

Instrumented Rocker-Bogie Chassis Design and Net Traction Estimation

by

Timothy P. Setterfield

A Thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfilment of
the requirements for the degree of
Master of Applied Science

Ottawa-Carleton Institute for
Mechanical and Aerospace Engineering

Department of Mechanical and Aerospace Engineering

Carleton University

Ottawa, Ontario, Canada

September 2011

Copyright ©

2011 - Timothy P. Setterfield



Library and Archives
Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*
ISBN: 978-0-494-83070-3
Our file *Notre référence*
ISBN: 978-0-494-83070-3

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

The undersigned recommend to
the Faculty of Graduate Studies and Research
acceptance of the Thesis

**Instrumented Rocker-Bogie Chassis Design and Net Traction
Estimation**

Submitted by **Timothy P. Setterfield**
in partial fulfilment of the requirements for the degree of
Master of Applied Science

Dr. Alex Ellery, Supervisor

Dr. Metin Yaras, Department Chair

Carleton University

2011

Abstract

This thesis presents the mechanical design of a rocker-bogie mobility system for the Canadian Space Agency's 30 kg *Kapvik* micro-rover. The design of the wheel drive system, a planetary differential mechanism, the integration of single-axis force sensors above the wheel hubs, structural analysis of the rocker and bogie links, and the cross-hill and uphill-downhill static stability are outlined.

The development of a net traction estimation algorithm is also presented. The algorithm uses multiple Unscented Kalman Filters to estimate drawbar pulls, resistive torques, wheel normal loads, wheel slips, wheel-terrain contact angles, and several other states in normal operation using on-board rover sensors. Polynomial fits to the estimated data are used to successfully recreate models of drawbar pull and resistive torque as functions of normal load and slip. A two-dimensional multibody dynamic simulation of *Kapvik* traversing rolling terrain is used to validate the efficacy of the algorithm.

Acknowledgments

I would like to thank my thesis supervisor, Dr. Alex Ellery. His ability to secure a contract with the Canadian Space Agency for the construction of a micro-rover allowed my Master's to not be just an intellectual exercise, but a project with a physical product. His unwavering confidence in the rover team inspired us to do our best work. The guidance Alex provided was specific enough to provide me with ideas, but open-ended enough to allow me to take those ideas in the direction of my choice.

I would like to thank my close friends and colleagues Rob Hewitt and Marc Gallant. Pursuing my Master's parallel to these two has been a great experience. Bouncing ideas back and forth in the lab, in meetings, or over pints really brought out my best work for both the *Kapvik* chassis and my thesis.

I would like to extend my gratitude to Ala' Qadi. Ala', as the project manager of *Kapvik*, had the tireless task of keeping a group of intellectuals on task and on schedule. Additionally, Ala' shielded the rover team from the mountains of documentation inherent in a project like *Kapvik*, allowing us to focus on the technical work.

I would like to thank the remainder of the Carleton *Kapvik* team: Brian Lynch, Matt Cross, Adam Mack, Cameron Frazier, Chris Nicol, Jesse Hiemstra, Adam Brierley, Amy Deeb, and Javier Romualdez. Carleton's contribution to *Kapvik* was a true team effort, and our accomplishments are the result of hard work and dedication.

I would like to acknowledge the fantastic work done at the Science and Technology Centre at Carleton, which manufactured the parts for *Kapvik*'s chassis. Steve Tremblay, Graham Beard, and Ron Roy created amazingly accurate parts; they also provided numerous suggestions for improvement to the *Kapvik* chassis design. Despite the difficulties associated with machining my parts they looked at my design with an open mind and took it as a challenge to create something unique. Without their positive attitude and skill, the *Kapvik* chassis would not have come to fruition.

I would like to thank Dr. Jo Wong. In addition to developing the theories of terramechanics utilized in this thesis he was of personal help to me: allowing me to take a four day mini-course at Carleton University free of charge; and responding to numerous emails to help me with the terramechanics aspect of my thesis work. Late in my Master's, a donation by Dr. Wong funded the creation of a mobile robotics laboratory in his name. It was in this laboratory that *Kapvik* took its first drive.

I would like to acknowledge some standout professors that either taught me or personally helped me in my research: Dr. Joshua Marshall, whose incredible course in Guidance, Navigation and Control provided me with a thorough understanding of estimation; Dr. Bruce Burlton, whose course in Space Mission Analysis and Design was as fascinating as it was informative; Dr. Mark Lanthier, whose Computer Science class in Mobile Robot Programming introduced me to the challenging but rewarding nature of robotics; and Dr. Mojtaba Ahmadi, who provided me with excellent guidance on motor selection and multibody dynamics simulation.

Finally, I would like to thank my family: my parents Tom and Penny, and my brother Jeremy. Your support throughout the completion of my Master's has been invaluable. I am particularly grateful to my father, who proofread my work on several occasions despite the large disparity between his research interests and mine.

Table of Contents

Abstract	iii
Acknowledgments	iv
Table of Contents	vi
List of Tables	xi
List of Figures	xii
Nomenclature	xv
1 Introduction	1
1.1 Problem Statement and Motivation	1
1.2 Purpose and Approach	3
1.3 Previous Work	4
1.3.1 Terramechanics	4
1.3.2 Rocker-Bogie Mobility System	5
1.3.3 Rover Multibody Dynamics	5
1.3.4 Soil Property and Net Traction Estimation	6
1.4 Outline	8

2	Terramechanics	9
2.1	Bekker’s Terramechanics Formulae	9
2.2	Wong’s Terramechanics Formulae	12
2.3	Wheel Dimensions	17
2.4	Soil Properties	17
2.5	Numerical Evaluation of Traction Parameters	18
2.5.1	Slip	19
2.5.2	Stress Distribution	19
2.5.3	Normal Force Equilibrium	20
2.5.4	Drawbar Pull and Resistive Torque	22
2.5.5	Traction Parameters for a Skidding Wheel	22
2.5.6	Traction Parameters Summary	24
2.6	Equation Simplification	24
3	<i>Kapvik</i> Chassis Design	29
3.1	Mobility System Overview	29
3.2	Wheel Drive System	31
3.3	Differential Mechanism	36
3.4	Force Sensor Integration	41
3.5	Structural Analysis	43
3.6	Static Stability	44
3.6.1	Cross-Hill Static Stability	45
3.6.2	Uphill-Downhill Static Stability	47
4	Multibody Dynamics	51
4.1	Lagrangian Mechanics	51
4.2	Two-Dimensional Rocker-Bogie Simulation	56
4.2.1	Co-ordinates, Naming Convention, and Configuration	57

4.2.2	Formulation of Dynamic Equations	58
4.2.3	Applied Forces and Torques	65
4.2.4	Simulation Implementation	68
4.2.5	Simulation Parameters	69
4.2.6	Simulation Results	71
5	Net Traction Estimation	73
5.1	Optimal State Estimation	73
5.1.1	Unscented Transform	74
5.1.2	Unscented Kalman Filter	75
5.2	Estimator Outline	78
5.3	Sensors	79
5.3.1	Motor Encoders	80
5.3.2	Bogie Angle Measurement	81
5.3.3	Force Sensors	82
5.3.4	Inertial Measurement Unit	83
5.3.5	Velocimeter	83
5.3.6	Sensor Measurement Vector and Noise Summary	85
5.4	Estimator 1: The General Estimator	86
5.4.1	States and Process Model	86
5.4.2	Measurements	90
5.4.3	Measurement Model	92
5.5	Estimator 2: Wheel Force Estimator with Two-Axis Force Sensors . .	93
5.5.1	States and Process Model	93
5.5.2	Measurements	94
5.5.3	Measurement Model	96
5.6	Estimator 3: Wheel Force Estimator with Single-Axis Force Sensors .	97

5.6.1	States and Process Model	97
5.6.2	Measurements	97
5.6.3	Measurement Model	99
5.7	Creating the Net Traction Model	99
5.8	Simulation Results and Discussion	101
5.9	Proposed Usage of Estimator	108
6	Conclusions and Recommendations	110
6.1	Summary of Contributions	111
6.2	Recommendations for Future Work	111
	List of References	114
	Appendix A Polynomial Traction Approximations	118
A.1	Ding Soil Simulant	118
A.2	Lunar Soil	119
A.3	Dry Sand	121
	Appendix B <i>Kapvik</i> Chassis Design Documentation	123
B.1	Overall Dimensions	123
B.2	Rocker Assembly	124
B.3	Bogie Assembly	125
B.4	Bogie Joint	125
B.5	Wheel	126
B.6	Preliminary Testing	127
	Appendix C Dynamic Simulation Variables	129
C.1	Common Functions	129
C.2	World Functions	131

C.3	Link Matrix	132
C.4	Joint and Link Positions	132
C.5	Wheel Force Jacobian	133
C.6	Link Jacobian	134
C.7	System Inertia Matrix	135
C.8	Time Derivative of the System Inertia Matrix	137
C.9	Rover Constraints	139
C.10	Jacobian of Constraints with respect to Configuration	139
C.11	Time Derivative of the Jacobian of Constraints with respect to Configuration	140
C.12	Derivative of Kinetic Energy with respect to Configuration	141
C.13	Derivative of Potential Energy with respect to Configuration	145
Appendix D Observability		146
D.1	Estimator 1: The General Estimator	147
D.2	Estimators 2 and 3: Wheel Force Estimators	150

List of Tables

2.1	Wheel dimensions.	17
2.2	Soil properties.	18
2.3	Treatment of different slip/skid cases.	20
3.1	Maxon RE25 motor specifications.	32
3.2	Wheel drive gear train specifications.	34
3.3	Specifications of <i>Kapvik</i> 's planetary gears.	39
3.4	Permissible off-axis loading for the THK RSR9ZM linear motion guide.	42
4.1	Parameters used in simulation.	70
5.1	Sensor measurement noises.	85
5.2	Coefficients of the 2 nd order polynomial true local model and 2 nd order polynomial fit to the estimated data.	106
A.1	Coefficients of the 4 th order polynomial model of drawbar pull DP and resistive torque τ_R for Ding et al.'s soil simulant.	118
A.2	Coefficients of the 4 th order polynomial model of drawbar pull DP and resistive torque τ_R for lunar soil.	119
A.3	Coefficients of the 4 th order polynomial model of drawbar pull DP and resistive torque τ_R for dry sand.	121

List of Figures

1.1	Jet Propulsion Laboratory’s Mars rovers.	6
2.1	Wheel co-ordinates, kinematic values, and dynamic values.	10
2.2	Co-ordinates and terms involved in wheel slip (left). Soil flow zones and angle of maximum normal stress (right).	15
2.3	An example stress distribution using the <i>Kapvik</i> wheel dimensions from Table 2.1, the soil properties from Table 2.2, a wheel-soil contact angle θ_0 of 45° , and a slip ι of 0.25.	21
2.4	The relationship between drawbar pull DP (left), resistive torque τ_R (right), and slip/skid ι for normal load $W = 50$ N.	25
2.5	Net traction parameters as functions of W and ι and their corresponding 4 th order polynomial approximations.	28
3.1	<i>Kapvik</i> rocker-bogie mobility system.	30
3.2	An exploded view of <i>Kapvik</i> ’s wheel drive system.	32
3.3	<i>Kapvik</i> wheel drive system operation.	35
3.4	General planetary gear.	37
3.5	Exploded views of left differential gearing system (top) and right differential gearing system (bottom).	40
3.6	Load cell and linear guide rail assembly (left) and linear guide rail off-axis forces and moments (right).	41

3.7	Rocker finite element analysis simulation (top) and bogie finite element analysis simulation (bottom).	44
3.8	Co-ordinates and variables involved in rover static stability.	45
3.9	Critical cross-hill stability angles.	46
3.10	Wheel vertical reaction forces on an uphill-downhill slope where the rover's center of gravity is at the rover centroid.	49
3.11	Critical uphill-downhill stability angles.	50
4.1	Two-dimensional rocker-bogie system configuration variables, vectors, and forces.	58
4.2	Wheel constraints.	63
4.3	Block diagram of simulation implementation.	69
4.4	Results from the 200 s simulation of the rover traversing gently rolling terrain.	71
4.5	Configuration results from the 200 s simulation of the rover traversing gently rolling terrain.	72
5.1	Block diagram of information flow through the estimator.	79
5.2	Sensor measurements.	80
5.3	Maxon magneto-resistant motor encoder.	81
5.4	Treatment of quantization noise as a normal distribution.	81
5.5	Bogie angle measurement.	82
5.6	Force sensors.	83
5.7	Memsense H3-IMU HP02-0300 inertial measurement unit.	84
5.8	Velocimeter principles of operation.	84
5.9	Forces applied at the wheel centroid shown in sensor directions.	98
5.10	Estimated resistive torques τ_R and drawbar pulls DP over the 200 s simulation.	103
5.11	Estimated normal loads W and slips i over the 200 s simulation.	104

5.12	Resistive torque data and polynomial fits using Estimators 1 and 2. . .	106
5.13	Drawbar pull data and polynomial fits using Estimators 1 and 2. . . .	107
5.14	Drawbar pull data and polynomial fits using Estimators 1 and 3. . . .	107
A.1	Net traction parameters as a function of W and ι , and their corresponding 4 th order polynomial approximation for lunar soil.	120
A.2	Net traction parameters as a function of W and ι , and their corresponding 4 th order polynomial approximation for dry sand.	122
B.1	Overall <i>Kapvik</i> chassis dimensions.	123
B.2	An exploded view of the <i>Kapvik</i> rocker assembly.	124
B.3	An exploded view of the <i>Kapvik</i> bogie assembly.	125
B.4	Exploded views of the <i>Kapvik</i> bogie joint.	126
B.5	The <i>Kapvik</i> wheel.	127
B.6	Photographs of the <i>Kapvik</i> micro-rover during outdoor testing.	128

Nomenclature

This thesis uses SI units throughout: kg, m, s.

Vectors and matrices are in boldface \mathbf{v} , \mathbf{M} .

The mean of a variable is denoted \bar{x} .

The standard deviation of a random variable is denoted σ_x .

Estimated variables are denoted \hat{x} .

Measured variables are denoted \check{x} .

Expected measurements are denoted $\hat{\hat{x}}$.

The first time derivative is denoted \dot{x} .

The second time derivative is denoted \ddot{x} .

Revolutions per minute is denoted rpm.

1×10^{-3} Newton-meters is denoted mNm.

Variables

Variable	Variable Meaning
$\mathbf{0}$	Zero vector or zero matrix
$\mathbf{a}_1, \mathbf{a}_2$	Constant vectors
\mathbf{a}, \mathbf{b}	Arbitrary vectors of dimension 3×1
a, b	Rocker-bogie characteristic lengths
α, β	Baumgarte stabilization constants
b_d	Wheel viscous damping coefficient

b_w	Wheel width
\mathbf{c}_{ii}	Vector from joint i to the associated link's center of mass
C, C_L, C_T	Linear guide rail dynamic load rating, reverse dynamic load rating, and lateral dynamic load rating
c	Soil cohesion
c_1, c_2	Soil constants for evaluation of the angle of maximum normal stress
dir	Direction in which to exert drawbar pull and resistive torque
DP	Drawbar pull
DP_1, DP_2, DP_3	Drawbar pulls for wheels 1, 2, and 3
\mathbf{E}	Observability matrix
$\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3$	Observability matrices for Estimators 1, 2, and 3
$e1, e2, e3$	Wheel-terrain contact points for wheels 1, 2, and 3
$\epsilon_1, \epsilon_2, \epsilon_3$	Angle between world vertical and primary sensing axes for force sensors 1, 2, and 3
$\mathbf{F}_{DP}, \mathbf{F}_{\tau R}$	Vector of external forces and moments on the wheel joints resultant from drawbar pull and resistive torque respectively
\mathbf{F}_{ext}	Vector of external forces and moments on the wheel joints
$\mathbf{F}_{w1}, \mathbf{F}_{w2}, \mathbf{F}_{w3}$	External force vectors exerted on wheel joints for wheels 1, 2, and 3
f	Discrete-time process model
f_1, f_2, f_3	Discrete-time process models for Estimators 1, 2, and 3
F_{L1}, F_{L2}, F_{L3}	Forces in the primary sensing axis above wheels 1, 2, and 3
F_{11}, F_{12}, F_{13}	Forces in the secondary sensing axis above wheels 1, 2, and 3
F_{112}	The sum of the force above wheels 1 and 2 in the secondary sensing axis
F_{xfw}, F_{zfw}	Net forces in the world horizontal and vertical directions respectively, from external wheel forces

Φ	Rover generalized configuration vector
$\dot{\Phi}$	Rover generalized velocity vector
Φ_s	Rover configuration state vector containing the generalized velocity vector and the configuration vector
ϕ_s	Soil internal angle of friction
g	Local gravitational acceleration
$\gamma_1, \gamma_2, \gamma_3$	Wheel-terrain contact angles for wheels 1, 2, and 3
γ_b	Bogie stability angle
γ_{ch}, γ_{ud}	Critical cross-hill and uphill-downhill static stability angles
\mathbf{H}	System inertia matrix
H	Wheel thrust
h	Measurement model
h_1, h_2, h_3	Measurement models for Estimators 1, 2, and 3
h_r	Height of rover cab centroid
\mathbf{I}	Inertia tensor
$\mathbf{I}_0, \mathbf{I}_1, \mathbf{I}_2, \mathbf{I}_3, \mathbf{I}_4$	Inertia tensors for bodies 0, 1, 2, 3, and 4
$\mathbf{I}_{q \times q}$	Identity matrix of dimensions $q \times q$
v	Slip or skid
v_1, v_2, v_3	Wheel slip(s)/skid(s) for wheels 1, 2, and 3
I_w	The wheel and gear inertia
I_{wg}	The wheel and gear inertia referred to the wheel
\mathbf{J}_a	Jacobian of link centers of mass's linear and angular velocities with respect to the generalized velocity vector
\mathbf{J}_v	Jacobian for linear velocity of body centers of mass with respect to the generalized velocity vector

$\mathbf{J}_{v_0}, \mathbf{J}_{v_1}, \mathbf{J}_{v_2}, \mathbf{J}_{v_3}, \mathbf{J}_{v_4}$	Jacobian for linear velocity of body centers of mass with respect to generalized velocity vector for bodies 0, 1, 2, 3, and 4
\mathbf{J}_w	Jacobian of generalized forces with respect to external wheel forces
\mathbf{J}_ω	Jacobian for angular velocity of bodies with respect to the generalized velocity vector
$\mathbf{J}_{\omega_0}, \mathbf{J}_{\omega_1}, \mathbf{J}_{\omega_2}, \mathbf{J}_{\omega_3}, \mathbf{J}_{\omega_4}$	Jacobian for angular velocity of bodies 0, 1, 2, 3, and 4 with respect to the generalized velocity vector
j	Shear displacement of the soil
\mathbf{K}	Kalman gain
k	Index of current simulation timestep
k_1, k_2	Soil pressure-sinkage constants
k_c	Soil modulus of cohesion
k_ϕ	Soil modulus of friction
K_p, K_i, K_d	Proportional, integral, and derivative control gains
K_s	Soil shear deformation parameter
\mathbf{L}	Matrix of zeros and ones describing the kinematic chain of the rover
\mathbf{l}_{ij}	Vector from point i to point j on the rover
L	Lagrangian
l_f	Length of footprint
λ	Lagrange multipliers
$\mathbf{M}_{w_1}, \mathbf{M}_{w_2}, \mathbf{M}_{w_3}$	External moment vectors exerted on wheel joints for wheels 1, 2, and 3
M_{0fw}, M_{1fw}	Net moments about combined body/rocker center of mass and bogie joint respectively, from external wheel forces
m_0, m_1, m_2, m_3, m_4	Masses of bodies 0, 1, 2, 3, and 4

M_A, M_B, M_C	Linear guide rail moment ratings
m_r	Total rover mass
$\mathbf{N}_1, \mathbf{N}_2, \mathbf{N}_3, \mathbf{N}_4$	Normal vector along axis of rotation of joints 1, 2, 3, and 4
n	Number of degrees of freedom of multibody system
n_b	Number of bodies
n_c	Number of constraints
N_{e1}, N_{e2}, N_{e3}	Vertical reaction forces at the wheel-terrain contact points $e1, e2, e3$
n_s	Soil deformation exponent
n_v	Number of joints
n_x	Number of states
O	Order of polynomial fit
\mathbf{P}	State covariance matrix
$\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4$	Position of joints 1, 2, 3, and 4 in the world frame
\mathbf{P}_{aa}	Covariance of vector \mathbf{a}
\mathbf{P}_{ab}	Cross-covariance between vectors \mathbf{a} and \mathbf{b}
$\mathbf{p}_{c1}, \mathbf{p}_{c2}, \mathbf{p}_{c3}, \mathbf{p}_{c4}$	Position of center of mass of links 1, 2, 3, and 4 in the world frame
p_{kl}	Coefficient of the term $i^k W^l$ in a polynomial approximation
\mathbf{Q}	Process noise covariance
q_1, q_2, q_3, q_4	Joint angles for joints 1, 2, 3, and 4
θ	Angle along the wheel rim
θ_0	Total wheel-soil contact angle
$\theta_{0x}, \theta_{0y}, \theta_{0z}$	Rover body orientation in world frame
θ_m	Angle of maximum normal stress along the wheel rim
\mathbf{r}_0	Position of lumped body/rocker center of mass in world frame

r_{0i}	Vector from lumped body/rocker center of mass to center of mass of link i in the world frame
r_w	Wheel radius vector from wheel centroid to wheel-soil contact point
R_y	Measurement noise covariance
R_c	Wheel compaction resistance
r_P, r_A, r_S	Radius of the planet gears, the annular gear, and the sun gear
r_w	Wheel radius
s_1, s_2	Roots of a characteristic equation
σ	Soil normal stress, or standard deviation of a random variable
σ_1, σ_2	Normal stress in forward and rearward regions respectively
σ_B	Bekker normal stress
σ_m	Maximum normal stress under the wheel rim
T	Kinetic energy
t	Time
T_e	Estimator timestep period
τ_a	Generalized applied forces acting on the multibody system
τ_c	Generalized constraint forces acting on the multibody system
τ_g	Generalized forces acting on the multibody system
$\tau_R, \tau_W, \tau_{DP}$	Generalized forces resultant from resistive torques, wheel torques, and drawbar pulls respectively
τ_B	Bekker shear stress
τ_R	Magnitude of the resistive torque of the soil on the wheel
$\tau_{R_1}, \tau_{R_2}, \tau_{R_3}$	Magnitude of the resistive torques on wheels 1, 2, and 3
τ_{rpt}	Harmonic Drive repeated peak torque rating
τ_s	Soil shear stress
τ_{s_1}, τ_{s_2}	Soil shear stress in forward and rearward regions respectively

τ_W	Motor torque referred to the wheel
$\tau_{W_1}, \tau_{W_2}, \tau_{W_3}$	Motor torque referred to the wheel for wheels 1, 2, and 3
τ_{Wmax}	Maximum attainable wheel torque
\mathbf{u}	Control vector
\mathbf{v}	Normally distributed measurement noise
$\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4$	Center of mass velocities for bodies 0, 1, 2, 3, and 4
$\mathbf{v}_a, \mathbf{v}_{a_n}, \mathbf{v}_{a_t}$	Soil particle velocity vector, and its radial and tangential components respectively
\mathbf{v}_j	Soil slip velocity vector
\mathbf{v}_w	Wheel centroid velocity vector
V	Potential energy
v	Velocity magnitude
v_d	Desired rover velocity
v_w	Wheel centroid velocity magnitude
\mathbf{w}	Normally distributed process noise
W	Normal load
W_1, W_2, W_3	Normal loads for wheels 1, 2, and 3
w_f	Width of footprint
$\boldsymbol{\omega}$	Angular velocity vector
$\boldsymbol{\omega}_0, \boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \boldsymbol{\omega}_3, \boldsymbol{\omega}_4$	Angular velocity vector for bodies 0, 1, 2, 3, and 4
ω	Angular velocity magnitude
$\omega_C, \omega_P, \omega_A, \omega_S$	Angular velocities of the carrier, the planet gears, the annular gear, and the sun gear
$\omega_{P/C}, \omega_{A/C}, \omega_{S/C}$	Angular velocities of the planet gears, the annular gear, and the sun gear with respect to the carrier
\mathbf{X}	Sigma points

$\dot{\mathbf{X}}_a$	Velocity vector containing the velocities of the centers of mass of the links
\mathbf{x}	State vector
$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$	State vectors for Estimators 1, 2, and 3
x_0, y_0, z_0	Lumped body/rocker center of mass co-ordinates in world frame
$\ddot{x}_0, \ddot{y}_0, \ddot{z}_0$	Lumped body/rocker center of mass acceleration in world frame
$\dot{x}_0^R, \dot{y}_0^R, \dot{z}_0^R$	Lumped body/rocker center of mass velocity in rover frame
$\ddot{x}_0^R, \ddot{y}_0^R, \ddot{z}_0^R$	Lumped body/rocker center of mass acceleration in rover frame
$x_{cg}^R, y_{cg}^R, z_{cg}^R$	Rover center of gravity co-ordinates in rover frame
X^R, Y^R, Z^R	Axes of the rover co-ordinate frame
x_t, y_t, z_t	Wheel-terrain contact co-ordinates in world frame
X_w, Y_w, Z_w	Axes of the world co-ordinate frame
x_{wc}, y_{wc}, z_{wc}	Wheel centroid path co-ordinates in world frame
\mathbf{Y}	Sigma points
\mathbf{y}	Measurement vector
$\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3$	Measurement vectors for Estimators 1, 2, and 3
Ψ	Motion constraints to the multibody system
Ψ_t	Derivative of motion constraints with respect to time
Ψ_{Φ}	Jacobian of motion constraints with respect to configuration vector
\mathbf{z}	Sensor measurement vector
z_0, z_1, z_2, z_3, z_4	Heights of bodies 0, 1, 2, 3, and 4 in world frame
z_s	Sinkage depth

Chapter 1

Introduction

1.1 Problem Statement and Motivation

Early exploration of planetary surfaces was performed predominantly by stationary landers. Although providing invaluable information about the area in their immediate vicinity, their immobility placed major constraints on the scientific return of these missions. The Russian Lunakhod rovers, the first of which landed on the moon in 1970, began the mission model of using unmanned robotic vehicles for space exploration. After a long hiatus, NASA's Mars Pathfinder mission landed on Mars with its rover, Sojourner, in 1997; this prompted a resurgence in research and interest in planetary rovers. Sojourner was the first planetary exploration rover to use the rocker-bogie mobility system, a spring-less suspension system designed to equilibrate ground pressure on all of its six wheels. In 2004, two NASA-built Mars Exploration Rovers, Spirit and Opportunity, landed on Mars [1]. Their planned three month mission was successful beyond all expectations: Spirit had a mission duration of over six years; Opportunity is still operational at the time of writing. The Mars Exploration Rovers, as well as the Mars Science Laboratory rover, Curiosity, all utilize the rocker-bogie mobility system, indicating NASA's continued confidence in the design.

Long time delays in transmissions and limited communication windows have motivated a trend toward increased levels of autonomy in planetary rover research. This trend has extended to many facets of rover design and operation: path planning, localization, mapping, acquisition of scientific targets, traversability analysis, traction control, and soil property estimation. Planetary terrain tends to be soft and sandy, making it easy for the rover to slip or become stuck; because of this, the fields of traversability analysis, traction control, and net traction estimation are particularly important to the continued operation of the rover. Slip decreases the efficiency with which the rover operates and in the worst case causes the rover to dig itself into a rut and become stuck. Large slips were frequently reported on the Mars Exploration Rovers, particularly when climbing inclines [2,3]. The Spirit rover eventually succumbed to a traction-related failure, becoming stuck in soft sand and unable to escape [4]. Optimal control of a rover for trafficability or power consumption is accomplished through the favourable distribution of forces amongst the rover wheels [5]. In order to apply the optimal distribution of forces it is necessary to model the rover's net traction relationships; the rover's net traction relationships are the net moment, termed resistive torque, and the net forward force, termed drawbar pull, applied to a wheel by the soil under specific operating conditions. For wheels driving on deformable terrain, there is no proportional relationship between the applied wheel torque and the force delivered by the terrain. Additionally, the force delivered by the terrain is sensitive to normal load and slip in a non-linear manner. Thus, in order to implement proper traction control there is a need to estimate the net traction relationships; it is preferable that this be accomplished using on-board sensors, since conventional methods require several instruments that are unlikely to be included on planetary exploration missions because of their additional mass and relatively low utility.

1.2 Purpose and Approach

The purpose of this thesis is to develop a method to perform net traction estimation using the rocker-bogie mobility system. This includes the mechanical design of an instrumented rocker-bogie mobility system, or “chassis”, as well as the development of a net traction estimation algorithm.

The Canadian Space Agency has recently invested heavily in the development of rover technologies. *Kapvik*, a 30 kg terrestrial prototype for a planetary micro-rover, was recently developed collaboratively by MPB Technologies, Carleton University, Ryerson University, the University of Toronto, Xiphos Technologies, MDA Space Missions, and the University of Winnipeg for the Canadian Space Agency. Carleton University’s role was to design navigation software, the rover’s cab, the rover’s avionics box, and the rover’s chassis. The assignment of the design of the rover’s chassis to the author provided a unique opportunity for the addition of sensors advantageous to net traction estimation. In addition to the typical suite of sensors included on a planetary rover, single-axis force sensors above the wheel hubs and a vision-based velocimeter were desired. Although the velocimeter was not included in the final design, its possible addition to future versions of *Kapvik* would not be problematic; the force sensors, the addition of which required extensive alterations to the typical rocker-bogie design, were included. At the time of writing, the chassis has been fully manufactured and has undergone assembly and preliminary testing.

In order to develop and test a net traction estimation algorithm without access to a real rover, a two-dimensional simulation environment was developed for a rocker-bogie mobility system. Since the most challenging environment in terms of traction is rolling terrain – not flat ground – the rover was simulated in the world horizontal and vertical directions, as opposed to from above. Fully dynamic two-dimensional multibody simulation is developed and performed. The wheel-terrain interaction forces and

moments are calculated using Wong’s terramechanics equations and incorporated into the dynamic simulation. A simplified but accurate polynomial approximation to the terramechanics equations is used to both increase the speed of the simulation, and as a means to reconstruct the net traction models through estimation. A set of three Unscented Kalman Filters are developed to perform net traction estimation; the use of both two-axis force sensors and single-axis force sensors (as are installed on *Kapvik*) above the wheel hubs is considered.

1.3 Previous Work

Previous work has been performed in all of the disciplines enlisted in this thesis. This section outlines previous work directly applicable to this thesis.

1.3.1 Terramechanics

Terramechanics, the study of the interaction of wheels and tracks with the ground, was pioneered by Bekker in the 1950’s. Bekker’s theories provided a means to analyze the steady-state motion of tracks and wheels on deformable terrain. Typically this analysis was used on large military and agricultural vehicles [2, 6], but it has found recent utility in analyzing the traction of planetary rovers. Wong refined Bekker’s theories to incorporate the effect of slip. Particularly important were his 1967 papers in which he used soil-wheel stresses to predict rigid wheel performance in deformable terrain [7, 8]. Despite recent interest in the field, there have not been any significant advancements in terramechanics since Wong’s improvements in the late 1960’s. Computationally heavy discrete element methods have been getting some attention for their potential to more accurately reproduce experimental results [9]. However, Bekker and Wong’s steady-state terramechanics equations are still typically used for analysis of tracked and wheeled vehicles on deformable terrain [2, 5, 10–12]; Wong’s

terramechanics equations are used in this thesis. Ding et al. recently published a thorough experimental study of rigid wheels driving through lunar soil simulant [12]. The soil properties as well as additional insights from this study are used in this thesis.

1.3.2 Rocker-Bogie Mobility System

The rocker-bogie articulated mobility system was invented at the Jet Propulsion Laboratory by Donald Bickler in 1989 [13] and is currently NASA's favoured planetary rover design. Thorough testing and validation of the mobility system has been performed both on Earth and on Mars [1]. In recent years the trend has been toward the development of larger rovers, as illustrated in Figure 1.1. The rocker-bogie mobility system was chosen for *Kapvik* because of its excellent flight heritage and obstacle negotiation capabilities; with reference to Figure 1.1, the size of *Kapvik* is between that of Sojourner (which is 0.65 m long) and the Mars Exploration Rover (which is 1.6 m long).

1.3.3 Rover Multibody Dynamics

Analysis of a rover with a rocker-bogie mobility system is a multibody dynamics problem. Several investigations into the kinematics and dynamics of articulated mobile robots have been performed [14–17]. When the accurate dynamic simulation of the rover is not the primary objective, such as when it is being used to test traction control, a quasi-static model is commonly used; this is the case for the research performed by both Hacot and Iagnemma [15, 16]. More recently, Ding et al. used Lagrangian multibody dynamics to create a high-fidelity simulation of an articulated mobile robot [18]. Extending multibody simulation techniques originally developed

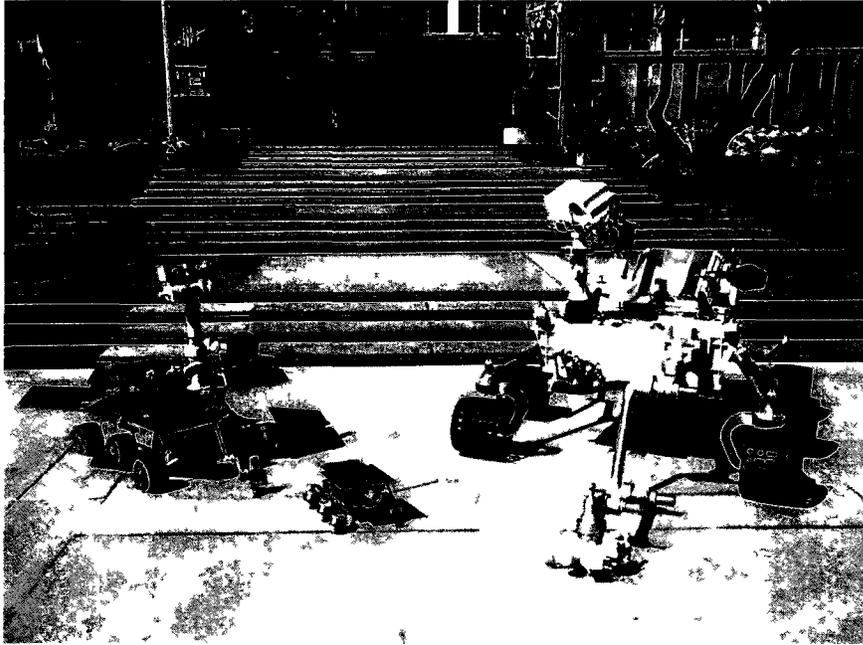


Figure 1.1: Jet Propulsion Laboratory’s Mars rovers. Mars Exploration Rover (left), Sojourner (middle), Mars Science Laboratory (right). All three rovers use the rocker-bogie mobility system.

for manipulators, Ding et al. were able to simulate the rover as an articulated multi-body system with a moving base. Through use of Wong’s terramechanics equations, an accurate simulation environment was created and experimentally verified [18]. A similar method to that developed by Ding et al. is used in this thesis to create a fully dynamic multibody rover simulation in two dimensions.

1.3.4 Soil Property and Net Traction Estimation

Preliminary estimates of the Martian soil properties were performed on the Viking missions. A small trenching experiment using a backhoe enabled the estimation of two soil properties: soil cohesion c and soil internal angle of friction ϕ_s [19]. This was a dedicated instrument that added mass to the overall payload, and could only determine the soil properties in the immediate vicinity of the landing site.

Estimation of soil properties using on-board rover sensors was performed on the Mars Pathfinder rover Sojourner in 1997 [20, 21]. The Sojourner rover was commanded to hold five wheels stationary whilst moving the sixth [21]. The wheel torque, obtained from the moving wheel's motor current, was used to deduce the soil shear stress τ_s ; several techniques were then used to obtain an estimate of soil cohesion c and soil internal angle of friction ϕ_s [20]. This experiment was outside of the normal operation of the rover, meaning that it consumed additional time and power. As is shown in Chapter 2, cohesion c and internal friction angle ϕ_s are just two of numerous soil parameters required to fully characterize the net traction relationships. Predicting the net traction given only these two parameters requires that representative values be chosen for the remainder of the soil properties.

An on-line, linear least-squares method for estimating the same two soil parameters, cohesion c and internal friction angle ϕ_s , was presented by Iagnemma et al. [10] using a simplified stress distribution [22] and the assumption that wheel sinkage can be measured. Measurement of the sinkage angle was later shown to be possible using a camera to image the wheel-terrain interface [3]. Iagnemma's technique does not require that the rover stop to perform a dedicated experiment; however, the downfalls of estimating only two of the many soil properties again apply.

The net traction relationships estimated in this thesis, resistive torque τ_R and drawbar pull DP , were estimated by Ray et al. in deformable terrain on an unarticulated four wheeled mobile robot using an Extended Kalman-Bucy filter and a fifth unpowered wheel to measure velocity [11]. Normal loads W were not sensed directly but instead calculated using sensed accelerations.

1.4 Outline

This thesis consists of six chapters and four appendices. This chapter provides an introduction to the problem that was undertaken. Chapter 2 reviews relevant terramechanics background, numerical evaluation of traction parameters, and develops a method for approximating the terramechanics equations using two-dimensional polynomial fits. Chapter 3 presents the design of *Kapvik*'s rocker-bogie mobility system together with supporting analysis. Chapter 4 outlines the use of Lagrangian mechanics for multibody dynamic simulation and its application to the *Kapvik* articulated rocker-bogie rover in two dimensions. Chapter 5 shows the development of the net traction estimation algorithm and presents simulated estimator results. Chapter 6 concludes the thesis, summarizing the main points and providing recommendations for future research in the field. Appendix A provides information on polynomial fitting of the net traction relationships. Appendix B contains additional details of the *Kapvik* chassis design. Appendix C outlines the calculation of dynamic simulation variables specific to the two-dimensional rover simulation. Finally, Appendix D proves the observability of the Unscented Kalman Filters using linearized process and measurement models.

Chapter 2

Terramechanics

This chapter provides an overview of both Bekker's and Wong's formulae. Bekker's formulae are briefly outlined to illustrate some basic concepts of terramechanics prior to refinement by Wong. The subscript B is used to indicate the Bekker version of formulae for which the Bekker and Wong forms vary. Wong's formulae are used for simulation in this thesis.

2.1 Bekker's Terramechanics Formulae

A diagram of the forces and stresses acting on a single rigid wheel driving in loose mineral terrain is shown in Figure 2.1. Terms in the following development adhere to the nomenclature outlined in this diagram. In this thesis, the wheel is assumed to lose contact with the terrain at a point directly below the wheel centroid. This approximation is commonly valid [7]. The x axis of the indicated co-ordinate system is parallel to the terrain; the z axis of the indicated co-ordinate system is perpendicular to the terrain.

A basic assumption made by Bekker in his derivation of wheel-soil interaction is that the normal stress acting radially at any point along the rim of the wheel is equal

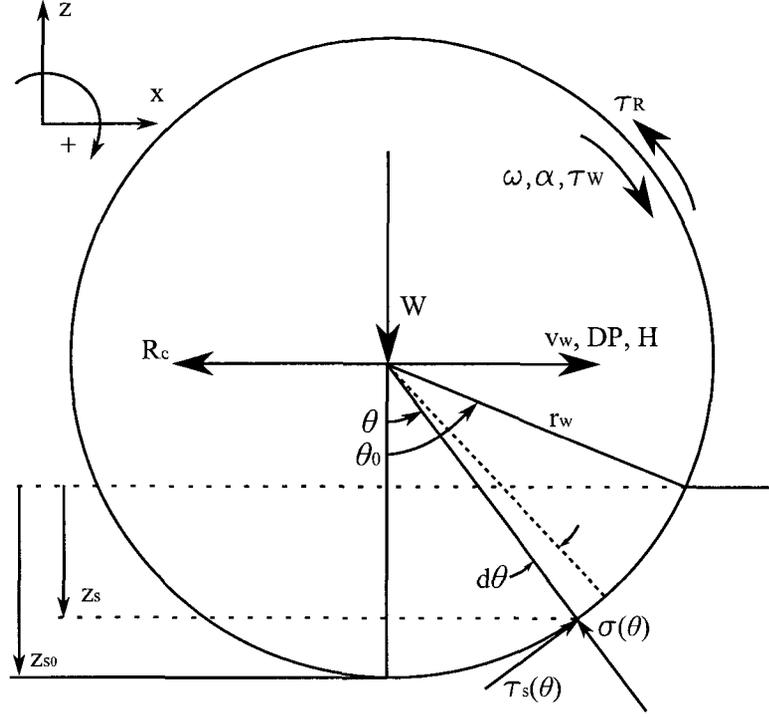


Figure 2.1: Wheel co-ordinates, kinematic values, and dynamic values. Adapted from [7].

to the pressure that would be exerted under a plate at the same depth [6]. This means that the normal stress at any angle θ between 0 and θ_0 along the wheel can be computed based on sinkage depth [6].

$$\sigma_B = \left(\frac{k_c}{b_w} + k_\phi \right) z_s^{n_s} = \left(\frac{k_c}{b_w} + k_\phi \right) [r_w (\cos \theta - \cos \theta_0)]^{n_s} \quad (2.1)$$

where σ_B is the Bekker normal stress, k_c is the soil modulus of cohesion, k_ϕ is the soil modulus of friction, b_w is the wheel width, z_s is the sinkage depth, n_s is the soil deformation exponent, which is usually close to one in sandy terrain, r_w is the wheel radius, θ is the angle along the wheel rim, where $\theta = 0$ directly below the wheel centroid in the z direction, and θ_0 is the total wheel-soil contact angle.

For loose mineral soil of the type experienced in many planetary environments, the shear stress τ_s developed by the soil is related to the displacement of the soil from its

original position. The relationship is given by [7]:

$$\tau_s = (c + \sigma \tan \phi_s) \left(1 - e^{-\frac{j}{K_s}}\right) \quad (2.2)$$

where c is the cohesion of the soil, σ is the normal stress acting on the soil, ϕ_s is the soil's internal angle of friction, j is the shear displacement of the soil, and K_s is the shear deformation parameter of the soil.

The maximum shear stress τ_{smax} of the soil is found by setting the shear displacement j to infinity.

$$\tau_{smax} = c + \sigma \tan \phi_s \quad (2.3)$$

Shear displacement is not properly considered in Bekker's formulation. It is assumed that the shear stress takes the maximum value τ_{smax} along the entire contact patch. Substituting Equation 2.1 into 2.3 the Bekker shear stress τ_B is given by:

$$\tau_B = \tau_{smax} = c + \left(\frac{k_c}{b_w} + k_\phi\right) [r_w (\cos \theta - \cos \theta_0)]^{n_s} \tan \phi_s \quad (2.4)$$

A reaction force is exerted by the ground to support the normal load W bearing down on the wheel. This is exerted by the components of normal and shear stress perpendicular to the ground. However, in Bekker's analysis, the perpendicular component of shear stress is neglected [7]. By force balance:

$$\begin{aligned} \sum F_z = 0 &= -W + \int_0^{\theta_0} b_w r_w \sigma_B \cos \theta d\theta \\ \sum F_z = 0 &= -W + b_w r_w \left(\frac{k_c}{b_w} + k_\phi\right) \int_0^{\theta_0} [r_w (\cos \theta - \cos \theta_0)]^{n_s} \cos \theta d\theta \end{aligned} \quad (2.5)$$

There are two contributors to the force parallel to the terrain: the parallel component of normal stress, which opposes motion, and the parallel component of shear stress, in the direction of motion. By integrating these stresses around the wheel rim, the

net force acting on the wheel can be found. The net parallel force is termed the drawbar pull DP . The resisting force is termed compaction resistance R_c , and the forward driving force is termed thrust H . The force balance parallel to the terrain is as follows:

$$\begin{aligned}
\sum F_x &= DP = H - R_c \\
DP &= \int_0^{\theta_0} r_w b_w \tau_B \cos \theta d\theta - \int_0^{\theta_0} r_w b_w \sigma_B \sin \theta d\theta \\
DP &= r_w b_w \int_0^{\theta_0} \left[c + \left(\frac{k_c}{b_w} + k_\phi \right) [r_w (\cos \theta - \cos \theta_0)]^{n_s} \tan \phi_s \cos \theta \right] d\theta \dots \\
&\dots - r_w b_w \int_0^{\theta_0} \left[\left(\frac{k_c}{b_w} + k_\phi \right) [r_w (\cos \theta - \cos \theta_0)]^{n_s} \sin \theta \right] d\theta \quad (2.6)
\end{aligned}$$

The magnitude of the resistive torque τ_R acting on the wheel using Bekker's formulae can be calculated by integrating the shear stress around the wheel rim:

$$\begin{aligned}
\tau_R &= \int_0^{\theta_0} r_w \tau_B b_w r_w d\theta \\
\tau_R &= b_w r_w^2 \int_0^{\theta_0} \left[c + \left(\frac{k_c}{b_w} + k_\phi \right) [r_w (\cos \theta - \cos \theta_0)]^{n_s} \tan \phi_s \right] d\theta \quad (2.7)
\end{aligned}$$

2.2 Wong's Terramechanics Formulae

The development of Wong's formulae is more involved because Wong considers the effect of wheel slip. Where appropriate, the same nomenclature as the previous section is used.

When a wheel's forward velocity v_w is less than the product of wheel radius and wheel angular velocity $r_w \omega$, the non-dimensional term slip i is used to describe the degree to which the wheel is slipping. When a wheel's forward velocity v_w is greater than the product of wheel radius and angular velocity $r_w \omega$, the non-dimensional term skid i is

used to describe the degree to which the wheel is skidding. The variable i is positive when the wheel is slipping and negative when the wheel is skidding [12].

$$i = 1 - \frac{v_w}{r_w \omega} \quad \text{when } |r_w \omega| \geq |v_w| \quad (\text{slip}) \quad (2.8)$$

$$i = \frac{r_w \omega}{v_w} - 1 \quad \text{when } |r_w \omega| < |v_w| \quad (\text{skid}) \quad (2.9)$$

This definition for skid is commonly used [2, 12, 23, 24] and was created so that slip i does not approach negative infinity for $\omega \rightarrow 0$. Note that all of the subsequent formulae in this section are for the case where the rover is slipping, not skidding (i.e. $|r_w \omega| \geq |v_w|$ and $i \geq 0$). The case of a skidding wheel is considered in Section 2.5.5.

In Wong's formulae, which are based largely on experimentation, there are two soil flow zones, each with different equations governing their normal stresses [7]. The forward flow zone is termed flow zone 1, and the rearward flow zone is termed flow zone 2. The maximum normal stress σ_m around the wheel rim was found to occur at the transition angle θ_m between the two flow zones. Experimental evidence suggests that the transition angle θ_m moves forward with increasing slip [7].

$$\theta_m = \theta_0 (c_1 + c_2 i) \quad (2.10)$$

where c_1 and c_2 are constants dependent on the type of soil.

The normal stress σ_1 in the forward flow zone (where $\theta_m \leq \theta \leq \theta_0$) is governed by an equation very similar to Equation 2.1:

$$\sigma_1 = (k_1 + k_2 b_w) \left(\frac{z_s}{b_w} \right)^{n_s} = (k_1 + k_2 b_w) \left[\frac{r_w}{b_w} (\cos \theta - \cos \theta_0) \right]^{n_s} \quad (2.11)$$

where k_1 and k_2 are pressure-sinkage constants.

By comparing Equation 2.11 to Equation 2.1, the constants k_c , k_ϕ , k_1 , and k_2 can be related in the following way:

$$k_1 = k_c b_w^{n_s - 1} \quad (2.12)$$

$$k_2 = k_\phi b_w^{n_s - 1} \quad (2.13)$$

In some sandy soils, where the soil deformation exponent n_s is one, the equations are identical, with $k_1 = k_c$ and $k_2 = k_\phi$. The normal stress distribution σ_2 in the rearward flow zone (where $0 \leq \theta < \theta_m$) was found to decrease toward $\theta = 0$ with the same shape as the curve from θ_m to θ_0 [7]. As a result, the stress distribution of the forward flow zone is scaled and reversed to form the stress distribution for the rearward region:

$$\sigma_2 = (k_1 + k_2 b_w) \left[\frac{r_w}{b_w} \left(\cos \left[\theta_0 - \frac{\theta}{\theta_m} (\theta_0 - \theta_m) \right] - \cos \theta_0 \right) \right]^{n_s} \quad (2.14)$$

Equation 2.2 shows that soil behaves elastically, and that non-zero soil displacement j is required in order to develop shear stress τ_s . The shear stress developed is zero if the soil is not displaced. The absolute displacement of the soil during a wheel passage can be found by considering the concept of slip velocity \mathbf{v}_j : the velocity with which the soil is slipping beneath the wheel. The absolute velocity of a particle of soil in contact with the wheel rim will be exactly the velocity of the wheel rim at that point [7]:

$$\mathbf{v}_a = \mathbf{v}_w + \boldsymbol{\omega} \times \mathbf{r}_w = \mathbf{v}_{a_n} + \mathbf{v}_{a_t} \quad (2.15)$$

where \mathbf{v}_a is the absolute velocity of the particle of soil, \mathbf{v}_w is the velocity of the wheel centroid, $\boldsymbol{\omega}$ is the angular velocity of the wheel, \mathbf{r}_w is the vector from the wheel centroid to the wheel-soil contact point being considered, \mathbf{v}_{a_n} is the radial component of \mathbf{v}_a , and \mathbf{v}_{a_t} is the tangential component of \mathbf{v}_a .

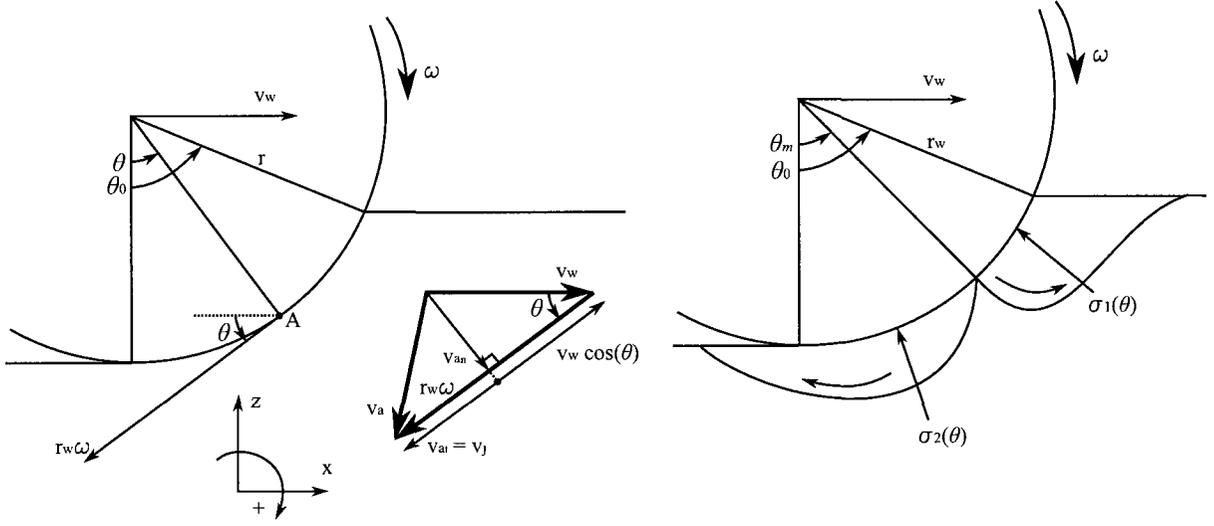


Figure 2.2: Co-ordinates and terms involved in wheel slip (left). Soil flow zones and angle of maximum normal stress (right).

The normal component v_{a_n} is the velocity at which the wheel is pushing into the soil and the tangential component v_{a_t} is the velocity of the sand tangential to the wheel. This is precisely the slip velocity (i.e. $v_j = v_{a_t}$). Referring to Figure 2.2, we have:

$$|v_{a_t}| = |v_j| = r_w \omega - v_w \cos \theta \quad (2.16)$$

This can then be expressed in terms of slip i :

$$\begin{aligned} |v_j| &= r_w \omega - r_w \omega \cos \theta + r_w \omega \cos \theta - v_w \cos \theta \\ |v_j| &= r_w \omega - r_w \omega \cos \theta + \left(\frac{r_w \omega - v_w}{r_w \omega} \right) r_w \omega \cos \theta \\ |v_j| &= r_w \omega (1 - (1 - i) \cos \theta) \end{aligned} \quad (2.17)$$

Using the fact that $dt = \frac{d\theta}{\omega}$, the shear displacement j at any angle θ along the wheel

rim can be calculated by integrating the slip velocity $|\mathbf{v}_j|$ [7].

$$j = \int_0^t |\mathbf{v}_j| dt = \int_{\theta}^{\theta_0} r_w \omega (1 - (1 - \iota) \cos \theta) \frac{d\theta}{\omega}$$

$$j = r_w [(\theta_0 - \theta) - (1 - \iota) (\sin \theta_0 - \sin \theta)] \quad (2.18)$$

Knowing the shear displacement j , the shear stress τ_s can be calculated at any point along the wheel rim by substituting Equation 2.18 into Equation 2.2.

$$\tau_s = (c + \sigma \tan \phi_s) \left(1 - e^{\frac{-r_w}{K_s} [(\theta_0 - \theta) - (1 - \iota) (\sin \theta_0 - \sin \theta)]} \right) \quad (2.19)$$

where $\sigma = \sigma_1$ for $\theta_m \leq \theta \leq \theta_0$ and $\sigma = \sigma_2$ for $0 \leq \theta < \theta_0$.

Forces perpendicular to the terrain include the normal load W , the perpendicular component of normal stress, and the perpendicular component of shear stress. Since Wong's analysis is performed for a wheel moving with a constant velocity entirely parallel to the ground, the perpendicular forces sum to zero.

$$\sum F_z = 0 = -W + b_w r_w \left(\int_0^{\theta_0} \sigma \cos \theta d\theta + \int_0^{\theta_0} \tau_s \sin \theta d\theta \right) \quad (2.20)$$

A root-finding algorithm must be used to solve Equation 2.20 for the wheel-soil contact angle θ_0 given normal load W and slip ι [7]. Once the wheel-soil contact angle θ_0 has been found, the drawbar pull DP and the magnitude of the resistive torque τ_R can be calculated:

$$\sum F_x = DP = H - R_c = b_w r_w \left(\int_0^{\theta_0} \tau_s \cos \theta d\theta - \int_0^{\theta_0} \sigma \sin \theta d\theta \right) \quad (2.21)$$

$$\tau_R = b_w r_w^2 \int_0^{\theta_0} \tau_s d\theta \quad (2.22)$$

where $\sigma = \sigma_1$ for $\theta_m \leq \theta \leq \theta_0$ and $\sigma = \sigma_2$ for $0 \leq \theta < \theta_0$.

The drawbar pull DP and resistive torque τ_R fully determine the net effect of the terrain on the wheel. Using Equations 2.20, 2.21, and 2.22 reduces drawbar pull DP and resistive torque τ_R to functions of normal load W and slip z exclusively for a wheel of fixed dimensions driving over homogeneous terrain with constant soil properties.

2.3 Wheel Dimensions

The wheel dimensions for *Kapvik* are necessary to determine its tractive performance. The wheel dimensions shown in Table 2.1 are used throughout this thesis in all calculations and simulations.

Table 2.1: Wheel dimensions.

Dimension	Value	Units
Wheel radius r_w	75	mm
Wheel width b_w	70	mm

2.4 Soil Properties

The soil properties used in this thesis are shown in Table 2.2. These values are predominantly from a soil simulant used by Ding et al. [12] in their thorough experimental study of driving wheels' performance on planetary soils; this soil simulant was made to closely resemble lunar soil. In Ding et al.'s paper, the pressure-sinkage constants are presented in terms of k_c and k_ϕ ; however, using Equations 2.12 and 2.13, Ding's value of soil deformation exponent n_s , and the *Kapvik* wheel width, equivalent values of k_1 and k_2 are found. It should be noted that these soil properties were chosen for two reasons: they are made to closely resemble values on the lunar surface; and this soil has undergone extensive experimental testing, allowing the results calculated in this thesis to be compared with empirical data.

Table 2.2: Soil properties.

Soil Property	Value	Units	Reference
Soil deformation exponent n_s	1.1	-	Ding soil simulant [12]
Cohesion c	250	Pa	Ding soil simulant [12]
Internal angle of friction ϕ_s	31.9	°	Ding soil simulant [12]
Shear deformation parameter K_s	11.4	mm	Average from Ding soil simulant [12]
Soil modulus of cohesion k_c	15.6	kPa/m ^{n_s-1}	Ding soil simulant [12]
Soil modulus of friction k_ϕ	2407.4	kPa/m ^{n_s}	Ding soil simulant [12]
Pressure-sinkage constant k_1	12.0	kPa	Ding soil simulant [12] and Equation 2.12
Pressure-sinkage constant k_2	1845.3	kPa/m	Ding soil simulant [12] and Equation 2.13
Maximum stress angle modulus c_1	0.18	-	Empirical, loose sand [7]
Maximum stress angle modulus c_2	0.32	-	Empirical, loose sand [7]

2.5 Numerical Evaluation of Traction Parameters

This section outlines the numerical methods used to calculate the traction parameters of interest: drawbar pull DP and resistive torque τ_R . The calculations are first shown for a slipping wheel, where $0 \leq i \leq 1$, in Sections 2.5.1-2.5.4 and then extended for a skidding wheel, where $-1 \leq i < 0$, in Section 2.5.5. The goal of this section is to create a function that takes as inputs the normal load W , the wheel centroid velocity v_w , and the wheel angular velocity ω , and outputs the drawbar pull DP and resistive torque τ_R exerted on the wheel by the soil. This function will be used for dynamic simulation in subsequent sections.

2.5.1 Slip

The evaluation of slip is complicated by ambiguities arising when the signs of v_w and ω are different, or when $v_w = 0$ and/or $\omega = 0$. Wong's formulae are developed for a wheel in steady-state (i.e. constant velocity). In this thesis Wong's principles were extended reasonably in order to resolve these ambiguous situations.

Consider a single wheel moving from left to right where a rightward wheel centroid velocity v_w is positive, and a clockwise angular velocity ω is positive. Two pieces of information are required for evaluation of DP and τ_R : the magnitude of the slip i , and the direction dir (either +1 or -1) in which to exert the drawbar pull DP and resistive torque τ_R . The treatment of different cases in finding i and dir is outlined in Table 2.3. The general function for slip/skid i for any combination of wheel centroid velocity v_w and wheel angular velocity ω is then:

$$i(v_w, \omega) = \begin{cases} 1 - \frac{v_w}{r_w \omega} & |r_w \omega| \geq |v_w| \\ \frac{r_w \omega}{v_w} - 1 & |r_w \omega| < |v_w| \end{cases} \quad (2.23)$$

where the function is subject to the case treatment in Table 2.3.

2.5.2 Stress Distribution

In order to calculate drawbar pull DP , resistive torque τ_R , and wheel-soil contact angle θ_0 it is necessary to calculate the normal and shear stress distribution around the wheel rim. Equations 2.11, 2.14, and 2.19 allow this calculation to be made for a given wheel soil contact angle θ_0 and a given slip i . These stress distributions will be integrated, so sample points are required. Two methods of integration are considered in this thesis: Riemann sum integration, which requires a large set of sample points; and Simpson's rule integration, which requires a total of five sample points (two in

Table 2.3: Treatment of different slip/skid cases.

Case	$ r_w\omega \geq v_w $	$ r_w\omega < v_w $	$\text{sgn } v_w = \text{sgn } \omega$	Action
1	✓	✗	✓	Normal range for slip $0 \leq \iota < 1$. Set $d\iota r = \text{sgn } \omega$.
2	✓	✗	✗	Wheel rotation and wheel centroid velocity are in opposite directions. Set slip to maximum value $\iota = 1$, and set $d\iota r = \text{sgn } \omega$.
3	✗	✓	✓	Wheel is skidding $1 < \iota < 0$. Set $d\iota r = \text{sgn } \omega$.
4	✗	✓	✗	Wheel rotation and wheel centroid velocity are in opposite directions. Set slip to maximum value $\iota = 1$, and set $d\iota r = \text{sgn } \omega$.
Case	$\omega = 0$	$v_w = 0$		Action
5	✓	✗		Wheel rotation locked $\iota = -1$. Set $d\iota r = \text{sgn } v_w$ since $\omega = 0$.
6	✗	✓		Wheel is spinning. Set slip to maximum value $\iota = 1$. Set $d\iota r = \text{sgn } \omega$.
7	✓	✓		Wheel is completely stationary. Set $\iota = 0$. Let τ_W represent the actuated wheel torque. If $\tau_W < \tau_R(\iota = 0)$, set $\tau_R = \tau_W$. Otherwise set $\tau_R = \tau_R(\iota = 0)$. If $DP(\iota = 0) < 0$, set $DP = 0$. Otherwise set $DP = DP(\iota = 0)$. Assume a positive direction $d\iota r = 1$.

the forward region, two in the rearward region, and one at the wheel-soil contact angle of maximum stress θ_m). An example stress distribution, and its approximation using five points are shown in Figure 2.3.

2.5.3 Normal Force Equilibrium

The normal load W must be countered by an equal and opposite reaction force. As shown in Equation 2.20, the reaction force is dependent on the total wheel-soil

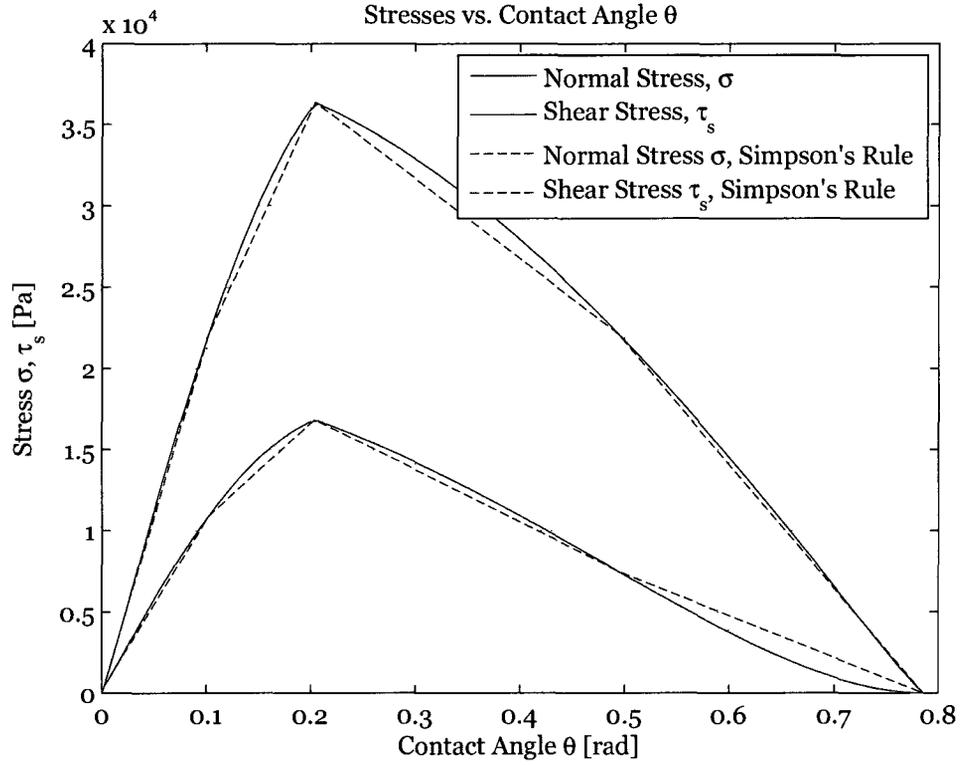


Figure 2.3: An example stress distribution using the *Kapvik* wheel dimensions from Table 2.1, the soil properties from Table 2.2, a wheel-soil contact angle θ_0 of 45° , and a slip ι of 0.25. Note the distinct transition between the forward and rearward regions at $\theta_m \approx 0.2$ rad.

contact angle θ_0 . Since θ_0 appears as a limit of integration and non-linearly inside the equation, the equation is too complex to solve using a closed form technique [7]. Instead, a routine using the `fzero` command in MATLAB is created to find the smallest angle θ_0 which will support the normal load W with a given slip ι . This is accomplished by finding the root of Equation 2.20.

$$\theta_0 = \text{root} \left(-W + b_w r_w \left[\int_0^{\theta_0} \sigma \cos \theta d\theta + \int_0^{\theta_0} \tau_s \sin \theta d\theta \right], \theta_0 \right) \quad (2.24)$$

2.5.4 Drawbar Pull and Resistive Torque

Once the contact angle θ_0 has been solved for using Equation 2.24, the normal and shear stress distributions can be obtained as shown in Section 2.5.2. With the stress distributions obtained, the integrations in Equations 2.21 and 2.22 can be performed numerically to solve for drawbar pull DP and resistive torque τ_R . This is done either using Riemann sum or Simpson's rule integration. An example of using Simpson's rule integration to approximate the resistive torque τ_R is shown below:

$$\begin{aligned}
 \tau_R &= b_w r_w^2 \int_0^{\theta_0} \tau_s(\theta) d\theta \\
 \tau_R &= b_w r_w^2 \left[\int_{\theta_m}^{\theta_0} \tau_{s_1}(\theta) d\theta + \int_0^{\theta_m} \tau_{s_2}(\theta) d\theta \right] \\
 \tau_R &\approx b_w r_w^2 \left[\frac{\theta_0 - \theta_m}{6} \left(\tau_{s_1}(\theta_m) + 4\tau_{s_1}\left(\frac{\theta_0 + \theta_m}{2}\right) + \tau_{s_1}(\theta_0) \right) \right] \dots \\
 &\quad \dots + b_w r_w^2 \left[\frac{\theta_m - 0}{6} \left(\tau_{s_2}(0) + 4\tau_{s_2}\left(\frac{\theta_m + 0}{2}\right) + \tau_{s_2}(\theta_m) \right) \right] \quad (2.25)
 \end{aligned}$$

where evaluation of the resistive torque is broken up into integrations of the forward region with shear stress τ_{s_1} and rearward region with shear stress τ_{s_2} .

Identical methodology is used to compute the result for drawbar pull DP . Simpson's rule approximation provides an accurate result with a decreased computational cost.

2.5.5 Traction Parameters for a Skidding Wheel

A wheel is considered to be skidding when the velocity of the wheel centroid v_w is larger than the no-slip velocity $r_w\omega$. Skidding wheels on deformable terrain have received very little attention in the literature compared to slipping wheels. A possible explanation for this is that in challenging environments where traction analysis needs to be performed, skidding is almost never encountered. Wong and Reece did perform an investigation of towed wheels on deformable terrain [8]; the formulae that they

developed are also valid for skidding wheels. However, by the nature of the resulting equations, the drawbar pull DP and resistive torque τ_R are discontinuous at zero slip ($\iota = 0$). This discontinuity is both physically unlikely and also introduces stiffness and instability into dynamic simulations. As a result, these equations are not used in this thesis.

Recent experiments by Ding et al. on rover wheels driving on planetary soil simulant were conducted down to skids of $\iota = -0.4$ [12]. Although this study does not cover the full spectrum of possible slips/skids (i.e. from $-1 \leq \iota \leq 1$), it does yield a continuous result at zero slip, and clear insight into the shapes of the drawbar pull DP and resistive torque τ_R curves for a skidding wheel in deformable terrain. The shapes of the curves in skid are very similar to those for slip, but are approximately anti-symmetric about the vertical axis ($\iota = 0$) intercept. The Pacejka “Magic Tyre Formula” used extensively for modelling the dynamics of road vehicles produces a curve of the character described above [25]. The drawbar pull curve for skidding is the same shape as the drawbar pull curve for slipping, but is anti-symmetric about the vertical axis intercept [11, 25]. Lhomme-Desages et al. use a similar model for drawbar pull that is also anti-symmetric about the vertical axis intercept. In this thesis, curves for both drawbar pull DP and resistive torque τ_R were made to be anti-symmetric about the vertical axis intercept. This ensured continuity at slip $\iota = 0$ and represented the best model available using existing theory and empirical results. When the wheel is skidding ($\iota < 0$), the following equations can be used to obtain the drawbar pull DP and resistive torque τ_R using the anti-symmetric curves:

$$DP(\iota) = -DP(\iota = -\iota) + 2DP(\iota = 0) \quad | \quad \iota < 0 \quad (2.26)$$

$$\tau_R(\iota) = -\tau_R(\iota = -\iota) + 2\tau_R(\iota = 0) \quad | \quad \iota < 0 \quad (2.27)$$

where DP is calculated using Equation 2.21 for $\iota \geq 0$, and τ_R is calculated using

Equation 2.22 for $\iota \geq 0$

2.5.6 Traction Parameters Summary

The full procedure for calculating the drawbar pull DP and resistive torque τ_R is shown in Algorithm 2.1. The resultant traction value curves are shown in Figure 2.4 for the wheel dimensions in Table 2.1, the soil properties in Table 2.2, and a normal load W of 50 N.

Algorithm 2.1 Calculate drawbar pull DP and resistive torque τ_R given normal load W , wheel centroid velocity v_w and wheel angular velocity ω

- 1 Calculate wheel slip/skid $\iota(v_w, \omega)$, and the direction d_{ir} from Equation 2.23 and associated conditions in Table 2.3
 - 2 If the wheel is skidding ($-1 \leq \iota < 0$), use Equations 2.26 and 2.27 to convert the problem into one for which slip ι is positive
 - 3 Find wheel-soil contact angle θ_0 using Equation 2.24, finding the first root of Equation 2.20 over the interval $0 < \theta_0 < \frac{\pi}{2}$
 - 4 Compute the normal stresses $\{\sigma_1, \sigma_2\}$ and shear stresses τ_s at either a large number of points (if using Riemann sum integration) or five points (if using Simpson's rule integration), with Equations 2.11, 2.14, and 2.19 respectively
 - 5 Using the wheel-soil contact angle θ_0 and slip ι obtained in the previous steps, calculate the drawbar pull DP and resistive torque τ_R using Equations 2.21, 2.22 and the calculated normal and shear stress distributions. Perform the integration using either Riemann sum integration or Simpson's rule integration and the sample points obtained in the previous step
 - 6 Multiply the drawbar pull DP and resistive torque τ_R by the direction d_{ir} found in Step 1
-

2.6 Equation Simplification

Despite the decrease in computational time afforded by the use of Simpson's rule integration, solving Equation 2.24 for θ_0 using a root-finding technique for every evaluation of drawbar pull DP and resistive torque τ_R is time consuming. The computational time can be greatly reduced by using a polynomial fit to approximate the

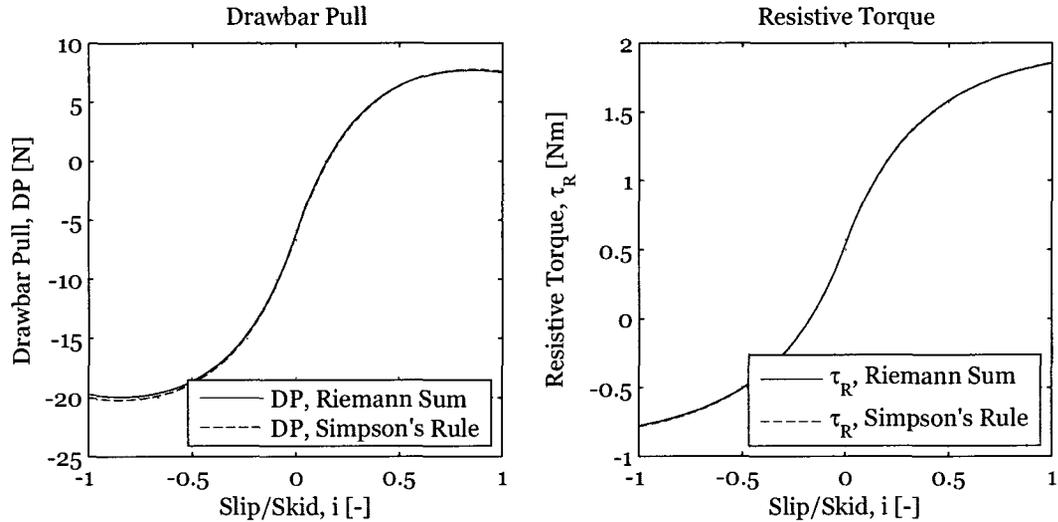


Figure 2.4: The relationship between drawbar pull DP (left), resistive torque τ_R (right), and slip/skid i for normal load $W = 50$ N. Both the Riemann sum method and Simpson's rule method are shown, with nearly identical results.

drawbar pull $DP(W, i)$ and resistive torque $\tau_R(W, i)$.

For a wheel of fixed dimensions driving in homogeneous terrain with a certain constant set of soil properties, drawbar pull DP and resistive torque τ_R are exclusively functions of normal load W and slip i ; this is demonstrated by Algorithm 2.1. Thus the drawbar pull DP and resistive torque τ_R both form surfaces in R^3 . In this thesis, O^{th} order polynomial fits in two variables are used to form accurate approximations of these functions. Over small parameter spaces, 2nd or 3rd order fits were found to be appropriate; over the large parameter space required for simulation, a 4th order fit was found to be appropriate (i.e. $O = 4$).

Using the soil properties defined in Table 2.2, a 100×100 array of results was calculated over the parameter space of $0 \leq i \leq 1$ and $0.1 \text{ N} \leq W \leq 200 \text{ N}$. The publicly available `polyfitweighted2` MATLAB function was used to find a 4th order

polynomial approximation of the form:

$$\begin{aligned}
 DP = & p_{00} + p_{10}v + p_{01}W + p_{20}v^2 + p_{11}vW + p_{02}W^2 + p_{30}v^3 + p_{21}v^2W + p_{12}vW^2 \dots \\
 & \dots + p_{03}W^3 + p_{40}v^4 + p_{31}v^3W + p_{22}v^2W^2 + p_{13}vW^3 + p_{04}W^4
 \end{aligned}
 \tag{2.28}$$

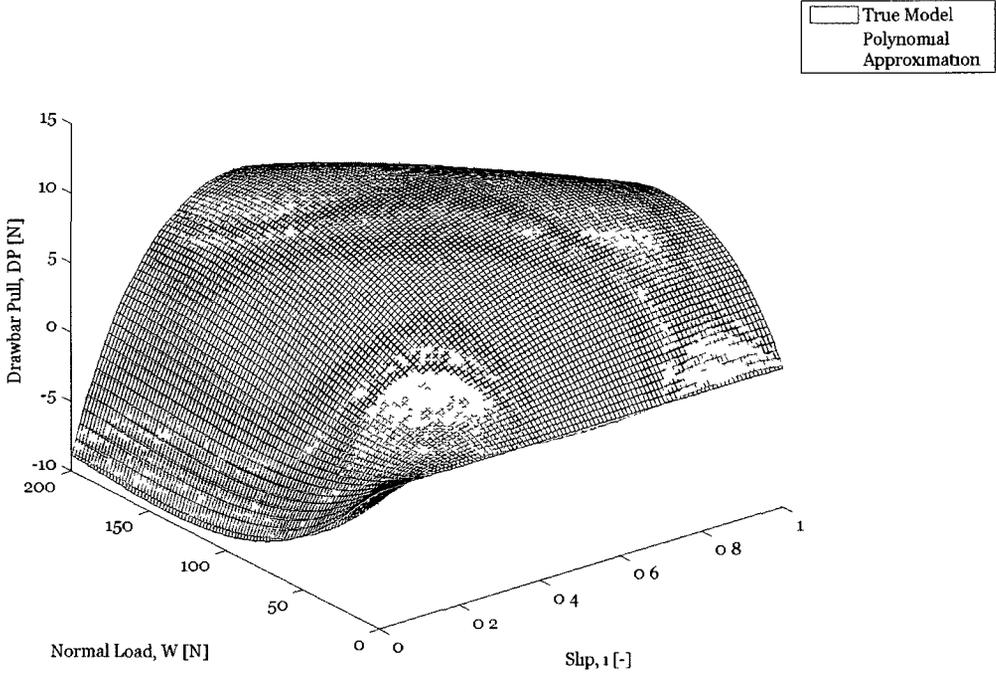
where p_{kl} is the coefficient for the term $v^k W^l$.

A 4th order polynomial was found to be the best balance of accuracy and simplicity over this parameter space. The true and approximated functions are shown in Figure 2.5 where the black surface represents the actual values found with Algorithm 2.1 using Riemann sum integration, and the green surface represents the 4th order polynomial approximation. The approximation was very accurate: the average error in drawbar pull DP was 0.10798 N and the average error in resistive torque was 0.0080327 Nm over the parameter space considered. A 3rd order approximation was found to have average errors of 2.3181 N and 0.14527 Nm for drawbar pull DP and resistive torque τ_R respectively. A 5th order approximation, which was found to have average errors of 0.024015 N and 0.0022771 Nm for drawbar pull DP and resistive torque τ_R respectively, was judged by the author to be past the point of diminishing returns. The coefficients resulting from the 4th order polynomial fits are shown in Table A.1 of Appendix A.1. For different soils and associated soil properties, a unique set of polynomial fit coefficients are produced. In Appendix A.2, a polynomial approximation is performed using recommended lunar soil properties from the “Lunar Sourcebook” [26]; in Appendix A.3, a polynomial approximation is performed using dry sand soil properties from Wong’s “Theory of Ground Vehicles” [6]. The accuracy of the results demonstrates that the polynomial approximation method is applicable to multiple soil types. Since the form of the equations for traction parameters are unaffected by soil type, it is proposed that a polynomial function will always be able to accurately approximate the drawbar pull DP and resistive torque τ_R relationships,

regardless of the soil properties.

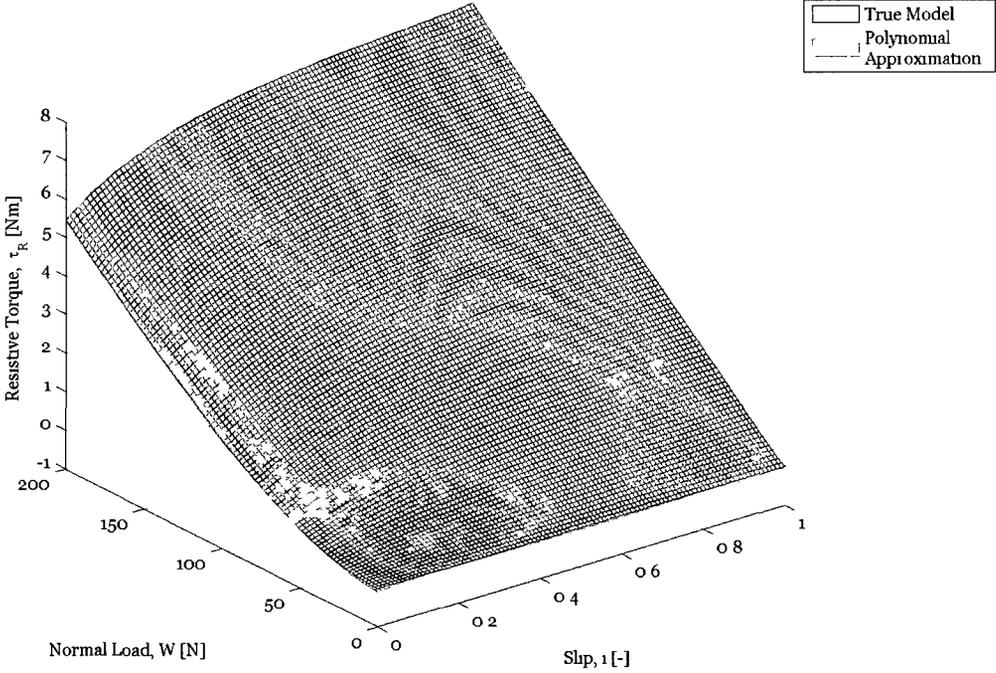
The polynomial approximation produced is used to drastically increase the simulation speed. Later in this thesis, a polynomial approximation is reconstructed through estimation of resistive torque τ_R , drawbar pull DP , normal load W , and slip i .

Drawbar Pull 4th Order Polynomial Fit



(a) Drawbar pull DP true model (black) and polynomial approximation (green).

Resistive Torque 4th Order Polynomial Fit



(b) Resistive torque τ_R true model (black) and polynomial approximation (green).

Figure 2.5: Net traction parameters as functions of W and i and their corresponding 4th order polynomial approximations.

Chapter 3

Kapvik Chassis Design

Kapvik is a 30 kg micro-rover prototype which was developed collaboratively by MPB Technologies, Carleton University, Ryerson University, the University of Toronto, Xiphos Technologies, MDA Space Missions, and the University of Winnipeg for the Canadian Space Agency. This chapter outlines some of the work done by the author on the design of the rocker-bogie mobility system.

3.1 Mobility System Overview

The rocker-bogie planetary rover mobility system was developed by the National Aeronautics and Space Administration (NASA) and the Jet Propulsion Laboratory (JPL) and used on both Sojourner and the Mars Exploration Rover rovers [1]. The rocker-bogie mobility system comprises a series of kinematic linkages without springs. One goal of this mobility system is to evenly distribute normal loads amongst the wheels, allowing each wheel to develop an equal tractive force. The rocker-bogie mobility system is also designed to traverse large obstacles of up to one wheel diameter in height [1]. The rocker-bogie mobility system designed for *Kapvik* is 0.782 m wide and 0.850 m long and is shown in Figure 3.1. In this thesis, the terms rover body and cab are used interchangeably to refer to the chassis' payload, shown in yellow in

Figure 3.1; the terms chassis and mobility system are also used interchangeably. The left and right sides of the rover body are connected to two rocker links via revolute joints. The rotation of these joints is limited to $\pm 16^\circ$ with hard stops. A differential mechanism ensures that the joint angles of the two rockers are equal in magnitude but opposite in direction with respect to the body, minimizing pitching of the cab. The front wheel is attached to the front end of the rocker. Another link, termed the bogie, is attached to the rear end of the rocker with a free revolute joint. The rotation of this joint is limited to $\pm 30^\circ$ with hard stops. Two rear wheels are attached to either end of the bogie. The mobility system thus has three wheels per side. The wheels are individually commanded by a wheel drive system that includes electric motors and a gear train. Since low power consumption is more important than high speed on a planetary rover, the gear ratio of the wheel drive system is very high. Typically, the rocker-bogie system is Ackermann steered, which requires four steering motors on the corner wheels. These motors were excluded from *Kapvik* in order to reduce mass and complexity; the rover is instead skid steered by sending different speed commands to the left and right wheels.

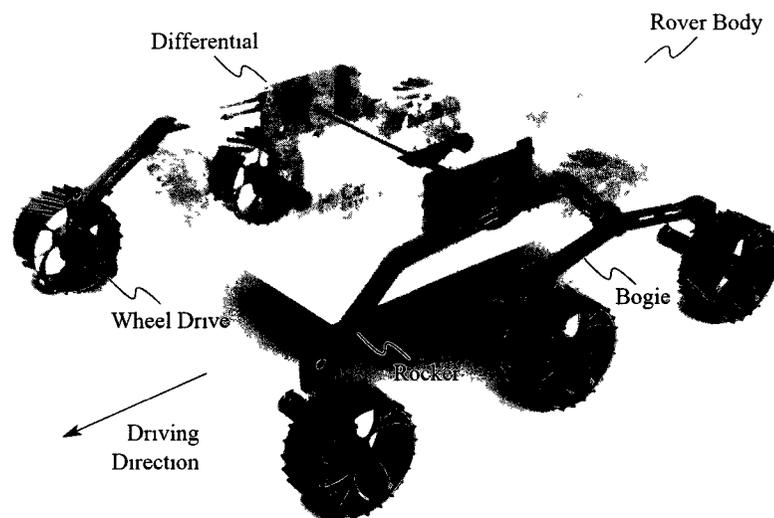


Figure 3.1: *Kapvik* rocker-bogie mobility system.

A manipulator, designed by Ryerson University and not shown in Figure 3.1, is attached to the top of the rover. It serves the dual functions of taking soil samples with an actuated scoop and supporting an elevated camera.

At the time of writing, the *Kapvik* chassis has been completely assembled and has undergone a small number of basic test drives. Additional chassis design documentation as well as a photographs of the assembled chassis can be found in Appendix B.

3.2 Wheel Drive System

Kapvik's wheel drive system is shown in Figure 3.2. The wheel is made of Aluminum and has 24 grousers, each 5 mm high and helical at an angle of 19.4° . The Harmonic Drive is a compact, low backlash, high ratio gearhead, the particular model used on *Kapvik* was the CSF-11-2XH-F with a gear ratio of 100:1, a flange output was chosen for direct attachment to the wheel. A custom-made intermediate plate connects the Harmonic Drive to the planetary gearhead. The planetary gearhead is a Maxon 2-stage GP26B with a gear ratio of 14:1. Thus the total gear ratio is 1400:1. A Maxon RE25, 43 mm long motor with graphite brushes was used to power the wheels. A 500 count per turn, magneto-resistant, three channel quadrature encoder was attached to the back shaft of the motor to measure motor revolutions. A motor enclosure (not shown) protects the wheel drive assembly from dust, moisture, and other contaminants.

Maxon brushed motors were selected primarily for their flight heritage on Sojourner and the Mars Exploration Rovers [1, 27]. The *Kapvik* rover prototype was designed with a clear path to flight in mind, the selection of these motors helped to fulfil this requirement. The motor that best met the operational requirements of *Kapvik* with the highest efficiency was found to be the Maxon RE25 motor, rated for 36 V, but

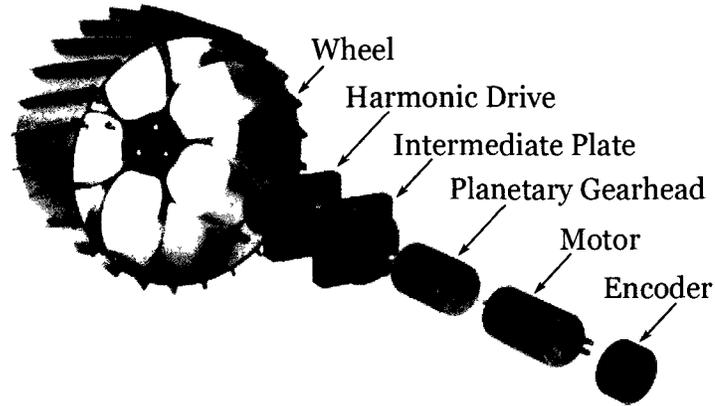


Figure 3.2: An exploded view of *Kapvik*'s wheel drive system.

run on the rover's 24 V solar array output. The details of the selected motor are shown in Table 3.1.

Table 3.1: Maxon RE25 motor specifications [28].

Variable	Value	Units
Rated Voltage	36	V
Operational Voltage	24	V
Torque Constant	32.9	mNm/A
Speed Constant	290	rpm/V
Resistance	4.37	Ω
Maximum Current	0.863	A
No-load Current	0.0575	A
Speed/Torque Gradient	38400	rpm/Nm
Inertia	13.4	g cm ²
Maximum Continuous Torque	30.8	mNm

The operational conditions of a rover in a planetary environment are unpredictable and it is therefore difficult to foresee all possible scenarios [29]. This makes it challenging to set a requirement for the maximum torque required from the wheel drive system. In the design of the Mars Exploration Rovers, JPL was faced with the same

problem and came up with a conservative specification: that in order to avoid being torque-limited, each wheel needed to be able to provide a force of half the total rover weight at the wheel rim [29]. This specification was followed on *Kapvik*. *Kapvik* is a terrestrial prototype, whereas the Mars Exploration Rovers were designed for the surface of Mars; thus, this requirement is more stringent for *Kapvik*, since Martian gravity is only 0.367 times that on Earth. For the 30 kg *Kapvik* micro-rover on Earth with wheels that are 15 cm in diameter, the wheel drive must generate a maximum output torque of 11 Nm. Note that the conditions under which a torque this high is necessary are rare, so it does not need to be provided continuously. To obtain this high torque with the low power Maxon motors and the transmission efficiencies of the available gears, it was found that the best combination of gears was a 14:1 Maxon planetary gearhead and a 100:1 Harmonic Drive gearhead. The designed top speed of the rover was 80 m/h, or approximately 2.2 cm/s; this speed, which is equivalent to a wheel angular velocity of 2.83 rpm under no-slip conditions, is attainable even with a combined gear ratio of 1400:1. The details of the selected gear train are shown in Table 3.2.

As shown in Table 3.2, the Harmonic Drive’s repeated peak output torque of 11 Nm exactly matches the required peak torque. With the motor outputting its maximum continuous torque of 30.8 mNm and the torque transmission efficiencies shown in Table 3.2, the torque output of the planetary gearhead is 0.2415 Nm, and the torque output of the Harmonic Drive is $\tau_{Wmax} = 12.24$ Nm. Under these conditions the planetary gearhead’s maximum continuous output torque of 0.6 Nm will not be exceeded; the Harmonic Drive’s recommended value for repeated peak torque of 11 Nm will limit the output of the wheel drive system. However, if absolutely necessary, this torque can be exceeded, as the maximum momentary torque specification is 25 Nm.

Table 3.2: Wheel drive gear train specifications [28, 30]. † Efficiencies vary with load and temperature; these representative values are taken as 80 % of the maximum efficiency for the planetary gearhead, and at 20°C and 1 Nm output load for the Harmonic Drive gearhead.

Variable	Value	Units
Maxon GP26B 14:1 Planetary Gearhead		
Gear Ratio	14	-
Transmission Efficiency †	0.56	-
Input Inertia	0.5	g/cm ²
Maximum Continuous Output Torque	0.6	Nm
Repeated Peak Output Torque	0.9	Nm
Harmonic Drive CSF-11-2XH-F 100:1 Gearhead		
Gear Ratio	100	-
Transmission Efficiency †	0.51	-
Input Inertia	14	g/cm ²
Maximum Continuous Output Torque	8.9	Nm
Repeated Peak Output Torque τ_{rpt}	11	Nm

These maximum design torques are far in excess of what will typically be required on *Kapvik*. A typical normal load on a *Kapvik* wheel will be 50 N. With reference to Figure 2.4, the maximum torque necessary in steady-state operation with a normal load of 50 N is 1.856 Nm, occurring at a slip $i = 1$. Note that this is only valid for the soil properties listed in Table 2.2. The excess torque will be useful for acceleration, for climbing grades, for different types of soil, for obstacle negotiation, if a wheel drive system is lost, or for any combination of these five scenarios.

The performance of the wheel drive system at the output, including the effect of all gear ratios and transmission efficiencies, is shown in Figure 3.3. The straight, sloped line indicates the limit of the wheel drive system's operation. All operating points below it and to the left of the maximum wheel torque τ_{Wmax} are attainable in continuous

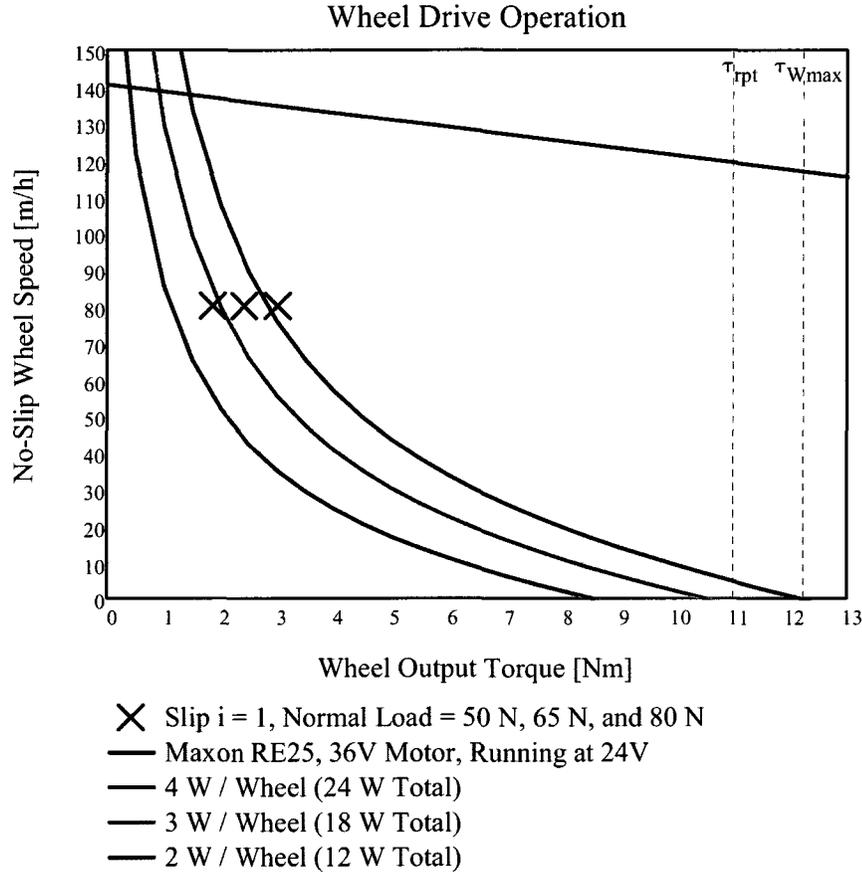


Figure 3.3: *Kapvik* wheel drive system operation. τ_{rpt} indicates the Harmonic Drive's maximum repeated peak torque; τ_{Wmax} indicates the maximum continuous torque possible if the Harmonic Drive's repeated peak torque specification is ignored. Operating points are shown for slip $i = 1$ and normal loads of 50 N, 65 N, and 80 N (left to right) using the soil properties listed in Table 2.2.

operation. The curved lines indicate lines of constant power consumption. *Kapvik* has a total of 24 W of power assigned to the operation of the wheel drive systems; thus the maximum power per wheel at which all wheels can be run simultaneously is 4 W. As can be seen in Figure 3.3, operation of every wheel simultaneously up to the maximum possible wheel drive torque τ_{Wmax} is possible if the wheel speed is reduced to nearly zero. The Harmonic Drive's maximum repeated peak torque τ_{rpt} is also indicated. Three representative operating points were chosen and are marked by \times 's in Figure 3.3. Each has the design top speed of 80 m/h as its no-slip wheel speed ($r_w\omega$).

The wheel torques are set to the largest possible resistive torques for wheel normal loads of 50 N, 65 N, and 80 N (left to right in Figure 3.3). These torques are obtained using the terramechanics equations developed Chapter 2, the soil properties listed in Table 2.2, and a slip of $i = 1$. All representative operating points are attainable using under 4 W of power per wheel except when the normal load is 80 N. To develop the necessary torque for this operating point, the speed could either be decreased, or the power delivered to this wheel could be increased.

As a structural consideration, the Harmonic Drives were placed so that their output bearings were coplanar with the wheel mid-planes. This was done to minimize the off-axis moment loading on the output bearing and thus maximize the lives of the Harmonic Drives.

3.3 Differential Mechanism

The differential mechanism minimizes the rover body's pitch by ensuring that the rocker joint angles are equal and opposite. It accomplishes this using two planetary gear assemblies; together with a torsion bar, these two planetary gear assemblies reverse and transfer the rotation between the two rockers. The torsion bar runs through the center of the rover body. The differential mechanism presented in this thesis is believed to be similar to that utilized on the Mars Exploration Rovers [1]; however, since a detailed description of its design is lacking in the literature, it was developed independently by the author.

Figure 3.4 shows a planetary gear comprised of a carrier, four planet gears, an annular gear, and a sun gear. In the general case, the carrier and all of the gears are allowed to rotate about their centroids. Consider all angular velocities relative to that of the

carrier:

$$\omega_P = \omega_C + \omega_{P/C} \quad (3.1)$$

$$\omega_A = \omega_C + \omega_{A/C} \quad (3.2)$$

$$\omega_S = \omega_C + \omega_{S/C} \quad (3.3)$$

where $\{\omega_C, \omega_P, \omega_A, \omega_S\}$ are the angular velocities of the carrier, the planet gears, the annular gear, and the sun gear respectively, and $\{\omega_{P/C}, \omega_{A/C}, \omega_{S/C}\}$ are the angular velocities of the planet gears, the annular gear, and the sun gear with respect to the carrier.

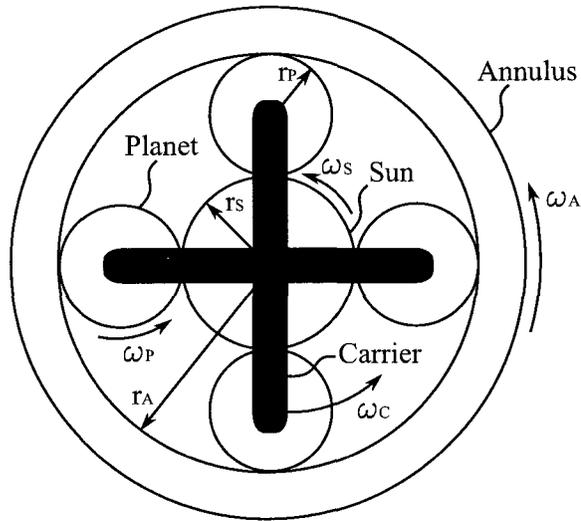


Figure 3.4: General planetary gear.

Considering the relative angular velocities alone is the same as considering the carrier as fixed, so:

$$\omega_{P/C} = \frac{r_A}{r_P} \omega_{A/C} \quad (3.4)$$

$$\omega_{P/C} = \frac{-r_S}{r_P} \omega_{S/C} \quad (3.5)$$

Combining Equations 3.4 and 3.5 gives a general equation for the motion of a planetary gear:

$$\begin{aligned} r_A \omega_{A/C} &= -r_S \omega_{S/C} \\ r_A (\omega_A - \omega_C) &= -r_S (\omega_S - \omega_C) \end{aligned} \quad (3.6)$$

The torsion bar was attached to the sun gear on both sides. The left hand side's planetary gear system has the carrier fixed to the rover body and the annulus attached to the rocker. If the angular velocity of the rover body is set to zero, $\omega_C = 0$, and:

$$\omega_S = \frac{-r_A}{r_S} \omega_A = -4\omega_A \quad (3.7)$$

The left hand side is used to reverse the direction of rotation. The angular velocity of the torsion bar ω_S is made to be four times the angular velocity of the rocker ω_A in order to reduce the torque being transmitted through the torsion bar by a factor of four. Considering the geometry of the planetary gear, where $r_S + 2r_P = r_A$, Equation 3.7 sets the ratio of gear radii on the left hand side:

$$r_P = \frac{3}{2} r_S \quad (\text{left hand side}) \quad (3.8)$$

On the right hand side's planetary gear system, the annulus is fixed to the rover body, and the carrier is attached to the rocker; this gives $\omega_A = 0$, and:

$$\omega_S = \frac{r_S + r_A}{r_S} \omega_C = 4\omega_C \quad (3.9)$$

Again considering the geometry of the planetary gear, Equation 3.9 sets the ratio of gear radii on the right hand side:

$$r_P = r_S \quad (\text{right hand side}) \quad (3.10)$$

When referring to gears and not to cylinders as depicted in Figure 3.4, the radius refers to the pitch radius, and all of the gears must have the same pitch. The details of the gears used on *Kapvik* are outlined in Table 3.3.

Table 3.3: Specifications of *Kapvik*'s planetary gears. All gears are stainless steel with a face width of 5 mm, a pitch of 0.5 mm, and were ordered from Stock Drive Parts / Sterling Instruments.

Gear	Pitch Diameter [mm]	Number of Teeth
Left Planet	18	36
Left Sun	12	24
Left Annulus	48	96
Right Planet	16	32
Right Sun	16	32
Right Annulus	48	96

An exploded view of the differential designed for *Kapvik* is shown in Figure 3.5. Ball bearings were press fit into housings to facilitate all rotational motion. A hollow shaft potentiometer was used to measure rotation of the torsion bar; with knowledge of the gearing relationship this allows for the calculation of both rocker joint angles. The gears and shafts were pair drilled/tapped and attached with set screws. Dust immunity was provided by a differential enclosure and the use of double sealed ball bearings at the rocker rotation joints. The differential enclosure was also designed to carry loads from each rocker to the rover body, leaving the gears free of load. Hard stops on the differential enclosure restrict the motion of the attached rocker to $\pm 16^\circ$.

The differential was successfully assembled and tested.

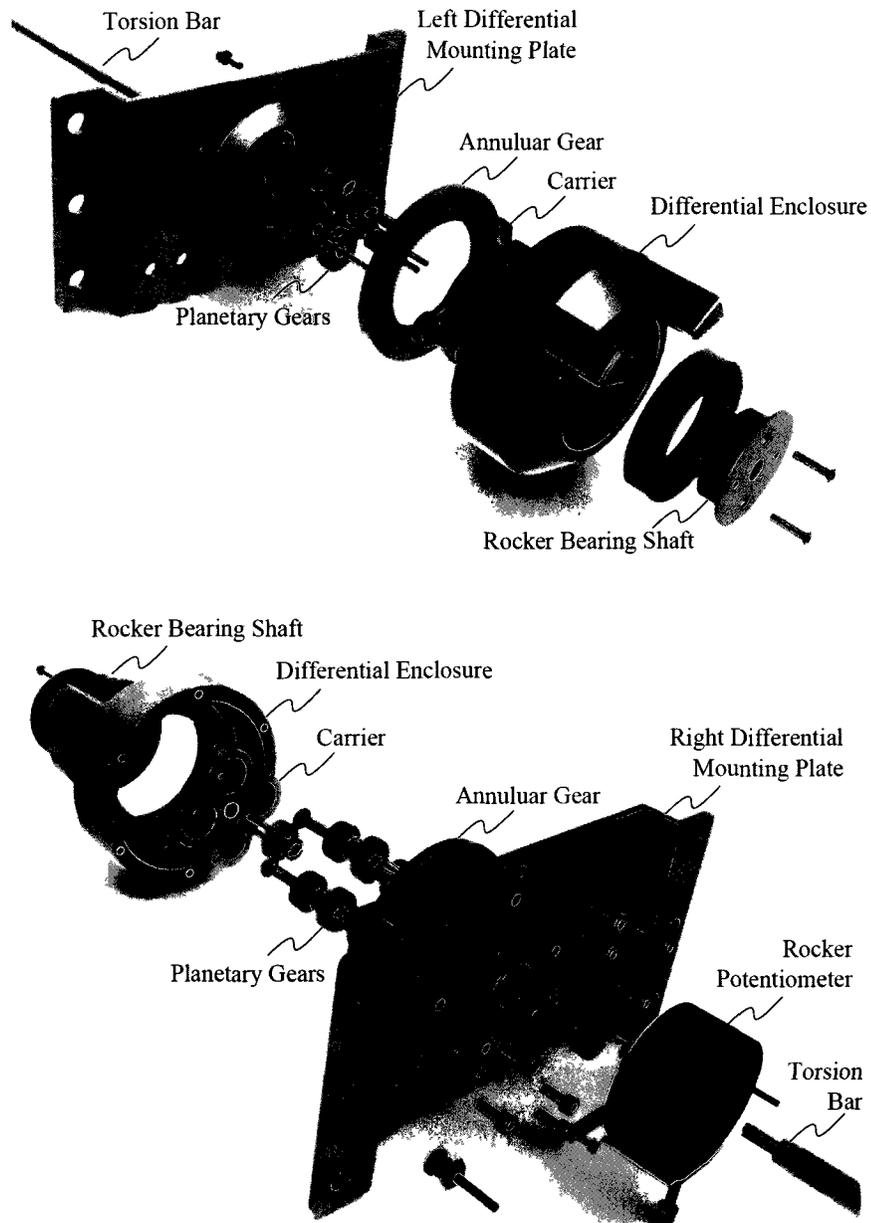


Figure 3.5: Exploded views of left differential gearing system (top) and right differential gearing system (bottom).

3.4 Force Sensor Integration

A novel feature of *Kapvik*'s rocker-bogie mobility system is the addition of load cells above the wheel hubs. As discussed later in this thesis, these sensors can be used to aid in net traction estimation. Further to this, they could also be used in tip-over stability control by sensing when a wheel is close to losing contact with the ground. *Kapvik*'s manipulator could then be used to appropriately adjust the center of gravity.

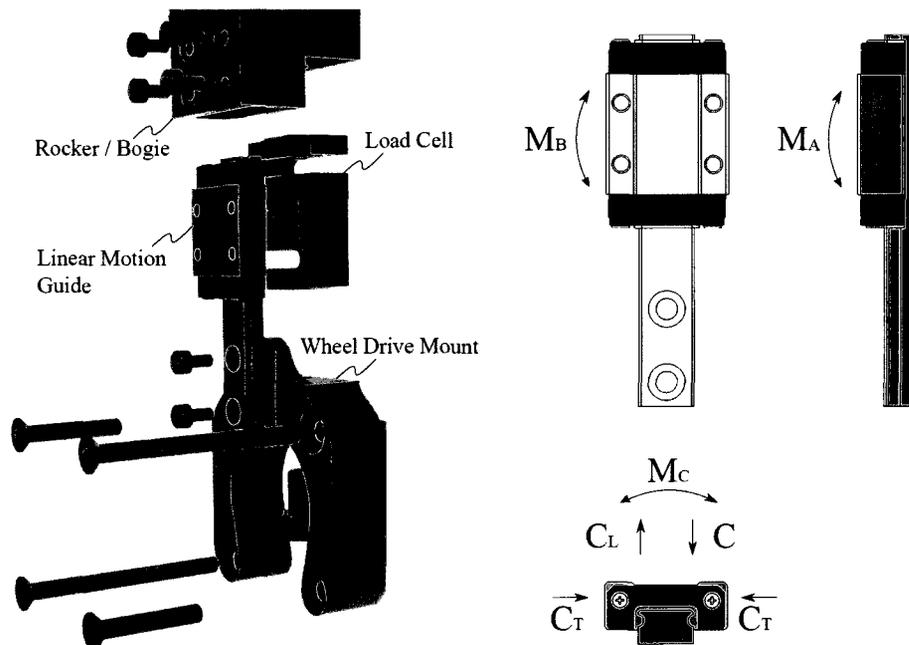


Figure 3.6: Load cell and linear guide rail assembly (left) and linear guide rail off-axis forces and moments (right).

To minimize the complexity and mass of the load sensing system, a single-axis load sensor of the appropriate dimensions and load range was sought. The Sherborne Sensors SS4000M miniature universal load cell, with a ± 200 N range, was chosen as the best candidate. Along the sensing axis, the overload capacity is 150% of the rated range, or ± 300 N [31]. Since the weight of the rover in Earth gravity is 294 N, the entire weight of the rover could be placed statically along the sensing axis without damaging the load cell. Despite the strength of the load cell along its sensing axis, its

ability to handle off-axis loading has not been thoroughly tested and was unable to be guaranteed by Sherborne Sensors. To take the load cell out of the off-axis load path, a linear guide rail is used. The guide rail's axis of motion is aligned with the sensing axis of the load cell; this allows the component of force in the sensing axis to pass through the load cell, but passes all off-axis forces and moments through the linear guide rail. A custom-sized THK linear motion guide RSR9ZM is used. The load cell and linear guide rail assembly, as well as the directions of the linear guide rail's load ratings are shown in Figure 3.6. The off-axis load capabilities of the RSR9ZM linear motion guide are shown in Table 3.4. It was determined that the linear guide rail had the strength to handle foreseeable off-axis forces and moments. Later in this thesis, it is shown that using multi-axis force sensors offers improvements to the net traction estimation algorithm; when the load sensing system was selected, these advantages were unknown to the author.

Table 3.4: Permissible off-axis loading for the THK RSR9ZM linear motion guide [32]. The loading directions are depicted in Figure 3.6. The static load rating was larger than the dynamic load rating: 2250 N as opposed to 1470 N.

Variable	Value	Units
Dynamic Load Rating C	1470	N
Reverse Dynamic Load Rating C_L	1470	N
Lateral Dynamic Load Rating C_T	1470	N
Moment Rating M_A	7.34	Nm
Moment Rating M_B	7.34	Nm
Moment Rating M_C	10.4	Nm

3.5 Structural Analysis

Finite element analysis performed in SolidWorks Simulation was used in the design of the rocker and bogie links. In order to reduce mass, material was removed from a baseline design in areas of low stress. The result was the removal of material in the center of the beams where the material's contribution to the second moment of inertia was minimal. The design remains very conservative. The links were all made of Aluminum 6061-T6 because of its strength, weight, and availability.

Several combinations of load magnitudes and directions were simulated. For the sake of brevity, the results of a single simulation of the rocker and a single simulation of the bogie are presented here. The links were fixed at their rotation point and 294 N (the weight of the rover) was applied at each wheel hub in the same direction as it would be on level ground. For the rocker link, this meant that 588 N was applied at the bogie joint. The results of this simulation are shown in Figure 3.7. The maximum stress in the rocker was 202.8 MPa, occurring close to the central hub where the rocker is attached to the differential. The maximum stress in the bogie was 100.0 MPa, occurring close to the central hub where the bogie is attached to the rocker. The yield stress of Aluminum 6061-T6 is 275 MPa; thus the links were able to withstand the applied force without exceeding the yield stress of the material. Since the weight of the rover will typically be divided between the six wheels, this simulation demonstrates the large safety factor built into the rocker and bogie link designs.

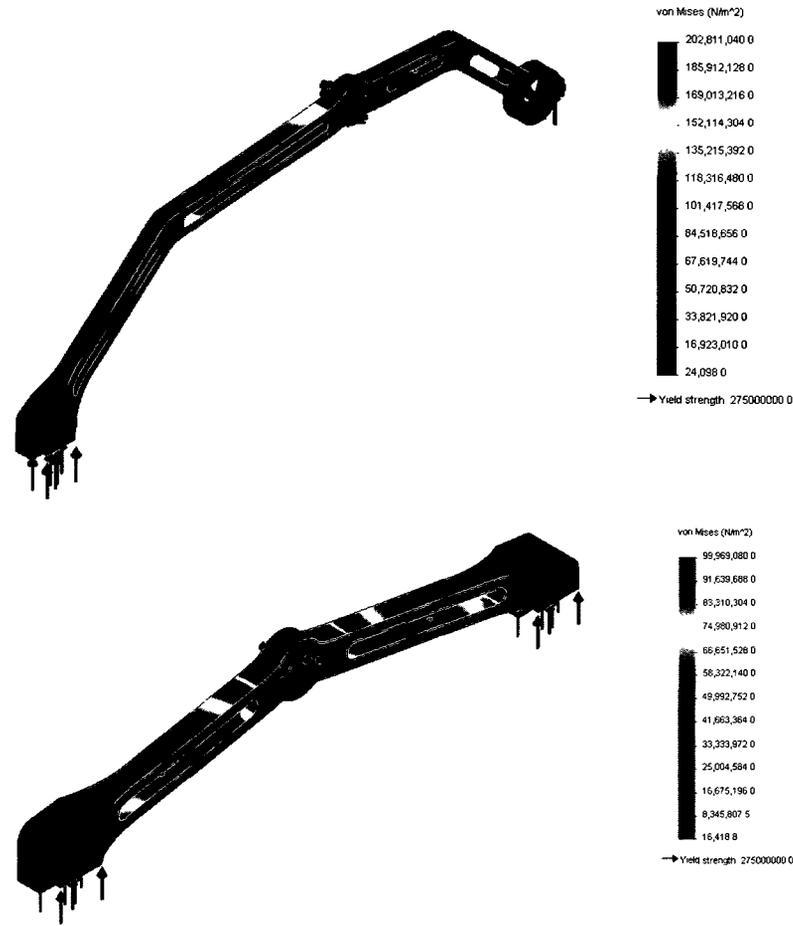


Figure 3.7: Rocker finite element analysis simulation (top) and bogie finite element analysis simulation (bottom). The weight of the entire rover is applied at each of the wheel hubs. The Von Mises stress is shown in N/m^2 .

3.6 Static Stability

When driving a rover on sloped terrain, a typical concern is the conditions under which the rover will tip over. Two scenarios, simplified to two dimensions, provide good insight into *Kapvik*'s static stability. The first scenario is *Kapvik* in an entirely cross-hill position as in Figure 3.8a; the second is *Kapvik* in an entirely uphill-downhill position as in Figure 3.8b. Consider a co-ordinate system $\{X^R, Y^R, Z^R\}$ aligned

with the rover body and with an origin at the center of the torsion bar (the rover's centroid). In this co-ordinate system the center of gravity of the entire rover has co-ordinates $\{x_{cg}^R, y_{cg}^R, z_{cg}^R\}$. The width of the footprint w_f is 782.25 mm, the length of the footprint l_f is 700 mm, the height of the rover cab centroid h_r is 250 mm, the rocker-bogie dimension a is 100 mm, and the rocker-bogie dimension b is 175 mm. Reaction forces at the wheels are assumed to be purely in the vertical direction in world co-ordinates $\{X_w, Y_w, Z_w\}$ (i.e. entirely in Z_w) and the rover is assumed to be stationary.

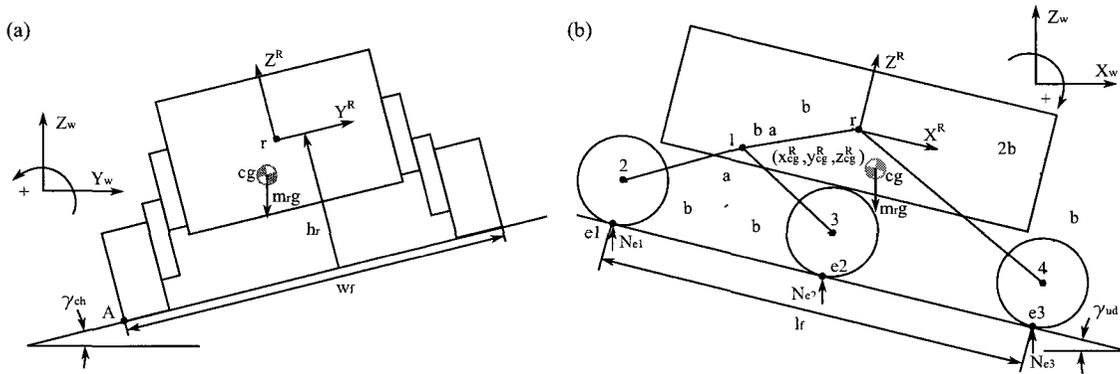


Figure 3.8: Co-ordinates and variables involved in rover static stability.

3.6.1 Cross-Hill Static Stability

The cross-hill static stability of the rover does not involve the rocker-bogie suspension system. Consider the moments about point A in Figure 3.8a. At the critical slope angle γ_{ch} , where the rover is at the limit of its stability, the sum of the moments at A will be zero, and the reaction force from the right set of wheels will be zero.

$$\sum M_A = -m_r g \cos \gamma_{ch} \left(\frac{w_f}{2} + y_{cg}^R \right) + m_r g \sin \gamma_{ch} (h_r + z_{cg}^R) = 0 \quad (3.11)$$

where m_r is the total rover mass, and g is the local gravitational acceleration.

Rearranging Equation 3.11 to solve for critical cross-hill stability angle γ_{ch} gives:

$$\gamma_{ch} = \tan^{-1} \left(\frac{\frac{w_f}{2} + y_{cg}^R}{h_r + z_{cg}^R} \right) \quad (3.12)$$

The critical cross-hill static stability angle γ_{ch} for various center of gravity positions is shown in Figure 3.9. As can be seen, a higher or more downhill center of gravity makes the rover less stable. The critical cross-hill static stability angle γ_{ch} is 57.4° when the rover's center of gravity is at the rover centroid (i.e. when $[x_{cg}^R, y_{cg}^R, z_{cg}^R] = [0, 0, 0]$). An estimate of the center of gravity was taken for a fully upright manipulator deployed as the camera mast; this represents the worst case for cross-hill static stability. This data point, shown in Figure 3.9, has a cross-hill static stability angle of 51.2° .

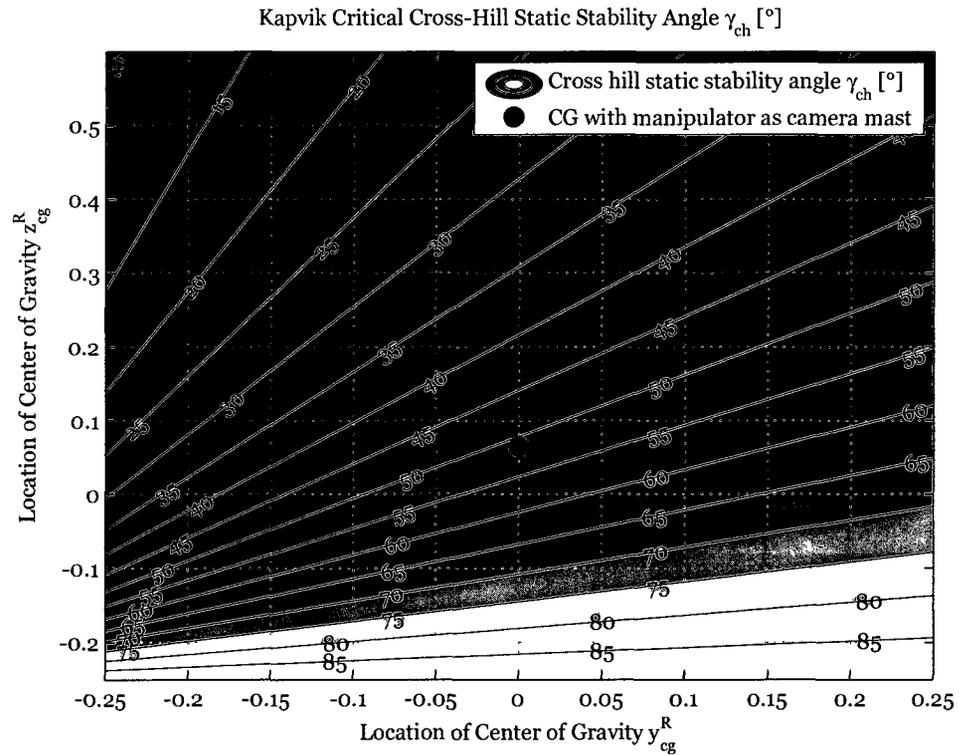


Figure 3.9: Critical cross-hill stability angles.

3.6.2 Uphill-Downhill Static Stability

Analysis of the uphill-downhill static stability of the rover involves the suspension system and is therefore more complex. Let the definition of uphill-downhill critical static stability angle γ_{ud} in this thesis be the angle at which any wheel loses contact with the ground. Although there are instances in which a wheel can lose contact with the ground without the rover tipping over, this definition will give a conservative estimate of uphill-downhill stability.

To determine the angle at which a rover wheel loses contact with the ground, one half of the rover is analyzed in two dimensions on planar, tilted ground. The wheels make contact with the terrain at points $e1$, $e2$, and $e3$, as shown in Figure 3.8b. The sum of the moments at $e3$ is given by:

$$\begin{aligned} \sum M_{e3} = & \frac{m_r g}{2} \left[\sin \gamma_{ud} (b + r_w + z_0^R) - \cos \gamma_{ud} (2b - x_0^R) \right] \dots \\ & \dots + N_{e1} \cos \gamma_{ud} 4b + N_{e2} \cos \gamma_{ud} 2b = 0 \end{aligned} \quad (3.13)$$

where $\{N_{e1}, N_{e2}, N_{e3}\}$ are the vertical reaction forces at wheel-terrain contact points $e1$, $e2$, and $e3$ respectively.

The sum of the moments at the bogie joint, joint 1, will also be zero since the bogie joint is a free joint connected by a bearing. This can be used twice: once for analyzing the bogie link, and once for analyzing the rocker link. Considering the bogie link:

$$\begin{aligned} \sum M_1 = & N_{e1} [\sin \gamma_{ud} (a + r_w) + \cos \gamma_{ud} b] \dots \\ & \dots + N_{e2} [\sin \gamma_{ud} (a + r_w) - \cos \gamma_{ud} b] = 0 \end{aligned} \quad (3.14)$$

Considering the rocker link:

$$\begin{aligned} \sum M_1 = \frac{m_r g}{2} [\sin \gamma_{ud} (b - a + z_0^R) + \cos \gamma_{ud} (b + x_0^R)] \dots \\ \dots + N_{e3} [\sin \gamma_{ud} (a + r_w) - \cos \gamma_{ud} 3b] = 0 \end{aligned} \quad (3.15)$$

Solving these three equations for $\{N_{e1}, N_{e2}, N_{e3}\}$ gives:

$$N_{e1} = \frac{m_r g [(2b - x_0^R) - \tan \gamma_{ud} (b + r_w + z_0^R)]}{\frac{b + \tan \gamma_{ud} (a + r_w)}{b - \tan \gamma_{ud} (a + r_w)} 4b + 8b} \quad (3.16)$$

$$N_{e2} = \frac{m_r g [(2b - x_0^R) - \tan \gamma_{ud} (b + r_w + z_0^R)]}{\frac{b - \tan \gamma_{ud} (a + r_w)}{b + \tan \gamma_{ud} (a + r_w)} 8b + 4b} \quad (3.17)$$

$$N_{e3} = \frac{m_r g [(b + x_0^R) + \tan \gamma_{ud} (b - a + z_0^R)]}{6b - 2 \tan \gamma_{ud} (a + r_w)} \quad (3.18)$$

where Equations 3.16, 3.17 and 3.18, also satisfy the vertical force balance $N_{e1} + N_{e2} + N_{e3} = \frac{m_r g}{2}$.

When the rover's center of gravity is at the rover centroid (i.e. when $[x_{cg}^R, y_{cg}^R, z_{cg}^R] = [0, 0, 0]$), the wheel reaction forces for different slope angles are shown in Figure 3.10. In both the forward tilting case and the rearward tilting case, a bogie wheel is the first to lose contact. The critical angle for bogie instability γ_b occurs when the bogie joint is directly overhead the wheel-terrain contact point (*e1* or *e2*).

$$\gamma_b = 90^\circ - \tan^{-1} \left(\frac{a + r_w}{b} \right) \quad (3.19)$$

For *Kapvik*, this means that in both the uphill and downhill case, the bogie wheel loses contact at 45° . Equation 3.19 also reveals the design motivation for a low bogie joint: to increase bogie stability.

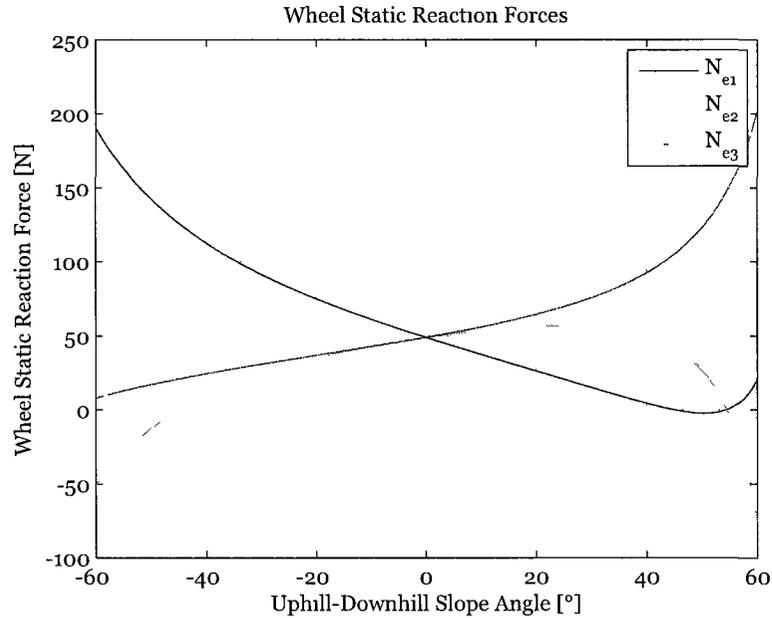


Figure 3.10: Wheel vertical reaction forces on an uphill-downhill slope where the rover’s center of gravity is at the rover centroid.

In the same way as for critical cross-hill stability angle γ_{ch} , the critical uphill-downhill static stability angle γ_{ud} can be found for various center of gravity locations. Figure 3.11 shows the center of gravity envelopes under which the wheels all maintain ground contact. For the critical uphill static stability angle shown in Figure 3.11a, the left-most vertical line and top-left diagonal line indicate the loss of contact at $e3$. The right-most vertical line represents the loss of contact with $e1$ and/or $e2$. For the critical downhill static stability angle shown in Figure 3.11b, the left-most vertical line and left-bottom diagonal line indicate the loss of contact at $e3$. The top-right diagonal line represents loss of contact with $e1$ and/or $e2$. To maintain full wheel contact at the angles indicated by the contour labels, the center of gravity must stay within the prescribed limits. An estimate of the center of gravity was taken for two extremes of manipulator positions: extended fully forwards as during sample collection; and deployed fully upright as the camera mast. These center of gravity locations are shown in Figure 3.11. With the manipulator extended fully forwards, the uphill

critical stability angle is -45° and the downhill critical stability angle is 38° ; with the manipulator acting as the camera mast, the uphill critical stability angle is 45° and the downhill critical stability angle is 45° .

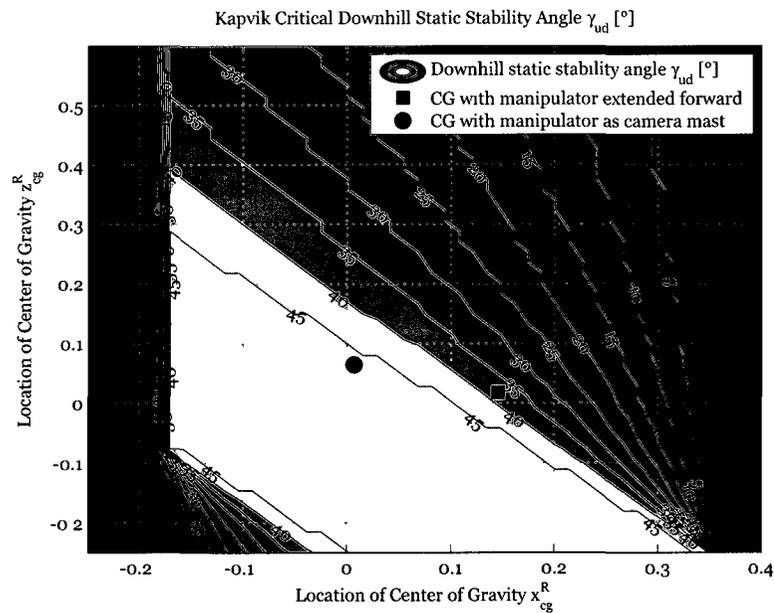
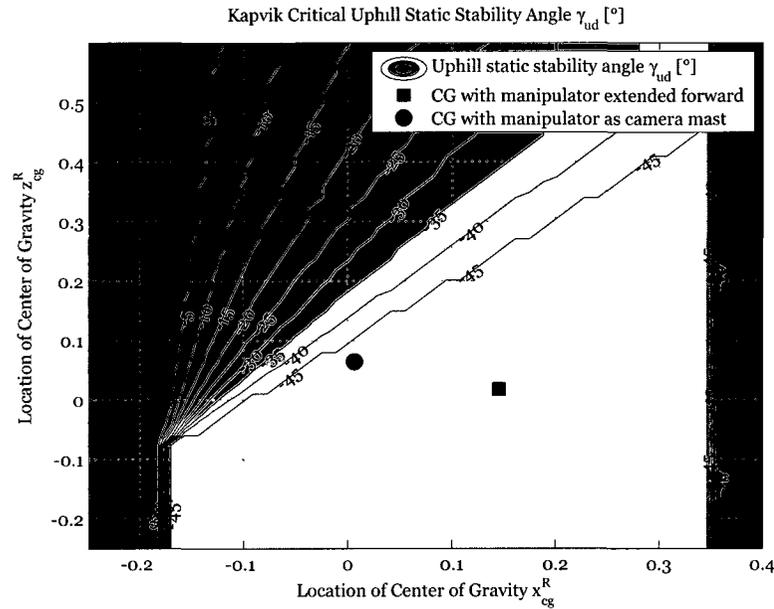


Figure 3.11: Critical uphill-downhill stability angles.

Chapter 4

Multibody Dynamics

This chapter first presents Lagrangian mechanics, which are used to simulate a rover equipped with a rocker-bogie mobility system traversing rolling terrain in two dimensions. The particulars of the rocker-bogie numerical simulation are then developed.

4.1 Lagrangian Mechanics

Lagrangian mechanics reformulate classical mechanics in terms of work and energy. These formulations are useful in multibody dynamic simulations and are typically used in manipulator simulations. A rover with a rocker-bogie suspension system comprises an articulated multibody system with a moving base [18]. As a result, the principles of Lagrangian mechanics are also useful in simulation of the rocker-bogie mobility system. The Lagrangian L of a system is defined as:

$$L = T - V \tag{4.1}$$

The term T represents the kinetic energy of the system [33]:

$$T = \sum_{i=1}^{n_b} \frac{1}{2} (m_i \mathbf{v}_i^T \mathbf{v}_i + \boldsymbol{\omega}_i^T \mathbf{I}_i \boldsymbol{\omega}_i) \tag{4.2}$$

for n_b bodies, where body i has mass m_i , absolute velocity \mathbf{v}_i at its center of mass, angular velocity $\boldsymbol{\omega}_i$ and inertia tensor \mathbf{I}_i .

The term V represents the potential energy of the system. In this thesis, this term represents the gravitational potential energy [33]:

$$V = g \begin{bmatrix} z_1 & z_2 & \dots & z_{n_b} \end{bmatrix} \begin{bmatrix} m_1 & m_2 & \dots & m_{n_b} \end{bmatrix}^T \quad (4.3)$$

where g is the local gravitational acceleration, $\{z_1, z_2, \dots, z_{n_b}\}$ are the heights of the mass centers of each body in world co-ordinates, and $\{m_1, m_2, \dots, m_{n_b}\}$ are the masses of each body.

The Euler-Lagrange equation gives utility to the Lagrangian [33]:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\boldsymbol{\Phi}}} - \frac{\partial L}{\partial \boldsymbol{\Phi}} = \boldsymbol{\tau}_g \quad (4.4)$$

where the system has n degrees of freedom, $\boldsymbol{\Phi}$ is a set of generalized co-ordinates $\{\Phi_1, \Phi_2, \dots, \Phi_n\}$, $\dot{\boldsymbol{\Phi}}$ is the generalized velocity vector, and $\boldsymbol{\tau}_g$ is a vector of generalized forces acting on the system.

If the Jacobian matrices of velocities \mathbf{v}_i and $\boldsymbol{\omega}_i$ with respect to generalized velocity $\dot{\boldsymbol{\Phi}}$ are given by $\mathbf{J}_{\mathbf{v}_i}$ and $\mathbf{J}_{\boldsymbol{\omega}_i}$ respectively, then Equation 4.2 becomes [33]:

$$\begin{aligned} T &= \sum_{i=1}^{n_b} \frac{1}{2} \left(m_i (\mathbf{J}_{\mathbf{v}_i} \dot{\boldsymbol{\Phi}})^T (\mathbf{J}_{\mathbf{v}_i} \dot{\boldsymbol{\Phi}}) + (\mathbf{J}_{\boldsymbol{\omega}_i} \dot{\boldsymbol{\Phi}})^T \mathbf{I}_i (\mathbf{J}_{\boldsymbol{\omega}_i} \dot{\boldsymbol{\Phi}}) \right) \\ T &= \sum_{i=1}^{n_b} \frac{1}{2} \left(m_i \dot{\boldsymbol{\Phi}}^T \mathbf{J}_{\mathbf{v}_i}^T \mathbf{J}_{\mathbf{v}_i} \dot{\boldsymbol{\Phi}} + \dot{\boldsymbol{\Phi}}^T \mathbf{J}_{\boldsymbol{\omega}_i}^T \mathbf{I}_i \mathbf{J}_{\boldsymbol{\omega}_i} \dot{\boldsymbol{\Phi}} \right) \\ T &= \frac{1}{2} \dot{\boldsymbol{\Phi}}^T \left(\sum_{i=1}^{n_b} m_i \mathbf{J}_{\mathbf{v}_i}^T \mathbf{J}_{\mathbf{v}_i} + \mathbf{J}_{\boldsymbol{\omega}_i}^T \mathbf{I}_i \mathbf{J}_{\boldsymbol{\omega}_i} \right) \dot{\boldsymbol{\Phi}} \\ T &= \frac{1}{2} \dot{\boldsymbol{\Phi}}^T \mathbf{H} \dot{\boldsymbol{\Phi}} \end{aligned} \quad (4.5)$$

where \mathbf{H} is an $n \times n$ symmetric system inertia matrix given by the term in parentheses on the second to last line.

The potential energy V is not dependent on generalized velocity $\dot{\Phi}$, so Equation 4.4 becomes [33]:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\Phi}} - \frac{\partial L}{\partial \Phi} = \mathbf{H}\ddot{\Phi} + \dot{\mathbf{H}}\dot{\Phi} - \frac{\partial L}{\partial \Phi} = \tau_g \quad (4.6)$$

where $\ddot{\Phi}$ is the generalized acceleration vector.

Algebraic constraints Ψ to the motion of the system can be implemented as follows [33, 34]:

$$\text{Position:} \quad \Psi(\Phi, t) = 0 \quad | \quad \Psi \in R^{n_c} \quad (4.7)$$

$$\text{Velocity:} \quad \dot{\Psi}(\Phi, t) = \Psi_t(\Phi, t) + \Psi_{\Phi}(\Phi, t)\dot{\Phi} = 0 \quad | \quad \Psi_{\Phi} \in R^{n_c \times n} \quad (4.8)$$

$$\text{Acceleration:} \quad \ddot{\Psi}(\Phi, t) = \dot{\Psi}_t(\Phi, t) + \dot{\Psi}_{\Phi}(\Phi, t)\dot{\Phi} + \Psi_{\Phi}(\Phi, t)\ddot{\Phi} = 0 \quad (4.9)$$

where Ψ_{Φ} is the Jacobian of the constraints Ψ with respect to configuration Φ , Ψ_t is the derivative of the constraints Ψ with respect to time t , and n_c is the number of constraints on the system.

The principles of constraints and virtual power can be combined to solve for the acceleration of the system. Firstly, the generalized forces τ_g in Equation 4.6 can be broken up into applied forces τ_a and constraint forces τ_c . The Euler-Lagrange equation then becomes [33]:

$$\mathbf{H}\ddot{\Phi} + \dot{\mathbf{H}}\dot{\Phi} - \frac{\partial L}{\partial \Phi} = \tau_a + \tau_c \quad (4.10)$$

The principle of virtual power states that the constraint forces τ_c deliver zero power along the possible directions of freedom of velocity [33]. In other words the dot

product $\boldsymbol{\tau}_c \cdot \dot{\boldsymbol{\Phi}}$ is equal to zero. The generalized constraint forces $\boldsymbol{\tau}_c$ can be set to [33]:

$$\boldsymbol{\tau}_c = \boldsymbol{\Psi}_{\boldsymbol{\Phi}}^T \boldsymbol{\lambda} \quad (4.11)$$

where $\boldsymbol{\lambda} \in R^{n_c}$ is a set of Lagrange multipliers $\{\lambda_1, \lambda_2, \dots, \lambda_{n_c}\}$ that form a linear combination of the columns of $\boldsymbol{\Psi}_{\boldsymbol{\Phi}}$ and allow the constraints to be satisfied. The Lagrange multipliers determine the magnitude of the force that acts along each constraint “direction”.

From here, Equation 4.11 can be substituted into Equation 4.10 to obtain:

$$\mathbf{H}\ddot{\boldsymbol{\Phi}} + \dot{\mathbf{H}}\dot{\boldsymbol{\Phi}} - \frac{\partial L}{\partial \boldsymbol{\Phi}} = \boldsymbol{\tau}_a + \boldsymbol{\Psi}_{\boldsymbol{\Phi}}^T \boldsymbol{\lambda} \quad (4.12)$$

Equation 4.12 can be combined with Equation 4.9 to form a system of $(n + n_c)$ equations in $(n + n_c)$ unknowns [33, 34]:

$$\begin{bmatrix} \mathbf{H} & \boldsymbol{\Psi}_{\boldsymbol{\Phi}}^T \\ \boldsymbol{\Psi}_{\boldsymbol{\Phi}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{\Phi}} \\ -\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau}_a - \dot{\mathbf{H}}\dot{\boldsymbol{\Phi}} + \frac{\partial L}{\partial \boldsymbol{\Phi}} \\ -\dot{\boldsymbol{\Psi}}_t - \dot{\boldsymbol{\Psi}}_{\boldsymbol{\Phi}}\dot{\boldsymbol{\Phi}} \end{bmatrix} \quad (4.13)$$

If the n_c constraint equations are independent, Equation 4.13 can be solved through matrix inversion [34]. However, enforcing the constraints using the acceleration equations leads to errors when the equations are integrated numerically. This is because $\ddot{\boldsymbol{\Psi}}(\boldsymbol{\Phi}, t) = 0$ has the following general solution [34]:

$$\boldsymbol{\Psi}(\boldsymbol{\Phi}, t) = \mathbf{a}_1 t + \mathbf{a}_2 \quad (4.14)$$

where \mathbf{a}_1 and \mathbf{a}_2 are constant vectors dependent on initial conditions.

If the initial conditions satisfy the constraints then both \mathbf{a}_1 and \mathbf{a}_2 will be zero vectors. However, due to roundoff errors during the simulation process, these vectors will assume non-zero values. Due to the $\mathbf{a}_1 t$ term in Equation 4.14, a non-zero vector \mathbf{a}_1 will cause a continuing increase in the motion constraints $\Psi(\Phi, t)$, leading to growing constraint violations [34]. In order to stabilize these equations during numerical simulation, Baumgarte stabilization is used. Baumgarte stabilization incorporates not only the second time derivative of the constraints $\ddot{\Psi}(\Phi, t)$, but also the first time derivative of constraints $\dot{\Psi}(\Phi, t)$ and the constraints $\Psi(\Phi, t)$ themselves. In Baumgarte stabilization, Equation 4.9 is replaced with the following [34]:

$$\ddot{\Psi}(\Phi, t) + 2\alpha\dot{\Psi}(\Phi, t) + \beta^2\Psi(\Phi, t) = 0 \quad (4.15)$$

where α and β are appropriately chosen constants.

The general solution to Equation 4.15 is [34]:

$$\Psi(\Phi, t) = \mathbf{a}_1 e^{s_1 t} + \mathbf{a}_2 e^{s_2 t} \quad (4.16)$$

where \mathbf{a}_1 and \mathbf{a}_2 are again constant vectors depending on initial conditions, and s_1 and s_2 are roots of the characteristic equation.

Setting α and β to positive constants ensures that both s_1 and s_2 will have a negative real part; this guarantees that Equation 4.16 is stable about zero [34]. The constants α and β are usually set to equal one another [34]. Substituting Equations 4.8 and 4.9 into Equation 4.15 gives:

$$\dot{\Psi}_t + \dot{\Psi}_\Phi \dot{\Phi} + \Psi_\Phi \ddot{\Phi} + 2\alpha \left(\Psi_t + \Psi_\Phi \dot{\Phi} \right) + \beta^2 \Psi = 0 \quad (4.17)$$

Rearranging Equation 4.17 for $\Psi_\Phi \ddot{\Phi}$ and substituting it into Equation 4.13, the

Baumgarte stabilized system of equations that represents the dynamic system is:

$$\begin{bmatrix} \mathbf{H} & \Psi_{\Phi}^T \\ \Psi_{\Phi} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\Phi} \\ -\lambda \end{bmatrix} = \begin{bmatrix} \tau_{\alpha} - \dot{\mathbf{H}}\dot{\Phi} + \frac{\partial L}{\partial \Phi} \\ -\dot{\Psi}_t - \dot{\Psi}_{\Phi}\dot{\Phi} - 2\alpha(\Psi_t + \Psi_{\Phi}\dot{\Phi}) - \beta^2\Psi \end{bmatrix} \quad (4.18)$$

When solved through matrix inversion, Equation 4.18 allows the accelerations in generalized co-ordinates $\ddot{\Phi}$ to be solved for in a manner that both satisfies the constraints and is immune to the destabilizing effects of numerical roundoff error.

4.2 Two-Dimensional Rocker-Bogie Simulation

A rover with a rocker-bogie suspension system comprises an articulated multibody system with a moving base. With slight modifications to include the moving base, Lagrangian mechanics can be used to deduce Equation 4.18 for an articulated rover driving over rolling terrain. The most challenging environment in terms of traction is one in which the rover is traversing slopes. For this reason, the two dimensions in which the simulation is performed are the world horizontal direction X_w and world vertical direction Z_w . The third axis in the Cartesian co-ordinate system Y_w is directed into the page, making clockwise the positive direction for rotational values.

In the two-dimensional simulation, a single side of the rocker-bogie suspension system is considered. This is equivalent to the three dimensional case in which the terrain extends infinitely into the page, and is a good approximation for slowly changing terrain on which the roll of the rover is negligible. The rocker-bogie's differential mechanism ensures that the rocker angle on one side is equal and opposite to the rocker angle on the other side. Given the assumption that the terrain extends infinitely into the page, the configuration of the suspension will be identical on both sides. This means that in order to satisfy the opposite angle constraint for rocker joint angle, the

rocker angle must be zero on each side. As a result, the rocker is treated as part of the body.

4.2.1 Co-ordinates, Naming Convention, and Configuration

The co-ordinates and naming conventions for the simulation are shown in Figure 4.1. The combined center of mass of the body and the rocker is at 0 ; the rocker joint is designated r ; the bogie joint/link is denoted 1 ; the wheel joints/links are given by 2 , 3 , and 4 ; joints/links 2 , 3 , and 4 are also referred to as wheels 1 , 2 , and 3 respectively; the wheel-terrain contact points are called $e1$, $e2$, and $e3$. A vector \mathbf{l}_{ij} represents the vector from i to j when the rover is resting on level ground. The vector \mathbf{c}_{11} represents the vector from joint 1 to the center of mass of the associated bogie link. The wheel centers of mass are located at the wheel joints, so the vectors from the wheel joints to their centers of mass are zero (i.e. $\mathbf{c}_{22} = \mathbf{c}_{33} = \mathbf{c}_{44} = \mathbf{0}$). The angular displacement of the rover body from the world horizontal, also termed pitch, is given by θ_{0_y} . Joint angles $\{q_1, q_2, q_3, q_4\}$ represent the angle of rotation of the bogie and three wheels respectively. The wheel-terrain contact angles from the horizontal at $e1$, $e2$, and $e3$ are given by $\{\gamma_1, \gamma_2, \gamma_3\}$ respectively. In deformable terrain, the wheel does not have a single discrete contact point. However, in simulation, a single effective point of contact can be used [5].

The configuration of the rover at any given time can be fully described by the vector:

$$\Phi = \left[x_0 \quad y_0 \quad z_0 \quad \theta_{0_x} \quad \theta_{0_y} \quad \theta_{0_z} \quad q_1 \quad q_2 \quad q_3 \quad q_4 \right]^T \quad (4.19)$$

where $\{x_0, y_0, z_0\}$ are the Cartesian co-ordinates of the combined body/rocker center of mass in the world coordinate system, $\{\theta_{0_x}, \theta_{0_y}, \theta_{0_z}\}$ are the rotation angles of the body about the world $\{X_w, Y_w, Z_w\}$ axes respectively, and $\{q_1, q_2, q_3, q_4\}$ are the joint

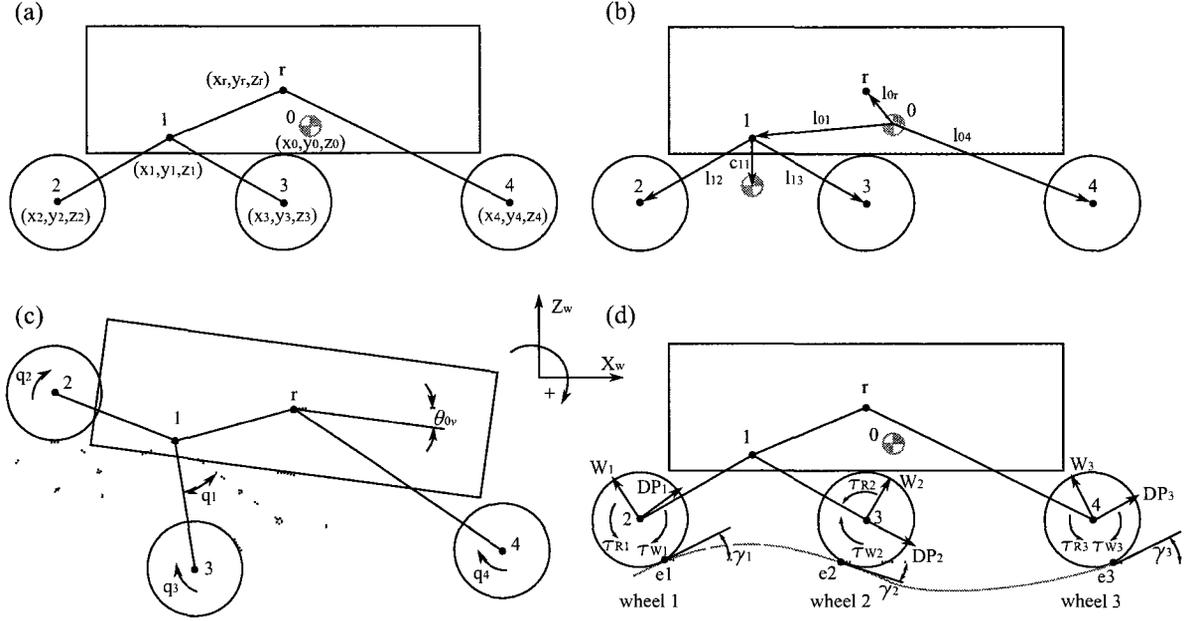


Figure 4.1: Two-dimensional rocker-bogie system configuration variables, vectors, and forces.

angles. Note that co-ordinates y_0 , θ_{0_x} , and θ_{0_z} are not required to describe the two-dimensional system. These co-ordinates are included because they facilitate easy rotations and cross products.

4.2.2 Formulation of Dynamic Equations

The goal of the following section is to describe the two-dimensional rocker-bogie system by the matrix relation in Equation 4.18 in order to solve for $\ddot{\Phi}$. The methodology to do this is developed in a similar manner as in Section 4.1.

The rocker-bogie mobility system as simulated in two dimensions consists of a moving base (the lumped body and rocker 0), four links (the bogie 1 and the three wheels 2 , 3 , and 4), and four revolute joints (the bogie joint 1 and the three wheel joints 2 , 3 , and 4). Therefore the number of bodies n_b is five and number of joints n_v is four.

The kinetic energy, required for the Lagrangian, L is then:

$$\begin{aligned}
T &= \sum_{i=0}^{n_v} \frac{1}{2} (m_i \mathbf{v}_i^T \mathbf{v}_i + \boldsymbol{\omega}_i^T \mathbf{I}_i \boldsymbol{\omega}_i) \\
T &= \frac{1}{2} \dot{\boldsymbol{\Phi}}^T \left(\sum_{i=0}^{n_v} m_i \mathbf{J}_{v_i}^T \mathbf{J}_{v_i} + \mathbf{J}_{\omega_i}^T \mathbf{I}_i \mathbf{J}_{\omega_i} \right) \dot{\boldsymbol{\Phi}} \\
T &= \frac{1}{2} \dot{\boldsymbol{\Phi}}^T \mathbf{H} \dot{\boldsymbol{\Phi}}
\end{aligned} \tag{4.20}$$

where $n_v = 4$.

In order to solve for the system inertia matrix \mathbf{H} it is necessary to solve for the linear and angular velocity Jacobians \mathbf{J}_{v_i} and \mathbf{J}_{ω_i} for each body i . To solve for \mathbf{J}_{v_i} and \mathbf{J}_{ω_i} , an expression for the linear and angular velocities \mathbf{v}_i and $\boldsymbol{\omega}_i$ of the mass centers of each body is required. The velocities must be described in terms of generalized velocity $\dot{\boldsymbol{\Phi}}$ for the rover body ($i = 0$) and all links ($i = \{1, 2, 3, 4\}$).

The velocity and angular velocity Jacobians for the body 0 are easily deduced since its velocity and angular velocity are explicitly included in the generalized velocity vector $\dot{\boldsymbol{\Phi}}$.

$$\mathbf{J}_{v_0} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 4} \end{bmatrix} \tag{4.21}$$

$$\mathbf{J}_{\omega_0} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 4} \end{bmatrix} \tag{4.22}$$

where $\mathbf{I}_{p \times p}$ is a $p \times p$ identity matrix, $\mathbf{0}_{p \times p}$ is a $p \times p$ zero matrix.

Finding the velocity and angular velocity Jacobians of the links 1 , 2 , 3 and 4 is more involved. Define the position of joint i to be \mathbf{p}_i . Define the position of the center of mass of the link corresponding to joint i to be \mathbf{p}_{c_i} . Define the position of point 0 to be \mathbf{r}_0 . These vectors all refer to positions in world co-ordinates. The velocity of the

center of mass of a single link \mathbf{v}_i is given by:

$$\mathbf{v}_i = \mathbf{v}_0 + \boldsymbol{\omega}_0 \times (\mathbf{p}_{c_i} - \mathbf{r}_0) + \sum_{j=1}^{n_v} \mathbf{L}_{i,j} \mathbf{N}_j \times (\mathbf{p}_{c_i} - \mathbf{p}_j) \dot{q}_j \quad | \quad i = \{1, 2, 3, 4\} \quad (4.23)$$

where \mathbf{v}_0 is the velocity of the lumped body/rocker θ , $\boldsymbol{\omega}_0$ is the angular velocity of the lumped body/rocker θ , \mathbf{L} is a matrix where $\mathbf{L}_{i,j} = 1$ if j is on the kinematic chain from θ to i and $\mathbf{L}_{i,j} = 0$ otherwise, \mathbf{N}_j is the normal vector along the axis of rotation of joint j , and \dot{q}_j is the time rate of change of joint angle for joint j .

The angular velocity of joint i is given by:

$$\boldsymbol{\omega}_i = \boldsymbol{\omega}_0 + \sum_{j=1}^{n_v} \mathbf{L}_{i,j} \mathbf{N}_j \dot{q}_j \quad (4.24)$$

The matrix \mathbf{L} for the two-dimensional rocker-bogie system is shown in Appendix C.3.

A skew symmetric matrix can be used to turn the cross product into matrix multiplication. With two vectors $\mathbf{a} = [a_1 \ a_2 \ a_3]^T$ and $\mathbf{b} = [b_1 \ b_2 \ b_3]^T$, a skew symmetric matrix $\tilde{\mathbf{a}}$ can be constructed with the following properties [18]:

$$\mathbf{a} \times \mathbf{b} = \tilde{\mathbf{a}} \mathbf{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}, \quad \mathbf{b} \times \mathbf{a} = -\tilde{\mathbf{a}} \mathbf{b} = \tilde{\mathbf{a}}^T \mathbf{b} \quad (4.25)$$

Using the insights from above, the Jacobians describing the velocity of the center of mass and angular velocity for a single link can be found such that $\mathbf{v}_i = \mathbf{J}_{v_i} \dot{\Phi}$ and $\boldsymbol{\omega}_i = \mathbf{J}_{\omega_i} \dot{\Phi}$:

$$\mathbf{J}_{v_i} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \tilde{\mathbf{r}}_{0i}^T & \mathbf{L}_{i,1} \mathbf{N}_1 \times (\mathbf{p}_{c_i} - \mathbf{p}_1) & \dots & \mathbf{L}_{i,4} \mathbf{N}_4 \times (\mathbf{p}_{c_i} - \mathbf{p}_4) \end{bmatrix} \quad (4.26)$$

$$\mathbf{J}_{\omega_i} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{L}_{i,1} \mathbf{N}_1 & \dots & \mathbf{L}_{i,4} \mathbf{N}_4 \end{bmatrix} \quad (4.27)$$

where $\mathbf{r}_{0i} = \mathbf{p}_{c_i} - \mathbf{r}_0$ is the vector from 0 to the center of mass of link $i = \{2, 3, 4\}$.

Stacking the Jacobians allows for the creation of a link Jacobian \mathbf{J}_a which when multiplied by the generalized velocity vector $\dot{\Phi}$ gives a vector containing the full set of link center of mass velocities $\dot{\mathbf{X}}_a$. These velocities are useful in net traction estimation.

$$\dot{\mathbf{X}}_a = \begin{bmatrix} \mathbf{v}_1 \\ \boldsymbol{\omega}_1 \\ \vdots \\ \mathbf{v}_{n_v} \\ \boldsymbol{\omega}_{n_v} \end{bmatrix} = \mathbf{J}_a \dot{\Phi} = \begin{bmatrix} \mathbf{J}_{v_1} \\ \mathbf{J}_{\omega_1} \\ \vdots \\ \mathbf{J}_{v_{n_v}} \\ \mathbf{J}_{\omega_{n_v}} \end{bmatrix} \dot{\Phi} \quad (4.28)$$

The MATLAB code used to evaluate \mathbf{J}_a is included in Appendix C.6.

To find the system inertia matrix \mathbf{H} , the values of \mathbf{J}_{v_i} and \mathbf{J}_{ω_i} derived above for the body and each link are substituted into the expression for \mathbf{H} implied by Equation 4.20. Note that the system inertia matrix \mathbf{H} is exclusively a function of the configuration vector Φ .

$$\mathbf{H} = \sum_{i=0}^{n_v} m_i \mathbf{J}_{v_i}^T \mathbf{J}_{v_i} + \mathbf{J}_{\omega_i}^T \mathbf{I}_i \mathbf{J}_{\omega_i} \quad (4.29)$$

The matrix \mathbf{H} for this system is shown in Appendix C.7.

The gravitational potential energy is found by applying Equation 4.3 to the two-dimensional rocker-bogie system:

$$V = g \begin{bmatrix} p_{c0z} & p_{c1z} & p_{c2z} & p_{c3z} & p_{c4z} \end{bmatrix} \begin{bmatrix} m_0 & m_1 & m_2 & m_3 & m_4 \end{bmatrix}^T \quad (4.30)$$

where p_{ci_z} is the z component of position of the center of mass of body i in world co-ordinates.

To finish the formulation of Equation 4.18 the following additional variables are required: $\dot{\mathbf{H}}$, Ψ , Ψ_t , $\dot{\Psi}_t$, Ψ_Φ , $\dot{\Psi}_\Phi$, and $\frac{\partial L}{\partial \Phi}$.

The time rate of change of the system inertia matrix $\dot{\mathbf{H}}$ is found by applying chain rule differentiation to Equation 4.29.

$$\dot{\mathbf{H}} = \sum_{i=0}^{n_v} m_i \frac{\partial}{\partial t} \mathbf{J}_{v_i}^T \mathbf{J}_{v_i} + m_i \mathbf{J}_{v_i}^T \frac{\partial}{\partial t} \mathbf{J}_{v_i} + \frac{\partial}{\partial t} \mathbf{J}_{\omega_i}^T \mathbf{I}_i \mathbf{J}_{\omega_i} + \mathbf{J}_{\omega_i}^T \mathbf{I}_i \frac{\partial}{\partial t} \mathbf{J}_{\omega_i} \quad (4.31)$$

The MATLAB code used to evaluate $\dot{\mathbf{H}}$ for this system is shown in Appendix C.8.

The constraints on the system Ψ are found by considering that the wheels must follow the terrain. A path for the wheel centroids is created using a continuously differentiable function $z_{wc}(x_{wc})$. The terrain is then formed by creating an offset function from the wheel centroid path. This offset function is displaced one wheel radius r_w downwards from the wheel centroid path. The wheel joints (2, 3, and 4) are constrained to the wheel centroid path $z_{wc}(x_{wc})$. Given a wheel centroid path $z_{wc}(x_{wc})$, the corresponding terrain co-ordinates x_t and z_t are given by the offset

function:

$$\begin{bmatrix} x_t \\ z_t \end{bmatrix} = \begin{bmatrix} x_{wc} \\ z_{wc}(x_{wc}) \end{bmatrix} + r_w \frac{\begin{bmatrix} \frac{dz_{wc}}{dx_{wc}}|_{x_{wc}} \\ -1 \end{bmatrix}}{\sqrt{1 + \frac{dz_{wc}}{dx_{wc}}^2}} \quad (4.32)$$

The motion constraints Ψ for the two-dimensional rocker-bogie simulation are:

$$\Psi = \begin{bmatrix} \Psi_1 \\ \Psi_2 \\ \Psi_3 \end{bmatrix} = \begin{bmatrix} p_{2z} - z_{wc}(p_{2x}) \\ p_{3z} - z_{wc}(p_{3x}) \\ p_{4z} - z_{wc}(p_{4x}) \end{bmatrix} = \mathbf{0}_{3 \times 1} \quad (4.33)$$

where p_{i_z} is the z component of joint i 's position and p_{i_x} is the x component of joint i 's position.

The MATLAB code used to evaluate the rover constraints is included in Appendix C.9. These constraints assure that the wheel centroids follow the wheel centroid path. Figure 4.2 illustrates the wheel centroid path and the corresponding offset function representing the terrain.

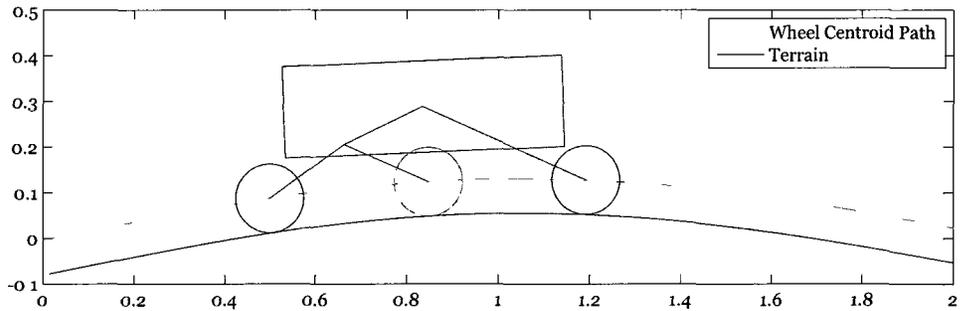


Figure 4.2: Wheel constraints.

Since the constraints are unchanging with time, $\Psi_t = \mathbf{0}$; thus $\dot{\Psi}_t = \mathbf{0}$ as well. The Jacobian of the constraints with respect to the configuration vector Ψ_{Φ} is non-zero since the constraints Ψ are dependent on the configuration vector Φ through the terms p_{i_z} and p_{i_x} .

$$\Psi_{\Phi} = \begin{bmatrix} \frac{\partial \Psi}{\partial x_0} & \frac{\partial \Psi}{\partial y_0} & \frac{\partial \Psi}{\partial z_0} & \frac{\partial \Psi}{\partial \theta_{0_x}} & \frac{\partial \Psi}{\partial \theta_{0_y}} & \frac{\partial \Psi}{\partial \theta_{0_z}} & \frac{\partial \Psi}{\partial q_1} & \frac{\partial \Psi}{\partial q_2} & \frac{\partial \Psi}{\partial q_3} & \frac{\partial \Psi}{\partial q_4} \end{bmatrix} \quad (4.34)$$

The term $\dot{\Psi}_{\Phi}$ is found by differentiating Equation 4.34 with respect to time. The MATLAB code used to evaluate Ψ_{Φ} and $\dot{\Psi}_{\Phi}$ is included in Appendices C.10 and C.11 respectively.

To calculate the derivative of the Lagrangian with respect to the configuration vector $\frac{\partial L}{\partial \Phi}$, the partial derivatives of kinetic energy T and potential energy V with respect to configuration Φ are required.

$$\frac{\partial L}{\partial \Phi} = \frac{\partial T}{\partial \Phi} - \frac{\partial V}{\partial \Phi} \quad (4.35)$$

The partial derivative of kinetic energy T with respect to configuration Φ is given by:

$$\begin{aligned} \frac{\partial T}{\partial \Phi} &= \frac{1}{2} \dot{\Phi}^T \frac{\partial \mathbf{H}}{\partial \Phi} \dot{\Phi} \\ \frac{\partial T}{\partial \Phi} &= \frac{1}{2} \dot{\Phi}^T \frac{\partial}{\partial \Phi} \left(\sum_{i=0}^{n_v} m_i \mathbf{J}_{v_i}^T \mathbf{J}_{v_i} + \mathbf{J}_{\omega_i}^T \mathbf{I}_i \mathbf{J}_{\omega_i} \right) \dot{\Phi} \\ \frac{\partial T}{\partial \Phi} &= \frac{1}{2} \dot{\Phi}^T \left(\sum_{i=0}^{n_v} m_i \frac{\partial \mathbf{J}_{v_i}^T}{\partial \Phi} \mathbf{J}_{v_i} + m_i \mathbf{J}_{v_i}^T \frac{\partial \mathbf{J}_{v_i}}{\partial \Phi} \dots \right. \\ &\quad \left. \dots + \frac{\partial \mathbf{J}_{\omega_i}^T}{\partial \Phi} \mathbf{I}_i \mathbf{J}_{\omega_i} + \mathbf{J}_{\omega_i}^T \mathbf{I}_i \frac{\partial \mathbf{J}_{\omega_i}}{\partial \Phi} \right) \dot{\Phi} \end{aligned} \quad (4.36)$$

The partial derivative $\frac{\partial \mathbf{H}}{\partial \Phi}$ is performed item by item for all of the bodies in the system; the results are then summed. The output $\frac{\partial T}{\partial \Phi}$ is a $(6 + n_v) \times 1$ vector representing

the sensitivity of kinetic energy to each variable in the configuration vector Φ . The partial derivative of potential energy V with respect to configuration Φ is given by:

$$\frac{\partial V}{\partial \Phi} = g \left[\frac{\partial p_{c0z}}{\partial \Phi}^T \quad \frac{\partial p_{c1z}}{\partial \Phi}^T \quad \frac{\partial p_{c2z}}{\partial \Phi}^T \quad \frac{\partial p_{c3z}}{\partial \Phi}^T \quad \frac{\partial p_{c4z}}{\partial \Phi}^T \right] \left[m_0 \quad m_1 \quad m_2 \quad m_3 \quad m_4 \right]^T \quad (4.37)$$

The MATLAB code to evaluate $\frac{\partial T}{\partial \Phi}$ and $\frac{\partial V}{\partial \Phi}$ is included in Appendices C.12 and C.13 respectively. Apart from the generalized applied forces τ_a , considered in the next section, all of the terms required to solve the dynamic system using Equation 4.18 have been derived.

4.2.3 Applied Forces and Torques

The non-gravitational forces and torques applied to the rover are shown in Figure 4.1d. The drawbar pulls $\{DP_1, DP_2, DP_3\}$, normal load reaction forces $\{W_1, W_2, W_3\}$, motor torques $\{\tau_{W_1}, \tau_{W_2}, \tau_{W_3}\}$, and resistive torques $\{\tau_{R_1}, \tau_{R_2}, \tau_{R_3}\}$ are all applied at the wheel joints. The generalized forces τ_g are divided into applied forces τ_a and constraint forces $\tau_c = \Psi_{\Phi}^T \lambda$. The constraint forces τ_c deliver zero power and represent the effect of the normal load reaction forces $\{W_1, W_2, W_3\}$ in generalized co-ordinates. They are found by solving Equation 4.18 for λ . The Lagrange multipliers λ represent the magnitude of the constraint forces in the constraint “directions”. Note that $\frac{\partial \Psi}{\partial z_0} = 1$, meaning that λ_i represents the Z_w component of the normal reaction force W_i for wheel i .

$$\lambda_i = W_i \cos \gamma_i \quad | \quad i = \{1, 2, 3\} \quad (4.38)$$

The X_w component of W_i can also be found for wheel i :

$$\Psi_{\Phi_i,1} \lambda_i = -W_i \sin \gamma_i \quad | \quad i = \{1, 2, 3\} \quad (4.39)$$

The set of generalized applied forces $\boldsymbol{\tau}_a$ consists of the effect of resistive wheel torques $\boldsymbol{\tau}_R$, the effect of drawbar pulls $\boldsymbol{\tau}_{DP}$, and the effect of wheel torques $\boldsymbol{\tau}_W$. Note that generalized forces represent forces and moments in the co-ordinates of the configuration vector $\boldsymbol{\Phi}$.

$$\boldsymbol{\tau}_a = \boldsymbol{\tau}_R + \boldsymbol{\tau}_{DP} + \boldsymbol{\tau}_W \quad (4.40)$$

The generalized forces $\boldsymbol{\tau}_R$ and $\boldsymbol{\tau}_{DP}$ are external and the generalized forces $\boldsymbol{\tau}_W$ are internal. The generalized external forces $\boldsymbol{\tau}_R$ and $\boldsymbol{\tau}_{DP}$ are found using Algorithm 2.1 in combination with calculated normal loads, wheel centroid velocities, and wheel angular velocities. The external forces, which are applied at the wheel joints, can be expressed in world co-ordinates as follows:

$$\begin{aligned} \mathbf{F}_{ext} &= \left[\mathbf{F}_{w_1}^T \quad \mathbf{M}_{w_1}^T \quad \mathbf{F}_{w_2}^T \quad \mathbf{M}_{w_2}^T \quad \mathbf{F}_{w_3}^T \quad \mathbf{M}_{w_3}^T \right]^T \\ \mathbf{F}_{w_i}^T &= \left[F_{w_{ix}} \quad F_{w_{iy}} \quad F_{w_{iz}} \right] \quad | \quad i = \{1, 2, 3\} \\ \mathbf{M}_{w_i}^T &= \left[M_{w_{ix}} \quad M_{w_{iy}} \quad M_{w_{iz}} \right] \quad | \quad i = \{1, 2, 3\} \end{aligned} \quad (4.41)$$

where \mathbf{F}_{w_i} is the external linear force vector exerted on wheel joint i , and \mathbf{M}_{w_i} is the external moment exerted on wheel joint i ; both are expressed in the world frame $\{X_w, Y_w, Z_w\}$.

In order to be applied to the simulation, the external forces need to be converted from the world co-ordinate frame into the generalized co-ordinate frame of the configuration vector $\boldsymbol{\Phi}$. This is accomplished through the use of a wheel force Jacobian \mathbf{J}_w .

$$\boldsymbol{\tau}_R = \mathbf{J}_w^T \mathbf{F}_{\tau_R} \quad (4.42)$$

$$\boldsymbol{\tau}_{DP} = \mathbf{J}_w^T \mathbf{F}_{DP} \quad (4.43)$$

where \mathbf{F}_{τ_R} and \mathbf{F}_{DP} are the set of external forces and moments in the form of \mathbf{F}_{ext} .

The external force vector from resistive torques in world co-ordinates \mathbf{F}_{τ_R} is:

$$\mathbf{F}_{\tau_R} = \begin{bmatrix} 0 & 0 & 0 & 0 & -\tau_{R1} & 0 & 0 & 0 & 0 & 0 & -\tau_{R2} & 0 & 0 & 0 & 0 & 0 & -\tau_{R3} & 0 \end{bmatrix}^T \quad (4.44)$$

The external force vector from drawbar pulls in world co-ordinates \mathbf{F}_{DP} is:

$$\mathbf{F}_{DP} = \begin{bmatrix} DP_1 \cos \gamma_1 & 0 & -DP_1 \sin \gamma_1 & 0 & 0 & 0 & DP_2 \cos \gamma_2 & 0 & -DP_2 \sin \gamma_2 & \dots \\ \dots & 0 & 0 & 0 & DP_3 \cos \gamma_3 & 0 & -DP_3 \sin \gamma_3 & 0 & 0 & 0 \end{bmatrix}^T \quad (4.45)$$

The wheel force Jacobian \mathbf{J}_w is shown in Appendix C.5. Note that it is exclusively a function of configuration vector Φ .

The internal forces consist of the set of wheel torques $\{\tau_{W1}, \tau_{W2}, \tau_{W3}\}$ and viscous damping torques $\{b_d \dot{q}_2, b_d \dot{q}_3, b_d \dot{q}_4\}$ on joints 2, 3, and 4, where b_d is the experimentally determined viscous damping coefficient. The bogie joint, joint 1, is not actuated. A bearing facilitates smooth motion of this joint, and the internal torque on joint 1 is assumed to be negligible. The wheel torques can be applied directly to the generalized forces:

$$\boldsymbol{\tau}_W = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & (\tau_{W1} - b_d \dot{q}_2) & (\tau_{W2} - b_d \dot{q}_3) & (\tau_{W3} - b_d \dot{q}_4) \end{bmatrix}^T \quad (4.46)$$

Equation 4.40 can be rewritten as:

$$\boldsymbol{\tau}_a = \mathbf{J}_w^T \mathbf{F}_{\tau_R} + \mathbf{J}_w^T \mathbf{F}_{DP} + \boldsymbol{\tau}_W \quad (4.47)$$

Equation 4.12 can be rewritten as:

$$\mathbf{H}\ddot{\Phi} + \dot{\mathbf{H}}\dot{\Phi} - \frac{\partial L}{\partial \Phi} = \mathbf{J}_w^T \mathbf{F}_{\tau_R} + \mathbf{J}_w^T \mathbf{F}_{DP} + \boldsymbol{\tau}_W + \Psi_{\Phi}^T \boldsymbol{\lambda} \quad (4.48)$$

Finally, Equation 4.18 can be rewritten to the form used in simulation:

$$\begin{bmatrix} H & \Psi_{\Phi}^T \\ \Psi_{\Phi} & 0 \end{bmatrix} \begin{bmatrix} \ddot{\Phi} \\ -\lambda \end{bmatrix} = \begin{bmatrix} \mathbf{J}_w^T \mathbf{F}_{\tau_R} + \mathbf{J}_w^T \mathbf{F}_{DP} + \tau_W - \dot{H}\dot{\Phi} + \frac{\partial L}{\partial \Phi} \\ -\dot{\Psi}_t - \dot{\Psi}_{\Phi}\dot{\Phi} - 2\alpha \left(\Psi_t + \Psi_{\Phi}\dot{\Phi} \right) - \beta^2 \Psi \end{bmatrix} \quad (4.49)$$

4.2.4 Simulation Implementation

The result of the preceding section is a solution to the second time derivative of configuration $\ddot{\Phi}$. For the purposes of propagating forward the configuration, a ‘‘configuration state vector’’ Φ_s is created:

$$\Phi_s = \begin{bmatrix} \dot{\Phi} \\ \Phi \end{bmatrix} \quad \dot{\Phi}_s = \begin{bmatrix} \ddot{\Phi} \\ \dot{\Phi} \end{bmatrix} \quad (4.50)$$

Each evaluation of $\dot{\Phi}_s$ is a function of Φ_s , the applied wheel torques $\{\tau_{W_1}, \tau_{W_2}, \tau_{W_3}\}$, and the previously evaluated generalized constraint forces τ_c . The term $\ddot{\Phi}$ is acquired from the solution to Equation 4.49, and the term $\dot{\Phi}$ is acquired directly from the input argument Φ_s . Numerical integration is then used to propagate forward the configuration state vector Φ_s . A block diagram of the simulation implementation is shown in Figure 4.3.

In simulation, proportional integral derivative (PID) controllers were used to set the approximate wheel angular velocities \dot{q}_w to the desired velocity divided by the wheel radius $\frac{v_d}{r_w}$. The controller was run at a frequency of 50 Hz; the configuration state vector Φ_s and all values of interest were also recorded at this frequency. Because of the sensitivity of drawbar pull DP and resistive torque τ_R to slip i , the MATLAB ordinary differential equation solver `ode45` was used for integration between timesteps. Over the duration of a single timestep the control input torques $\{\tau_{W_1}, \tau_{W_2}, \tau_{W_3}\}$ and

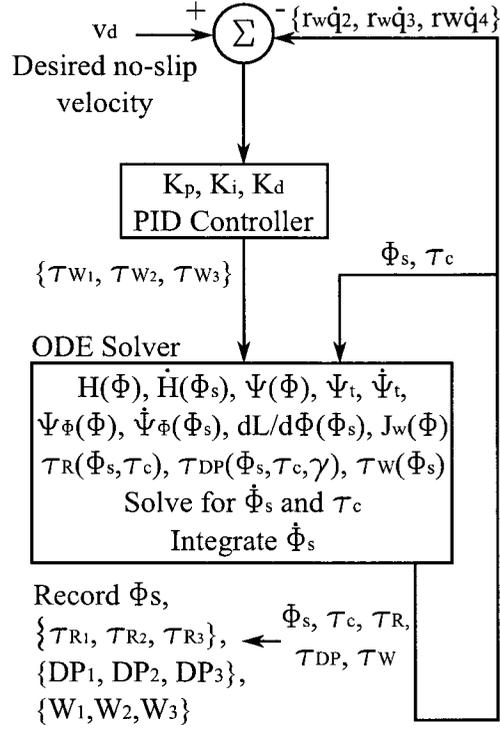


Figure 4.3: Block diagram of simulation implementation.

normal loads $\{W_1, W_2, W_3\}$ were presumed to be constant.

The partial derivative of kinetic energy with respect to configuration vector $\frac{\partial T}{\partial \Phi}$ involves the calculation of the partial derivative of the system inertia matrix $\frac{\partial H}{\partial \Phi}$. This step is computationally expensive, and was found to account for approximately one third of the computation time. The term $\frac{\partial H}{\partial \Phi}$ is typically very small, as is the generalized velocity $\dot{\Phi}$ for the slowly moving rover. As a result, the term $\frac{\partial T}{\partial \Phi}$ as defined in Equation 4.36 was found to be negligible and was excluded from the simulation in order to reduce computation time.

4.2.5 Simulation Parameters

The parameters used in the simulation were selected to represent one side of the *Kapvik* micro-rover and are shown in Table 4.1. Inertias were evaluated about the Y_w

axis only. The wheel dimensions from Table 2.1 were used. Gradually rolling terrain was created for simulation, with a wheel centroid path given by:

$$z_{wc}(x_{wc}) = 0.12 \cos\left(\frac{\pi}{2} x_{wc} - 1.6\right) + 0.01 x_{wc} \quad (4.51)$$

Table 4.1: Parameters used in simulation.

Parameter	Value	Units	Source
Mass of body and rocker m_0	12.49	kg	SolidWorks model
Mass of link 1 m_1	1.562	kg	SolidWorks model
Mass of links 2, 3, and 4 m_2, m_3, m_4 (wheel mass)	0.3174	kg	SolidWorks model
Inertia of body and rocker I_0	0.5045	kg m ²	SolidWorks model
Inertia of link 1 I_1	0.05248	kg m ²	SolidWorks model
Wheel inertia I_w	0.003584	kg m ²	SolidWorks model and drivetrain manufacturers
Wheel and gear inertia referred to wheel I_{wg}	1.082	kg m ²	SolidWorks model and drivetrain manufacturers
Wheel angular viscous damping coefficient b_d	2.28855	N m s	Experiment
Vector \mathbf{c}_{11}	$\begin{bmatrix} 0 \\ 0 \\ -0.07322 \end{bmatrix}$	m	SolidWorks model
Vector $\mathbf{c}_{22} = \mathbf{c}_{33} = \mathbf{c}_{44}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	m	SolidWorks model
Baumgarte stabilization constant α	500	-	Trial and error
Baumgarte stabilization constant β	500	-	Trial and error
Desired velocity v_d	2	cm/s	Design speed
Proportional control gain K_p	25	-	Trial and error
Integral control gain K_i	0.5	-	Trial and error
Derivative control gain K_d	12.5	-	Trial and error

4.2.6 Simulation Results

The simulation was run for 200 s at a desired velocity of 2 cm/s following the implementation details outlined in Section 4.2.4, and the simulation parameters shown in Section 4.2.5. The results of the simulation are shown in Figures 4.4 and 4.5. The simulation performs well and produces results within the range of expected magnitudes; this simulation is used later in this thesis to test the net traction estimation algorithm.

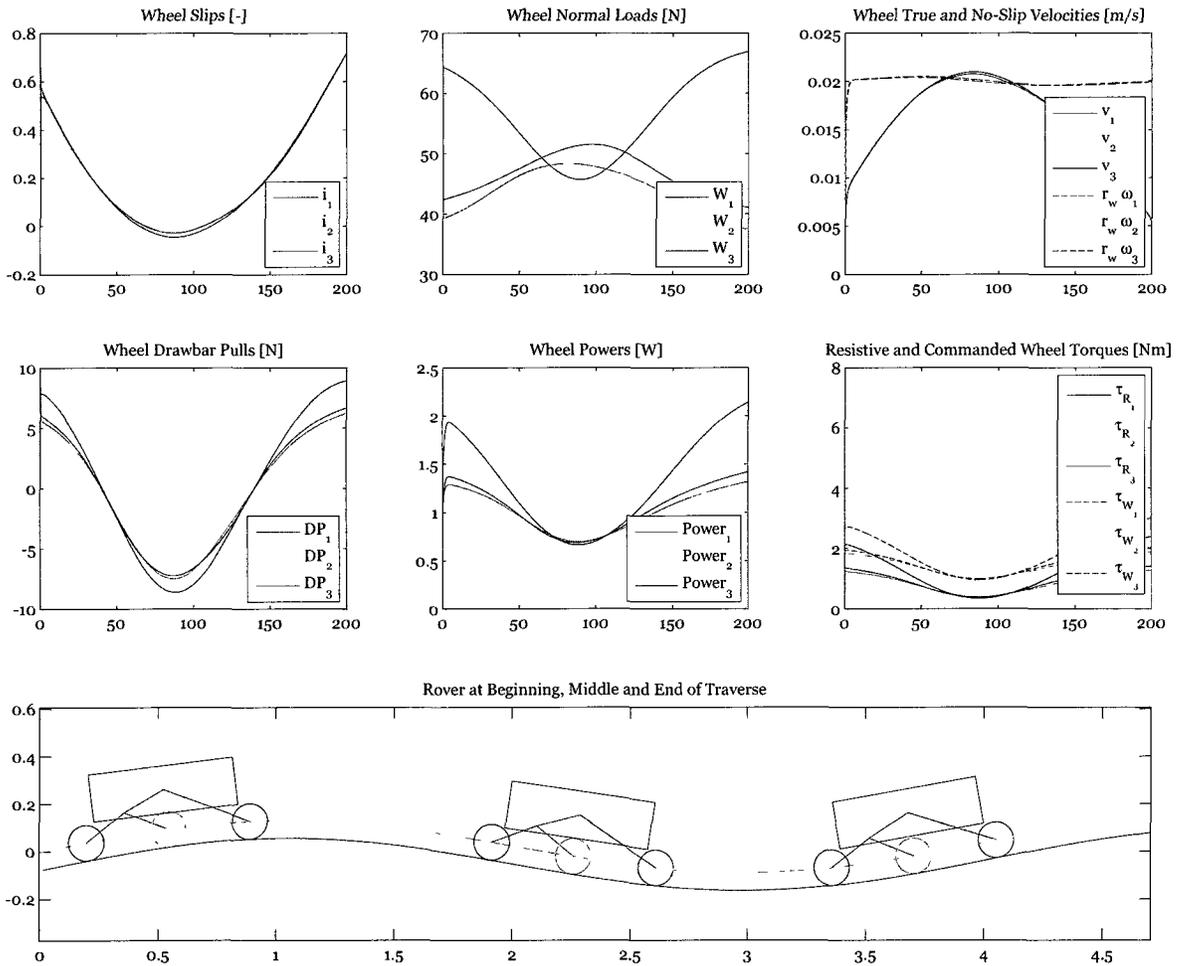


Figure 4.4: Results from the 200 s simulation of the rover traversing gently rolling terrain. Time is shown on the x axis in seconds. Wheel powers were calculated from the mechanical power delivered and Joule losses to heat caused by current flowing through the motor windings; the drivetrain's mechanical efficiency was incorporated into the calculation.

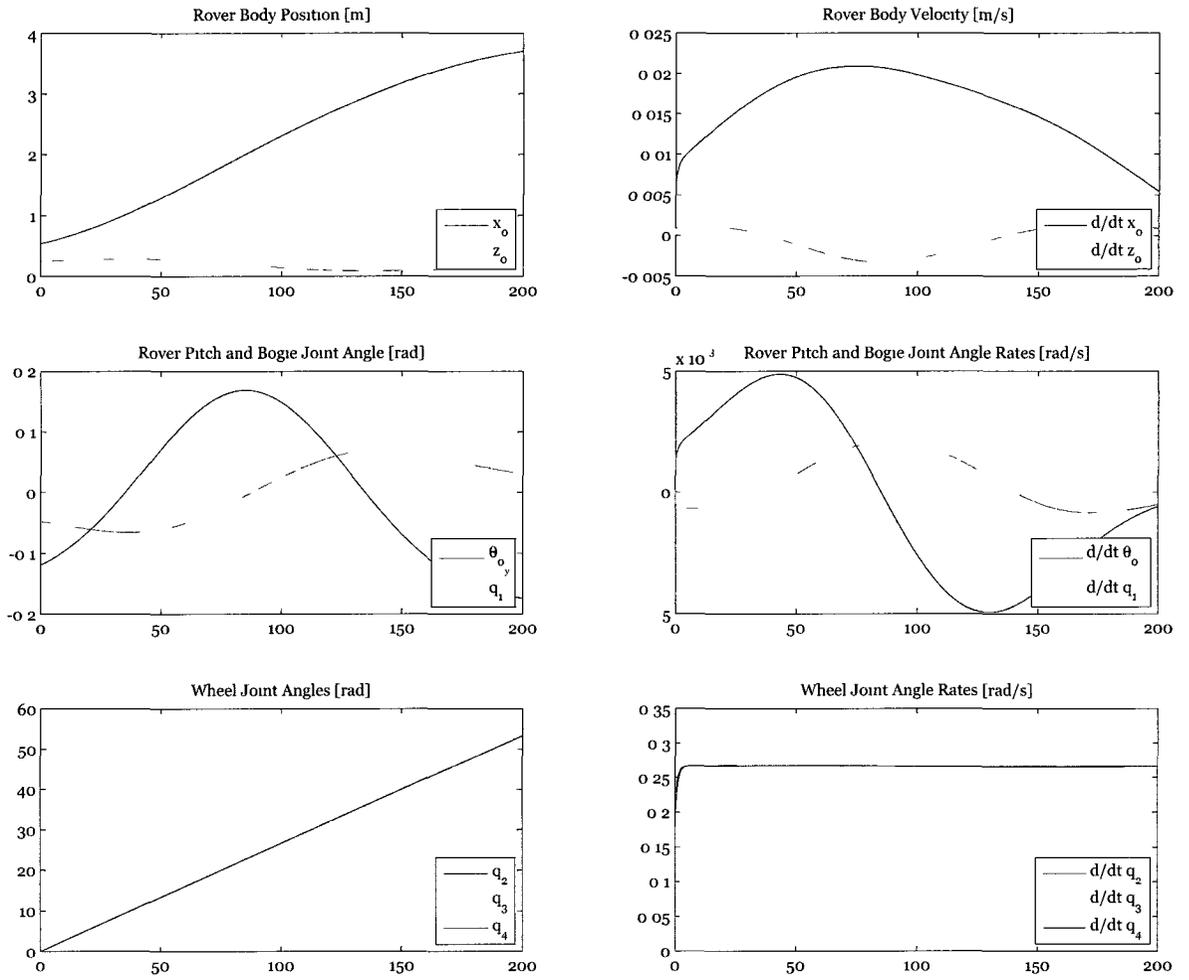


Figure 4.5: Configuration results from the 200 s simulation of the rover traversing gently rolling terrain.

Chapter 5

Net Traction Estimation

This chapter develops an estimator for the two key net traction relationships: resistive torque τ_R and drawbar pull DP . For wheels of fixed dimensions driving on homogeneous terrain with constant soil properties, τ_R and DP will be functions of normal load W and slip i exclusively. The estimator is developed for the two-dimensional rocker-bogie simulation presented in Chapter 4. It is assumed that there is no *a priori* knowledge of soil properties or net traction relationships; the estimator builds a net traction model from scratch. Two separate estimation techniques have been developed, depending on the type of force sensors installed above the wheel hubs: one technique for two-axis force sensors, and one technique for single-axis force sensors. *Kapvik* currently is equipped with single-axis force sensors.

5.1 Optimal State Estimation

A Kalman Filter is a mathematical method used to estimate variables, or “states” in a real system. A model of reality, termed the process model, is used to predict the states; measurements are used to verify this prediction. Since reality does not exactly follow the process model, and since the measurements carry with them inherent noise, neither calculation is perfectly accurate. States are grouped into a vector and treated

as random variables with a mean and co-variance. The application of statistical mathematics allows the prediction and measurement to be combined in a weighted fashion, resulting in an optimal estimate of the desired states. The Kalman Filter also allows states that are not directly measured to be estimated; when it is possible to obtain a stable estimate of a state, the state is said to be “observable”.

The original Kalman filter assumed that the process being estimated was entirely linear. However, most real world systems, including the one studied in this thesis, are non-linear. The Extended Kalman Filter (EKF) addressed this concern by locally linearizing the system about the current state estimate. The EKF has two main downfalls. Firstly, it requires the calculation of large Jacobian matrices. Secondly, it is inaccurate to propagate the probability density function of highly non-linear system through its linear approximation; doing this can introduce large errors which can jeopardize the stability of the estimate.

The Unscented Kalman Filter (UKF), used for estimation in this thesis, works around these downfalls by eliminating the need to linearize the system. Using the UKF, it is not necessary to calculate large Jacobian matrices, and some of the error associated with propagating probability density functions through linear approximations is mitigated. The development of the Unscented Kalman Filter below closely follows that in Dan Simon’s book “Optimal State Estimation” [35].

5.1.1 Unscented Transform

The Unscented Transform is a method for propagating a random variable through a non-linear function. Consider a vector \mathbf{x} containing n_x states with a mean $\bar{\mathbf{x}}$ and covariance matrix \mathbf{P} . A set of $2n_x$ “sigma points” \mathbf{X} can be formed that approximates the random variable \mathbf{x} [35, 36]. Together, these sigma points have an ensemble mean

$\bar{\mathbf{x}}$ and covariance \mathbf{P} equal to that of the state vector \mathbf{x} [35, 36].

$$\begin{aligned} \mathbf{X}_i &= \bar{\mathbf{x}} + \left(\sqrt{n_x \mathbf{P}}\right)_i^T & | & \quad i = \{1, \dots, n_x\} \\ \mathbf{X}_{n_x+i} &= \bar{\mathbf{x}} - \left(\sqrt{n_x \mathbf{P}}\right)_i^T & | & \quad i = \{1, \dots, n_x\} \end{aligned} \quad (5.1)$$

where $\left(\sqrt{n_x \mathbf{P}}\right)_i$ is the i^{th} row of the matrix square root of $n_x \mathbf{P}$.

This set of $2n_x$ vectors, termed sigma points \mathbf{X} , can then be propagated one at a time through a non-linear function f to give another set of sigma points \mathbf{Y} .

$$\mathbf{Y}_i = f(\mathbf{X}_i) \quad | \quad i = \{1, \dots, 2n_x\} \quad (5.2)$$

The mean $\bar{\mathbf{y}}$ and covariance \mathbf{P}_{yy} of the transformed sigma points \mathbf{Y} can then be recovered using weighted averages [35].

$$\bar{\mathbf{y}} = \frac{1}{2n_x} \sum_{i=1}^{2n_x} \mathbf{Y}_i \quad (5.3)$$

$$\mathbf{P}_{yy} = \frac{1}{2n_x} \sum_{i=1}^{2n_x} (\mathbf{Y}_i - \bar{\mathbf{y}})(\mathbf{Y}_i - \bar{\mathbf{y}}) \quad (5.4)$$

To determine the cross-covariance of two sets of sigma points \mathbf{X} and \mathbf{Y} , Equation 5.4 is extended [35].

$$\mathbf{P}_{xy} = \frac{1}{2n_x} \sum_{i=1}^{2n_x} (\mathbf{X}_i - \bar{\mathbf{x}})(\mathbf{Y}_i - \bar{\mathbf{y}}) \quad (5.5)$$

5.1.2 Unscented Kalman Filter

The Unscented Kalman Filter is very similar to the original Kalman Filter, except that it uses the Unscented Transform in order to propagate the mean and covariances. Consider a non-linear system with n_x states that can be modelled as a discrete-time

system.

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, t_k) + \mathbf{w}_k \quad (5.6)$$

$$\mathbf{y}_k = h(\mathbf{x}_k, t_k) + \mathbf{v}_k \quad (5.7)$$

$$\mathbf{w}_k \sim N(0, \mathbf{Q}_k) \quad (5.8)$$

$$\mathbf{v}_k \sim N(0, \mathbf{R}_{\mathbf{y}_k}) \quad (5.9)$$

where \mathbf{x} is the state vector, f is the discrete-time process model, \mathbf{u} is the control vector, t is the time, \mathbf{w} is the normally distributed process noise with covariance \mathbf{Q} , \mathbf{y} is the measurement vector, h is the measurement model, \mathbf{v} is the normally distributed measurement noise with covariance \mathbf{R}_y , and k is the current timestep.

The filter is initialized with the state $\hat{\mathbf{x}}_0^+$ and state covariance estimate \mathbf{P}_0^+ . The “+” superscript indicates an *a posteriori* estimate, and the “-” superscript indicates an *a priori* estimate. Following initialization, the estimator repeats the steps below indefinitely.

The previous *a posteriori* estimate $\hat{\mathbf{x}}_k^+$ is transformed into sigma points $\hat{\mathbf{X}}_k^+$ using Equation 5.1. These sigma points are then passed through the discrete-time process model f . This constitutes the prediction step, and forms the *a priori* state estimate sigma points $\hat{\mathbf{X}}_{k+1}^-$.

$$\begin{aligned} \hat{\mathbf{x}}_{i_k}^+ &= \hat{\mathbf{x}}_k^+ + \left(\sqrt{n_x \mathbf{P}_k^+} \right)_i^T & | & \quad i = \{1, \dots, n_x\} \\ \hat{\mathbf{x}}_{(n_x+i)_k}^+ &= \hat{\mathbf{x}}_k^+ - \left(\sqrt{n_x \mathbf{P}_k^+} \right)_i^T & | & \quad i = \{1, \dots, n_x\} \end{aligned} \quad (5.10)$$

$$\hat{\mathbf{x}}_{i_{k+1}}^- = f\left(\hat{\mathbf{x}}_{i_k}^+, \mathbf{u}_k, t_k\right) \quad | \quad i = \{1, \dots, 2n_x\} \quad (5.11)$$

The *a priori* state covariance \mathbf{P}_{k+1}^- can now be determined using Equation 5.4 and adding the process noise covariance \mathbf{Q}_k .

$$\mathbf{P}_{k+1}^- = \frac{1}{2n_x} \sum_{i=1}^{2n_x} \left(\hat{\mathbf{X}}_{i_{k+1}}^- - \hat{\mathbf{x}}_{k+1}^- \right) \left(\hat{\mathbf{X}}_{i_{k+1}}^- - \hat{\mathbf{x}}_{k+1}^- \right) + \mathbf{Q}_k \quad (5.12)$$

where the mean $\hat{\mathbf{x}}_{k+1}^-$ is recovered using Equation 5.3.

Here, a new set of sigma points can be created using the *a priori* state estimate covariance \mathbf{P}_{k+1}^- , or the previous sigma points can be used to save computational effort at the expense of performance [35]. In this thesis, a new set of sigma points is not created. Estimated measurement vector sigma points $\hat{\mathbf{Y}}$ are calculated by propagating the *a priori* state estimate $\hat{\mathbf{X}}_{i_{k+1}}^-$ through the non-linear measurement model h .

$$\hat{\mathbf{Y}}_{i_{k+1}} = h \left(\hat{\mathbf{X}}_{i_{k+1}}^- \right) \quad | \quad i = \{1, \dots, 2n_x\} \quad (5.13)$$

The Kalman gain \mathbf{K}_{k+1} is given by the cross-covariance between the *a priori* state estimate $\hat{\mathbf{x}}_{k+1}^-$ and the estimated measurement vector $\hat{\mathbf{y}}_{k+1}$ divided by the covariance of $(\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1})$ [35]. The mean of the estimated measurement vector $\hat{\mathbf{y}}_{k+1}$ is recovered using Equation 5.3.

$$\begin{aligned} \mathbf{K}_{k+1} &= \frac{\mathbf{P}_{\hat{\mathbf{x}}_{k+1}^- \hat{\mathbf{y}}_{k+1}}}{\mathbf{P}_{(\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1})(\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1})}} \\ \mathbf{K}_{k+1} &= \frac{\frac{1}{2n_x} \sum_{i=1}^{2n_x} \left(\hat{\mathbf{X}}_{i_{k+1}}^- - \hat{\mathbf{x}}_{k+1}^- \right) \left(\hat{\mathbf{Y}}_{i_{k+1}} - \hat{\mathbf{y}}_{k+1} \right)}{\frac{1}{2n_x} \sum_{i=1}^{2n_x} \left(\hat{\mathbf{Y}}_{i_{k+1}} - \hat{\mathbf{y}}_{k+1} \right) \left(\hat{\mathbf{Y}}_{i_{k+1}} - \hat{\mathbf{y}}_{k+1} \right) + \mathbf{R}_{\mathbf{y}_k}} \end{aligned} \quad (5.14)$$

The Kalman gain \mathbf{K}_{k+1} is used to weight the prediction against the measurement and

form the *a posteriori* state estimate $\hat{\mathbf{x}}_{k+1}^+$ and a *a posteriori* state covariance \mathbf{P}_{k+1}^+ [35].

$$\hat{\mathbf{x}}_{k+1}^+ = \hat{\mathbf{x}}_{k+1}^- + \mathbf{K}_{k+1} (\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}) \quad (5.15)$$

$$\mathbf{P}_{k+1}^+ = \mathbf{P}_{k+1}^- - \mathbf{K}_{k+1} \mathbf{P}_{(\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1})(\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1})} \mathbf{K}_{k+1}^T \quad (5.16)$$

where \mathbf{y}_{k+1} are the actual measurements.

The algorithm is then restarted at Equation 5.10 and run recursively to obtain a state estimate $\hat{\mathbf{x}}$ at every timestep.

5.2 Estimator Outline

Net traction estimation is accomplished in this thesis through the use of two Unscented Kalman Filters running simultaneously. A total of three estimators are presented. Estimator 1, with state vector \mathbf{x}_1 , is a general estimator which estimates a large set of states essential for net traction estimation, including resistive torques $\{\tau_{R_1}, \tau_{R_2}, \tau_{R_3}\}$ and wheel slips $\{\iota_1, \iota_2, \iota_3\}$. It is run regardless of the type of force sensors available. Estimator 2, with state vector \mathbf{x}_2 , estimates wheel-terrain contact angles $\{\gamma_1, \gamma_2, \gamma_3\}$, normal loads $\{W_1, W_2, W_3\}$, and drawbar pulls $\{DP_1, DP_2, DP_3\}$ in the case where two-axis force sensors are available. Estimator 3, with state vector \mathbf{x}_3 , estimates wheel-terrain contact angles $\{\gamma_1, \gamma_2, \gamma_3\}$, normal loads $\{W_1, W_2, W_3\}$, and drawbar pulls $\{DP_1, DP_2, DP_3\}$ in the case where only single-axis force sensors are available.

A set of sensor measurements \mathbf{z} is taken; each is assumed to be a Gaussian random variable. The sensor measurements are used to create the measurement vector \mathbf{y}_1 for use in Estimator 1. Some of the measurements in \mathbf{y}_1 are taken directly from the sensor measurements, and some are calculated using a combination of sensor measurements.

The measurement vectors \mathbf{y}_2 and \mathbf{y}_3 , used in Estimators 2 and 3 respectively, contain sensor measurements as well as measurements calculated using a combination of state estimates from Estimator 1. The motivation behind having two estimators running simultaneously is twofold: some estimated states from Estimator 1 are used in the measurement vectors of Estimators 2 and 3; and in order to separately consider the use of two-axis and single-axis force sensors in this thesis it was instructive to separate the two cases. Figure 5.1 illustrates the flow of information in the net traction estimator.

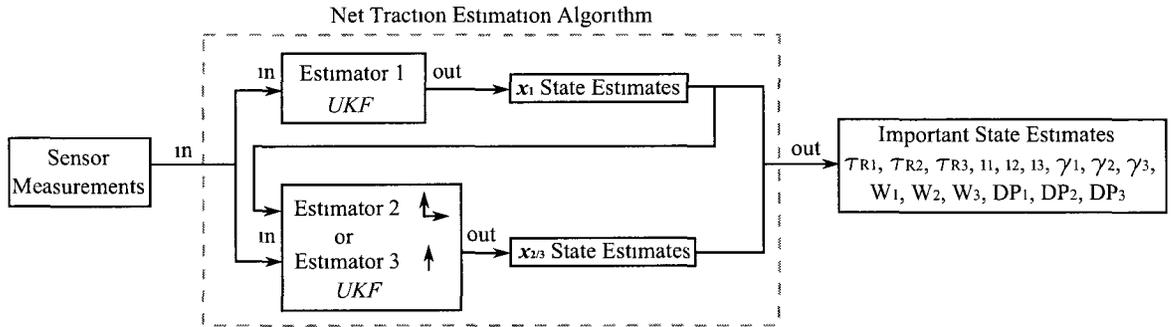


Figure 5.1: Block diagram of information flow through the estimator.

5.3 Sensors

To perform accurate net traction estimation a comprehensive suite of sensors is required. Fortunately, several of these sensors are typically included on planetary rover platforms. The required measurements include wheel angular velocity, bogie angle, rover accelerations, rover pitch rate, forward velocity, and force above the wheel hubs (either two-axis or single-axis). The set of measurements is illustrated in Figure 5.2. In this thesis a measurement of x is denoted \check{x} and the current timestep is denoted k .

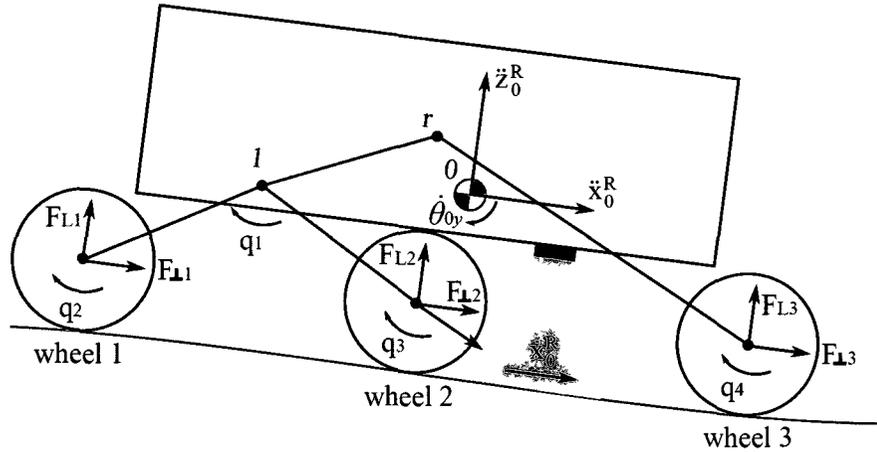


Figure 5.2: Sensor measurements. The wheel angular positions $\{q_2, q_3, q_4\}$ are measured by motor encoders. The bogie joint angle q_1 is measured by a potentiometer or an absolute encoder. The forward velocity of the rover in rover co-ordinates \dot{x}_0^R is measured by a downward facing camera-based velocimeter. The wheel forces $\{F_{L1}, F_{L2}, F_{L3}, F_{\perp 1}, F_{\perp 2}, F_{\perp 3}\}$ are measured using force sensors. Note that the forces $\{F_{\perp 1}, F_{\perp 2}, F_{\perp 3}\}$ are only measured if two-axis force sensors are available. The linear body accelerations in rover co-ordinates $\{\ddot{x}_0^R, \ddot{z}_0^R\}$ are measured by accelerometers in the inertial measurement unit (IMU). The pitch rate $\dot{\theta}_{0y}$ is measured by a rate gyroscope in the IMU.

5.3.1 Motor Encoders

Wheel angle measurements $\{\check{q}_2, \check{q}_3, \check{q}_4\}$ are provided by motor encoders that measure the number of “counts” traversed by the wheel. The encoders used on *Kapvik* were Maxon magneto-resistant encoders with a total of 2000 counts per motor revolution (500 counts per turn with quadrature) shown in Figure 5.3 [28]. With the *Kapvik* gear ratio of 1400:1, the resolution of a single count is $1.286 \times 10^{-4} \text{ }^\circ$. The encoder position noise is approximated by a normal distribution for which the quantization error 3σ bounds are given by: $\pm 3\sigma = \pm \frac{1}{2}$ resolution. This concept is shown in Figure 5.4 and results in a wheel angle standard deviation of $\sigma_{\check{q}_2} = \sigma_{\check{q}_3} = \sigma_{\check{q}_4} = 2.143 \times 10^{-5} \text{ }^\circ$.

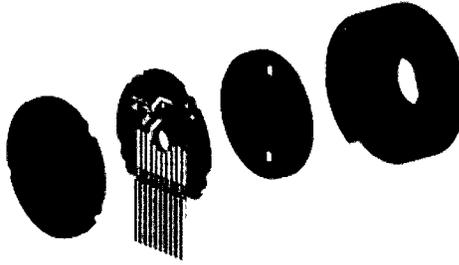


Figure 5.3: Maxon magneto-resistant motor encoder.

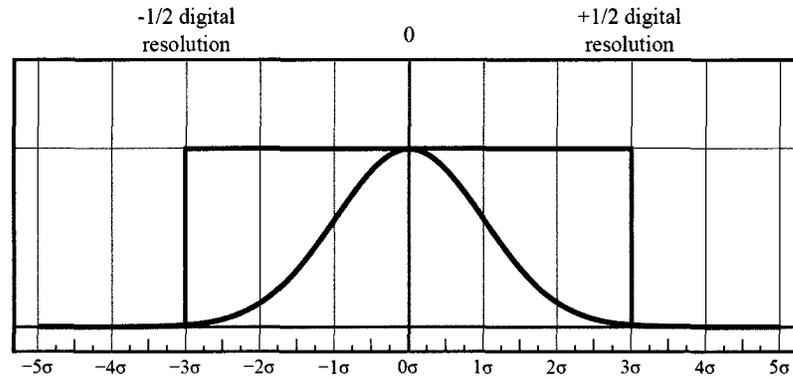
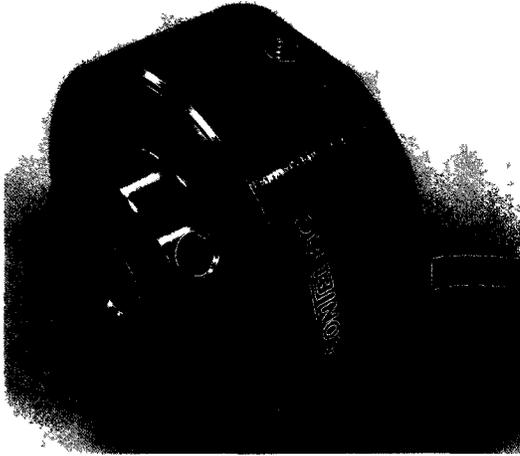


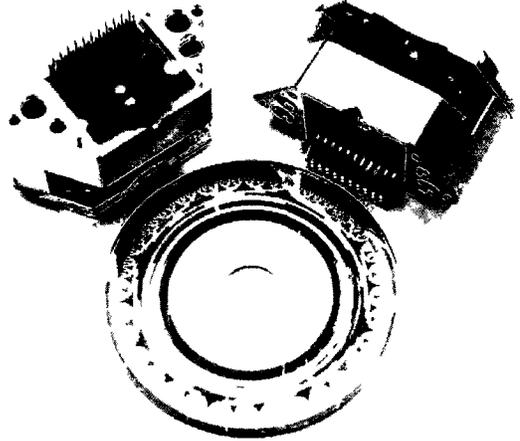
Figure 5.4: Treatment of quantization noise as a normal distribution.

5.3.2 Bogie Angle Measurement

An accurate absolute measurement of bogie angle \check{q}_1 is required for the estimator to succeed in measuring wheel centroid velocities and in estimating slip and wheel-terrain contact angles for wheels 1 and 2. On *Kapvik*, Inscale GL60 hollow-shaft potentiometers, shown in Figure 5.5a, are used to sense this angle. The repeatability of this sensor is 0.1° [37]; although sufficient for normal operation, this was found to be inadequate for net traction estimation. As a result, a more capable sensor was used in simulation: the Avago AEAS-7000 16-bit grey code hollow-shaft absolute encoder, depicted in Figure 5.5b. The dimensions of this absolute encoder [38] are such that they could be incorporated into a later generation of *Kapvik* with minimal modifications. Using the treatment of quantization error shown in Figure 5.4, the standard deviation of the 16-bit absolute encoder noise $\sigma_{\check{q}_1}$ was found to be $9.155 \times 10^{-4}^\circ$.



(a) Inscale GL60 hollow-shaft potentiometer



(b) Avago AEAS-7000 ultra-precision 16-bit grey code hollow-shaft absolute encoder.

Figure 5.5: Bogie angle measurement.

5.3.3 Force Sensors

The single-axis force sensors used on *Kapvik* to measure $\{F_{L_1}, F_{L_2}, F_{L_3}\}$ are Sherborne Sensors S4000M miniature universal load cells (shown in Figure 5.6a). They operate in tension and compression, accommodating a full range load of ± 200 N [31]. The load cell output is sampled by a 10-bit analog to digital converter. The noise is taken to be the quantization error of this signal, giving a noise standard deviation of $\sigma_{\tilde{F}_L}$ of 3.255×10^{-2} N.

A two-axis load cell yields better performance of the net traction estimator, but was not included on *Kapvik*. The ATI Nano25 six-axis force transducer is of the appropriate size and range to be used on *Kapvik*, and could be incorporated into a later generation with some modifications to the chassis. The main sensing axis, which would measure $\{F_{L_1}, F_{L_2}, F_{L_3}\}$ in Figure 5.2 has a sensing range of 500 N, whereas the secondary sensing axis, which would measure $\{F_{\perp 1}, F_{\perp 2}, F_{\perp 3}\}$ in Figure 5.2 has a sensing range of 125 N [39]. The specified resolution in the main sensing axis is 6.25×10^{-2} N and 2.083×10^{-2} N in the secondary sensing axis [39]. Using the quantization error

approach of Figure 5.4 to find the standard deviation of this error yields a result more accurate than the Sherborne Sensors S4000M miniature universal load cell. However, the noise in all force sensors were set to the same value in simulation to allow a proper comparison of the two estimation techniques. Thus $\sigma_{\check{F}_L} = \sigma_{\check{F}_\perp} = 3.255 \times 10^{-2} \text{ N}$ for the two-axis force sensor.

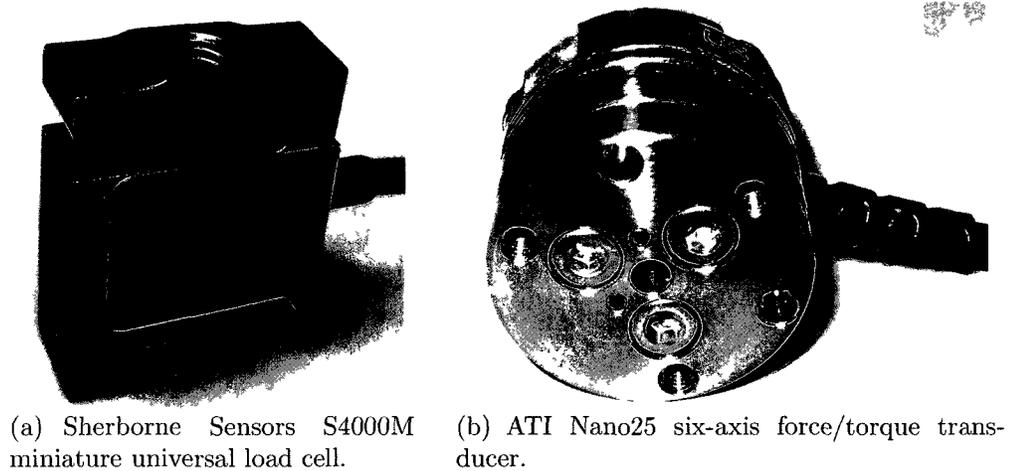


Figure 5.6: Force sensors.

5.3.4 Inertial Measurement Unit

The inertial measurement unit (IMU) used on *Kapvik* is a Memsense H3-IMU HP02-0300, shown in Figure 5.7. It is equipped with three orthogonal accelerometers and three orthogonal rate gyroscopes. The accelerometers have a dynamic range of $\pm 2 g$, and a typical noise standard deviation $\sigma_{\check{x}_0^v} = \sigma_{\check{z}_0^v} = 1.27 \times 10^{-4} g$ [40]. The rate gyroscopes have a typical noise standard deviation $\sigma_{\check{\theta}_{0y}^v}$ of $0.56^\circ/\text{s}$ [40].

5.3.5 Velocimeter

A velocimeter is a device that measures velocity independently of wheel angular velocity or integration of acceleration. It is an invaluable sensor when slip is being

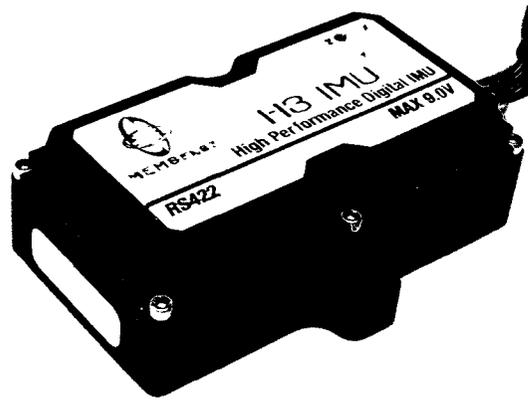


Figure 5.7: Memsense H3-IMU HP02-0300 inertial measurement unit.

estimated. Although currently not implemented on *Kapvik*, a vision-based velocimeter is under development at Carleton University. The velocimeter uses the optic flow from a sequence of images to measure the forward speed of the rover \dot{x}_0^R . A low-cost rangefinder could be used to measure the distance from the camera to the ground and accurately scale the optic flow measurements from camera co-ordinates to world co-ordinates. The velocimeter could be mounted to the bottom of the rover and pointed at the ground. Preliminary tests have been promising, with a conservatively estimated noise of $\sigma_{\dot{x}_0^R} = 1$ mm/s; this noise is used in simulation. An image illustrating how the velocimeter functions is shown in Figure 5.8.



Figure 5.8: Velocimeter principles of operation. (Left) A red box surrounding a central region of the image is considered. (Middle) An algorithm searches within the green search window for a region resembling that within the red box. It finds one and identifies it as the blue box. (Right) The flow vector, displayed as an arrow, is determined using the difference in position between the red and blue boxes. Image courtesy of Adam Mack.

5.3.6 Sensor Measurement Vector and Noise Summary

In reference to the two-dimensional rocker-bogie simulation, a vector of sensor measurements \mathbf{z} can be formed:

$$\mathbf{z} = \left[\check{q}_1 \quad \check{\theta}_{0_y} \quad \check{x}_0^R \quad \check{x}_0^R \quad \check{z}_0^R \quad \check{q}_2 \quad \check{q}_3 \quad \check{q}_4 \quad \check{F}_{L_1} \quad \check{F}_{L_2} \quad \check{F}_{L_3} \quad \check{F}_{\perp_1} \quad \check{F}_{\perp_2} \quad \check{F}_{\perp_3} \right]^T \quad (5.17)$$

where \check{q}_1 is the bogie potentiometer angle, $\check{\theta}_{0_y}$ is the pitch rate measured by the IMU, \check{x}_0^R is the forward speed of the rover measured by the velocimeter, $\{\check{x}_0^R, \check{z}_0^R\}$ are the rover accelerations measured by the IMU along its x and z axes respectively, $\{\check{q}_2, \check{q}_3, \check{q}_4\}$ are the angular positions of the wheels measured by the motor encoders, $\{\check{F}_{L_1}, \check{F}_{L_2}, \check{F}_{L_3}\}$ are the sensed wheel loads along the primary sensing axis, and $\{\check{F}_{\perp_1}, \check{F}_{\perp_2}, \check{F}_{\perp_3}\}$ are the sensed wheel loads along the secondary sensing axis, available only when the rover is equipped with two-axis force sensors.

The noise of each term in the sensor measurement vector \mathbf{z} is summarized in Table 5.1.

Table 5.1: Sensor measurement noises.

Measured Variable (*)	Sensor	Noise Std. Deviation. (σ_*)	Units
Bogie angle \check{q}_1	16-bit Absolute encoder	9.155×10^{-4}	°
Pitch rate $\check{\theta}_{0_y}$	H3-IMU-HP02-0300	0.56	°/s
Velocity in rover coordinates \check{x}_0^R	Velocimeter	1	mm/s
Accelerations in rover coordinates $\{\check{x}_0^R, \check{z}_0^R\}$	H3-IMU-HP02-0300	1.245×10^{-3}	m/s ²
Wheel positions $\{\check{q}_2, \check{q}_3, \check{q}_4\}$	Motor encoders	2.143×10^{-5}	°
Wheel vertical loads $\{\check{F}_{L_1}, \check{F}_{L_2}, \check{F}_{L_3}\}$	Force sensor	3.255×10^{-2}	N
Wheel perpendicular loads $\{\check{F}_{\perp_1}, \check{F}_{\perp_2}, \check{F}_{\perp_3}\}$	Force sensor	3.255×10^{-2}	N

5.4 Estimator 1: The General Estimator

The general estimator estimates 22 states related to net traction estimation. In this thesis an estimate of x is denoted \hat{x} . Observability of the general estimator is proved in Appendix D.1.

5.4.1 States and Process Model

The states estimated by the general estimator are:

$$\mathbf{x}_1 = \begin{bmatrix} \dot{q}_2 & \dot{q}_3 & \dot{q}_4 & \tau_{R_1} & \tau_{R_2} & \tau_{R_3} & \theta_{0_y} & q_1 & \dot{\theta}_{0_y} & \dot{q}_1 & \ddot{\theta}_{0_y} & \ddot{q}_1 & \dots \\ \dots & \ddot{x}_0 & \ddot{z}_0 & M_{0f_w} & M_{1f_w} & F_{xf_w} & F_{zf_w} & \dot{x}_0^R & i_1 & i_2 & i_3 \end{bmatrix}^T \quad (5.18)$$

where M_{0f_w} is the net moment exerted about the lumped body/rocker center of mass 0 by wheel forces, M_{1f_w} is the net moment exerted about the bogie joint 1 by wheel forces, F_{xf_w} is the net force in the world horizontal direction X_w exerted by wheel forces, F_{zf_w} is the net force in the world vertical direction Z_w exerted by wheel forces, and $\{i_1, i_2, i_3\}$ are the wheel slip(s)/skid(s). Note that $\{M_{0f_w}, M_{1f_w}, F_{xf_w}, F_{zf_w}\}$ describe the net effects of wheel *linear* forces, but do not include the net effects of wheel resistive torques.

An estimated configuration $\hat{\Phi}$ can be calculated for subsequent use. The estimated configuration $\hat{\Phi}$ will be used below in the calculation of variables \hat{H} , $\frac{\partial V}{\partial \Phi}$, \hat{J}_w , and \hat{J}_a . Since the wheel mass centers lie directly at the wheel joints (i.e. $\mathbf{c}_{22} = \mathbf{c}_{33} = \mathbf{c}_{44} = \mathbf{0}$), none of the above variables to be calculated are dependent on wheel angles $\{q_2, q_3, q_4\}$. The position of the rover in the world frame, given by $\{x_0, y_0, z_0\}$, is also irrelevant. Thus it is sufficient to include just $\hat{\theta}_{0_{y_k}}$ and \hat{q}_{1_k} in the estimated configuration $\hat{\Phi}$:

$$\hat{\Phi}_k = \begin{bmatrix} 0 & 0 & 0 & 0 & \hat{\theta}_{0_{y_k}} & 0 & \hat{q}_{1_k} & 0 & 0 & 0 \end{bmatrix}^T \quad (5.19)$$

Similarly, the estimated generalized velocity vector $\hat{\Phi}$ is given by:

$$\hat{\Phi}_k = \left[\hat{x}_{0_k}^R \cos \hat{\theta}_{0_{y_k}} \quad 0 \quad -\hat{x}_{0_k}^R \sin \hat{\theta}_{0_{y_k}} \quad 0 \quad \hat{\theta}_{0_{y_k}} \quad 0 \quad \hat{q}_{1_k} \quad \hat{q}_{2_k} \quad \hat{q}_{3_k} \quad \hat{q}_{4_k} \right]^T \quad (5.20)$$

Note that the estimate $\hat{\Phi}$ involves the assumption that the velocity in the rover's z direction \dot{z}_0^R is zero. This approximation is reasonable when the rover is traversing gently rolling terrain. It is unreasonable in rough terrain such as in rock fields; however, in this sort of terrain the soil is unlikely to be homogeneous and thus net traction estimation should not be performed.

The process model is a non-linear discrete-time model. The state is propagated forward using the function f_1 :

$$\hat{\mathbf{x}}_{1_{k+1}} = f_1(\hat{\mathbf{x}}_{1_k}, \mathbf{u}_k) \quad (5.21)$$

where \mathbf{u} is the control vector, equal to the set of input wheel torques:

$$\mathbf{u} = [\tau_{W_1} \quad \tau_{W_2} \quad \tau_{W_3}]^T \quad (5.22)$$

The first three states are wheel angular velocities $\{\dot{q}_2, \dot{q}_3, \dot{q}_4\}$. Their process model is straightforward, propagating the previous velocity forward by the estimator timestep T_e using the estimated angular acceleration.

$$\hat{q}_{i_{k+1}} = \hat{q}_{i_k} + \hat{q}_{i_k} T_e \quad | \quad i = \{2, 3, 4\} \quad (5.23)$$

$$\hat{q}_{i_{k+1}} = \left(\tau_{W_{i-1_k}} - \hat{\tau}_{R_{i-1_k}} - \hat{q}_{i_k} b_d \right) \frac{1}{I_{wg}} \quad | \quad i = \{2, 3, 4\} \quad (5.24)$$

where \hat{q}_i is the estimated angular acceleration; although not a state, it is calculated using other states and control inputs and is used multiple times in the process model.

All accelerations calculated in the estimator use a particular row of Equation 4.48, solved for the acceleration of interest. When using the dynamic equations in the estimator, the terms $\dot{\mathbf{H}}\dot{\Phi}$ and $\frac{\partial T}{\partial \Phi}$ are not included. In Equation 5.24, rows 8-10 of Equation 4.48 are solved for $\{\ddot{q}_2, \ddot{q}_3, \ddot{q}_4\}$; negligible inertial terms in \mathbf{H} relating $\{\ddot{q}_2, \ddot{q}_3, \ddot{q}_4\}$ to $\ddot{\theta}_{0_y}$ and \ddot{q}_1 are neglected. The wheel angular accelerations are dependent on the applied wheel torques $\{\tau_{W_{1_k}}, \tau_{W_{2_k}}, \tau_{W_{3_k}}\}$, the estimated resistive torques $\{\hat{\tau}_{R_{1_k}}, \hat{\tau}_{R_{2_k}}, \hat{\tau}_{R_{3_k}}\}$, the estimated wheel angular velocities $\{\hat{q}_{2_k}, \hat{q}_{3_k}, \hat{q}_{4_k}\}$, the viscous angular damping coefficient b_d , and the inertia of the gear train and the wheel referred to the wheel output I_{wg} .

The next three estimated states are the resistive torques exerted on the wheels $\{\tau_{R_{1_k}}, \tau_{R_{2_k}}, \tau_{R_{3_k}}\}$. As the estimator has no *a priori* knowledge of the model for resistive torque τ_R , a random walk is assumed. Thus the predicted value for the next resistive torque is simply the previously estimated resistive torque:

$$\hat{\tau}_{R_{i_{k+1}}} = \hat{\tau}_{R_{i_k}} \quad | \quad i = \{1, 2, 3\} \quad (5.25)$$

The resistive torques $\{\tau_{R_1}, \tau_{R_2}, \tau_{R_3}\}$, wheel angular accelerations $\{\ddot{q}_2, \ddot{q}_3, \ddot{q}_4\}$, and wheel angular velocities $\{\dot{q}_2, \dot{q}_3, \dot{q}_4\}$ are all observable through measurement of the wheel angular velocities [11]. The wheel angular velocity measurements are obtained using the finite difference method; this is shown in the next section.

The process models for θ_{0_y} and q_1 use second order Taylor series approximations; the

process models for $\hat{\theta}_{0y}$, and \hat{q}_1 use first order Taylor series approximations.

$$\hat{\theta}_{0y_{k+1}} = \hat{\theta}_{0y_k} + \hat{\theta}_{0y_k} T_e + \hat{\theta}_{0y_k} \frac{T_e^2}{2} \quad (5.26)$$

$$\hat{q}_{1_{k+1}} = \hat{q}_{1_k} + \hat{q}_{1_k} T_e + \hat{q}_{1_k} \frac{T_e^2}{2} \quad (5.27)$$

$$\hat{\theta}_{0y_{k+1}} = \hat{\theta}_{0y_k} + \hat{\theta}_{0y_k} T_e \quad (5.28)$$

$$\hat{q}_{1_{k+1}} = \hat{q}_{1_k} + \hat{q}_{1_k} T_e \quad (5.29)$$

The process models for $\ddot{\theta}_{0y}$ and \ddot{q}_1 use rows 5 and 7 from Equation 4.48, solved for the terms $\ddot{\theta}_{0y}$ and \ddot{q}_1 respectively.

$$\begin{aligned} \hat{\ddot{\theta}}_{0y_{k+1}} = & \left(\hat{M}_{0fw_k} - \hat{\mathbf{J}}_{w5,5k} \hat{r}_{R1_k} - \hat{\mathbf{J}}_{w11,5k} \hat{r}_{R2_k} - \hat{\mathbf{J}}_{w17,5k} \hat{r}_{R3_k} - \left[\frac{\partial \hat{V}}{\partial \hat{\Phi}} \right]_5 - \hat{\mathbf{H}}_{5,1k} \hat{x}_{0k} \dots \right. \\ & \left. \dots - \hat{\mathbf{H}}_{5,3k} \hat{z}_{0k} - \hat{\mathbf{H}}_{5,7k} \hat{q}_{1k} - \hat{\mathbf{H}}_{5,8k} \hat{q}_{2k} - \hat{\mathbf{H}}_{5,9k} \hat{q}_{3k} - \hat{\mathbf{H}}_{5,10k} \hat{q}_{4k} \right) \frac{1}{\hat{\mathbf{H}}_{5,5k}} \end{aligned} \quad (5.30)$$

$$\begin{aligned} \hat{\ddot{q}}_{1_{k+1}} = & \left(\hat{M}_{1fw_k} - \hat{\mathbf{J}}_{w5,7k} \hat{r}_{R1_k} - \hat{\mathbf{J}}_{w11,7k} \hat{r}_{R2_k} - \left[\frac{\partial \hat{V}}{\partial \hat{\Phi}} \right]_7 - \hat{\mathbf{H}}_{7,1k} \hat{x}_{0k} \dots \right. \\ & \left. \dots - \hat{\mathbf{H}}_{7,3k} \hat{z}_{0k} - \hat{\mathbf{H}}_{7,5k} \hat{\theta}_{0y_k} - \hat{\mathbf{H}}_{7,8k} \hat{q}_{2k} - \hat{\mathbf{H}}_{7,9k} \hat{q}_{3k} \right) \frac{1}{\hat{\mathbf{H}}_{7,7k}} \end{aligned} \quad (5.31)$$

where $\frac{\partial \hat{V}}{\partial \hat{\Phi}}$, $\hat{\mathbf{H}}$, and $\hat{\mathbf{J}}_w$ are obtained using the estimated configuration $\hat{\Phi}$, and the terms $\dot{\mathbf{H}}\dot{\Phi}$ and $\frac{\partial T}{\partial \hat{\Phi}}$ are not included.

The process models for \ddot{x}_0 and \ddot{z}_0 use rows 1 and 3 from Equation 4.48 solved for the terms \ddot{x}_0 and \ddot{z}_0 respectively. All terms except for $\dot{\mathbf{H}}\dot{\Phi}$ and $\frac{\partial T}{\partial \hat{\Phi}}$ are included.

$$\hat{\ddot{x}}_{0_{k+1}} = \left(\hat{F}_{xw_k} - \hat{\mathbf{H}}_{1,5k} \hat{\theta}_{0y_k} - \hat{\mathbf{H}}_{1,7k} \hat{q}_{1k} \right) \frac{1}{\hat{\mathbf{H}}_{1,1k}} \quad (5.32)$$

$$\hat{\ddot{z}}_{0_{k+1}} = \left(\hat{F}_{zw_k} - \left[\frac{\partial \hat{V}}{\partial \hat{\Phi}} \right]_3 - \hat{\mathbf{H}}_{3,5k} \hat{\theta}_{0y_k} - \hat{\mathbf{H}}_{3,7k} \hat{q}_{1k} \right) \frac{1}{\hat{\mathbf{H}}_{3,3k}} \quad (5.33)$$

The net moments $\{M_{0fw}, M_{1fw}\}$ and forces $\{F_{xfw}, F_{zfw}\}$ resultant from the wheel-terrain interaction forces $\{DP_1, DP_2, DP_3, W_1, W_2, W_3\}$ are assumed to follow a random walk. Their process models are then:

$$\hat{M}_{0fw_{k+1}} = \hat{M}_{0fw_k} \quad \hat{M}_{1fw_{k+1}} = \hat{M}_{1fw_k} \quad \hat{F}_{xfw_{k+1}} = \hat{F}_{xfw_k} \quad \hat{F}_{zfw_{k+1}} = \hat{F}_{zfw_k} \quad (5.34)$$

The process model for the velocity along the rover \hat{x}_0^R is a first order Taylor series approximation using the estimated acceleration of the rover along the rover body.

$$\hat{x}_{0k+1}^R = \hat{x}_{0k}^R + \left(\hat{x}_{0k} \cos \hat{\theta}_{0y_k} - \hat{z}_{0k} \sin \hat{\theta}_{0y_k} \right) T_e \quad (5.35)$$

The slip(s)/skid(s) of the rover wheels are assumed to follow a random walk.

$$\hat{i}_{v_{k+1}} = \hat{i}_{v_k} \quad | \quad i = \{1, 2, 3\} \quad (5.36)$$

5.4.2 Measurements

The measurement vector \mathbf{y}_1 for the general estimator is given by:

$$\mathbf{y}_1 = \left[\check{q}_2 \quad \check{q}_3 \quad \check{q}_4 \quad \check{\theta}_{0y} \quad \check{q}_1 \quad \check{x}_0^R \quad \check{z}_0^R \quad \check{x}_0^R \quad \check{i}_1 \quad \check{i}_2 \quad \check{i}_3 \right]^T \quad (5.37)$$

The first three measurements are wheel angular velocities. These are obtained by applying the first order finite difference method to the wheel angle measurements.

$$\check{q}_{i_k} = \frac{\check{q}_{i_k} - \check{q}_{i_{k-1}}}{T_e} \quad | \quad i = \{2, 3, 4\} \quad (5.38)$$

A measurement of the rover's pitch angle $\check{\theta}_{0y}$ is required in order to make the system observable. To accomplish this, a measurement of rover pitch is constructed using the direction of the gravity vector as sensed by the IMU; this will be accurate under slow

acceleration and steady-state conditions. Since net traction estimation is nominally performed when the rover is in steady-state, and not when it is undergoing large accelerations, it is typically accurate when required. The measurement of the pitch angle is given by:

$$\check{\theta}_{0_{y_k}} = \tan^{-1} \left(\frac{-\check{\ddot{x}}_{0_k}^R}{\check{\ddot{z}}_{0_k}^R} \right) \quad (5.39)$$

The measurements \check{q}_1 , $\check{\ddot{x}}_0^R$, $\check{\ddot{z}}_0^R$, and $\check{\ddot{x}}_0^R$ are taken directly from the sensor measurement vector \mathbf{z} .

To provide a measurement of wheel slips $\{\check{i}_1, \check{i}_2, \check{i}_3\}$, the wheel centroid velocities must first be found. A measured configuration vector $\check{\Phi}$ and its time derivative $\check{\dot{\Phi}}$ are defined:

$$\check{\Phi}_k = \left[0 \ 0 \ 0 \ 0 \ \check{\theta}_{0_{y_k}} \ 0 \ \check{q}_{1_k} \ 0 \ 0 \ 0 \right]^T \quad (5.40)$$

$$\check{\dot{\Phi}}_k = \left[\check{\ddot{x}}_{0_k}^R \cos \check{\theta}_{0_{y_k}} \ 0 \ -\check{\ddot{x}}_{0_k}^R \sin \check{\theta}_{0_{y_k}} \ \check{\dot{\theta}}_{0_{y_k}} \ 0 \ \check{q}_{1_k} \ \check{q}_{2_k} \ \check{q}_{3_k} \ \check{q}_{4_k} \right]^T \quad (5.41)$$

Note that the measured pitch rate $\check{\dot{\theta}}_{0_{y_k}}$ is utilized in the measured generalized velocity vector $\check{\dot{\Phi}}$. The measured link Jacobian \check{J}_{a_k} and measured set of link center of mass velocities \check{X}_{a_k} can then be calculated as in Equation 4.28.

$$\check{X}_{a_k} = \check{J}_{a_k} \check{\dot{\Phi}}_k \quad (5.42)$$

Since the distances from joints 2, 3, and 4 to their corresponding link centers of mass are zero ($\mathbf{c}_{22} = \mathbf{c}_{33} = \mathbf{c}_{44} = 0$) the wheel centroid velocities are given by $\{\mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4\}$ in Equations 4.28 and 5.42. The measurements of slip can be calculated using the wheel linear and angular velocity magnitudes found above, and Equation 2.23. It is

assumed that the rover is moving in the direction of positive X_w

$$\check{v}_{i_k} = i \left(\text{sgn} \left(\check{\mathbf{X}}_{\mathbf{a}(6i+1)_k} \right) |\check{v}_{i+1_k}|, \check{q}_{i+1_k} \right) \quad | \quad i = \{1, 2, 3\}$$

where \check{v}_{i+1_k} is the wheel centroid velocity vector of wheel i , a term in $\check{\mathbf{X}}_{\mathbf{a}}$ (see Equations 4 28 and 5 42), note that the angular velocities $\{\theta_{0_y}, q_1\}$ are treated as negligible in comparison with the wheel joint velocities $\{q_2, q_3, q_4\}$ when inputting the wheel angular velocities to this equation

5.4.3 Measurement Model

The measurement model h_1 predicts the expected set of measurements $\hat{\mathbf{y}}_1$ based on the estimated state vector $\hat{\mathbf{x}}_1$. In this thesis, the expected measurement of a variable x is denoted \hat{x}

$$\hat{\mathbf{y}}_{1k} = h_1(\hat{\mathbf{x}}_{1k}) \quad (5 43)$$

The wheel angular velocities $\{q_2, q_3, q_4\}$, rover pitch θ_{0_y} and bogie joint angle q_1 are states in $\hat{\mathbf{x}}_1$ and thus have a simple measurement model

$$\hat{q}_{2k} = \hat{q}_{2k} \quad \hat{q}_{3k} = \hat{q}_{3k} \quad \hat{q}_{4k} = \hat{q}_{4k} \quad \hat{\theta}_{0_{y_k}} = \hat{\theta}_{0_{y_k}} \quad \hat{q}_{1k} = \hat{q}_{1k} \quad (5 44)$$

The estimated accelerations in world co-ordinates $\{\hat{x}_0, \hat{z}_0\}$ must be rotated into rover co-ordinates to match the measurements $\{\check{x}_0^R, \check{z}_0^R\}$. The effect of gravity, which is measured by the accelerometers, must also be included. Note that gravity, although directed downwards, produces the same signal as an upward acceleration g

$$\hat{x}_{0_k}^R = \hat{x}_{0_k} \cos \hat{\theta}_{0_{y_k}} - \hat{z}_{0_k} \sin \hat{\theta}_{0_{y_k}} - g \sin \hat{\theta}_{0_{y_k}} \quad (5 45)$$

$$\hat{z}_{0_k}^R = \hat{x}_{0_k} \sin \hat{\theta}_{0_{y_k}} + \hat{z}_{0_k} \cos \hat{\theta}_{0_{y_k}} + g \cos \hat{\theta}_{0_{y_k}} \quad (5 46)$$

The slips $\{i_1, i_2, i_3\}$ are states in $\hat{\mathbf{x}}_1$ and thus have a simple measurement model.

$$\hat{i}_{1k} = \hat{i}_{1k} \quad \hat{i}_{2k} = \hat{i}_{2k} \quad \hat{i}_{3k} = \hat{i}_{3k} \quad (5.47)$$

5.5 Estimator 2: Wheel Force Estimator with Two-Axis Force Sensors

Estimator 2 estimates the wheel-terrain contact angles and the wheel forces when two-axis force sensors are available above each wheel. The observability of Estimators 2 and 3 is proved in Appendix D.2.

5.5.1 States and Process Model

The two wheel force estimators, Estimator 2 and Estimator 3, both estimate nine states: the wheel-terrain contact angles $\{\gamma_1, \gamma_2, \gamma_3\}$, the wheel normal loads $\{W_1, W_2, W_3\}$, and the wheel drawbar pulls $\{DP_1, DP_2, DP_3\}$.

$$\mathbf{x}_2 = \mathbf{x}_3 = \left[\gamma_1 \ \gamma_2 \ \gamma_3 \ W_1 \ W_2 \ W_3 \ DP_1 \ DP_2 \ DP_3 \right]^T \quad (5.48)$$

The process model is a non-linear discrete-time model. The state is propagated forward using the function f_2 :

$$\hat{\mathbf{x}}_{2k+1} = f_2(\hat{\mathbf{x}}_{2k}) \quad (5.49)$$

The technique used to estimate wheel-terrain contact angles $\{\gamma_1, \gamma_2, \gamma_3\}$ is identical for both Estimators 2 and 3. An accurate estimate of the wheel-terrain contact angles $\{\gamma_1, \gamma_2, \gamma_3\}$ is essential to properly apportion the sensed forces into normal load W and drawbar pull DP . The wheel-terrain contact angles are also important when

optimally distributing tractive forces in traction control [5, 16]. Evolution of the terrain is assumed to be a random process since it is inherently unpredictable [5]. Without a detailed model of the terrain ahead, reasonable predictions of the wheel-terrain contact angle at the next timestep are not feasible. Thus, the process model for the wheel-terrain contact angles is assumed to be a random walk:

$$\hat{\gamma}_{i_{k+1}} = \hat{\gamma}_{i_k} \quad | \quad i = \{1, 2, 3\} \quad (5.50)$$

Since the estimator has no *a priori* knowledge of the net traction relationships, both normal load W and drawbar pull DP are unmodelled phenomena; their process model is also assumed to be a random walk:

$$\hat{W}_{i_{k+1}} = \hat{W}_{i_k} \quad | \quad i = \{1, 2, 3\} \quad (5.51)$$

$$\hat{DP}_{i_{k+1}} = \hat{DP}_{i_k} \quad | \quad i = \{1, 2, 3\} \quad (5.52)$$

5.5.2 Measurements

The measurement vector \mathbf{y}_2 contains wheel-terrain contact angle measurements and force sensor measurements.

$$\mathbf{y}_2 = \left[\check{\gamma}_1 \quad \check{\gamma}_2 \quad \check{\gamma}_3 \quad \check{F}_{L_1} \quad \check{F}_{L_2} \quad \check{F}_{L_3} \quad \check{F}_{\perp_1} \quad \check{F}_{\perp_2} \quad \check{F}_{\perp_3} \right]^T \quad (5.53)$$

The measurements of the wheel-terrain contact angles $\{\check{\gamma}_1, \check{\gamma}_2, \check{\gamma}_3\}$ are obtained from the kinematics of the rover. The velocity of the wheel centroid will be parallel to the slope of the terrain at the wheel-terrain contact point. Thus if a wheel centroid has velocity vector \mathbf{v}_w , the wheel-terrain contact angle is given by:

$$\gamma = \tan^{-1} \left(\frac{-v_{w_z}}{v_{w_x}} \right) \quad (5.54)$$

The estimated configuration $\hat{\Phi}$ and estimated generalized velocity $\hat{\dot{\Phi}}$, calculated in Estimator 1, are used in the measurement of wheel-terrain contact angles. Note that wheel-terrain contact angle estimation is very sensitive to the validity of the assumption that $\dot{z}_0^R \approx 0$, because an accurate estimate of the joints' velocity in Z_w is required. The estimated joint velocities $\hat{\dot{X}}_{\alpha k}$ can be found using the estimated joint velocity Jacobian $\hat{J}_{\alpha k}$ and the estimated generalized velocity $\hat{\dot{\Phi}}_k$:

$$\hat{\dot{X}}_{\alpha k} = \hat{J}_{\alpha k} \hat{\dot{\Phi}}_k \quad (5.55)$$

This is identical to the manner in which measured joint velocities $\check{\dot{X}}_{\alpha}$ were determined for Estimator 1, but is more accurate since estimated values are used in place of measured ones. The additional accuracy is a benefit of separating the estimators. A measurement of the wheel-terrain contact angle for each wheel can then be determined:

$$\check{\gamma}_{i_k} = \tan^{-1} \left(\frac{-\hat{v}_{i+1z_k}}{\hat{v}_{i+1x_k}} \right) = \tan^{-1} \left(\frac{-\hat{\dot{X}}_{\alpha(6i+3)_k}}{\hat{\dot{X}}_{\alpha(6i+1)_k}} \right) \quad | \quad i = \{1, 2, 3\} \quad (5.56)$$

where $\hat{J}_{\alpha k}$ is a function of $\hat{\Phi}_k$, and \hat{v}_{i+1x_k} and \hat{v}_{i+1z_k} are respectively the X_w and Z_w components of wheel i 's centroid velocity.

The wheel-terrain contact angle estimation presented above is similar to the method used by Iagnemma et al. [5, 16]. Iagnemma et al. use the rover pitch, rover pitch rate, and no-slip wheel velocities to kinematically determine the wheel-terrain contact angles of a simulated two wheeled rover [5, 16]. Unlike Iagnemma et al.'s method, and owing to the addition of a velocimeter, the method presented in this thesis does not rely on the assumption of low slip. The measurements $\{\check{F}_{L_1}, \check{F}_{L_2}, \check{F}_{L_3}, \check{F}_{\perp_1}, \check{F}_{\perp_2}, \check{F}_{\perp_3}\}$ are taken directly from the force sensors.

5.5.3 Measurement Model

The measurement model h_2 predicts the expected set of measurements $\hat{\mathbf{y}}_2$ based on the state vector \mathbf{x}_2 .

The wheel-terrain contact angles are states in \mathbf{x}_2 and thus have a simple measurement model.

$$\hat{\gamma}_{i_k} = \hat{\gamma}_{i_k} \quad | \quad i = \{1, 2, 3\} \quad (5.57)$$

In order to predict the force sensor measurements $\{F_{L_1}, F_{L_2}, F_{L_3}, F_{\perp_1}, F_{\perp_2}, F_{\perp_3}\}$, it is necessary to consider the angle between the estimated forces and the measured forces. The drawbar pulls $\{DP_1, DP_2, DP_3\}$ will act parallel to the ground, at angles of $\{\gamma_1, \gamma_2, \gamma_3\}$ with respect to the world horizontal X_w . The normal loads $\{W_1, W_2, W_3\}$ will act perpendicular to the ground, at angles of $\{\gamma_1, \gamma_2, \gamma_3\}$ from the world vertical Z_w . Forces along the primary sensing axes $\{F_{L_1}, F_{L_2}, F_{L_3}\}$ will be sensed at angles presently defined as $\{\epsilon_1, \epsilon_2, \epsilon_3\}$ from the world vertical Z_w ; the angles $\{\epsilon_1, \epsilon_2, \epsilon_3\}$ are shown in Figure 5.9. Forces along the secondary sensing axes $\{F_{\perp_1}, F_{\perp_2}, F_{\perp_3}\}$ will be sensed at angles $\{\epsilon_1, \epsilon_2, \epsilon_3\}$ from the world horizontal X_w . Thus, the expected force measurements are given by:

$$\hat{F}_{L_{i_k}} = \hat{W}_{i_k} \cos(\hat{\gamma}_{i_k} - \hat{\epsilon}_{i_k}) - \hat{D}P_{i_k} \sin(\hat{\gamma}_{i_k} - \hat{\epsilon}_{i_k}) \quad | \quad i = \{1, 2, 3\} \quad (5.58)$$

$$\hat{F}_{\perp_{i_k}} = \hat{W}_{i_k} \sin(\hat{\gamma}_{i_k} - \hat{\epsilon}_{i_k}) + \hat{D}P_{i_k} \cos(\hat{\gamma}_{i_k} - \hat{\epsilon}_{i_k}) \quad | \quad i = \{1, 2, 3\} \quad (5.59)$$

where $\hat{\epsilon}_{1_k} = \hat{\epsilon}_{2_k} = \hat{\theta}_{0_{y_k}} + \hat{q}_{1_k}$ and $\hat{\epsilon}_{3_k} = \hat{\theta}_{0_{y_k}}$.

5.6 Estimator 3: Wheel Force Estimator with Single-Axis Force Sensors

Estimator 3 estimates the wheel-terrain contact angles and the wheel forces when only single-axis force sensors are available above each wheel, as is the case on the *Kapvik* micro-rover.

5.6.1 States and Process Model

The state vector for Estimators 2 and 3 are identical.

$$\mathbf{x}_3 = \mathbf{x}_2 = \left[\gamma_1 \ \gamma_2 \ \gamma_3 \ W_1 \ W_2 \ W_3 \ DP_1 \ DP_2 \ DP_3 \right]^T \quad (5.60)$$

The process models are also the same: all states are assumed to undergo a random walk.

$$\hat{\mathbf{x}}_{3k+1} = \hat{\mathbf{x}}_{3k} \quad (5.61)$$

5.6.2 Measurements

The measurement vector contains a set of measurements identical to Estimator 2; however, the perpendicular forces $\{F_{\perp 1}, F_{\perp 2}, F_{\perp 3}\}$ are calculated artificially since they are not explicitly sensed.

$$\mathbf{y}_3 = \mathbf{y}_2 = \left[\check{\gamma}_1 \ \check{\gamma}_2 \ \check{\gamma}_3 \ \check{F}_{L_1} \ \check{F}_{L_2} \ \check{F}_{L_3} \ \check{F}_{\perp 1} \ \check{F}_{\perp 2} \ \check{F}_{\perp 3} \right]^T \quad (5.62)$$

The procedure for calculating measurements of wheel-terrain contact angles $\{\check{\gamma}_1, \check{\gamma}_2, \check{\gamma}_3\}$ is the same as in Estimator 2 (see Equation 5.56). The measurements $\{\check{F}_{L_1}, \check{F}_{L_2}, \check{F}_{L_3}\}$ are taken directly from the force sensors. The wheel forces in the secondary sensing axes $\{\check{F}_{\perp 1}, \check{F}_{\perp 2}, \check{F}_{\perp 3}\}$ must be solved for using the estimated net

effects of wheel forces $\{\hat{M}_{0fw}, \hat{M}_{1fw}, \hat{F}_{xfw}, \hat{F}_{zfw}\}$ from Estimator 1. As shown in Figure 5.9, the forces $F_{\perp 1}$ and $F_{\perp 2}$ share the same line of action. It is impossible to uniquely determine these two values with the observed net effects of wheel forces $\{\hat{M}_{0fw}, \hat{M}_{1fw}, \hat{F}_{xfw}, \hat{F}_{zfw}\}$, so they are combined into a single variable $F_{\perp 12}$.

$$F_{\perp 12} = F_{\perp 1} + F_{\perp 2} \quad (5.63)$$

With only two unknowns, $F_{\perp 12}$ and $F_{\perp 3}$, only two of the four net effects of wheel forces $\{\hat{M}_{0fw}, \hat{M}_{1fw}, \hat{F}_{xfw}, \hat{F}_{zfw}\}$ need to be considered to find a solution. The equation for M_{1fw} is used to solve for $F_{\perp 12}$. With knowledge of $F_{\perp 12}$, the equation for F_{xfw} is used to solve for $F_{\perp 3}$. Referring to Figure 5.9, expressions for the forces $F_{\perp 12}$ and $F_{\perp 3}$ are:

$$\check{F}_{\perp 12k} = \left(-\hat{M}_{1fwk} + \left(\check{F}_{L1k} - \check{F}_{L2k} \right) b \right) \frac{1}{a} \quad (5.64)$$

$$\check{F}_{\perp 3k} = \left(\hat{F}_{xfwk} - \left(\check{F}_{L1k} + \check{F}_{L2k} \right) \sin \hat{\epsilon}_{1k} - \check{F}_{\perp 12k} \cos \hat{\epsilon}_{1k} - \check{F}_{L3k} \sin \hat{\epsilon}_{3k} \right) \frac{1}{\cos \hat{\epsilon}_{3k}} \quad (5.65)$$

Thus, it is possible to provide an accurate measurement of $F_{\perp 3}$.

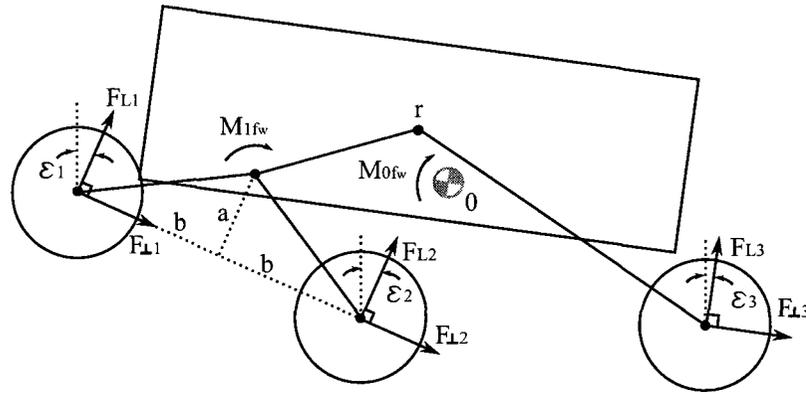


Figure 5.9: Forces applied at the wheel centroid shown in sensor directions.

The lumped force $\check{F}_{\perp 12}$ can then be apportioned based on the sensed loads \check{F}_{L1} and \check{F}_{L2} , which up to moderate values of $\{\epsilon_1, \epsilon_2\}$ approximately represent the normal loads

W_1 and W_2 .

$$\check{F}_{\perp 1k} = \frac{\check{F}_{\perp 12k} \check{F}_{L1k}}{\check{F}_{L1k} + \check{F}_{L2k}} \quad \check{F}_{\perp 2k} = \frac{\check{F}_{\perp 12k} \check{F}_{L2k}}{\check{F}_{L1k} + \check{F}_{L2k}} \quad (5.66)$$

The apportionment of forces is performed in this manner because the drawbar pull is larger for larger normal loads. However, assuming proportionality is an approximation, since the relationship between drawbar pull and normal load is not a proportional one. Additionally, $F_{\perp 1}$ and $F_{\perp 2}$ do not directly represent the drawbar pull, as can be seen in Equation 5.59. Nevertheless, apportioning the forces in this way provides a reasonable approximation. Ray et al. use a similar method to apportion drawbar pulls based on normal loads [11].

5.6.3 Measurement Model

The measurement model for the single-axis wheel force estimator h_3 is identical to that for the two-axis wheel force estimator.

$$\hat{\mathbf{y}}_{3k} = h_3(\hat{\mathbf{x}}_{3k}) = h_2(\hat{\mathbf{x}}_{3k}) \quad (5.67)$$

5.7 Creating the Net Traction Model

The result of running the estimators developed above is a set of estimated net traction values $\{\tau_R, DP, W, i\}$ for each wheel. These are all treated as sample points in the two net traction relationships: resistive torque τ_R as a function of normal load W and slip i , and drawbar pull DP as a function of normal load W and slip i . The sample points are filtered to exclude data collected when the rover is accelerating or decelerating above a certain threshold. The reason for this is twofold: Wong's terramechanics equations describe constant velocity operation and cannot be expected to hold during periods of acceleration and deceleration; additionally, since the measurement of

rover pitch is based on the gravity vector from the IMU, the estimator performs better without a corrupting acceleration. The estimated wheel angular accelerations $\{\hat{q}_2, \hat{q}_3, \hat{q}_4\}$ are used to determine whether or not to keep a data point.

Creating a two-dimensional polynomial fit with thousands of data points can be computationally expensive. To make it more manageable, sample data can be temporally averaged, decreasing the total number of sample points.

After filtering and temporal averaging, a two-dimensional polynomial function is fit to the sample points. The best order O to use for the polynomial fit will depend on the separation of the data points obtained from different wheels and the size of the sampled parameter space. For example, it is possible that two wheels had a normal load of approximately 60 N and that the third wheel had a normal load of approximately 30 N for the entire sampling period. In this case there would be a large gap in the parameter space due to separation of the sampled data. The same scenario is possible with slip. A low order fit should be used for data sets with large sample separation; a low order fit should also be used for sample data spanning a small parameter space. Fits between 2nd and 4th order were found to be appropriate.

The drawbar pulls, and to a lesser extent the normal loads, estimated using single-axis force sensors and Estimator 3 will be of lower accuracy on wheels 1 and 2 than on wheel 3. If desired, wheel 3 can be used exclusively to create the model, serving as a sensing wheel. Drawbar pull estimates using wheels 1 and 2 will be biased toward a proportional relationship between drawbar pull DP and normal load W because of the force apportioning performed using Equation 5.66.

5.8 Simulation Results and Discussion

A 200 s simulation of the rover traversing the gently rolling terrain shown in Figure 4.4 with a desired speed v_d of 2 cm/s was performed. White, Gaussian noise of the magnitude outlined in Table 5.1 was added to the sensor measurements. Figure 5.10 shows the real values, estimated values, and 3σ uncertainty bounds of the resistive torques τ_R and drawbar pulls DP over the simulation time period. The average error of each estimate over the simulation is shown in the title block of each plot.

Resistive torques are very well estimated because of the precise measurement of wheel angles $\{q_2, q_3, q_4\}$ (Figure 5.10a-c). Drawbar pulls are well estimated by Estimator 2 using a two-axis force sensor (Figure 5.10d-f). The small systematic error is caused by imperfect estimation of wheel-terrain contact angles $\{\gamma_1, \gamma_2, \gamma_3\}$. The velocity of the rover in its z direction \dot{z}_0^R is closer to zero in some periods of the simulation than in others. Since $\dot{z}_0^R = 0$ is assumed in the estimate of generalized velocity $\hat{\Phi}$, the estimates of wheel-terrain contact angles, which are very sensitive to velocity in the Z_w direction, will be inaccurate when \dot{z}_0^R is non-negligible. A systematic error in estimation of the wheel-terrain contact angles causes a similar error in the apportionment of sensed wheel forces into drawbar pulls and normal loads.

The white noise in drawbar pull is more prevalent for wheels 1 and 3 than for wheel 2. This is because the estimated wheel-terrain contact angle $\hat{\gamma}_2$ is less sensitive to noise in the estimated rover pitch rate $\hat{\theta}_{0_y}$ than the estimated wheel-terrain contact angles $\hat{\gamma}_1$ and $\hat{\gamma}_3$. Since the estimated rover pitch rate $\hat{\theta}_{0_y}$ is noisier than the other variables affecting the estimate of wheel-terrain contact angle, estimated wheel-terrain contact angles $\hat{\gamma}_1$ and $\hat{\gamma}_3$ experience more white noise than does $\hat{\gamma}_2$.

As expected, the performance of the wheel force estimator using single-axis force sensors is poorer than that using two-axis force sensors for wheels 1 and 2 (Figure 5.10g-h). Since the drawbar pull and normal load do not follow a proportional relationship as implied by Equation 5.66, there is an additional source of systematic error for wheels 1 and 2. However, the drawbar pull for wheel 3 is estimated with an accuracy nearly identical to that when using two-axis force sensors.

Figure 5.11 shows the estimated normal loads and slips over the 200 s simulation. Normal load is accurately estimated in both Estimator 2 and Estimator 3; since both estimators perform nearly identically, the results from Estimator 3 not shown. Normal load is far less sensitive than drawbar pull to the aforementioned errors in estimated wheel-terrain contact angles $\{\hat{\gamma}_1, \hat{\gamma}_2, \hat{\gamma}_3\}$. Additionally, it is nearly directly sensed by force sensors in both estimators.

Slips are also accurately estimated, with some white noise. Slip measurements are much less sensitive to the assumption that $\dot{z}_0^R = 0$ than are the estimates of wheel-terrain contact angles; as a result, slip estimates have minimal systematic error.

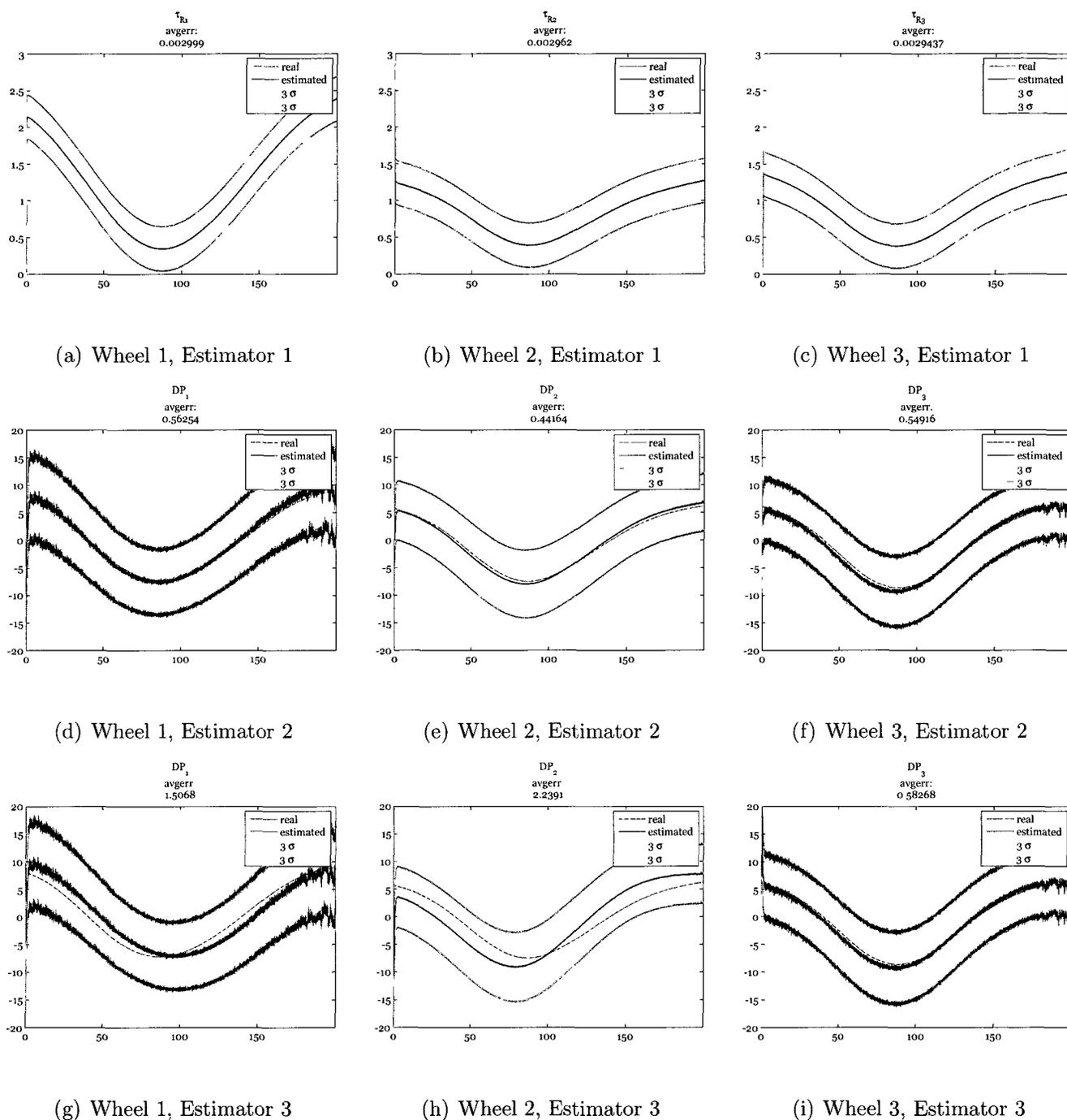


Figure 5.10: Estimated resistive torques τ_R and drawbar pulls DP over the 200 s simulation. Time is on the x axis in seconds. **(a)-(c)** Resistive torques estimated using Estimator 1 [Nm]. **(d)-(f)** Drawbar pulls estimated with two-axis force sensors and Estimator 2 [N]. **(g)-(i)** Drawbar pulls estimated with single-axis force sensors and Estimator 3 [N]. The average error of each estimate over the simulation is shown in the title block of each plot.

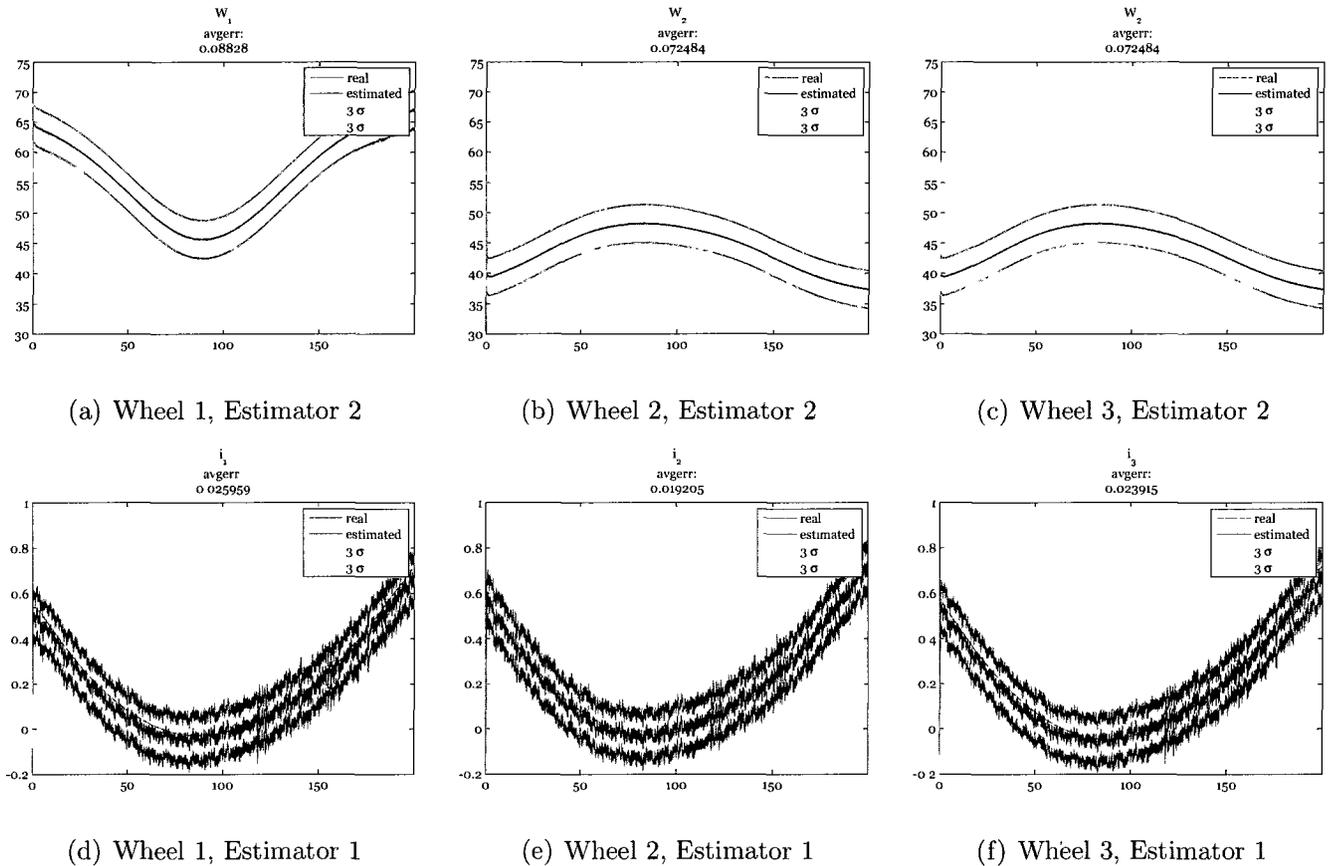


Figure 5.11: Estimated normal loads W and slips i over the 200 s simulation. Time is on the x axis in seconds. (a)-(c) Normal load estimated using Estimator 2 [N]. (d)-(f) Slip estimated using Estimator 1 [-]. The average error of each estimate over the simulation is shown in the title block of each plot.

The estimated data was filtered to exclude data for which the estimated wheel accelerations were greater than 6.079×10^{-3} radians/s. To reduce the amount of data, the estimates were temporally averaged to 5 samples/s from the original sampling rate of 50 samples/s.

Figures 5.12, 5.13, and 5.14 depict the reformulated net traction models alongside their true local models; the estimated fits are formed entirely using estimated data points, with no *a priori* knowledge of the net traction relationships. Figure 5.12 shows the resistive torque data and models using Estimators 1 and 2; Figure 5.13 shows the

drawbar pull data and models using Estimators 1 and 2; and Figure 5.14 shows the drawbar pull data and models using Estimators 1 and 3. The estimated points, shown as blue \circ 's, represent the estimated resistive torques and drawbar pulls in terms of estimated slip and estimated normal load; the co-ordinates of the estimated points are then $(\hat{\iota}, \hat{W}, \hat{\tau}_R)$ in the resistive torque plot, and $(\hat{\iota}, \hat{W}, \hat{DP})$ in the drawbar pull plots. Data from all three wheels is used. The ground truth points, shown as green \times 's, represent the true resistive torques and drawbar pulls experienced on the rover's traverse in terms of true slip and true normal load; the co-ordinates of the ground truth points are then (ι, W, τ_R) for the resistive torque plot, and (ι, W, DP) for the drawbar pull plots. Data from all three wheels is again used. A two-dimensional, 2nd order, polynomial function, with the same form as Equation 2.28, was fit to the estimated data; the estimated fit is represented by a blue surface and is formed entirely from scratch using only estimated data from the rover's traverse. The actual terramechanics models were approximated using 2nd order polynomial fits over the same parameter space; they are represented by red surfaces. Thus, there are two types of polynomial functions that can be compared in order to judge the efficacy of the estimation algorithm: the estimated fits (blue) and the true local models (red). The coefficients for the estimated models are compared with the true local models in Table 5.2.

Examining the plots and the coefficients of the polynomial functions, it is clear that the estimated fits provide a good approximation of the actual net traction relationships over the traversed parameter space of $(-0.2 < \iota < 0.6)$ and $(30 \text{ N} < W < 70 \text{ N})$. As would be expected, the estimated fits are more accurate for normal loads and slips close to the sampled data. The ability to reconstruct accurate net traction models for resistive torque τ_R and drawbar pull DP as a function of normal load W and slip ι , using the estimators presented in this thesis, has been demonstrated.

Table 5.2: Coefficients of the 2nd order polynomial true local model and 2nd order polynomial fit to the estimated data. The coefficients describe the fits shown in Figures 5.12, 5.13, and 5.14.

	τ_R True Local Model	$\hat{\tau}_R$ Fit, Estimator 1	DP True Local Model	$\hat{D}P$ Fit, Estimator 2	$\hat{D}P$ Fit, Estimator 3
p_{00}	-0.36555	-0.1486	-4.1510	7.0170	11.652
p_{10}	2.1782	2.5482	31.811	41.150	45.136
p_{01}	0.013773	0.0013583	-3.3989×10^{-2}	-0.55302	-0.73482
p_{20}	-2.7207	-2.5991	-39.134	-41.883	-41.933
p_{11}	0.026712	0.014448	0.27556	0.068862	-2.0886×10^{-2}
p_{02}	8.2251×10^{-5}	2.6943×10^{-4}	-1.6247×10^{-4}	5.9339×10^{-3}	7.7134×10^{-3}

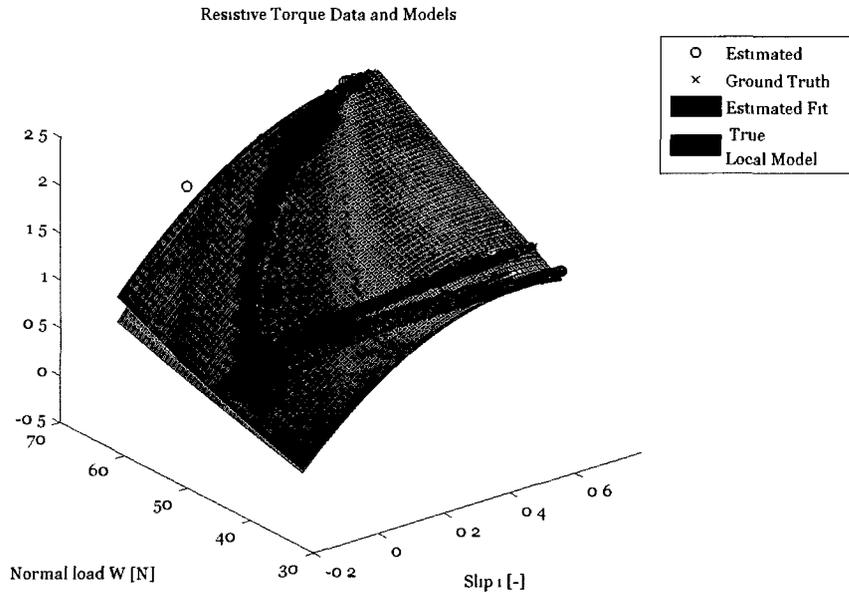


Figure 5.12: Resistive torque data and polynomial fits using Estimators 1 and 2. Estimator 1 was used for the slip \hat{i} and resistive torque $\hat{\tau}_R$ data; Estimator 2 was used for the normal load \hat{W} data.

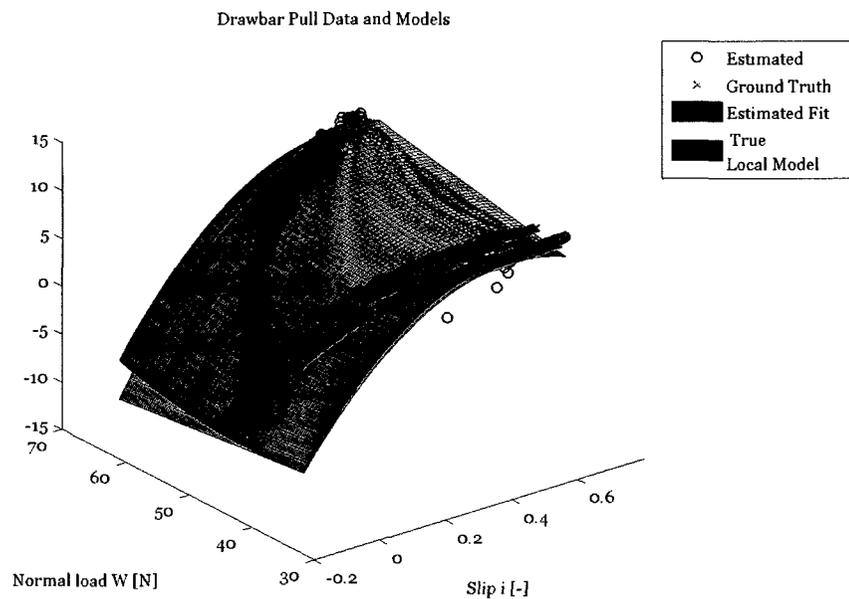


Figure 5.13: Drawbar pull data and polynomial fits using Estimators 1 and 2. Estimator 1 was used for the slip \hat{i} data; Estimator 2 was used for the normal load \hat{W} and drawbar pull $\hat{D}P$ data.

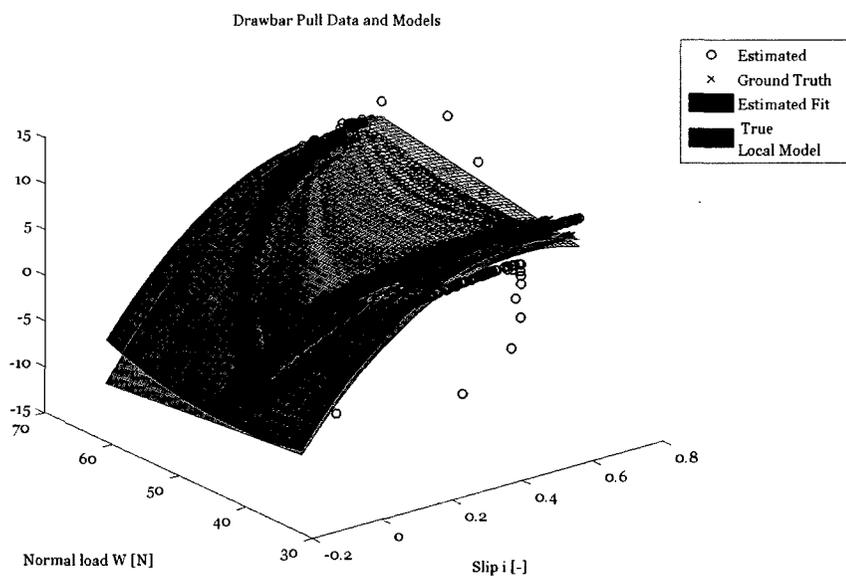


Figure 5.14: Drawbar pull data and polynomial fits using Estimators 1 and 3. Estimator 1 was used for the slip \hat{i} data; Estimator 3 was used for the normal load \hat{W} and drawbar pull $\hat{D}P$ data.

5.9 Proposed Usage of Estimator

Ideally, the estimator would be run in real-time, providing current estimates of resistive torques, drawbar pulls, wheel normal loads, wheel slips, and wheel-terrain contact angles, amongst other states. Knowledge of these states could be used to perform real-time traction control, with forces being optimally applied, or high slip conditions being avoided [5, 16, 41]. An algorithm estimating all of these mobility-related states in unison does not, to the author's knowledge, yet exist; thus there exists great potential for the development of new traction control algorithms. The estimator simulated in MATLAB in this thesis is not capable of running in real-time; its optimization for speed is outside of the scope of this thesis.

The end result of the net traction estimation algorithm will be polynomial models of resistive torque τ_R and drawbar pull DP as functions of normal load W and slip ι . It is possible that the polynomial models created could contain information about the soil properties of the terrain, whose recovery would represent increased scientific return of a mission. Creating these models using polynomial fitting techniques requires numerous estimated data points; it is necessary that the rover drive for a certain amount of time in order to collect enough estimated data. The net traction estimator is controller independent, meaning that data collected using anything from simple PID control to sophisticated traction control can be used. Creation of the polynomial model from estimated data would likely be performed when the rover is stopped. The net traction estimation algorithm requires that the terrain be homogeneous and gently rolling. Forward looking range data could be used to assess the character of the terrain ahead, and perhaps to autonomously make a decision about whether or not to perform net traction estimation. Throughout the rover's mission, the soil properties of the terrain over which it is driving may change, changing the net traction relationships. To mitigate the negative consequences of using an outdated net traction model, only

a certain number of the latest estimated data points could be used. This way, the model would gradually adjust itself to match the new net traction relationships.

Chapter 6

Conclusions and Recommendations

The goal of this thesis is to develop a method to perform terrain estimation using the rocker-bogie mobility system. This problem was approached from the perspectives of terramechanics, hardware, simulation, and software. A net traction model was presented in Chapter 2. Polynomial fitting of terramechanics equations provided a means to perform faster simulations, as well as to rebuild the net traction model. The design of *Kapvik*'s rocker-mobility system hardware was presented in Chapter 3. Analysis of the drive system, differential mechanism, structural integrity and static stability were presented. Force sensors above the wheel hubs were added to the typical rocker-bogie design to facilitate future work on net traction estimation. The application of multibody dynamics to articulated mobile robots was presented in Chapter 4. A fully dynamic two-dimensional simulation of a rocker-bogie equipped rover traversing rolling terrain with a proper wheel-soil interaction model was created. Net traction estimation was developed for two different classes of force sensors in Chapter 5. Two-dimensional polynomial models of the critical net traction relationships, resistive torque τ_R and drawbar pull DP , were constructed without *a priori* knowledge of the soil properties using estimators developed in this thesis. Performance of the net traction estimation algorithm was validated in simulation using realistic sensor noises and soil properties. Thus, the goal of this thesis was fulfilled.

6.1 Summary of Contributions

1. The creation of a method to accurately approximate the net traction relationships for resistive torque τ_R and drawbar pull DP using a two-dimensional polynomial approximation. This method of approximation both expedites simulation and allows for the net traction relationships to be reformulated from estimated data points.
2. The detailed design of a rocker-bogie mobility system for the *Kapvik* micro-rover, including: design of the wheel drive system; design the differential mechanism; integration of force sensors above the wheel hubs; structural analysis; and static stability analysis.
3. The development of a fully dynamic, two-dimensional multibody simulation of a rover with a rocker-bogie mobility system traversing rolling terrain. Realistic wheel-soil interaction forces are obtained using Wong's terramechanics formulae.
4. An algorithm for net traction estimation using the rocker-bogie mobility system and a suite of sensors. The estimator can be used either with two-axis or single-axis force sensors above the wheel hubs. Wheel-soil contact angles and wheel slips, important variables in traction control, are also estimated.

6.2 Recommendations for Future Work

This thesis involved several disciplines and the creation of *Kapvik*, a terrestrial prototype of a planetary micro-rover. As a result, there are a large number of opportunities for future work.

Investigation into whether soil properties can be recovered from the estimated net traction data points and/or the two-dimensional polynomial net traction model could

be undertaken. Since the data points contain a full record of resistive torques τ_R and drawbar pulls DP as a function of normal loads W and slips i , it is quite possible that several important soil properties could be extracted, increasing the scientific return of *Kapvik*, or a similar rover. A method similar to Iagnemma et al.'s least squares estimation is a likely starting point [10].

A more detailed study of the appropriate order O of the polynomial approximation to use in different separations of estimated data would help to automate the creation of the net traction model. Creating a grid system in the parameter space of normal load W and slip i would allow for assessment of data sparsity or separation. By studying an occupancy grid, it is possible that through trial and error or through a mathematically rigorous method, an optimal order O could be found that decreases the likelihood of inaccurate interpolation between data points.

If net traction estimation is desired on future iterations of *Kapvik*, some improvements to the sensors should be implemented. A two-axis force sensor equivalent to that shown in Figure 5.6b should be installed; the hollow shaft potentiometers should be replaced by 16-bit grey code hollow-shaft absolute encoders as shown in Figure 5.5b; a downward facing velocimeter should be installed to measure the rover's forward velocity. These changes would make it possible to experimentally achieve the same performance as the net traction estimator presented in this thesis.

Extending the multibody simulations to three dimensions would be laborious but possible. The treatment of constraints may need to be changed and side-slip of wheels would need to be considered. Ishigami presents a three dimensional simulation of a four wheeled rover with a rocker and differential in his thesis [2]. Extending the simulation environment to three dimensions would allow for a full comparison of simulated and real-world results. A well instrumented "Mars yard", under construction at Carleton University at the time of writing, would provide an ideal test environment

for net traction estimation.

Creating a traction control algorithm is far easier when a model for the net traction relationships is available than when it is not. The net traction estimation algorithm presented in this thesis could be run under PID control for a short period of time to establish the polynomial model. Once the model has converged acceptably, the PID controller could be switched off, and a traction controller, using the acquired model, could be switched on. Since the estimator is controller independent, it could continue to run; thus, continually updated models of the net traction relationships would be available to the controller. The net traction model would be most accurate in predicting performance close to the rover's current or previously seen operating point. A method for optimal traction control is outlined by Iagnemma [5].

For rovers performing long traverses, net traction relationships can change with location. When this arises, it may be advantageous to form the net traction model based solely on recent data points. Older data points could be decreased in significance until they are completely excluded from the formation of the polynomial model. This possibility could be investigated, particularly for rovers performing long traverses, or driving in an area with small patches of dissimilar soil. Additionally, net traction models could be monitored for potentially dangerous terrain. For example, when in a soil for which positive drawbar pull DP is only possible with high slip i , the rover is more likely to dig itself into a rut and become permanently stuck. In dangerous cases such as this, the rover could reverse or proceed cautiously.

List of References

- [1] R. Lindemann and C. Voorhees, “Mars Exploration Rover Mobility Assembly Design, Test and Performance,” *2005 IEEE International Conference on Systems, Man and Cybernetics*, pp. 450–455, 2005.
- [2] G. Ishigami, *Terramechanics-based Analysis and Control for Lunar / Planetary Exploration Robots*. Doctor of philosophy thesis, Tohoku University, 2008.
- [3] C. A. Brooks, *Terrain Identification Methods for Planetary Rovers*. Master of science thesis, Massachusetts Institute of Technology, 2004.
- [4] NASA, “Now a Stationary Research Platform , NASA’s Mars Rover Spirit Starts a Mars Exploration Rover Mission.” Internet, <http://marsrover.nasa.gov/newsroom/pressreleases/20100126a.html> [July 26, 2011].
- [5] K. Iagnemma, “Traction Control of Wheeled Robotic Vehicles in Rough Terrain with Application to Planetary Rovers,” *The International Journal of Robotics Research*, vol. 23, pp. 1029–1040, Oct. 2004.
- [6] J. Y. Wong, *Theory of Ground Vehicles*. John Wiley & Sons, 4th ed., 2008.
- [7] J. Wong and A. Reece, “Prediction of rigid wheel performance based on the analysis of soil-wheel stresses: Part I. Performance of Driven Rigid Wheels,” *Journal of Terramechanics*, vol. 4, no. 1, pp. 81–98, 1967.
- [8] J. Wong and A. Reece, “Prediction of rigid wheel performance based on the analysis of soil-wheel stresses: Part II. Performance of towed rigid wheels,” *Journal of Terramechanics*, vol. 4, no. 2, pp. 7–25, 1967.
- [9] H. Nakashima, H. Fujii, A. Oida, M. Momozu, H. Kanamori, S. Aoki, T. Yokoyama, H. Shimizu, J. Miyasaka, and K. Ohdoi, “Discrete element method analysis of single wheel performance for a small lunar rover on sloped terrain,” *Journal of Terramechanics*, vol. 47, pp. 307–321, Oct. 2010.

- [10] K. Iagnemma, H. Shibly, and S. Dubowsky, "On-line terrain parameter estimation for planetary rovers," in *Proceedings 2002 IEEE International Conference on Robotics and Automation*, (Washington, DC), pp. 3142–3147, IEEE, 2002.
- [11] L. R. Ray, D. C. Brande, and J. H. Lever, "Estimation of net traction for differential-steered wheeled robots," *Journal of Terramechanics*, vol. 46, no. 3, pp. 75–87, 2009.
- [12] L. Ding, H. Gao, Z. Deng, K. Nagatani, and K. Yoshida, "Experimental study and analysis on driving wheels performance for planetary exploration rovers moving in deformable soil," *Journal of Terramechanics*, vol. 48, pp. 27–45, Feb. 2011.
- [13] D. B. Bickler, "Articulated Suspension System." US Patent 4840394, 1989.
- [14] J. Chottiner, *Simulation of a Six Wheel Martian Rover Called the Rocker Bogie*. Master of science thesis, The Ohio State University, 1992.
- [15] H. Hacot, *Analysis and Traction Control of a Rocker-Bogie Planetary Rover*. Master of science thesis, Massachusetts Institute of Technology, 1998.
- [16] K. Iagnemma, *Rough-Terrain Mobile Robot Planning and Control with Application to Planetary Exploration*. Doctor of philosophy thesis, Massachusetts Institute of Technology, 2001.
- [17] M. Tarokh and G. McDermott, "Kinematics modeling and analyses of articulated rovers," *IEEE Transactions on Robotics*, vol. 21, pp. 539–553, Aug. 2005.
- [18] L. Ding, K. Nagatani, K. Sato, A. Mora, K. Yoshida, H. Gao, Z. Deng, and R. Analysis, "Terramechanics-based High-Fidelity Dynamics Simulation for Wheeled Mobile Robot on Deformable Rough Terrain," in *2010 IEEE International Conference on Robotics and Automation*, pp. 4922–4927, 2010.
- [19] H. J. Moore, G. D. Clow, and R. E. Hutton, "A Summary of Viking Sample-Trench Analyses for Angles of Internal Friction and Cohesions," *Journal of Geophysical Research*, vol. 87, no. B12, pp. 10043–10050, 1982.
- [20] Rover Team, "Characterization of the Martian Surface Deposits by the Mars Pathfinder Rover, Sojourner," *Science*, vol. 278, pp. 1765–1768, Dec. 1997.
- [21] H. J. Moore, D. B. Bickler, J. A. Crisp, H. J. Eisen, J. A. Gensler, A. F. C. Haldermann, J. R. Matijevic, L. K. Reid, and F. Pavlics, "Soil-like deposits observed by Sojourner, the Pathfinder rover," *Journal of Geophysical Research*, vol. 104, no. E4, pp. 8729–8746, 1999.

- [22] H. Shibly, K. Iagnemma, and S. Dubowsky, “An equivalent soil mechanics formulation for rigid wheels in deformable terrain, with application to planetary exploration rovers,” *Journal of Terramechanics*, vol. 42, pp. 1–13, Jan. 2005.
- [23] D. Lhomme-Desages, C. Grand, and J. Guinot, “Trajectory control of a four-wheel skid-steering vehicle over soft terrain using a physical interaction model,” in *2007 IEEE International Conference on Robotics and Automation*, pp. 1164–1169, IEEE, 2007.
- [24] L. Ding, H. Gao, Z. Deng, K. Yoshida, and K. Nagatani, “Slip Ratio for Lugged Wheel of Planetary Rover in Deformable Soil: Definition and Estimation,” in *The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3343–3348, 2009.
- [25] H. B. Pacejka and E. Bakker, “The Magic Formula Tyre Model,” *Vehicle System Dynamics*, vol. 21, no. S1, pp. 1–18, 1992.
- [26] G. Heiken, D. Vaniman, and B. French, *Lunar sourcebook: a user’s guide to the moon*. CUP Archive, 1991.
- [27] H. J. Eisen, C. W. Buck, G. R. Gillis-Smith, and J. W. Umland, “Mechanical Design of the Mars Pathfinder Mission,” tech. rep., Jet Propulsion Laboratory, 1997.
- [28] Maxon Motors, *Maxon Program 09/10*. Maxon Motors, 09/10 ed., 2009.
- [29] S. Krishnan and C. Voorhees, “The Use of Harmonic Drives on NASA’s Mars Exploration Rover,” in *Harmonic Drive International Symposium*, 2001.
- [30] Harmonic Drive Gearheads, “CSF Mini Series.” Internet, <http://www.harmonicdrive.net/media/support/catalogs/pdf/csf-mini-5-14.pdf> [October 13, 2009].
- [31] Sherborne Sensors, “SS4000M Series Miniature Universal Load Cell.” Internet, <http://www.sherbornesensors.com/dyn/datasheets/ss4000m-iss4.pdf> [June 24, 2010].
- [32] THK, “RSR-Z Linear Motion Guide.” Internet, https://tech.thk.com/en/products/pdfs/en_b1.130.pdf [December 10, 2010].
- [33] D. Dimitrov, “Multibody simulation dynamics of a multibody system (Euler-Lagrange formulation),” in *Multibody Simulation Course Notes*, Orebro University, 2009.

- [34] J. Garcia de Jalon and J. Bayo, *Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time Challenge*. Springer-Verlag, 1994.
- [35] D. Simon, *Optimal State Estimation*. John Wiley & Sons, 2006.
- [36] S. J. Julier and J. K. Uhlmann, “New extension of the Kalman filter to nonlinear systems,” *Proceedings of SPIE*, vol. 3068, pp. 182–193, 1997.
- [37] Inscale Measurement Technology Ltd., “Series GL Hollow Shaft Sensors.” Internet, <http://www.inscale-potentiometers.co.uk/pdf/gl.pdf> [August 10, 2010].
- [38] Avagao Technologies, “AEAS-7000 Ultra-precision 16 bit Gray Code Absolute Encoder Module.” Internet, <http://www.avagotech.com/docs/5989-4140EN> [June 29, 2011].
- [39] ATI Industrial Automation, “ATI Force/Torque Sensor: Nano25.” Internet, http://www.ati-ia.com/products/ft/ft_models.aspx?id=Nano25 [July 25, 2011].
- [40] Memsense, “H3-IMU High Performance Inertial Measurement Unit.” Internet, http://www.memsense.com/docs/H3IMU/DSS-1001_Rev-B_H3-IMU.pdf [May 20, 2011].
- [41] K. Yoshida, H. Hamano, and T. Watanabe, “Slip-based Traction Control of a Planetary Rover,” in *Experimental Robotics VIII*, pp. 644–653, Springer, 5th ed., 2003.

Appendix A

Polynomial Traction Approximations

A.1 Ding Soil Simulant

The coefficients for the 4th order polynomial approximation to the traction parameters using the lunar soil simulant properties from Table 2.2 are shown in Table A.1.

Table A.1: Coefficients of the 4th order polynomial model of drawbar pull DP and resistive torque τ_R for Ding et al.'s soil simulant. Evaluation is performed using Equation 2.28, and the equivalent equation for τ_R .

Coefficient	Drawbar Pull DP	Resistive Torque τ_R
p_{00}	-0.6941499871492659	-0.074692940580283043
p_{10}	12.356691584745722	0.87250171008904021
p_{01}	-0.14547626576234601	0.0025510696731966612
p_{20}	-51.172683793239997	-3.5754007449661298
p_{11}	1.0186604150069647	0.077927916745546
p_{02}	0.00077088050338964945	0.00022574422704184664
p_{30}	73.685592302972822	5.113008216555694
p_{21}	-0.92078957333278266	-0.066560998614018846
p_{12}	-0.0037098437195073634	-0.00026073204427577175
p_{03}	$-7.6064003510828028 \times 10^7$	$-6.9834433035713326 \times 10^7$
p_{40}	-33.969670188744821	-2.3538819513518461
p_{31}	0.2697640745286296	0.020589265359729405
p_{22}	0.0011396447157889238	0.00010285518082128865
p_{13}	$4.0276952489236379 \times 10^6$	$2.9133317838241482 \times 10^7$
p_{04}	$-2.3251679995188414 \times 10^9$	$9.9975206352688892 \times 10^{10}$

A.2 Lunar Soil

The coefficients of the 4th order polynomial approximation to the traction parameters using the recommended lunar soil properties from the “Lunar Sourcebook” [26] are shown in Table A.2. The lunar soil properties used were: cohesion $c = 170$ Pa; internal friction angle $\phi_s = 35^\circ$; shear deformation parameter $K_s = 17.8$ mm; deformation exponent $n_s = 1$; modulus of cohesion $k_c = 1.4$ kPa; and modulus of friction $k_\phi = 820$ kPa/m [26]. Dry sand values for c_1 and c_2 were used (i.e. $c_1 = 0.18$ and $c_2 = 0.32$) [7]. The average errors in the approximated functions were 0.037682 N for drawbar pull, and 0.0023752 Nm for resistive torque. Plots of the polynomial approximations are shown in Figure A.1. The parameter space was restricted to $0 \leq i \leq 1$ and $0.1 \text{ N} \leq W \leq 75 \text{ N}$ because above 75 N of normal load, the wheel-soil contact angle θ_0 exceeded 90° .

Table A.2: Coefficients of the 4th order polynomial model of drawbar pull DP and resistive torque τ_R for lunar soil. Evaluation is performed using Equation 2.28, and the equivalent equation for τ_R .

Coefficient	Drawbar Pull DP	Resistive Torque τ_R
p_{00}	-0.32105374537956471	-0.030083337889648551
p_{10}	3.9731094735808967	0.21914763075672361
p_{01}	-0.15332437307337637	0.0048677080187868567
p_{20}	-12.465213892689725	-0.69783601074798929
p_{11}	0.76462077735366407	0.06322086975638018
p_{02}	0.0023582023048187048	0.00080963089584658548
p_{30}	14.899457777750635	0.85322110414713626
p_{21}	-0.474149273373837	-0.040696869962136409
p_{12}	-0.011223389436071415	-0.00085335807749506865
p_{03}	$-1.0014815342230956 \times 10^{-5}$	$-7.0299522227661702 \times 10^{-6}$
p_{40}	-5.8783305810824125	-0.34550123020828138
p_{31}	0.061512790614153663	0.0089135686199323539
p_{22}	0.0022124624564960828	0.00029053384009854216
p_{13}	$3.6344047637651598 \times 10^{-5}$	$3.3660207854645556 \times 10^{-6}$
p_{04}	$-6.6123839669122175 \times 10^{-8}$	$2.5930452515527823 \times 10^{-8}$

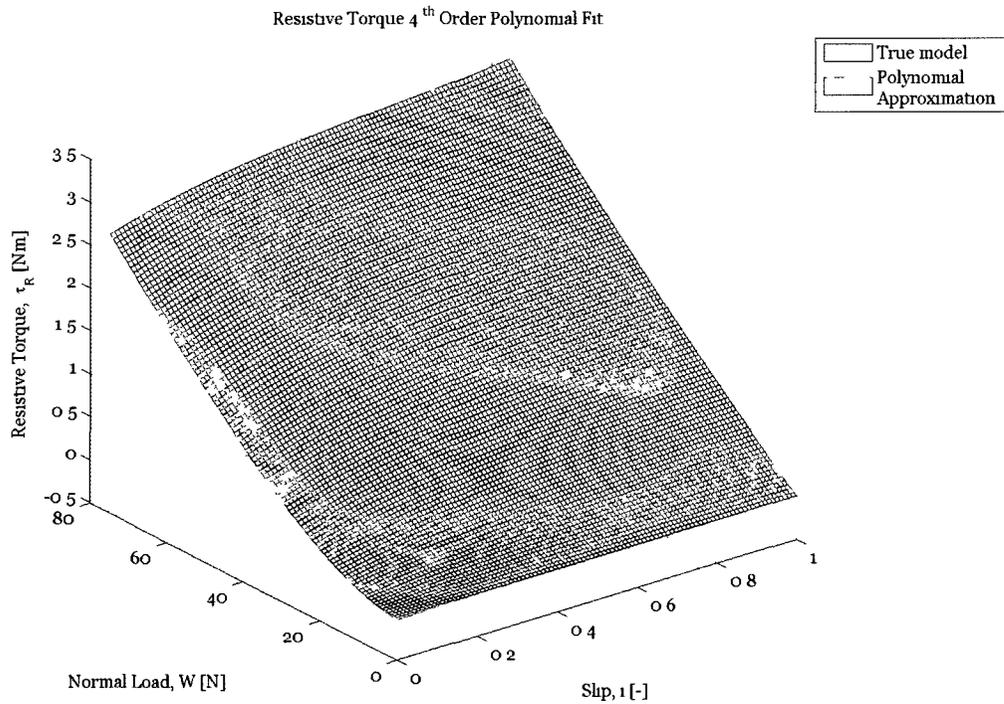
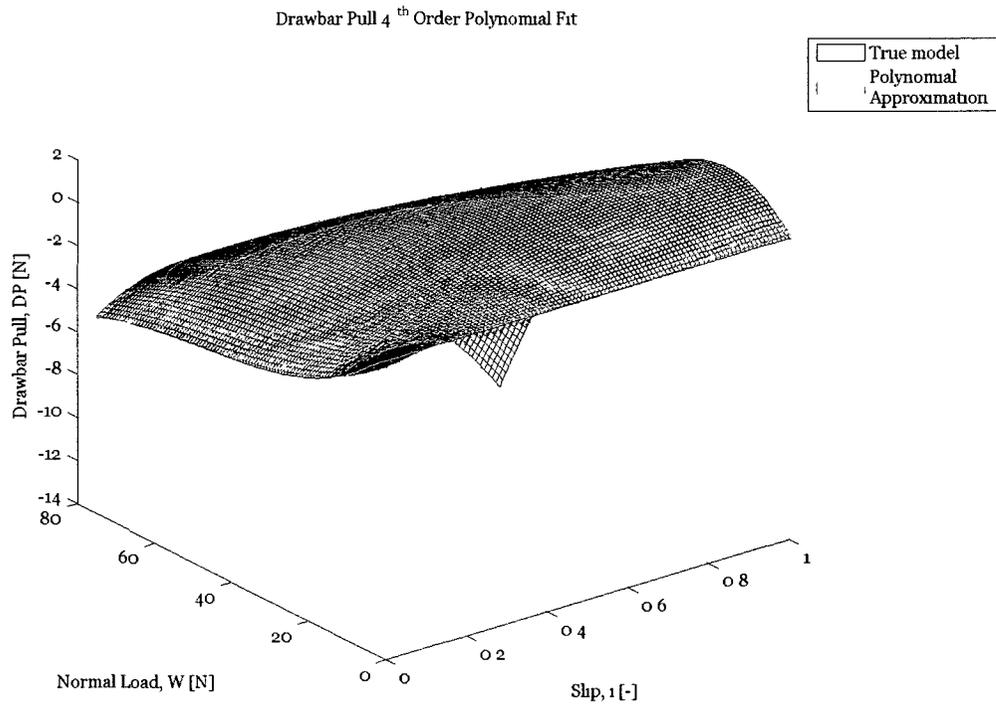


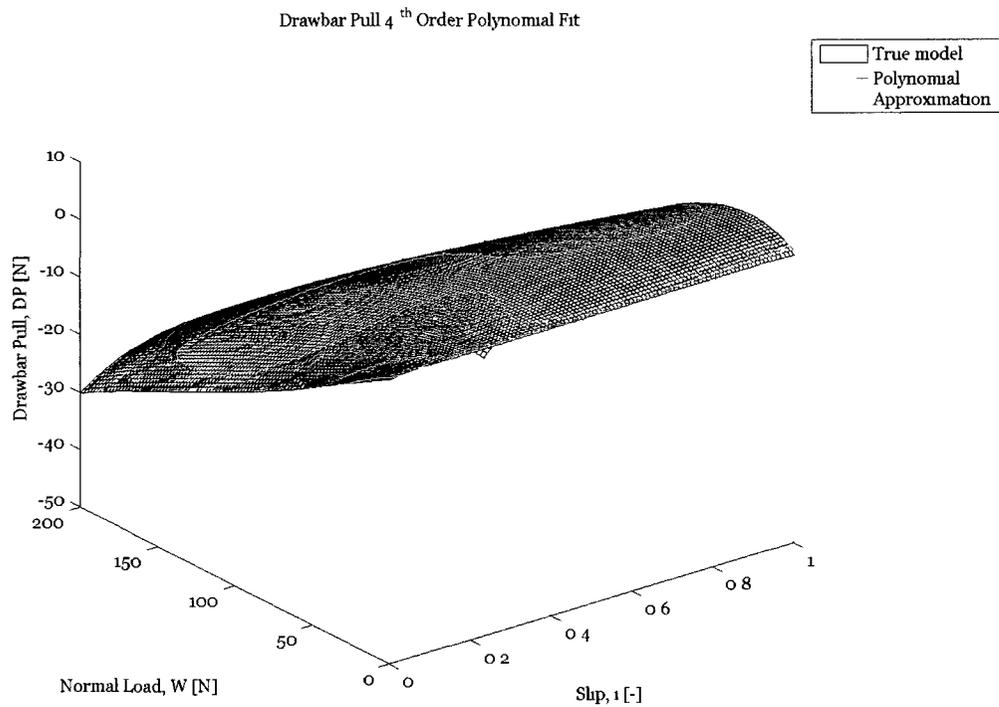
Figure A.1: Net traction parameters as a function of W and i , and their corresponding 4th order polynomial approximation for lunar soil.

A.3 Dry Sand

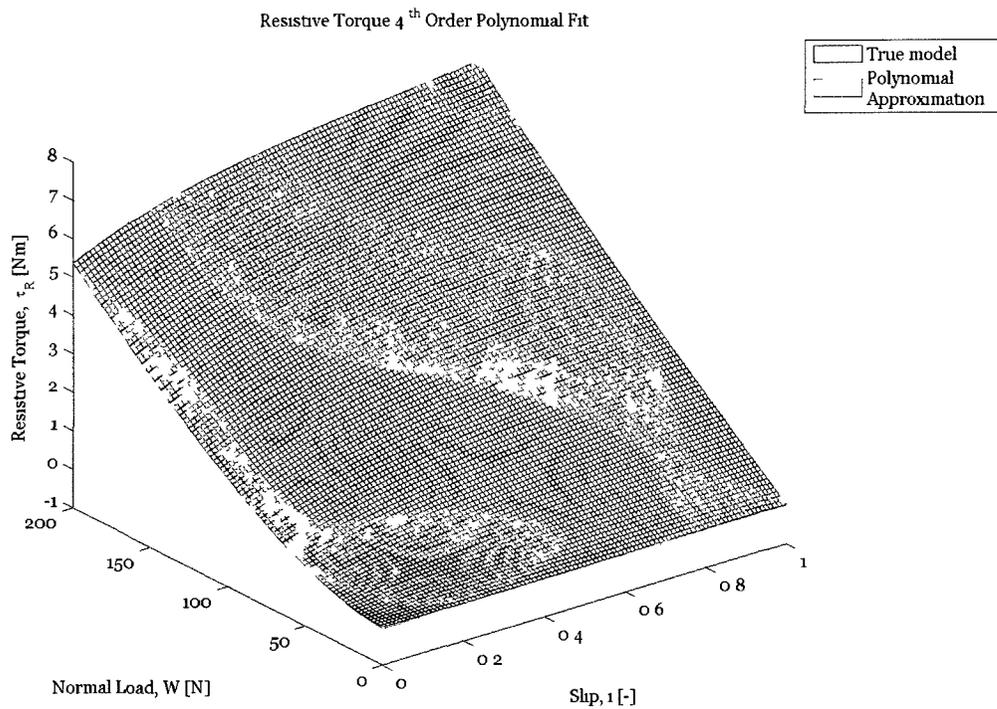
The coefficients of the 4th order polynomial approximation to the traction parameters using dry sand soil properties in are shown in Table A.3. The dry sand soil properties used were: cohesion $c = 1040$ Pa; internal friction angle $\phi_s = 28^\circ$; shear deformation parameter $K_s = 17.5$ mm; deformation exponent $n_s = 1.1$; modulus of cohesion $k_c = 0.99$ kPa/m $^{n_s-1}$; and modulus of friction $k_\phi = 1528.43$ kPa/m [6]. Dry sand values for c_1 and c_2 were used (i.e. $c_1 = 0.18$ and $c_2 = 0.32$) [7]. The average errors in the approximated functions were 0.081406 N for drawbar pull, and 0.0040465 Nm for resistive torque. Plots of the polynomial approximations are shown in Figure A.2.

Table A.3: Coefficients of the 4th order polynomial model of drawbar pull DP and resistive torque τ_R for dry sand. Evaluation is performed using Equation 2.28, and the equivalent equation for τ_R .

Coefficient	Drawbar Pull DP	Resistive Torque τ_R
p_{00}	-0.43634860986048812	-0.048548957940936721
p_{10}	8.0391549475404549	0.44477880843841894
p_{01}	-0.15964302178324102	0.0034437525653181339
p_{20}	-26.88374042429561	-1.5306057969972653
p_{11}	0.70331116406455385	0.058235021532663821
p_{02}	-0.00019112338240838124	0.00020026074414595776
p_{30}	35.311211792900018	2.0841767416645345
p_{21}	-0.49679615820237949	-0.04089965845449986
p_{12}	-0.0031864658669679821	-0.00023359914680999097
p_{03}	$3.3460890098786386 \times 10^{-6}$	$-5.4199020346198563 \times 10^{-7}$
p_{40}	-15.038162064115175	-0.91747236834681123
p_{31}	0.095561833031966165	0.010706738123251739
p_{22}	0.00057418539873414689	$8.3146573858595469 \times 10^{-5}$
p_{13}	$3.564385354916015 \times 10^{-6}$	$3.0533571621104358 \times 10^{-7}$
p_{04}	$-1.0626391803475609 \times 10^{-8}$	$6.5654910931650938 \times 10^{-10}$



(a) Drawbar pull DP true model (black) and polynomial approximation (green).



(b) Resistive torque τ_R true model (black) and polynomial approximation (green).

Figure A.2: Net traction parameters as a function of W and i , and their corresponding 4th order polynomial approximation for dry sand.

Appendix B

Kapvik Chassis Design Documentation

B.1 Overall Dimensions

The overall dimensions of the *Kapvik* chassis are shown in Figure B.1.

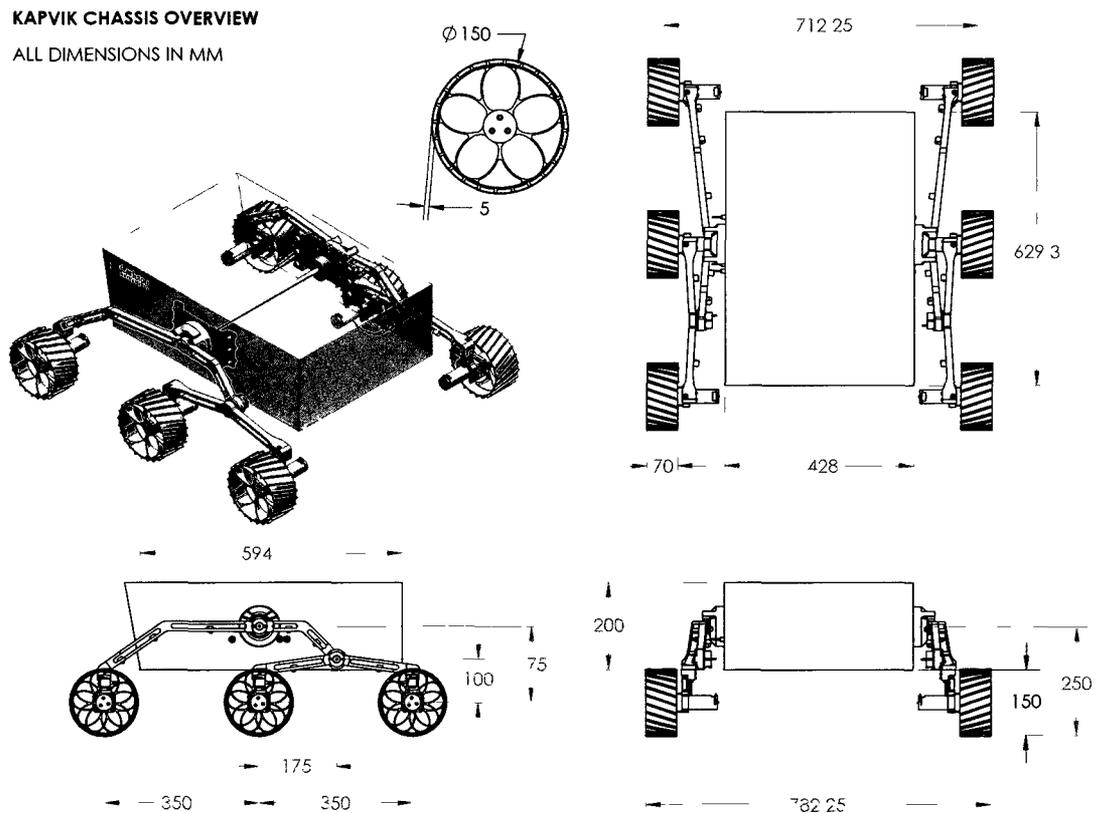


Figure B.1: Overall *Kapvik* chassis dimensions.

B.2 Rocker Assembly

An exploded view of the rocker assembly is shown in Figure B.2. The wheel drive system mounts to the front of the rocker via five screws: four to mount the linear guide rail and one to mount the single-axis force sensor. The rocker is connected to the differential at its center using just three screws. *Kapvik* was designed to be modular at this point, so that future generations can use tracks or an elastic loop mobility system. Requiring only three screws at the interface ensures the ability to quickly change the chassis. The potentiometer is attached to a flat on a mounting bracket with a single screw. These two parts together are mounted to the rocker using four screws. Five wiring clips are placed along the rocker for routing of the motor power, the encoder signals, the load cell signals, and the potentiometer signals.

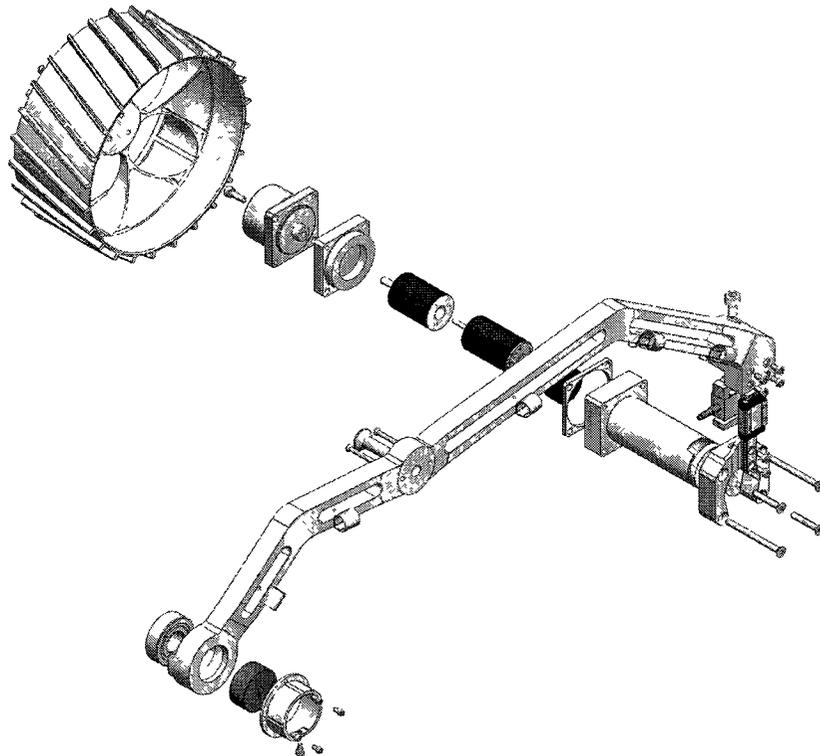


Figure B.2: An exploded view of the *Kapvik* rocker assembly.

B.3 Bogie Assembly

An exploded view of the bogie assembly is shown in Figure B.3. A wheel drive system is mounted to each end of the bogie in an identical manner as to the rocker assembly. At its center, the bogie mounts to a small shaft using a single screw and two dowel pins that ensure alignment and transfer rotational motion. Two wiring clips are placed along the bogie for routing of the motor power, the encoder signals, and the load cell signals.

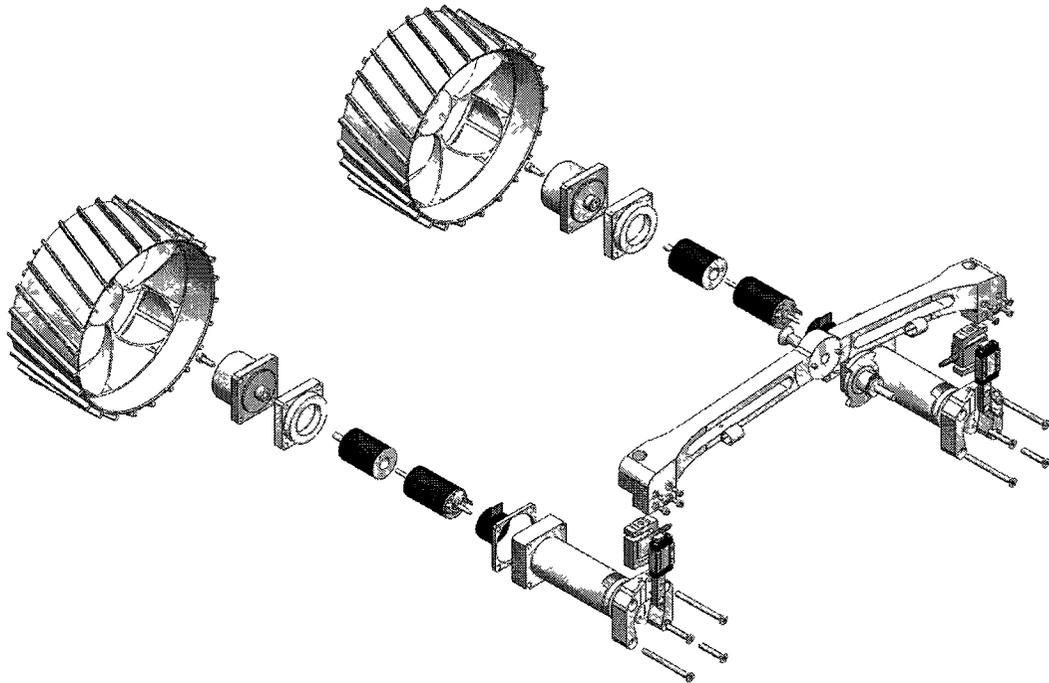


Figure B.3: An exploded view of the *Kapvik* bogie assembly.

B.4 Bogie Joint

An exploded view of the bogie assembly side of the bogie joint is shown in Figure B.4a, and an exploded view of the rocker assembly side of the bogie joint is shown in Figure B.4b. The shaft attached to the bogie is fit through a shielded ball bearing

with a tight sliding fit. It is retained on the other side with an external retaining ring. The shaft extends through the hollow shaft potentiometer; the potentiometer's rotating component is coupled to the central shaft using a set screw. The motion of the bogie is limited to $\pm 30^\circ$ in order to prevent dangerous configurations of the mobility system; this is done through the use of hard stops pictured in Figure B.4. Four optional screw holes were kept on the rocker to accommodate the future possibility of locking bogie joint's motion entirely.

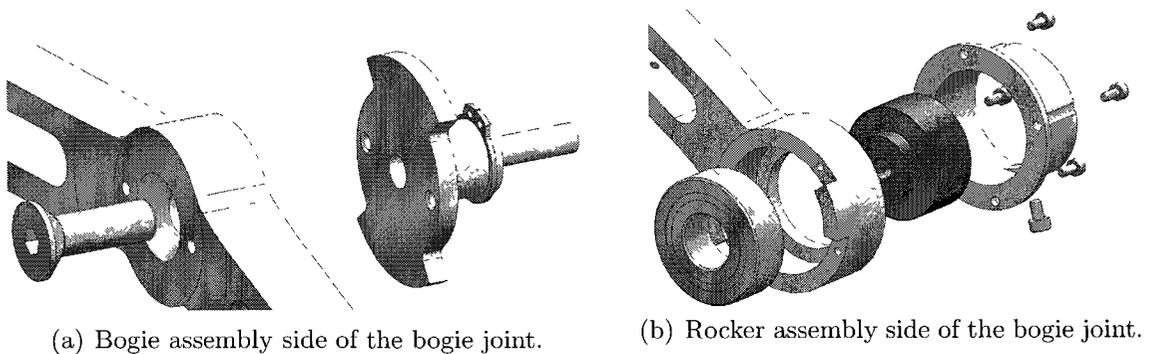


Figure B.4: Exploded views of the *Kapvik* bogie joint.

B.5 Wheel

The wheel design used on *Kapvik* is shown in Figure B.5. The wheel diameter at the base of the grousers is 145 mm. The grousers are 5 mm high and 3 mm wide, giving the wheel an outer diameter 155 mm and an average diameter of 150 mm. The wheel was designed to have low mass, and the wheel rim is just 2 mm thick. Slanting of the grousers has been shown to produce an application of forward force that is closer to continuous [12]. This is equivalent to the operation of a helical gear in which the teeth are gradually engaged versus that of a spur gear in which the teeth are abruptly engaged. The grousers were angled by 19.4° on *Kapvik*, and the direction of the helix was designed to be opposite on the left and right wheels for symmetry.

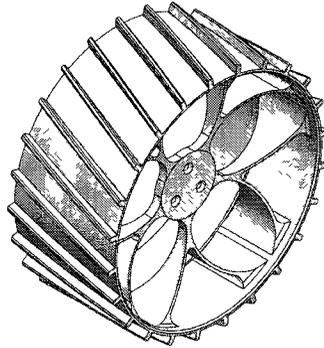


Figure B.5: The *Kapvik* wheel.

B.6 Preliminary Testing

At the time of thesis submission, the chassis has been fully assembled and undergone some initial testing, with good results. The mobility system has demonstrated that it can traverse obstacles up to 21 cm (or 1.4 wheel diameters) in height. It can successfully climb a 30° grade and has yet to be tested in a situation where its mobility is limited by motor torque. Bogie and rocker potentiometers as well as load cells have been connected and tested using amplification circuits. The outputs from these sensors are within the expected ranges. Photographs of *Kapvik* during outdoor testing, performed on Petrie Island on September 16th and 17th, are shown in Figure B.6.

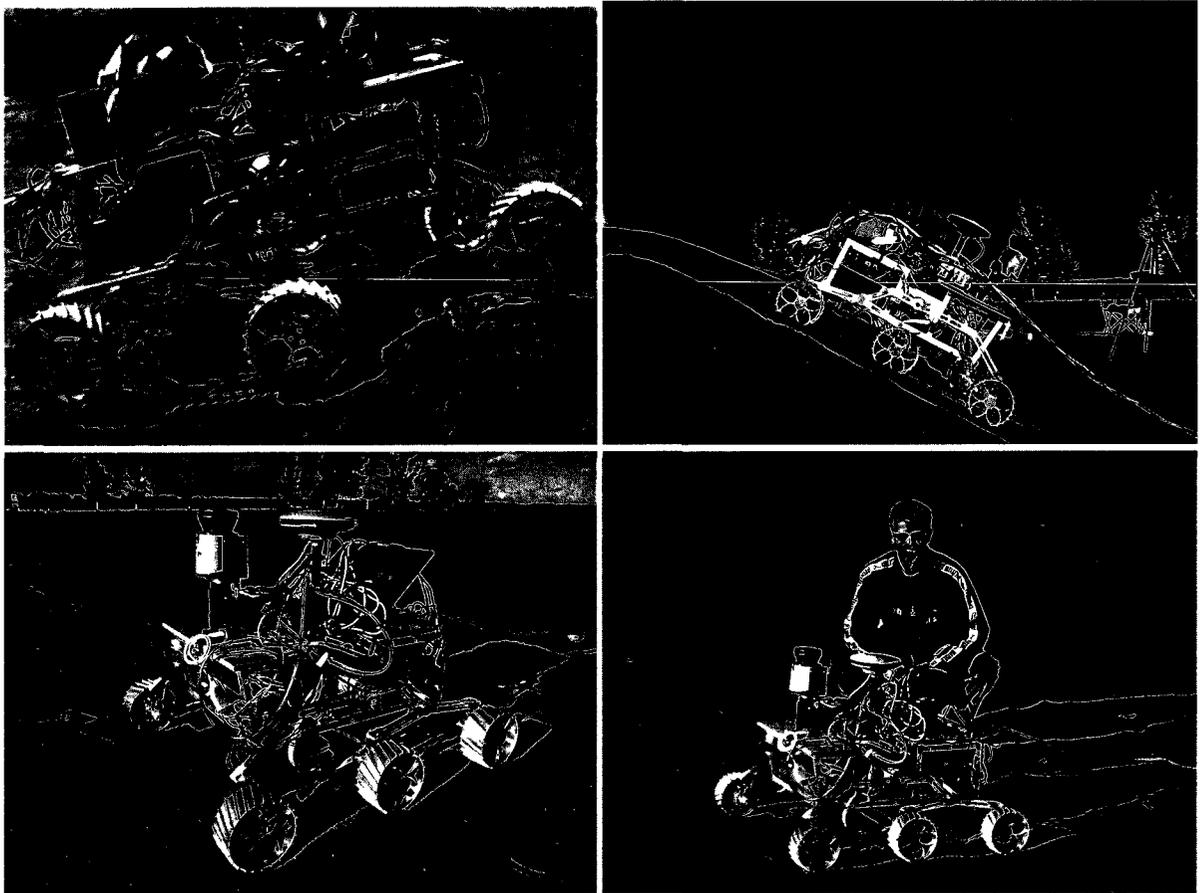


Figure B.6: Photographs of the *Kapvik* micro-rover during outdoor testing. *Kapvik* negotiates a 21 cm obstacle (top-left). *Kapvik* climbs a 30° slope (top-right). The *Kapvik* micro-rover equipped with a laser scanner, GPS, and stereoscopic cameras (bottom-left). The author with the *Kapvik* micro-rover (bottom-right).

Appendix C

Dynamic Simulation Variables

This appendix outlines the dynamic simulation variables. Certain less involved dynamic simulation variables are included as mathematical markup accompanied by MATLAB code. The more involved dynamic simulation variables are included using only the MATLAB code used to calculate their values in simulation.

C.1 Common Functions

Listing C.1 presents some MATLAB functions which are used in the subsequent evaluation of dynamic simulation variables. These functions are part of the `c` class, and are referred to in future functions as `c.function`.

Listing C.1: Common functions.

```
% Forms a rotation matrix of angle about the specified axis
function r = rot(a,axis)
% Inputs: Angle, a [rad], Axis of rotation ('x','y','z')
% Outputs: Rotation matrix to perform transformation, r 3x3
    if strcmp(axis,'x')
        r = [1 0 0; 0 cos(a) -sin(a); 0 sin(a) cos(a)];
    elseif strcmp(axis,'y')
        r = [cos(a) 0 sin(a); 0 1 0; -sin(a) 0 cos(a)];
    elseif strcmp(axis,'z')
        r = [cos(a) -sin(a) 0; sin(a) cos(a) 0; 0 0 1];
    end
end
% Forms the derivative of a rotation matrix with respect to its rotation angle
function r = drot(a,axis)
% Inputs: Angle, a [rad], Axis of rotation ('x','y','z')
% Outputs: Derivative of the rotation matrix wrt a, r 3x3
```

```

% Notes: See: http://naif.jpl.nasa.gov/pub/naif/toolkit\_docs/FORTRAN/spicelib/drotat.html
    if strcmp(axis,'x')
        r = [0 0 0; 0 -sin(a) -cos(a); 0 cos(a) -sin(a)];
    elseif strcmp(axis,'y')
        r = [-sin(a) 0 cos(a); 0 0 0; -cos(a) 0 -sin(a)];
    elseif strcmp(axis,'z')
        r = [-sin(a) -cos(a) 0; cos(a) -sin(a) 0; 0 0 0];
    end
end
% Forms the derivative of a rotation matrix with respect to time
function r = drottdt(a,adot,axis)
% Inputs: Angle, a [rad], rate of change of angle, Axis of rotation ('x','y','z')
% Outputs: Derivative of the rotation matrix wrt time, r 3x3
    r = c.drottda(a,axis)*adot; % derivative with respect to time
end
% Forms the derivative of the derivative of the rotation matrix with respect to its
rotation angle d/dt(drot/da)
function r = ddrottdadt(a,adot,axis)
% Inputs: Angle, a [rad], rate of change of angle, Axis of rotation ('x','y','z')
% Outputs: Derivative of the rotation matrix wrt time, r 3x3
    if strcmp(axis,'x')
        r = [0 0 0; 0 -cos(a) sin(a); 0 -sin(a) -cos(a)]*adot;
    elseif strcmp(axis,'y')
        r = [-cos(a) 0 -sin(a); 0 0 0; sin(a) 0 -cos(a)]*adot;
    elseif strcmp(axis,'z')
        r = [-cos(a) sin(a) 0; -sin(a) -cos(a) 0; 0 0 0]*adot;
    end
end
% Forms a skew symmetric cross product matrix a~
function ssm = skew(a)
% Inputs: Vector a = [a1 a2 a3]
% Outputs: Skew symmetric cross product matrix a~ [0 -a3 a2; a3 0 -a1; -a2 a1 0]
    ssm = [0 -a(3) a(2); a(3) 0 -a(1); -a(2) a(1) 0];
end
function s = sym(a)
% Inputs: Upper triangular square matrix, a
% Outputs: Symmetric matrix, s
s = a + a' - diag(diag(a)); % add the matrix to its transpose and subtract the extra
diagonal terms
end
% Find the clockwise angle between two vectors in two dimensions where the
dimensions are x and z
function a = ang(u,v)
    cr = cross(u,v);
    a = sign(cr(2))*acos(dot(u,v)/(norm(u)*norm(v))); % find the angle from the
inverse cos of the dot product
end
% Get the cross product in a faster way than done by MATLAB's native "cross"
function cp = cross(u,v)
% Inputs: Vectors u and v
% Outputs: Cross product u x v
    cp = c.skew(u)*v; % multiply v by the skew symmetric matrix of u
end
% Get the subarray of A from rows a to b and columns c to d
function arr = subarray(A,a,b,c,d)
% Inputs: A two dimensional array, A
% Bounds of the subarray, a,b,c,d
% Outputs: A(a:b,c:d)
    arr = A(a:b,c:d); % get the subarray
end

```

C.2 World Functions

Listing C.2 presents the MATLAB functions pertaining to the environment in which the rover is driving, such as terrain and wheel contact path functions. These functions are part of the `w` class, and are referred to in future functions as `w.function`.

Listing C.2: World functions.

```
% Find the elevation of the wheel center path at a row vector of x0 values
function e = zw(x)
    e(1,:) = x; % x0 values
    e(2,:) = 0.12*cos(2*p1/4*x - 1.6) + 0.01*x; % function for wheel center
        path elevation
end
% Find the slope of the wheel center path at a row vector of x0 values
function d = dzwdx(x)
    d(1,:) = x; % x0 values
    d(2,:) = -0.12*2*p1/4*sin(2*p1/4*x - 1.6) + 0.01; % derivative of wheel
        center path function wrt x
end
% Find the angle of the wheel center path at a row vector of x0 values
function g = gw(x)
    g(1,:) = x; % x0 values
    dzwdx = w.dzwdx(x); % get the slope at x0
    g(2,:) = atan(dzwdx(2,:)); % wheel center path angle [rad]
end
% Find the (x1,z1) coordinates of the terrain when the wheel center is at a row
vector of x0 values
function e = zt(x)
    dzwdx = w.dzwdx(x); % derivative of wheel center path wrt x0
    zw = w.zw(x); % wheel center elevations at x0
    e = [x; zw(2,:)] + r.rw*[dzwdx(2,:); -ones(1,length(x))]./[sqrt(1^2+dzwdx
        (2,:).^2); sqrt(1^2+dzwdx(2,:).^2)]; % terrain elevation (x1,z1)
end
% Find the slope of the terrain (dz1/dx1) at a row vector of x0 values
function d = dztdx(x)
    zt = w.zt(x); % use the terrain elevation function to
        find the corresponding x1 value
    dzwdx = w.dzwdx(x); % get the slope of the wheel center path at
        x0
    d(1,:) = zt(1,:); % x1 value for x0 inputs
    d(2,:) = dzwdx(2,:); % the slope of the terrain at x1 is the
        same as the slope of the terrain at x0
end
% Find the angle of the terrain at a row vector of x0 values
function g = gt(x)
    dztdx = w.dztdx(x); % get the slope at x0
    g(1,:) = dztdx(1,:); % x1 values for x0 inputs
    g(2,:) = -atan(dztdx(2,:)); % terrain angle [rad]
end
```

C.3 Link Matrix

The link matrix \mathbf{L} is a matrix of zeros and ones where $L_{i,j}$ indicates whether j is on the kinematic chain from 0 to i . In the MATLAB code, it is referred to as `r.L`.

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{C.1})$$

C.4 Joint and Link Positions

Listing C.3 presents the MATLAB code to find the world co-ordinates of the rover's joints \mathbf{p}_i and link centers of mass \mathbf{p}_{c_i} .

Listing C.3: Joint and link positions.

```
% Find the joint positions in inertial coordinates given phi
function pp = p(phi)
% Inputs: Configuration, phi [(1)x0 (2)y0 (3)z0 (4)theta0x (5)theta0y (6)theta0z (7)
        q1 (8)q2 (9)q3 (10)q4]'
% Outputs: Joint positions, pp [ [p1x p1y p1z]' [p2x p2y p2z]' [p3x p3y p3z]' [p4x
        p4y p4z]' ]
        Rq1 = c.rot(phi(7),'y');           % rotation by q1
        l01 = r.l01,                       % vector from body to link 1
        l(:,1) = l01;                      % vector to link 1 rover coords
        l(:,2) = l01 + Rq1*r.l1            % vector to link 2 rover coords
        l(:,3) = l01 + Rq1*r.l1            % vector to link 3 rover coords
        l(:,4) = r.l04,                   % vector to link 4 rover coords
        r0 = phi(1:3),                   % rover position inertial coords
        pp = [r0 r0 r0 r0] + c.rot(phi(5),'y')*l; % link positions [p1 p2 p3 p4 p5]
        inertial coords
end
% Find the link CG positions in inertial coordinates given phi
function pp = pc(phi)
% Inputs: Configuration, phi [(1)x0 (2)y0 (3)z0 (4)theta0x (5)theta0y (6)theta0z (7)
        q1 (8)q2 (9)q3 (10)q4]'
% Outputs: Link CG positions, pp [ [pc1x pc1y pc1z]' [pc2x pc2y pc2z]' [pc3x pc3y
        pc3z]' [pc4x pc4y pc4z]' ]
        p = r.p(phi),
        % vectors to links
        Rt0y = c.rot(phi(5),'y');
        Rq1 = c.rot(phi(7),'y'),
        pp(:,1) = p(:,1) + Rt0y*Rq1*r.c11; % vector to link 1 CG
        inertial coords
        pp(:,2) = p(:,2) + Rt0y*Rq1*c.rot(phi(8),'y')*r.c22; % vector to link 2 CG
        inertial coords
        pp(:,3) = p(:,3) + Rt0y*Rq1*c.rot(phi(9),'y')*r.c33, % vector to link 2 CG
        inertial coords
        pp(:,4) = p(:,4) + Rt0y*c.rot(phi(10),'y')*r.c44, % vector to link 4 CG
        inertial coords
end
```

C.5 Wheel Force Jacobian

The wheel force Jacobian \mathbf{J}_w in Equation C.2 was developed using the MATLAB Symbolic Math Toolbox. Listing C.4 presents the MATLAB function used to evaluate the wheel force Jacobian. This function is part of the `r` class, so is referred to as `r.Jw`.

$$\mathbf{J}_w = \begin{bmatrix} 1 & 0 & 0 & r_{02z} & 0 & p_{2z} - p_{1z} & 0 & 0 & 0 \\ 0 & 1 & 0 & -r_{02z} & 0 & -r_{02x} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -r_{02x} & 0 & p_{1x} - p_{2x} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & r_{03z} & 0 & p_{3z} - p_{1z} & 0 & 0 & 0 \\ 0 & 1 & 0 & -r_{03z} & 0 & -r_{03x} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -r_{03x} & 0 & p_{1x} - p_{3x} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & r_{04z} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -r_{04z} & 0 & -r_{04x} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -r_{04x} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (\text{C.2})$$

where $\mathbf{r}_{0i} = (\mathbf{p}_i - \mathbf{r}_0)$, \mathbf{r}_0 is the position of the rover lumped body/rocker center of mass, and \mathbf{p}_i is the position of joint i , all in the world frame.

Listing C.4: Wheel force Jacobian.

```
% Find the wheel jacobian given phi
function j = Jw(phi)
% Inputs: Configuration, phi [(1)x0 (2)y0 (3)z0 (4)theta0x (5)theta0y (6)theta0z
(7)q1 (8)q2 (9)q3 (10)q4]'
% Outputs: Sensitivity of the system to forces and moments applied at the wheel
centroids, j [Jw1; Jw2; Jw3] where J1 = [JBT JMT; JBR JMR]
% Notes: This jacobian is used in the following expression Fsys = Jw'*[Fw1 Nw1 Fw2
Nw2 Fw3 Nw3]
    JMT = zeros(3,r.nv); % jacobian dv/dqdot
    JMR = zeros(3,r.nv); % jacobian dom/dqdot
    J = zeros(6,6+r.nv,r.nv); % whole jacobian for each joint
    p = r.p(phi); % position of the links in
        inertial coords
    r0 = phi(1:3); % position of the rover in
        inertial coords
    Y = [0 1 0]; % out of the plane joint vector
    L = r.L - eye(r.nv); % since the force is being
        applied at the wheel centroid and not the wheel CG, take the joint off
        the kinematic chain of its own link
    JBR = [zeros(3) eye(3)]; % create jacobian dom/d(vc,omc)
    for i = 1:r.nw % for each wheel
        JBT = [eye(3) c.skew(p(:,i+1)-r0)']; % create jacobian dv
            /d(vc,omc)
        for j = 1:r.nv % for each link
            JMT(:,j) = L(1+1,j)*c.cross(Y,p(:,i+1)-p(:,j)); % create a column
                of jacobian dv/dqdot
            JMR(:,j) = r.L(1+1,j)*Y; % create a column of
                jacobian dom/dqdot (here we use the original r.L since the
                torque applied still affects the rotation)
        end
    end
```

```

        J(:,:,1) = [JBT JMT; JBR JMR];           % combine into the whole
            jacobian for the wheel
    end
    j = [J(:,:,1); J(:,:,2); J(:,:,3)];         % stack the jacobians to form
        Jw
end

```

C.6 Link Jacobian

Listing C.5 presents the MATLAB function used to evaluate the velocity Jacobian for the link centers of mass \mathbf{J}_a . This function is part of the `r` class, so is referred to as `r.Ja`.

Listing C.5: Link Jacobian.

```

% Find the link jacobians given phi
function j = Ja(phi)
% Inputs: Configuration, phi [(1)x0 (2)y0 (3)z0 (4)theta0x (5)theta0y (6)theta0z
        (7)q1 (8)q2 (9)q3 (10)q4]'
% Outputs: Jacobian of link centroids, j [J1; J2; J3] where J1 = [JBT JMT; JBR JMR]
% Notes: This jacobian is used in the following expression Xadot = Ja*phidot.
    JMT = zeros(3,r.nv);           % jacobian dv/dqdot
    JMR = zeros(3,r.nv);           % jacobian dom/dqdot
    J = zeros(6,6+r.nv,r.nv);      % whole jacobian for each joint
    pc = r.pc(phi);                % position of the endpoints in
        inertial coords
    p = r.p(phi);                  % position of the links in inertial
        coords
    r0 = phi(1:3);                 % position of the rover in inertial
        coords
    Y = [0 1 0];                  % out of the plane joint vector
    JBR = [zeros(3) eye(3)];       % create jacobian dom/d(vc,omc)
    for i = 1:r.nv                 % for each link
        JBT = [eye(3) c.skew(pc(:,1)-r0)']; % create jacobian dv/d(vc,omc)
        for j = 1:r.nv             % for each link
            JMT(:,j) = r.L(1,j)*c.cross(Y,pc(:,1)-p(:,j)); % create a column
                of jacobian dv/dqdot
            JMR(:,j) = r.L(1,j)*Y; % create a column of jacobian
                dom/dqdot
        end
        J(:,:,1) = [JBT JMT; JBR JMR]; % combine into the whole
            jacobian for the system
    end
    j = [J(:,:,1); J(:,:,2); J(:,:,3); J(:,:,4)]; % stack the jacobians to form
        Ja
end

```

C.7 System Inertia Matrix

The system inertia matrix \mathbf{H} , developed in the MATLAB Symbolic Math Toolbox, is given row by row in Equation C.3. Listing C.6 presents the MATLAB function used to evaluate the system inertia matrix \mathbf{H} . This function is part of the `r` class, so is referred to as `r.H`.

$$\mathbf{H} = [\mathbf{H}_1 \mathbf{H}_2 \mathbf{H}_3 \mathbf{H}_4 \mathbf{H}_5 \mathbf{H}_6 \mathbf{H}_7 \mathbf{H}_8 \mathbf{H}_9 \mathbf{H}_{10}]^T$$

$$\mathbf{H}_1^T = \begin{bmatrix} m_0+m_1+m_2+m_3+m_4 \\ 0 \\ 0 \\ 0 \\ m_1r_{0c1z}+m_2r_{0c2z}+m_3r_{0c3z}+m_4r_{0c4z} \\ 0 \\ -m_1(p_{1z}-p_{c1z})-m_2(p_{1z}-p_{c2z})-m_3(p_{1z}-p_{c3z}) \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{H}_2^T = \begin{bmatrix} 0 \\ m_0+m_1+m_2+m_3+m_4 \\ 0 \\ -m_1r_{0c1z}-m_2r_{0c2z}-m_3r_{0c3z}-m_4r_{0c4z} \\ -m_1r_{0c1x}-m_2r_{0c2x}-m_3r_{0c3x}-m_4r_{0c4x} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{H}_3^T = \begin{bmatrix} 0 \\ 0 \\ m_0+m_1+m_2+m_3+m_4 \\ 0 \\ -m_1r_{0c1x}-m_2r_{0c2x}-m_3r_{0c3x}-m_4r_{0c4x} \\ 0 \\ m_1(p_{1x}-p_{c1x})+m_2(p_{1x}-p_{c2x})+m_3(p_{1x}-p_{c3x}) \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{H}_4^T = \begin{bmatrix} 0 \\ -m_1r_{0c1z}-m_2r_{0c2z}-m_3r_{0c3z}-m_4r_{0c4z} \\ 0 \\ m_1r_{0c1z}^2+m_2r_{0c2z}^2+m_3r_{0c3z}^2+m_4r_{0c4z}^2 \\ 0 \\ m_1r_{0c1x}r_{0c1z}+m_2r_{0c2x}r_{0c2z}+m_3r_{0c3x}r_{0c3z}+m_4r_{0c4x}r_{0c4z} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{aligned}
\mathbf{H}_5^T &= \begin{bmatrix} m_1 r_{0c1z} + m_2 r_{0c2z} + m_3 r_{0c3z} + m_4 r_{0c4z} \\ 0 \\ -m_1 r_{0c1x} - m_2 r_{0c2x} - m_3 r_{0c3x} - m_4 r_{0c4x} \\ 0 \\ m_1 r_{0c1x}^2 + m_2 r_{0c2x}^2 + m_1 r_{0c1z}^2 + m_3 r_{0c3x}^2 + m_2 r_{0c2z}^2 + m_4 r_{0c4x}^2 \\ + m_3 r_{0c3z}^2 + m_4 r_{0c4z}^2 + I_{02,2} + I_{12,2} + 3I_w \\ 0 \\ I_{12,2} + 2I_w - m_1 r_{0c1x}(p_{1x} - p_{c1x}) - m_1 r_{0c1z}(p_{1z} - p_{c1z}) - m_2 r_{0c2x}(p_{1x} - p_{c2x}) \\ - m_2 r_{0c2z}(p_{1z} - p_{c2z}) - m_3 r_{0c3x}(p_{1x} - p_{c3x}) - m_3 r_{0c3z}(p_{1z} - p_{c3z}) \\ I_w \\ I_w \\ I_w \end{bmatrix} \\
\mathbf{H}_6^T &= \begin{bmatrix} 0 \\ -m_1 r_{0c1x} - m_2 r_{0c2x} - m_3 r_{0c3x} - m_4 r_{0c4x} \\ 0 \\ m_1 r_{0c1x} r_{0c1z} + m_2 r_{0c2x} r_{0c2z} + m_3 r_{0c3x} r_{0c3z} + m_4 r_{0c4x} r_{0c4z} \\ 0 \\ m_1 r_{0c1x}^2 + m_2 r_{0c2x}^2 + m_3 r_{0c3x}^2 + m_4 r_{0c4x}^2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\
\mathbf{H}_7^T &= \begin{bmatrix} -m_1(p_{1z} - p_{c1z}) - m_2(p_{1z} - p_{c2z}) - m_3(p_{1z} - p_{c3z}) \\ 0 \\ m_1(p_{1x} - p_{c1x}) + m_2(p_{1x} - p_{c2x}) + m_3(p_{1x} - p_{c3x}) \\ 0 \\ I_{12,2} + 2I_w - m_1 r_{0c1x}(p_{1x} - p_{c1x}) - m_1 r_{0c1z}(p_{1z} - p_{c1z}) - m_2 r_{0c2x}(p_{1x} - p_{c2x}) \\ - m_2 r_{0c2z}(p_{1z} - p_{c2z}) - m_3 r_{0c3x}(p_{1x} - p_{c3x}) - m_3 r_{0c3z}(p_{1z} - p_{c3z}) \\ 0 \\ I_{12,2} + 2I_w + m_1(p_{1x} - p_{c1x})^2 + m_1(p_{1z} - p_{c1z})^2 \\ + m_2(p_{1x} - p_{c2x})^2 + m_2(p_{1z} - p_{c2z})^2 + m_3(p_{1x} - p_{c3x})^2 + m_3(p_{1z} - p_{c3z})^2 \\ I_w \\ I_w \\ 0 \end{bmatrix} \\
\mathbf{H}_8 &= [0 \ 0 \ 0 \ 0 \ I_w \ 0 \ I_w \ I_{wg} \ 0 \ 0] \\
\mathbf{H}_9 &= [0 \ 0 \ 0 \ 0 \ I_w \ 0 \ I_w \ 0 \ I_{wg} \ 0] \\
\mathbf{H}_{10} &= [0 \ 0 \ 0 \ 0 \ I_w \ 0 \ 0 \ 0 \ 0 \ I_{wg}] \tag{C.3}
\end{aligned}$$

where $\mathbf{r}_{0c_i} = (\mathbf{p}_{c_i} - \mathbf{r}_0)$, \mathbf{r}_0 is the position of the rover lumped body/rocker center of mass, \mathbf{p}_{c_i} is the position of the center of mass of joint i , and $I_{i2,2}$ represents the inertia of body i in the Y_w axis.

Listing C.6: System inertia matrix.

```

% Find the system inertia matrix given phi
function h = H(phi)
% Inputs: Configuration, phi [(1)x0 (2)y0 (3)z0 (4)theta0x (5)theta0y (6)theta0z
(7)q1 (8)q2 (9)q3 (10)q4]'
% Outputs: System inertia matrix H (6+nv)x(6+nv)
% Notes: The inertia matrix is used in the expression for kinetic energy. T =
1/2*phidot' * H * phidot
h = r.H0; % body inertia matrix
Ja = r.Ja(phi); % get the link jacobian matrix
for i = 1:r.nv % for all of the links, add the jacobian of the link
to the system inertia matrix
h = h + r.ml(i)*Ja(1+6*(i-1):3+6*(i-1),1:6+r.nv)*Ja(1+6*(i-1):3+6*(i-1),1:6+r.nv) + Ja(4+6*(i-1):6+6*(i-1),1:6+r.nv)*r.Il(1:3,1:3,i)
*Ja(4+6*(i-1):6+6*(i-1),1:6+r.nv);

```

```

end
h(end-r.nw+1:end,end-r.nw+1:end) = diag([r.Iwg r.Iwg r.Iwg]); % different
    inertias for wheel rotation
end

```

C.8 Time Derivative of the System Inertia Matrix

Listing C.7 presents the MATLAB function used to evaluate the time derivative of system inertia matrix \dot{H} . This function is part of the `r` class, so is referred to as `r.Hdot`.

Listing C.7: Time derivative of the system inertia matrix.

```

% Get the time derivative of inertia matrix H
function dhdt = Hdot (phi,phidot)
% Inputs: Configuration, phi [(1)x0 (2)y0 (3)z0 (4)theta0x (5)theta0y (6)theta0z
    (7)q1 (8)q2 (9)q3 (10)q4]'
% Time rate of change of configuration, phidot [(1)v0x (2)v0y (3)v0z (4)
    thetadot0x (5)thetadot0y (6)thetadot0z (7)qdot1 (8)qdot2 (9)qdot3 (10)qdot4]'
% Outputs: Time derivative of system inertia matrix H, Hdot (6+nv)x(6+nv)
% Notes: The time derivative of the inertia matrix is used in the Euler-Lagrange
    equation: H*qddot + Hdot*qdot - dL/dq = tau + tauC
    Y = [0 1 0]'; % y unit vector
    Rt0y = c.rot(phi(5),'y') % theta0y rotation matrix
    Rq1 = c.rot(phi(7),'y') % joint angle q1 rotation matrix
    Rq2 = c.rot(phi(8),'y') % joint angle q2 rotation matrix
    Rq3 = c.rot(phi(9),'y') % joint angle q3 rotation matrix
    Rq4 = c.rot(phi(10),'y') % joint angle q4 rotation matrix
    dRt0ydt = c.drottdt(phi(5),phidot(5),'y'); % d/dt(Rtheta0y)
    dRq1dt = c.drottdt(phi(7),phidot(7),'y'); % d/dt(Rq1)
    dRq2dt = c.drottdt(phi(8),phidot(8),'y'); % d/dt(Rq2)
    dRq3dt = c.drottdt(phi(9),phidot(9),'y'); % d/dt(Rq3)
    dRq4dt = c.drottdt(phi(10),phidot(10),'y'); % d/dt(Rq4)
    r0 = phi(1:3); % rover body position
    p = r.p(phi); % joint positions inertial coords
    pc = r.pc(phi);
    r01 = pc - [r0 r0 r0 r0];
    r11 = pc - [p(:,1) p(:,1) p(:,1) p(:,1)];
    r21 = pc - [p(:,2) p(:,2) p(:,2) p(:,2)];
    r31 = pc - [p(:,3) p(:,3) p(:,3) p(:,3)];
    r41 = pc - [p(:,4) p(:,4) p(:,4) p(:,4)];

    Jv0 = [eye(3) zeros(3,r.nv+3)]; % velocity Jacobian for body 0
    dh0dtV = zeros(6+r.nv,6+r.nv); % d/dt(Jv0'*Jv0) = d/dt(Jv0')*Jv0 + Jv0'*d/dt(
        Jv0)

    Jv1 = [eye(3) c.skew(r01(:,1))' c.cross(Y,r11(:,1)) zeros(3,3)]; %
        velocity Jacobian for link 1
    dJv1dt(1:3,1:3) = zeros(3,3); % d(eye(3))/dt
    dJv1dt(1:3,4:6) = c.skew( dRt0ydt*(r.l01+Rq1*r.c11) + Rt0y*dRq1dt*r.c11 )'; %
        d/dt(r^-)
    dJv1dt(1:3,7) = c.cross(Y, dRt0ydt*Rq1*r.c11 + Rt0y*dRq1dt*r.c11 ); % d/dt(Y
        x (pc1-p1))
    dJv1dt(1:3,8:6+r.nv) = zeros(3,3); % links 2, 3 and 4 are not on the
        kinematic chain of 1

```

```

dh1dtV = r.ml(1) * ( dJv1dt' * Jv1 + Jv1' * dJv1dt ); % chain rule
differentiation d/dt(Jv1'*Jv1) = d/dt(Jv1')*Jv1 + Jv1'*d/dt(Jv1)

Jv2 = [eye(3)   c.skew(r01(:,2))'   c.cross(Y,r11(:,2))   c.cross(Y,r21(:,2))
        zeros(3,2)]; % velocity Jacobian for link 2
dJv2dt(1:3,1:3) = zeros(3,3); % d(eye(3))/dt
dJv2dt(1:3,4:6) = c.skew( dRt0ydt*(r.101+Rq1*(r.112+Rq2*r.c22)) + Rt0y*dRq1dt
        *(r.112+Rq2*r.c22) + Rt0y*Rq1*dRq2dt*r.c22 ); % skew(d/dt(r'))
dJv2dt(1:3,7) = c.cross(Y, dRt0ydt*Rq1*(r.112+Rq2*r.c22) + Rt0y*dRq1dt*(r.112
        +Rq2*r.c22) + Rt0y*Rq1*dRq2dt*r.c22 ); % d/dt(Y x (pc2-p1))
dJv2dt(1:3,8) = c.cross(Y, dRt0ydt*Rq1*Rq2*r.c22 + Rt0y*dRq1dt*Rq2*r.c22 +
        Rt0y*Rq1*dRq2dt*r.c22 ); % d/dt(Y x (pc2-p2))
dJv2dt(1:3,9:6+r.nv) = zeros(3,2); % links 3 and 4 are not on the kinematic
chain of 2
dh2dtV = r.ml(2) * ( dJv2dt' * Jv2 + Jv2' * dJv2dt ); % chain rule
differentiation d/dt(Jv2'*Jv2) = d/dt(Jv2')*Jv2 + Jv2'*d/dt(Jv2)

Jv3 = [eye(3)   c.skew(r01(:,3))'   c.cross(Y,r11(:,3))   zeros(3,1)   c.
        cross(Y,r31(:,3))   zeros(3,1)]; % velocity Jacobian for link 3
dJv3dt(1:3,1:3) = zeros(3,3); % d(eye(3))/dt
dJv3dt(1:3,4:6) = c.skew( dRt0ydt*(r.101+Rq1*(r.113+Rq3*r.c33)) + Rt0y*dRq1dt
        *(r.113+Rq3*r.c33) + Rt0y*Rq1*dRq3dt*r.c33 ); % skew(d/dt(r'))
dJv3dt(1:3,7) = c.cross(Y, dRt0ydt*Rq1*(r.113+Rq3*r.c33) + Rt0y*dRq1dt*(r.113
        +Rq3*r.c33) + Rt0y*Rq1*dRq3dt*r.c33 ); % d/dt(Y x (pc3-p1))
dJv3dt(1:3,8) = zeros(3,1); % link 2 is not on the kinematic chain of 3
dJv3dt(1:3,9) = c.cross(Y, dRt0ydt*Rq1*Rq3*r.c33 + Rt0y*dRq1dt*Rq3*r.c33 +
        Rt0y*Rq1*dRq3dt*r.c33 ); % d/dt(Y x (pc3-p3))
dJv3dt(1:3,10) = zeros(3,1); % link 4 is not on the kinematic chain of 3
dh3dtV = r.ml(3) * ( dJv3dt' * Jv3 + Jv3' * dJv3dt ); % chain rule
differentiation d/dt(Jv3'*Jv3) = d/dt(Jv3')*Jv3 + Jv3'*d/dt(Jv3)

Jv4 = [eye(3)   c.skew(r01(:,4))'   zeros(3,3)   c.cross(Y,r41(:,4))]; %
velocity Jacobian for link 4
dJv4dt(1:3,1:3) = zeros(3,3); % d(eye(3))/dt
dJv4dt(1:3,4:6) = c.skew( dRt0ydt*(r.104+Rq4*r.c44) + Rt0y*dRq4dt*r.c44 ); %
d/dt(r')
dJv4dt(1:3,7:9) = zeros(3,3); % links 1, 2 and 3 are not on the kinematic
chain of 4
dJv4dt(1:3,10) = c.cross(Y, dRt0ydt*Rq4*r.c44 + Rt0y*dRq4dt*r.c44 ); % d/dt(Y
x (pc4-p4))
dh4dtV = r.ml(4) * ( dJv4dt' * Jv4 + Jv4' * dJv4dt ); % chain rule
differentiation d/dt(Jv4'*Jv4) = d/dt(Jv4')*Jv4 + Jv4'*d/dt(Jv4)

Jom0 = [ zeros(3)   eye(3)   zeros(3,4) ]; % angular velocity Jacobian of link
2
dh0dt0m = zeros(6+r.nv,6+r.nv); % none of the terms in Jom0 are time
dependent
Jom1 = [ zeros(3)   eye(3)   Y   zeros(3,3)]; % angular velocity Jacobian of
link 1
dh1dt0m = zeros(6+r.nv,6+r.nv); % none of the terms in Jom1 are time
dependent; the normal Y is time independent because we are in 2D
Jom2 = [ zeros(3)   eye(3)   Y   Y   zeros(3,2)]; % angular velocity Jacobian
of link 2
dh2dt0m = zeros(6+r.nv,6+r.nv); % none of the terms in Jom2 are time
dependent; the normal Y is time independent because we are in 2D
Jom3 = [ zeros(3)   eye(3)   Y   zeros(3,1)   Y zeros(3,1)]; % angular
velocity Jacobian of link 3
dh3dt0m = zeros(6+r.nv,6+r.nv); % none of the terms in Jom3 are time
dependent; the normal Y is time independent because we are in 2D
Jom4 = [ zeros(3)   eye(3)   zeros(3,3) Y ]; % angular velocity Jacobian of
link 4
dh4dt0m = zeros(6+r.nv,6+r.nv); % none of the terms in Jom4 are time
dependent; the normal Y is time independent because we are in 2D

```

```

dhdt = dh0dtV + dh1dtV + dh2dtV + dh3dtV + dh4dtV + dh0dt0m + dh1dt0m +
dh2dt0m + dh3dt0m + dh4dt0m; % the sum of the derivatives is the
derivative of H with respect to time
end

```

C.9 Rover Constraints

Listing C.8 presents the MATLAB function used to evaluate the rover constraints Ψ .

This function is part of the `r` class, so is referred to as `r.Psi`.

Listing C.8: Rover constraints.

```

% Get the constraints on the rover
function ps = Psi(phi)
% Inputs: Configuration, phi [(1)x0 (2)y0 (3)z0 (4)theta0x (5)theta0y (6)theta0z (7)
q1 (8)q2 (9)q3 (10)q4]'
% Outputs: Constraints to keep the wheel centers on the wheel center path, Psi [(1)
p3z-zw(p3x) (2)p4z-zw(p4x) (3)p5z-zw(p5x)]'
p = r.p(phi); % joint positions inertial coords
zw = w.zw(p(1,2:4)); % wheel center z coordinates
ps = p(3,2:4)' - zw(2,:)'; % constraint vector Psi (should be = 0)
end

```

C.10 Jacobian of Constraints with respect to Configuration

Listing C.9 presents the MATLAB function used to evaluate the Jacobian of constraints with respect to configuration Ψ_{Φ} . This function is part of the `r` class, so is referred to as `r.Psphi`.

Listing C.9: Jacobian of constraints with respect to configuration.

```

% Get the Jacobian of rover constrains wrt phi
function psp = Psphi(phi)
% Inputs: Configuration, phi [(1)x0 (2)y0 (3)z0 (4)theta0x (5)theta0y (6)theta0z
(7)q1 (8)q2 (9)q3 (10)q4]'
% Outputs: Jacobian of constraints wrt phi, psp (m x (3+nv)) where m is the number
of constraints
% Notes: This function finds the Jacobian of the constraints below. These
constraints assure that the z position of the wheel centers is on the
% wheel center path, as desired. By default gte is not required, but if
calling from an estimator it must be provided because the function
% dzwdx is unavailable.

```

```

%      Joint 3 Constraint: Psi3 = z0 + [Rt0y*(l01+Rq1*l12)]_z - w.zw(x0+[Rt0y*(
101+Rq1*l12)]_x)
%      Joint 4 Constraint: Psi4 = z0 + [Rt0y*(l01+Rq1*l13)]_z - w.zw(x0+[Rt0y*(
101+Rq1*l13)]_x)
%      Joint 5 Constraint: Psi5 = z0 + [Rt0y*l04]_z - w.zw(x0+[Rt0y*l04]_x)
p = r.p(phi); % joint positions inertial
               coords
Rt0y = c.rot(phi(5),'y'); % theta0y rotation matrix
Rq1 = c.rot(phi(7),'y'); % joint angle q1 rotation matrix
dRt0y = c.drot da(phi(5),'y'); % derivative of rotation matrix theta0y wrt
theta0y
dRq1 = c.drot da(phi(7),'y'); % derivative of rotation matrix q1 wrt q1
dzwdx = w.dzwdx([p(1,2) p(1,3) p(1,4)]'); % the slope of the
wheel path at the wheel positions
dPsidx0 = [-dzwdx(2,1) -dzwdx(2,2) -dzwdx(2,3) ]'; % dPsi/dx0 = -(dzw/dx
)*(dx/d0) = -dzw/dx*1
dPsidy0 = [0 0 0]'; % dPsi/dy0 = 0
dPsidz0 = [1 1 1]'; % dPsi/dz0 = 1
dPsidtheta0x = [0 0 0]'; % dPsi/dtheta0x = 0
dPsidtheta0y(1,1) = dRt0y(3,:)*(r.l01+Rq1*r.l12) - dzwdx(2,1)*dRt0y(1,:)*(r.
101+Rq1*r.l12); % dPsi/dtheta0y = [dRt0y/dt0y*pr]_z - (dzw/dx)*(dx/dt0y)
dPsidtheta0y(2,1) = dRt0y(3,:)*(r.l01+Rq1*r.l13) - dzwdx(2,2)*dRt0y(1,:)*(r.
101+Rq1*r.l13);
dPsidtheta0y(3,1) = dRt0y(3,:)*r.l04 - dzwdx(2,3)*dRt0y(1,:)*r.l04;
dPsidtheta0z = [0 0 0]'; %
dPsi/dtheta0z = 0
dPsidq1(1,1) = Rt0y(3,:)*dRq1*r.l12 - dzwdx(2,1)*Rt0y(1,:)*dRq1*r.l12; %
dPsi1/dq1 = [Rt0y*(dRq1/q1)*l12]_z - (dzw/dx)*(dx/dq1)
dPsidq1(2,1) = Rt0y(3,:)*dRq1*r.l13 - dzwdx(2,2)*Rt0y(1,:)*dRq1*r.l13; %
dPsi2/dq1 = [Rt0y*(dRq1/q1)*l13]_z - (dzw/dx)*(dx/dq1)
dPsidq1(3,1) = 0; % dPsi3/dq1 = 0
dPsidq2 = [0 0 0]'; % dPsi/dq2 = 0
dPsidq3 = [0 0 0]'; % dPsi/dq3 = 0
dPsidq4 = [0 0 0]'; % dPsi/dq4 = 0
psp = [dPsidx0 dPsidy0 dPsidz0 dPsidtheta0x dPsidtheta0y dPsidtheta0z dPsidq1
dPsidq2 dPsidq3 dPsidq4]; % concatenate sensitivities to form Jacobian
end

```

C.11 Time Derivative of the Jacobian of Constraints with respect to Configuration

Listing C.10 presents the MATLAB function used to evaluate the time derivative of the Jacobian of constraints with respect to configuration $\dot{\Psi}_{\Phi}$. This function is part of the `r` class, so is referred to as `r.Psiphidot`.

Listing C.10: Time derivative of the Jacobian of constraints with respect to configuration.

```

% Get the time derivative Jacobian of rover constrains wrt phi
function dpspdt = Psiphidot(phi,phidot)
% Inputs: Configuration, phi [(1)x0 (2)y0 (3)z0 (4)theta0x (5)theta0y (6)theta0z
(7)q1 (8)q2 (9)q3 (10)q4]

```

```

%           Time rate of change of configuration, phidot [(1)v0x (2)v0y (3)v0z (4)
           thetadot0x (5)thetadot0y (6)thetadot0z (7)qdot1 (8)qdot2 (9)qdot3 (10)qdot4]'
% Outputs:  Time derivative of Jacobian of constraints wrt phi, dpspdt (m x (3+nv))
           where m is the number of constraints
% Notes:    This function finds the time derivative of the constraint Jacobian found
           in Ps1phi
           p = r.p(phi); % joint positions inertial coords
           Rt0y = c.rot(phi(5),'y'); % theta0y rotation matrix
           Rq1 = c.rot(phi(7),'y'); % joint angle q1 rotation matrix
           dRt0y = c.drotdda(phi(5),'y'); % derivative of rotation matrix theta0y wrt
           theta0y
           dRq1 = c.drotdda(phi(7),'y'); % derivative of rotation matrix q1 wrt q1
           dRt0ydt = c.drotddt(phi(5),phidot(5),'y'); % d/dt(Rtheta0y)
           dRq1dt = c.drotddt(phi(7),phidot(7),'y'); % d/dt(Rq1)
           ddRt0ydt = c.ddrotddadt(phi(5),phidot(5),'y'); % d/dt(dRtheta0y/dtheta0y)
           ddRq1dt = c.ddrotddadt(phi(7),phidot(7),'y'); % d/dt(dRq1/dq1)
           dzwdx = w.dzwdx([p(1,2) p(1,3) p(1,4)]'); % d/dx(zw) slope of wheel
           centroid path
           ddPs1dx0dt = [0 0 0]'; % d/dt(dPs1/dx0)
           ddPs1dy0dt = [0 0 0]'; % d/dt(dPs1/dy0)
           ddPs1dz0dt = [0 0 0]'; % d/dt(dPs1/dz0)
           ddPs1dt0xdt = [0 0 0]'; % d/dt(dPs1/dtheta0x)
           ddPs1dt0ydt(1,1) = ddRt0ydt(3,:)*(r.l01+Rq1*r.l12) + dRt0y(3,:)*dRq1dt*r.l12
           - dzwdx(2,1)*(ddRt0ydt(1,:)*(r.l01+Rq1*r.l12) + dRt0y(1,:)*dRq1dt*r.l12);
           % d/dt(dPs1/dtheta0y)
           ddPs1dt0ydt(2,1) = ddRt0ydt(3,:)*(r.l01+Rq1*r.l13) + dRt0y(3,:)*dRq1dt*r.l13
           - dzwdx(2,2)*(ddRt0ydt(1,:)*(r.l01+Rq1*r.l13) + dRt0y(1,:)*dRq1dt*r.l13);
           ddPs1dt0ydt(3,1) = ddRt0ydt(3,:)*r.l04 - dzwdx(2,3)*ddRt0ydt(1,:)*r.l04;
           ddPs1dt0zdt = [0 0 0]'; % d/dt(
           dPs1/dtheta0z)
           ddPs1dq1dt(1,1) = dRt0ydt(3,:)*dRq1*r.l12 + Rt0y(3,:)*ddRq1dt*r.l12 - dzwdx
           (2,1)*(dRt0ydt(1,:)*dRq1*r.l12 + Rt0y(1,:)*ddRq1dt*r.l12); % d/dt(dPs1/
           dq1)
           ddPs1dq1dt(2,1) = dRt0ydt(3,:)*dRq1*r.l13 + Rt0y(3,:)*ddRq1dt*r.l13 - dzwdx
           (2,2)*(dRt0ydt(1,:)*dRq1*r.l13 + Rt0y(1,:)*ddRq1dt*r.l13); % d/dt(dPs1/
           dq1)
           ddPs1dq1dt(3,1) = 0; % d/dt(dPs13/dq1)
           ddPs1dq2dt = [0 0 0]'; % d/dt(dPs1/dq2)
           ddPs1dq3dt = [0 0 0]'; % d/dt(dPs1/dq3)
           ddPs1dq4dt = [0 0 0]'; % d/dt(dPs1/dq4)
           dpspdt = [ddPs1dx0dt ddPs1dy0dt ddPs1dz0dt ddPs1dt0xdt ddPs1dt0ydt
           ddPs1dt0zdt ddPs1dq1dt ddPs1dq2dt ddPs1dq3dt ddPs1dq4dt]; % concatenate
           sensitivities to form Jacobian
end

```

C.12 Derivative of Kinetic Energy with respect to Configuration

Listing C.11 presents the MATLAB function used to evaluate the derivative of kinetic energy with respect to configuration $\frac{\partial T}{\partial \Phi}$. This function is part of the `r` class, so is referred to as `r.dTdphi`.

Listing C.11: Derivative of kinetic energy with respect to configuration.

```

% Get the derivative of kinetic energy T with respect to phi
function dtdp = dTdphi (phi,phidot)
% Inputs: Configuration, phi [(1)x0 (2)y0 (3)z0 (4)theta0x (5)theta0y (6)theta0z
(7)q1 (8)q2 (9)q3 (10)q4]'
% Time rate of change of configuration, phidot [(1)v0x (2)v0y (3)v0z (4)
thetadot0x (5)thetadot0y (6)thetadot0z (7)qdot1 (8)qdot2 (9)qdot3 (10)qdot4]'
% Outputs: Derivative of kinetic energy, T wrt phi (6+nv)x1
% Notes: Finds the derivative of the kinetic energy T (which is a scalar) with
respect to each variable in phi, returning a column matrix. The
% approach used is the following: T = 1/2*phidot'*H*phidot. dT/dphi(1) =
1/2*phidot'*dH/dphi(1)*phidot
Y = [0 1 0]'; % y unit vector
Rt0y = c.rot(phi(5),'y'); % theta0y rotation matrix
Rq1 = c.rot(phi(7),'y'); % joint angle q1 rotation matrix
Rq2 = c.rot(phi(8),'y'); % joint angle q2 rotation matrix
Rq3 = c.rot(phi(9),'y'); % joint angle q3 rotation matrix
Rq4 = c.rot(phi(10),'y'); % joint angle q4 rotation matrix
dRt0y = c.drotdda(phi(5),'y'); % derivative of rotation matrix theta0y wrt
theta0y
dRq1 = c.drotdda(phi(7),'y'); % derivative of rotation matrix q1 wrt q1
dRq2 = c.drotdda(phi(8),'y'); % derivative of rotation matrix q2 wrt q2
dRq3 = c.drotdda(phi(9),'y'); % derivative of rotation matrix q3 wrt q3
dRq4 = c.drotdda(phi(10),'y'); % derivative of rotation matrix q4 wrt q4
pc = r.pc(phi); % link CG positions
p = r.p(phi); % joint positions inertial coords
r0 = phi(1:3); % rover body position
r01 = pc - [r0 r0 r0 r0];
r11 = pc - [p(:,1) p(:,1) p(:,1) p(:,1)];
r21 = pc - [p(:,2) p(:,2) p(:,2) p(:,2)];
r31 = pc - [p(:,3) p(:,3) p(:,3) p(:,3)];
r41 = pc - [p(:,4) p(:,4) p(:,4) p(:,4)];

% Find the derivative of the angular velocity Jacobians with respect to phi
where dJom1dphi(:,j) dJom1/dphi(j)
Jom0 = [ zeros(3) eye(3) zeros(3,4) ]; % angular velocity Jacobian of link
2
dJom0dphi(1:3,1:6+r.nv,1:6+r.nv) = zeros(3,6+r.nv,6+r.nv); % d/dphi(Jom0)
Jom1 = [ zeros(3) eye(3) Y zeros(3,3) ]; % angular velocity Jacobian of
link 1
dJom1dphi(1:3,1:6+r.nv,1:6+r.nv) = zeros(3,6+r.nv,6+r.nv); % d/dphi(Jom1)
Jom2 = [ zeros(3) eye(3) Y Y zeros(3,2) ]; % angular velocity Jacobian
of link 2
dJom2dphi(1:3,1:6+r.nv,1:6+r.nv) = zeros(3,6+r.nv,6+r.nv); % d/dphi(Jom2)
Jom3 = [ zeros(3) eye(3) Y zeros(3,1) Y zeros(3,1) ]; % angular velocity
Jacobian of link 3
dJom3dphi(1:3,1:6+r.nv,1:6+r.nv) = zeros(3,6+r.nv,6+r.nv); % d/dphi(Jom3)
Jom4 = [ zeros(3) eye(3) zeros(3,3) Y ]; % angular velocity Jacobian of
link 4
dJom4dphi(1:3,1:6+r.nv,1:6+r.nv) = zeros(3,6+r.nv,6+r.nv); % d/dphi(Jom4)

% Find the derivative of the velocity Jacobians with respect to phi where
dJv1dphi(:,j) dJv1/dphi(j)
Jv0 = [eye(3) zeros(3,r.nv+3)]; % velocity Jacobian for body 0
dJv0dphi = zeros(3,6+r.nv,6+r.nv);

Jv1 = [eye(3) c.skew(r01(:,1))' c.cross(Y,r11(:,1)) zeros(3,3)]; %
velocity Jacobian for link 1 [eye(3) r'' (Y x (pc1-p1)) zeros(3,3)]
dJv1dphi(1:3,1:3,1:6+r.nv) = zeros(3,3,6+r.nv); % d(eye(3))/dphi
dJv1dphi(1:3,4:6,1:4) = zeros(3,3,4); % d/dx0(r''), d/dy0(r''), d/dx0(r''), d
/dt0x(r'')
dJv1dphi(1:3,4:6,5) = c.skew( dRt0y*(r.l01+Rq1*r.c11) )'; % d/dt0y(r'')
dJv1dphi(1:3,4:6,6) = zeros(3,3); % d/dt0x(r'')
dJv1dphi(1:3,4:6,7) = c.skew( Rt0y*dRq1*r.c11 )'; % d/dq1(r'')

```

```

dJv1dphi(1:3,4:6,8:6+r.nv) = zeros(3,3,r.nv-1);% d/dq2(r''), d/dq3(r''), d/
dq4(r'')
dJv1dphi(1:3,7,1:4) = zeros(3,1,4); % d/dx0(Y x (pc1-p1)), d/dy0(Y x (pc1-p1)
), d/dx0(Y x (pc1-p1)), d/dt0x(Y x (pc1-p1))
dJv1dphi(1:3,7,5) = c.cross(Y, dRt0y*Rq1*r.c11 ); % d/dt0y(Y x (pc1-p1))
dJv1dphi(1:3,7,6) = zeros(3,1); % d/dt0x(Y x (pc1-p1))
dJv1dphi(1:3,7,7) = c.cross(Y, Rt0y*dRq1*r.c11 ); % d/dq1(Y x (pc1-p1))
dJv1dphi(1:3,7,8:6+r.nv) = zeros(3,1,r.nv-1), % d/dq2(Y x (pc1-p1)), d/dq3(Y
x (pc1-p1)), d/dq4(Y x (pc1-p1))
dJv1dphi(1:3,8:6+r.nv,1:6+r.nv) = zeros(3,3,6+r.nv); % d(zeros(3,4))/dphi

Jv2 = [eye(3) c.skew(r01(:,2))' c.cross(Y,r1l(:,2)) c.cross(Y,r2l(:,2))
zeros(3,2)]; % velocity Jacobian for link 2
dJv2dphi(1:3,1:3,1:11) = zeros(3,3,11); % d(eye(3))/dphi
dJv2dphi(1:3,4:6,1:4) = zeros(3,3,4); % d/dx0(r''), d/dy0(r''), d/dx0(r''), d
/dt0x(r'')
dJv2dphi(1:3,4:6,5) = c.skew( dRt0y*(r.l01+Rq1*(r.l12+Rq2*r.c22)) )'; % d/
dt0y(r'')
dJv2dphi(1:3,4:6,6) = zeros(3,3); % d/dt0x(r'')
dJv2dphi(1:3,4:6,7) = c.skew( Rt0y*dRq1*(r.l12+Rq2*r.c22) )'; % d/dq1(r'')
dJv2dphi(1:3,4:6,8) = c.skew( Rt0y*Rq1*dRq2*r.c22 )'; % d/dq2(r'')
dJv2dphi(1:3,4:6,9:6+r.nv) = zeros(3,3,r.nv-2);% d/dq3(r''), d/dq4(r'')
dJv2dphi(1:3,7,1:4) = zeros(3,1,4); % d/dx0(Y x (pc2-p1)), d/dy0(Y x (pc2-p1)
), d/dx0(Y x (pc2-p1)), d/dt0x(Y x (pc2-p1))
dJv2dphi(1:3,7,5) = c.cross(Y, dRt0y*Rq1*(r.l12+Rq2*r.c22) ); % d/dt0y(Y x (
pc2-p1))
dJv2dphi(1:3,7,6) = zeros(3,1); % d/dt0z(Y x (pc2-p1))
dJv2dphi(1:3,7,7) = c.cross(Y, Rt0y*dRq1*(r.l12+Rq2*r.c22) ); % d/dq1(Y x (
pc2-p1))
dJv2dphi(1:3,7,8) = c.cross(Y, Rt0y*Rq1*dRq2*r.c22 ); % d/dq2(Y x (pc2-p1))
dJv2dphi(1:3,7,9:6+r.nv) = zeros(3,1,r.nv-2); % d/dq3(Y x (pc2-p1)), d/dq4(Y
x (pc2-p1))
dJv2dphi(1:3,8,1:4) = zeros(3,1,4); % d/dx0(Y x (pc2-p2)), d/dy0(Y x (pc2-p2)
), d/dx0(Y x (pc2-p2)), d/dt0x(Y x (pc2-p2))
dJv2dphi(1:3,8,5) = c.cross(Y, dRt0y*Rq1*Rq2*r.c22 ); % d/dt0y(Y x (pc2-p2))
dJv2dphi(1:3,8,6) = zeros(3,1); % d/dt0x(Y x (pc2-p2))
dJv2dphi(1:3,8,7) = c.cross(Y, Rt0y*dRq1*Rq2*r.c22 ); % d/dq1(Y x (pc2-p2))
dJv2dphi(1:3,8,8) = c.cross(Y, Rt0y*Rq1*dRq2*r.c22 ); % d/dq2(Y x (pc2-p2))
dJv2dphi(1:3,8,9:6+r.nv) = zeros(3,1,r.nv-2); % d/dq3(Y x (pc2-p2)), d/dq4(Y
x (pc2-p2))
dJv2dphi(1:3,9:6+r.nv,1:6+r.nv) = zeros(3,r.nv-2,6+r.nv); % d(zeros(3,2))/
dphi

Jv3 = [eye(3) c.skew(r01(:,3))' c.cross(Y,r1l(:,3)) zeros(3,1) c.cross
(Y,r3l(:,3)) zeros(3,1)]; % velocity Jacobian for link 3
dJv3dphi(1:3,1:3,1:11) = zeros(3,3,11); % d(eye(3))/dphi
dJv3dphi(1:3,4:6,1:4) = zeros(3,3,4); % d/dx0(r''), d/dy0(r''), d/dx0(r''), d
/dt0x(r'')
dJv3dphi(1:3,4:6,5) = c.skew( dRt0y*(r.l01+Rq1*(r.l13+Rq3*r.c33)) )'; % d/
dt0y(r'')
dJv3dphi(1:3,4:6,6) = zeros(3,3); % d/dt0x(r'')
dJv3dphi(1:3,4:6,7) = c.skew( Rt0y*dRq1*(r.l13+Rq3*r.c33) )'; % d/dq1(r'')
dJv3dphi(1:3,4:6,8) = zeros(3,3); % d/dq2(r'')
dJv3dphi(1:3,4:6,9) = c.skew( Rt0y*Rq1*dRq3*r.c33 )'; % d/dq3(r'')
dJv3dphi(1:3,4:6,10) = zeros(3,3); % d/dq4(r'')
dJv3dphi(1:3,7,1:4) = zeros(3,1,4); % d/dx0(Y x (pc3-p1)), d/dy0(Y x (pc2-p1)
), d/dx0(Y x (pc3-p1)), d/dt0x(Y x (pc3-p1))
dJv3dphi(1:3,7,5) = c.cross(Y, dRt0y*Rq1*(r.l13+Rq3*r.c33) ); % d/dt0y(Y x (
pc3-p1))
dJv3dphi(1:3,7,6) = zeros(3,1); % d/dt0x(Y x (pc3-p1))
dJv3dphi(1:3,7,7) = c.cross(Y, Rt0y*dRq1*(r.l13+Rq3*r.c33) ); % d/dq1(Y x (
pc3-p1))
dJv3dphi(1:3,7,8) = zeros(3,1); % d/dq2(Y x (pc3-p1))
dJv3dphi(1:3,7,9) = c.cross(Y, Rt0y*Rq1*dRq3*r.c33 ); % d/dq3(Y x (pc2-p1))
dJv3dphi(1:3,7,10) = zeros(3,1); % d/dq4(Y x (pc3-p1))

```

```

dJv3dphi(1:3,8,1:6+r.nv) = zeros(3,1,6+r.nv); % d(zeros(3,1))/dphi
dJv3dphi(1:3,9,1:4) = zeros(3,1,4); % d/dx0(Y x (pc3-p3)), d/dy0(Y x (pc3-p3)
), d/dx0(Y x (pc3-p3)), d/dt0x(Y x (pc3-p3))
dJv3dphi(1:3,9,5) = c.cross(Y, dRt0y*Rq1*Rq3*r.c33 ); % d/dt0y(Y x (pc3-p3))
dJv3dphi(1:3,9,6) = zeros(3,1); % d/dt0z(Y x (pc3-p3))
dJv3dphi(1:3,9,7) = c.cross(Y, Rt0y*dRq1*Rq3*r.c33 ); % d/dq1(Y x (pc3-p3))
dJv3dphi(1:3,9,8) = zeros(3,1); % d/dq2(Y x (pc3-p3))
dJv3dphi(1:3,9,9) = c.cross(Y, Rt0y*Rq1*dRq3*r.c33 ); % d/dq3(Y x (pc3-p3))
dJv3dphi(1:3,9,10) = zeros(3,1); % d/dq4(Y x (pc3-p3))
dJv3dphi(1:3,10,1:6+r.nv) = zeros(3,1,6+r.nv); % d(zeros(3,1))/dphi

Jv4 = [eye(3) c.skew(r01(:,4))' zeros(3,3) c.cross(Y,r41(:,4))]; %
velocity Jacobian for link 1 [eye(3) r'' (Y x (pc1-p1)) zeros(3,4)]
dJv4dphi(1:3,1:3,1:6+r.nv) = zeros(3,3,6+r.nv); % d(eye(3))/dphi
dJv4dphi(1:3,4:6,1:4) = zeros(3,3,4); % d/dx0(r''), d/dy0(r''), d/dx0(r''), d
/dt0x(r'')
dJv4dphi(1:3,4:6,5) = c.skew( dRt0y*(r.l04+Rq4*r.c44) )'; % d/dt0y(r'')
dJv4dphi(1:3,4:6,6:9) = zeros(3,3,4); % d/dt0x(r''), d/dq1(r''), d/dq2(r''),
d/dq3(r'')
dJv4dphi(1:3,4:6,10) = c.skew( Rt0y*dRq4*r.c44 )'; % d/dq4(r'')
dJv4dphi(1:3,7:9,1:6+r.nv) = zeros(3,3,6+r.nv); % d(zeros(3,3))/dphi
dJv4dphi(1:3,10,1:4) = zeros(3,1,4); % d/dx0(Y x (pc4-p4)), d/dy0(Y x (pc4-p4)
), d/dx0(Y x (pc4-p4)), d/dt0x(Y x (pc4-p4))
dJv4dphi(1:3,10,5) = c.cross(Y, dRt0y*Rq4*r.c44 ); % d/dt0y(Y x (pc4-p4))
dJv4dphi(1:3,10,6:9) = zeros(3,1,4); % d/dt0x(Y x (pc4-p4)), d/dq1(Y x (pc4-
p4)), d/dq2(Y x (pc4-p4)), d/dq3(Y x (pc4-p4))
dJv4dphi(1:3,10,10) = c.cross(Y, Rt0y*dRq4*r.c44 ); % d/dq4(Y x (pc4-p4))

dhdphi = zeros(6+r.nv,6+r.nv,6+r.nv); % dH/dphi wrt each phi
dtdp = zeros(6+r.nv,1); % dT/dphi wrt each phi
for j = 1:6+r.nv % for all phi
    dhdphi(:,j) = r.m0*(dJv0dphi(:,j))*Jv0 + Jv0'*dJv0dphi(:,j)) +
    dJom0dphi(:,j)*r.I0*Jom0 + Jom0'*r.I0*dJom0dphi(:,j); % get
    dH/dphi(j) for body
    dhdphi(:,j) = dhdphi(:,j) + r.m1*(dJv1dphi(:,j))*Jv1 + Jv1'*
    dJv1dphi(:,j) + dJom1dphi(:,j)*r.I1(1:3,1:3,1)*Jom1 + Jom1
    '*r.I1(1:3,1:3,1)*dJom1dphi(:,j); % get dH/dphi(j) for joints
    dhdphi(:,j) = dhdphi(:,j) + r.m1*(dJv2dphi(:,j))*Jv2 + Jv2'*
    dJv2dphi(:,j) + dJom2dphi(:,j)*r.I1(1:3,1:3,2)*Jom2 + Jom2
    '*r.I1(1:3,1:3,2)*dJom2dphi(:,j);
    dhdphi(:,j) = dhdphi(:,j) + r.m1*(dJv3dphi(:,j))*Jv3 + Jv3'*
    dJv3dphi(:,j) + dJom3dphi(:,j)*r.I1(1:3,1:3,3)*Jom3 + Jom3
    '*r.I1(1:3,1:3,3)*dJom3dphi(:,j);
    dhdphi(:,j) = dhdphi(:,j) + r.m1*(dJv4dphi(:,j))*Jv4 + Jv4'*
    dJv4dphi(:,j) + dJom4dphi(:,j)*r.I1(1:3,1:3,4)*Jom4 + Jom4
    '*r.I1(1:3,1:3,4)*dJom4dphi(:,j);
    dtdp(j) = 1/2 * phidot' * dhdphi(:,j) * phidot; % dT/dphi(j) = 1/2 *
    phidot' * dH/dphi(j) * phidot
end
end
end

```

C.13 Derivative of Potential Energy with respect to Configuration

Listing C.12 presents the MATLAB function used to evaluate the derivative of potential energy with respect to configuration $\frac{\partial V}{\partial \Phi}$. This function is part of the `r` class, so is referred to as `r.dVdphi`.

Listing C.12: Derivative of potential energy with respect to configuration.

```
% Get the derivative of potential energy V with respect to phi
function dvdp = dVdphi (phi)
% Inputs: Configuration, phi [(1)x0 (2)y0 (3)z0 (4)theta0x (5)theta0y (6)theta0z
(7)q1 (8)q2 (9)q3 (10)q4]'
% Outputs: The derivative of potential energy V with respect to phi as a column
vector, dvdp (6+nv)x1
% Notes: The derivative of potential energy with respect to phi is used in the
euler lagrange equation d/dt(dL/dphidot)-dL/dphi = tau + tauC. It is
% in the equation for the Lagrangian, as: dL/dq = dT/dq - dV/dq.
Rt0y = c.rot(phi(5),'y'); % body theta0y rotation matrix
Rq1 = c.rot(phi(7),'y'); % joint angle q1 rotation matrix
Rq2 = c.rot(phi(8),'y'); % joint angle q2 rotation matrix
Rq3 = c.rot(phi(9),'y'); % joint angle q3 rotation matrix
Rq4 = c.rot(phi(10),'y'); % joint angle q4 rotation matrix
dRt0y = c.drot da(phi(5),'y'); % derivative of rotation matrix theta0y
wrt theta0y
dRq1 = c.drot da(phi(7),'y'); % derivative of rotation matrix q1 wrt q1
dRq2 = c.drot da(phi(8),'y'); % derivative of rotation matrix q2 wrt q2
dRq3 = c.drot da(phi(9),'y'); % derivative of rotation matrix q3 wrt q3
dRq4 = c.drot da(phi(10),'y'); % derivative of rotation matrix q4 wrt q4
% x0,y0,z0,t0x,t0y,t0z,q1,q2,q3,q4
dz0dphi = [0 0 1 0 0 0 0 0 0 0]'; % derivative of the height with respect to
phi for the body and each link
dz1dphi = [0 0 1 0 dRt0y(3,:)*(r.l01+Rq1*r.c11) 0
Rt0y(3,:)*dRq1*r.c11 0 0 0]';
dz2dphi = [0 0 1 0 dRt0y(3,:)*(r.l01+Rq1*(r.l12+Rq2*r.c22)) 0
Rt0y(3,:)*dRq1*(r.l12+Rq2*r.c22) Rt0y(3,:)*Rq1*dRq2*r.c22 0
0]';
dz3dphi = [0 0 1 0 dRt0y(3,:)*(r.l01+Rq1*(r.l13+Rq3*r.c33)) 0
Rt0y(3,:)*dRq1*(r.l13+Rq3*r.c33) 0 Rt0y(3,:)*Rq1*dRq3*r.
c33 0]';
dz4dphi = [0 0 1 0 dRt0y(3,:)*(r.l04+Rq4*r.c44) 0 0
0 0 Rt0y(3,:)*dRq4*r.c44
]';
dvdp = w.g*[dz0dphi dz1dphi dz2dphi dz3dphi dz4dphi]*[r.m0; r.m1]; %
derivative of potential energy with respect to phi as a column vector
end
```

Appendix D

Observability

Observability for a discrete-time system is defined as the ability to uniquely determine the initial state \mathbf{x}_0 at some final time k for any initial state \mathbf{x}_0 given knowledge of the control inputs \mathbf{u}_i and measurements \mathbf{y}_i for all $i \in [0, k]$ [35]. This appendix proves the observability of the Unscented Kalman Filters presented in this thesis. In a linear system, observability can be proved using an “observability matrix”. Consider a linear discrete-time system similar to the non-linear discrete-time system in Equations 5.6 and 5.7:

$$\mathbf{x}_{k+1} = \frac{\partial f}{\partial \mathbf{x}} \mathbf{x}_k + \frac{\partial f}{\partial \mathbf{u}} \mathbf{u}_k \quad (\text{D.1})$$

$$\mathbf{y}_k = \frac{\partial h}{\partial \mathbf{x}} \mathbf{x}_k \quad (\text{D.2})$$

For this system, the observability matrix \mathbf{E} is given by:

$$\mathbf{E} = \begin{bmatrix} \frac{\partial h}{\partial \mathbf{x}} \\ \frac{\partial h}{\partial \mathbf{x}} \frac{\partial f}{\partial \mathbf{x}} \\ \vdots \\ \frac{\partial h}{\partial \mathbf{x}} \left(\frac{\partial f}{\partial \mathbf{x}}^{n_x-1} \right) \end{bmatrix} \quad (\text{D.3})$$

The system is said to be observable if the observability matrix \mathbf{E} has a rank equal to the number of states n_x [35]. In this thesis the system being analyzed is non-linear. However, *linearized observability* will be proved by linearizing the process and measurement models, and proving that the linearized observability matrix has a rank of n_x . The symbolic mathematics required for these proofs was performed using the MATLAB symbolic toolbox. Intuitive explanations of the observability of the filters are also provided.

D.1 Estimator 1: The General Estimator

The state vector \mathbf{x}_1 was found to be observable using the measurement vector \mathbf{y}_1 . The wheel joint angular velocities $\{\dot{q}_2, \dot{q}_3, \dot{q}_4\}$ are directly observed through encoder measurements. When a certain state is observable in a Kalman Filter, its time-derivative is also observable. This is equivalent to the Kalman Filter being able to differentiate an observable state. In this manner, the wheel joint angular accelerations $\{\ddot{q}_2, \ddot{q}_3, \ddot{q}_4\}$ are also observable. Despite the wheel joint angular velocities not being included in the state vector \mathbf{x}_1 , their observability, combined with a knowledge of the commanded wheel torques $\{\tau_{W_1}, \tau_{W_2}, \tau_{W_3}\}$ renders the resistive torques $\{\tau_{R_1}, \tau_{R_2}, \tau_{R_3}\}$ observable. The rover pitch θ_{0_y} and bogie angle q_1 are directly observed through measurements. From these measurements, the first and second time derivatives of these quantities are also observable $\{\dot{\theta}_{0_y}, \dot{q}_1, \ddot{\theta}_{0_y}, \ddot{q}_1\}$. The rover accelerations in the world frame $\{\ddot{x}_0, \ddot{z}_0\}$ are observable through the accelerometer measurements $\{\ddot{x}_0^R, \ddot{z}_0^R\}$ and the observable estimate of rover pitch θ_{0_y} ; the observability of rover pitch θ_{0_y} allows the accelerations to be properly rotated from rover co-ordinates to world co-ordinates. The net effects of wheel forces $\{M_{0_{fw}}, M_{1_{fw}}, F_{x_{fw}}, F_{z_{fw}}\}$ are observable because they are in the process model equations for $\{\ddot{\theta}_{0_y}, \ddot{q}_1, \ddot{x}_0, \ddot{z}_0\}$ respectively (Equations 5.30-5.33); since there are four unknowns with four equations and all of the other terms in the equations have

been confirmed to be observable above, the remaining terms are observable. The forward velocity of the rover \dot{x}_0^R and the wheel slips $\{\iota_1, \iota_2, \iota_3\}$ are directly observed through measurements.

The observability matrix of Estimator 1 \mathbf{E}_1 is, for the sake of brevity, shown to have full rank using only the rows that were necessary; since there are 22 states in \mathbf{x}_1 , there will be 22 rows in \mathbf{E}_1 . The notation $\left[\frac{\partial h_1}{\partial \mathbf{x}_1}\right]_{\iota_j}$ refers to the sub-matrix which includes rows ι to j inclusive of $\frac{\partial h_1}{\partial \mathbf{x}_1}$. Note that the system inertia matrix \mathbf{H} is symmetric (i.e. $\mathbf{H}_{k,l} = \mathbf{H}_{l,k}$). The matrix \mathbf{E}_1 is shown row-by-row in Equation D.5.

$$\mathbf{E}_1 = \begin{bmatrix} \frac{\partial h_1}{\partial \mathbf{x}_1} \\ \left[\frac{\partial h_1}{\partial \mathbf{x}_1}\right]_{1\ 7} \frac{\partial f_1}{\partial \mathbf{x}_1} \\ \left[\frac{\partial h_1}{\partial \mathbf{x}_1}\right]_{4\ 7} \left(\frac{\partial f_1}{\partial \mathbf{x}_1}\right)^2 \end{bmatrix} \quad (\text{D.4})$$

$$[\mathbf{E}_1]_1 = [1, 0]$$

$$[\mathbf{E}_1]_2 = [0, 1, 0]$$

$$[\mathbf{E}_1]_3 = [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

$$[\mathbf{E}_1]_4 = [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

$$[\mathbf{E}_1]_5 = [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

$$[\mathbf{E}_1]_6 = [0, 0, 0, 0, 0, 0, -g \cos \theta_{0_y} - z_0 \cos \theta_{0_y} - x_0 \sin \theta_{0_y}, 0, 0, 0, 0, 0, \cos \theta_{0_y}, -\sin \theta_{0_y}, 0, 0, 0, 0, 0, 0, 0, 0]$$

$$[\mathbf{E}_1]_7 = [0, 0, 0, 0, 0, 0, x_0 \cos \theta_{0_y} - g \sin \theta_{0_y} - z_0 \sin \theta_{0_y}, 0, 0, 0, 0, 0, \sin \theta_{0_y}, \cos \theta_{0_y}, 0, 0, 0, 0, 0, 0, 0, 0]$$

$$[\mathbf{E}_1]_8 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]$$

$$[\mathbf{E}_1]_9 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]$$

$$[\mathbf{E}_1]_{10} = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]$$

$$[\mathbf{E}_1]_{11} = [0, 1, 0]$$

$$[\mathbf{E}_1]_{12} = [1 - (Tb_d)/I_{wg}, 0, 0, T/I_{wg}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

$$[\mathbf{E}_1]_{13} = [0, 1 - (Tb_d)/I_{wg}, 0, 0, T/I_{wg}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

$$\begin{aligned}
[\mathbf{E}_1]_{22} = & \left[- (b_d(\mathbf{H}_{5,8}\mathbf{H}_{7,7}T^2z_0 \sin \theta_{0_y} + \mathbf{H}_{5,8}\mathbf{H}_{7,7}T^2g \sin \theta_{0_y} - \mathbf{H}_{5,8}\mathbf{H}_{7,7}T^2x_0 \cos \theta_{0_y})) / (2\mathbf{H}_{5,5}\mathbf{H}_{7,7}I_{wg}) \right. \\
& - (b_d(2\mathbf{H}_{5,3}\mathbf{H}_{5,8}\mathbf{H}_{7,7} \cos \theta_{0_y} + 2\mathbf{H}_{5,5}\mathbf{H}_{7,3}\mathbf{H}_{7,8} \cos \theta_{0_y} + 2\mathbf{H}_{5,1}\mathbf{H}_{5,8}\mathbf{H}_{7,7} \sin \theta_{0_y} + 2\mathbf{H}_{5,5}\mathbf{H}_{7,1}\mathbf{H}_{7,8} \sin \theta_{0_y})) \\
& / (2\mathbf{H}_{1,1}\mathbf{H}_{5,5}\mathbf{H}_{7,7}I_{wg}), - (b_d(\mathbf{H}_{5,9}\mathbf{H}_{7,7}T^2z_0 \sin \theta_{0_y} + \mathbf{H}_{5,9}\mathbf{H}_{7,7}T^2g \sin \theta_{0_y} - \mathbf{H}_{5,9}\mathbf{H}_{7,7}T^2x_0 \cos \theta_{0_y})) / (2\mathbf{H}_{5,5}\mathbf{H}_{7,7}I_{wg}) \\
& - (b_d(2\mathbf{H}_{5,3}\mathbf{H}_{5,9}\mathbf{H}_{7,7} \cos \theta_{0_y} + 2\mathbf{H}_{5,5}\mathbf{H}_{7,3}\mathbf{H}_{7,9} \cos \theta_{0_y} + 2\mathbf{H}_{5,1}\mathbf{H}_{5,9}\mathbf{H}_{7,7} \sin \theta_{0_y} + 2\mathbf{H}_{5,5}\mathbf{H}_{7,1}\mathbf{H}_{7,9} \sin \theta_{0_y})) \\
& / (2\mathbf{H}_{1,1}\mathbf{H}_{5,5}\mathbf{H}_{7,7}I_{wg}), - (\mathbf{H}_{5,10}b_d(T^2g \sin \theta_{0_y} - T^2x_0 \cos \theta_{0_y} + T^2z_0 \sin \theta_{0_y})) / (2\mathbf{H}_{5,5}I_{wg}) \\
& - (\mathbf{H}_{5,10}b_d(2\mathbf{H}_{5,3} \cos \theta_{0_y} + 2\mathbf{H}_{5,1} \sin \theta_{0_y})) / (2\mathbf{H}_{1,1}\mathbf{H}_{5,5}I_{wg}), - \cos \theta_{0_y} ((\mathbf{H}_{5,3}(\mathbf{H}_{5,8}/I_{wg} - 1)) / (\mathbf{H}_{1,1}\mathbf{H}_{5,5})) \\
& + (\mathbf{H}_{7,3}(\mathbf{H}_{7,8}/I_{wg} - 1)) / (\mathbf{H}_{1,1}\mathbf{H}_{7,7})) - \sin \theta_{0_y} ((\mathbf{H}_{5,1}(\mathbf{H}_{5,8}/I_{wg} - 1)) / (\mathbf{H}_{1,1}\mathbf{H}_{5,5})) + (\mathbf{H}_{7,1}(\mathbf{H}_{7,8}/I_{wg} - 1)) / (\mathbf{H}_{1,1}\mathbf{H}_{7,7})) \\
& - (T^2(\mathbf{H}_{5,8}/I_{wg} - 1)(g \sin \theta_{0_y} - x_0 \cos \theta_{0_y} + z_0 \sin \theta_{0_y})) / (2\mathbf{H}_{5,5}), - \cos \theta_{0_y} ((\mathbf{H}_{5,3}(\mathbf{H}_{5,9}/I_{wg} - 1)) / (\mathbf{H}_{1,1}\mathbf{H}_{5,5})) \\
& + (\mathbf{H}_{7,3}(\mathbf{H}_{7,9}/I_{wg} - 1)) / (\mathbf{H}_{1,1}\mathbf{H}_{7,7})) - \sin \theta_{0_y} ((\mathbf{H}_{5,1}(\mathbf{H}_{5,9}/I_{wg} - 1)) / (\mathbf{H}_{1,1}\mathbf{H}_{5,5})) + (\mathbf{H}_{7,1}(\mathbf{H}_{7,9}/I_{wg} - 1)) / (\mathbf{H}_{1,1}\mathbf{H}_{7,7})) \\
& - (T^2(\mathbf{H}_{5,9}/I_{wg} - 1)(g \sin \theta_{0_y} - x_0 \cos \theta_{0_y} + z_0 \sin \theta_{0_y})) / (2\mathbf{H}_{5,5}), - ((\mathbf{H}_{5,10} - I_{wg})(T^2g \sin \theta_{0_y} - T^2x_0 \cos \theta_{0_y} + T^2z_0 \sin \theta_{0_y})) \\
& / (2\mathbf{H}_{5,5}I_{wg}) - ((\mathbf{H}_{5,10} - I_{wg})(2\mathbf{H}_{5,3} \cos \theta_{0_y} + 2\mathbf{H}_{5,1} \sin \theta_{0_y})) / (2\mathbf{H}_{1,1}\mathbf{H}_{5,5}I_{wg}), x_0 \cos \theta_{0_y} - g \sin \theta_{0_y} - z_0 \sin \theta_{0_y}, 0, \\
& (-2)T(g \sin \theta_{0_y} - x_0 \cos \theta_{0_y} + z_0 \sin \theta_{0_y}), 0, (\mathbf{H}_{5,7}\mathbf{H}_{7,3} \cos \theta_{0_y}) / (\mathbf{H}_{1,1}\mathbf{H}_{7,7}) - (3T^2(g \sin \theta_{0_y} - x_0 \cos \theta_{0_y} + z_0 \sin \theta_{0_y})) / 2 \\
& + (\mathbf{H}_{5,7}\mathbf{H}_{7,1} \sin \theta_{0_y}) / (\mathbf{H}_{1,1}\mathbf{H}_{7,7}), (\mathbf{H}_{5,7}(T^2g \sin \theta_{0_y} - T^2x_0 \cos \theta_{0_y} + T^2z_0 \sin \theta_{0_y})) / (2\mathbf{H}_{5,5}) + (\mathbf{H}_{5,7}(2\mathbf{H}_{5,3} \cos \theta_{0_y} \\
& + 2\mathbf{H}_{5,1} \sin \theta_{0_y})) / (2\mathbf{H}_{1,1}\mathbf{H}_{5,5}), \cos \theta_{0_y} ((\mathbf{H}_{5,1}\mathbf{H}_{5,3}) / (\mathbf{H}_{1,1}\mathbf{H}_{5,5})) + (\mathbf{H}_{7,1}\mathbf{H}_{7,3}) / (\mathbf{H}_{1,1}\mathbf{H}_{7,7})) + \sin \theta_{0_y} (\mathbf{H}_{5,1}^2 / (\mathbf{H}_{1,1}\mathbf{H}_{5,5}) \\
& + \mathbf{H}_{7,1}^2 / (\mathbf{H}_{1,1}\mathbf{H}_{7,7})) + (\mathbf{H}_{5,1}T^2(g \sin \theta_{0_y} - x_0 \cos \theta_{0_y} + z_0 \sin \theta_{0_y})) / (2\mathbf{H}_{5,5}), \sin \theta_{0_y} ((\mathbf{H}_{5,1}\mathbf{H}_{5,3}) / (\mathbf{H}_{1,1}\mathbf{H}_{5,5})) \\
& + (\mathbf{H}_{7,1}\mathbf{H}_{7,3}) / (\mathbf{H}_{1,1}\mathbf{H}_{7,7})) + \cos \theta_{0_y} (\mathbf{H}_{5,3}^2 / (\mathbf{H}_{1,1}\mathbf{H}_{5,5}) + \mathbf{H}_{7,3}^2 / (\mathbf{H}_{1,1}\mathbf{H}_{7,7})) + (\mathbf{H}_{5,3}T^2(g \sin \theta_{0_y} - x_0 \cos \theta_{0_y} + z_0 \sin \theta_{0_y})) \\
& / (2\mathbf{H}_{5,5}), - ((T^2g \sin \theta_{0_y}) / 2 - (T^2x_0 \cos \theta_{0_y}) / 2 + (T^2z_0 \sin \theta_{0_y}) / 2) / \mathbf{H}_{5,5} - (\mathbf{H}_{5,3} \cos \theta_{0_y} + \mathbf{H}_{5,1} \sin \theta_{0_y}) / (\mathbf{H}_{1,1}\mathbf{H}_{5,5}), \\
& - (\mathbf{H}_{7,3} \cos \theta_{0_y} + \mathbf{H}_{7,1} \sin \theta_{0_y}) / (\mathbf{H}_{1,1}\mathbf{H}_{7,7}), \sin \theta_{0_y} / \mathbf{H}_{1,1}, \cos \theta_{0_y} / \mathbf{H}_{1,1}, 0, 0, 0, 0 \Big] \tag{D 5}
\end{aligned}$$

The observability matrix \mathbf{E}_1 has a rank of 22. This is equal to the number of states in Estimator 1, meaning that it is observable.

D.2 Estimators 2 and 3: Wheel Force Estimators

Estimators 2 and 3 possess the same process and measurement models, so will be considered together. In Estimators 2 and 3, the state vectors $\{\mathbf{x}_2, \mathbf{x}_3\}$ are assumed to follow a random walk. The first three state variables, the wheel-terrain contact angles $\{\gamma_1, \gamma_2, \gamma_3\}$ are directly observable through measurements. Since these measurements are constructed using the rover kinematics, it can be said that these angles are only observable when the rover is in motion (i.e. when measurements of $\{\gamma_1, \gamma_2, \gamma_3\}$ are available). The next six state variables, the wheel normal loads

$\{W_1, W_2, W_3\}$ and drawbar pulls $\{DP_1, DP_2, DP_3\}$ are made observable through the combination of observable wheel-terrain contact angles $\{\gamma_1, \gamma_2, \gamma_3\}$ and measured forces $\{F_{L_1}, F_{L_2}, F_{L_3}, F_{\perp_1}, F_{\perp_2}, F_{\perp_3}\}$. The measured forces need only to be apportioned into forces perpendicular to the ground and forces parallel to the ground for the normal loads $\{W_1, W_2, W_3\}$ and drawbar pulls $\{DP_1, DP_2, DP_3\}$ to be obtained. Since the wheel-terrain contact angles $\{\gamma_1, \gamma_2, \gamma_3\}$ are observable, as are the rover pitch θ_{0_y} and bogie angle q_1 (from Estimator 1), the measured forces can be properly apportioned, making the normal loads and drawbar pulls observable.

Assuring full rank of the observability matrix of Estimator 2 \mathbf{E}_2 requires only the $\frac{\partial h_2}{\partial \mathbf{x}_2}$ term:

$$\begin{aligned} \mathbf{E}_2 = \mathbf{E}_3 &= \frac{\partial h_2}{\partial \mathbf{x}_2} = [[\mathbf{E}_2]_1, [\mathbf{E}_2]_2, [\mathbf{E}_2]_3, [\mathbf{E}_2]_4, [\mathbf{E}_2]_5, [\mathbf{E}_2]_6, [\mathbf{E}_2]_7, [\mathbf{E}_2]_8, [\mathbf{E}_2]_9]^T \\ [\mathbf{E}_2]_1 &= [1, 0, 0, 0, 0, 0, 0, 0, 0] \\ [\mathbf{E}_2]_2 &= [0, 1, 0, 0, 0, 0, 0, 0, 0] \\ [\mathbf{E}_2]_3 &= [0, 0, 1, 0, 0, 0, 0, 0, 0] \\ [\mathbf{E}_2]_4 &= [-DP_1 \cos(\gamma_1 - \epsilon_1) - W_1 \sin(\gamma_1 - \epsilon_1), 0, 0, \cos(\gamma_1 - \epsilon_1), 0, 0, -\sin(\gamma_1 - \epsilon_1), 0, 0] \\ [\mathbf{E}_2]_5 &= [0, -DP_2 \cos(\gamma_2 - \epsilon_1) - W_2 \sin(\gamma_2 - \epsilon_1), 0, 0, \cos(\gamma_2 - \epsilon_1), 0, 0, -\sin(\gamma_2 - \epsilon_1), 0] \\ [\mathbf{E}_2]_6 &= [0, 0, -DP_3 \cos(\gamma_3 - \epsilon_3) - W_3 \sin(\gamma_3 - \epsilon_3), 0, 0, \cos(\gamma_3 - \epsilon_3), 0, 0, -\sin(\gamma_3 - \epsilon_3)] \\ [\mathbf{E}_2]_7 &= [W_1 \cos(\gamma_1 - \epsilon_1) - DP_1 \cos(\gamma_1 - \epsilon_1), 0, 0, \sin(\gamma_1 - \epsilon_1), 0, 0, -\sin(\gamma_1 - \epsilon_1), 0, 0] \\ [\mathbf{E}_2]_8 &= [0, W_2 \cos(\gamma_2 - \epsilon_1) - DP_2 \cos(\gamma_2 - \epsilon_1), 0, 0, \sin(\gamma_2 - \epsilon_1), 0, 0, -\sin(\gamma_2 - \epsilon_1), 0] \\ [\mathbf{E}_2]_9 &= [0, 0, W_3 \cos(\gamma_3 - \epsilon_3) - DP_3 \cos(\gamma_3 - \epsilon_3), 0, 0, \sin(\gamma_3 - \epsilon_3), 0, 0, -\sin(\gamma_3 - \epsilon_3)] \end{aligned} \quad (\text{D.6})$$

The rank of matrix \mathbf{E}_2 is nine, which is the number of states in Estimators 2 and 3. Therefore Estimators 2 and 3 are observable.