

System Design, Motion Modelling and Planning for a Reconfigurable Wheeled Mobile Robot

William Reid

A thesis submitted in fulfillment
of the requirements of the degree of
Doctor of Philosophy



Australian Centre for Field Robotics
School of Aerospace, Mechanical and Mechatronic Engineering
The University of Sydney

April 2018

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the University or other institute of higher learning, except where due acknowledgement has been made in the text.

William Reid

2 April 2018

Abstract

William Reid
The University of Sydney

Doctor of Philosophy
April 2018

System Design, Motion Modelling and Planning for a Reconfigurable Wheeled Mobile Robot

Over the past five decades the use of mobile robotic rovers to perform in-situ scientific investigations on the surfaces of the Moon and Mars has been tremendously influential in shaping our understanding of these extraterrestrial environments. As robotic missions have evolved there has been a greater desire to explore more unstructured terrain. This has exposed mobility limitations with conventional rover designs such as getting stuck in soft soil or simply not being able to access rugged terrain. Additionally, future rovers will need to assist human explorers with surface operations such as haulage and construction during future long duration human missions to the Moon or Mars. Increased mobility and terrain traversability are key requirements when considering designs for next generation planetary rovers. Coupled with these requirements is the need to autonomously navigate unstructured terrain by taking full advantage of increased mobility.

To address these issues, a high degree-of-freedom reconfigurable platform that is capable of energy intensive legged locomotion in obstacle-rich terrain as well as wheeled locomotion in benign terrain is proposed. This pseudo-omnidirectional wheel-on-leg platform may reconfigure its footprint and clamber. The computational challenges of the planning task that considers the high degree-of-freedom state space of this platform are considerable. A variant of asymptotically optimal sampling-based planners that exploits the presence of dominant sub-spaces within a reconfigurable mobile robot's kinematic structure is proposed to increase path quality and ensure platform safety. It is also conjectured that the use of a mechanical work cost function within the planner that minimizes an energy objective facilitates the exploitation of the platform's various mobility modes.

The contributions of this thesis include: the design and implementation of a highly mobile planetary analogue rover; motion modelling of the platform to enable locomotion modes such as actively articulated suspension, rowing and inchworming, along with experimental validation of each of these capabilities; the sampling-based Hierarchical Bidirectional Fast Marching Trees (HBFMT*) planner that hierarchically considers sub-spaces to better guide search of the complete state space; and experimental validation of the planner with the physical platform that demonstrates how the planner exploits the robot's capabilities to fluidly transition between various physical geometric configurations and wheeled/legged locomotion modes.

Acknowledgements

Firstly, I would like to thank Dr. Ali Haydar Göktoğan for advising me all the way through this endeavour. I am very grateful for your attention to detail, kind mentorship and high expectations. Thanks also goes to Assoc. Prof. Robert Fitch. Thank you for the many discussions about motion planning, your critical analysis and enthusiasm for new ideas. To Prof. Salah Sukkarieh, thank you for your guidance and for providing me with many great opportunities to work on novel robotic platforms and to collaborate with overseas research institutions.

The ACFR is a research and engineering group that does extraordinary work on a daily basis. It has been a privilege to work with and learn from true experts. To my good friends and fellow MAMMOTH wranglers Muhammand Esa Attia and Javier Martinez, thank you for your unwavering support in the quest to develop some pretty outlandish robotic concepts. Designing the MAMMOTH system with you has been an invaluable learning experience. Steven Potiris, thank you for helping me out of so many technical conundrums. You are an excellent engineer and friend. Warwick, Justin, Oliver, Felix, Nathan, Graeme, Suchet, John and the rest of my ACFR colleagues, thank you for all the cups of coffee and endless robot talk.

To my overseas colleague Francisco Javier Pérez Grau, thank you for your collaboration on making the MAMMOTH rover see the terrain around it. To Florian Cordes, thank you for hosting me in Bremen and your enthusiasm for wheel-on-leg robots.

To Belinda and all of the housemates at 368, thank you for the many conversations and glasses of wine. You have all made the Sydney experience a wonderful one.

Jack, thank you for always reminding me to not take things too seriously. It has been a great pleasure working alongside you for all this time.

Rieteke, Chris, Jack and Jim, thank you for being so supportive and being such a wonderful gang of outlaws.

Mum, Dad and Bella, thank you for putting up with what seems like an eternity of university. I am so lucky to have such a loving, interested and supportive family who have always been there for me no matter what.

Genevieve, I could not have done this without you. Thank you for always being there and always making things better. We have had some great adventures together, here's to many more!

Contents

Declaration	ii
Abstract	iii
Acknowledgements	iv
Contents	v
List of Figures	ix
List of Tables	xii
List of Algorithms	xiii
Nomenclature	xiv
Abbreviations	xx
Definitions	xxii
1 Introduction	1
1.1 The Problem of Planning for Increased Mobility	2
1.2 Thesis Objectives	3
1.3 Contributions	4
1.4 Thesis Structure	5
1.5 Publications	6
1.6 Supplementary Material	6
2 Related Work	7
2.1 Mobile Martian and Lunar Exploration Robots	7
2.1.1 Space Race Rovers	7
2.1.2 The NASA Mars Rover Missions	9
2.1.3 Actively Articulated Planetary Analogue Rovers	11

2.2	Kinematic Modelling of WMRs	13
2.2.1	Geometric Models	14
2.2.2	Transformation-based Models	14
2.2.3	Velocity Propagation Models	15
2.2.4	Stability Models	16
2.3	Planning for RWMRs	17
2.3.1	Grid-Based Planning Techniques	17
2.3.2	Sampling-Based Planning	18
2.3.3	The Fast Marching Tree Planner	19
2.3.4	Hierarchical Planning	22
2.4	Summary	25
3	The MAMMOTH Rover System	26
3.1	Mechanical Design	26
3.1.1	The MAMMOTH Rover Leg	27
3.1.2	Structure	29
3.2	Electrical Design	31
3.2.1	Power and Safety	31
3.2.2	Computing and Communications	33
3.2.3	Actuation	34
3.2.4	Sensing	35
3.2.5	Expandability	36
3.3	Software Design	36
3.3.1	Low Level Mobility Control Software	37
3.3.2	High Level Mobility Control Software	38
3.4	The Mars Lab	39
4	RWMR Motion Modelling	41
4.1	Kinematic Notation	41
4.2	Position Kinematics	44
4.2.1	The RWMR Kinematic Tree	45
4.2.2	Static Stability	47
4.3	Velocity Kinematics	48
4.3.1	General Velocity Kinematic Model	49
4.3.2	Steering and Driving	50
4.3.3	Mobility Analysis	51
4.4	Alternative Mobility Modes	52
4.4.1	Rowing	53
4.4.2	Inchworming	55
4.5	Actively Articulated Suspension	57
4.5.1	Articulation Solution	57
4.5.2	Terrain Mapping	58

5	Kinematic Modelling Validation Experiments	61
5.1	Kinematic Model Validation	61
5.1.1	Setup	61
5.1.2	Discussion	64
5.2	Alternative Mobility Mode Demonstration	65
5.2.1	Setup	65
5.2.2	Rowing	65
5.2.3	Inchworming	70
5.2.4	Energy Usage	71
5.3	Actively Articulated Suspension Demonstration	73
5.3.1	Experimental Setup	73
5.3.2	Results and Discussion	73
6	Hierarchical Sampling-Based Motion Planning for RWMRs	78
6.1	Problem Formulation and Background	79
6.1.1	Problem Definition	79
6.1.2	Bidirectional Fast Marching Tree	80
6.2	Hierarchical Bidirectional Fast Marching Tree	81
6.3	Analysis	85
6.4	Planning for the MAMMOTH Rover	88
6.4.1	Collision-Based Feedback Sampling	88
6.4.2	Cost Function	90
6.5	Numerical Experiments	90
6.5.1	Environments	91
6.5.2	Experimental Setup	92
6.5.3	Results	93
6.5.4	Parameter Tuning	98
6.6	Summary	99
7	Plan Following Experiment	105
7.1	Setup	105
7.2	Results	107
7.2.1	Trial 56: Unexpected Emergent Behaviour	108
7.2.2	Trial 61: Clambering Example	110
7.2.3	Planner Validation	112
7.3	Summary	114

8 Conclusion	117
8.1 Principal Contributions	117
8.2 Future Work	118
Bibliography	120
A Kinematics	130
A.1 Recursive Kinematic Propagation Examples	130
A.1.1 MAMMOTH Rover Position Kinematics	130
A.1.2 MAMMOTH Rover Leg Relative Positions and Rotations Using DH Parameters	130
A.1.3 Jacobian Formulation Example	131
A.2 Body Rotation Rates	133
A.3 Centre of Mass Calculation	133
A.4 Euler Angles	134
A.5 Thigh Joint Kinematics	134

List of Figures

1.1	The Spirit embedding event.	2
1.2	Thesis outline.	5
2.1	Lunar and Martian exploration rovers.	8
2.2	An autonomous drive by Mars Exploration Rover (MER) Opportunity. . . .	10
2.3	The ATHLETE and SherpaTT robots.	11
2.4	Diagram depicting the force-angle stability margin.	16
2.5	Robots that have used hierarchical planners.	22
3.1	Labelled view of the MAMMOTH rover.	27
3.2	Labelled view of the MAMMOTH rover leg.	28
3.3	Illustration of the MAMMOTH rover's reconfigurability.	29
3.4	Labelled views of the MAMMOTH rover's body structure.	30
3.5	System architecture of the MAMMOTH rover.	32
3.6	Two separate MAMMOTH rover sensor configurations.	35
3.7	MAMMOTH rover software diagram.	37
3.8	The Mars Lab, an analogue Martian terrain.	40
4.1	Labelled view of the Mars Analogue Multi-MOde Traverse Hybrid (MAMMOTH) rover.	42
4.2	Labelled view of the MAMMOTH rover leg.	43
4.3	The wheel contact patch and multiple contact points.	44
4.4	The MAMMOTH rover's stability model.	47
4.5	The MAMMOTH rover steering and wheel drive joints.	51
4.6	The rowing manoeuvre.	53

4.7	The inchworming manoeuvre.	56
4.8	Actively articulated suspension.	57
4.9	Block diagram for the actively articulated suspension controller.	59
4.10	A terrain point cloud augmented with RGB data.	60
5.1	Kinematic model validation experiment path.	62
5.2	Timelapses from each of the kinematic model validation trials.	63
5.3	A timelapse of a single rowing cycle.	66
5.4	Rowing wheel tracks.	66
5.5	Rowing trial translation plots.	67
5.6	Kinematic model validation experimental results.	68
5.7	Hip position, rate and current plots for rowing.	69
5.8	A timelapse of a single inchworming cycle.	70
5.9	Inchworming translation and thigh position plots.	71
5.10	Actively articulated suspension trial 1 timelapse.	74
5.11	Actively articulated suspension trial 1 results.	75
5.12	Actively articulated suspension trial 2 timelapse.	76
5.13	Actively articulated suspension trial 2 results.	76
5.14	Actively articulated suspension trial 3 timelapse.	77
5.15	Actively articulated suspension trial 3 results.	77
6.1	Outline of the HBFMT* algorithm.	82
6.2	The ‘Spin to Win’ environment.	86
6.3	Comparison of HBFMT* single radius values.	87
6.4	Comparison of HBFMT* tunnel radii sizes.	88
6.5	HBFMT* numerical experiment environments.	91
6.6	Gauntlet experimental results.	94
6.7	The simulated MAMMOTH rover passing through a narrow passageway.	95
6.8	Alternatives experimental results.	96
6.9	Mars Lab Corridors experimental results.	97
6.10	Mars Lab Tunnels experimental results.	98
6.11	HBFMT* parameter tuning sensitivity experimental results.	100

6.12	A MAMMOTH rover path in the Gauntlet environment.	101
6.13	A MAMMOTH rover path in the Alternatives environment.	102
6.14	A MAMMOTH rover path in the Mars Lab Corridors environment.	103
6.15	A MAMMOTH rover path in the Mars Lab Tunnels environment.	104
7.1	Environments used for the plan following experiments.	106
7.2	Path following experiment timelapse from Trial 56.	108
7.3	Path following experiment timelapse from Trial 61.	110
7.4	Comparison between the physical paths followed and the planned paths. . .	111
7.5	Successful traversal mechanical work costs.	112
7.6	Successful traversal energy costs.	113
7.7	Trial 56 body position, actuator position and sub-space cost plots.	115
7.8	Trial 61 body position, actuator position and sub-space cost plots.	116
A.1	A labelled view of the MAMMOTH rover leg i thigh.	135
A.2	MAMMOTH rover thigh joint kinematic variables.	136

List of Tables

1.1	Physical experiment video playlist URLs.	6
4.1	Kinematic relationships within the MAMMOTH rover structure.	45
5.1	Distances traversed in the alternative mobility mode experiments.	66
5.2	Energy usage by the MAMMOTH rover during the alternative mobility modes.	72
6.1	The planners used in the numerical benchmarking experiment.	92
6.2	The tuning parameters used for HBFMT* for each environment.	92
7.1	The path following experiment initial and goal states for each trial.	106
7.2	Plan following setup details for the successful path traversals.	107

List of Algorithms

1	Fast Marching Tree (FMT*) [60]	21
2	Bidirectional Fast Marching Tree (BFMT*)	80
3	Fast Marching Tree (FMT*) Expansion [118]	81
4	Initialize a Fast Marching Tree [118]	81
5	Hierarchical Bidirectional Fast Marching Tree (HBFMT*)	83
6	Singe Branch	83
7	Sample Free	84
8	Sample from the Biased Distribution.	84
9	Sample Free MAMMOTH Rover State with Collision Feedback	89

Nomenclature

Generic Notation

\underline{u} A column vector.

$[\underline{u}]^\times$ A vector that has the skew symmetric operation performed to it.

$I_{g \times g}$ A $g \times g$ identity matrix.

$\underline{0}_{g \times h}$ A $g \times h$ zero vector or matrix.

$\underline{u}(1 : h)$ A vector composed of the 1 to h rows of column vector \underline{u} . All vectors and matrices are one-base indexed unless otherwise specified.

$M(1 : h, 1 : g)$ The sub-matrix of M that is composed of M 's 1 to h rows and 1 to g columns.

Coordinate Frames

I The inertial coordinate frame.

B The body coordinate frame.

H_i The hip coordinate frame for leg i .

U_i The upper thigh coordinate frame for leg i .

Th_i The thigh coordinate frame for leg i .

Lo_i The ankle coordinate frame for leg i .

A_i The wheel coordinate frame for leg i .

A'_i The wheel coordinate frame for leg i that rotates with the wheel.

C_i The wheel/terrain coordinate frame for leg i .

Kinematic Indices

i The kinematic branch index.

j The kinematic branch base frame index.

k A generic frame index.

n The kinematic branch terminal frame index.

Position Kinematics

${}^{F_0}r_{F_2}^{F_1}$	A 3×1 vector expressing the position of frame F_2 relative to frame F_1 , expressed in F_0 .	[m]
$\underline{r}_{F_2}^{F_1}$	A 3×1 vector expressing the position of frame F_2 relative to frame F_1 . As a short-hand notation, if there is not a frame written in the super-script preceding the position vector it is assumed that the position is expressed in the frame written in the super-script after the position vector.	[m]
$R_{F_2}^{F_1}$	A 3×3 rotation matrix expressing a rotation from frame F_2 to frame F_1 .	
x_B^I	The translation of the B frame relative to the I frame along the x -axis of the I frame.	[m]
y_B^I	The translation of the B frame relative to the I frame along the y -axis of the I frame.	[m]
z_B^I	The translation of the B frame relative to the I frame along the z -axis of the I frame.	[m]
ϕ	Body roll angle.	[°]
θ	Body pitch angle.	[°]
ψ	Body yaw angle.	[°]
\underline{q}_H	The 4×1 vector of the hip joint angular positions.	[°]
\underline{q}_U	The 4×1 vector of the thigh joint angular positions.	[°]
q_{H_i}	The hip joint angular position in leg i .	[°]
q_{U_i}	The upper thigh joint angular position in leg i .	[°]
q_{Lo_i}	The lower thigh joint angular position in leg i .	[°]
$q_{A'_i}$	The wheel drive angular position in leg i .	[°]
$\underline{\sigma}_i$	The 6×1 slip vector of the contact frame for leg i .	[m]
$\underline{\gamma}_i$	The variable length articulation vector for leg i .	[°]
$\underline{\gamma}$	The variable length articulation vector.	[°]
\underline{p}	The 6×1 body pose vector.	
ε_i	The angle between the y -axis of frame U_i and a line that is parallel to the z -axis of I .	[°]
κ_i	The angle between the x -axis of U_i and the plane that is parallel to the xy -plane that intersects with I .	[°]

Static Stability

r_{\bullet}^I	A 3×1 position vector that describes the position of the centre of mass of the robot relative to the inertial frame. [m]
\underline{f}_g	The 3×1 weight vector extending from the rover's centre of mass parallel to the inertial frame's z -axis. [N]
\underline{a}_i	A 3×1 unit vector that extends from wheel contact point i to wheel contact point $i + 1$. If $i = n$, then the vector extends to contact point 1.
$\underline{d}_{C_i}^B$	A 3×1 position vector, which is the position of wheel i 's contact point expressed relative to the B frame. [m]
\underline{l}_i	A 3×1 position vector that extends from the centre of mass and is orthogonal to \underline{a}_i . [m]
η_i	The force angle stability measure for tip-over axis i . [$^\circ$]
β	The minimum force angle stability measure, which is the force angle stability metric for a given robot pose. [$^\circ$]
m	The number of contact points.

Velocity Kinematics

$\underline{v}_{F_2}^{F_1}$	A 3×1 translational velocity vector of frame F_2 relative to frame F_1 . [m/s]
$\underline{\omega}_{F_2}^{F_1}$	A 3×1 vector angular velocity vector of frame F_2 relative to frame F_1 . [$^\circ$ /s]
$J_{F_2}^{F_1}$	The variable sized Jacobian matrix that relates the velocities of all joints along the kinematic chain between frame F_1 and frame F_2 to the translational and angular velocities of frame F_1 .
P	The 6×6 body rate Jacobian.
$Q_{C_i}^I$	The 6×3 articulation rate Jacobian for leg i .
P_{full}	6×24 full body rate Jacobian containing stacked P matrices.
Q_{full}	The 24×12 joint articulation rate Jacobian containing all of the individual leg Q_i Jacobians.
$P_{\underline{v}_B}^I$	The 6×3 body rate Jacobian that maps the translational velocities of the body frame.
$Q_{C_i, \underline{q}_H}^I$	The 6×1 articulation rate Jacobian for leg i that maps the hip angular rates.
$Q_{C_i, \underline{q}_U}^I$	The 6×1 articulation rate Jacobian for leg i that maps the upper thigh angular rates.

\underline{d}_{k+1}^k	A 6×1 vector that maps the joint rate associated with frame k to the adjacent $k + 1$ frame.
Ω	A Jacobian is composed of an identity matrix, P_{full} and Q_C^I .
$\underline{\dot{q}}$	A vector that is composed of the stacked wheel contact rate, body rate and joint rate vectors.

Alternative Mobility Modes

ζ	The quadrant index that a WMR's velocity vector is in with respect to the inertial frame.
I'	The inertial coordinate frame that has its x -axis parallel to the WMRs direction of travel.
V	A 4×4 permutation matrix that is used to transform terminal joint angle vectors or discrete leg index vectors.
ι	An offset angle that is used to find initial and terminal hip joint angles for the rowing manoeuvre.
Λ	A 6×6 Jacobian that is used to transform a velocity vector into the I' coordinate frame.
e	The index basis vector used in formulating inchworming.
e'	The base index basis vector used in formulating inchworming.
S	A selection vector or matrix used to reduce a kinematic system of equations.

Actively Articulated Suspension

K_r	A position gain for the actively articulated suspension control policy.
d_{look}	A lookahead distance. [m]

Planning

\mathcal{X}	A state space.
$\mathcal{X}^\#$	A sub-state space that is contained within \mathcal{X} , $\mathcal{X}^\# \subset \mathcal{X}$.
\mathcal{X}_{obs}	The region of the state space occupied by obstacles.
\mathcal{X}_{free}	The region of the state space that is free of obstacles.
d	The dimension of a state space.
\underline{x}_{init}	A $d \times 1$ vector that describes the initial state.
\underline{x}_{goal}	A $d \times 1$ vector that describes the goal state.
\mathcal{X}_{goal}	A goal region.

e	The dimension of a sub-state space.
Σ	The set of all feasible paths from the initial to goal state.
π	A feasible path from \underline{x}_{init} to \underline{x}_{goal} .
π^*	An optimal path from \underline{x}_{init} to \underline{x}_{goal} .
Π	A set of paths from \underline{x}_{init} to \underline{x}_{goal} .
c	An arc cost function.
δ	A distance from \mathcal{X}_{obs} along all points within a path that defines a robustly feasible path.
n	The number of states successfully sampled by a sampling-based planner.
V	A set of states.
E	A set of edges that connect states.
T	A tree composed of a set of states V and a set of edges E .
H	A set of states called the open set. This set is used in the Expansion routine of the FMT* algorithm to label nodes that are currently being considered for expansion.
W	A set of states called the unvisited set. This set is used in the Expansion routine of the FMT* algorithm to label nodes that have yet to be expanded into.
r_n	A radius that defines a state's set of neighbouring states.
k_n	The number of nearest neighbours to a state.
r_{singl}	The singe radius defines the minimum allowable distance between BFMT* collision nodes within the HBFMT* algorithm.
r_{tunnel}	The tunnel radius defines the size of the region around the sub-space paths to be sampled in the full-space search within the HBFMT* algorithm.
ℓ	The probability that states will be sampled uniformly from the state space within the HBFMT* planner.
k	The number of samples taken per batch in the BIT* algorithm.
$\mu(\mathcal{X}_{free})$	The Lebesgue measure of the free space in \mathcal{X} .
ζ_d	The Lebesgue measure of a unit-cost ball with dimensionality d .

Plan Following Experiments

ϵ	The margin used within the path following controller to determine if a waypoint in a path has been reached and if the next waypoint may be pursued.
Q	The total traversal energy expenditure.

I_i	The current draw of actuator i .
V_i	The actuator voltage of actuator i .
$\hat{\underline{c}}$	The vector of accumulated mechanical work costs for a sub-space i in a physical path following trial.
\underline{c}	The vector of accumulated mechanical work costs for a sub-space i returned by the path planner.
m	The number of actuators.

Abbreviations

ACFR	Australian Centre for Field Robotics
aFMT*	Anytime Fast Marching Tree
AO	asymptotically optimal
ATHLETE	All-Terrain Hex-Limbed Extra-Terrestrial Explorer
AutoNav	Autonomous Navigation
BBB	BeagleBone Black
BFMT*	Bidirectional Fast Marching Trees
BIT*	Batch Informed Tree
C-FOREST	Coupled Forest Of Random Engrafting Search Trees
CHIMP	CMU Highly Intelligent Mobile Platform
CMU	Carnegie Mellon University
CPU	central processing unit
DAE	differentiable algebraic equation
DARPA	Defense Advanced Research Projects Agency
DC	direct current
DH	Denavit-Hartenberg
DOF	degree-of-freedom
DRC	DARPA Robotics Challenge
DXL-Pro	Dynamixel Pro
EET	Exploration Exploitation Tree
EST	Expansive Space Tree
E-STOP	emergency stop
FIDO	Field Integrated Design and Operations rover
FK	forward kinematics
FMT*	Fast Marching Tree
FS	feedback sampling
GESTALT	Grid-based Estimation of Surface Traversability Applied to Local Terrain
HBFMT*	Hierarchical Bidirectional Fast Marching Trees
HIL	hardware-in-the-loop
HRRT*	Hierarchical Rapidly Exploring Random Trees
ICP	Iterative Closest Point
IK	inverse kinematics
IMU	inertial measurement unit
IP	Internet Protocol

KAIST	Korean Advanced Institute of Science and Technology
LCD	liquid crystal display
LTS	Long Term Support
Li-Ion	lithium ion
LRV	Lunar Roving Vehicle
MAMMOTH	Mars Analogue Multi-MODE Traverse Hybrid
MER	Mars Exploration Rover
MPLB	Motion Planning using Lower Bounds
MSL	Mars Science Laboratory
MSR	Mars Sample Return
NASA	National Aeronautics and Space Administration
NFS	no feedback sampling
OMPL	Open Motion Planning Library
OS	operating system
P	proportional
PC	probabilistically complete
PI	proportional integral
PID	proportional integral derivative
PLC	programmable logic controller
PoE	Power over Ethernet
PRM	Probabilistic Roadmap
PSPACE	polynomial space
RGB	red, green and blue
RGB-D	red, green, blue and depth
RKP	recursive kinematic propagation
ROS	Robotics Operating System
RRT	Rapidly-exploring Random Tree
RWMR	reconfigurable wheeled mobile robot
SBL	Single-query, Bidirectional, Lazy in collision-checking
SBP	sampling-based planning
SDK	software development kit
SRR	Sample Return Rover
SyCLoP	Synergistic Combination of Layers of Planning
TCP/IP	Transmission Control Protocol/Internet Protocol
UDP	User Datagram Protocol
WMR	wheeled mobile robot

Definitions

Clambering: The act of a reconfigurable wheeled mobile robot (RWMR) lifting one or more legs over obstacles so as to traverse the obstacles with no pre-defined gait and remaining statically stable.

Configuration Space: The set of possible transformations that may be applied to a robot. A configuration describes the translational and angular positions of each of the generalized coordinates of a robot.

Hierarchical Planner: A motion planning technique that comprises a set of motion planners that plan in different state spaces and use solutions from the other planners to guide their own planning. At the *high level* of the hierarchy, a low resolution planning problem is addressed. Low-resolution refers to either the state space representation and/or a reduced dimensionality of the state space. At the low-level of the hierarchy, high resolution state spaces are planned through.

Lunar/Martian Analogue Terrain: A terrain that simulates surface conditions on the Moon or Mars.

Node (Software): Within the context of software development for the Robotics Operating System (ROS), a node is a process that performs some computation. Nodes interact with each other by sending messages over the Transmission Control Protocol/Internet Protocol (TCP/IP) or the User Datagram Protocol (UDP).

Node (Planning): Within the context of robot motion planning, a node is a state sampled from a state space. Motion planners attempt to connect nodes to form a path or trajectory from an initial to a goal state.

Pose: The six dimensional vector in the special Euclidean space $\underline{p} \in SE(3)$. The pose of a wheeled mobile robot refers to the position and rotation of the robot's body frame with respect to an inertial frame.

Pseudo-Omnidirectional: An ability of a wheeled mobile robot (WMR) with all independently steered and driven wheels to move in any direction on a plane, however not instantaneously due to the non-holonomic constraints introduced by the steering actuators [24].

Reconfigurable Wheeled Mobile Robot (RWMR): A cyber-physical agent that is propelled by wheels and may morph the shape of its body through joint articulation.

In this thesis a RWMR typically refers to a robot that may actively drive its joints to change its shape to affect its static stability, conform to different terrain profiles and to clamber or walk in combination with wheel drive locomotion.

State Space: The set of situations that a robot may be in. In this thesis the term state space is used to describe all possible positions, velocities and/or accelerations of a robot.

Unstructured Terrain: A terrain containing obstacles or hazards that are not regularly spaced.

Workspace: The three dimensional Euclidean space $\mathcal{W} \in \mathbb{R}^3$. It is typically associated with the position of a wheeled mobile robot's body frame with respect to an inertial frame.

Chapter 1

Introduction

To conduct in-situ surface investigations of rocky extraterrestrial environments, the most cost-effective and technologically feasible option has been to use wheeled mobile robots (WMRs) complete with suites of sensors and investigatory tools. From the Soviet Union's tank-like lunar exploration *Lunokhod* rovers of the early 1970s, to the United States' increasingly sophisticated National Aeronautics and Space Administration (NASA) Mars rovers of the 1990s and early 21st century, these systems have been used with great effect as proxies for human explorers. NASA's three most recent Mars rovers, landers and array of Martian satellites have returned compelling data that show a present abundance of water ice on Mars and have highlighted a complex water history as detailed in [115], [116] and [117]. Most recently, the Mars Science Laboratory (MSL) *Curiosity* has collected data sets from Gale Crater that indicate past habitable conditions in this ancient lake bed [45].

Much of the scientifically interesting terrain that rovers are used to explore is unstructured. Unlike a road or path, these environments are full of hazards such as rock obstacles, soft soil and uneven surfaces. The NASA Mars rovers have used the very reliable and robust passively articulated rocker-bogie suspension system to conform to rough terrain and maximize wheel traction [16]. Some of these rovers have encountered situations in which terrain traversal has been impeded. The most notable example of this was when the Mars Exploration Rover (MER) *Spirit* became stuck in 2009 and was unable to be freed [126]. This failure was due to the soft soil being driven over as well as the failure of two wheel driving actuators. *Spirit*'s final resting place at Troy, Gusev Crater, is shown in Figure 1.1. A more recent and less dramatic example of mobility difficulties was seen with the MSL rover, in which significant slippage and sinkage into sand was observed during traverses over megaripple sand dune fields [4]. Additionally, wheel puncturing due to traversal over sharp rocks has been observed.

The challenges faced in remotely operating a mobile science laboratory on another planet such as Mars are immense. Due to the significant light distances separating the robot from its human operators on Earth, tele-operation is significantly delayed. Compounding this is the limited communication bandwidth and communication windows that are a function of the direct line of sight between Earth and the rover. The robot must have some degree of autonomy to make decisions about its instructions and state in real time when no humans

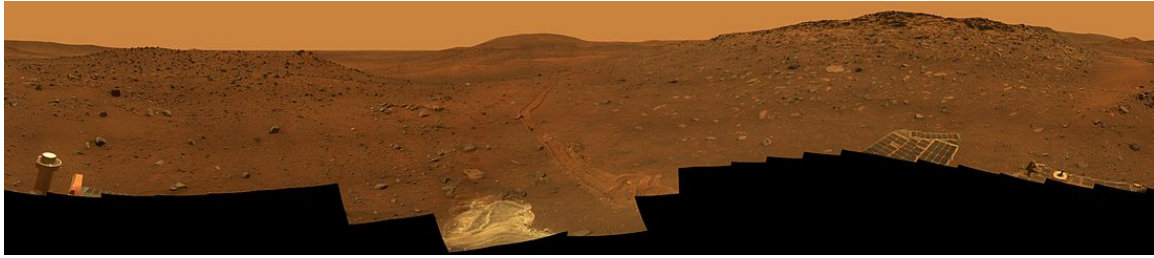


Figure 1.1 – A panorama mosaic of images taken by the MER-A (Spirit) at Troy, the site of the rover’s mobility ending ‘embedding event’ [80]. This environment is an example of unstructured terrain full of rock obstacles and soft soil that the rover had to navigate throughout its five year mobile surface operations.

are available to supervise. Given the significant operational constraints and need to explore more rugged terrain, a greater reliance on rover autonomy is required. Introducing autonomy to planetary exploration is a popular area of research as highlighted in [10], [23], [74], [121] and [124]. Given basic instructions from human operators about where to move to and what sites to monitor, the robot should be able to perceive the space around it, plan a set of actions to pass through this space, and then perform these actions to follow the plan. Specifically looking at the planning problem, the rover should be able to reason about the safe and efficient traverses it is capable of performing based on a pre-defined motion model and knowledge of the terrain around it

1.1 The Problem of Planning for Increased Mobility

A type of robot that exhibits a high degree of mobility is known as a reconfigurable wheeled mobile robot (RWMR). Such a robot can be used to overcome the mobility problems previously seen by planetary rovers. This is done by combining wheeled and legged locomotion modes to fluidly transition over, under or around unstructured obstacles by driving pseudo-omnidirectionally in combination with morphing its geometric structure.

With an increased mobility the reliance on autonomous selection of motions to achieve mission goals becomes increasingly important due to the many locomotion modes available and the possibility of unsafe geometric states. Autonomous planning for a high degree of freedom rover is challenging due to the size of its state space, in addition to the complexity of the motion planning problem itself.

Such a planning problem may be addressed with sampling-based planning (SBP) algorithms that randomly probe the configuration space with a collision checking routine and then attempt to connect valid states. Even though variants of these algorithms have probabilistically complete (PC) and asymptotically optimal (AO) guarantees, they are still limited in terms of computational efficiency and returned path quality. When considering the high degree-of-freedom (DOF) planning problem for the distinct geometry of an RWMR with strict safety constraints, it is desirable to consider new ways in which SBP algorithms may be modified to increase path quality and computational efficiency.

The problem addressed in this thesis relates to the following question: *with driving, clambering and footprint morphing abilities, can a path be planned for an RWMR by hierarchically considering sub-spaces of the state space to leverage the platform's locomotive capabilities in minimizing energy cost and remaining safe?*

It is envisioned that such a planner would cause locomotion behaviours to emerge that fully exploit the array of mobility modes on offer, even resulting in behaviours that may be unintuitive to a human operator.

To address this planning problem and validate methodologies developed to solve this problem, it is a prerequisite to develop a real-world RWMR system and develop a low-level control architecture for this system. The design of a highly reconfigurable robot and a kinematics-based motion control architecture are the preliminary contributions of this thesis. A hierarchical sampling-based planning algorithm is then proposed, followed by validation of this algorithm on the real robotic hardware to address the RWMR planning problem.

1.2 Thesis Objectives

The first objective of this thesis is to develop a robotic rover that can traverse unstructured terrain, reconfigure its shape and be able to continue traversing terrain after actuator failures. More specifically, the rover shall be able to reconfigure its geometry to maintain static stability, clamber over obstacles and morph its shape to conform to terrain obstacles. It is desired that the rover move using a variety of different mobility modes that suit various tasks. For example, the rover should be able to adopt an energy efficient driving mode when traversing benign terrain, while adopting a clambering manoeuvre when traversing step obstacles. It is also desired that the rover have failure mode redundancy so that if actuators break or there are sensor pointing constraints imposed that the rover can still achieve its motion goals.

The second objective is to develop a motion model of the newly developed high degree-of-freedom RWMR. This model may be used to define the number of independently controlled motions of the platform given the geometric constraints of the platform and the terrain it is operating on. The model is used to develop relationships between each of the degrees-of-freedom and the relationships between the independent and dependent platform motions. From this model, locomotion modes that are alternatives to simple driving may be realized as well as control policies that reconfigure the vehicle as a function of the terrain and desired workspace motion.

The third objective is to develop a robot motion planner that exploits increased mobility and redundancy while retaining platform safety during a traverse over mapped unstructured terrain. Conventional grid-based and potential field optimal robotic motion planning techniques suffer from the *curse of dimensionality* and are therefore unsuitable for RWMR motion planning. Instead, it is the aim of this thesis to develop a variant of state-of-the-art asymptotically optimal SBP techniques that biases state space search based on the presence of dominant RWMR motions. It is desired to leverage SBP techniques to expose fluid locomotion between different modes that may not necessarily be intuitive to a human operator. The motion planner shall also provide a means to impose motion constraints based

on actuator failures or sensor pointing requirements and still return feasible, efficient and safe motion plans.

1.3 Contributions

This work proposes and validates solutions for increased planetary analogue rover traversability that include the design of a novel wheel-on-leg mobile robot, motion modelling for such a system to develop versatile locomotion modes and a motion planning technique that finds better quality safe paths faster than other state-of-the-art motion planners.

To meet the aforementioned thesis objectives, the design of a wheel-on-leg robot called the Mars Analogue Multi-MOde Traverse Hybrid (MAMMOTH) rover is presented. The MAMMOTH rover system has 16 points of actuation, which it may use to reconfigure its pose and footprint so as to drive or clamber over or around obstacles.

A motion model of the robotic system is formulated based on the recursive kinematic propagation (RKP) framework proposed in [65]. Kinematic analysis results in the formulation of various complex locomotion manoeuvres that the robot may utilize to traverse challenging terrain as well as a novel methodology for synthesizing vision-based sensor data and motion modelling to actively conform to uneven terrain.

With this motion model, a motion planning technique for reconfigurable mobility systems is developed to guide the robot over unstructured terrain. The state-of-the-art Fast Marching Tree (FMT*) asymptotically optimal sampling-based planning algorithm, proposed in [60], is adapted to take advantage of the kinematic structure of reconfigurable rovers. A hierarchical bidirectional planner called Hierarchical Bidirectional Fast Marching Trees, which re-uses FMT* at each of its levels is proposed to reduce computation time and return better quality paths. The planner begins by exploring the workspace of the robot, rapidly finding alternative paths through the sub-dimensional space. These paths are then used to seed exploration in the full state space. Asymptotic optimality guarantees of the FMT* algorithm are shown to be preserved. Better path quality and reduction in computation time are demonstrated through numerical benchmarking against state-of-the-art asymptotically optimal sampling-based planners. Additionally, feedback sampling is used to find when wheel contact constraints should be broken and clambering manoeuvres attempted.

All theory presented is validated using the physical MAMMOTH rover platform. Experimental validation of the rover's kinematic model and its various mobility modes is performed in a lab setting as well as in a Martian analogue environment. Additionally, validation of the motion planner is performed with the MAMMOTH rover following a variety of planned paths within a variety of obstacle-rich environments. These experiments illustrate how the proposed planner fluidly transitions the robot between mobility modes, taking full advantage of the platform's mobility to traverse unstructured terrain. Furthermore, motion constraints are applied to the planning space, illustrating how motion redundancy may be taken advantage of after application of virtual failure modes or operator defined sensor pointing constraints.

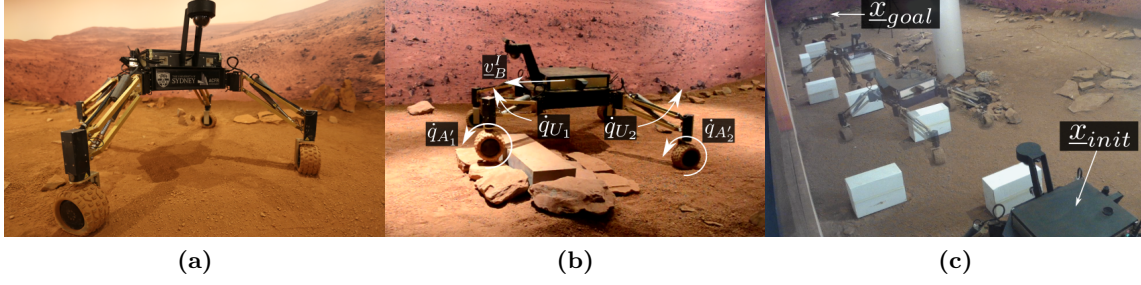


Figure 1.2 – The contributions of this thesis are presented in (a) **Chapter 3**: MAMMOTH Rover System, (b) **Chapter 4**: RWMR Motion Modelling and (c) **Chapter 6**: Hierarchical Sampling-Based Motion Planning for RWMRs. The MAMMOTH rover system and all theoretical contributions are validated in **Chapters 5 and 7**.

1.4 Thesis Structure

In **Chapter 2**, existing planetary rover mobility systems are investigated along with some of the traversability challenges faced by these platforms. Generalized motion modelling techniques for WMRs are reviewed to establish a foundation for modelling of the MAMMOTH rover system. Additionally, motion planning for high degree of freedom systems, hierarchical planning techniques and motion planners that have been validated in the field are explored.

The design of the reconfigurable wheel-on-leg MAMMOTH rover is then presented in **Chapter 3**. The architecture of this novel system’s mechanics, electronics and software is summarised for the reader who is both interested in the engineering implementation of a physical RWMR and to help motivate solutions for its motion modelling and planning.

Chapter 4 presents forward and inverse position and velocity kinematic models for the MAMMOTH rover using the generalized RKP technique for WMRs given in [65]. This model is used to analyze the number of independently controlled degrees of freedom that such a vehicle has and to define a stability metric that quantifies how safe an RWMR is during a traverse. The motion model is manipulated to formulate locomotion modes such as rowing and inchworming that demonstrate how RWMRs may utilize their redundant degrees of freedom for locomotion. Lastly, a novel actively articulated suspension technique that utilizes the MAMMOTH rover motion model and structured-light terrain mapping to conform the robots limbs to unstructured terrain is presented. Validation of the rover’s kinematic model and alternative locomotion modes are presented in **Chapter 5**.

Chapter 6 considers the motion planning problem for reconfigurable rovers and presents the novel HBFMT* algorithm. Performance guarantees, limitations and sensitivity to parameter tuning are all investigated. This algorithm is benchmarked in simulation against state-of-the-art motion planners in a variety of environments composed of unstructured terrain. Validation of the the HBFMT* planner being used to plan a path for the physical MAMMOTH rover is presented in **Chapter 7**. Lastly, conclusions and a summary of future work are provided in **Chapter 8**.

1.5 Publications

- W. Reid, R. Fitch, A. H. Göktoğan, and S. Sukkarieh, “Motion planning for reconfigurable mobile robots using hierarchical fast marching trees,” in *Proc. of WAFR*, 2016.
- W. Reid, A. H. Göktoğan, F. J. Pérez-Grau, and S. Sukkarieh, “Actively articulated suspension for a wheel-on-leg rover operating on a Martian analogue surface,” in *Proc. of IEEE ICRA*, 2016, pp. 5596-5602.
- W. Reid, A. H. Göktoğan, and S. Sukkarieh, “Complex manoeuvres for the wheel-on-leg planetary analogue MAMMOTH rover,” in *Proc. of IAF IAC*, 2017. *Accepted*.
- W. Reid, A. H. Göktoğan, and S. Sukkarieh, “Moving MAMMOTH: Stable motion for a reconfigurable wheel-on-leg rover,” in *Proc. of ARAA ACRA*, 2014. *Nominated for best paper award*.
- W. Reid, R. Fitch, A. H. Göktoğan, and S. Sukkarieh, “Motion planning for a reconfigurable wheeled mobile robot using hierarchical fast marching trees.” *Journal article in preparation*.

1.6 Supplementary Material

Recordings of all physical experiments may be viewed at <https://goo.gl/W9xk0z>, while Table 1.1 provides web addresses for each of the experiment video playlists. A summary of the MAMMOTH rover may be viewed at <https://goo.gl/RHNEYm>.

Table 1.1 – Web addresses for videos of the MAMMOTH rover experiments.

Experiment	URL
Kinematic Model Validation	https://goo.gl/hQuMx3
Actively Articulated Suspension	https://goo.gl/IGIDCi
Alternative Mobility Modes	https://goo.gl/fMiT5v
Plan Following	https://goo.gl/5MC6Ib

Chapter 2

Related Work

In this literature review the history of planetary rover systems is discussed to highlight mobility system design and how autonomous navigation has been used. Kinematic modelling techniques for highly articulated vehicles are reviewed along with how they may be adapted to varying mobility modes and how they may be used within motion planning routines. Lastly, a broad overview of WMR planning is discussed followed by a specific look at how to adapt high degree of freedom sampling-based planning (SBP) to RWMRs.

2.1 Mobile Martian and Lunar Exploration Robots

Mobile robotic agents have been sent to Mars and the Moon over the past five decades to facilitate in-situ exploration of the surface environments of these two extraterrestrial bodies. These platforms have helped shape our understanding of how the Moon and Mars have formed, what surface conditions are like, and helped prepare for future robotic and manned missions. In this section a brief history of these rovers, their mobility and their autonomy is discussed. This section seeks to highlight both the utility of planetary rovers as human proxies and the operational limitations of these systems.

2.1.1 Space Race Rovers

Robotic exploration of the Moon began with the Soviet Union's Lunokhod rovers during the Space Race of the 1960s and 1970s. The Lunokhod 1 and 2 rovers (Figure 2.1a) were each eight-wheeled, solar powered, tub-like vehicles that each had a mass of 840 kg [75]. Their designs originated in the Soviet tank industry and were teleoperated over significant lunar distances. Lunokhod 1 drove 10.5 km over its 10 month lifetime, while Lunokhod 2 drove an impressive 42 km over 5 months. Each vehicle was internally heated by a plutonium heat source. Until 2014, Lunokhod 2 still had the record for furthest distance travelled on the Moon or Mars. The vehicles were operated by a team of five people. The three-second round-trip communications delay impeded teleoperation, however it was still feasible without any on-board mobility autonomy.

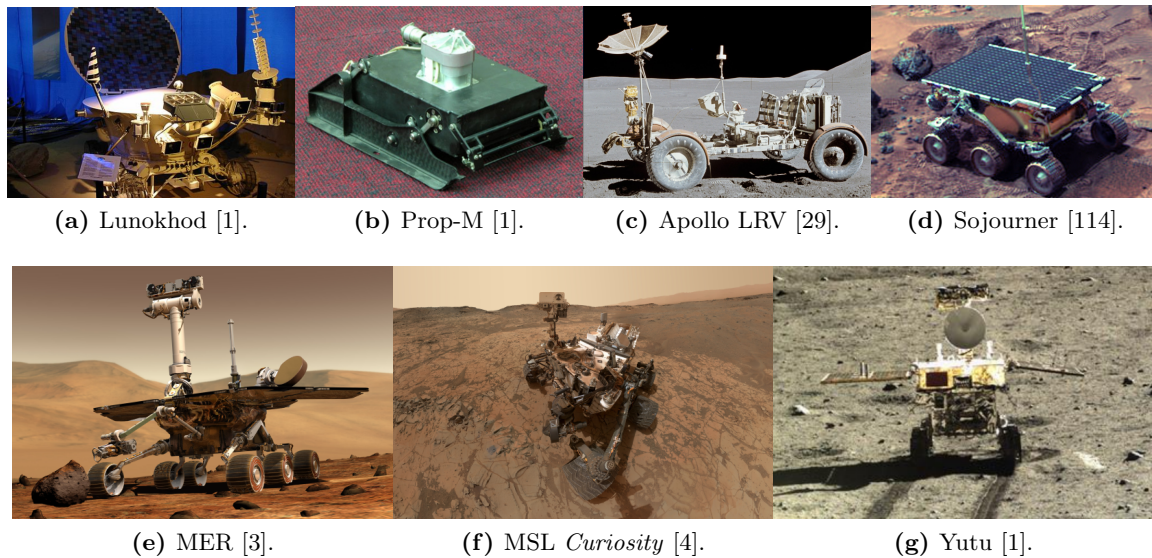


Figure 2.1 – The mobile robotic systems that have been deployed on both the Moon and Mars by space agencies from the Soviet Union, the United States and the People’s Republic of China.

Alexander Kemurdjian, the renowned Soviet tank designer, along with heading the Lunokhod programme, was also responsible for developing the first Martian rover. Substantially more primitive to the Lunokhods, the Prop-M rover, shown in Figure 2.1b, was a 4.5 kg lander-tethered vehicle that propelled itself on rotating skis [1]. The 1971 Soviet Mars 2 and Mars 3 missions each carried a Prop-M rover, however in both cases lander failures resulted in neither rover being operated on the Martian surface. Given the average 20-minute round trip communications delay between Earth and Mars, any Earth operated Martian robot has to have some form of autonomy and not rely on low-latency feedback telecommunication. The Prop-M rover was therefore designed with a reactive navigation system. It was to drive out from the lander, taking soil measurements every metre. It used reactive navigation by detecting collisions with bump sensors and changing heading accordingly.

The Apollo Lunar Roving Vehicle (LRV), shown in Figure 2.1c, was the first lunar wheeled vehicle operated by the United States’ NASA. It was used by a crew of two astronauts from the Apollo 15, 16 and 17 missions between 1971 and 1972. The astronauts used the LRV to increase their scientific investigation zone, and transport a greater number of rocks back to the lunar lander [29]. The range of the rover from the lander was limited by the maximum distance the astronauts could walk back to the lander at any point in an extra-vehicular activity. Unlike the Lunokhods, the LRV could not be teleoperated and had to have a human to drive the vehicle in situ.

The Chinese Yutu rover, a 160 kg solar powered vehicle was landed on the Lunar surface in 2013 [1]. It conducted mobile operations for about a month and became the first rover to operate on the Moon since the Lunokhod’s in the 1970s. The rover was teleoperated and had basic hazard avoidance system.

2.1.2 The NASA Mars Rover Missions

In 1996, the NASA Pathfinder Mars lander carried a microwave oven-sized rover called Sojourner (Figure 2.1d), which was NASA's first robotic extraterrestrial rover. It was also the first rover to perform an in situ investigation of the Martian surface and operated semi-autonomously [114]. Desired waypoints were instructed by the NASA ground control team and the rover autonomously drove and steered towards these commanded positions. Laser strip and camera-based obstacle avoidance techniques were used to steer around hazards in the robot's path. Wheel odometry was used as the primary localization method. It was the first robot to use the rocker-bogie suspension system, an articulated mobility system that promotes wheel contact for each of the rover's six wheels [16]. Sojourner proved the utility of this mobility system, which was adopted in the designs of the three future NASA Mars rovers.

The twin MERs, shown in Figure 2.1e, were named Spirit (MER-A) and Opportunity (MER-B). They arrived on Mars at two separate landing sites in 2004. Each rover was designed to last for 90 Martian solar days (sols). Spirit drove 7.5 km and survived for five Earth years uncovering the previous existence of sulfate rich deposits that indicate the previous existence of soil water [5]. In this article, Arvidson et al. claim that it seems likely that the Home Plate region of Gusev Crater, where Spirit ultimately finished its mission, was once an habitable environment. Ironically, many of these deposits were found due to embedding events where wheels slipped and dug the rover into the soil. Spirit's ultimately failed due to such an embedding event on sol 1871 of its mission. Repeated attempts to drive and 'wiggle' the rovers wheels out of its Home Plate embedding site were ultimately unsuccessful. A view from Spirit's final resting place at the Troy site is shown in Figure 1.1. The Opportunity rover is still conducting surface operations at Endeavour Crater, almost 13 Earth years after arrival with almost 44 km on its odometer. Like Spirit it has discovered evidence of previous existence of the existence of water on the Martian surface that may have been favourable for microorganisms to exist [6].

Autonomy has been demonstrated extensively in both MER missions [17]. Visual odometry has been used to localize the rovers and track slippage based on common landscape features between consecutive image frames. Autonomous Navigation (AutoNav) was also used by both rovers to select motion primitives and avoid hazards to get to operator-defined waypoints. The AutoNav system is able to draw on wheel and visual odometry as well as inertial measurement unit (IMU) and actuator encoder data to monitor the state of the rover during its traverse. Local path planning is performed using the Grid-based Estimation of Surface Traversability Applied to Local Terrain (GESTALT) algorithm [73]. It evaluates the traversability of nearby terrain that has been divided into discrete grid cells. This evaluation is performed based on surface roughness, step variation and inclination metrics. Global path planning is performed by human operators, which is transmitted to the rovers as a series of waypoints to follow. Global path planning is also performed autonomously as part of AutoNav using the Field D* algorithm, first presented in [34]. The algorithm is an anytime grid-based planner that incorporates terrain traversability into traverse cost evaluation. Given the MERs' limited 20 MHz radiation hardened central processing units (CPUs), use of AutoNav makes for slow progress (≈ 10 m/h). Figure 2.2 shows wheel tracks made by the Opportunity rover after an autonomous traverse near Victoria crater.

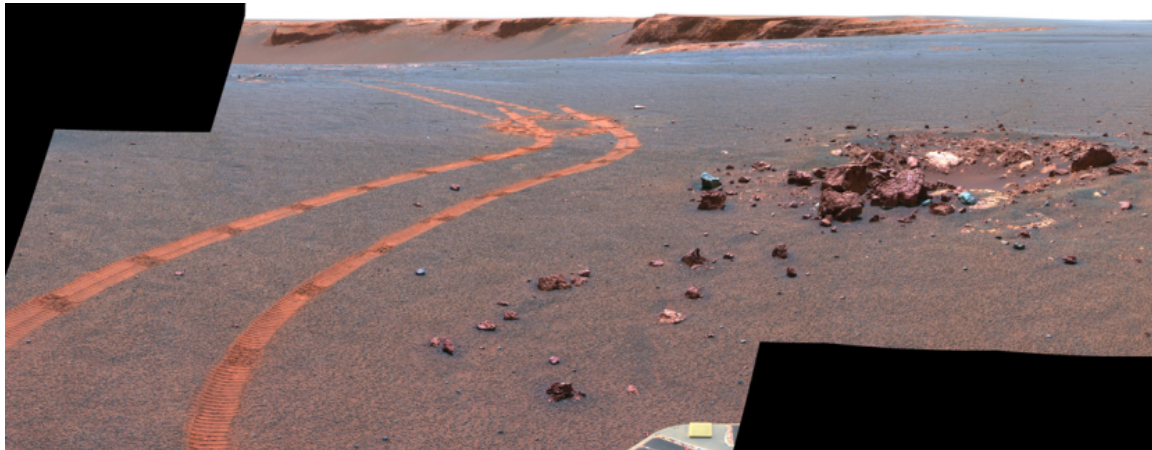


Figure 2.2 – Wheel tracks made by the Opportunity MER during an autonomous drive around a hazardous boulder field [23].

The MSL *Curiosity* is the latest and most sophisticated NASA Mars rover. After landing in 2012, the system has transformed our understanding of the water history of Mars. It has shown that the Gale Crater region is an ancient lake-bed and once had conditions that would have been habitable to ancient lifeforms [45]. The MSL features faster autonomous navigation capabilities than its predecessor MERs with a 200 MHz radiation hardened CPU, however the autonomous navigation used by the rover are very similar to what is used by the MERs.

The MSL has been traversing the base of Mount Sharp that lies within Gusev Crater since it landed in mid-2012. During this traverse it has encountered a series of mobility problems including wheel punctures from crossing sharp rocks and significant slip and skid behaviour while crossing megaripples (areas in the troughs of valleys where wind blown sand has collected and is covered by coarser grains) [4].

As stated in a review of the autonomy of Mars rovers [10], future rovers will need to rely much more on autonomous navigation. The example of a rover used for gathering rock cores for the Mars Sample Return (MSR) mission is cited. Due to thermal cycling of the fuel of the MSR ascent vehicle, the MSR rover would have approximately one year to travel up to 5 km, take varied cores samples and return to the lander. Autonomous planning capabilities that do not rely on the constant supervision of Earth-based controllers would be required to swiftly and safely guide the rover between various sites of interest.

Research into rover autonomy is a mature field and there have been many examples of robotic agents exploring lunar or Martian analogues on Earth. Systems with rocker-bogie mobility systems such as the Field Integrated Design and Operations Field Integrated Design and Operations rover (FIDO) rover [107], Rocky-7 [51] and Athena [18] rovers have been used extensively in testing guidance, navigation and control techniques in preparation for NASA's Mars rover missions. The Rocky-7 vehicle has been commonly used as standard for kinematic and dynamic modelling simulation experiments. Long endurance missions have been conducted using the Zoë [130] and Nomad ([132], [99], [11]) rovers in the Atacama desert. More recently, the Scarecrow rover has been used to test the MSL's mobility system

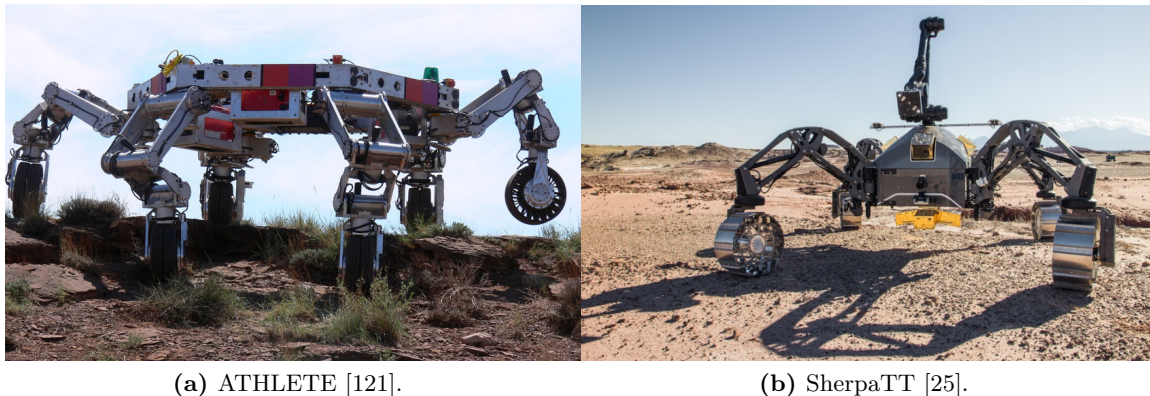


Figure 2.3 – The ATHLETE and SherpaTT wheel-on-leg actively articulated reconfigurable mobile robots. Each of these robots is capable of reconfiguring its limbs to traverse across unstructured terrain using varied mobility modes.

performance in soft sandy and rough rocky terrain [54]. All of these systems utilize mobility systems that conform to the terrain and manage wheel traction by passively reconfiguring articulation angles between internal joints. These platforms have shown to be robust, however susceptible to embedding events and are limited in the types of terrain that they are able to explore. Obstacles such as discontinuous terrain (step obstacles), rough surfaces and significant inclines may result in unsafe operation.

2.1.3 Actively Articulated Planetary Analogue Rovers

Actively articulated mobility systems provide a versatility that addresses these problems. These systems rely on actuators embedded within their structure to reconfigure the vehicle. Early efforts that looked at this problem included the NASA Sample Return Rover (SRR) that was capable of reconfiguring its footprint and centre of mass to prevent the vehicle from tipping over as it traversed undulating terrain as described in [57]. The authors of this work demonstrated a velocity kinematic formulation that was used within a posture optimization routine to find safe configurations as a function of the terrain being traversed.

The NASA All-Terrain Hex-Limbed Extra-Terrestrial Explorer (ATHLETE) rover, shown in Figure 2.3a, is a more complicated example of this type of system as detailed in [134]. It is a six-limbed platform that has 42 actuators with seven actuators per leg. It is termed a ‘wheel-on-leg’ robot given that it has wheels mounted at the end of each leg. The platform may drive over benign terrain, while it may also walk over complex terrain. It can reconfigure itself into a large variety of shapes and sizes depending on its use case. The platform may even split apart and function as two separate three-legged robots. Aside from exploration tasks, it has been demonstrated that the ATHLETE rover is capable of performing construction and haulage tasks that would be required in the construction of a lunar or Martian habitat.

The Sherpa rover is another example of a reconfigurable wheel-on-leg system proposed for lunar exploration. Sherpa is a vehicle with 24 actuated joints, with four legs and six joints

in each leg and is discussed within the context of a heterogeneous team of robotic lunar explorers in [98]. SherpaTT, shown in Figure 2.3b, is the second iteration of the Sherpa robot, which has 20 points of actuation [26]. Even though it has less points of actuation than the first, the reachability of the the robot is significantly increased as it has a knee joint in its leg providing a walking capability and body raise/lower motions that do not change the wheel contact point locations. Kinematic formulations for wheel odometry-based localization and slip monitoring of the Sherpa rover are provided in [55]. The Sherpa system has been used to illustrate some of the many exploration and manipulation tasks that high-DOF reconfigurable wheel-on-leg systems are capable of. In addition to their limbed locomotion, the Sherpa platforms may perform extra manipulation capabilities using a manipulator arm mounted on top of the central chassis. In [26] and [98] the Sherpa platform is used to demonstrate the importance of geometric morphing when it comes not only to crossing varied terrain, but in the manipulation of additional infrastructure such as localization beacons and other robots.

Both the SherpaTT and ATHLETE rovers have independently centre-steered and driven wheels, allowing each platform to drive in any direction. This ability is called *pseudo-omnidirectionality* given that the vehicle must wait for each steering actuator to adopt the correct position before it can be driven along the desired velocity vector. Each of these rovers is also capable of *clambering*, meaning that they may lift one or more legs over obstacles and remain statically stable by virtue of the support polygon formed by the remaining legs in contact with the terrain.

Another significantly simpler yet highly mobile platform is the Scarab rover designed for navigating polar cold traps [12]. This system is a four-wheeled differentially driven platform that can actively adjust the height of pairs of wheels using a linear actuator driven linkage. In [133], an inchworming manoeuvre is proposed to reduce slip on inclines as compared to simple driving. Omni-directional inchworming is implemented for the Sherpa rover as demonstration of its kinematic capabilities [27]. A similar inchworming capability is described in Chapter 4 of this thesis to demonstrate how the kinematic model of the rover may be manipulated to form alternative mobility modes.

Some of the most recent advances in reconfigurable mobile robots have occurred with the DARPA Robotics Challenge (DRC) finals [68]. Competing teams were challenged to build a robotic system that could be used in a variety of manipulation and traversal tasks that a disaster relief robot may have to execute, such as driving a car, opening a valve or traversing an area filled with debris. A total of 24 teams took part in the finals and the resulting robot designs were highly varied, all using many degrees of actuation. The vast majority were humanoid robots that relied on bi-pedal walking for locomotion. Some of the most successful robots however were those that relied on a variety of both legged and wheeled mobility modes. Examples of these robots included the RoboSimian quadruped [53], [63], the CMU Highly Intelligent Mobile Platform (CHIMP) tracked-legged robot [119], the Nimbro Rescue quadruped wheel-on-leg system [110] and the winner of the competition, the bipedal/wheeled Team Korean Advanced Institute of Science and Technology (KAIST) HUBO robot [72].

The RoboSimian and CHIMP robots are reviewed here and in Section 2.3 as both teams have described in detail their motion planning strategies. RoboSimian is a quadruped with

seven degrees of freedom in each leg. To traverse uneven terrain and perform manipulation tasks it used a Rapidly-exploring Random Tree (RRT) sampling-based planner variant as described in [106]. Like the ATHLETE planner, RoboSimian relies on a hierarchy of foot fall planning and sampling-based planning for legged traversals of terrain. When operating on benign terrain the RoboSimian drove around on wheels mounted to its central chassis. The CHIMP robot is another quadruped that relies on walking and tracked locomotion. In this case, a set of planners available within the Open Motion Planning Library (OMPL) [120] was used to execute all planning queries [52]. Both of these robots are high DOF systems that are operated in a semi-autonomous mode. The use of sampling-based planning to formulate low-level motion plans for individual limbs is used to great effect. Both robots placed in the top five of the competition. The challenge with such robots is to formulate global plans that take advantage of the multiple mobility modes on offer. This thesis addresses this challenge by proposing and validating a sampling-based motion planner that generates high quality global motion plans by first exploring the the workspace of an RWMR and using solutions from this search to bias a full state space search. Such a technique enables a higher resolution search of the regions of the full state space that likely contain a high quality path.

2.2 Kinematic Modelling of WMRs

The large set of motions available to RWMRs is central to their increased mobility. In this section kinematic modelling methodologies for these class of robots are reviewed. In order to address the RWMR motion planning problem it is important to understand which system motions may be controlled independently, how each of these motions affects dependent joint motions, and how the application of joint constraints may affect overall motion and the safety of the platform. It is also necessary to understand the efficiency of motion modelling techniques as they may become a computational bottleneck within the motion planning process.

Kinematic models describe the relationship between coordinate frames in a robotic system and in the environment its operating in. Within the robot manipulation arm community kinematics is divided into forward and inverse classes. The forward kinematics (FK) calculates the manipulator arm’s end effector pose as a function of the arm’s joint angles, while inverse kinematics calculates the joint angles as a function of the end effector pose. In wheeled mobile robot kinematic modelling, the terms inverse and forward kinematics are slight misnomers, as the ‘end effector’ coordinate frame may be positioned either at the body centred coordinate frame, at the wheel-contact points, or anywhere on the robot for that matter. A wheeled mobile robot generally has a *tree* structure in which it has a central body or chassis that is the *trunk*, while the attached limbs leading to the wheels are the *branches*. For the purposes of this thesis, the FK is finding the state of a WMR’s wheel contact points as a function of the platform’s body motion and internal joint articulation. The inverse kinematics (IK) is the the process of finding the state of either the body motion or internal joint actuation as a function of the wheel contact state. For each of these cases, the term *position kinematics* relates the positions and rotations of the platform and *velocity kinematics* relates the translational and angular rates of the platform.

Techniques for formulating kinematic models of highly articulated wheeled mobile robots are

varied. *Geometric* kinematic modelling techniques directly look at the geometric structure of a robot and analytically describe the relationship between various coordinate frames positioned along the kinematic tree. This technique is platform-specific and difficult to generalize. Geometric modelling is presented in the original kinematic modelling paper about the MAMMOTH rover [90] and in the WMR kinematic model formulation for the Hylos rover in [44]. *Transformation techniques* develop chains of translation and rotation translations along each of the branches of the WMR kinematic tree. The velocity kinematics is formulated by formulating transformation branches from the body frame to the contact frame and taking the symbolic derivative of this transformation. Such techniques have been utilized in [79] and [122]. *Velocity propagation* techniques are another example of generalized kinematic modelling that propagate velocity transformation along kinematic chains to formulate a velocity kinematic model. These techniques are solely based on the relative positions and rotations between coordinate frames. Such techniques are used in [38], [65] and [123].

2.2.1 Geometric Models

The application of these various techniques to highly articulated wheeled mobile robots is now presented. In [44] the authors formulate a geometric formulation of a velocity kinematic model that is used to characterize the mobility of the wheel-on-leg Hylos rover. Grand et al. [44] present a physical experiment with the Hylos robot in which they apply a balance controller so that the robot keeps a ‘floating-base’ configuration while it traverses irregular terrain. Force feedback in each leg is used to correct the leg joint velocities as the robot traverses over irregular terrain. This decoupling of vehicle posture from joint articulation to meet the terrain profile illustrates some of the significant capabilities of redundant wheel-on-leg systems. Another important aspect of [44] is the mobility analysis performed. The authors use the kinematic model of the system to determine its overall mobility, or the number of dimensions along which it can independently control its motion. This analysis is critical for the formulation of the motion planning and control problem and is discussed further in Section 4.3.3.

2.2.2 Transformation-based Models

Generalized kinematic modelling for wheeled mobile robots (WMRs) was first formalized in [79], in which a transformation matrix approach was used. In [22], a vector-based approach was used for generalized kinematic modelling of WMRs operating on 2D terrains. Additionally, mobility analyses identifying the independent degrees of motion of five different types of planar mobility systems were investigated.

The three dimensional case was studied in [122]. This work developed a transformation technique for formulating the Jacobian relating the wheel and joint motions of a WMR to the body motion. This is done by first generating the homogeneous transformation matrices characterizing the positions and orientations of each wheel frame in relation to an inertial frame. The derivative of the homogeneous transformation matrix expressing the transformation between the body coordinate frame between time steps is then taken and

equated with a general velocity expression for the workspace of the vehicle. The Jacobian relating the velocity of the body frame to the velocity of each of the platform's joints is then formulated from this. Manipulation of the derived kinematic model is used to perform platform localization, slip estimation and determine actuation rates to meet desired platform motion. This technique was applied to a kinematic model of the Rocky-7 rover in [122], and to the Sherpa robot in [55].

The disadvantage in using this technique is the reliance on taking a symbolic derivative of the homogeneous transformation matrix, which can be quite a cumbersome and a computationally inefficient operation. Additionally, when applying wheel-contact constraints, a computationally expensive optimization routine is performed where the error between the terrain and the wheel surface is minimized at each time step.

2.2.3 Velocity Propagation Models

The symbolic derivative of the the homogeneous transformation matrix is addressed in [65] and [123] by re-formulating the generalized kinematic model as a set of differentiable algebraic equations (DAEs). In each of these papers the expression for the Jacobians relating velocities of the robot workspace to the the velocities of contact frames and joint frames are purely functions of the position and rotation transformations between frames. This generalized kinematic model makes the construction of vehicles relatively straight-forward to implement. It is shown in [65] and [123] that a standardized kinematic representation such as Denavit-Hartenberg (DH) parameters, desired body rates and knowledge of local terrain profiles are all that is required for formulating a kinematic model of an articulated WMR. DH parameters, presented in [47], are a standard and compact way of defining transformations between coordinate frames within a mechanism.

An alternative velocity propagation technique is presented in [38] in which twist vectors and adjoint transformations are used to formulate the Jacobian for velocity kinematic solutions. This technique is compact and does not rely on a symbolic derivative. The result of this work is similar to that [65] and [123] in that all that is required to generate the kinematic model is the position and rotation matrices that relate kinematic frames. This work also presents experiments where control techniques using the kinematic formulation are applied to a reconfigurable wheel-on-leg robot operating on flat surfaces.

The work presented presented by Kelly and Seegmiller in [65] presents a generalized WMR kinematic modelling technique called recursive kinematic propagation (RKP). Aside from being an excellent and comprehensive summary of WMR kinematic modelling, this work presents a method that relies on recursive application of the transport theorem along branches of a WMR's kinematic tree to develop expressions for both the velocity and acceleration Jacobians. The technique is simple to implement and may be easily adapted given a standard kinematic representation of a wheeled mobile robot. A discussion of application of kinematic constraints is provided along with simulation of the Rocky-7 rover on uneven undulating terrain. Experimental data validating the kinematic model is also provided in [111] using the Zoë rover. RKP has been chosen as the kinematic modelling framework in this thesis to aid in describing the various complex manoeuvres reconfigurable wheel-on-leg platforms are capable of performing and in developing motion planning techniques for this class

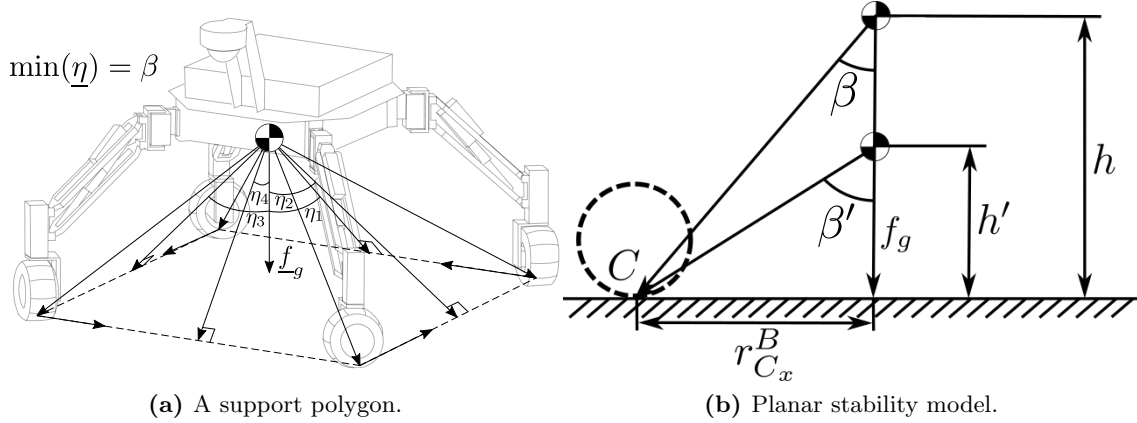


Figure 2.4 – An example support polygon formed by a quadrupedal RWMR is shown in (a). It is shown in (b) that a vehicle on a flat plane with a taller centre of mass has a smaller *force angle* stability measure as defined in [83], however it will have the same *stability margin* measure as defined in [76].

of robots. Seegmiller and Kelly have also expanded on recursive kinematic propagation to formulate dynamic models of wheeled mobile robots in [112]. incorporation of computationally efficient dynamic modelling into planning and control strategies is highly desirable, especially when optimizing for energy consumption of the robot over unstructured terrain.

2.2.4 Stability Models

The most significant motion constraint on rovers is that they remain safe during any traverse and not tip over. Tip-over is a function of body geometry as well as orientation of the body relative to terrain. Given the slow-speed of many of these rovers, static stability metrics are typically used to measure how close the platform is to tipping-over. In all stability metrics the concept of a support polygon is used. The support polygon is the shape that is formed by connecting each of the wheel-ground contact points to their adjacent contact points. A diagram showing an example support polygon of a RWMR is shown in Figure 2.4a. In discussing safe motion for the Hylos rover, the authors of [43] breakdown stability metrics into three separate categories. The *stability margin* developed in [76] is the minimum distance between the projection of the centre of mass of the vehicle and the edges of the support polygon. The *force angle stability margin* presented in [83] takes the minimum angle between the gravity force vector extending from the centre of mass and the vectors extending from the centre of mass that are also orthogonal to the edges of the support polygon. The *energy stability margin* given in [41] and [77] is the minimum amount of energy required to tip a platform over the edges of the support polygon.

The *force angle* stability margin from [83] has been chosen to evaluate tip-over stability in this thesis due to it biasing top-heaviness as compared with the *stability margin* defined in [76], and its relative simplicity of implementation compared to the *energy stability margin* defined in [77]. As shown in Figure 2.4b, the force angle β is smaller with increased centre of mass height. If the centre of mass were simply projected onto the support polygon

on the flat plane, and the distance to the edges from this projected point taken as the metric, height of the centre of mass would not have an effect. Given that RWMRs may change their height quite considerably, any stability measure should be sensitive to top-heaviness.

The *force angle* stability margin has been utilized in the motion control architecture of the reconfigurable wheel-on-leg Sample Return Rover (SRR) in [57].

2.3 Planning for RWMRs

With methodologies for developing wheeled mobile robot kinematics reviewed, the use of such a motion model within a motion planning scheme is now addressed. Given their increased mobility, RWMRs have a large *configuration space*, or set of positions they may occupy within their environment. They have an even larger *state space*, or possible set of positions, velocities, and/or accelerations they may occupy. In this thesis a robot *state* may also be a *configuration*, however a *configuration* is not always a *state*. It is the task of a motion planner to find a *path* through a configuration space or a *trajectory* through a state space that is kinematically and/or dynamically feasible, ensures the safety of the platform, and can successfully drive the robot to its pre-defined goal state. A path or trajectory that fulfils these criteria is defined as *feasible*. The best possible path or trajectory with respect to some cost function is defined as the *optimal*. A review of the major classes of WMR planning is first presented, followed by a description of SBP and its suitability in planning for high degree of freedom systems, and lastly hierarchical planning methods for RWMRs are reviewed in detail as background for the hierarchical planning methodology presented in Chapter 6.

2.3.1 Grid-Based Planning Techniques

Some of the most widely used and robust WMR planning methods are grid-based search techniques. In these planning algorithms the configuration space of the robot is first discretized into a grid. A graph search dynamic programming based method such as Dijkstra's algorithm [31], A* [46] or variants are then used to find the optimal path with respect to some cost function. For planetary exploration, the MER and MSL rovers use a global grid-based planner called Field D* [34]. This algorithm is a variant of A* that may re-plan according changes in perception of the surrounding environment. Unlike conventional grid-based planners that use discrete headings to transition between states, Field D* uses linear interpolation between candidate headings to calculate direct paths between grid cells. Optimized variants of Field D* that reduce the size of the grid-based cost map are described in [23] as part of efforts to implement the planner within the MER AutoNav system. In this work, three separate trials that use the Field D* algorithm on board the MER Opportunity rover are presented. Figure 2.2 is an image showing the wheel tracks made by Opportunity as it executed a path around hazardous rock obstacles planned by Field D*.

To help meet the challenges of autonomous operation of robotic systems in unstructured environments this thesis focuses on the motion planning problem for reconfigurable wheel-on-leg robots. This problem may be considered an instance of the *Piano Mover's Problem*

formulated in [109]. This problem has been defined as polynomial space (PSPACE) hard in Reif’s seminal work [93]. With the introduction of reconfiguration degrees of freedom and multiple locomotion modes, conventional graph search techniques quickly become intractable due to the *curse of dimensionality*, or the exponential increase of the size of the grid to search over.

A recent adaptation of grid-based search to state spaces, which is particularly applicable to differentially constrained rovers, are state lattice techniques [84]. By propagating a grid of states via a discrete set of motion primitives, the technique generates a discrete set of feasible states. Conventional grid-search techniques are then used to search the resulting graph. This technique has been shown to work well in practice for non-holonomic rover systems and directly addresses the non-holonomic constraints present in the MER and MSL Mars rover mobility systems. These rovers do not have steerable centre wheels and therefore are not pseudo-omnidirectional. Differentially constrained Ackerman steering manoeuvres and spot turns are the only driving motions available. In [84] the authors present a field trial with the non-holonomic FIDO Mars analogue rover where global plans through rock fields are generated. Notably, the rover uses the state-lattice formulation to re-plan differentially constrained ‘three point turn’ style manoeuvres when sensor information reveals that rocks are covering paths that it had initially been planned.

2.3.2 Sampling-Based Planning

Over the past two decades, sampling-based planning (SBP) algorithms have become the go-to methodology for solving high-dimensional planning problems. SBP attempts to find the connectivity of an environment by sampling a robot’s state space and using a collision detection routine to determine the validity of samples and connections between samples. SBP avoids the explicit construction of a state space, thereby reducing the complexity of the problem [70]. Ease of implementation, guarantees on probabilistic completeness, and more recently, asymptotic optimality are attractive properties of these techniques. A planner that is probabilistically complete (PC) guarantees that as it nears an infinite number of state samples, it will find a solution to the planning problem with a probability of one. Similarly, an asymptotically optimal (AO) planner guarantees that as it nears an infinite number of samples it will find an optimal solution to the planning problem.

There are two major classes of SBP algorithms: multi-query and single query. Multi-query planners construct a roadmap of a space that can be queried with different initial and goal states multiple times. Single-query planners solve a planning problem for a single initial and goal state pair. Multi-query sampling-based techniques include the probabilistically complete Probabilistic Roadmap (PRM) [64] and its asymptotically optimal variant PRM* [62]. These techniques first sample a batch of random nodes from a space, which are then connected to their nearest neighbours to form a roadmap. This roadmap may then be searched using standard graph-search algorithms such as Dijkstra or A*. PRM* ensures asymptotic optimality by considering edges to nodes within a nearest neighbourhood. The nearest neighbourhood is defined by a distance r_n that is a function of the number of nodes within the current tree, the dimensionality of the planning problem and the volume of the space being explored:

$$r_n \geq 2 \left(\frac{1}{d} \right)^{\frac{1}{d}} \left(\frac{\mu(\mathcal{X}_{free})}{\zeta_d} \right)^{\frac{1}{d}} \left(\frac{\log(n)}{n} \right)^{\frac{1}{d}} \quad (2.1)$$

where ζ_d is the Lebesgue measure of the unit-cost ball of dimension d and $\mu(\mathcal{X}_{free})$ is the Lebesgue measure of the free space. Eq. 2.1 is originally derived in [62]. Much effort has been focused on developing variants of PRMs that address the computational bottleneck of edge construction [32], [33], [103]. Additional improvements include heuristic techniques that lazily check obstacle collisions along promising candidate paths as described for PRM in [19] and for PRM* in [50].

Examples of single-query planning algorithms include the Rapidly-exploring Random Tree (RRT) [71] and its asymptotic optimal counterpart RRT* [62], the Expansive Space Tree (EST) [56] and the Single-query, Bidirectional, Lazy in collision-checking (SBL) [104] algorithms. In the RRT variants a state is sampled from a uniform distribution and a nearest neighbour state within the tree is grown towards the sampled state. On each iteration of the RRT expansion a tree edge growth is attempted and is successful if the edge has no collision with the environment or breaks an internal constraint of the robot model. The algorithm iterates until the tree reaches a goal region or an expiry time is reached. RRT* grows its tree in a similar fashion, however it also attempts to rewire the tree on every iteration, thereby optimizing tree branches relative to some cost function. Tree re-wiring occurs by inspecting the nearest neighbourhood of nodes around a newly added node and checking if a better edge exists than the one just added. The nearest neighbourhood is defined by a distance r_n as defined in Eq. (2.1). RRT* is also an anytime algorithm in that even after a path is found it continues to sample new nodes and attempts to find lower cost paths.

As with the PRM variants, much emphasis has been put on improving the computational efficiency of single query SBP planners. In [2], the AO RRT[#] planner is proposed where tree leaves are characterized based on their potential to be apart of a better solution once an initial solution has been found. Only the leaves that can possibly improve the solution may be re-wired. A similar idea is used in Anytime RRTs [35], Coupled Forest Of Random Engrafting Search Trees (C-FOREST) RRT [81] and Informed RRT* [39] where once the RRT finds a solution only samples within an informed hyper-ellipse are taken. The ellipse represents a bound on the largest possible cost-to-come plus cost-to-go distance of a node. In [39] the authors' major contribution is an analytical method of directly sampling the ellipse as opposed to using rejection sampling.

2.3.3 The Fast Marching Tree Planner

A recently proposed AO SBP is the Fast Marching Tree (FMT*) planner. The algorithm performs a recursive dynamic programming procedure to simultaneously explore and construct a single-query tree through a state space. The algorithm begins by sampling n nodes. Similar to the Fast Marching Method presented in [113], the tree is grown outward using a wavefront expansion, always finding the optimal solution to the current node with respect to the available nodes and the prescribed metric cost function.

In terms of AO SBP computational performance, FMT* and its variants represent the current state-of-the-art. This thesis presents a variant of the FMT* planner and a summary of the algorithm is given here as background to the algorithms presented in Chapter 6. FMT* expansion is performed by maintaining three sets of nodes: a closed, an open and an unvisited set. More concisely, tree $T = (V, E, H, W)$ is composed of tree nodes V , and edges E . The open set of nodes is denoted as H , and W is the unvisited set. Tree T is initialized with n nodes sampled from free space, \mathcal{X}_{free} , in addition to the start node, \underline{x}_{start} and goal node, \underline{x}_{goal} . The algorithm maintains an open set, H , and an unvisited set, W . Initially, \underline{x}_{start} is placed in H , while every other node as well as the goal node is placed in W . The node set, V is initialized containing \underline{x}_{start} , while the edge set E is initialized as empty. The algorithm then performs a recursive dynamic programming procedure that propagates a wavefront of state transitions (tree edges) outwards from the start node.

The expansion procedure starts by taking the lowest cost node \underline{z} within the open set H and finding its nearest neighbourhood Z_{near} of nodes that are also within the unvisited set W (line 7). The Z_{near} neighbourhood is iterated over and connections are attempted between each of these nodes and the closest nodes to them that are also in the open set X_{near} . A connection to a node \underline{x}_{min} in the nearest neighbourhood is attempted only once; if a collision is detected on this candidate connection, no more attempts will be made to connect \underline{x}_{min} to any other node in the open set during this expansion iteration (lines 8-15). This is termed *lazy collision detection*. Once all nearest neighbourhood nodes have been iterated over, all of the newly connected nodes are added to the tree, taken out of the unvisited set and placed in H . The original lowest cost node within the open set is also removed from the open set, never to be re-visited (line 16). Lastly, a new minimum open set node \underline{z} is assigned in preparation for the next expansion iteration (line 19). This expansion procedure continues until the goal node is successfully found, or the open set is empty, implying that no nodes are left to be expanded into and failure is returned. Like PRM* and RRT*, the nearest neighbourhood in FMT* is defined by the radial distance given in Eq. (2.1). Pseudo-code for unidirectional FMT* can be found in [60] and is listed in Algorithm 1.

FMT* is advantageous in that it uses lazy collision detection to reduce the number of expensive collision checking operations, and converges to optimal solutions faster than RRT* and PRM* (shown experimentally in [60] and [118]). Like RRT* and PRM*, FMT* is proven to be asymptotic optimal [60]. The proof may be modified to allow for non-uniform sampling distributions. This property was recently exploited to handle narrow passageways in [138]. The HBFMT* planner presented in this thesis also exploits non-uniform sampling by biasing solutions found from FMT* searches in sub-dimensional spaces. It is shown that the conditions for asymptotic optimality are preserved in Section 6.3.

Additional asymptotically optimal single-query planners that, unlike RRT planners, rely on sampling batches of states prior to building a tree include Batch Informed Tree (BIT*) [40] and Motion Planning using Lower Bounds (MPLB) [102]. Unlike FMT*, BIT* and MPLB are anytime or quasi-anytime, implying that once a solution is found the algorithm refines it and only stops planning when an expiry time is reached. In [102], the authors propose a naive anytime variant of FMT* called Anytime Fast Marching Tree (aFMT*), which simply re-runs FMT* with double the number of nodes during each planning iteration. To

Algorithm 1: Fast Marching Tree (FMT*) [60]

Data: $\mathcal{X}_{free}, \underline{x}_{init}, \underline{x}_{goal}, n$
Result: π, T

```

1  $z \leftarrow \underline{x}_{init}$ 
2  $H \leftarrow \underline{x}_{init}$ 
3  $S \leftarrow \text{SampleFree}(n, \mathcal{X}_{free})$ 
4  $W \leftarrow S \setminus \underline{x}_{init}$ 
5 while  $success = false$  do
6    $H_{new} \leftarrow \emptyset$ 
7    $Z_{near} \leftarrow \text{Near}(V \setminus \{z\}, z, r_n) \cap W$ 
8   for  $\underline{x} \in Z_{near}$  do
9      $X_{near} \leftarrow \text{Near}(V \setminus \{\underline{x}\}, \underline{x}, r_n) \cap H$ 
10     $\underline{x}_{min} \leftarrow \text{argmin}_{\underline{x} \in X_{near}} \{\text{Cost}(\underline{x}, T) + \text{Cost}(\overline{\underline{x}} \underline{x})\}$ 
11    if  $\text{CollisionFree}(\underline{x}_{min}, \underline{x})$  then
12       $E \leftarrow E \cup \{(\underline{x}_{min}, \underline{x})\}$ 
13       $H_{new} \leftarrow H_{new} \cup \{\underline{x}\}$ 
14       $W \leftarrow W \setminus \{\underline{x}\}$ 
15       $T = (V, E)$ 
16      if  $\underline{x}_{min} = \underline{x}_{goal}$  then
17         $success = true$ 
18   $H \leftarrow (H \cup H_{new}) \setminus \{z\}$ 
19  if  $H = \emptyset$  then
20    return Failure
21   $z \leftarrow \text{argmin}_{\underline{x} \in H} \{\text{Cost}(\underline{x}, T)\}$ 
22 return  $T = (V, E)$ 

```

formulate MPLB, the authors modify aFMT* by using a cost-to-come plus heuristic cost-to-go bound on the sampled state space so as not to wastefully explore parts of the space that would not return a better solution. The heuristic used to estimate the cost-to-go is found from a collision-free roadmap of the nodes being planned over.

The BIT* planner proposed in [40] uses a similar anytime batch-based strategy. During each algorithm iteration, k new samples are taken from the state space. Like the A* planner, BIT* employs a heuristic distance to the goal state to manage tree expansion. Candidate expansion states are kept in a priority queue so that they are expanded according to their heuristic cost-to-go plus cost-to-come value. BIT* requires a tree re-wiring procedure in the case that a newly sampled node will improve the cost of existing tree branches. Given the use of cost bounds, both MPLB and BIT* are useful when there is a relatively direct path to the goal state. When this is not the case, the performance of FMT* is faster given that it does not have to repeatedly re-visit nodes and/or perform tree re-wiring procedures. This is empirically demonstrated between BIT* and FMT* in the simulation-based benchmarking experiment presented in Section 6.5.

Bidirectional search is a common method for improving performance of sampling-based plan-

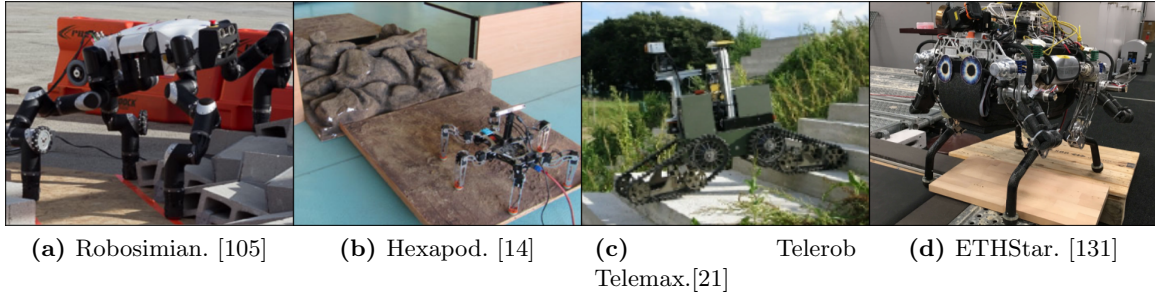


Figure 2.5 – Four examples of robotic systems that use hierarchical planners with SBP at the low-level.

ners and has been demonstrated with Bidirectional Fast Marching Trees (BFMT*) [118], RRT-Connect [69] and bidirectional RRT* [61]. In this thesis, bidirectional search is employed for its performance boost as well and for finding alternative paths, which is a recent popular topic in motion planning [86] and [136].

2.3.4 Hierarchical Planning

SBP has been used extensively to address reconfigurable systems given its applicability to high degree of freedom systems. In work presenting planners for high dimensional legged, tracked or wheel-on-leg robotic systems, SBPs are typically used as part of a hierarchical planner. Typically, a coarse footfall planner at the high level, and then an SBP for low-level leg swing and body posture planning.

The hierarchical motion planner used by NASA’s RoboSimian quadruped (Figure 2.5a) is described in [106] and [105]. Initially, a footfall planner returns a global path over unstructured terrain, a body pose search then returns a central body motion that complies with inverse kinematic solutions constrained by leg stances, and finally an RRT-Connect search returns a path for a swing-leg between footfall states constrained by the central body motion. By planning in a hierarchical manner the authors seek to constrain the enormous configuration space available to RoboSimian, which has 28 points of actuation (7 actuators per leg). Hierarchical search used to constrain configuration space size is also employed in the hexapod planner (Figure 2.5b) described in [14] where a high-level A* search within the robot’s workspace is followed by a lower level guided RRT search of the higher dimensioned robot body pose. In [21] a planner for a reconfigurable tracked vehicle that can traverse very rough terrain is presented. Once again, A* is used at the high level for workspace planning, however the low level RRT* planner is used to plan in the full configuration space of the articulated vehicle.

The ATHLETE rover, shown in Figure 2.3a, with all of its complex mobility also presents a challenging planning, perception and control problem as discussed in [121]. This work presents extensive field trials of the ATHLETE and Tri-ATHLETE systems on lunar analogue terrains. A hierarchical planning system that combines foot fall planning and the SBL sampling-based planner [104] is used to plan safe walking paths over obstacle-littered

terrain. Field testing demonstrated the use of the planner to get around local rock obstacles and also dismount from a mock habitat. The developed planner is able to check kinematic constraints, however is focussed on local leg raising and lowering paths. The sampling-based methodology applied does not consider the full motion of the vehicle over the terrain. Such full body motion planning for ATHLETE is proposed in [49]. This work incorporates actuator torque limits when evaluating motion feasibility and presents experimental results in simulation.

The Hierarchical Bidirectional Fast Marching Trees (HBFMT*) planner presented in this thesis leverages the configuration space size reduction enabled by hierarchical planning, however unlike the hierarchical planners presented so far, HBFMT* operates completely in continuous space by using SBP at each hierarchy level. Ultimately, the performance of discrete grid-based planners is limited by grid resolution. This is not the case when using sampling-based planners at each level of the hierarchy. An additional advantage of using SBP at each level is that dimensionality of the planning problem is easily modified by simply adding or subtracting dimensions from the sampler. In discrete planning subtracting dimensions may be cumbersome in implementation, while adding dimensions quickly becomes computationally intractable. A recent hierarchical planner that uses RRT* at each level is presented in [131], presented at a similar time to HBFMT* [91]. In [131], the three tiered hierarchical planner, which will be denoted as Hierarchical Rapidly Exploring Random Trees (HRRT*), is used to explore the space with successively finer resolution footprint models of the ETHStar quadruped walking robot (Figure 2.5d). The technique lends itself to planning within the coarse global map available to the robot before its traverse with the coarse high level RRT* planner. Local terrain information gleaned from laser sensors is used to iteratively plan over with the lower level RRT* planners using an increasingly higher resolution model of the platform’s footprint. HRRT* is targeted at the anytime planning problem where local sensor information is updated throughout the robot’s traverse, whereas HBFMT* is focussed on reconfigurable systems and exploiting the various modes of traversing a terrain in a computationally efficient manner.

Hierarchical planning is a popular technique within the SBP research community to boost computational efficiency. Major contributions to this field include the Synergistic Combination of Layers of Planning (SyCLOP) algorithm detailed in [85]. This algorithm initially decomposes the workspace of the robot into a discrete grid or set of triangular regions, which is searched using a standard graph search technique. The initial search returns a set of ‘lead’ regions, which are used to guide an SBP through the full state space of the robot. As the SBP exploration progresses, the corresponding discrete regions are updated based on how successful the SBP explores. A similar methodology is used within the Exploration Exploitation Tree (EET) planner proposed in [94]. At the high-level, the robot workspace is explored using a wavefront expansion ‘bubble’ planner that finds collision free ‘tunnels’ through the workspace, which is based on a workspace decomposition planner proposed in [20]. The purpose of the high-level planner is to identify a coarse obstacle free structure of the environment. At the low-level, an RRT-Connect algorithm is used to rapidly perform a bi-directional search of the full state space. The RRT-Connect is guided by choosing frontier ‘bubbles’ that are good candidate regions for extending the plan towards the goal. As the RRT-Connect progresses, the exploitation score of the associated workspace bubble is updated based on how successful it is in progressing through the bubble. The authors

of both [85] and [94] advertise their methodologies as trading off between exploration and exploitation. The low-level workspace plans provide knowledge of the environment that the high level planners may exploit. If the high level planner does not progress through certain regions they both have mechanisms by which exploration of new workspace regions is promoted.

Neither of the algorithms presented in [85] nor [94] maintain the probabilistic completeness guarantees of the sampling-based planners they use at the low-level of their hierarchies. Both sets of authors argue that in practice the reliance on exploitation of the workspace plan results in two to three orders of magnitude improvement in computational efficiency as compared with RRT based search of the full state space. The HBFMT* algorithm presented in this thesis exploits knowledge from a lower level sub-space plan and at the same time takes advantage of the FMT* asymptotic optimality guarantee with biased sampling presented to retain the asymptotic optimality and therefore completeness.

A shortcoming of these hierarchical techniques is seen when the workspace plan does not capture the necessary complexities of moving through a high dimensional space. An example discussed in [94] is when a manipulator has to rotate a bar after removing the bar from a hole and then having to place the other end of the bar in the same hole. The workspace plan is relatively simple given the end effector of the manipulator arm would not have to move very far in the workspace between start and goal workspace locations. The path that captures removing the bar, rotating it and placing it back in the whole would not be embedded within the simple workspace path returned. Fortunately, for the reconfigurable wheeled mobile robot planning problem, most navigation cases have the workspace of the rover's body frame embedded within the full configuration space of the robot.

Similar hierarchical techniques that do not necessarily focus on a workspace decomposition but a more generalized sub-space have been used to address the multi-robot planning problem. In [129] a high dimensional multi-robot path planning problem is addressed using a hierarchical planner. At the high level, a plan is generated for each planar robot. At the lower level all of the high-level paths are combined to form a single space with the dimensionality of a single robot. As this space is explored robot collisions are found, forcing the the dimensionality of the collision region to be expanded to incorporate the separate dimensions of the colliding robots. A similar state decomposition is also used in [9] where obstacle free polytope sub-regions are found within the planar environment and then route planning within each multi-robot configuration space within the associated sub-region.

For the hierarchical planners already discussed, a sub-space is selected by intuition. Automatic sub-space decomposition remains open in general [66]. In [101] the planning problem for a planar mobile dual-arm manipulation robot is presented. A low-dimensional configuration space projection is learned from a succession of paths being run between randomly selected start and goal states in an example environment. An SBP is then run within this projected space. In [128] low-dimensional structure is learned by finding the directions in which the planning cost function varies principally. A set of the most influential principal components together describe basis motions of the robot, which are used to describe the low-dimensional sub-space to plan in. This technique is applied to a manipulation arm as well as a PR2 dual manipulator arm and mobile platform system. Unlike these planners, the HBFMT* planning algorithm presented in this thesis assumes a given state variable

decomposition, but is the first principled hierarchical sampling-based planner that retains performance guarantees.

2.4 Summary

The related work summarized in this chapter highlights the state-of-the-art in planetary exploration rover mobility systems and autonomy. This summary demonstrates that there is a need for RWMRs that may address some of the mobility problems encountered by conventional planetary rover systems. Various generalized kinematic modelling techniques are compared and the velocity propagation RKP technique is chosen as a foundation for kinematic modelling in this thesis. This choice is made due to RKP's relatively straight forward implementation and computational efficiency. Motion planning techniques for highly articulated mobile robots are reviewed with special emphasis placed on SBP techniques due to their effectiveness in addressing the curse of dimensionality. It is shown in the literature that hierarchical methods are an effective way of improving returned path quality and reducing computation time.

Chapter 3

The MAMMOTH Rover System

The MAMMOTH rover is an actively articulated reconfigurable wheeled mobile robot that is used as the example case for a reconfigurable system in this thesis. This chapter presents the design of the MAMMOTH rover hardware and software. The design description is a high-level overview of the novel wheel-on-leg system and robotic system architecture.

The major motivation for the MAMMOTH rover system design and for this thesis is to develop new methods for autonomous high degree of freedom locomotion over unstructured terrain. Therefore, the most significant design consideration of the MAMMOTH rover is to maximize reconfigurability while also providing a reliable means to stay statically stable during operation.

The complete system is shown in Figure 3.1. It is a wheel-on-leg rover with 16 actuated joints capable of multi-modal locomotion. It is composed of five major structural elements: four legs and a body. Each of the four legs have four actuated joints: a hip, a thigh, an ankle joint and a wheel. The body contains all computing and power sub-systems, and is the structural root of the limbed system's kinematic tree. The rover is capable of pseudo-omnidirectional driving by virtue of its four independently steered and driven wheels, footprint reconfiguration through hip actuation, as well as body height, roll and pitch reconfiguration through thigh actuation. Moreover, the rover is capable of lifting one leg off the ground at a time to clamber over obstacles. This chapter provides an overview of the physical MAMMOTH rover system with discussions of the robot's mechanical, electrical, and software sub-systems.

3.1 Mechanical Design

A major mechanical design goal is to maximize platform versatility and reconfigurability to accommodate for unstructured obstacles that may be encountered while traversing a terrain. It is also a major design consideration to develop a modular system that is composed of a common core of actuator and structural elements. Additionally, it is desired that the rover have a size and weight that allow it to be handled by two people during both operation and maintenance. The rover's minimum size shall be small enough to fit into a sports

utility vehicle so that the rover may be transported without the use of a trailer. In terms of performance, it shall be able to operate on slopes up to 20° . It shall also be able to clamber over obstacles that have a maximum height of three times the wheel diameter. The platform should be as modular as possible so as to promote design reuse within a potentially very complex reconfigurable mobility system. Additionally, given the dusty Mars analogue environment the rover operates it shall have a means by which it can protect its electronics from exposure to dust.

3.1.1 The MAMMOTH Rover Leg

The resulting design is a 75 kg platform with four identical wheel-on-leg mobility elements. As shown in Figure 3.2, a single leg is composed of a hip, a thigh, an ankle joint and a wheel. Each MAMMOTH leg weighs 10 kg, with its structure made predominantly of anodised aluminium. Each of the MAMMOTH legs are numbered in a counter-clockwise direction starting at the rover's front left from 1 to 4 as labelled in Figure 3.1.

To enable pseudo-omnidirectional driving, each wheel is independently driven and steered. An actuator is embedded within each wheel hub for driving. The wheels have rubber tyres that are packed with foam. Each wheel is steered via a centre-oriented ankle joint, which means that the joint has a rotary actuator mounted along the central axis of the wheel. To remove any angular position constraints at the ankle joint, while still maintaining electrical connection with the wheel actuator, all signals are passed through a slip ring

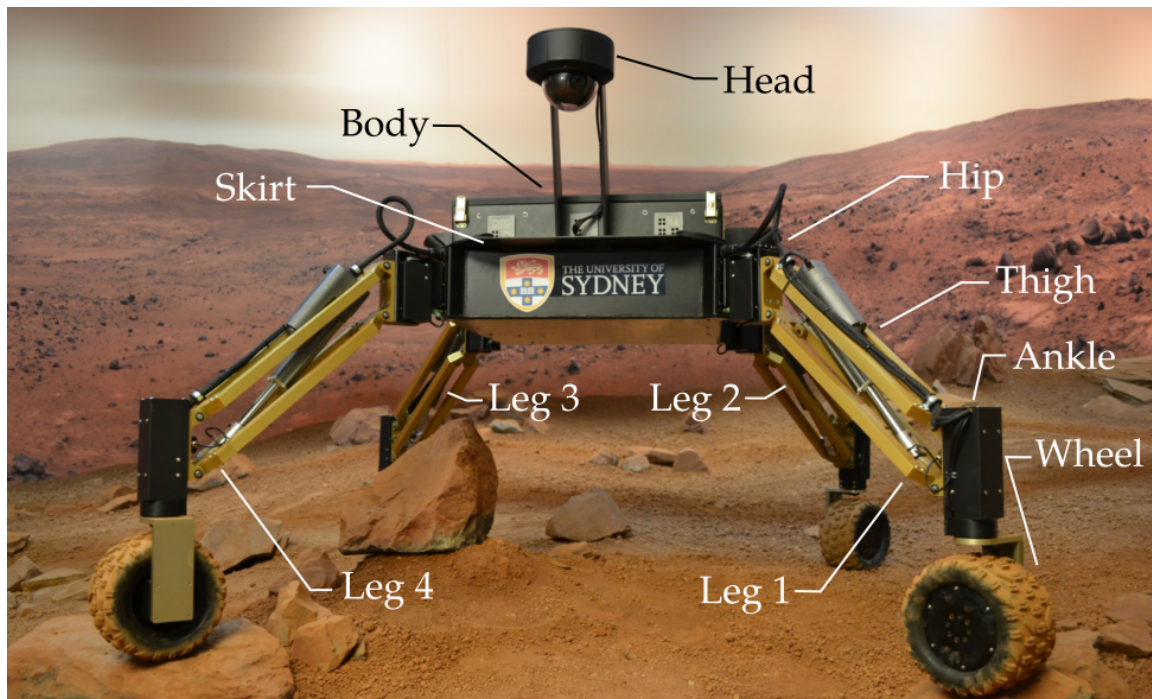


Figure 3.1 – Labelled view of the MAMMOTH rover showing the various structural components and the leg enumeration.

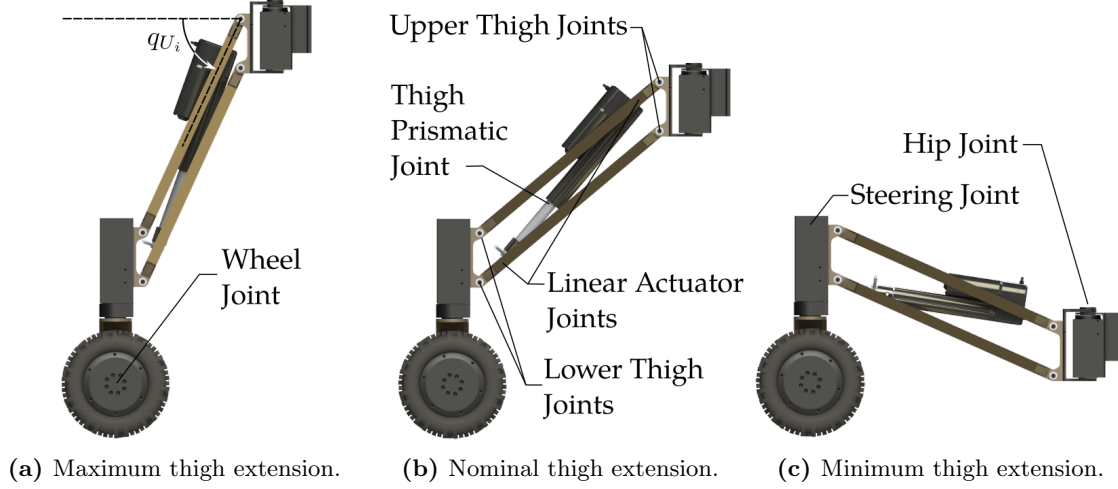


Figure 3.2 – Three labelled views of the MAMMOTH rover leg with varying thigh extensions: (a) $q_{U_i} = 65^\circ$, (b) $q_{U_i} = 40^\circ$, (c) $q_{U_i} = -24^\circ$.

located directly below the ankle actuator. Each wheel and ankle actuator is the same Dynamixel Pro (DXL-Pro) L54 [96] unit that has a nominal torque of 11.8 Nm and peak torque of 24.0 Nm.

The thighs are parallelogram structures shown in detail in Figure 3.2. A single thigh has a total of seven joints: four passive joints at each of the parallelogram corners, two passive joints inside the parallelogram that attach to each end of the linear actuator, and an active prismatic joint inside the linear actuator. The hip and thigh are joined at the upper thigh joints, while the thigh and ankle joint are joined at the lower thigh joints as labelled in Figure 3.2b. At each of these joints a steel shaft secured via circlips is used as the rotational interface.

The role of the thigh is to raise and lower the wheel relative to the body. Any thigh configuration results in the driving axis of the ankle joint being parallel with the driving axis of the hip joint by virtue of the parallel structure. The thigh is actuated via a standard ball-screw linear actuator positioned along the thigh diagonal. By extending the linear actuator's stroke the wheel is lowered with respect to the body, whereas if it is contracted, the wheel is raised. A linear actuator was selected to drive the thigh joints for multiple reasons. The thigh joint needs to be constantly fixed to maintain body height and orientation. A ball-screw linear actuator is preferable for this task given that it is not back-drivable and takes the load of the rover without being powered. Additionally, the linear actuator has 'hard-stop' position limits that ensure that the thighs stay within their allowable angular range. There is a non-linear relationship between linear actuator stroke and thigh angular position as described in Section 4.2. As defined in Figure 3.2, the upper thigh joint q_{U_i} for leg i , has minimum value of -24° and a maximum of 65° . Assuming a flat plane and zero roll and pitch, this thigh angular range allows for a variable body ground clearance between 0.05 m and 0.8 m. The maximum load that can be handled by each linear actuator is 1500 N, which is adequate in providing lifting force in worst case scenarios when the platform is standing on three legs at a roll and pitch of 20° .

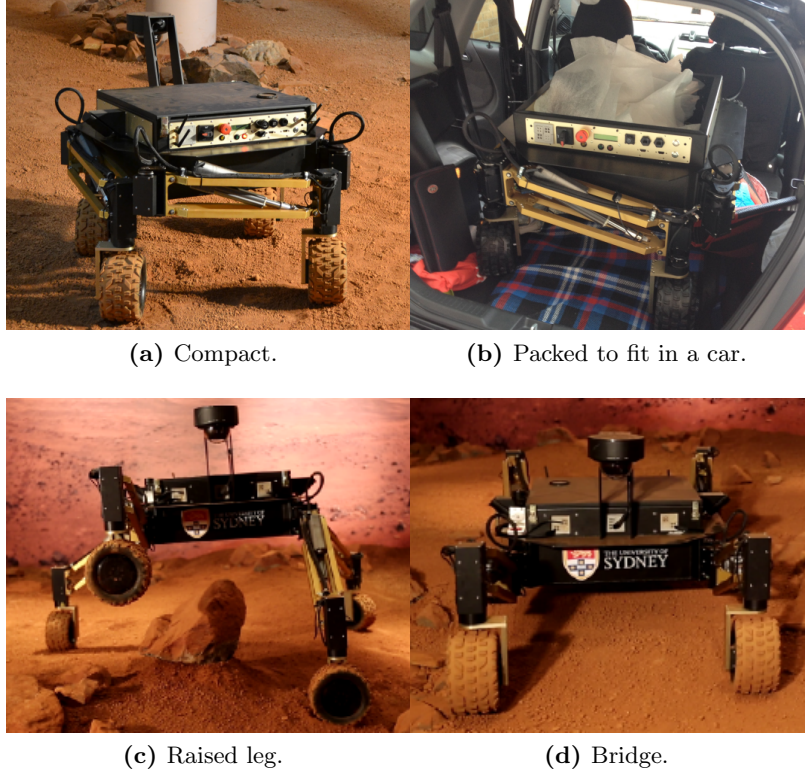


Figure 3.3 – The MAMMOTH rover in a variety of different poses that illustrate its reconfigurability. A compact configuration for driving through small spaces is shown in (a), while the same compact configuration for being transported is shown in (b). A tripod configuration for clambering over obstacles is shown in (c) and a bridge configuration for squeezing between obstacles is shown in (d) .

The hip joints are responsible for footprint reconfiguration. Each individual hip joint q_{H_i} has an angular range of 270° , with a deflection of $\pm 135^\circ$ centred at the $q_{H_i} = 0^\circ$ position shown in Figure 3.1. This results in a wide variety of footprint configurations as shown in Figure 3.3. For example, the robot may morph into a deployed state for nominal stability and reachability (Figure 3.1), a cube-like configuration for compact operations and transport (Figures 3.3a and 3.3b), a tripod state for clambering over obstacles (Figure 3.3c) and a bridge-like configuration (Figure 3.3d) that allows for maximum pitch change and minimal hip torque when driving in the forward direction relative to the body frame. Each hip actuator is a DXL-Pro H54 unit [95] with a nominal torque of 25.3 Nm and a peak torque of 65.0 Nm. The C-bracket interfaces with the hip’s actuator via a shaft with a keyway. The base of the C-bracket slides against a hub affixed to the base of the actuator via a plane bearing.

3.1.2 Structure

The MAMMOTH body is the core structural element of the robot. Made of 4 mm thick anodized aluminium sheeting, the body holds all of the legs in place and houses all of the

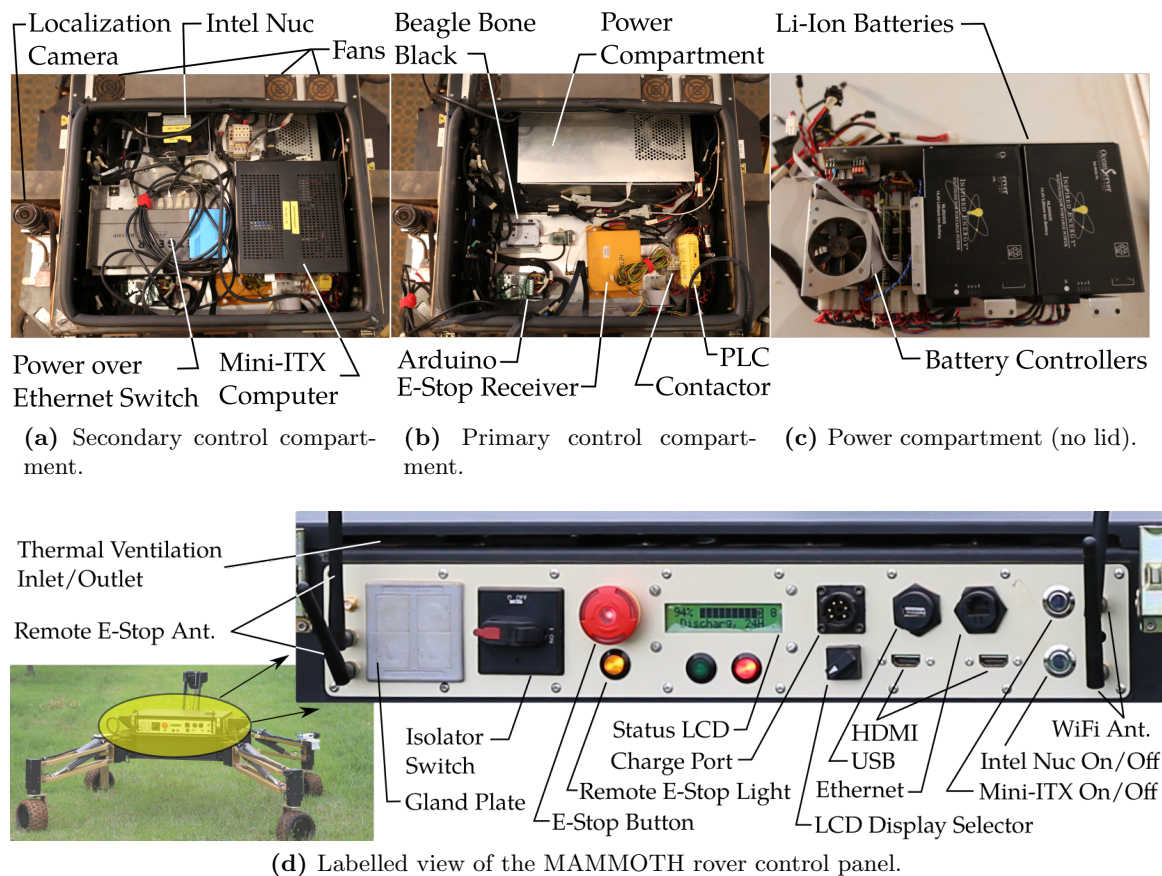


Figure 3.4 – Labelled views of the MAMMOTH rover's body structure.

power and control electronics. The body weighs approximately 30 kg, is a 650 mm by 650 mm by 200 mm box, with each of the legs attached at the corners of the box. A skirt shelf is located around the exterior of the MAMMOTH body, upon which extra sensors or payload may be affixed. The skirting is labelled in Figure 3.1. Gland plates at the front and rear of the body allow for extra data and power lines to be extended to any equipment mounted on the skirt.

A lid on the top of the body secured by four latches may be removed to access the interior of the body. The top of the body is lined with rubber tubing so as to create a dust seal when the lid is secured onto the platform. Three fans are mounted underneath the lid for thermal regulation as shown in Figure 3.4a. Dust filters are placed over each of the intake fans to prevent dust from entering the body. When the lid is on the robot, an a thermal ventilation inlet/outlet is formed between the lid and the body frame as labelled in Figure 3.4d.

The rover's back panel contains various displays, ports and communications equipment and is shown in Figure 3.4d. The back panel alerts an operator of the robot's operational status via an emergency stop (E-STOP) light, a remote E-STOP light, and a liquid crystal display (LCD) display showing the status of the rover's batteries. An operator may turn the rover on or off via the isolator switch or cut power to each of the motors by pressing the

E-STOP button. The operator may also plug into the on board computers via Ethernet, USB or HDMI ports on the back panel. Power buttons for each of the primary controller computers described in Section 3.2.2 with power indication LEDs mounted into the buttons are provided. A charging port for the rover batteries along with WiFi and remote E-STOP antennas are also located on the back panel.

The interior of the body contains three separate compartments: power, core and expandable. The power and core compartments are located at the base of the body interior, while the expandable compartment is on a removable shelf placed on stand-offs above the base layer. Figures 3.4a, 3.4b and 3.4c show the arrangement of each of these compartments. The power system is composed of batteries and battery controller boards and is housed inside its own aluminium box that protects the sensitive batteries and battery controller electronics. All electronics within the primary control compartment is affixed to the body shell via screws, velcro or double-sided tape tabs. All electronics housed on the second layer shelf is affixed with velcro so that it may be easily removed from the shelf.

3.2 Electrical Design

A high level diagram of the MAMMOTH rover's electronics architecture is provided in Figure 3.5. The electronics is divided into the core and expandable sub-systems. The core sub-system encompasses all of the components that are required for basic teleoperation of the vehicle. The core sub-system includes the power, safety, communications, embedded controllers, actuators and actuator sensors sub-systems. The expandable sub-system includes all components related to higher level rover operation, specifically relating to platform autonomy. The expandable sub-system includes the primary controllers, the localization sensors and the power over Ethernet switch within the communications sub-system. Any expansion of the MAMMOTH rover computing or sensing is included within this sub-system.

The electrical design of the MAMMOTH rover is primarily driven by operation duration, safety and expandability requirements. It is required that the rover operate for a full day of experiments or demonstrations, which is defined as 6 hours with a 10% duty cycle on all actuators being driven at nominal torque. Additionally, an emergency stop safety system that turns off all power to driven joints that can be activated both remotely and in local proximity to the robot is required. Given the high torque actuators and the danger these present to an individual operating the MAMMOTH rover, this requirement is critical. It is also desired that the electronics design should be adaptable so that new sensors or computers may be easily added to the existing infrastructure.

3.2.1 Power and Safety

The MAMMOTH rover system is powered by 12×95 Wh lithium ion (Li-Ion) batteries [58]. The raw battery voltage is 14.4 V, which is regulated to 24 V, 12 V, 5 V and 3.3 V busses. The 24 V bus powers all actuators, the 12 V bus powers computing and communications, while 5 V is reserved for embedded computing and sensor power. An E-STOP system

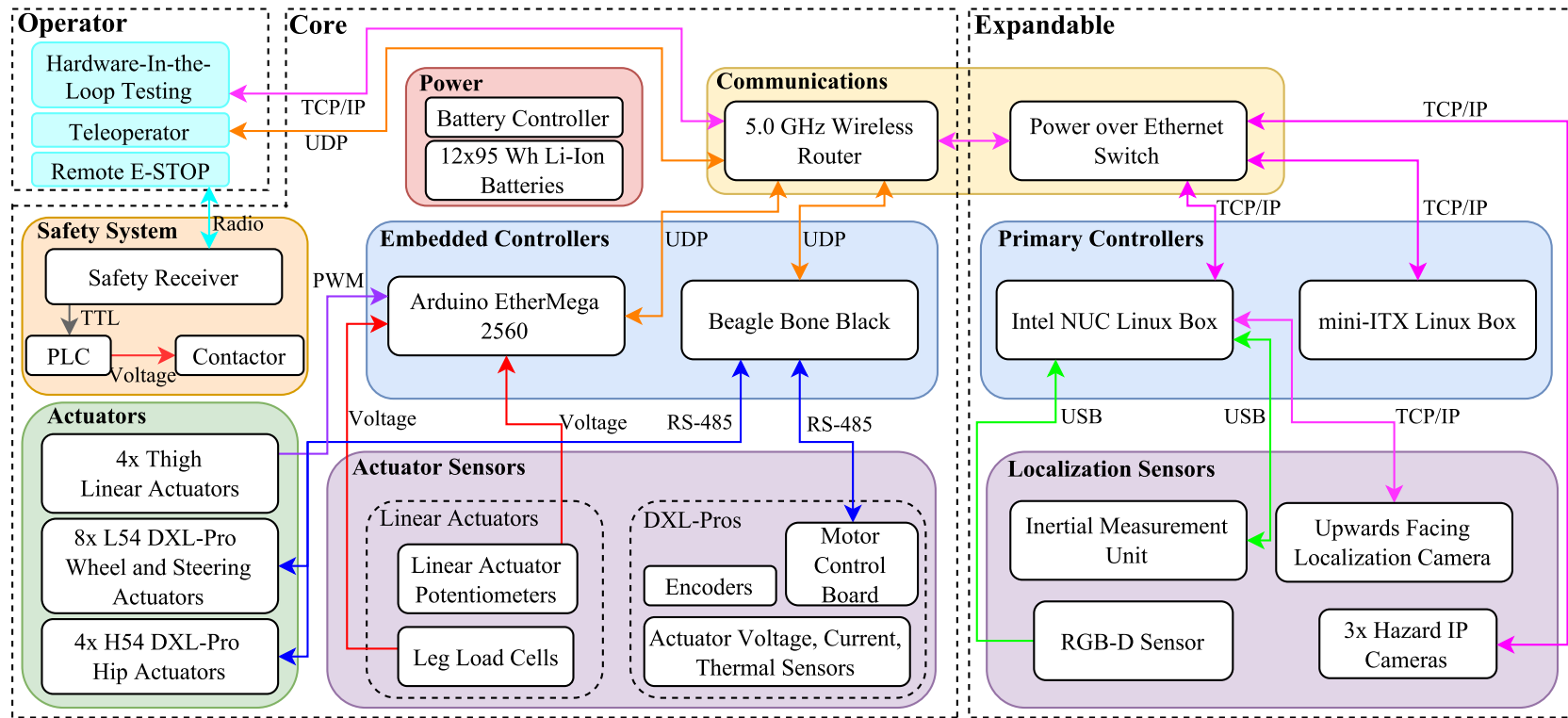


Figure 3.5 – System architecture of the MAMMOTH rover.

composed of a programmable logic controller (PLC) based safety controller with a safety rated wireless receiver is installed. When either the E-STOP button on the back of the rover or the E-STOP button on a wireless remote is pressed, a contactor is triggered by the PLC, which cuts power to all actuators. The emergency stop system may be activated via a button on the robot's control panel or via a remote control. Additional power supply lines are made available for each of the power buses so that new computers or sensors may be added to the existing platform. A Power over Ethernet (PoE) switch is also included within the communications system to allow extra sensors to be powered over their data transfer lines.

3.2.2 Computing and Communications

Each computer or microcontroller on board the rover will be denoted as a processor. There are five processors on board the rover: the safety system PLC, the embedded processors that manage the low level mobility control and the primary processors that are responsible for platform autonomy.

The embedded processors are the Arduino EtherMega and the BeagleBone Black (BBB). The Arduino EtherMega [37] is an Arduino compatible board based on the Atmel ATmega2560 microcontroller [8] with a WIZnet W5100 Ethernet chip [135]. Two dual H-bridge motor driver shields [87] are used with the Arduino to provide up to 24 A continuous current to a single linear actuator motor. The BBB is a credit card sized computer that has a 1 GHz ARM Cortex-A8 processor with 512MB DDR3 RAM running the Debian operating system (OS) [13]. The BBB interfaces with the MAMMOTH local area network via an Ethernet connection, while it interfaces with the DXL-Pro actuators over an RS-485 bus via a USB to RS-485 adaptor.

The two primary processors are the mini-ITX Linux Box and the Intel NUC Linux Box. Higher level mobility tasks such as planning and path following are managed by the KTQM77 mini-ITX [67] 3.1 GHz computer running the Ubuntu 16.04 Long Term Support (LTS) OS. This computer is used to run ROS to manage communications between software nodes operating on controllers within the expandable sub-system. Additionally, an Intel NUC computer [59] with a 2.6 GHz processor, 16 GB RAM, an i5 processor and running Ubuntu 16.04 LTS is responsible for all localization computation.

All of the MAMMOTH embedded and primary processors interface via a distributed communications architecture via TCP/IP and UDP over an on-board local area network. A wireless router hosts a 5 GHz WiFi network and is also an Ethernet switch for the core electronics system. This wireless router is connected via Ethernet to a PoE switch, which acts as the main networking interface for the expandable system of components. The rover's wireless network is used to teleoperate the robot as well as run hardware-in-the-loop (HIL) testing. All Internet Protocol (IP) cameras including the hazard cameras and localization camera are both powered and interface with the various system controllers via the power over Ethernet switch. It is straight forward to include new computing or sensors into this distributed system. They may be included as either as a sub-system within the MAMMOTH rover or as an external agent by interfacing via UDP or TCP/IP over the local area network.

The computational capability of the MAMMOTH rover with its distributed architecture and multi-core higher level processors is significantly greater than what would be typically found on a planetary rover. Planetary rovers such as the MERs or the MSL have single core, low power processors that run at clockspeeds an order of magnitude slower than the MAMMOTH rover's processors [4].

More powerful processors have been chosen due to their off-the-shelf availability as well as their necessity for the planning algorithms developed in Chapter 6. To consider the adoption of such planning algorithms on board an actual wheel-on-leg planetary rover, processors of a similar performance on the MAMMOTH rover would be required. In the development of a simulated planetary exploration system, it is assumed that such computational capabilities would be available by the time an actual wheel-on-leg planetary rover were to be developed.

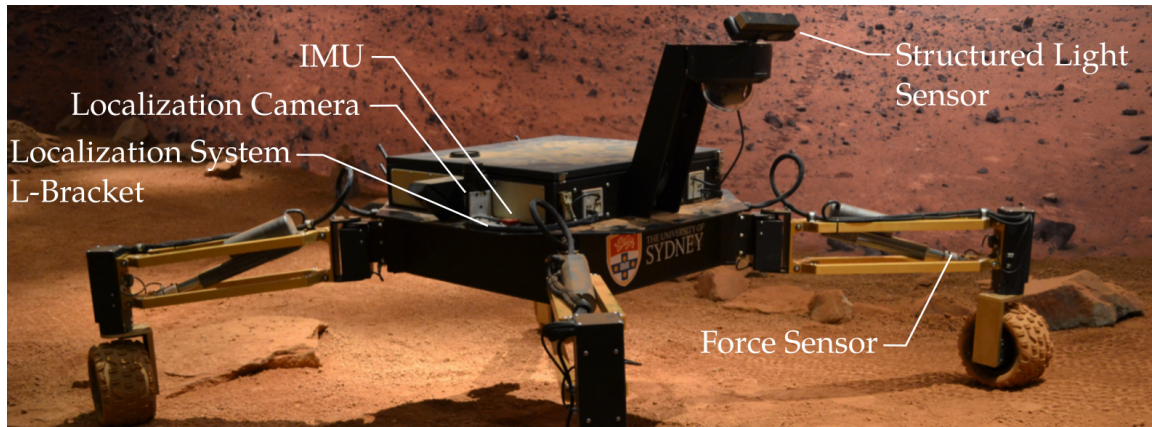
3.2.3 Actuation

All rotary actuators are DXL-Pro series servo units complete with a direct current (DC) motor, an harmonic gear head and a programmable motor control board. Data is transferred to and from the actuators via an RS-485 bus. The bus is arranged in a star topology with the rotary actuators in each leg daisy-chained. Control and data management is managed by the two embedded processors.

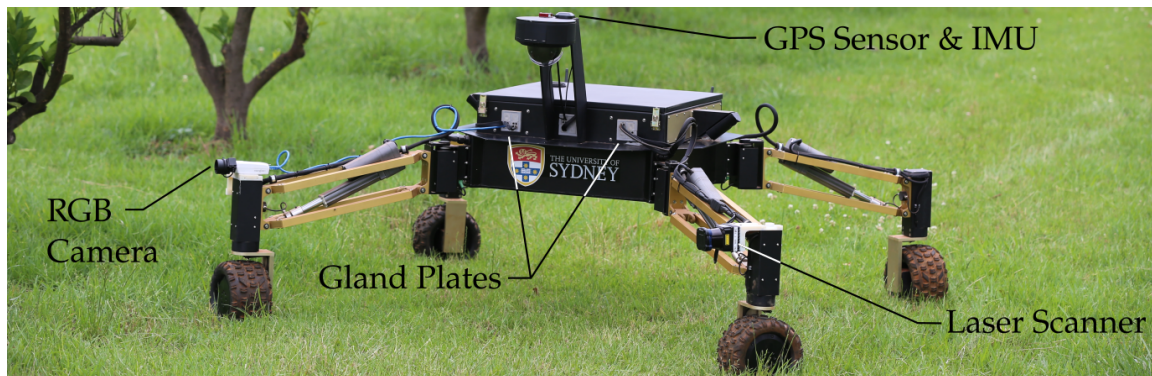
Each of the DXL-Pro actuators may be controlled in different modes based on their use. The control mode is programmed via the actuator unit's programmable motor control board. The wheel drive actuators are set to 'velocity mode'. A desired rotational speed is given to each of the wheel drive actuators and it may continuously rotate. A proportional integral (PI) controller is implemented for both velocity and torque control in this mode. The ankle and hip actuators are controlled in 'extended position mode' so that position, velocity and torque may be controlled. A proportional (P) controller is implemented for the position controller, and PI controllers for the velocity and torque. In this mode the actuators are free to continuously rotate, which is desirable given that there is no constraint prohibiting continuous rotation at the ankle joint. Position limits are programmed to the hip actuators so that they may not attempt to drive past their limits of $\pm 135^\circ$. All controller gains are manually tuned by the operator.

The hip actuators have a power rating of 100 W, while the ankle and wheel actuators have power ratings of 50 W. All of the rotary actuators operate at 24 V. At their stall torque the hip actuators pull 14 A, while the ankle and wheel actuators pull 5.6 A at stall. During nominal continuous operation the wheel actuators are driven at 22.1 RPM, which is equivalent to a rover forward speed of 0.23 m/s. At this rate the actuators deliver up to 4.8 Nm and pull 1.9 A.

The linear actuators are controlled via a PWM input signal sourced from the Arduino EtherMega microcontroller. Analogue voltages are read by the Arduino EtherMega from the linear actuator potentiometer and force sensors. A PI position controller is used to control the linear actuator position. The linear actuator's velocity may not be controlled as well with the rotary actuators. Non-linear velocity control is observed when operating below 10 mm/s, which is compounded when the linear actuators are loaded. The linear



(a) A typical Mars Yard sensor configuration.



(b) The MAMMOTH rover with an extra camera, GPS and laser scanner mounted to it. This is an example of the expandability of the MAMMOTH rover given its distributed communications architecture, power over Ethernet and extra power lines.

Figure 3.6 – Two separate MAMMOTH rover sensor configurations.

actuators have a maximum speed of 11.2 mm/s. Given this behaviour, it is desired that the linear actuators operate near their maximum speed during all operations. When variable velocity is required, a PWM limit is included within the proportional controller so that the controller may not issue a velocity command larger than the desired velocity. Such an implementation was used in the actively articulated suspension experiments presented in Section 5.3.

3.2.4 Sensing

For localization, a VN-200 IMU [127] and Blackfly red, green and blue (RGB) camera unit [36] are mounted on the MAMMOTH rover body's skirt. The two sensors are both attached to a single L-bracket aluminium frame so that the geometric relationship between the two components are held constant. These two sensors make up the localization system that is used to formulate a 6-DOF pose solution for the vehicle. The localization system components are shown in Figure 3.6a. The camera is used to sight fiducial markers mounted

at known locations on the ceiling of the environment (see Figure 3.8b). The fiducial marker locations are used to triangulate the position and orientation of the rover. The orientation result may be fused with the output of the angular position, rates and accelerations from the IMU to generate a the 6-DOF pose. Details regarding the localization system implementation may be found in [125].

Hazard cameras mounted on the rover body skirt as well as on the rover head allow an operator to observe the rover’s surroundings during teleoperation. Each of the hazard cameras is an IP camera that is powered via Ethernet. An ASUS Xtion Pro red, green, blue and depth (RGB-D) sensor [7] is also mounted on the rover’s mast. This sensor is used to perform localization of the robot and mapping of the terrain in front of the vehicle. This sensor is used within the actively articulated suspension control scheme described in Section 4.5. This sensor is powered via a USB connection to the mini-ITX primary controller computer. The sensor is mounted on top of the rover’s mast as shown in Figure 3.6a at 42° relative to the body horizontal plane. In this configuration the field of view of the sensor does not see the rover’s front legs when their respective hip angles are equal to 0° . This is preferable so that features from the rover’s legs are do not interfere with the mapping and localization algorithms that use the RGB-D sensor data. This configuration is used for the actively articulated suspension experiments described in Section 5.3.

Absolute angular encoders are mounted at each driven joint, which is used to formulate localization of the entire kinematic structure. Each DXL-Pro rotary actuator has a 1000 pulse incremental encoder before gear reduction and a 4096 step absolute encoder after gear reduction. Each linear actuator has an analogue potentiometer that returns a voltage between 0 V and 5 V as a function of the position of the linear actuator stroke. Current and thermal sensors monitor each rotary actuator, while current sensors are also included inside each linear actuator. A ± 1 kN force sensor is mounted on the end of each linear actuator to indicate the terrain contact status of each wheel.

3.2.5 Expandability

In November 2014 the MAMMOTH rover was demonstrated on a farm, shown in Figure 3.6b and <https://goo.gl/NmM54G>. The rover is shown with an extra laser scanner on leg 1, a camera on leg 4, and a GPS sensor and IMU attached to its head. Each of these sensors used either the spare power lines or power over Ethernet for power and an Ethernet connection to the local area network for data interface. Gland plates are positioned on the front (Figure 3.6b) and the back (Figure 3.4d) to allow new data and power lines to be extended into the interior of the robot’s body while maintaining a dust seal. Mounting points on the rover’s head, ankle joints and body skirt are provided to mount new sensors as demonstrated in 3.6b.

3.3 Software Design

The MAMMOTH rover software is a set of components developed using a model-based design methodology. A high level diagram of the various software models is shown in

Figure 3.7. The MAMMOTH software components are run on four separate processors in a *distributed* system. As shown in Figure 3.5 the four processors are divided into the embedded and primary controllers. The embedded controllers are those that are on when the mobility system is on, and are required for basic operation of the mobility system. Primitive operation of the robot, such as teleoperation may be performed with just the embedded controllers activated. The primary controllers handle all of the higher level tasks related to platform autonomy: planning, environment sensing, localization and path following control.

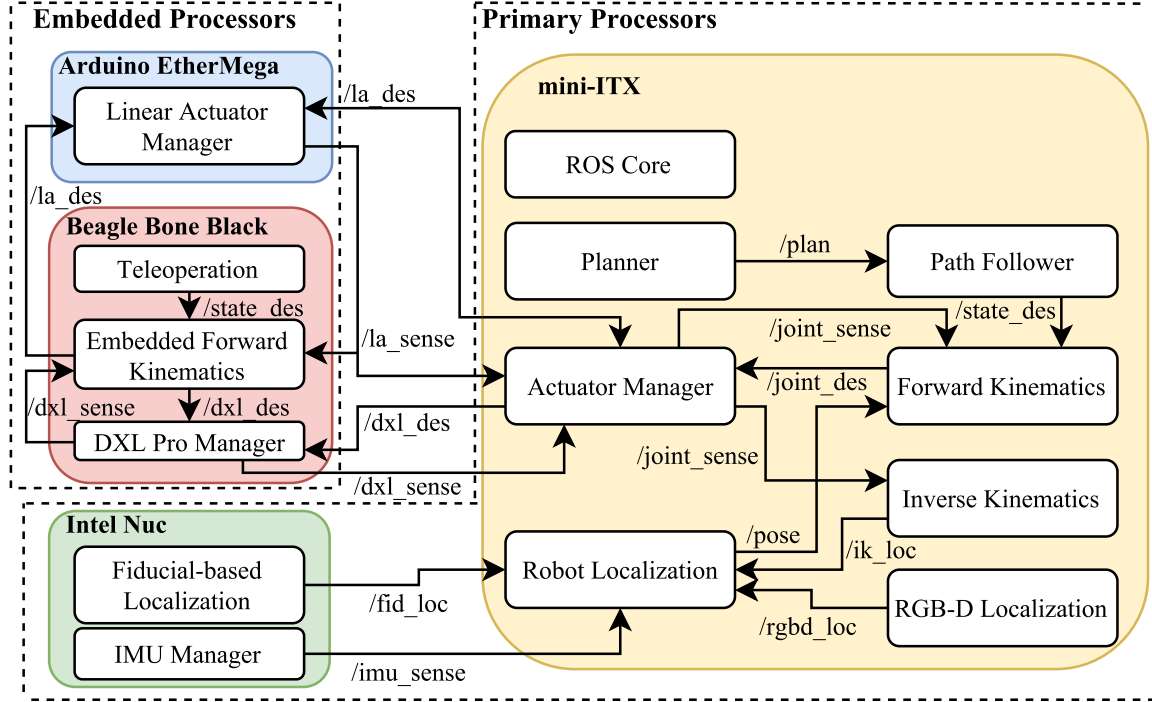


Figure 3.7 – The software components running on the four MAMMOTH rover processors and the interfaces between the components.

3.3.1 Low Level Mobility Control Software

The *DXL-Pro Manager* software component runs on board the BBB and manages writing to and reading from all of the 12 rotary actuators over an RS-485 interface. Actuator read and write operations are performed using the DXL-Pro software development kit (SDK) [97]. Simultaneous read and write commands are sent to all twelve actuators at 20 Hz. Write commands send a desired position and velocity to each actuator. Read commands are separated into two types of commands: mobility read and status read. Mobility read commands read the position and velocity of each actuator, while status read commands read the temperature, current and voltage of each actuator. A cycle of read and write commands occurs every 250 ms. Every cycle there are two write, two mobility read commands and a single status read command.

The *Linear Actuator Manager* component running on the Arduino EtherMega embedded controller manages the linear actuators. The microcontroller runs position controllers for

each of the linear actuators. Along with managing linear actuator control, the Arduino is responsible for reading linear actuator current, voltage, encoder and force sensors in each MAMMOTH leg. The Arduino communicates by accepting position and velocity commands and broadcasts sensor data all via UDP at 10 Hz.

If teleoperation is to be run, a forward kinematics model of the MAMMOTH rover is run on board the BBB. This *Embedded Forward Kinematics* software component accepts desired positions and velocities for the rover's pose and joint coordinates. The component accepts UDP pose velocity and joint position and velocity commands. Depending on the desired motion, it outputs the corresponding joint actuator desired position and velocity commands, which are then forwarded to the *DXL-Pro Manager* and *Linear Actuator Manager* software components.

3.3.2 High Level Mobility Control Software

The software running on the primary controllers is used for autonomous operation of the MAMMOTH rover. The ROS distributed communications architecture [89] is used for communication between all software nodes running on board the mini-ITX Linux Box. All software components that are run within ROS are denoted as nodes. Each of the ROS nodes as well as software components running on board the Intel NUC are as seen in Figure 3.7 are now described.

The *Actuator Manager* node forwards desired position and velocity commands to the embedded processors and also receives actuator sensor data from them. This component is responsible for packaging all data related to the rover's actuators.

The *Forward Kinematics* node transforms all actuator and localization data into a rover state solution. It is also responsible for all kinematic calculations associated with complex manoeuvres and actively articulated suspension. This component uses recursive kinematic propagation [65] to perform kinematic calculations. Dependent joints such as the thigh, ankle and wheel drive joints have their desired positions and rates calculated by this node so that desired full state motions may be met. The kinematic models used by this node are described in Chapter 4. The *Inverse Kinematics* node performs the necessary velocity kinematic calculations to form an odometry-based localization solution. The kinematic model used in this case is given in Eq. 4.20.

The *Fiducial-based Localization* node interfaces with the localization RGB camera and performs all computer vision tasks, outputting a 6-DOF localization solution. The *IMU Manager* component packages and forwards the angular position, velocity and acceleration of the rover. These two localization solutions are fused with the solution from the *Inverse Kinematics* node within the third-party *robot_localization* ROS node [78]. The *RGB-D Localization* node is performs localization and mapping given the relative displacement of terrain features. It is also used to identify the height of the terrain that a wheel is about to drive over so that the associated may be actively articulated to conform to the terrain while preserving a desired body velocity. Actively articulated suspension is discussed in Section 4.5.

All path planning, as described in Chapter 6, is implemented within the Open Motion Planning Library (OMPL) [120]. The ROS node *Path Planner* is a wrapper that calls OMPL. This wrapper outputs generated paths to the *Path Follower* node. All path planning for the robot assumes a-priori knowledge of the environment and its obstacles. This information is loaded into the OMPL environment during software initialization. The planner node is used within all plan following experiments described in Chapter 7.

The *Path Follower* node implements a pure pursuit style path following control system [30]. It receives an array of waypoints, with each waypoint being a state within an 11-DOF MAMMOTH configuration space. Initially the module attempts to drive the robot towards the first waypoint in the plan. A waypoint is reached when it is less than an ϵ tolerance from the waypoint. This tolerance is tuned by the operator. The difference between a waypoint and the robot's current location is calculated using the sum of the norms of waypoint and current location sub-space values. This difference is used as the input to a proportional integral derivative (PID) controller to determine output control commands. Control commands are issued in the form of desired rates in each of the platform's controllable state dimensions. If a specific command exceeds the maximum allowable rate for the associated state dimension the command is set to its saturation limit and the remaining control commands are linearly scaled. Once a waypoint is reached the next waypoint in the plan is adopted as the next goal. When the last waypoint in the plan is reached the robot is instructed to stop moving.

3.4 The Mars Lab

The Mars Lab is a 17 m by 7 m space that is a simulated Martian terrain where experiments with the MAMMOTH rover are run. The space is located within a museum exhibit at the Powerhouse Museum in Sydney, Australia. It is a recreation of the Columbia Hills Complex in the Gusev Crater region of Mars, near where the Spirit Mars Exploration Rover landed in 2004. The variety of terrain available to traverse including the flat sandy highway region, the undulating mounds region, and the crater all labelled in Figure 3.8a.

An array of fiducial markers are mounted on the ceiling of the Mars Lab. There are a total of 152 distinct fiducial markers that have all been surveyed relative to an inertial coordinate frame I shown in Figure 3.8c. The x_I axis points along the long terrain axis of the yard, while the y_I axis points along the short terrain axis of the yard. The z_I axis points upwards. The localization system is a custom designed solution developed at the Australian Centre for Field Robotics (ACFR) and is discussed in [125].

The fiducial-based localization system is used as a ground truth for the alternative mobility mode experiments discussed in Section 5.2 and the actively articulated suspension experiments discussed in Section 5.3. The localization system is used in the control loop for the plan following experiments described in Chapter 7. In an actual planetary exploration mission, such a localization system would not be available hence its use as only ground truth in the alternate mobility mode and actively articulated suspension experiments. The major contribution of the path following experiments is the validation of the planning techniques and not localization. For this reason, localization is assumed. In a real planetary

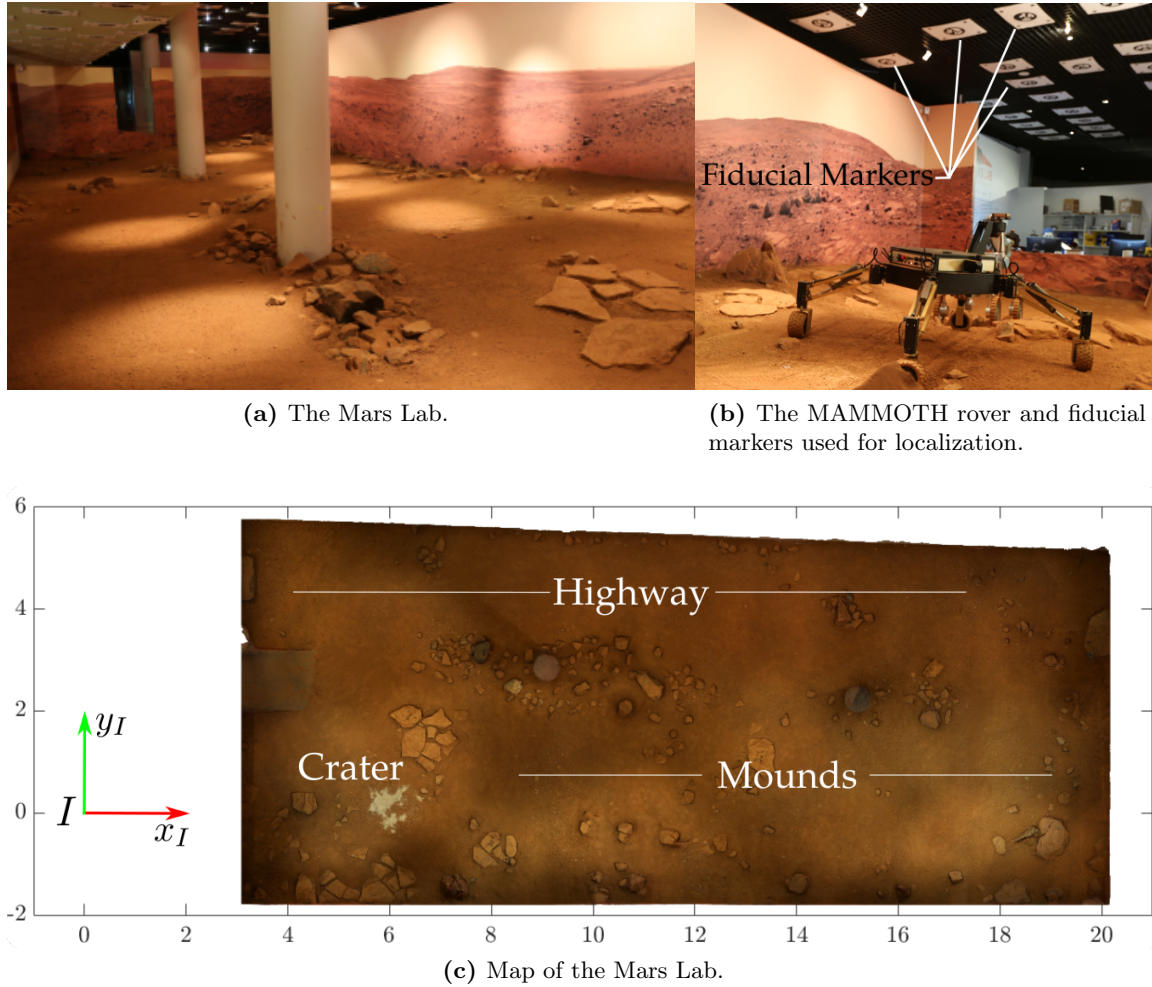


Figure 3.8 – The Mars Lab, a 17 m by 7 m Martian analogue at the Sydney Powerhouse Museum. The coordinate system used is relative to the inertial coordinate frame I .

exploration mission global localization and mapping would have to be solved via alternative means.

All experiments included in this thesis excluding the kinematic model validation experiments presented in Section 5.1 are performed on the Mars Yard. The actively articulated suspension experiments presented in Section 5.3 are performed on the mounds region, while the alternative mobility mode experiments from Section 5.2 and the plan following experiments from Chapter 7 are performed on the highway region. Extra obstacles are positioned on this section of terrain. Each of the obstacles is mapped relative to the inertial frame.

Chapter 4

RWMR Motion Modelling

This chapter presents the motion model of the MAMMOTH rover system. The rover's kinematic model is presented using the recursive kinematic propagation (RKP) technique described in [65] and is used in all numerical and physical experiments throughout the thesis. Novel locomotion modes and capabilities are presented to illustrate the versatility of actively articulated RWMRs. Additionally, a novel method for actively articulated suspension that aids in traversing uneven terrain is presented. This technique uses the vehicle kinematics and a terrain point cloud from a structured light sensor to generate desired wheel-ground contact trajectories.

4.1 Kinematic Notation

Throughout this thesis a vector notation similar to that used in [65] is used. A generic expression for a vector is written with an underline such as \underline{u} . A position vector is denoted by ${}^{F_0}\underline{r}_{F_1}^{F_1}$, which is a 3×1 vector that is the position of frame F_2 relative to frame F_1 , expressed in frame F_0 . The expression $\underline{r}_{F_2}^{F_1}$ is used as short-hand for ${}^{F_1}\underline{r}_{F_2}^{F_1}$. The rotation from frame F_2 to F_1 is given by the 3×3 orthonormal rotation matrix $R_{F_2}^{F_1}$. The vector $\underline{v}_{F_1}^{F_2}$ denotes the translational velocity of frame F_1 with respect to F_2 , while $\underline{\omega}_{F_1}^{F_2}$ denotes the angular velocity of frame F_1 with respect to F_2 . To denote the size of matrices and vectors, products as subscripts are used. Examples include $I_{g \times g}$, which denotes a $g \times g$ identity matrix, while $\underline{0}_{g \times h}$ denotes a $g \times h$ zero vector or matrix. To define a sub-region of a matrix, the indexing notation $M(g : l, :)$ is used. The g through l rows of the M matrix are selected, while all columns of M are selected, represented by the ':' operator. All matrices and vectors are one-base indexed. To express a cross product in matrix form the skew-symmetric operation is used on a vector. The skew-symmetric operation performed

on the position vector \underline{r} is given by $[\underline{r}]^\times = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix}$.

Planning and control of the MAMMOTH rover is dependent on the kinematic model of the multi-link vehicle. This model relates the motion of the robot's body frame, its actuated joints and its wheel contact points relative to the inertial frame. Figure 4.1 shows the full

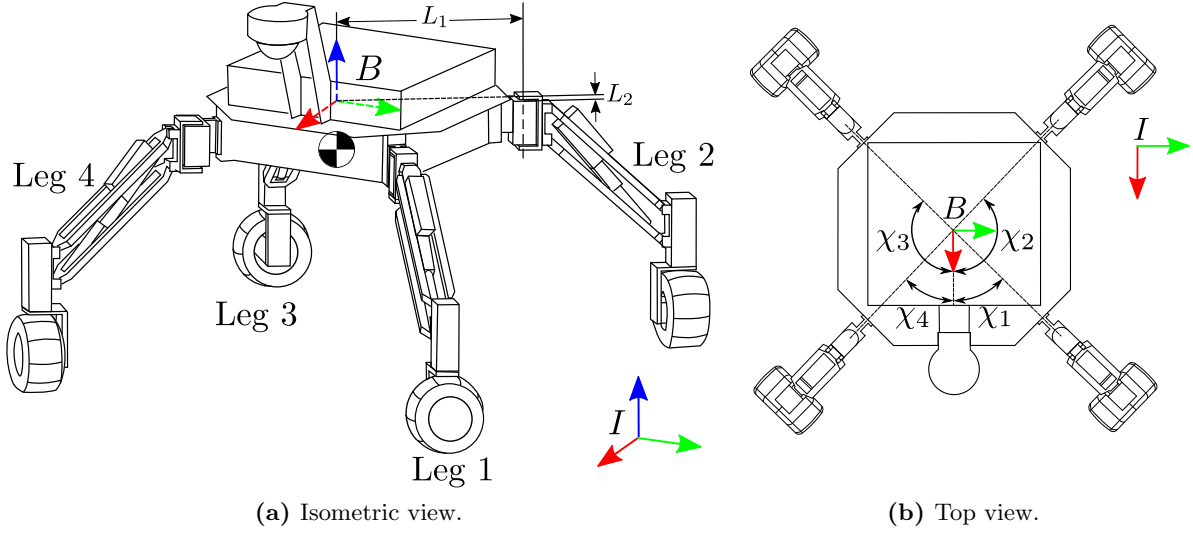


Figure 4.1 – Labeled view of the MAMMOTH rover, its main components, dimensions and coordinate frames.

MAMMOTH rover structure, the major structural elements of the platform and the inertial and body coordinate frames. As seen in this figure, the central structural element of the robot is the box-like body where all computing and power systems are housed. The body coordinate frame, B , is located at the geometric centre of this box. The robot moves relative to frame I , which is placed in inertial space. Coordinate frames are presented in Figures with different coloured axes: the x -axis is red, the y -axis is green and the z -axis is blue. The position of the centre of mass (\odot) is a function of the configuration of the robot. The technique used to calculate the position of the centre of mass, \underline{r}_{\odot}^I , is given in Appendix A.3.

The MAMMOTH rover has a total of 16 actuators, with 4 actuators located in each leg. A leg is shown in Figure 4.2 and is the main mobility element of the robot. A leg is referred to by its index $i = \{1...4\}$, labelled in Figure 4.1. The coordinate frames used within the leg structure to define the motion of the robot are shown in Figure 4.2. Starting from the top of the leg at the hip frame, H_i , the hip joint rotates the leg about H_i 's z -axis. This motion contributes to changing the footprint size of the robot. The thigh is a parallel structure that forms the major structural element of the leg. A linear actuator is embedded within the thigh and is positioned along a diagonal of the parallel structure. The upper thigh coordinate frame, U_i , is located at the thigh joint closest to the body, while the lower thigh coordinate frame, Lo_i , is at the thigh joint located closest to the wheel. The thigh coordinate frame, Th_i , is located at the interface between the linear actuator housing and the linear actuator ram. The geometric relationship between U_i and Th_i is detailed in Appendix A.5. The wheel is steered using a centre-steered configuration; the steering frame, S_i , is aligned with the centre of the wheel. The wheel frame, A_i , is located at the wheel centre and the wheel is driven about the z -axis of this frame. The A_i frame is fixed with respect to the wheel rotation. A second wheel frame, A'_i , is also located at the centre of the wheel, however this frame rotates with the wheel. Lastly, a contact frame C_i is located at the interface between the wheel surface and the terrain surface. Table 4.1

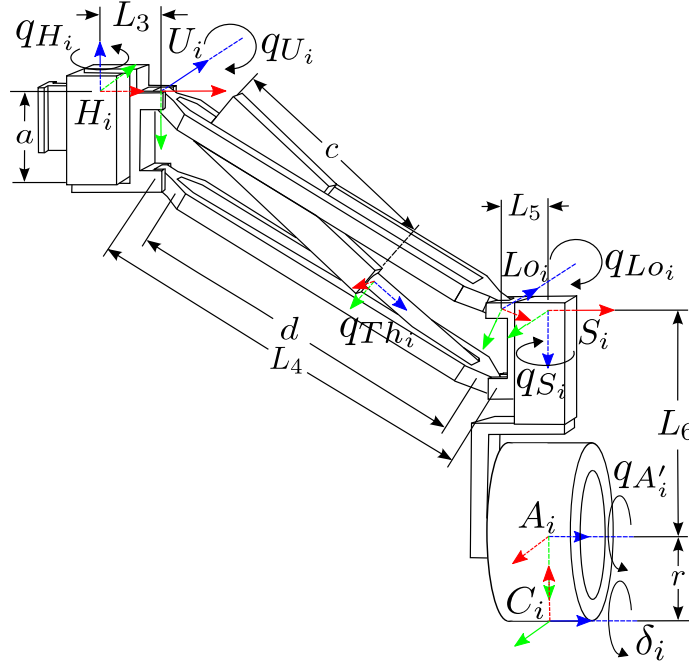


Figure 4.2 – Labeled view of the MAMMOTH rover leg i showing its coordinate frames, dimensions and joint angular position variables.

presents a standardized formulation of the relationship between each of the MAMMOTH rover's coordinate frames in the form of Denavit-Hartenberg (DH) parameters from the body frame onwards.

The wheel/terrain contact point is assumed to be a point, however in reality it is a contact patch given the compliance of each of the wheels' tyres. Depending on the terrain structure and wheel compliance, there may be multiple wheel-ground contacts for a single wheel. Diagrams showing a compliant wheel and an approximate contact patch are shown in Figure 4.3. Given terrain interaction, the contact points may slip in the longitudinal driving direction as well as the lateral direction. The 6×1 vector $\underline{\sigma}_i$ describes the translational and rotational slip along each of these directions. In this thesis it is assumed that there is a single contact point and that the wheel tyre is not compliant so that there is a constant wheel radius between A_i and C_i . Ideal wheel/terrain interactions are also assumed, making $\underline{\sigma}_i$ a zero vector. In all motion modelling it is also assumed that the contact angle between the surface of each wheel and the terrain is zero. This assumption is made given the inability to adequately sense the wheel-ground contact angle with the MAMMOTH rover's sensing configuration.

When considering how each of the 16 actuators affect the total motion of the vehicle, the independently controllable motions are first identified. The robot may translate along each axis of B relative to I , resulting in a translational position defined by x_B^I , y_B^I and z_B^I . The body's angular rotation relative to I is given by the Euler angles: roll (ϕ), pitch (θ) and yaw (ψ). The Euler angles used in this thesis follow the 'XYZ' rotation convention as summarised in Appendix A.4. The six position and rotation variables describing the relationship between I and B comprise the six DOF pose vector $\underline{p} = [x_B^I \ y_B^I \ z_B^I \ \phi \ \theta \ \psi]^T$. Additional

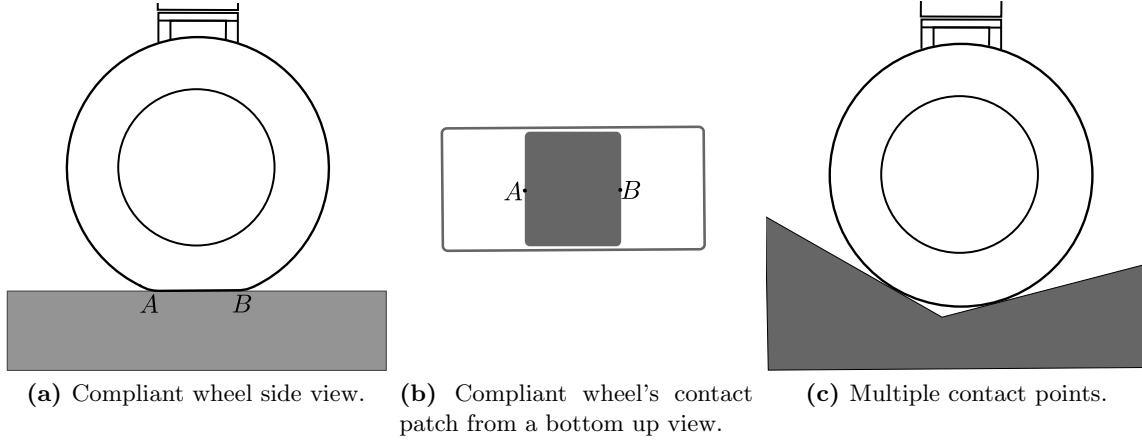


Figure 4.3 – In reality, a wheel is compliant causing a contact patch as opposed to a single contact point. In (a) a compliant wheel with a compressed base is shown. In (b) a bottom up-view of the contact patch of the compliant wheel is shown. In (c) the case of multiple contact points on uneven terrain is highlighted.

independently controlled motions are performed by the four hip joints. These four rotations are denoted by $\underline{q}_H = [q_{H_1} \dots q_{H_4}]^T$. In total, the MAMMOTH rover has 10 DOF when it has all four wheels in contact with the ground and there is no slip. This is expanded to 11 DOF when a single leg is lifted off the ground and the upper thigh rotation q_{U_i} of the raised leg i may also be independently controlled. An analysis of the MAMMOTH rover's mobility index is performed in Section 4.3.3.

Two variables that describe dependent motions within the platform's kinematics are the upper and lower thigh joint rotations. For leg i , the upper thigh rotation is denoted by q_{U_i} , while the lower thigh joint rotation is denoted by q_{Lo_i} . Given the parallel structure of the MAMMOTH rover thighs, the equality constraint $q_{U_i} = -q_{Lo_i}$ is enforced. The steering joint position, q_{S_i} , changes the driving direction of wheel i , while the wheel drive joint position, q_{A_i} , tracks the rotational displacement of wheel i . Lastly, the wheel-ground contact angle, δ_i , is dependent on both the rover configuration and the terrain profile. The variables that describe leg i 's articulation angular positions are stacked into the leg articulation vector $\underline{\gamma}_i = [q_{H_i} \ q_{U_i} \ q_{S_i} \ \delta_i]^T$. The full articulation vector for the whole robot is composed of the four individual leg articulation vectors: $\underline{\gamma} = [\underline{\gamma}_1^T \dots \underline{\gamma}_4^T]^T$.

4.2 Position Kinematics

The position kinematics establishes relative position and orientation relationships between the coordinate frames within a robot's structure. This section defines how the position kinematic model of the MAMMOTH rover is formulated. Initially, the position kinematics are used to define positions of coordinate frames along each MAMMOTH rover leg. A model of static stability of the MAMMOTH rover is also defined. This model is used as a safety metric during all rover operations.

Table 4.1 – Summary of the kinematic constants and variables within the kinematic branches that go along each MAMMOTH rover leg. Presented are the variables associated with the transformation from I to B , the DH parameters describing the transformations along leg i 's links and values associated with each of the kinematic constants. The \underline{d}_{k-1}^k vectors relate to the velocity kinematics of the rover and are described in Section 4.3.1. The Rate column lists the joint rate that controls the rotational rate of the associated frame.

Inertial to Body Frame Transformation						
k	Frame	R_k^{k-1}	r_k^{k-1}	\underline{d}_k^{k-1}	Rate	
1	I	-	-	-	-	
2	B_a	$I_{3 \times 3}$	$\begin{bmatrix} x_B^I \\ y_B^I \\ z_B^I \end{bmatrix}$	$\begin{bmatrix} 1 \\ \underline{0}_{1 \times 5} \end{bmatrix}$	-	
3	B_b	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix}$	$\underline{0}_{3 \times 1}$	$\begin{bmatrix} \underline{0}_{3 \times 1} \\ 1 \\ \underline{0}_{2 \times 1} \end{bmatrix}$	$\dot{\phi}$	
4	B_c	$\begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$	$\underline{0}_{3 \times 1}$	$\begin{bmatrix} \underline{0}_{4 \times 1} \\ 1 \\ 0 \end{bmatrix}$	$\dot{\theta}$	
5	B	$\begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\underline{0}_{3 \times 1}$	$\begin{bmatrix} \underline{0}_{5 \times 1} \\ 1 \end{bmatrix}$	$\dot{\psi}$	
Leg DH Parameters						
k	Frame	θ_{dh}	d_{dh}	a_{dh}	α_{dh}	Rate
6	H_i	χ_i	$-L_2$	L_1	0	0
7	U_i	q_{H_i}	0	L_3	-90°	\dot{q}_{H_i}
8	Lo_i	q_{U_i}	0	L_4	0	\dot{q}_{U_i}
9	S_i	$-q_{U_i}$	0	L_5	-90°	$-\dot{q}_{U_i}$
10	A_i	$q_{S_i} + 90^\circ$	L_6	0	90°	0
11	C_i	$\delta_i - 90^\circ$	0	$-L_7$	0	$\dot{\delta}_i$
Kinematic Constants						
	$[\circ]$	$[\text{mm}]$		$[\text{mm}]$		$[\text{mm}]$
χ_1	45	L_1	442.0	L_5	50.2	a 90.0
χ_2	135	L_2	65.0	L_6	238.5	c 257.0
χ_3	-135	L_3	63.5	L_7	100.0	d 337.5
χ_4	-45	L_4	431.5			

4.2.1 The RWMR Kinematic Tree

A position kinematic model of a wheeled mobile robot expresses the position of each of the wheel-ground contact points as a function of the body pose and the actuator angular positions. The model for a RWMR rover is expressed as a stacked vector of four *kinematic*

branches, where each branch is a sequence of translations and rotations moving along each leg from the I frame to the C_i frame:

$$\underline{r}_C^I = [(\underline{r}_{C_1}^I)^T \quad \dots \quad (\underline{r}_{C_4}^I)^T]^T. \quad (4.1)$$

Together, these branches form a *kinematic tree*, with I at the root of the tree and each of the C_i frames at the tree leaves. Eq. (4.1) assumes that all four wheels are in contact with the terrain. If leg i is not in contact with the terrain, the associated $\underline{r}_{C_i}^I$ is removed from \underline{r}_C^I .

For any kinematic branch the root frame is denoted by the index j , while the leaf frame is the n index, where $j \leq n$. A kinematic branch is formulated from the relative position vectors and rotation matrices between each adjacent robot coordinate frame. An adjacent frame translation is defined as \underline{r}_k^{k-1} , while an adjacent frame rotation is defined as R_k^{k-1} . In these expressions, the index k is a generic coordinate frame index. The expression for a position vector between a kinematic branch's root frame and a leaf frame may be formulated by recursively applying the following expression:

$$\underline{r}_n^{j+k-1} = \underline{r}_{j+k}^{j+k-1} + R_{j+k}^{j+k-1} \underline{r}_n^{j+k}. \quad (4.2)$$

In this expression the index k starts at 1 and counts upwards on each recursion. Recursion stops when the \underline{r}_n^{j+k} term expresses the relationship between adjacent frames. A simple example showing the use of Eq. 4.2 may be found in Appendix A.1.1. Similarly, rotation matrices may be solved for by recursively applying

$$R_n^{j+k-1} = R_{j+k}^{j+k-1} R_n^{j+k}. \quad (4.3)$$

Recursion once again stops when R_n^{j+k} is a rotation matrix between adjacent frames.

If it is desired to express a position vector in a different coordinate frame, say $j-1$, and it is currently expressed in j , then we may simply multiply the rotation matrix going from j to $j-1$ by the position vector:

$${}^{j-1}r_n^j = R_j^{j-1} r_n^j. \quad (4.4)$$

The inverse of a rotation matrix is equal to its transpose. Therefore, if we want to transform the position vector ${}^{j-1}r_n^j$ to be expressed in the j frame we may perform the following operation:

$$r_n^j = R_{j-1}^j {}^{j-1}r_n^j = (R_j^{j-1})^T {}^{j-1}r_n^j. \quad (4.5)$$

For the MAMMOTH rover, a single leg kinematic branch extends from frame I ($j = 1$) to frame C_i ($n = 11$). The first five frames are common for all of the MAMMOTH rover legs as they deal with the transformation from I to B . The relative position vectors and rotation matrices for all of the frame transformations within the first five frames are given as part of the Inertial to Body Frame Transformation section of Table 4.1. Frames 6 (H_i) through 11 (C_i) are specific to the individual leg i . The position vectors and rotation matrices for each of these adjacent frame transformations may be formulated using the DH parameters given in the Leg DH Parameters section of Table 4.1. Using these parameters, the position

vector r_{k+1}^k that expresses the position of frame $k + 1$ relative to frame k is expressed as:

$$\underline{r}_{k+1}^k = [a_{dh,k+1} \cos \theta_{dh,k+1} \quad a_{dh,k+1} \sin \theta_{dh,k+1} \quad d_{dh,k+1}]^T. \quad (4.6)$$

The rotation matrix R_{k+1}^k may also be expressed as:

$$R_{k+1}^k = \begin{bmatrix} \cos \theta_{dh,k+1} & -\sin \theta_{dh,k+1} \cos \alpha_{dh,k+1} & \sin \theta_{dh,k+1} \sin \alpha_{dh,k+1} \\ \sin \theta_{dh,k+1} & \cos \theta_{dh,k+1} \cos \alpha_{dh,k+1} & -\cos \theta_{dh,k+1} \sin \alpha_{dh,k+1} \\ 0 & \sin \alpha_{dh,k+1} & \cos \alpha_{dh,k+1} \end{bmatrix}^T. \quad (4.7)$$

With Eqs. (4.2), (4.3), (4.6), (4.7) and the data presented in Table 4.1, it is now possible to solve for all position vectors and rotation matrices that make up the forward position kinematic model in Eq. (4.1). A simple example showing the use of DH parameters is given in Appendix A.1.2.

4.2.2 Static Stability

Given that the MAMMOTH rover is intended to operate at relatively slow speeds, a static tip-over stability model based on the force angle stability margin originally formulated in [83] is used to evaluate how close the robot is to tipping over. This model is based on the stability polygon formed by the rover's wheel-ground contact points and the location of the platform's centre of mass. The rover is assumed to be in a stable state if the weight vector, \underline{f}_g , intersects with the interior of the stability polygon formed by each of the contact points. Figure 4.1 shows the stability polygon formed when four wheels are in contact with the ground.

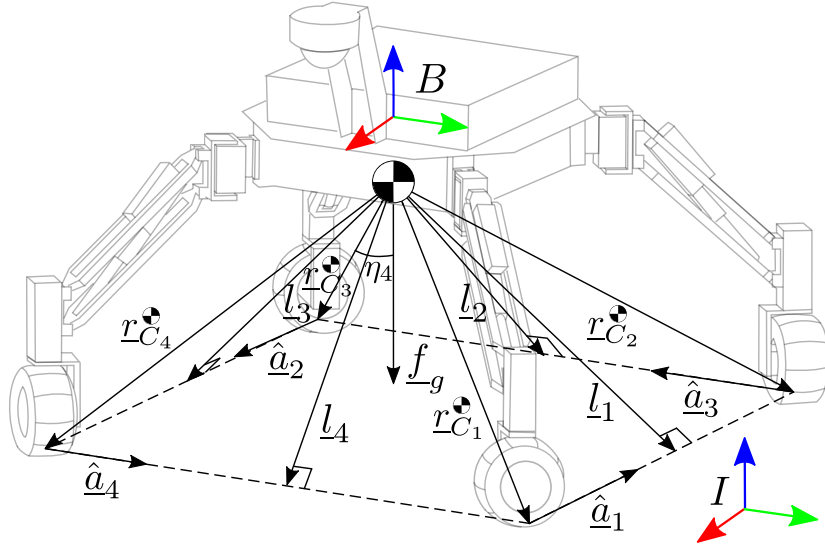


Figure 4.4 – A labelled view of the MAMMOTH rover's stability model that is a function of the stability polygon and centre of mass.

In this stability model, a contact vector, \underline{c} , is defined as containing the ordered leg indices of

the legs that are in contact with the ground. If for example leg 1 is raised, then $\underline{c} = [2 \ 3 \ 4]$, while if all legs are in contact with the ground $\underline{c} = [1 \ 2 \ 3 \ 4]$. The variable m is the length of \underline{c} . The stability metric, β , is the minimum $\eta_{\underline{c}(i)}$ value, the angle between the weight vector, \underline{f}_g , and the vector $\underline{l}_{\underline{c}(i)}$. The metric β quantifies how close the rover is to tipping over. Each $\eta_{\underline{c}(i)}$ angle is found by first defining the $\underline{a}_{\underline{c}(i)}$ vectors that connect each of the wheel-ground contact points:

$$\begin{aligned}\underline{a}_{\underline{c}(i)} &= \underline{r}_{C_{\underline{c}(i+1)}}^{\bullet} - \underline{r}_{C_{\underline{c}(i)}}^{\bullet}, \quad i = \{1, \dots, m-1\} \\ \underline{a}_{\underline{c}(i)} &= \underline{r}_{C_{\underline{c}(1)}}^{\bullet} - \underline{r}_{C_{\underline{c}(i)}}^{\bullet}, \quad i = m.\end{aligned}\tag{4.8}$$

The vector $\underline{l}_{\underline{c}(i)}$ originates at \bullet and is orthogonal to $\underline{a}_{\underline{c}(i)}$. It can therefore be expressed using an orthogonal projection operator:

$$\underline{l}_{\underline{c}(i)} = \left(I - \hat{\underline{a}}_{\underline{c}(i)} \hat{\underline{a}}_{\underline{c}(i)}^T \right) \underline{r}_{C_{\underline{c}(i)}}^{\bullet}.\tag{4.9}$$

A stability angle, $\eta_{\underline{c}(i)}$ is then found from:

$$\eta_{\underline{c}(i)} = \mu \arccos \left(\hat{\underline{f}}_g \hat{\underline{l}}_{\underline{c}(i)} \right),\tag{4.10}$$

where

$$\mu = \begin{cases} +1 & \left(\left[\hat{\underline{l}}_{\underline{c}(i)} \right]^{\times} \hat{\underline{f}}_g \right) \hat{\underline{a}}_{\underline{c}(i)} > 0 \\ -1 & \text{otherwise,} \end{cases}\tag{4.11}$$

which leads to an expression for the stability metric:

$$\beta = \min \eta_i, \quad i = 1, \dots, m.\tag{4.12}$$

This stability metric indicates an unstable tip-over state when it is less than zero.

4.3 Velocity Kinematics

The velocity kinematics of a WMR define the relationship between workspace rates, actuator rates and wheel contact rates. In this section the forward velocity kinematics is used to solve the wheel-ground contact point rates as a function of the workspace rates and the joint rates. The inverse velocity kinematics is used to solve the velocity of the body frame as a function of the contact and joint rates. From the velocity kinematics model the mobility of the platform is then analyzed, which leads to the identification of the independently controlled degrees of freedom of the MAMMOTH platform. The steering and wheel drive rates may not be determined directly from this model, therefore additional expressions are provided for each.

4.3.1 General Velocity Kinematic Model

A general velocity kinematic model of the kinematic branch i of a WMR is given as:

$$\begin{bmatrix} \underline{v}_{C_i}^I \\ \underline{\omega}_{C_i}^I \end{bmatrix} = J_{C_i}^I \begin{bmatrix} \underline{\dot{p}} \\ \underline{\dot{\gamma}} \end{bmatrix} + \underline{\dot{\sigma}}_i, \quad (4.13)$$

where J_C^I is the Jacobian mapping the pose and joint rates, $[\underline{\dot{p}} \ \underline{\dot{\gamma}}]^T$, to contact point rates $[\underline{v}_{C_i}^I \ \underline{\omega}_{C_i}^I]^T$. The Jacobian is formulated using the recursive kinematic propagation technique presented in [65] and summarised below for completeness.

The Jacobian is defined using a recursive expression, in which the base case is at the terminal index n .

$$J_n^{n-1} = \underline{d}_n^{n-1}. \quad (4.14)$$

The recursion expression is defined as:

$$J_n^{j+k-1} = \left[\begin{bmatrix} I_{3 \times 3} & -[j+k-1] \underline{r}_n^{j+k} \times \\ 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix} \underline{d}_{j+k}^{j+k-1} \right] \left[\begin{bmatrix} R_{j+k}^{j+k-1} & 0_{3 \times 3} \\ 0_{3 \times 3} & R_{j+k}^{j+k-1} \end{bmatrix} J_n^{j+k} \right] \quad (4.15)$$

where j is the root index, and k is an index that starts at 1 and counts upwards towards n on each recursion. The variable $\underline{d}_{j+k}^{j+k-1}$ is a 6×1 vector that maps the joint rate associated with frame $j+k$ to the adjacent frame velocity vector $\begin{bmatrix} \underline{v}_{j+k}^{j+k-1} & \underline{\omega}_{j+k}^{j+k-1} \end{bmatrix}$:

$$\begin{bmatrix} \underline{v}_{j+k}^{j+k-1} \\ \underline{\omega}_{j+k}^{j+k-1} \end{bmatrix} = \underline{d}_{j+k}^{j+k-1} \dot{q}_{j+k}. \quad (4.16)$$

For the MAMMOTH rover, values for \underline{d}_{k+1}^k may be found for all frames between I and B in Table 4.1, while for the remaining leg i frames \underline{d}_{k+1}^k may be expressed as a function of the DH parameters:

$$\underline{d}_{k+1}^k = [-a_{dh,k+1} s\theta_{dh,k+1} \quad a_{dh,k+1} c\theta_{dh,k+1} \quad 0 \quad 0 \quad 0 \quad 1]^T. \quad (4.17)$$

An example Jacobian formulation that applies Eqs. (4.14) to (4.17) to find the inertial to contact frame Jacobian for the MAMMOTH rover leg is given in Appendix A.1.3.

For ease of notation the Jacobian may be decomposed into a linear combination of the workspace Jacobian, P that relates $\underline{\dot{p}}$ to the contact point velocity and the articulation Jacobian, $Q_{C_i}^I$, which relates $\underline{\dot{\gamma}}_i$ to the contact point velocity:

$$\begin{bmatrix} \underline{v}_{C_i}^I \\ \underline{\omega}_{C_i}^I \end{bmatrix} = P \underline{\dot{p}} + Q_{C_i}^I \underline{\dot{\gamma}}_i + \underline{\dot{\sigma}}_i. \quad (4.18)$$

For the full velocity kinematic model, Eq. (4.18) is stacked in the following arrangement:

$$\begin{aligned} \underline{v}_C^I = \begin{bmatrix} \underline{v}_{C_1}^I \\ \underline{\omega}_{C_1}^I \\ \vdots \\ \underline{v}_{C_4}^I \\ \underline{\omega}_{C_4}^I \end{bmatrix} &= \begin{bmatrix} P & Q_{C_1}^I & \mathbf{0}_{6 \times 3} & \mathbf{0}_{6 \times 3} & \mathbf{0}_{6 \times 3} \\ P & \mathbf{0}_{6 \times 3} & Q_{C_2}^I & \mathbf{0}_{6 \times 3} & \mathbf{0}_{6 \times 3} \\ P & \mathbf{0}_{6 \times 3} & \mathbf{0}_{6 \times 3} & Q_{C_3}^I & \mathbf{0}_{6 \times 3} \\ P & \mathbf{0}_{6 \times 3} & \mathbf{0}_{6 \times 3} & \mathbf{0}_{6 \times 3} & Q_{C_4}^I \end{bmatrix} \begin{bmatrix} \underline{\dot{p}} \\ \underline{\dot{\gamma}}_1 \\ \underline{\dot{\gamma}}_2 \\ \underline{\dot{\gamma}}_3 \\ \underline{\dot{\gamma}}_4 \end{bmatrix} + \begin{bmatrix} \underline{\dot{\sigma}}_1 \\ \underline{\dot{\sigma}}_2 \\ \underline{\dot{\sigma}}_3 \\ \underline{\dot{\sigma}}_4 \end{bmatrix} \\ &= [P_{full} \quad Q_C^I] \begin{bmatrix} \underline{\dot{p}} \\ \underline{\dot{\gamma}} \end{bmatrix} + \underline{\dot{\sigma}}. \end{aligned} \quad (4.19)$$

The workspace Jacobian P is a function of the body frame's translation relative to the inertial frame and roll, pitch and yaw. The workspace rate vector $\underline{\dot{p}}$ is composed of the translational velocities as well as the roll, pitch, yaw rates $(\dot{\phi}, \dot{\theta}, \dot{\psi})$. The relationship between $\dot{\phi}, \dot{\theta}, \dot{\psi}$ and $\underline{\omega}_B^I$ is discussed in Appendix A.2.

If it is desired to develop an inverse kinematic solution that tracks the position and orientation of the rover's body frame relative to the inertial frame by integrating $\underline{\dot{p}}$, the velocity kinematic model may be manipulated to find $\underline{\dot{p}}$ by the following:

$$\underline{\dot{p}} = (P_{full})^\dagger (\underline{v}_C^I - Q_C^I \underline{\dot{\gamma}}) \quad (4.20)$$

assuming no slip. The \dagger symbol denotes the use of the Moore-Penrose pseudo inverse.

4.3.2 Steering and Driving

Given the centre-oriented steering joint configuration, the angular steering position for each leg, q_{S_i} , may not be defined as a function of the workspace and articulation vectors alone. It is defined as a function of the workspace and articulation rates and positions. The steering joint angle must align with the desired direction of travel of the steering frame as shown in Figure 4.5a. From Figure 4.5a it is seen that the steering angular position q_{S_i} for leg i is defined as the negative multi-valued inverse tangent (atan2) of the velocity along the steering frame's x -axis over the the velocity along the steering frame's y -axis:

$$q_{S_i} = -\text{atan2} \left({}^{S_i}v_{S_{i,x}}^I, {}^{S_i}v_{S_{i,y}}^I \right), \quad (4.21)$$

where

$${}^{S_i}v_{S_i}^I = R_I^{S_i} J_{S_i}^I (1:3, :) \begin{bmatrix} \underline{\dot{p}} \\ \underline{\dot{\gamma}}_i \end{bmatrix}. \quad (4.22)$$

The wheel joint rate, $\dot{q}_{A'_i}$, is the rotation rate of the the A'_i frame. As shown in Figure 4.5b, the A_i frame is held stationary relative to the steering frame S_i so as to preserve the correct position of the contact point frame C_i . The A'_i frame rotates with the wheel. The rate $\dot{q}_{A'_i}$ is solved by finding the magnitude of the velocity of the contact point, and dividing it by L_5 , the radius of the wheel:

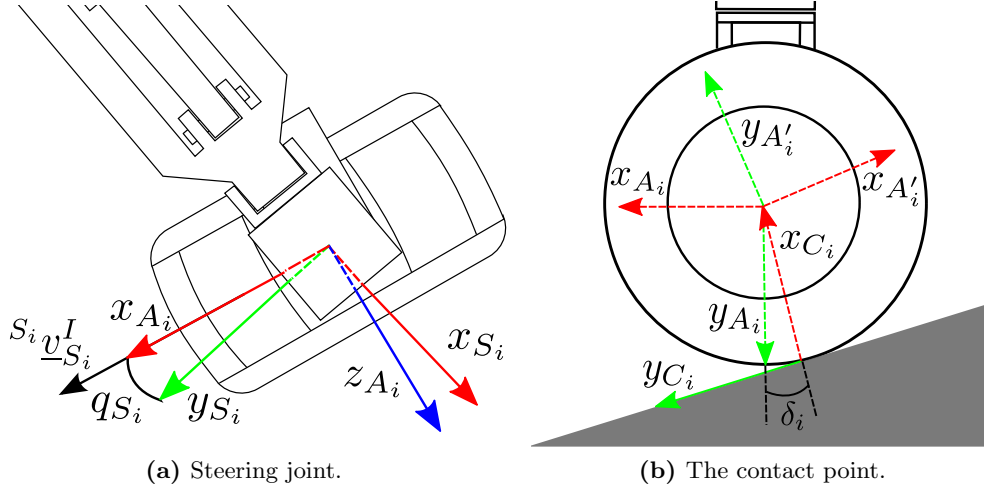


Figure 4.5 – The MAMMOTH rover steering joint and wheel drive joint. The steering joint angle q_{S_i} is the angle between the y_{S_i} -axis and the desired steering frame velocity vector $\underline{v}_{S_i}^I$.

$$\dot{q}_{A'_i} = \frac{|C_i \underline{v}_{C_i,y}^I|}{L_5} = \frac{|R_I^{C_i} \underline{v}_{C_i,y}^I|}{L_5}, \quad (4.23)$$

where $\underline{v}_{C_i}^I$ can be found from Eq. (4.19). The equation works under the assumption that there is no compliance within the wheel. For leg i , the wheel-ground contact angle, δ_i , is a function of both the terrain slope and the configuration of the platform. A diagram showing δ_i is presented in Figure 4.5b.

4.3.3 Mobility Analysis

When modelling any mechanism it is critical to establish the number of independent degrees of freedom, or mobility of the system. The velocity kinematic model from Eq. (4.19) may be used to perform this analysis. This model may be re-formulated as

$$\Omega \underline{\dot{q}} = [I, \quad -P_{full}, \quad -Q_C^I] \underline{\dot{q}} = \underline{\dot{\sigma}} = 0, \quad (4.24)$$

where Ω is composed of the full system Jacobians and $\underline{\dot{q}}$ is the body and joint rate vector including the wheel contact rates. It is assumed in Eq. (4.24) that there is no slip, hence setting $\underline{\dot{\sigma}}$ to zero. This expression is composed of 24 linearly independent equations. To describe the constrained motion of the MAMMOTH rover eight more equations are required. It is desired that all four wheels stay in contact with the terrain: $\underline{v}_{C,z}^I = \underline{0}_{4 \times 1}$. Also due to the contact condition, the angular rotation rate of each wheel-ground contact point must be equal to the wheel-ground contact angular rate: $\underline{\omega}_{C,z}^A = \underline{\dot{\delta}}$. These eight additional equations may be appended to the model to generate the full kinematic system of equations. It is assumed within these system of equations that the MAMMOTH rover is operating on a

surface with wheel-ground contact angles of zero degrees. Otherwise, additional constraints relating to motion of each of the contact frames may need to be applied.

As posed in [44], the mobility of the platform, m_r , or the number of independently controllable degrees of freedom is defined as the difference between the number of variables in $\underline{\dot{q}}$ and the rank of the Ω matrix:

$$m_r = \dim(\underline{\dot{q}}) - \text{rank}(\Omega). \quad (4.25)$$

Given that there are now 32 linearly independent equations, $\text{rank}(\Omega) = 32$, and the dimension of the $\underline{\dot{q}}$ matrix is 42. The mobility is therefore $m_r = 10$. Intuitively, the 10 independently controlled degrees of freedom in this case are the six variables in the body rate vector $\underline{\dot{p}}$ and the four hip rate variables $\underline{\dot{q}}_H$.

If wheel i is lifted off the ground, the contact point at the origin of C_i is defined as being rigidly attached to the surface of the wheel. In this case, its x_{C_i} axis is co-linear with the y_{A_i} axis. The associated $v_{C_i,z}^I = 0$ constraint equation is removed from the model given that the wheel is lifted off the ground. The mobility calculation therefore becomes: $m_r = 42 - 31 = 11$. An extra degree of mobility is gained with \dot{q}_{U_i} being independently controllable.

In this analysis, the non-holonomic constraint at the steering joints have not been included. The effect of these joints will cause the platform to realize a 10 or 11 degree of freedom velocity only after steering its wheels to the appropriate configuration. This definition of mobility does not comply with the definition of mobility posed in [22] in which mobility is defined as the dimensionality of the instantaneously accessible velocity space.

4.4 Alternative Mobility Modes

To demonstrate the use of the velocity kinematic model, a set of manoeuvres composed of motion primitives are now described. These complex manoeuvres demonstrate various mobility modes that the MAMMOTH rover is capable of performing. Experimental validation of each of the described complex manoeuvres is provided in Section 5.2. In each sub-section the associated complex manoeuvre is described using a finite state machine representation along with the velocity kinematic formulation of the underlying motion primitives. Before presenting each of the complex manoeuvres it is first necessary to describe a transformation method that is a function of the robot's direction of travel.

It is first assumed that each of these complex manoeuvres is performed on a flat plane. The robot's direction of travel is defined using a new inertial coordinate frame I' that has its x -axis parallel to the direction of travel of the robot. The rotational transformation $R_I^{I'}$ is a function of a yaw motion $\psi_I^{I'}$. This yaw is used throughout this section to define the direction of travel. Each of the complex manoeuvres is reliant on initial joint actuator positions and leg index vectors that are functions of $\psi_I^{I'}$. To find these vectors the quadrant index of the vehicle's body velocity vector is first found:

$$\zeta = \left\lfloor \frac{\text{atan2}(\psi_I^{I'} + \pi/4)}{\pi/2} \right\rfloor. \quad (4.26)$$

To transform terminal joint angle vectors or discrete leg index vectors ζ is used as an exponent of a permutation matrix. A generic vector u' is transformed to u by

$$\underline{u} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}^{\zeta} \underline{u}' = V^{\zeta} \underline{u}'. \quad (4.27)$$

4.4.1 Rowing

The rowing manoeuvre utilizes the robot's hips for locomotion. Figure 4.6a outlines the manoeuvre's finite state machine. The manoeuvre is divided into two main motion primitives: body motion and hip motion. Initially, during the hip motion, all hip joints are moved towards the the initial configuration, \underline{q}_{H_o} . The hip joints and wheel joints are actuated, as shown in Figure 4.6b. The body motion then propels the robot body along its direction of travel using the four hip actuators, while also driving the wheel contact points orthogonally to the body's direction of travel, as shown in Figure 4.6c. During the body motion, the hips are driven to the final configuration, \underline{q}_{H_f} . Rowing continues by repeating the hip motion then body motion cycle. If the desired yaw changes at any time then the rowing manoeuvre resets with a hip motion.

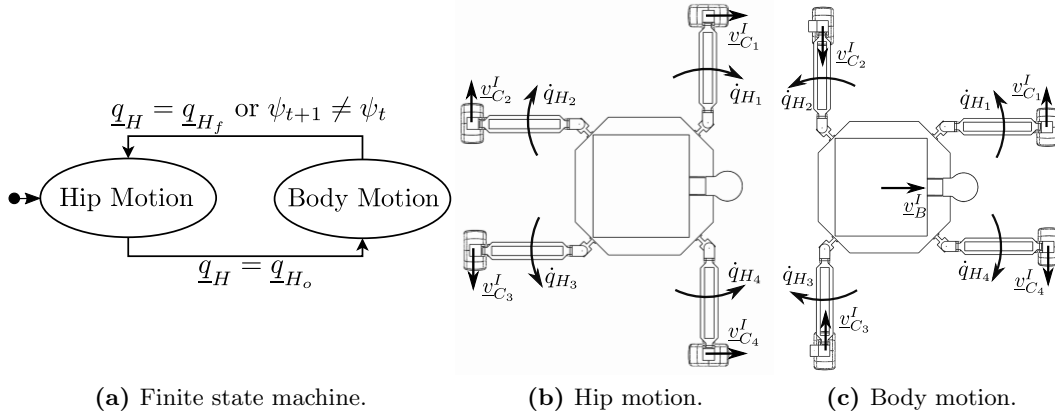


Figure 4.6 – The rowing manoeuvre.

The terminal hip configurations are determined by first expressing their base vectors:

$$\underline{q}'_{H,o} = [45^\circ \quad 45^\circ \quad -45^\circ \quad -45^\circ]^T, \underline{q}'_{H,f} = [-45^\circ \quad -45^\circ \quad 45^\circ \quad 45^\circ]^T. \quad (4.28)$$

These base vectors are transformed as a function of $\psi_I^{I'}$. The permutation matrix transfor-

mation from Eq. (4.27) is applied to do so:

$$\underline{q}_{Ho} = \begin{cases} V^\zeta \underline{q}'_{H,o} + \iota, & \text{if } \zeta \geq 0 \\ V^{-\zeta} \underline{q}'_{H,o} + \iota, & \text{if } \zeta < 0 \end{cases}, \underline{q}_{Hf} = \begin{cases} V^\zeta \underline{q}'_{H,f} + \iota, & \text{if } \zeta \geq 0 \\ V^{-\zeta} \underline{q}'_{H,f} + \iota, & \text{if } \zeta < 0 \end{cases}. \quad (4.29)$$

An offset angle ι is added to the transformed basis vectors to find the initial and final hip angles. The offset angle is found by

$$\iota = \text{mod} \left((\psi_I^{I'} + \pi/4), \pi/2 \right) - \pi/4. \quad (4.30)$$

These initial and final conditions have been selected to allow for body motion while retaining static stability for all MAMMOTH rover heights.

During the body motion, it is assumed that the thigh actuators are held stationary ($\dot{\underline{q}}_U = \underline{0}_{4 \times 1}$). It is also assumed that the wheel contact frames only move orthogonal to the direction of travel of the robot body. The body motion is solved for using a sequence of operations. The input to this sequence is an initial body rate vector $\dot{\underline{p}}_{init}$ that describes the direction of travel of the robot during rowing. Eq. (4.19) is transformed into the new inertial coordinate frame I' using the transformation matrix $\Lambda = \begin{bmatrix} R_I^{I'} & 0_{3 \times 3} \\ 0_{3 \times 3} & R_I^{I'} \end{bmatrix}$ and constrained using the selection vector S :

$$S\Lambda \underline{v}_C^I = \underline{0}_{4 \times 1} = S\Lambda P_{full} \dot{\underline{p}}_{init} + S\Lambda Q_{C,\dot{\underline{q}}_H}^I \dot{\underline{q}}_{H,init}. \quad (4.31)$$

$S = [1 \ 0 \ 0 \ 0 \ 0 \ 0]$ selects the wheel contact velocity component that is aligned with the $x_{I'}$ axis, which is constrained to zero. The Jacobian $Q_{C,\dot{\underline{q}}_H}^I$ maps the hip rates to the wheel contact rates. An initial hip motion, $\dot{\underline{q}}_{H,init}$, may then be solved for by re-arranging Eq. (4.31):

$$\dot{\underline{q}}_{H,init} = (S\Lambda Q_{C,\dot{\underline{q}}_H}^I)^\dagger (-S\Lambda P_{full} \dot{\underline{p}}_{init}), \quad (4.32)$$

where \dagger is the Moore-Penrose pseudo inverse.

The next operation is to calculate the $\dot{\underline{q}}_H$ by imposing a rate limit on $\dot{\underline{q}}_{H,init}$:

$$\dot{\underline{q}}_H = \begin{cases} \dot{\underline{q}}_{H,init} & \text{if } \max(|\dot{\underline{q}}_{H,init}|) \leq \dot{q}_{H,max} \\ \dot{\underline{q}}_{H,max} \frac{\dot{\underline{q}}_{H,init}}{\max(|\dot{\underline{q}}_{H,init}|)} & \text{otherwise} \end{cases} \quad (4.33)$$

where $|\dot{\underline{q}}_{H,init}|$ is element-wise absolute value of $\dot{\underline{q}}_{H,init}$. The rate limit is defined by the constant $\dot{q}_{H,max}$, which for the MAMMOTH rover is typically set at $6^\circ/\text{s}$. The body rate in the direction of travel, $\dot{x}_B^{I'}$, is solved for as a function of $\dot{\underline{q}}_H$:

$$\dot{x}_B^{I'} = (-S\Lambda P_{full})^\dagger (S\Lambda Q_{C,\dot{\underline{q}}_H}^I \dot{\underline{q}}_H). \quad (4.34)$$

During the hip motion, it is assumed that the thighs are held stationary with $\dot{\underline{q}}_U = \underline{0}_{4 \times 1}$, and the body frame also remains stationary along the y_I and z_I axes. By substituting these

static variables, Eqs. (4.33) and (4.34) into Eq. (4.19), the wheel contact velocities \underline{v}_C^I for the hip motion primitive may be solved.

4.4.2 Inchworming

The inchworming manoeuvre utilizes the linear actuators embedded in each thigh for platform motion. This may be especially useful when attempting to traverse soft soil or inclined terrain. Inchworming is summarised in Figure 4.7. Like the rowing manoeuvre, inch-worming requires initial leg articulation angles. The robot will maintain the same hip angular positions, \underline{q}_H , throughout the entire manoeuvre. The hip positions may be calculated similarly to Eq. (4.29):

$$\underline{q}_{H_o} = V^\zeta [-45^\circ \quad 45^\circ \quad -45^\circ \quad 45^\circ]^T + \iota. \quad (4.35)$$

The robot starts a raise manoeuvre at $\underline{q}_U = 10^\circ$ and ends at $\underline{q}_U = 50^\circ$, while the lower manoeuvre has the opposite terminal thigh positions.

To describe this motion we must constrain the velocity kinematic model by first identifying which variables are held at zero. These include the hip joint rates $\dot{\underline{q}}_H = \underline{0}_{4 \times 1}$, and the body rotation rates $\underline{\omega}_B^I = \underline{0}_{3 \times 1}$. For the lower motion primitive, the two front wheels relative to the direction of travel are only constrained along the inertial z -axis as the wheel's should not leave the terrain surface. The back two wheels are constrained to not move at all so as to anchor the motion. It is the opposite for the raise motion primitive: the front wheels relative to the direction of travel are held stationary while the back wheels are only constrained to stay stationary along the inertial z -axis.

The velocity kinematics may be expressed as a modification of the velocity kinematic model from Eq. (4.19):

$$\underline{v}_C^I = P_{\underline{v}_B^I}^I \underline{v}_B^I + P_{\underline{\omega}_B^I}^I \underline{\omega}_B^I + Q_{C,\dot{\underline{q}}_H}^I \dot{\underline{q}}_H + Q_{C,\dot{\underline{q}}_U}^I \dot{\underline{q}}_U, \quad (4.36)$$

where $P_{\underline{v}_B^I}^I$ is the Jacobian mapping body translation rate, $P_{\underline{\omega}_B^I}^I$ is the Jacobian mapping body rotation rate, $Q_{C,\dot{\underline{q}}_H}^I$ is the Jacobian mapping hip rates and $Q_{C,\dot{\underline{q}}_U}^I$ is the Jacobian mapping thigh rates. Substituting $\dot{\underline{q}}_H = \underline{0}_{4 \times 1}$ and $\underline{\omega}_B^I = \underline{0}_{3 \times 1}$ into Eq. 4.36 gives:

$$\underline{v}_C^I = P_{\underline{v}_B^I}^I \underline{v}_B^I + Q_{\dot{\underline{q}}_U}^I \dot{\underline{q}}_U. \quad (4.37)$$

Furthermore, the selection matrix S is applied to impose the wheel-contact constraints, from which the body rates may be solved for:

$$\begin{aligned} S \underline{v}_C^I &= \underline{0}_{8 \times 1} = S P_{\underline{v}_B^I}^I \underline{v}_B^I + S Q_{\dot{\underline{q}}_U}^I \dot{\underline{q}}_U \\ \underline{v}_B^I &= (-S P_{\underline{v}_B^I}^I)^\dagger S Q_{\dot{\underline{q}}_U}^I \dot{\underline{q}}_U. \end{aligned} \quad (4.38)$$

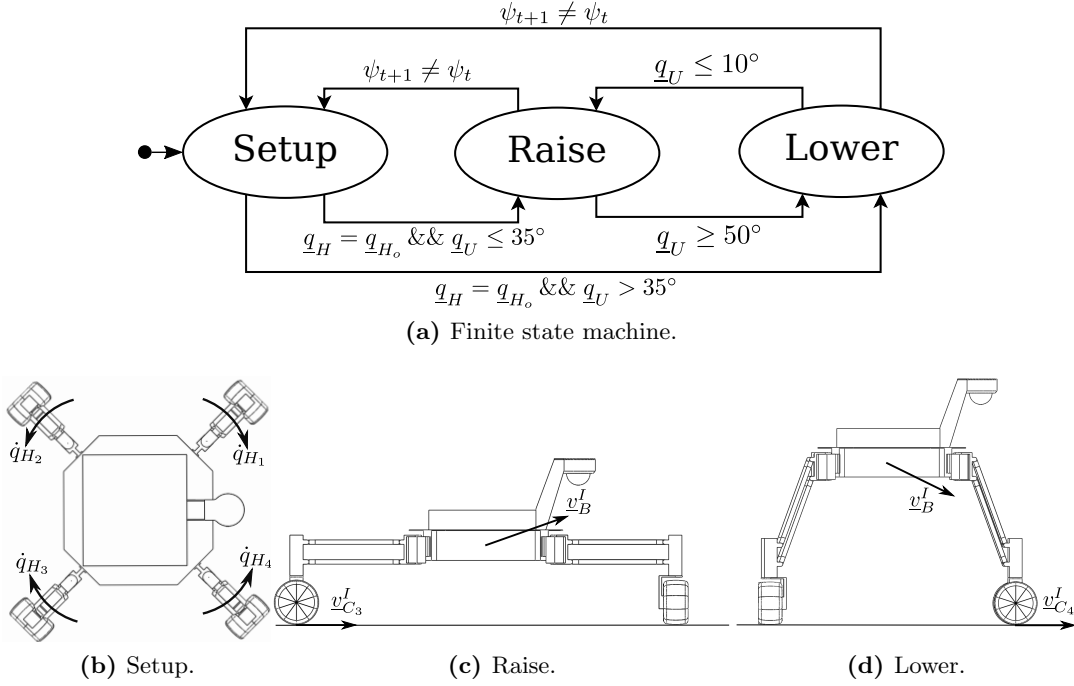


Figure 4.7 – The inchworming manoeuvre.

For this manoeuvre the magnitude of each element of the thigh joint velocity vector $\dot{\underline{q}}_U$ is set as a function of the maximum thigh actuation rate $\dot{q}_{Th,max}$. The relationship between the thigh actuation rate and the upper thigh actuation rate is defined in Appendix A.5. During the raise motion each element of $\dot{\underline{q}}_U$ is positive and negative during the lower motion. The selection matrix S is a function of the raise or lower motion primitive and the direction of travel. To develop an expression for S we first have to generate two matrices that select the wheel-contact dimensions that are to be set to zero. For a wheel that cannot move at all we use

$$S_A = \begin{bmatrix} I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix}, \quad (4.39)$$

while for a wheel that may move freely except that it may not come off the ground we use

$$S_B = \begin{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix}. \quad (4.40)$$

To develop an expression for S we again use the base index basis vector $\underline{e}' = [1 \ 2 \ 3 \ 4]^T$. The index basis vector \underline{e} is found from $\underline{e} = V^Z \underline{e}'$. With this notation established, S is expressed as $\text{diag}(S_{e(1)}, S_{e(2)}, S_{e(3)}, S_{e(4)})$. For the lower motion primitive the sub-matrices are defined as $S_{e(1)} = S_A$, $S_{e(2)} = S_B$, $S_{e(3)} = S_B$ and $S_{e(4)} = S_A$. For the raise motion primitive the sub-matrices are defined as $S_{e(1)} = S_B$, $S_{e(2)} = S_A$, $S_{e(3)} = S_A$ and $S_{e(4)} = S_B$.

4.5 Actively Articulated Suspension

Actively articulated suspension of a RWMR relates to the robot's ability to actuate its joints so as to conform to the terrain as it drives over it. By doing so, the robot may overcome uneven terrain and attempt to keep all wheels in contact to maximize wheel traction. This problem may be addressed by developing a control policy that attempts to keep each wheel-ground contact point in contact with the terrain. To accomplish this, sensory information regarding the shape of the terrain profile along with the pose and joint configuration of the robot are required. Terrain profile data along with localisation are provided by an RGB-D sensor mounted on the head of the MAMMOTH rover. Joint configuration is sensed using angular encoder data and the position of each of the robot frames is found using the joint data and the rover's kinematic model. In this section, the control strategy is first described followed by a summary of how the terrain model is generated. Experiments demonstrating actively articulated suspension with the MAMMOTH rover are presented in Section 5.3 and in [92].

4.5.1 Articulation Solution

The control policy used to keep each wheel in contact with the terrain drives the wheel away or towards the terrain based on the distance between the surface of the wheel and the surface of the terrain. Additionally, a feed-forward component that predicts the necessary wheel contact velocity based on the vertical rate of the terrain profile as the rover moves over it is used. The control policy for leg i is expressed as:

$$v_{C_i,z}^I = K_r(r_{terrain,i,z}^I - r_{C_i,z}^I) + \frac{\Delta r_{terrain,i,z}^I}{\Delta t} \quad (4.41)$$

A diagram showing a wheel moving over undulating terrain and the various variables used within the control policy is shown in Figure 4.8. A block diagram for the controller is shown in Figure 4.9.

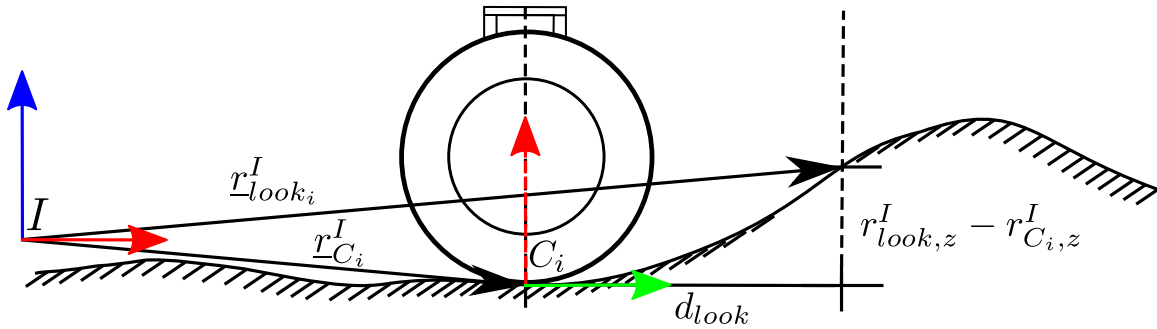


Figure 4.8 – Wheel i crossing undulating terrain and the variables used to formulate the actively articulated suspension control policy.

The desired vertical position of the terrain at contact point i is denoted by $r_{terrain,z}^I$. It is determined by finding the point on the surface mesh generated by the RGB-D camera that

is directly below or above $r_{C,z}^I$, which in turn is determined by the robot's angular encoder data and kinematic model.

The feed-forward component of the control policy uses the rate of the terrain height with respect to time. This rate is approximated by finding the rate of change of terrain profile over a lookahead distance, d_{look} , given a constant platform velocity, \underline{v}_B^I . The expression for this approximation is given as:

$$\frac{\Delta r_{terrain_i,z}^I}{\Delta t} \approx (r_{look_i,z}^I - r_{C_i,z}^I) \left(\frac{d_{look}}{|\underline{v}_{B,xy}^I|} \right). \quad (4.42)$$

Given the desired contact point velocity, the thigh articulation rates, $\dot{\underline{q}}_U$ are then found that meet $\begin{bmatrix} \dot{\underline{p}} \\ \dot{\underline{q}}_H \end{bmatrix}$, the rates of each of the independently driven degrees of freedom. The velocity kinematic model from Eq. (4.19) is reduced using the selection matrix S to express just the wheel contact velocity along the z_I axis:

$$\underline{v}_{C,z}^I = SP_{full}(\underline{p})\dot{\underline{p}} + SQ_{\underline{q}_H,full}\dot{\underline{q}}_H + SQ_{\underline{q}_U,full}\dot{\underline{q}}_U. \quad (4.43)$$

where S is 4×24 matrix that may be generate with the following kronecker product:

$$S = I_{4 \times 4} \otimes [0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0]. \quad (4.44)$$

The S matrix selects all of the rows within the Jacobians that contribute towards the vertical rate of the wheel-ground contact point.

The thigh joint rates may then be solved for by re-arranging Eq. (4.43) to give:

$$\dot{\underline{q}}_U = (SQ_{\underline{q}_U})^{-1} \left(\underline{v}_{C,z}^I - SP_{full}(\underline{p})\dot{\underline{p}} - SQ_{\underline{q}_H,full}\dot{\underline{q}}_H \right) \quad (4.45)$$

This thigh joint rate solution is an approximation given that a vertical displacement of the contact frame relative to the inertial frame will have a corresponding displacement along the $x_I y_I$ plane due to the kinematic configuration of the MAMMOTH rover's thigh joint. To improve upon this solution, the hip actuators could be used as dependently controlled joints. Ideally, it is desired that the wheel-ground contact point be moved vertically relative to the inertial frame without any displacement along the $x_I y_I$ plane. A possible item of future work is to re-design the MAMMOTH leg to incorporate a knee joint to enable this isolated vertical motion.

4.5.2 Terrain Mapping

To find a reliable solution to the actively articulated suspension problem, the structure of the terrain being traversed must be known. In this thesis, direct terrain profile information is gathered using a a combined RGB-D sensor mounted on the MAMMOTH rover. The

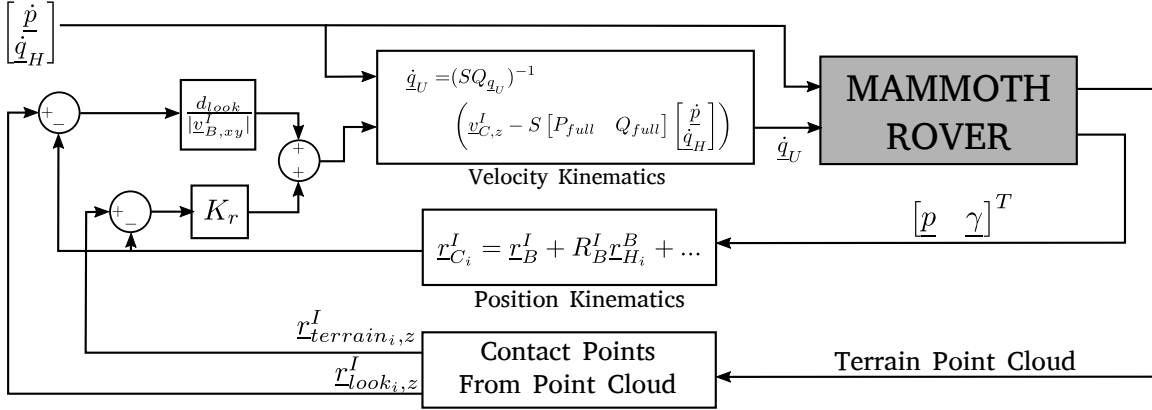


Figure 4.9 – Block diagram for the actively articulated suspension controller.

selected sensor is an ASUS Xtion Pro Live. This sensor provides both images and point clouds of the terrain in front of the rover.

At each frame, a set of visual features is extracted from the colour image and mapped to their 3D locations using the associated point cloud. The set of 3D points are matched across consecutive frames to finally estimate sensor pose increments. This method does not rely on any assumption about the robot kinematics, however it benefits from the fact that the rover motion is relatively slow on the order of centimetres per second. An example terrain map stitched together from consecutive frames is shown in Figure 4.10.

The features are extracted using the Features from the Accelerated Segment Test (FAST) algorithm [100], which is a computationally efficient method for corner detection. The identified features contain local information that ideally make them repeatable across consecutive frames. The approach is not based on image feature matching though. Instead, the associated depth of each feature point is computed using the point cloud provided by the sensor, hence building a set of FAST features enhanced with depth information.

In order to overcome the problem of texture uniformity in the scene, and/or the grouping of features in specific strong-detailed regions of the image, an image bucketing strategy is used. The image is divided into several regions, and a minimum number of FAST features are required to be extracted from each region.

RGB-D sensors provide depth information for most of the image pixels, but not all of them due to the physical properties of the surfaces or the environment conditions. As a result, an image feature point might not have an associated depth value. In this case such points are rejected from the feature set. This strategy does not impact on the overall performance of the terrain mapping since the size of the feature set is relatively large (several hundred points).

Once the feature set has been filtered, the Iterative Closest Point (ICP) technique is used to find the best fit between the previous and the current sets. ICP is a method capable of providing a computationally efficient pose estimate in complex, unstructured environments [88]. ICP has the advantage of locally solving the problem of matching and localization, being generic and potentially used for real-time applications. Moreover, the computational

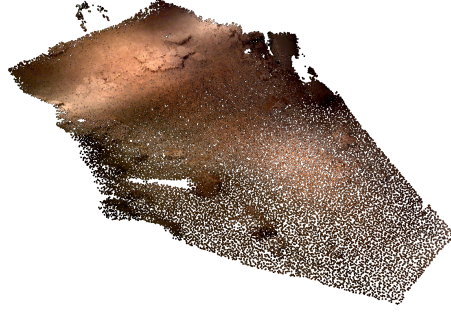


Figure 4.10 – A point cloud map superimposed with RGB data generated after driving the MAMMOTH rover over a section of the Mars Yard. An RGB-D sensor mounted on top of the rover’s mast is used to collect point cloud data.

load is reduced by selecting a set of interest points instead of using the whole point cloud.

The goal of ICP is to find a rigid geometric transformation $\begin{bmatrix} R_{point}^{sensor} & r_{point}^{sensor} \\ 0 & 1 \end{bmatrix}$ that can be applied to the previous set of 3D points such that it is most closely aligned with the current set. The closeness of the alignment is most often based on a cost function such as the sum of mean square distances between point associations. Due to the sparse distribution of the 3D points, the standard point-to-point error metric, E , is used in the iterative process:

$$E = \sum_{j=1}^N w_j \left\| \begin{bmatrix} R_{point}^{sensor} & r_{point}^{sensor} \\ 0 & 1 \end{bmatrix} \underline{u}_j - \underline{z}_j \right\|, \quad (4.46)$$

where (u_j, z_j) are the j th of the N pairs of correspondences from the previous set of 3D points to the current set, w_i is the weight assigned to the j th correspondence.

The sensor pose update is transformed to the inertial frame, I , and along with the kinematic state of the rover, used to calculate the new desired contact points for each rover wheel. In order to obtain the 3D location of the current desired contact point for wheel i , $\underline{r}_{terrain_i, z}^I$, and the desired lookahead contact point, $\underline{r}_{look_i, z}^I$, the generated map point cloud is placed in a kd -tree data structure so that nearest neighbour points from the point cloud can be found efficiently. These positions estimated using k nearest neighbour points in the map, which are weighted according to their distance to the desired contact point.

Chapter 5

Kinematic Modelling Validation Experiments

A contribution of this thesis is the experimental validation of both the MAMMOTH rover motion model presented in Chapter 3 and the planning methodology presented in Chapter 6. All validation is performed using the MAMMOTH rover system described in Chapter 3. Section 5.1 presents experiments performed to showcase the platform’s velocity kinematic model, which is also presented in [90]. In Section 5.2 each of the alternative mobility modes described in Section 4.4 are performed and the energy required for each of these manoeuvres is presented. Experiments that demonstrate actively articulated suspension as described in Section 4.5 and [91] are presented in Section 5.3.

5.1 Kinematic Model Validation

The following experimental trials are used to demonstrate the variety of motions that the MAMMOTH rover is capable of performing. In doing so, the kinematic model of the rover from Chapter 3 is showcased. Additionally, the experiments highlight estimates of the static stability of the platform during operations according to the stability model from Section 4.2.2. Lastly, the performance of the rover in driving along a pre-defined path while using multiple primitive motions synchronously as compared to using a single driving motion primitive is evaluated. Videos of each of the kinematic model validation trials may be viewed at <https://goo.gl/IGIDCi>.

5.1.1 Setup

Five trials are performed along a common workspace path to allow for comparison between each experiment. Each trial is run twice: the results presented are representative examples of each trial. The workspace path followed in each experiment is shown in Figure 5.1. When the rover reaches a target on the path, the rover is reliant on the operator to identify that it

has reached the target and to issue a new driving instruction. In all trials the robot drives along the $x_I y_I$ plane with a velocity of 0.03 m/s.

The rover starts at target 1 with a yaw of $\psi = 0^\circ$, it then drives straight ahead towards target 2. Upon reaching target 2 the rover drives along the y_I -axis, keeping $\psi = 0^\circ$ as it moves towards target 3. At target 3, the rover begins to drive back towards target 1 by turning about an instantaneous centre of rotation (ICR), which is located at target 4. Lastly, upon reaching target 1, the rover performs a spot turn in an anti-clockwise direction until it's yaw is once again at $\psi = 0^\circ$. In each trial the rover follows this workspace path while performing distinct manoeuvres. The manoeuvres and numbering of each trial are as follows:

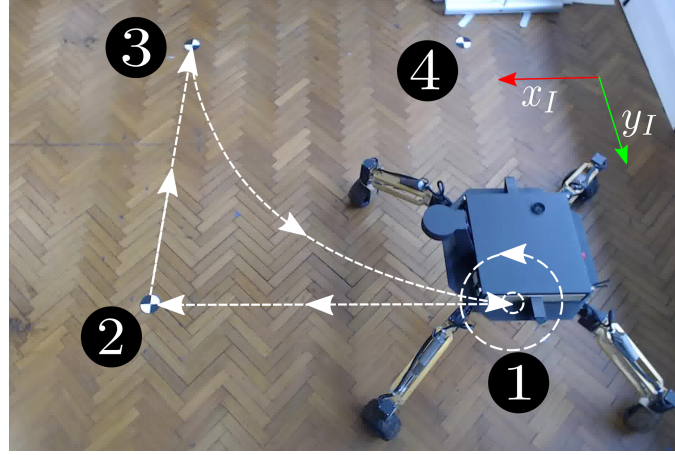


Figure 5.1 – The path followed by the MAMMOTH rover in each experiment with the four labelled targets and inertial coordinate frame.

1. Driving with all hips held at $q_{H_i} = 0^\circ$ and all thighs held at $q_{U_i} = 30^\circ$ ($q_{Th_i} = 80$ mm).
2. Driving with all thighs held at $q_{U_i} = 22^\circ$ ($q_{Th_i} = 70$ mm), while packing and unpacking the hips between $q_{H_i} = 130^\circ$ and $q_{H_i} = -130^\circ$.
3. Driving with all hips held at $q_{H_i} = 0^\circ$, while raising and lowering the thighs between $q_{U_i} = -19^\circ$ ($q_{Th_i} = 10$ mm) and $q_{U_i} = 45^\circ$ ($q_{Th_i} = 95$ mm).
4. Driving with all hips packing and unpacking between between $q_{H_i} = 130^\circ$ and $q_{H_i} = -90^\circ$ and all thighs raising and lowering between $q_{U_i} = -19^\circ$ ($q_{Th_i} = 10$ mm) to $q_{U_i} = 45^\circ$ ($q_{Th_i} = 95$ mm).
5. Driving on three wheels with leg 4 raised to $q_{U_i} = -25^\circ$ ($q_{Th_i} = 0$ mm) and the remaining legs at $q_{U_i} = 22^\circ$ ($q_{Th_i} = 70$ mm). The hip angles are used to adopt a stable configuration with hip 1 at $q_{H_1} = -90^\circ$, hip 2 at $q_{H_2} = 0^\circ$, hip 3 at $q_{H_3} = 90^\circ$ and hip 4 at $q_{H_4} = 0^\circ$.



Figure 5.2 – Snapshots of the MAMMOTH rover between targets. Each row showing an individual experiment. The columns show driving steps, where step 1 is driving straight between targets 1 and 2, step 2 is driving to the side between targets 2 and 3, step 3 is between targets 3 and 1 driving along a curved path centred at target 4, while step 4 is spot turning at target 1. Ghost images of the rover are shown of the rover at the previous step for reference.

5.1.2 Discussion

The sequence of motions performed in each experiment can be seen from the snapshots in Figure 5.2. The figure's rows each show an individual experiment, while each column shows the rover at various steps in the path followed. Each experiment step shows the rover while it is executing a motion between path targets. Step 1 shows the rover moving from target 1 to 2, step 2 shows driving between targets 2 to 3, step 3 shows the ICR turn from targets 3 to 1 and step 4 shows the spot turn about target 1.

Figure 5.6 presents data collected from actuator encoders, stability estimates, β , and the heading angle, ψ of the rover for each of the trials. The hip and thigh positions match the desired motions given that they are based on operator defined inputs to the system. From observing the ankle position and wheel speed plots it is observed that they are dependent on the behaviour of the thigh and hip angular positions. This is quite noticeable when looking at the q_S plot for trial 2 between $t = 150$ s and $t = 197$ s, where there is a symmetric angular profile between the four ankle angles while the hip angles are changing. The wheel speed in the \dot{q}_A plot for the same trial during the same time period also exhibits a symmetric profile between the four legs. At various points in the trials where simultaneous motions were used, it is observed that the wheel rate goes to $0^\circ/\text{s}$ for brief periods of time. This is attributed to the associated ankle driving to an ankle actuator angle equal to or greater than 45° . This angle limit has been chosen to limit the mechanical stresses imposed on the ankle joint.

Based on each trial having the same desired yaw as a function of the global position of the platform, the heading is a metric for defining how well synchronous movements were performed. Figure 5.6 shows the rover's heading angle as a function of trial time for each of the trials in the bottom row of plots. In an ideal case, each of the heading versus time data sets should be the same, being 0° from target 1 to 3, change linearly between 0° and -90° between targets 2 and 3 and then change linearly between -90° and -360° as the spot turn about target 1 is performed.

For each of the trials, the heading plots follow the same trend. Discrepancies between heading plots are due to multiple items not captured in the kinematic model. A velocity limit of $60^\circ/\text{s}$ was placed on each steering actuator to ensure safety of the ankle's mechanical joint. In some situations this resulted in a lag in ankle angle and the wheel being driven in a slightly incorrect direction and in turn wheel slippage. Additionally, it is assumed by the kinematic model that the wheel has a contact point with the ground at any point in time, whereas it actually has a contact ellipse. This contact over a surface area results in wheel slippage and sheer stress on the wheel, especially during significant changes in ankle angular position while the wheel is being driven. This effect may be noticed in each of the plots from trial 2 between times $t = 271$ s and $t = 295$ s and in trial 4 between $t = 272$ s and $t = 300$ s. These time ranges are highlighted by green boxes in Figure 5.6.

The stability estimate for each trial is shown in the second last row of Figure 5.6. Given that the rover stayed stable throughout all of the trials, the stability estimates, β , were always greater than zero. The stability of the platform changes whenever the footprint of the rover changes. This is particularly noticeable in trials 2-4. In trial 2, the stability angles follow a hyperbolic path as the hip angles change at a constant rate. In this trial the

platform is at its least stable when its legs are in their fully packed state when $q_H = 130^\circ$. In trial 3, the stability angles are also seen to change as the body is raised and lowered and the contact points get closer and further away from the body respectively. From all of the trials performed, the platform adopts its most stable configuration when it is closest to the ground plane, with the hips at $q_H = 0^\circ$ as seen in trials 3 and 4. The most unstable configuration is observed in trial 4 when the hips are in their packed configuration and the body is raised to a maximum height of $z_B^I = 0.64$ m above the ground.

5.2 Alternative Mobility Mode Demonstration

In this set of experiment trials each complex manoeuvre described in Section 4.4 is demonstrated with the physical MAMMOTH rover. The aim of this set of experiments is to validate the kinematic models associated with each manoeuvre and highlight the versatility of RWMRs. The energy usage of each complex manoeuvre is also compared against a simple driving mode over the same distance.

5.2.1 Setup

Three trials are presented for the rowing manoeuvre in which the robot rows forwards in the first two and rows backwards in the third. Two trials are also presented for the inchworming manoeuvre on the same surface. In the first inchworming trial the platform is propelled forward, while in the second the platform is propelled backwards. Additional control trials in which the rover is driven without any other joint motion at varying speeds along the same terrain are performed. The traverses performed in each trial take place over 2.6 m to 3.1 m distances on the same terrain type. All manoeuvres are performed on a flat terrain that has a hard base with sandy soil on its surface. Results from kinematic simulations are performed for both rowing and inchworming to validate the kinematic model for each of these manoeuvres.

5.2.2 Rowing

As described in Section 4.4.1, rowing is composed of a body motion and a hip motion. The body motion is responsible for propelling the robot along its direction of travel by using both its hip and wheel actuators. The hip motion is responsible for resetting the position of the wheel contact points for the next body motion. Figure 5.3 shows four separate configurations along a single cycle of rowing from trial 1. The three rowing trials may each be viewed in the MAMMOTH rover rowing video at https://youtu.be/as6j9xVNt_8. This rowing cycle is repeated between 8 and 9 times for each of the three rowing trials. The entire traverse and position of each of the wheel contact points along a single traverse is highlighted by the ground tracks left by the rover. Figure 5.4 shows these tracks from rowing trial 1.

Table 5.1 shows the distances travelled in each rowing and inchworming trial and compares these distances to kinematic simulations performed for each mobility mode. In each rowing

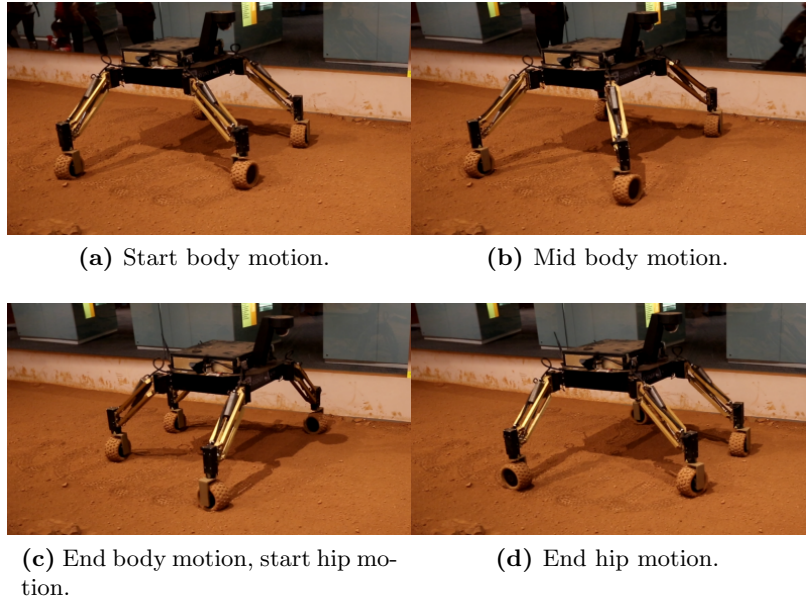


Figure 5.3 – A timelapse of a single rowing cycle.

Table 5.1 – The distances traversed by the MAMMOTH rover in each rowing and inchworming traversal as well as the kinematic simulations.

	Rowing				Inchworming		
	Trial 1	Trial 2	Trial 3	Sim.	Trial 1	Trial 2	Sim.
Dist. [m]	2.64	2.81	2.65	2.9	2.45	3.06	3.86
Num. Cycles	8	9	8	8	21	20.5	16
Dist. per Cycle [m]	0.33	0.31	0.33	0.36	0.12	0.15	0.24
Error [m]				0.04			0.10

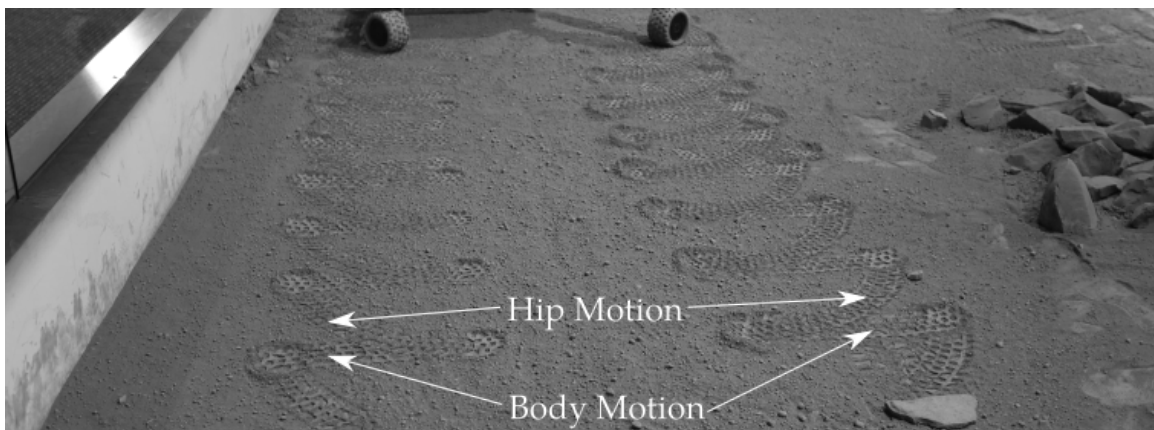


Figure 5.4 – The wheel tracks created by the MAMMOTH rover after performing a rowing manoeuvre. The tracks that make an arc are caused by hip motions, while the tracks that are orthogonal to the direction of travel of the robot are caused by body motions.

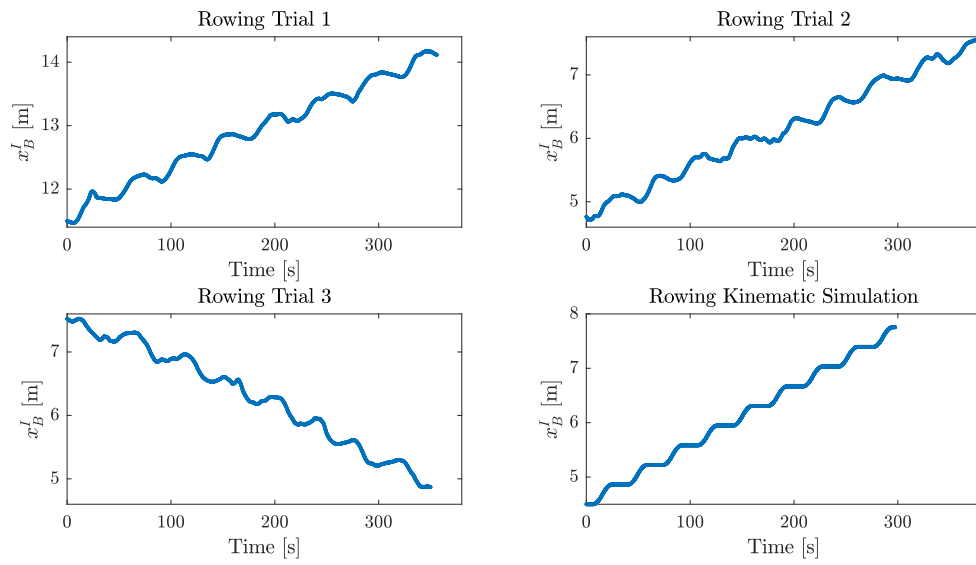


Figure 5.5 – The translation along the x_B^I direction recorded from the fiducial-based localization system for each rowing trial. Additionally, body translation from a kinematic simulation is presented.

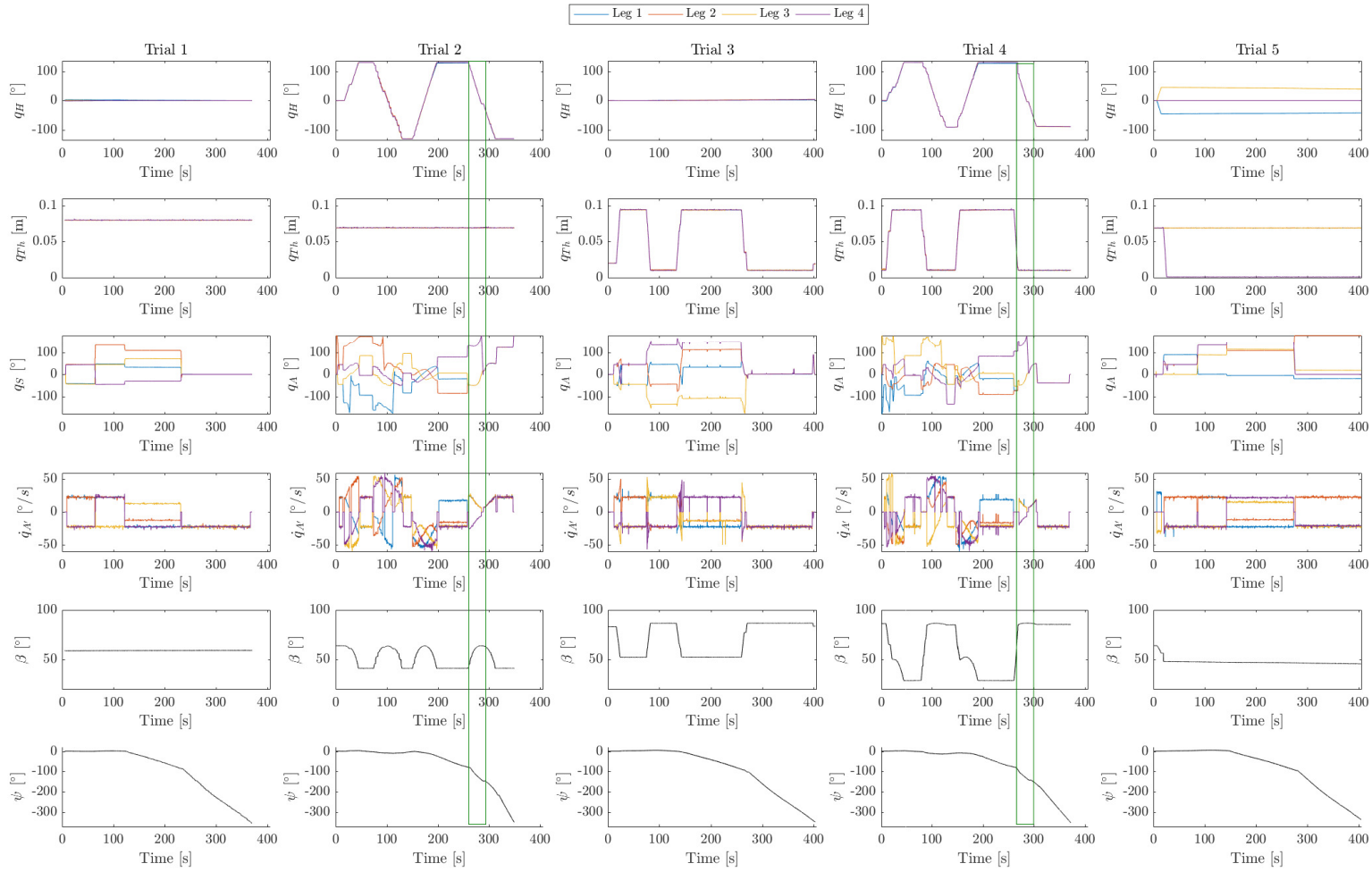


Figure 5.6 – Data collected from each of the five trials. The hip angles, q_H , thigh stroke positions, q_{Th} , steering joint angles, q_S , and wheel rotational rates, q'_A , are presented for each leg. Additionally, the stability estimate, β , and the heading, ψ , are shown for each experiment. The green boxes highlight areas specific data ranges of interest.

trial, a single cycle is responsible for an average body translation of 0.32 m. In the kinematic simulation the total body translation over 7 cycles is 2.82 m, resulting in 0.36 m per cycle. The 0.04 m discrepancy between the simulation and experiment is primarily caused by the hip motion part of the rowing cycle. It is observed in the rowing trial videos that the body moves in an opposite direction to the desired direction of travel during the hip motion. During this phase the rover's body should not be moving at all. This is also highlighted in Figure 5.5 where in each rowing trial the x_B^I translation changes slightly during the body motions. This unwanted translation may be caused by a discrepancy between where the kinematic model locates the wheel-ground contact points versus where they are in reality. This is most likely due to uncertainties in the system such as the wheel-ground contact being a contact patch as opposed to a point. Additional uncertainty comes from backlash within the thigh joint.

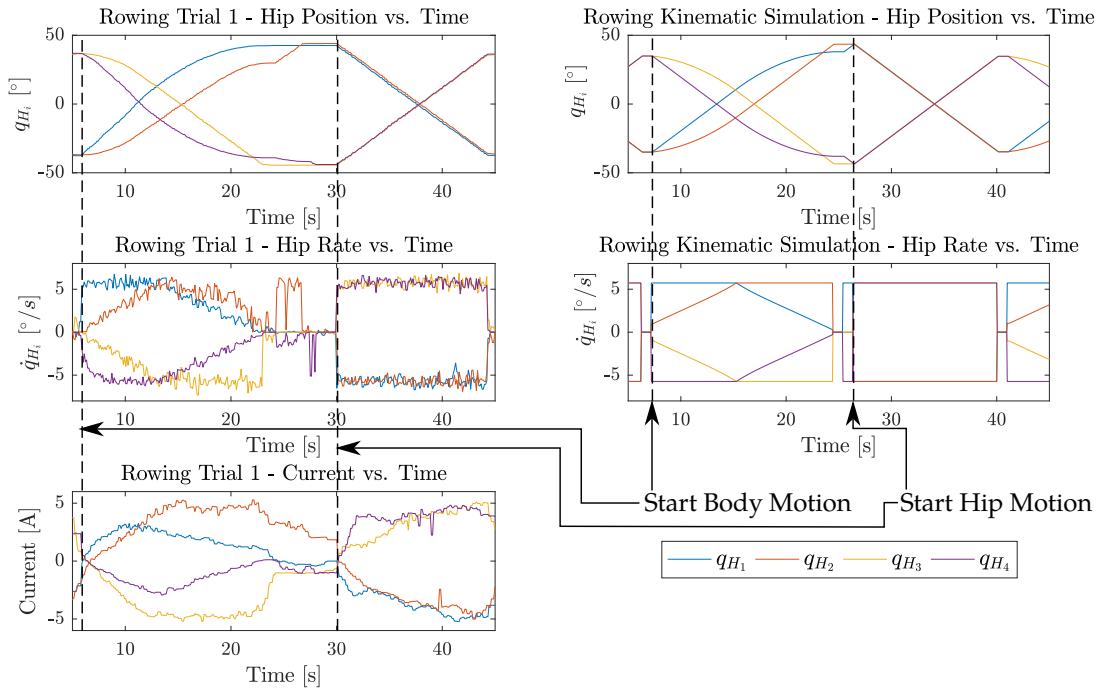


Figure 5.7 – The hip position, rates and currents from a representative rowing cycle from rowing trial 1 and a kinematic simulation. The cycle starts with a body motion and is followed by a hip motion.

A single representative rowing cycle is presented in Figure 5.7. The cycle starts with a body motion. Both the hip 1 and hip 4 actuators start driving at $6^\circ/\text{s}$, while hips 2 and 3 accelerate constantly until they get close to the 0° position. At $t = 14$ s in trial 1 and $t = 16$ s in the simulation, hips 1 and 4 begin to decelerate, while hips 2 and 3 attempt to drive at a constant speed of $6^\circ/\text{s}$. In simulation this is the case, however in trial 1, hip 2 is observed to lag behind hip 3. It is observed that hip 3 beats hip 2 to 0° , which in turn causes hip 3 to move faster than hip 2 for the rest of its motion. Overall, this does not impact the overall motion of the body, it just results in a slightly asymmetric leg motion. Another effect of this behaviour is observed in Figure 5.5, where the kinematic simulation traverse occurs 75 s faster than the trial 2, which has the same number of cycles. The physical

rowing trials are inevitably slower given that time has to be spent to correct the position of any actuators that have become out of sync. An example of this correction is observed between 24 s and 27 s in rowing trial 1, where hip 2 is driven to its terminal position after the other actuators have already reached their terminal positions.

5.2.3 Inchworming

The purpose of the inchworming manoeuvre is to propel the rover along a desired direction of travel by using both the thigh and wheel actuators. As described in Section 4.4.2, a single inchworming cycle is composed of a raise motion and a lower motion. The two inchworming trials may be viewed in the MAMMOTH rover inchworming video at <https://youtu.be/doDyYZ1lKRI>. Trial 1 illustrates the inchworming manoeuvre being used to move forward, while trial 2 is used to move the robot backwards. In each of these trials the rover's body frame moves at approximately 0.14 m per cycle, with a 0.07 m translation per raise or lower. Figure 5.8 shows a sequence of images taken from inchworming trial 1 that highlight a single raise/lower cycle.

Table 5.1 shows the traverse distances performed by the rover during each inchworming trial and compares them with a kinematic simulation. The simulated rover is shown to travel 0.22 m per cycle, on average 0.09 m more per cycle than in the trials. This discrepancy can be attributed to the large amount of slip witnessed at the anchored wheels during both raise and lower motions. The slip is easily observed in the video of each of the trials. Even though the objective of moving forward is achieved, the need for better anchoring to reduce slip is highlighted. Techniques for anchoring wheels may include adding grousers or spikes

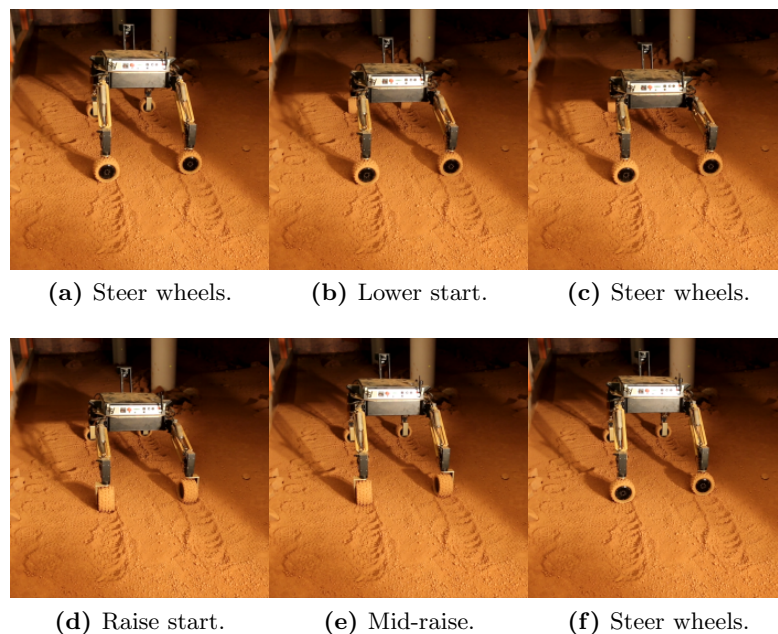


Figure 5.8 – A timelapse of a single inchworming cycle.

to the surface of the wheels and/or changing the orientation of the platform to distribute more weight on the moving wheels.

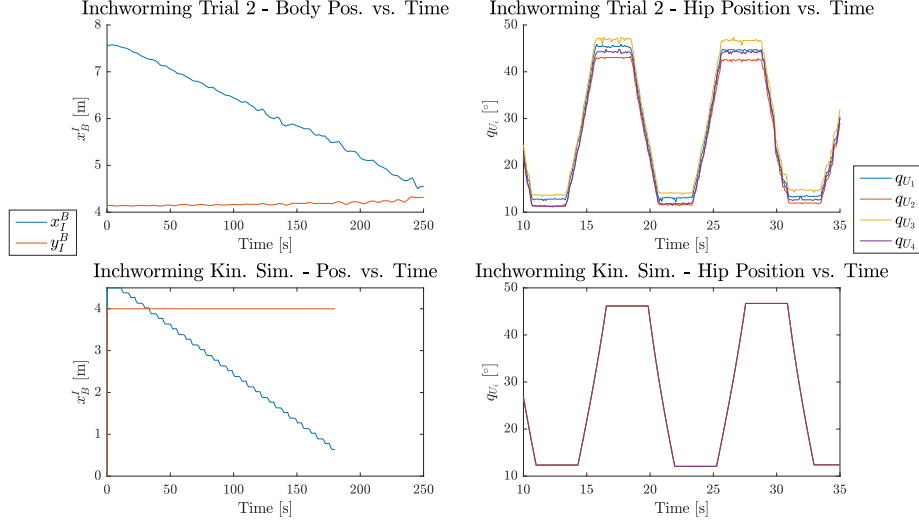


Figure 5.9 – The position of the rover’s body frame during Trial 2 of inchworming in addition to the thigh positions from two representative inchworming cycles.

Figure 5.9 shows shows thigh actuator position from a representative set of inchworming cycles from trial 2 and from the kinematic simulation. The translation of the rover’s body frame along the x_B^I and y_B^I axes is also shown. It is noted that there are discrepancies between the terminal thigh positions in trial 2. The effect of these differences may be seen in the y_B^I position, in that it increases over the traverse. This increase is due to cumulatively less thigh motion on the side of the robot with legs 1 and 2 than there is on the side of the robot with legs 3 and 4 over the entire traverse.

5.2.4 Energy Usage

It is envisioned that rowing or inchworming could be used to traverse regions of soft soil or inclines. The trade-off for using these modes over simple driving is that they are more energy intensive. In this section the energy usage of each mode is quantified. Current and voltage data is collected for each of the rover’s actuators. The energy usage of a single traverse is calculated as the sum of each actuator voltage, current and time-step duration product:

$$Q = \sum_{i=0}^m \sum_{t_j=1}^{t_f} I_i(t_j) V_i(t_j) (t_j - t_{j-1}), \quad (5.1)$$

where Q is the total traversal energy expenditure in Joules, $I_i(t_j)$ is the current drawn by actuator i at time t_j , $V_i(t_j)$ is the operating voltage of actuator i at time t_j , and m is the number of actuators.

Six driving trials were conducted in which the platform was driven either forwards or backwards over the same terrain type as was driven over in the rowing and inchworming trials. Table 5.2 shows the accumulated energy expenditure and energy to traverse distance ratios for each rowing, inchworming and driving trial. It is shown that the average energy to distance ratio is 47.5 kJ/m for rowing and 32.6 kJ/m for inchworming. These figures are an order of magnitude larger than driving which has an energy to distance ratio of 2.2 kJ/m. It is noted that the energy usage between driving trials varies, with the energy decreasing as the driving rate increases. This somewhat counter-intuitive behaviour may be explained by the hip actuator energy expenditure required to hold the desired zero degree hip positions as the rover is driving. At smaller driving velocities, the hip actuators have to hold their positions and expend energy for a longer time than at larger driving velocities over the same traversal distance.

The results presented in Table 5.2 demonstrate the energy efficiency of a simple driving mode compared to alternative techniques. A topic of future work is to test the alternative mobility modes on terrain types that simple driving may not work on such as deep soft sandy soils with inclines.

The presented alternative mobility mode trials validate the kinematic model for each mode presented in Chapter 4. Discrepancies between the kinematic model and the physical trials are attributed to affects not accounted for in the kinematic model such as wheel deformation at the contact point, thigh backlash and wheel slippage. The average energy usage of the platform in each mobility mode is also quantified demonstrating that the proposed alternative mobility modes required significantly more energy than driving with a fixed footprint.

Table 5.2 – Energy usage by the MAMMOTH rover’s mobility system during rowing, inchworming and driving traverses.

Trial	Rowing				Inchworming		
	1	2	3	Avg.	1	2	Avg.
Distance	2.64	2.81	2.65	2.7	2.45	3.06	2.76
Speed [mm/s]	7	7	8	7	10	12	11
Energy [kJ]	126.8	138.3	120.2	128.4	89.2	88.0	89.2
Energy/Dist. [kJ/m]	48.0	49.3	45.3	47.5	36.4	28.8	32.6
Trial	Driving						
	1	2	3	4	5	6	Avg.
Distance	2.70	2.70	2.69	2.70	2.71	2.76	2.71
Speed [mm/s]	59	75	116	113	151	153	111
Energy [kJ]	9.3	7.5	5.3	5.3	3.7	4.2	5.9
Energy/Dist. [kJ/m]	3.4	2.8	2.0	2.0	1.4	1.5	2.2

5.3 Actively Articulated Suspension Demonstration

This section provides experimental results from three trials in which the MAMMOTH rover performs successful traverses over a step obstacle using the actively articulated suspension technique presented in Section 4.5 and [92]. The rover is given a desired body velocity along the x_I -axis relative to the inertial frame and changes its leg geometries to traverse the step obstacle. The experimental setup along with discrepancies between desired and actual rover behaviour are discussed. Videos of the three obstacle traversals presented may be viewed at <https://goo.gl/hQuMx3>.

5.3.1 Experimental Setup

During each attempted traverse the MAMMOTH rover moves over a bed of rocks that surround a 0.15 m tall cinder block, shown in Figure 5.10. In trials 1 and 2 the rover initially drives 4 m so as to allow each of its wheels to be within the mapped terrain region, whereas in trial 3 only 3 m is initially driven. Once all wheels are within this region, the active articulation controller is activated by an operator. The initial configuration of the robot for both trials has each of the hip joints at $q_{H_i} = 0^\circ$. The desired pose to be kept for each trial is zero translational motion along the y_I - and z_I -axes, and zero rotational motion for ϕ , θ , ψ , and \underline{q}_H . In trial 1 and trial 2, the desired rate of the rover is 0.05 m/s along the x_I axis, while in trial 3 it is 0.04 m/s. The lookahead distance, d_{look} , as described in Eq. (4.42) is set to 0.3 m for trial 1 and 0.1 m for trial 2 and 3. The inertial frame is defined at the base of the rover at the beginning of its traverse. The x_I and y_I axes are coincident with the body frame's x_B and y_B initial positions. The z_I axis originates from the average initial height of each of the wheel contact frames relative to the body frame. The roll, pitch and yaw angles are presented for each trial. In this section, these Euler angles are extrinsic rotations (elemental rotations that occur about a fixed coordinate system) as opposed to the intrinsic rotations used in the rest of the thesis. The reason for the use of this representation is for easier data visualization.

5.3.2 Results and Discussion

During the first trial the rover traverse over the block obstacle by raising and lowering legs 1 and 2 to meet the obstacle profile. A timelapse of the trial is presented in Figure 5.10. Figure 5.11 shows the RGB-D based translation and orientation of the vehicle during its traverse. Orientation data from an on-board IMU is also provided. A video of the trial is available at <https://youtu.be/uN1uU-eD5nI>. Assuming the IMU orientation data to be ground truth, discrepancies in the RGB-D sensor orientation solution are apparent. For the entirety of the traverse the pitch data from the two localization systems are similar and indicate the largest pitch to be no greater than 7° . During the initial approach to the obstacle between 0 s and 50 s it is observed that the yaw deviates by 2.5° in the IMU solution. This deviation is due to operator re-alignment commands to ensure that the obstacle is traversed. According to the RGB-D localization solution the yaw is approximately four times larger. A similar drift is seen on the roll axis.

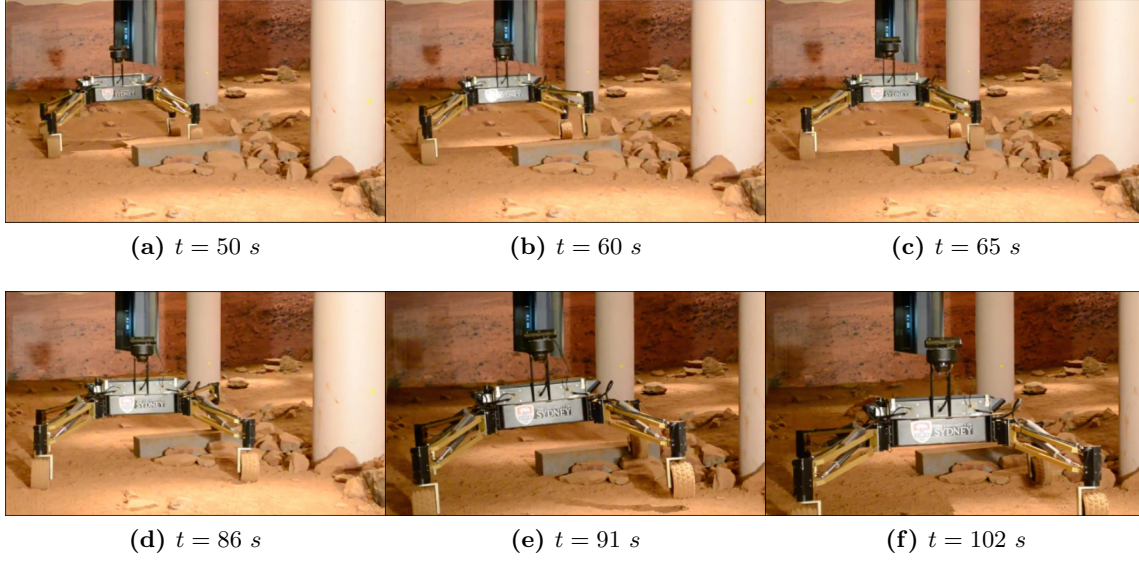


Figure 5.10 – A sequence of images from trial 2, in which the MAMMOTH rover traversed a step obstacle with its left legs. The images sequence progresses from left to right by row.

During the obstacle traverse, between 50 s and 102 s, multiple unwanted behaviours are observed. Wheel 1 starts by raising to meet the terrain profile. A slight pitch and roll deviation each of 1° is observed. Between 60 s and 65 s as wheel 1 is driving off the block surface the linear actuator begins to actuate too soon. From the wheel contact position plot in Figure 5.13, it can be seen that the wheel 1 contact point profile, between 4 m and 4.25 m along the x_I -axis, rises to a peak as opposed to following the profile of a block with a flat top. At 60 s, the rover attempts to conform to the downward slope of the peak when it is actually still on top of the block. This causes wheel 4 to lift off the ground between 63 s and 74 s. A similar behaviour is observed when wheel 2 drives off the block. In this case, leg 1 is raised off the ground between 87 s and 94 s as a result. These behaviours may be attributed to the back section of the obstacle being shielded from the view of the RGB-D sensor. The k-nearest neighbour contact point interpolation results in a terrain profile with a sloped surface as opposed to a step change. Additionally, the use of a lookahead distance of 0.3 m causes the wheels to be pushed down slightly early.

A cause of the drift in the orientation solution from the RGB-D localization was the lighting conditions present in the environment. Glare from a spotlight positioned behind the camera views in Figure 5.10 has a detrimental affect on the sensor. For this reason, more homogeneous lighting conditions are introduced in trial 2.

During the second trial the MAMMOTH rover traverses the rock obstacle once again with both legs 1 and 2. A timelapse of the trial 2 traverse is presented in Figure 5.12 along with a ground track and orientation data presented in Figure 5.13. A video of the trial is available at <https://youtu.be/Ym2yGejuNbo>. In this trial the rover drives along the x_I -axis at 0.05 m/s with a shorter lookahead distance of 0.1 m. Overall, this trial results in an improved traversal performance. Unlike trial 1, for the most part all four wheels stay in contact with the ground, resulting in a more fluid obstacle traversal. The only exception is

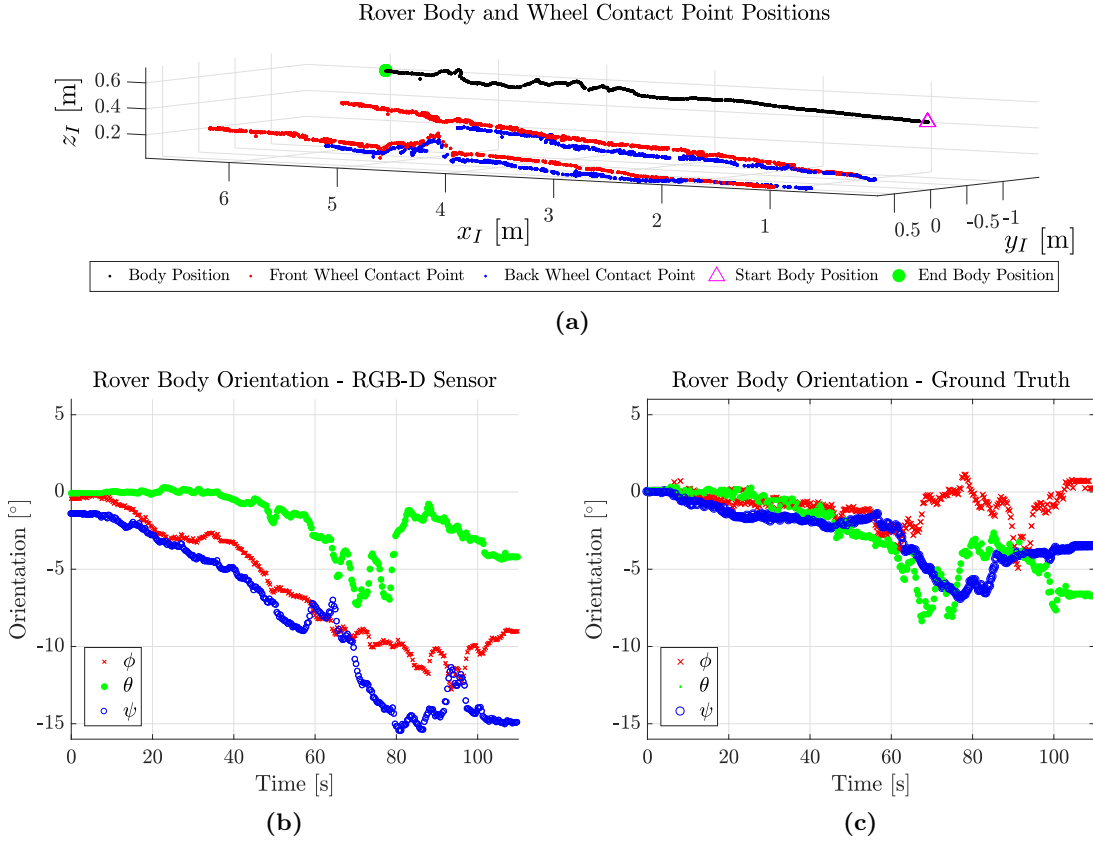


Figure 5.11 – Position and orientation plots of the MAMMOTH rover from trial 1.

between 70 and 73 s when wheel 3 lifts off the ground due to wheel 1 not being pushed down enough to follow the end of the step obstacle. This may be due to too short a lookahead distance. A smaller drift in the roll solution is seen in this trial, most likely due to improved lighting conditions. The yaw motion of the platform also decreases between 0 s and 50 s due to yaw re-alignments by the operator. During the obstacle traverse a slight yaw is observed as shown in the orientation data. The yaw stabilizes once wheel 2 has finished traversing the obstacle.

In trial 3 only leg 1 is raised over the obstacle as leg 2 misses the obstacle due to the initial yaw of the robot. A timelapse of the trial is shown in Figure 5.14, while ground-track and localization data is shown in Figure 5.15. Video of trial 3 is available at <https://youtu.be/v5N-3hFQHq0>. In this trial the speed along the x_I -axis is set slower at 0.04 m/s and the lookahead distance is 0.1 m. The effect of these changes is noticeable with leg 1 being lowered too late as the wheel drives off the obstacle. An undesirable drift that does not agree with the actual orientation of the robot is noted in this trial.

In summary, the actively articulated suspension technique is demonstrated to work with the physical MAMMOTH rover in the three trials presented. The technique uses RGB-D sensor data to localize the platform and map the terrain being traversed. Even though

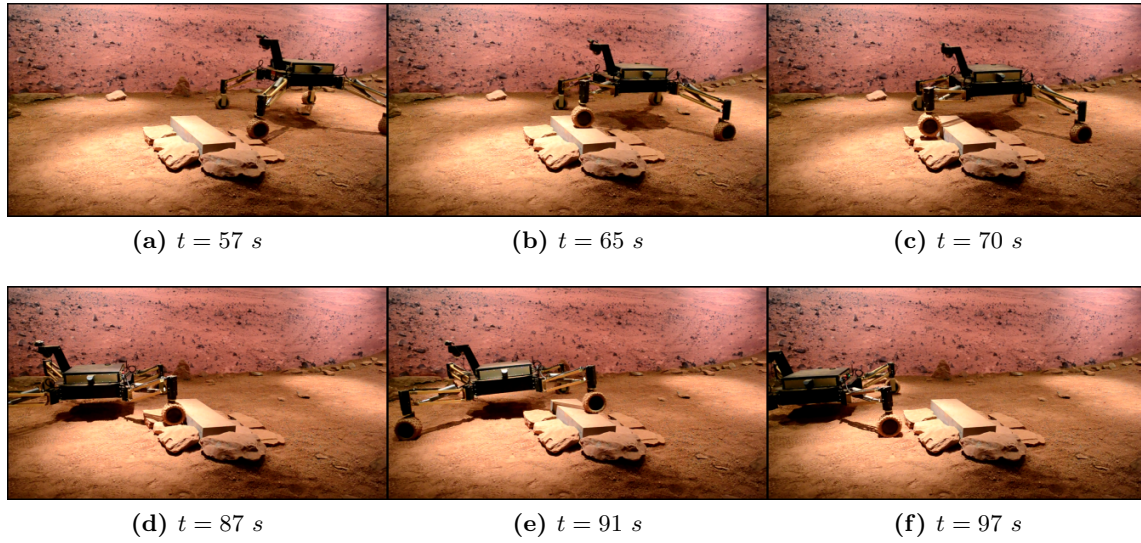


Figure 5.12 – A sequence of images from trial 1, in which the MAMMOTH rover traversed a step obstacle with its left legs. The images sequence progresses from left to right by row.

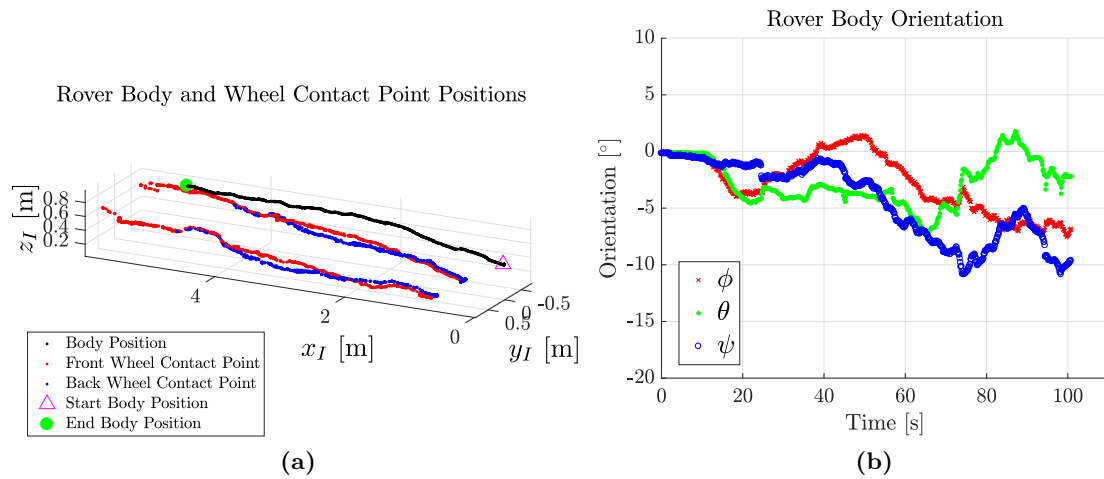


Figure 5.13 – Position and orientation plots of the MAMMOTH rover from trial 2.

the technique is demonstrated to work, a more robust localization and control scheme is desired. The localization solution is susceptible to changes in lighting, which causes drift in the orientation solution. Within the control scheme a direct measurement of wheel contact using the leg force sensors would be beneficial and is an item of future work.

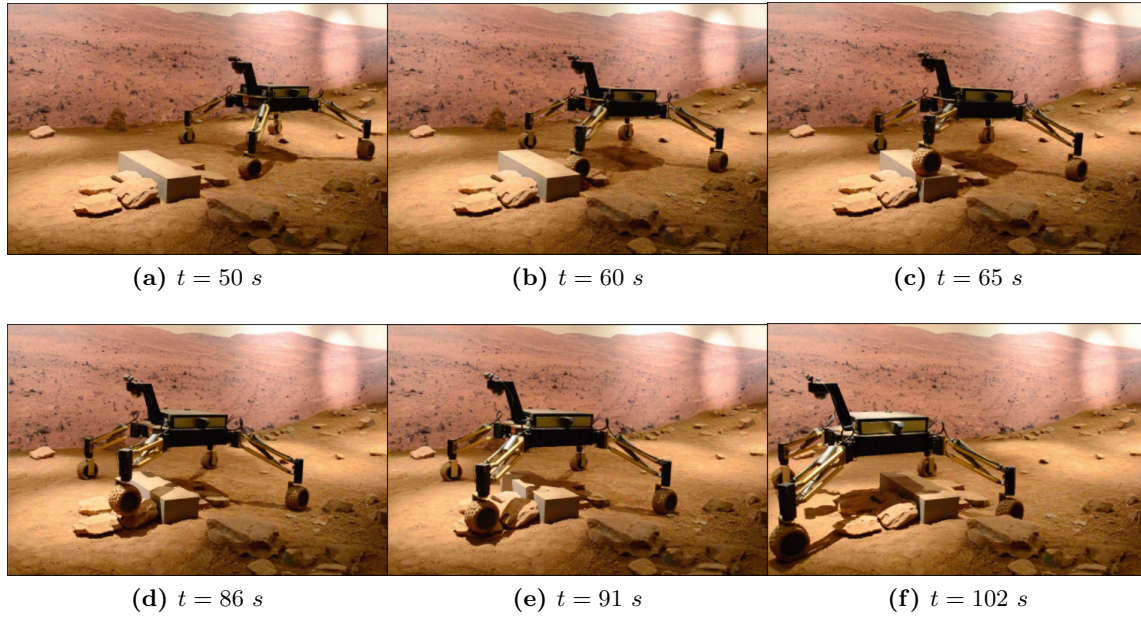


Figure 5.14 – A sequence of images from trial 3, in which the MAMMOTH rover traversed a step obstacle with its left leg. The images sequence progresses from left to right by row.

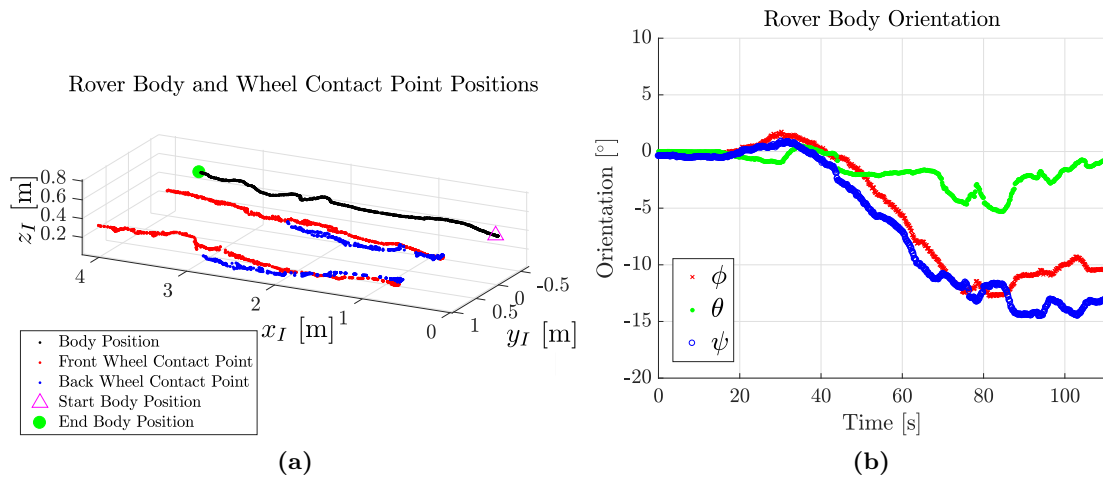


Figure 5.15 – Position and orientation plots of the MAMMOTH rover from trial 3.

Chapter 6

Hierarchical Sampling-Based Motion Planning for RWMRs

As shown in the previous chapters, RWMRs are platforms that gain versatility through the ability to dynamically alter certain elements of their geometric structure. This versatility is demonstrated by the novel locomotion modes of the MAMMOTH RWMR described in Chapter 4. This platform is novel in how it combines wheeled and legged locomotion modes to traverse over, under or around challenging obstacles. It may drive pseudo-omnidirectionally while raising and lowering its body, packing or unpacking each of its legs between large and small footprints, as well as lifting or lowering individual legs to clamber over terrain. Even though the rover may need to reconfigure to achieve a successful traversal, the dominant mode of motion remains the translation of its body frame relative to an inertial frame. A hierarchical motion planning technique that exploits this property by rapidly searching the dominant sub-space to find paths that are then used to bias search within the full state space is proposed.

The main challenge in planning for RWMRs is to efficiently find a feasible low cost path given potentially many degrees of freedom. The problem is to generate a sequence of transitions between initial and terminal system states while both avoiding environmental obstacles and satisfying internal kinematic constraints. The sequence of transitions is to be optimized with respect to a cost function that is used to define the objectives of the robot's overall motion. In this thesis an approximation of the energy expenditure of the robot is minimized.

Sampling-based planners are a class of path planning algorithms that are not as susceptible to the *curse of dimensionality* as their discrete grid-based planning or artificial potential field counterparts. With asymptotically optimal (AO) sampling-based planners it is feasible to plan paths for high degree of freedom robots and converge towards an optimal solution. This thesis describes a modification to existing AO sampling-based planners that both preserves AO and exploits prior knowledge of task structure to promote faster convergence. Additionally, the proposed planning architecture assists in the search for narrow passageways, a notable problem for many sampling-based planners. This approach enables the use of prior knowledge of kinematic structure to inform the choice of hierarchical decomposi-

tion, but defined in a way that is more flexible than in other hierarchical approaches. The state-space decomposition must still be provided a priori, but the subtasks are continuous and generated probabilistically by the planner. The extent of the state space explored by subtasks is defined by continuous parameters that may be automatically tuned.

A main technical challenge in developing a hierarchical sampling-based planner is to define the information flow between levels in a way that does not sacrifice completeness and optimality guarantees. The planner decomposes the state space as a hierarchy of increasingly larger sub-spaces, each searched by a Fast Marching Tree (FMT*) variant. Solutions from adjacent levels are used to bias search, thereby increasing path quality with equivalent computational resources. This technique is suited to platforms that have intuitively identifiable low-dimensional kinematic structure, such as RMRs, but may also generalize to other high-DOF robots such as redundant manipulator arms.

It is shown that this approach retains the completeness and asymptotic optimality guarantees of FMT* [60] by virtue of a fraction of its samples, ℓ being taken from a uniform distribution. Experimental results in simulation show that HBFMT* outperforms FMT*, BMFT*, RRT*, Informed RRT* and BIT* in a variety of environments.

6.1 Problem Formulation and Background

In this section the path planning problem for RMRs is formulated. Notation that agrees with previous work relevant to FMT* in [60] and [118] has been used.

6.1.1 Problem Definition

Let $\mathcal{X} = [0, 1]^d$ be a state space with dimension $d \in \mathbb{N}$ constrained to $d \geq 2$. Let \mathcal{X}_{obs} be the obstacle region such that $\mathcal{X} \setminus \mathcal{X}_{obs}$ is an open set. This implies that the free state space is a closed set $\mathcal{X}_{free} = \text{cl}(\mathcal{X} \setminus \mathcal{X}_{obs})$. A set of n states are randomly sampled from \mathcal{X}_{free} . A state \underline{x} is a vector that defines a discrete point within \mathcal{X} . Let $\mathcal{X}^\# = [0, 1]^e$ be a sub-space that is embedded within \mathcal{X} where $2 \leq e \leq d$. Any variable with the $\#$ super-script is used to denote a variable related to the sub-space $\mathcal{X}^\#$.

A path planning problem is denoted by a triplet $(\mathcal{X}_{free}, \underline{x}_{init}, \mathcal{X}_{goal})$ where \underline{x}_{init} is the initial state and \mathcal{X}_{goal} is the goal region. In this work, the goal region is considered as a single state, described by \underline{x}_{goal} . A path is defined by a continuous mapping $\sigma : [0, 1] \rightarrow \mathbb{R}^d$ such that $0 \mapsto x_1$ and $1 \mapsto x_2$. Let Σ be the set of all paths in \mathcal{X} . A feasible path, π , in the planning problem $(\mathcal{X}_{free}, \underline{x}_{init}, \mathcal{X}_{goal})$ is a path that is collision free with $\pi(0) = \underline{x}_{init}$ and $\pi(1) = \underline{x}_{goal}$. $\Pi = \{\pi_0, \pi_1, \dots, \pi_k\}$ is a set of k feasible paths sorted according to path cost.

The optimal path planning problem is to find a path π^* , assuming $(\mathcal{X}_{free}, \underline{x}_{init}, \mathcal{X}_{goal})$ and an arc cost function $c : \Sigma \rightarrow \mathbb{R}_{\geq 0}$ such that $c(\pi^*) = \min \{c(\pi)\}$, or to report failure. The optimal path π^* is δ -robustly feasible if every point along it is a minimum of δ away from \mathcal{X}_{obs} .

Algorithm 2: Bidirectional Fast Marching Tree (BFMT*)

```

function BFMT* ( $\mathcal{X}_{free}, \underline{x}_{init}, \underline{x}_{goal}, \Pi, \ell, r_{singe}, r_{tunnel}, tc$ )
1   $S \leftarrow \underline{x}_{init} \cup \underline{x}_{goal} \cup \text{SampleFree}(n, \Pi, \ell, r_{tunnel})$ 
2   $T \leftarrow \text{Initialize}(S, \underline{x}_{init}), T' \leftarrow \text{Initialize}(S, \underline{x}_{goal})$ 
3   $\Pi \leftarrow \emptyset, \underline{z} \leftarrow \underline{x}_{init}, \underline{x}_{collision} \leftarrow \emptyset, pathFound \leftarrow false$ 
4   $success = false$ 
5  while  $success = false$  do
6     $\text{Expand}(T, \underline{z}, \underline{x}_{collision})$ 
7    if  $(tc = \text{FIRST and } \underline{x}_{collision} \neq \emptyset)$  or  $(tc = \text{BEST and } \underline{z} \in (V' \setminus H'))$  then
8       $\pi \leftarrow \text{Path}(\underline{x}_{collision}, T) \cup \text{Path}(\underline{x}_{collision}, T'), \Pi \leftarrow \Pi \cup \pi$ 
9       $success = true$ 
10   else if  $tc = \text{EXHAUSTIVE and } \underline{z} \in (V' \setminus H')$  then
11      $\pi \leftarrow \text{Path}(\underline{x}_{collision}, T) \cup \text{Path}(\underline{x}_{collision}, T'), \Pi \leftarrow \Pi \cup \pi$ 
12      $\text{SingeBranch}(\underline{x}_{collision}, T, r_{singe}), \text{SingeBranch}(\underline{x}_{collision}, T', r_{singe})$ 
13      $\underline{x}_{collision} \leftarrow \emptyset, pathFound \leftarrow true$ 
14   if  $H = \emptyset$  and  $H' = \emptyset$  and  $pathFound = false$  then
15     return Failure
16   else if  $H = \emptyset$  and  $H' = \emptyset$  and  $pathFound = true$  then
17      $success = true$ 
18      $\underline{z} \leftarrow \text{argmin}_{\underline{x}' \in H'} \{\text{Cost}(\underline{x}', T')\}$ 
19      $\text{Swap}(T, T')$ 
20 return  $\Pi$ 

```

6.1.2 Bidirectional Fast Marching Tree

In general, the usage of bidirectional search within the planning problem may reduce the amount of required exploration [70]. Additionally, bidirectional search has been shown to aid in the discovery of narrow passageways for sampling-based planners [61], [118]. Within the hierarchical planning framework proposed in this thesis, bidirectional search is employed for this reduction in exploration while searching \mathcal{X}_{free} . It is also used to find multiple candidate paths while searching $\mathcal{X}_{free}^\#$ as described in Section 6.2. Bidirectional Fast Marching Tree (BFMT*) is a bidirectional variant of FMT* proposed in [118]. Pseudo-code for the variant of BFMT* used in this thesis is provided in Algorithm 2. The FMT* algorithm is also described in Section 2.3.3.

BFMT* grows two FMT*s, $T = (V, E)$ and $T' = (V', E')$, rooted at the start and goal nodes respectively. BFMT* starts by sampling a set of states, S , from \mathcal{X}_{free} . The two trees are then initialized. The samples S , the start state \underline{x}_{init} and the goal state \underline{x}_{goal} are first inserted into the V set of T , with \underline{x}_{init} as the root node. The states S , \underline{x}_{init} and \underline{x}_{goal} are also inserted into V' of T' , with \underline{x}_{goal} at its root. Each tree is alternately expanded using the **Expand** function. Pseudo-code for this function is presented in Algorithm 3 and in [118].

The termination condition of BFMT* may occur in one of two separate conditions [118]. The first is when the two trees initially intersect, occurring when a node is in the open set

Algorithm 3: Fast Marching Tree (FMT*) Expansion [118]

```

function Expand( $T = (V, E, H, W), T' = (V', E', H', W'), z, \underline{x}_{collision}$ )
1   $H_{new} \leftarrow \emptyset$ 
2   $Z_{near} \leftarrow \text{Near}(V \setminus \{z\}, z, r_n) \cap W$ 
3  for  $x \in Z_{near}$  do
     $X_{near} \leftarrow \text{Near}(V \setminus \{x\}, x, r_n) \cap H$ 
     $\underline{x}_{min} \leftarrow \text{argmin}_{\underline{x}' \in X_{near}} \{\text{Cost}(\underline{x}', T') + \text{Cost}(\overline{x'x})\}$ 
    if CollisionFree( $\underline{x}_{min}, x$ ) then
4       $E \leftarrow E \cup \{(x_{min}, x)\}$ 
5       $H_{new} \leftarrow H_{new} \cup \{x\}$ 
6       $W \leftarrow W \setminus \{x\}$ 
      if ( $x \in H'$  or  $x \in W'$ ) and ( $\text{Cost}(x, T) + \text{Cost}(x, T') < \text{Cost}(\underline{x}_{collision}, T)$ 
        +  $\text{Cost}(\underline{x}_{collision}, T')$ ) then
7         $\underline{x}_{collision} \leftarrow x$ 
8   $H \leftarrow (H \cup H_{new}) \setminus \{z\}$ 
  return  $T = (V, E, H, W), \underline{x}_{collision}$ 

```

Algorithm 4: Initialize a Fast Marching Tree [118]

```

1 function Initialize( $S, x_o$ )
2    $V \leftarrow S$ 
3    $E \leftarrow \emptyset$ 
4    $W \leftarrow V \setminus \{x_o\}$ 
5    $H \leftarrow \{x_o\}$ 
6   return  $T = (V, E, H, W)$ 

```

of both trees. This is the ‘first’ path termination condition, however it does not guarantee an optimal solution. The second termination condition occurs when a node is in the open set of one tree, while not inside the open or unvisited set of the other. This second ‘best’ path condition results in a case where the node can no longer improve its cost-to-come value from either tree root and is therefore the optimal solution given the sampled nodes.

6.2 Hierarchical Bidirectional Fast Marching Tree

Intuitively, HBFMT* biases search of the full state space based on knowledge gathered from an initial rapid search of a sub-space. To introduce HBFMT* an example with the MAM-MOTH rover constrained to an 8-DOF space \mathcal{X} operating within a confined environment is shown in Figure 6.1. The robot starts in an open-stance state (Figure 6.1b) at the diamond position, while its goal is to achieve a similar stance at the star position both shown in the map shown in Figure 6.1c. Any feasible path requires reconfiguration of the robot to either traverse over and around block obstacles or clamber over step obstacles. Initially, as in Figure 6.1c, a sub-space $\mathcal{X}_{free}^\#$ in \mathbb{R}^3 is searched using BFMT*. The sub-space describes

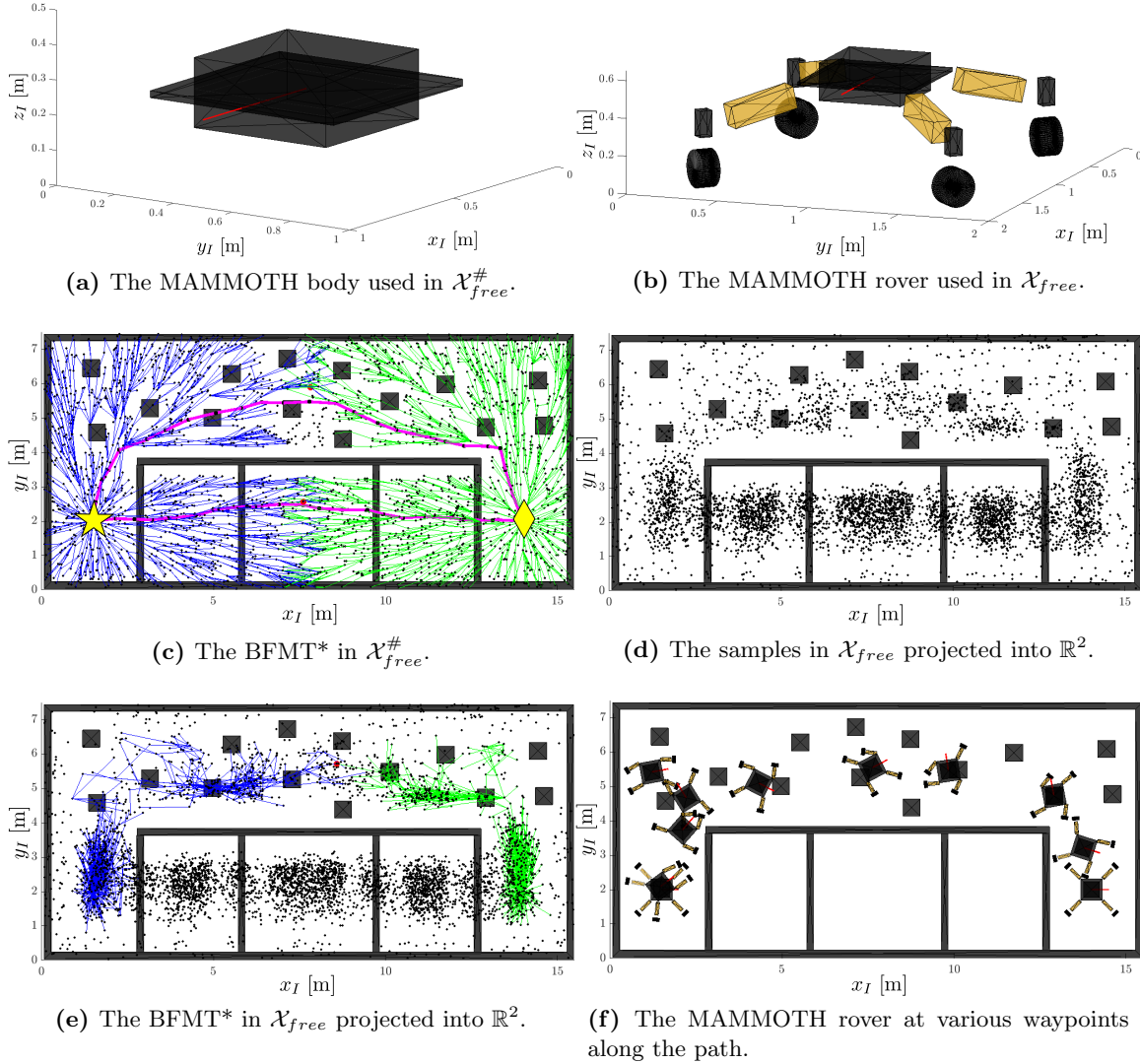


Figure 6.1 – A planning problem is addressed, which plans for the MAMMOTH rover shown in (a) and (b). In the HBFMT* algorithm an initial exhaustive BFMT* search (c) of a uniformly sampled sub-space returns a set of feasible paths. In (d) states from the full space are sampled from a biased distribution focused around the sub-space paths. A second BFMT* shown in (e) searches through the full state samples. Lastly, in (f) a full state-space path is returned.

the translation of the robot's body with respect to the inertial frame. Collision checking is performed between the environment and the translating body of the robot without its legs, shown in Figure 6.1a. A set $\Pi^\#$, composed of feasible paths through $\mathcal{X}_{free}^\#$ is returned as shown in Figure 6.1c. The full 8-DOF state space of the robot is then sampled with a bias around the set of returned paths through $\mathcal{X}_{free}^\#$ as shown in Figure 6.1d. Another BFMT* instance then searches this biased distribution as in Figure 6.1e. The bias is implemented using a tunable *tunnel radius*, r_{tunnel} that defines how focused the search of the solutions from the previous level of the hierarchy is. A final path that traverses around and over the box obstacles is returned and shown in Figure 6.1f.

Algorithm 5: Hierarchical Bidirectional Fast Marching Tree (HBFMT*)**Data:** $\mathcal{X}_{free}, \underline{x}_{init}, \underline{x}_{goal}, r_{singe}, r_{tunnel}$ **Result:** Π

```

1 begin
2    $\Pi^\# \leftarrow \text{BFMT}^*(\mathcal{X}_{free}^\#, \underline{x}_{init}^\#, \underline{x}_{goal}^\#, \emptyset, 1, r_{singe}, r_{tunnel}, \text{EXHAUSTIVE})$ 
3    $\Pi \leftarrow \text{BFMT}^*(\mathcal{X}_{free}, \underline{x}_{init}, \underline{x}_{goal}, \Pi^\#, \ell, \emptyset, r_{tunnel}, \text{BEST})$ 

```

Algorithm 6: Singe Branch

```

1 function SingeBranch( $z, T, r_{singe}$ )
2   /* Perform a breadth first search of the branch of the sub-tree
      with its root at node z.                                     */
3    $A \leftarrow \text{BreadthFirstSearch}(z, T, T')$ 
4    $A \leftarrow A \cup \text{Near}(T, A, r_{singe})$ 
5    $H \leftarrow H \setminus A, V \leftarrow V \setminus A$ 
6    $H' \leftarrow H' \setminus A, V' \leftarrow V' \setminus A$ 

```

Before HBFMT* is defined, a modification to the termination condition of BFMT* is described. An ‘exhaustive’ termination condition is introduced; pseudo-code is provided in Algorithm 2. When a ‘best’ or ‘first’ termination condition is found at a tree collision node $x_{collision,i}$, BFMT* returns the feasible path π_i , where i is the path index, and adds it to the set of alternative paths $\Pi^\#$. When a collision node is found it is singed. All of its descendant nodes are found via a breadth first search starting from the collision node. All nodes within a distance r_{singe} of the collision node or its descendants are removed from the open and unvisited sets so that tree expansion may not continue through them. This singeing procedure is performed in both FMT*s. Pseudo-code for the singeing procedure is presented in Algorithm 6. BFMT* continues this search and singe procedure until both trees’ open sets are exhausted.

HBFMT* is now formalized. Pseudo-code is listed in Algorithm 5. The first step is to plan a path through the sub-space $\mathcal{X}_{free}^\#$ using an exhaustive BFMT*, sampling $n^\#$ nodes uniformly from $\mathcal{X}_{free}^\#$. Depending on the space $\mathcal{X}_{free}^\#$ and the performance of the BFMT* search, there may be multiple unique paths $\Pi^\#$ returned. An example where two alternative paths are found is shown in Figure 6.1c.

The set of paths $\Pi^\#$ is then passed to a second BFMT* instance along with a sampling variable ℓ . This second BFMT* search starts by sampling n states from \mathcal{X}_{free} to create the tree nodes V . A collection of $n\ell$ samples are taken from a uniform distribution over \mathcal{X}_{free} to comply with asymptotic optimality conditions discussed in Section 6.3. Every second path within $\Pi^\#$ is reversed and then all of the paths within $\Pi^\#$ are concatenated to form a path λ that goes back and forth between the goal and end nodes. The sub-space components of $n(1 - \ell)$ states are sampled from a Gaussian distribution $\mathcal{N}(d_\lambda, r_{tunnel})$ where d_λ is a uniformly sampled distance along λ . The parameter r_{tunnel} , the tunnel radius, determines how wide the biased search region around λ is. Currently, this variable is tuned manually,

Algorithm 7: Sample Free

```

1 function SampleFree( $n, \Pi, l, r_{tunnel}$ )
2    $\lambda \leftarrow []$ 
3   for  $i = 1$  to  $|\Pi|$  do
4     if  $i \bmod 2 = 0$  then
5        $\lambda \leftarrow [\lambda \quad \text{Reverse}(\Pi_i)]$ 
6     else
7        $\lambda \leftarrow [\lambda \quad \Pi_i]$ 
8    $m \leftarrow n$ 
9   while  $m > 0$  do
10    if  $m < (1 - \ell)n$  then
11       $\underline{x} \leftarrow \text{SampleBiased}(\lambda, r_{tunnel})$ 
12    else
13       $\underline{x} \leftarrow \text{SampleUniform}$ 
14    if IsValid( $\underline{x}$ ) then
15       $m \leftarrow m - 1, S \leftarrow S \cup \underline{x}$ 
16  return  $S$ 

```

Algorithm 8: Sample from the Biased Distribution.

```

1 function SampleBiased( $\lambda, r_{tunnel}, c_\lambda$ )
2    $d \leftarrow \mathcal{U}(0, c_\lambda)$ 
3   for  $i = 0$  to  $|\lambda| - 2$  do
4     if  $c(\lambda(:, 0 : i + 1)) > d$  then
5        $t \leftarrow J(\underline{\lambda}_i, \underline{\lambda}_{i+1})$  /* Column vector notation:  $\underline{\lambda}_i = \lambda(:, i)$  */
6        $u \leftarrow d - c(\lambda(:, 0 : i))$ 
7        $v \leftarrow u/t$ 
8        $\underline{y} \leftarrow \underline{\lambda}_i + (1 - v)\underline{\lambda}_i + v\underline{\lambda}_{i+1}$ 
9        $\underline{x} \leftarrow \mathcal{N}(\underline{y}, r_{tunnel})$ 
10      break
11  return  $\underline{x}$ 

```

however in future work it is desired that this variable be based on results from multiple searches of $\mathcal{X}_{free}^\#$. An example of a set of sampled nodes in \mathcal{X}_{free} projected into \mathbb{R}^2 is shown in Figure 6.1d. Once the biased set of samples S has been found, the second BFMT* instance explores the set and returns the best path through the resulting tree as shown in Figure 6.1f.

6.3 Analysis

In this section the asymptotic optimality of HBFMT* is shown and the effects of its tuning parameters are discussed. AO (and thus, probabilistic completeness) of HBFMT* is proved by observing that the terminal stage of the algorithm is BFMT* with a non-uniform sampling distribution and a metric cost function. The following theorem characterizes AO in terms of the number of sample nodes.

Theorem 6.1. *Let $\pi : [0, 1]$ be a feasible path with strong δ -clearance, $\delta > 0$. ζ is the Lebesgue measure of the unit-cost ball and $\mu(\mathcal{X}_{free})$ is the Lebesgue measure of the free space. Consider running HBFMT* by sampling n nodes from a distribution φ and executing to completion using any termination criteria and a radius*

$$r_n = 2(1 + \eta) \left(\frac{1}{d} \right)^{\frac{1}{d}} \left(\frac{\mu(\mathcal{X}_{free})}{\zeta} \right)^{\frac{1}{d}} \left(\frac{\log(n\ell)}{n} \right)^{\frac{1}{d}} \left(\frac{1}{\ell} \right)^{\frac{1}{d}}, \quad (6.1)$$

for a parameter $\eta \geq 0$ and $n\ell > 1$. Let c_n denote the cost of the path returned by HBFMT* and c^* be the optimal path cost, then $\lim_{n \rightarrow \infty} \mathbb{P}(c_n > (1 + \varepsilon)c^*) = 0$ for all $\varepsilon > 0$ (which defines AO).

Proof. The terminal stage of HBFMT* implements BFMT* operating in \mathcal{X}_{free} , using a metric cost function and samples from probability distribution φ that is lower bounded by ℓ . To prove AO of HBFMT* it therefore suffices to show that BFMT* is AO under these two conditions. Firstly, any metric cost function satisfies the AO conditions for FMT* (Section 5 of [60]). Secondly, φ can be represented as a mixture distribution, composed of a uniform distribution that is sampled with probability ℓ and a non-uniform distribution that is sampled with probability $1 - \ell$ [60]. This results in a uniform distribution being sampled $n\ell$ times, and thus the proof of AO of FMT* is preserved. Likewise, for BFMT*, $\lim_{n \rightarrow \infty} \mathbb{P}(c_n > (1 + \varepsilon)c^*) = 0$ for all $\varepsilon > 0$ [118].

□

The performance of HBFMT* compared with BFMT* and FMT* is largely dependent on the sub-space $\mathcal{X}_{free}^\#$ that is initially sampled and the resulting distribution φ generated in \mathcal{X}_{free} . It is generally the case for reconfigurable mobile robots that $\mathcal{X}^\#$ is the workspace of the vehicle, which is in the translational \mathbb{R}^3 Euclidean space. However, if the platform requires a specific orientation or joint articulation motion that is not contained within the biased search region in \mathcal{X}_{free} , the algorithm may not perform as well as FMT* and BFMT* and may be dependent on samples taken from the uniform component of φ . Currently, the sub-space $\mathcal{X}_{free}^\#$ is chosen a priori, however a topic of future work is incorporation of autonomous selection of these spaces given a certain environment and robot model. Examples of such low-dimensional structure identification can be found in [101] and [128].

An example environment in which a dominant dimension outside of \mathbb{R}^3 is present is called ‘Spin to Win’ and is shown in Figure 6.2. The MAMMOTH rover starts in a narrow passageway, and its goal position is inside the same narrow passageway, however the yaw angle is changed by 180° . To get to the goal state from the initial state the rover must drive

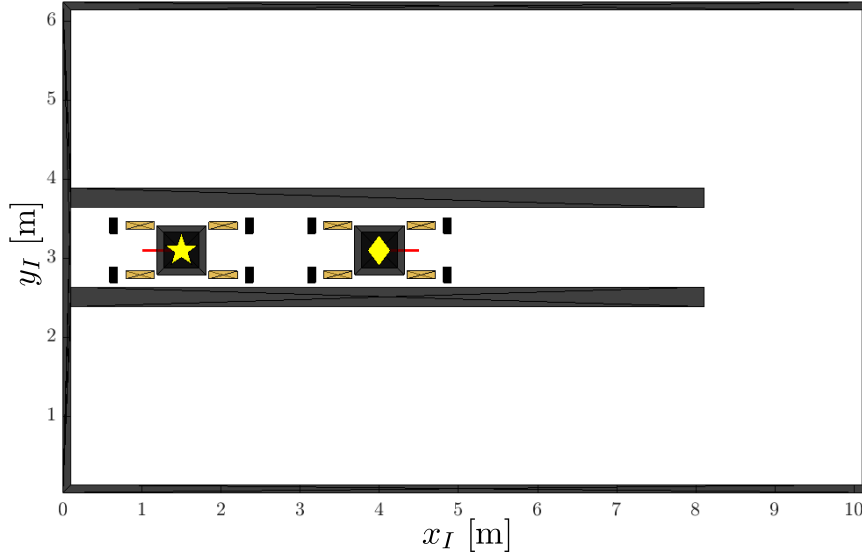


Figure 6.2 – The ‘Spin to Win’ environment in which the MAMMOTH rover must drive out of the narrow passageway, change its yaw angle by 180° and then drive back into the narrow passageway. The start state is at the diamond location, while the goal state is at the star location.

out of the passageway, change its heading and then drive back into the passageway. The use of HBFMT* exploring \mathbb{R}^3 in its first hierarchy level would not perform very well given that the yaw dimension is a dominant one within the full planning problem. This problem may be addressed by including the yaw of the robot within $\mathcal{X}^\#$. By doing this, a second problem arises. If the body without legs is used within the first hierarchy level it will be able to yaw by 180° within the narrow passageway, unlike the full rover model. It therefore becomes apparent that the footprint dimensions are also dominant within this problem and a BFMT* or FMT* that uniformly samples \mathcal{X} would be more useful.

The singe radius, r_{singe} parameter affects the number of alternative paths within $\Pi^\#$. By increasing the size of the singe radius a larger sub-set of the two trees’ nodes are prohibited from being expanded through. The affect of varying r_{singe} is shown in Figure 6.3. The figure illustrates the resulting alternative paths through \mathcal{X}_{free} in an environment with many possible homotopies to choose from. As a rule of thumb the r_{singe} parameter should define a ball that is larger than the radius that defines the ball that can completely envelop the body of the robot being planned for. Any smaller value will most likely result in redundant sub-space paths that will implicitly create a sampling bias towards the region of the full space that they lie in. As r_{singe} gets larger the number of alternate paths found will decrease. Therefore, if it is desired that the number of sub-space paths be smaller so as to focus on a smaller exploration region in the final hierarchy level, the r_{singe} value should be increased. The choice of r_{singe} is a trade-off between a focussed sampling distribution in \mathcal{X}_{free} and variance of sub-space paths embedded within \mathcal{X}_{free} .

An additional parameter that affects the performance of HBFMT* is the biased sampling distribution variance r_{tunnel} . A small value for r_{tunnel} results in focused sampling around the nodes in $\Pi^\#$, thereby increasing the planner’s reliance on the sub-state solutions $\Pi^\#$,

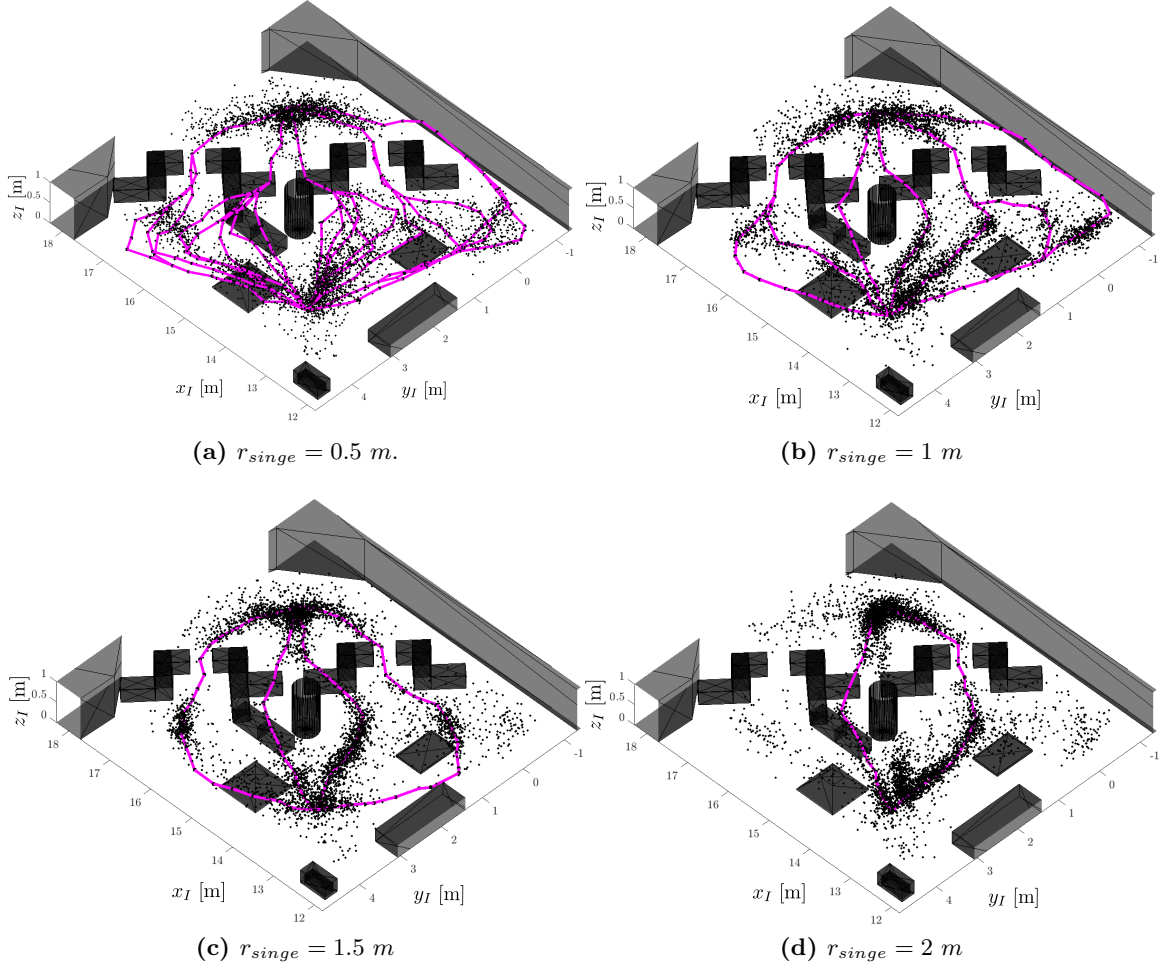


Figure 6.3 – The paths through $\mathcal{X}_{free}^\#$ generated by the exhaustive BFMT* with varying r_{single} values. The samples taken from the biased sampling distribution generated around the $\mathcal{X}_{free}^\#$ paths are also shown. For each of these cases $r_{tunnel} = 0.2 \text{ kJ}$ and $\ell = 0.2$

and making the planner more susceptible to missing necessary motions. A large value for r_{tunnel} results in a distribution resembling uniformity or a distribution with an accumulation of samples at the boundaries of \mathcal{X} . This removes any advantage or even worsens HBFMT*'s performance compared with BFMT* or FMT*. Figure 6.4 highlights the effect of different values for r_{tunnel} on the sampling distribution in \mathcal{X}_{free} .

Both r_{single} and r_{tunnel} require tuning given robot and environment models. An experiment that varies r_{single} and r_{tunnel} for planning problems in different environments is provided in Sec. 6.5 to empirically demonstrate the effect that these parameters have on algorithm performance.

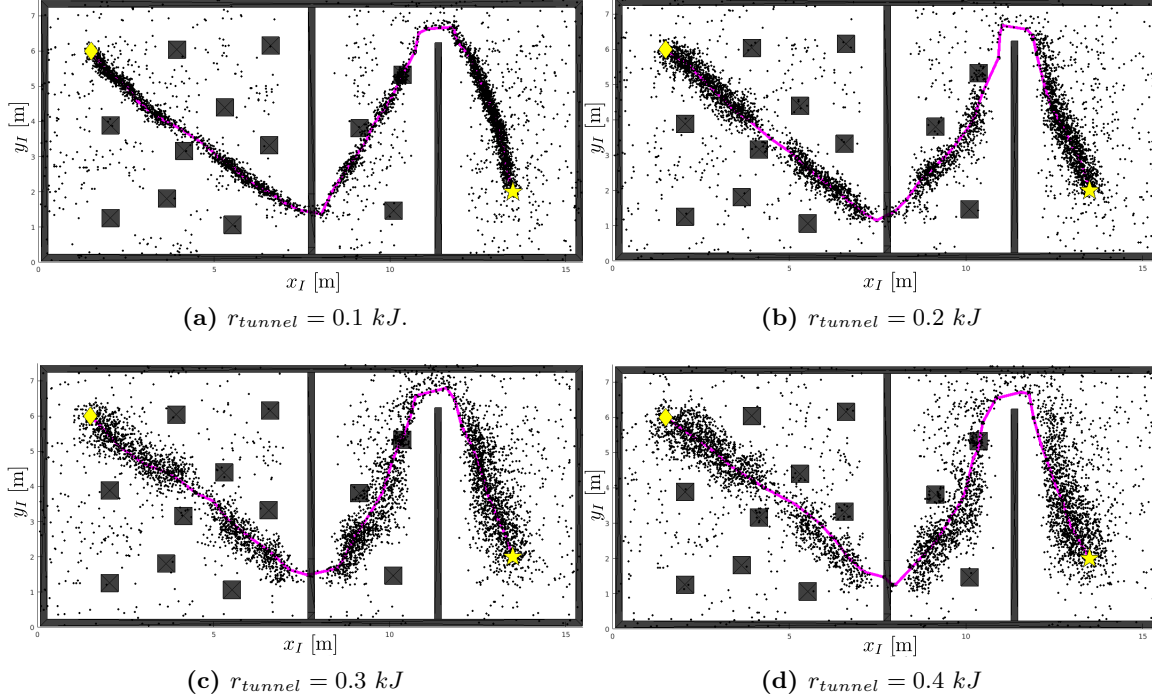


Figure 6.4 – The effect of different tunnel radii sizes on samples in $\mathcal{X}_{\text{free}}$ taken from a biased distribution focused around a path found from an exploration of $\mathcal{X}_{\text{free}}^{\#}$. In each of these cases $\ell = 0.2$.

6.4 Planning for the MAMMOTH Rover

The RMR used in all experiments is a simulated version of the MAMMOTH rover shown in Figure 6.1b. The simulated model has an 8-DOF state space that excludes the roll and pitch from the typical 10-DOF state described in Chapter 4. A state \underline{x} contains randomly sampled values for each of the 8-DOF. The state \underline{x} also contains the four thigh joint variables that are calculated as a function of the eight independent degrees-of-freedom as well as the local terrain profile. The physical properties of the actual MAMMOTH rover including its joint limits and footprint size are presented in Chapter 3.

6.4.1 Collision-Based Feedback Sampling

The 8-DOF state space does not characterize the capability of the MAMMOTH rover to raise one leg off the ground at a time. This ability is critical when the rover needs to clamber over obstacles. To allow this clambering behaviour to emerge, a sampling feedback method is used. If it is found that a sample from the 8-DOF space is deemed invalid due to leg i colliding with an obstacle, the sample is checked again with leg i raised. This is done by changing the leg's q_{U_i} value so that the leg is in its fully raised configuration. If this new leg-raised state is not in collision with obstacles and obeys static stability constraints, then it is inserted into $\mathcal{X}_{\text{free}}$. Alg. 9 provides pseudo-code for the sampling implementation used for the MAMMOTH rover planning problem. The subroutine `GetThighAngles(\underline{x})` is used

to find the \underline{q}_U vector that corresponds to a certain MAMMOTH rover pose and ensures that the wheels are in contact with the ground. This function is based on the thigh position kinematics presented in Appendix A.5. new space being planned through has 9-DOF as there can only be one leg raised at a time.

By not uniformly sampling a state space that includes dimensions that describe the robot's ability to raise and lower its legs off the ground, asymptotic optimality and completeness guarantees are removed from the planner. In practice however, this sampling strategy is shown to be effective. As part of the collision detection procedure, the rover's static stability is checked according to the model described in [83] and Section 4.2.2. In all of the experiments performed to validate the motion planner a stability margin of 10° is used. This value has been selected given empirical observations of safe operations from plan following experiments presented in Chapter 7.

Algorithm 9: Sample Free MAMMOTH Rover State with Collision Feedback

```

1 function SampleFree( $n, \Pi, l, r_{tunnel}$ )
2    $\lambda \leftarrow []$ 
3   for  $i = 1$  to  $|\Pi|$  do
4     if  $i \bmod 2 = 0$  then
5        $\lambda \leftarrow [\lambda \text{ Reverse}(\Pi_i)]$ 
6     else
7        $\lambda \leftarrow [\lambda \ \Pi_i]$ 
8    $m \leftarrow n$ 
9   while  $m > 0$  do
10    if  $m < (1 - \ell)n$  then
11       $\underline{x} \leftarrow \text{SampleBiased}(\lambda, r_{tunnel})$ 
12    else
13       $\underline{x} \leftarrow \text{SampleUniform}$ 
14     $\underline{q}_U \leftarrow \text{GetThighAngles}(\underline{x})$ 
15    if IsValid( $\underline{x}$ ) then
16       $m \leftarrow m - 1, S \leftarrow S \cup \underline{x}$ 
17    else
18      for  $i \leftarrow 1$  to 4 do
19        if LegCollision( $i$ ) then
20           $\underline{q}_U(i) \leftarrow q_{U_{min}}$ 
21          if IsValid( $\underline{x}$ ) then
22             $m \leftarrow m - 1, S \leftarrow S \cup \underline{x}$ 
23  return  $S$ 

```

6.4.2 Cost Function

The function that is used to evaluate state transition costs approximates the amount of mechanical work in a state transition:

$$J(\underline{x}_i, \underline{x}_{i+1}) = \mu Mg \sqrt{(\Delta x_B^I)^2 + (\Delta y_B^I)^2} + (m_{body} + 2m_{leg})g\Delta z_B^W + \mu m g r_{leg} \Delta \psi \\ + \mu m_{leg} g r_{leg} (\Delta q_{H_1} + \Delta q_{H_2} + \Delta q_{H_3} + \Delta q_{H_4}) + (r_{leg}/2)m_{leg}g\Delta q_{U_j}, \quad (6.2)$$

where Δ is the absolute difference over a single dimension. Additionally, $m_{body}=35$ kg is the mass of the rover's central chassis, $m_{leg}=10$ kg is the mass of a single leg, $M = (m_{body} + 4m_{leg})$ is the total mass of the rover, $r_{leg} = 0.7$ m is the average reach of the centre of mass of a single rover leg, and $\mu = 0.1$ is an approximate rolling resistance coefficient for the rover's wheels. The constants used to generate J are an approximation; an item of future work is to incorporate a high fidelity physical model of the rover into the planner to more accurately calculate energy expenditure. The traversal cost of a path is the sum of its state transition costs:

$$c(\pi) = \sum_{i=0}^{b-1} J(\underline{x}_i, \underline{x}_{i+1}), \quad (6.3)$$

where b is the number of waypoints in the path. The reason for using a mechanical work cost is because it is proportional to translational and rotational displacement between states and therefore a metric cost function.

6.5 Numerical Experiments

The aims of the numerical experiments are to:

1. evaluate HBFMT*'s performance against state-of-the-art sampling-based planners, including FMT*, BFMT*, RRT*, Informed RRT* and BIT*;
2. demonstrate the algorithm's suitability to path planning for RMRs using the system described in Section 6.4 and four distinct environments described in Sec. 6.5.1;
3. and to evaluate the sensitivity of the hierarchical planning framework to parameter tuning.

Numerical experiments are performed using a variety of environments that are designed to highlight the advantages and disadvantages of HBFMT*. Planners are evaluated on the returned path cost, required computation time, as well as the probability of successfully finding a path at a certain computation time. All experiments are performed on a flat terrain that is cluttered with obstacles.

All planners are implemented in simulation within the Open Motion Planning Library (OMPL) version 1.1.0 [120]. OMPL implementations for FMT*, RRT*, Informed RRT* and BIT* have been used. The BFMT* and HBFMT* implementations were written by the

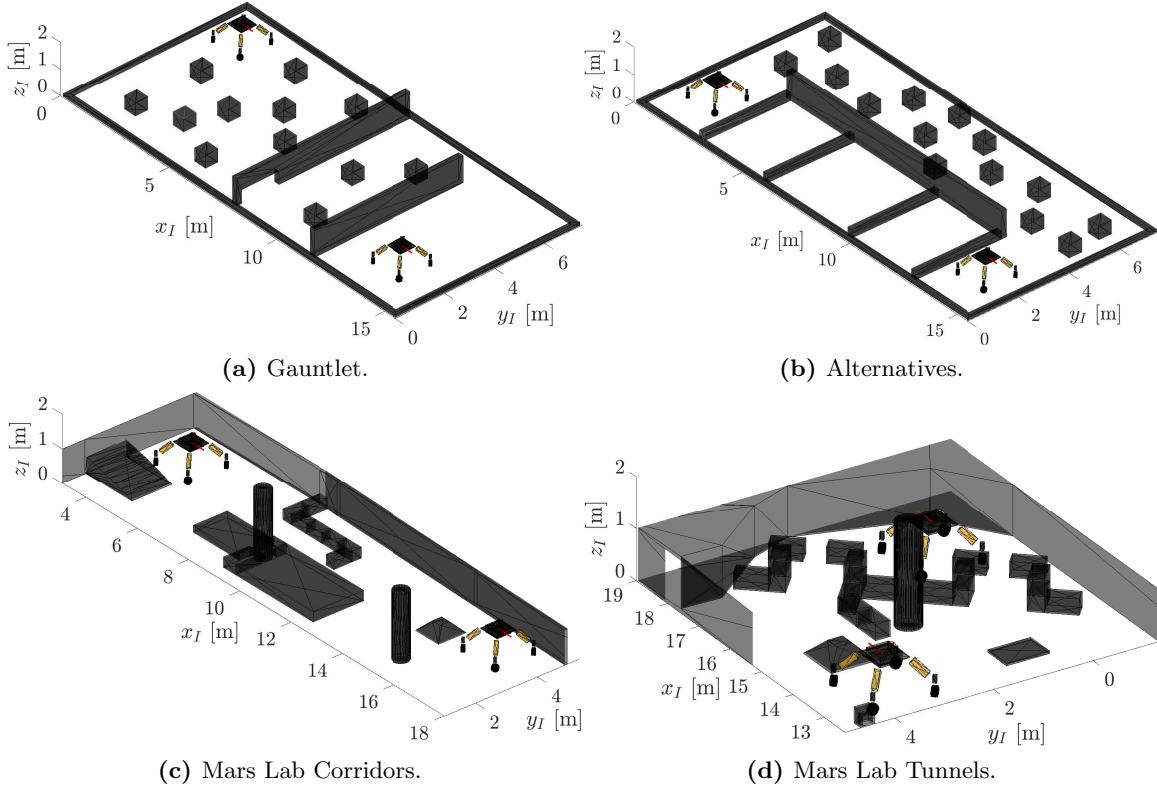


Figure 6.5 – The four environments used in the numerical experiments along with the initial and goal states of the rover.

author. Experiments are run on a 3.4 GHz Intel i7 processor with 32 GB of RAM running Linux Ubuntu 16.04 LTS. In all planners the GNAT nearest neighbour algorithm [42] is used to perform nearest neighbourhood queries, while the FCL collision detection library [82] is used to perform all collision checks.

6.5.1 Environments

Four separate environments are used to perform the experiments. The first two are purely simulated environments, however the last two are based on actual environments constructed within a Mars analogue environment called the Mars Lab, described in Section 3.4. The four environments have the following designations: (a) Gauntlet, (b) Alternatives, (c) Mars Lab Corridors and (d) Mars Lab Tunnels. Each of the environments are shown in Figure 6.5 along with the initial and goal states of the rover.

In the Gauntlet environment, the rover must navigate around/over box obstacles, through a tunnel and through a narrow passageway to get to its goal. This environment contains a single homotopy class of paths embedded within the Euclidean translational sub-space of the rover. HBFMT* is able to focus its search on this single family of paths. The Alternatives environment is designed to contain two homotopy classes of paths within the Euclidean translational sub-space. The robot may choose to clamber over the four step obstacles,

or drive around and over the many block obstacles. Clambering over obstacles is energy intensive, so an optimal solution should lie in the region with the block obstacles. In order to successfully traverse the Mars Lab Corridors environment the robot must clamber over the set of step obstacles in the middle of the environment. A solution for this environment is only achievable if the planner considers the robot's ability to raise individual legs off the terrain. This is achieved by using the collision-based sampling feedback strategy described in Section 6.4.1. Mars Lab Tunnels is used to showcase planners' ability to navigate through narrow passageways and choose between two homotopy classes.

6.5.2 Experimental Setup

For the benchmarking experiments, paths are found for the four environments using a set of sampling-based planners: HBFMT*, BFMT*, FMT*, BIT*, RRT*, Informed RRT*. A batch of trials is composed of 50 planner executions. For HBFMT*, BFMT* and FMT*, distinct batches are assigned between 1000 and 5000 nodes with 1000 node increments and between 5000 and 25000 nodes with 5000 node increments. HBFMT* additionally uses 1000 nodes to sample $\mathcal{X}_{free}^\#$. For BIT*, RRT* and Informed RRT*, single batches of 50 trials are used with an expiry time of 350 s. For RRT* and Informed RRT*, a goal-bias probability of 5% is used. This setup data is detailed in Table 6.1.

Table 6.1 – The planners used in the numerical benchmarking experiment. References to the planner's name, the work in which it was first presented, the number of trials per experiment batch, the number of samples per batch and the planning expiration time are given. The numbers of samples per batch is given in MATLAB vector notation $\langle \text{starting number} \rangle : \langle \text{increment number} \rangle : \langle \text{finishing number} \rangle$. All planners used are asymptotically optimal planners.

Planner	Ref.	OMPL	Trials/batch	# Samples/batch	Exp. Time [s]
HBFMT*	[92]	No	50	1000:1000:4000, 5000:5000:25000	450
FMT*	[60]	Yes	50	1000:1000:4000, 5000:5000:25000	450
BFMT*	[118]	No	50	1000:1000:4000, 5000:5000:25000	450
BIT*	[40]	Yes	50	-	350
RRT*	[62]	Yes	50	-	350
Informed RRT*	[39]	Yes	50	-	350

Table 6.2 – The tuning parameters used for HBFMT* for each environment.

Environment	r_{single} [m]	r_{tunnel} [kJ]	ℓ [-]
Gauntlet	2	0.1	0.2
Alternatives	3	0.75	0.2
Mars Lab Corridors	2	0.3	0.2
Mars Lab Tunnels	2	0.1	0.2

Tuning parameters for HBFMT* are different for each environment as detailed in Table 6.2. HBFMT* begins by sampling the translational \mathbb{R}^3 state space of the rover and uses the

rover's body without legs (Figure 6.1a) for collision checking. Based on observations of best practice [60], k-nearest neighbour versions of all of the planners are used. The value k_n , which denotes the number of nearest neighbour nodes, is calculated as $k_{n,FMT*} = 2^d e / d \log(n)$ for the FMT* variants, and $k_{n,RRT*} = (e + e/d) \log(n)$ for RRT*. For HBFMT*, k_n is determined by changing the number of nodes n to $n\ell$ as in Eq. 6.1.

6.5.3 Results

All experimental results are presented in path cost versus time and success rate versus time plots. Results for the Gauntlet environment are shown in Figure 6.6, Alternatives in Figure 6.8, Mars Lab Corridors in Figure 6.9 and Mars Lab Tunnels in Figure 6.10. In each of these figures, excluding Figure 6.9 for Mars Lab Corridors, separate results are presented with no feedback sampling (NFS) and with feedback sampling (FS). Only results using feedback sampling are presented for Mars Lab Corridors given that leg lift manoeuvres are required to successfully traverse the environment. Additionally, the cost versus time plots in these figures are in a standard box plot format. Each box for the FMT* variant planners represents a distinct number of samples as described in Table 6.1. Similarly, in the success rate versus time plots, circles represent a distinct number of samples from the FMT* variant planners. Timelapses of feasible solutions for each environment are presented in Figures 6.12-6.15 at the end of the chapter. The examples shown in these figures illustrate fluid transition between intuitively discrete locomotion modes such as driving, clambering and footprint reconfiguration.

Gauntlet

The first obvious observation from the Gauntlet experiments is HBFMT*'s ability to find feasible solutions. For both the NFS and FS cases, HBFMT* returns more successful paths than any of the other planners. For the NFS case HBFMT* begins returning feasible paths 100% of the time in the 20,000 node batch. BFMT* and FMT* are the next most successful planners with an approximately 70% success rate by the 20,000 to 25,000 node batches. The anytime planners perform poorly in this environment for the NFS case with an approximately 5% success rate at the 350 s expiry time. Another improvement by the HBFMT* planner is the computation time when compared with BFMT* and FMT*. For 25,000 nodes the average computation time difference is 80 s with FMT* and 100 s with BFMT*.

When analyzing path cost results it must be noted that the reported cost data is only for the trials that successfully returned a path. For both the NFS and FS cases HBFMT* outperforms all other planners and begins to find successful solutions faster than all other planners. For the FS case there is an approximate cost difference of 0.25 kJ between the HBFMT* and FMT* variants between 120 s and 250 s. For the NFS case on the other hand there is a much larger 1 kJ margin during the same time period. To explain this difference, it first must be noted that HBFMT* does a similar job between the NFS and FS cases. BFMT* and FMT* have difficulty finding a path through the first narrow passageway shown in Figure 6.7 in the NFS case. With FS, FMT* and BFMT* are able to use samples where

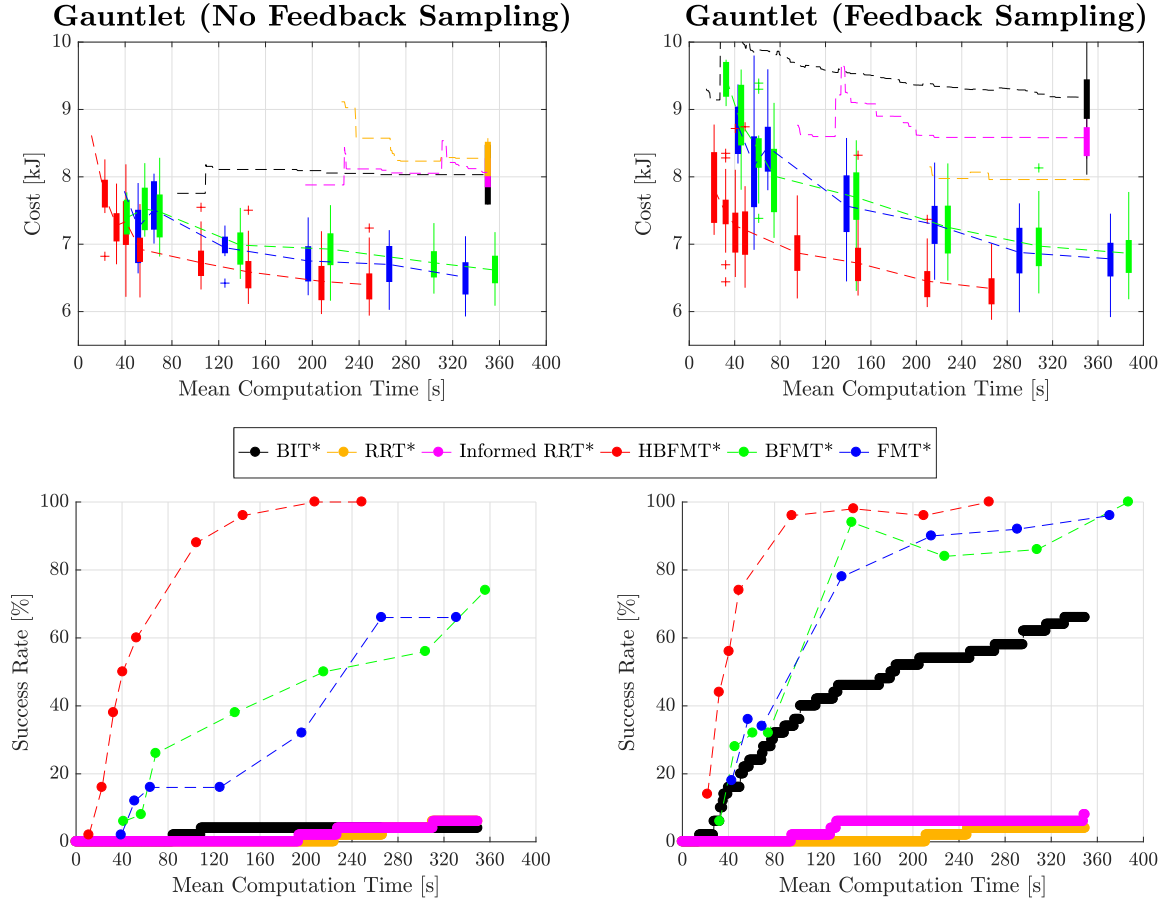


Figure 6.6 – Experimental results from planners solving for a path through the Gauntlet environment. The left column shows results with no feedback sampling and the right column shows results with feedback sampling.

the robot is straddling the low boundary walls. This results in a significant improvement in successfully returning paths, however due to energy intensive leg raising manoeuvres being utilized the average path cost increases. BIT* also has a notable increase in success rate when FS is enabled, however the cost of BIT*'s returned paths is 2 kJ to 3 kJ more expensive than the FMT* variants.

Alternatives

In the Alternatives environment HBFMT* will find paths through $\mathcal{X}_{free}^\#$ over the two homotopic regions. These regions include the area with the four steps and the area with block obstacles. The $\mathcal{X}_{free}^\#$ search by HBFMT* results in a biased sampling distribution in \mathcal{X}_{free} that is focused over both distinct sections of the space. The cheaper cost region will lie in the region riddled with box obstacles. To traverse the area with four steps would require many energy intensive leg lifts. Additionally, finding a path over the step obstacles with feedback sampling is extremely unlikely given the four high-dimensional narrow passageways that are generated by the four required clambering manoeuvres.

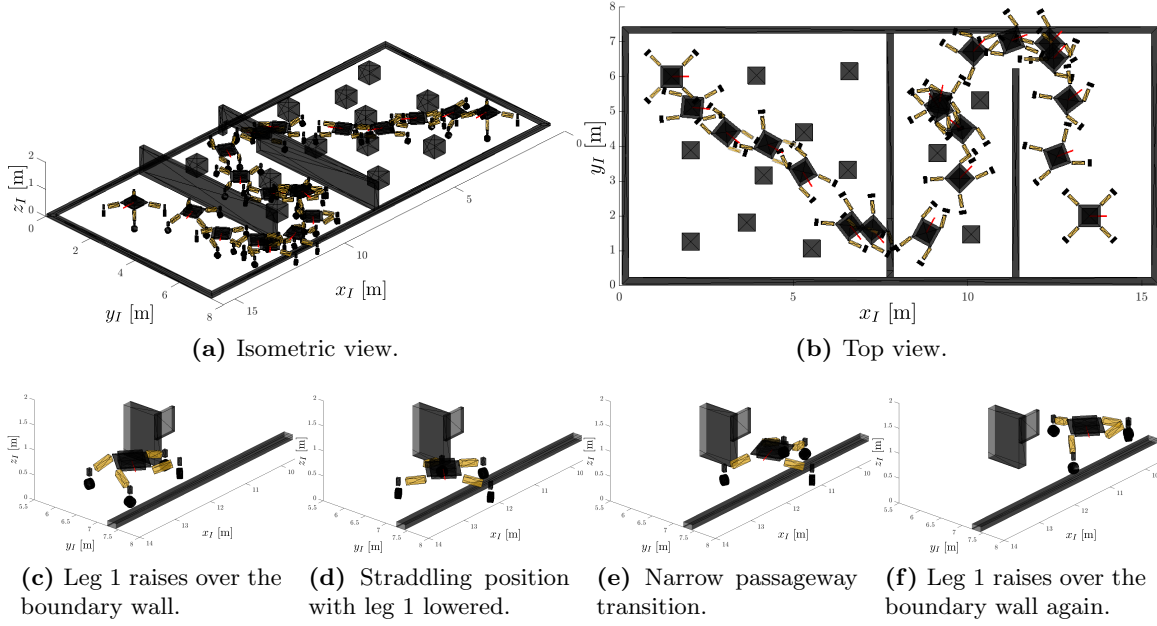


Figure 6.7 – A path generated by BFMT* using 25,000 nodes in which the MAMMOTH rover straddles the low boundary wall in the narrow passageway at $x \approx 11$ m and $y \approx 7$ m. The bottom row of images show the leg raising manoeuvres the MAMMOTH rover performs to clamber over the boundary wall.

Even though the number of samples available in the box obstacle region is diluted by sampling of the step region, a notable cost improvement between HBFMT* and the other planners is observed, especially for batches with smaller numbers of samples. Due to its bias around a sub-region of the environment's box region, HBFMT* converges to 100% successful solutions by 20 s; faster than any other planner when using NFS. BIT*, FMT* and BFMT* however are also relatively quick to converge to 100%, with BIT* reaching 100% convergence by 65 s, and BFMT*/FMT* by 100 s.

For the feedback sampling case the FMT* variants get close to but do not completely converge to 100% success rates. BIT* on the other hand does converge to 100% by 45 s. As mentioned in Section 6.4.1, due to the ninth 'leg-raise' dimension not being included in \mathcal{X}_{free} and only being sampled if a leg collision occurs, AO and PC guarantees are removed. This affect is not noticeable with the anytime BIT* planner however given that it may resort to previously found solutions and not rely on a wavefront of edges expanding through a single batch of nodes.

HBFMT* once again consistently returns better cost paths faster than all of the other planners, especially at lower computation times (< 100 s). For the NFS case there is an average cost difference of ~ 1.5 kJ at 60 s and ~ 0.25 kJ at 180 s between HBFMT* and BFMT*/FMT*. For the FS case, the margins are similar with ~ 1.8 kJ at 60 s and ~ 0.25 kJ at 195 s.

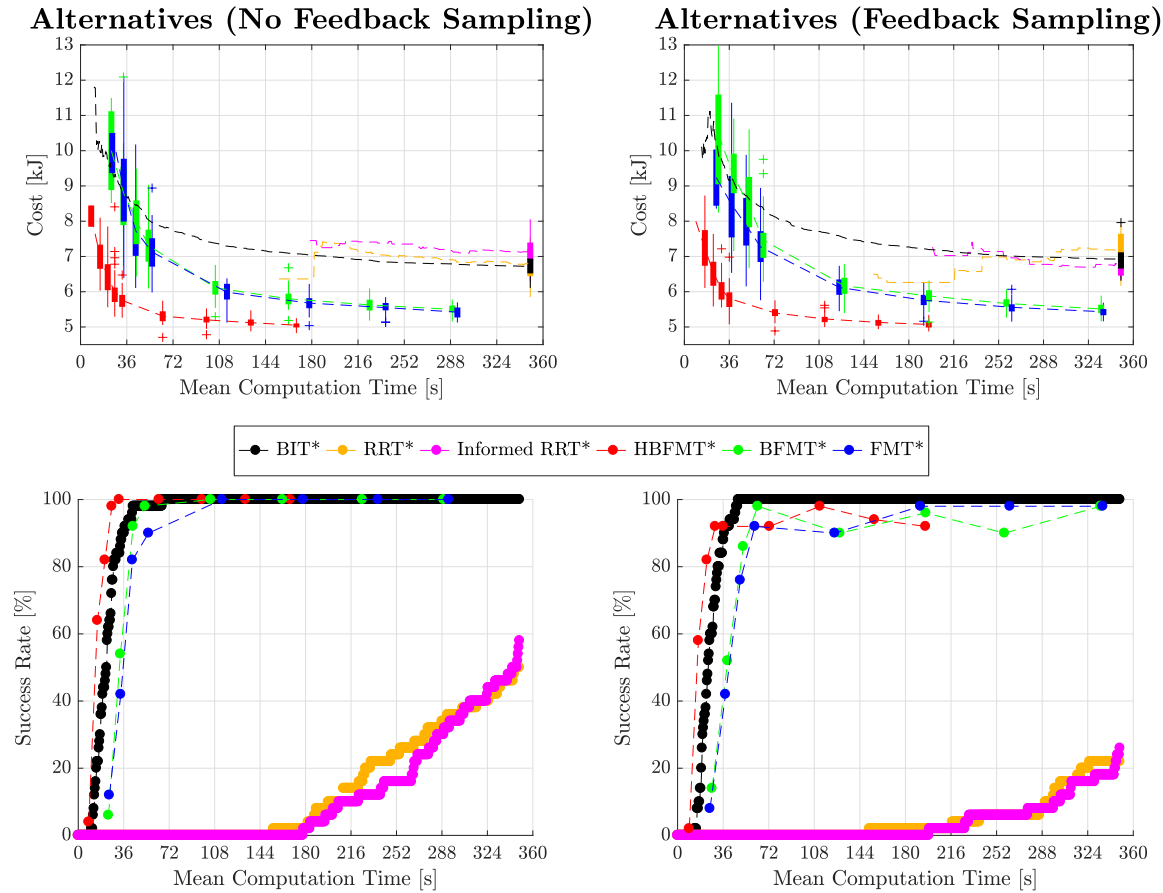


Figure 6.8 – Experimental results from planners solving for a path through the Alternatives environment. The top row shows successful path cost versus time in standard box plot format. The left column shows results with no feedback sampling and the right column shows results with feedback sampling.

Mars Lab Corridors

The Mars Lab Corridors environment forces the MAMMOTH rover to perform a clambering manoeuvre to traverse the ‘S’-shaped obstacle at the centre of the space. For this reason, only the FS case is used in the Mars Lab Corridors experiment. The manoeuvre required for the robot to clamber over the obstacle represents a high-dimensional narrow passageway where the robot must find a set of sequential moves to clamber over the obstacle while remaining stable. This narrow passageway sequence of manoeuvres includes one or more leg raises and yaw re-orientations as shown in Figure 6.14. Bidirectional search is commonly used to address the narrow passageway problem and its efficacy is evident from the success versus computation time plot in Figure 6.9. HBFMT* gets to 90% success by the 15,000 node batch, while BFMT* gets to just above 50% by 15,000 nodes. FMT* and the unidirectional anytime planners see success rates at or below 20% for all computation times. Due to its focussed exploration, HBFMT* is able to outperform BFMT* in terms of success rate.

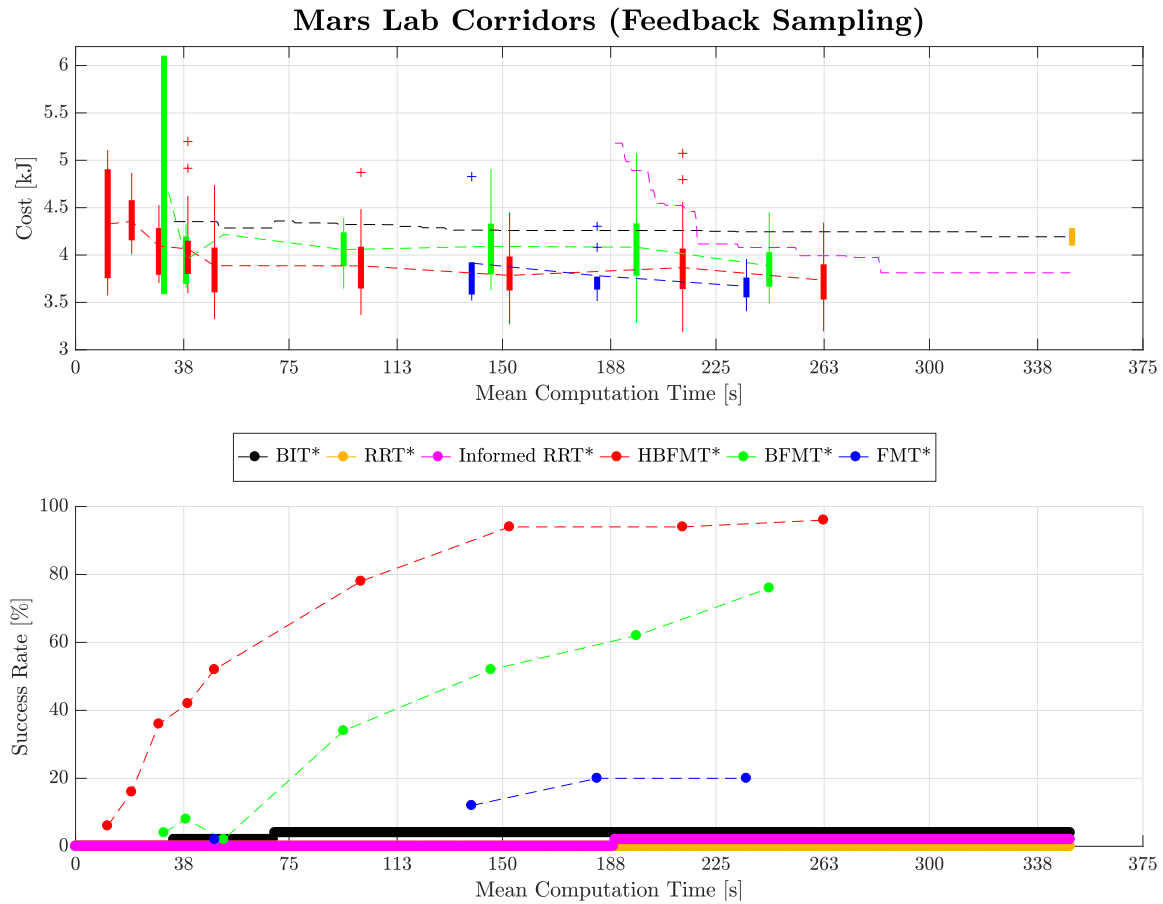


Figure 6.9 – Experimental results from planners solving for a path through the Mars Lab Corridors environment using feedback sampling.

In terms of cost, BFMT* and HBFMT* perform relatively similarly for successful trials. Besides the clambering manoeuvre, the rest of the traverse is relatively easy to find due to the rest of the environment being relatively open space. Therefore, if the clambering manoeuvre is found, it is straight forward for any of the planners to achieve a solution that is close to the optimal. Such a solution is shown in Figure 6.14.

Mars Lab Tunnels

Mars Lab Tunnels presents an environment with many narrow passageways in $\mathcal{X}^\#$. Two of these narrow passageways will be embedded within the set of possible feasible paths through \mathcal{X}_{free} . These two paths pass through the two tunnels. By biasing search towards the $\mathcal{X}^\#$ narrow passageways HBFMT* is more likely to find a successful path than the other planners, as shown in the success rate versus computation time plot in Figure 6.10.

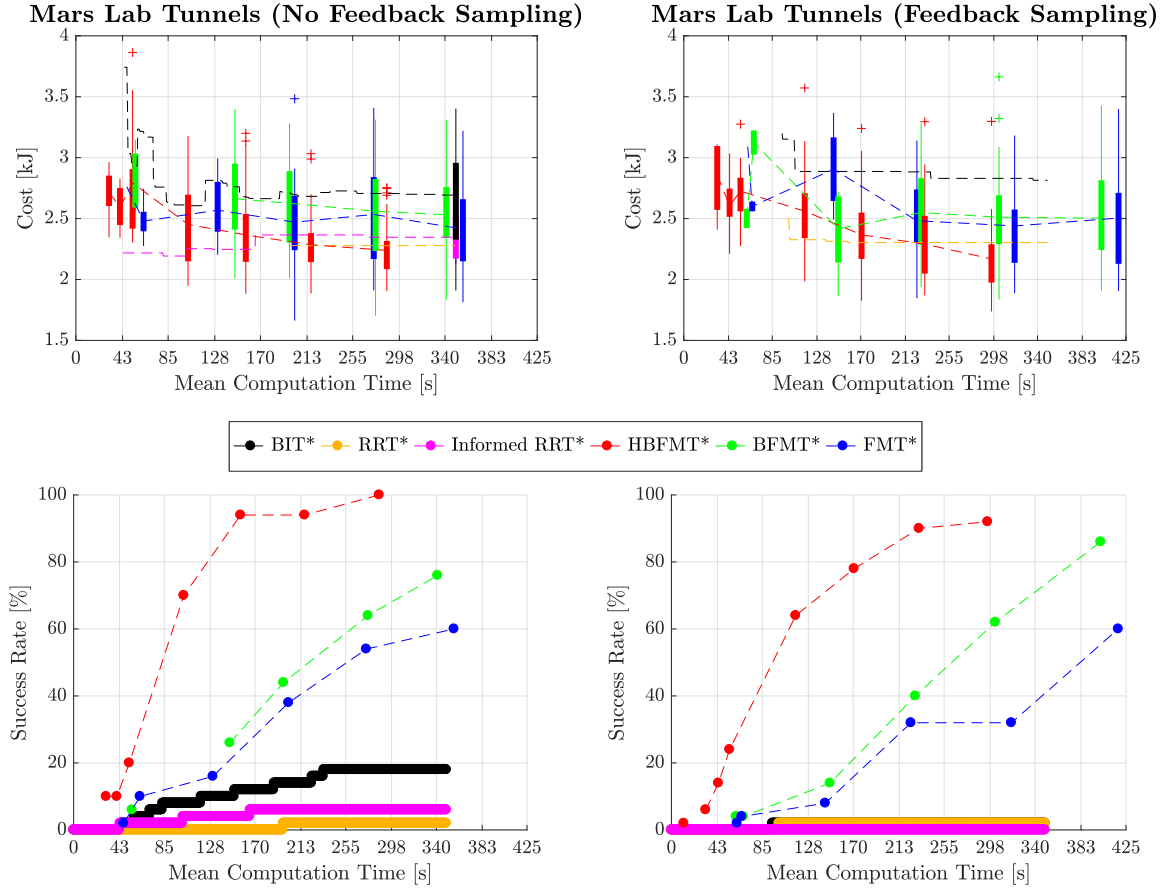


Figure 6.10 – Experimental results from planners solving for a path through the Mars Lab Tunnels environment. The left column shows results with no feedback sampling and the right column shows results with feedback sampling.

6.5.4 Parameter Tuning

An experiment investigating the robustness of the algorithm given varying parameter values of singe radius r_{singe} and tunnel radius r_{tunnel} is performed. In this experiment the HBFMT* planner is run for the MAMMOTH rover in the Alternatives, Gauntlet and Mars Lab Corridors environments with the same start and goal configurations shown in Figure 6.5. A total of 100 separate batches of trials are run, with 50 trials per batch. Each batch has a different r_{singe} and r_{tunnel} combination. The singe radius r_{singe} is varied between 0.5 m and 5 m, while r_{tunnel} is varied between 0.1 kJ and 1.8 kJ. Feedback sampling is used in the Gauntlet and Mars Lab Corridors environments, however it is not used in the Alternatives environment.

The results of this experiment are shown in Figure 6.11. Each mesh vertex represents the average of the 50 trials run for a specific r_{singe} and r_{tunnel} combination. The results for the Gauntlet environment demonstrate that lower cost is returned as the biased sampling gets more focused around the single $\mathcal{X}_{free}^\#$ homotopy class available in the environment. In Alternatives, it is seen that when r_{singe} is large and r_{tunnel} is small the number of

successful traverses drops. This is caused by biased samples not being placed in the half of the environment filled with block obstacles and only being placed over the step obstacles. The planner has difficulty finding a path over the four step obstacles and therefore the success rate drops. Additionally, the Alternatives plot shows that if r_{tunnel} is too small then the biased sampling is too focused and cannot find configurations that successfully avoid the large number of block obstacles along the optimal path. The Mars Lab Corridors experiment shows that the returned path cost stays relatively constant across the r_{tunnel} and r_{singe} ranges. It is noted that for small r_{tunnel} values the success rate drops off to almost zero. This behaviour is due to the biased sampling being too focussed over the step obstacle region. A small amount of variance in body translation is required for formulate a statically stable clambering manoeuvre. It is also noted that as the r_{singe} variable decreases less successful traverses are found. This is due to an increase in alternative workspace paths, many of which will not be embedded within the regions encompassing the possible full state space paths as seen in Figure 6.3a.

6.6 Summary

Generally, HBFMT* is observed to improve path quality with a smaller computation time and return a greater number of feasible solutions. The HBFMT* performance improvement is due to it biasing search of the high-dimensional state space by first considering the dominant translational sub-space of RWMRs. This biasing allows for constrained exploration in the full state space, allowing the sampling-based planner to more frequently probe regions that are more likely to return a high quality solution. The planner retains the FMT* AO guarantee by imposing a base uniform distribution in its sampling strategy. Static stability constraints are incorporated by treating them as state space obstacles within the collision checking routine. Feedback sampling is utilized to expose clambering behaviours. Even though AO is removed when clambering is introduced, in practice the addition of feedback sampling to all algorithms tested allows for greater mobility without a significant reduction in returned feasible paths or path quality. Lastly, a parameter tuning experiment is performed that demonstrates the effect of changing the HBFMT* tuning parameters. The effects of the tuning parameters are shown to be dependent on the type of environment being explored. It is generally the case however that an r_{tunnel} value that corresponds to a state space motion equivalent to two to three rover body lengths and an equivalent singe radius return adequate results.

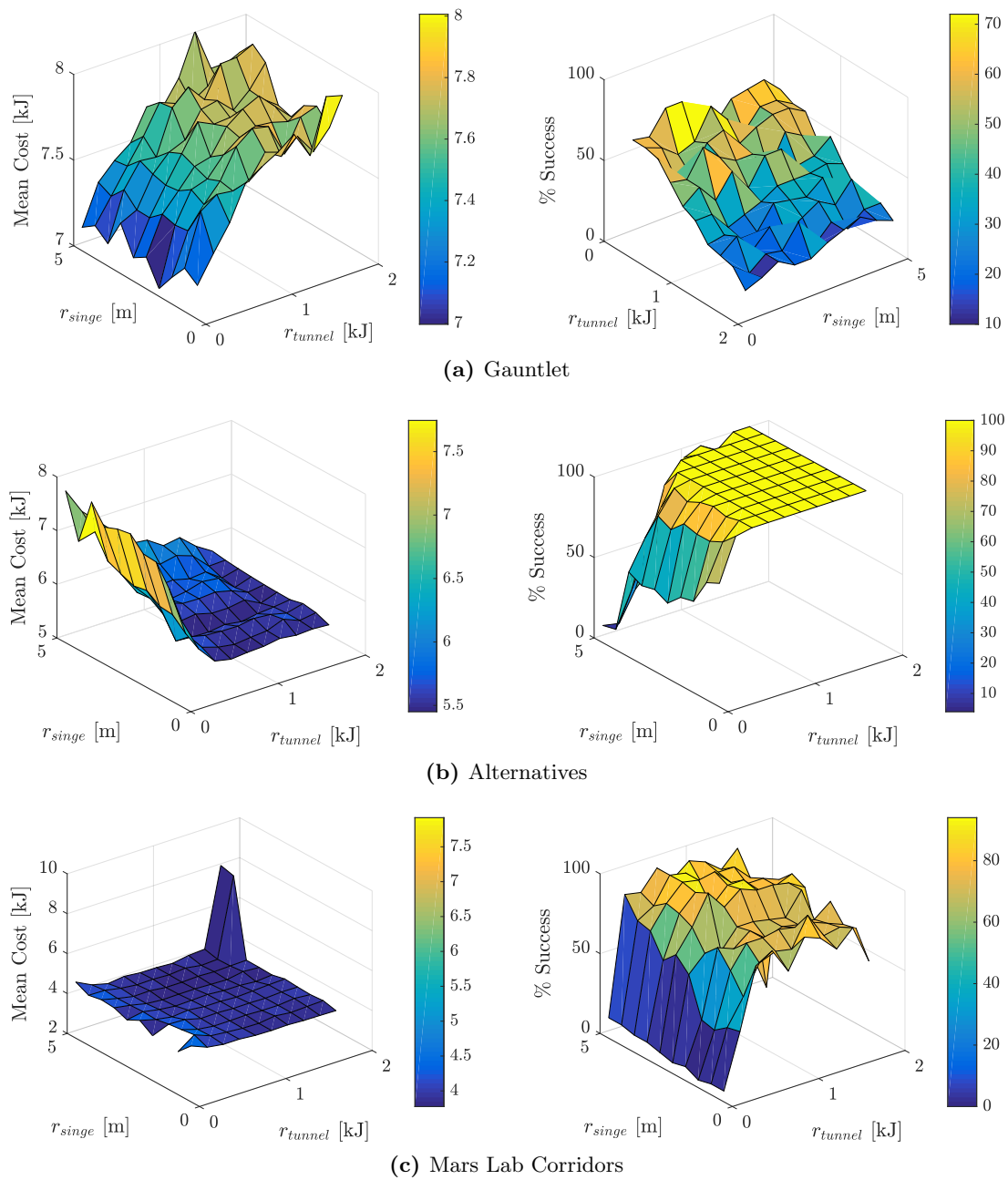
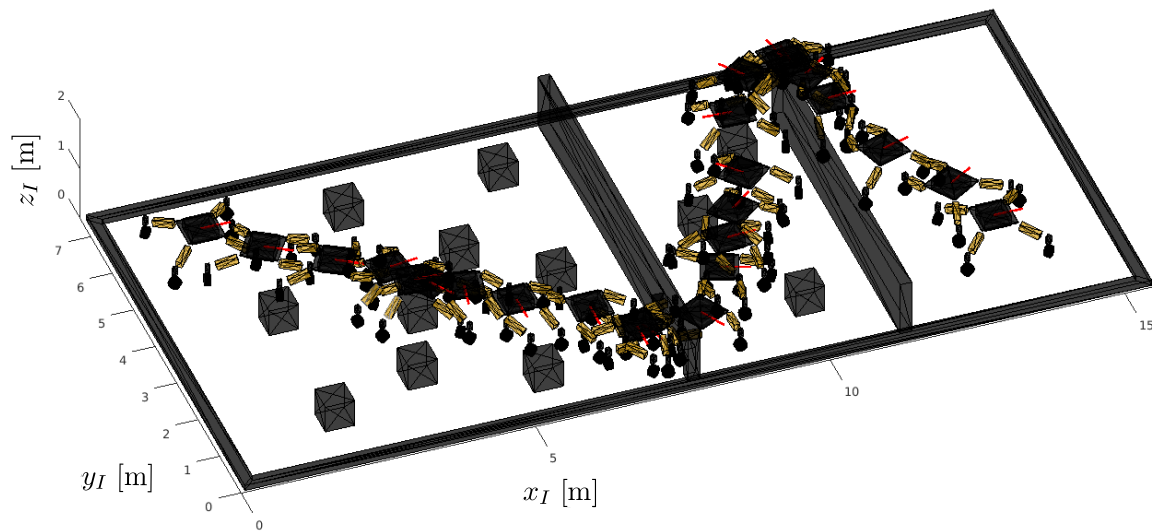
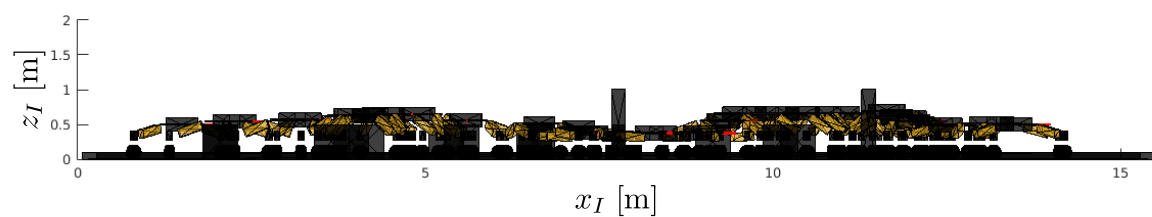


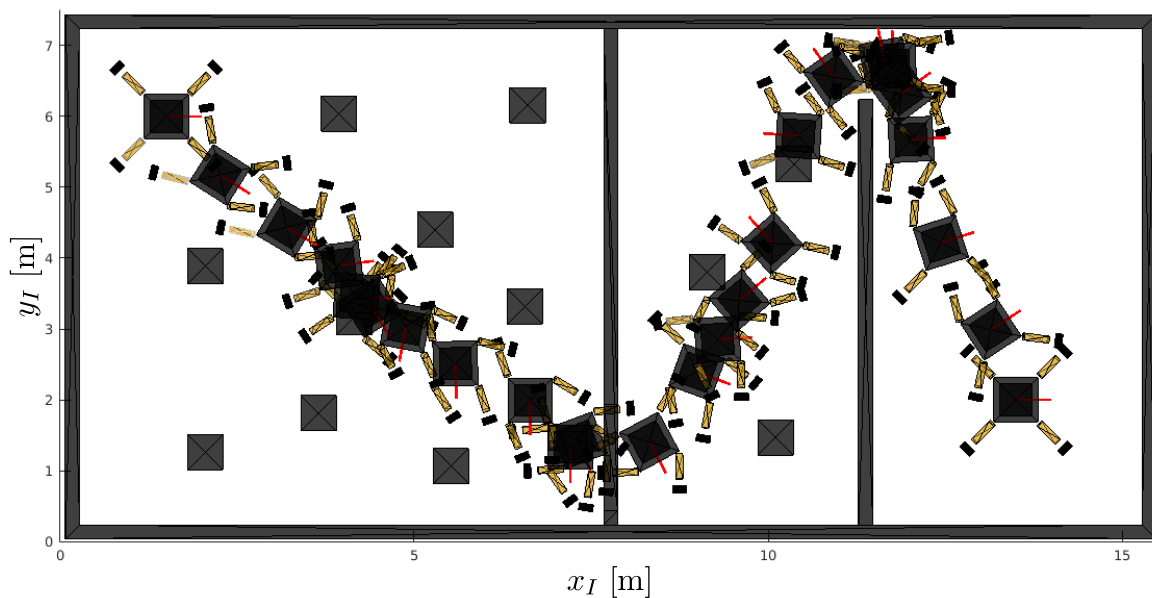
Figure 6.11 – Path cost and success rates for HBFMT* with 5000 nodes in each environment with varying r_{singe} and r_{tunnel} values.



(a) Isometric view.



(b) Side view.



(c) Top view.

Figure 6.12 – The Gauntlet environment seen from (a) an isometric view, (b) a side view and (c) a top view. The successful MAMMOTH traversal was generated using HBFMT* with 25,000 nodes.

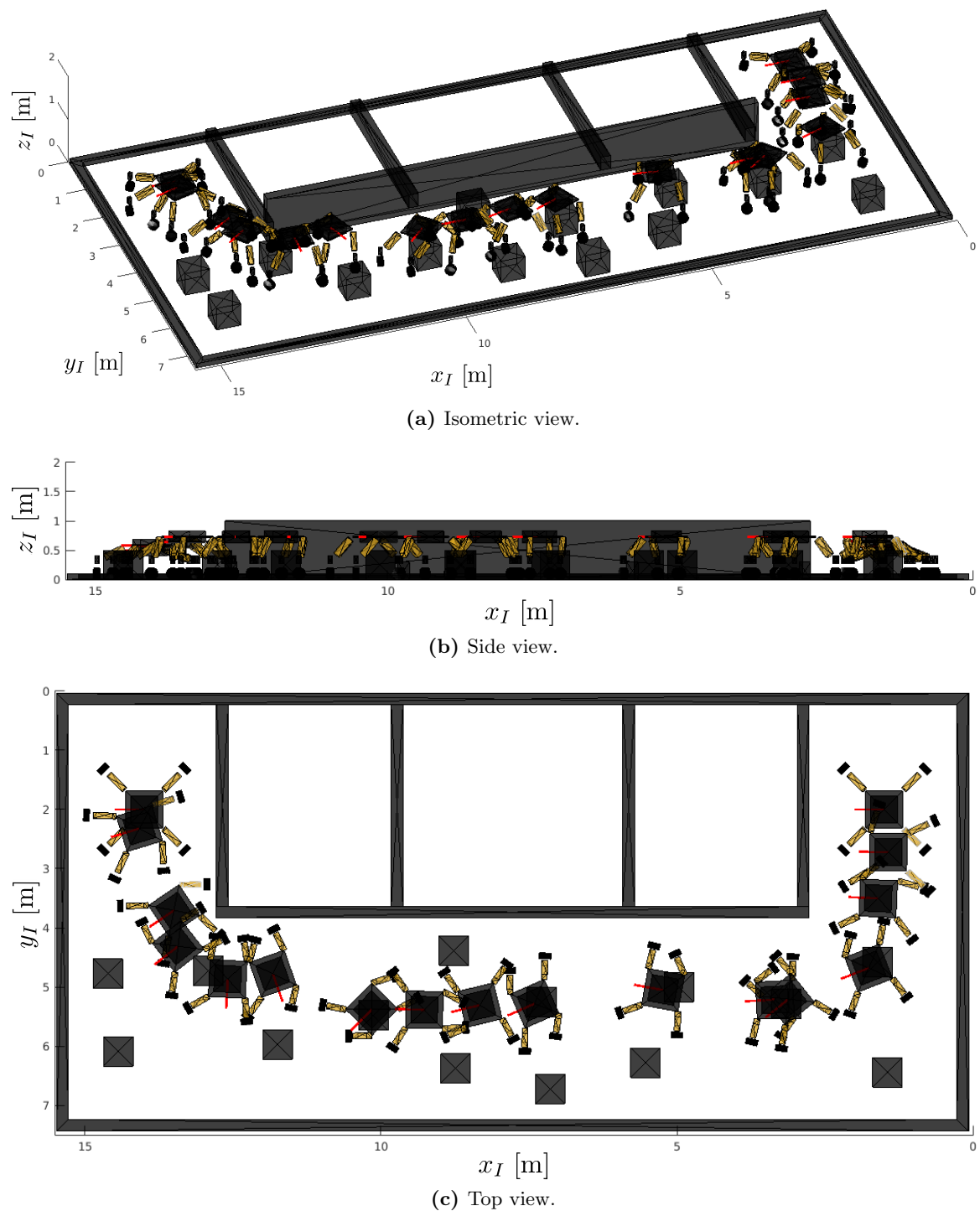


Figure 6.13 – The Alternatives environment seen from (a) an isometric view, (b) a side view and (c) a top view. The successful MAMMOTH traversal was generated using HBFMT* with 25,000 nodes.

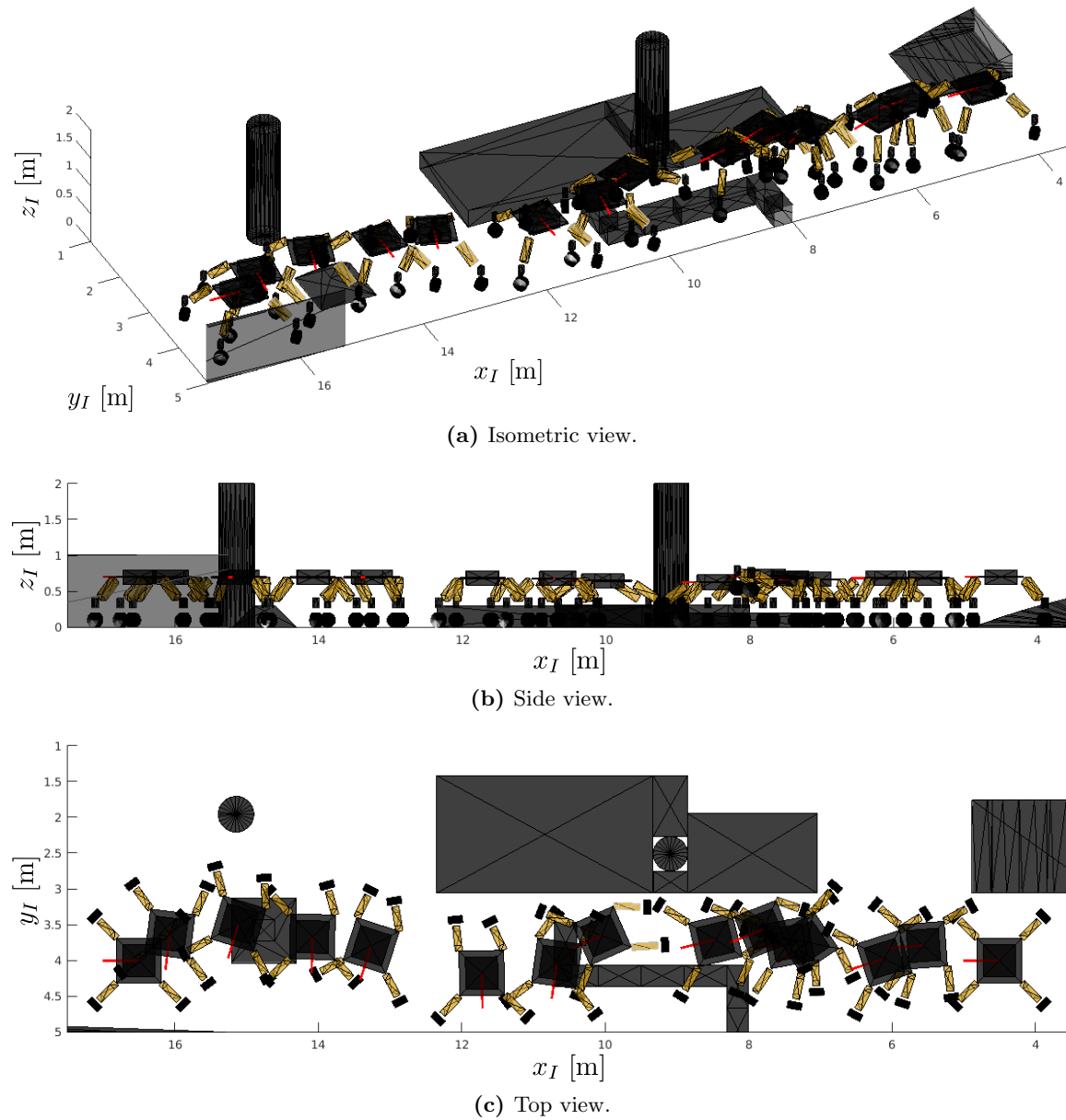


Figure 6.14 – The Mars Lab Corridors environment seen from (a) an isometric view, (b) a side view and (c) a top view. The successful MAMMOTH traversal was generated using HBFMT* with 25,000 nodes.

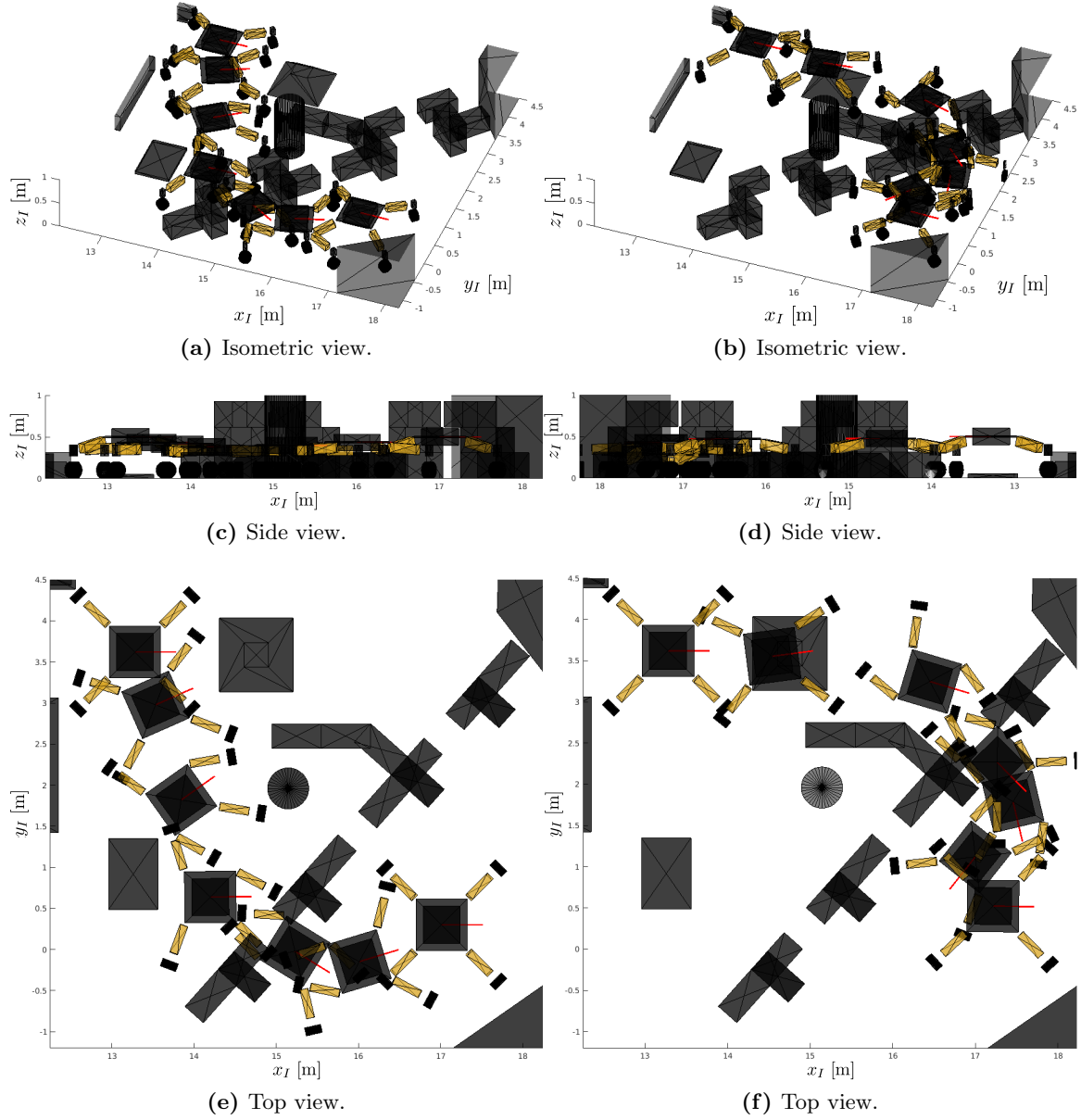


Figure 6.15 – The Mars Lab Tunnels environment seen from (a-b) an isometric view, (c-d) a side view and (e-f) a top view. Views (a,c and e) show the rover traversing one possible homotopy class, while views (b, d and f) show the rover traversing another homotopy class. These successful MAMMOTH traversals were generated using HBFMT* with 25,000 samples.

Chapter 7

Plan Following Experiment

The objective of the plan following experiment is to (a) demonstrate the physical MAMMOTH rover following paths generated by the proposed HBFMT* planner; to (b) validate the cost function used in the planner against the actual energy expenditure by the physical platform; to (c) demonstrate actuator constraints (faults or sensor pointing restrictions) being incorporated into the plan; and to (d) highlight practical considerations of the MAMMOTH planning problem not originally considered in the numerical experiments.

7.1 Setup

The experiment is divided into trials where a single traverse of an environment is attempted. The execution time for a traverse is started as soon as motion in any actuator starts. Traverse execution time ends when the rover is within a tolerance ϵ of the final state. All experimental trials are performed by running the motion planner before the traverse is begun. The planner's generated path is given to the path follower which is used to control the motion of the rover. The rover drives along the path until it reaches its goal state.

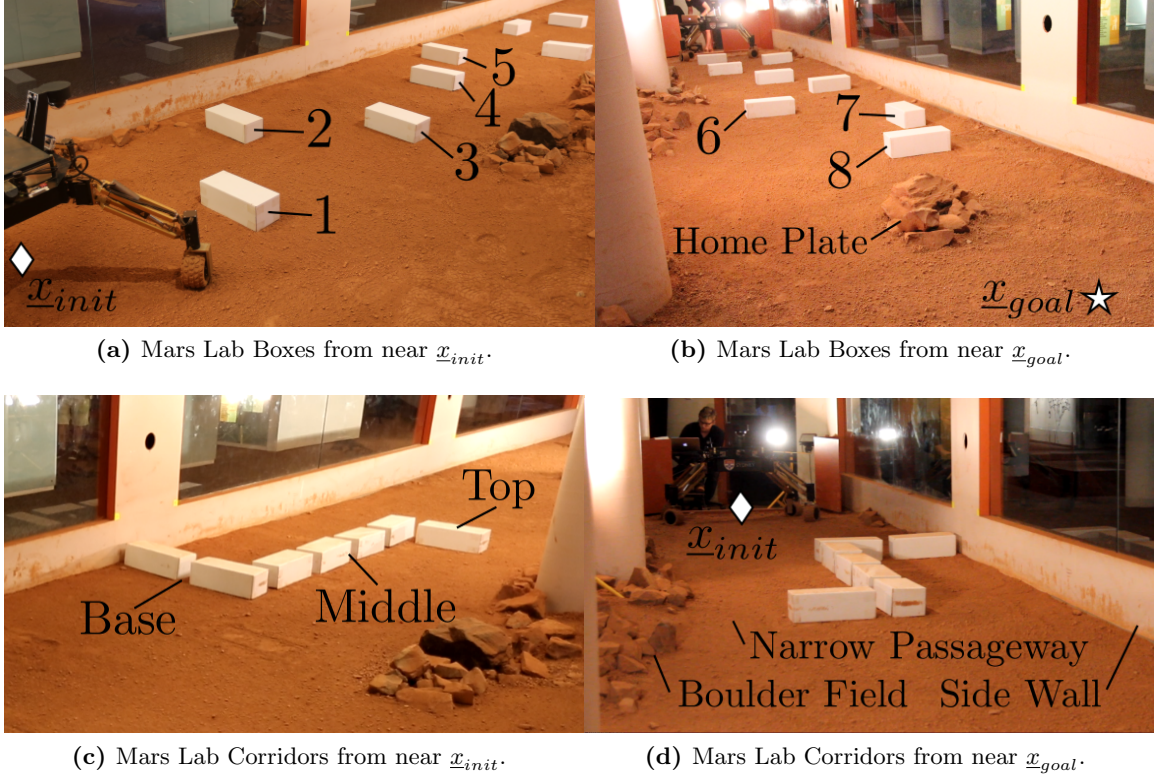
Trials are run for all environments using either 5000 or 10000 samples in \mathcal{X}_{free} and 5000 samples in $\mathcal{X}_{free}^{\#}$. Varying stability margins are used, while HBFMT* parameters are held constant over each trial at $r_{sing} = 0.2$ m, $r_{tunnel} = 0.2$ kJ and $\ell = 0.2$. Obstacle-based feedback sampling is enabled for all trials to allow the robot to raise legs over obstacles if necessary.

The initial and goal states for all trials are given in Table 7.1. Obstacles within the environments consist of rocks and cardboard boxes, as can be seen in pictures of the two separate environments in Figure 7.1. The planner has a map of the environment and the obstacles. All obstacles are inflated by a radius of 10 cm in the map.

Obstacle inflation is needed due to uncertainty in the translational localization solution of the platform. Robot localization is performed using an array of fiducial markers mounted on the ceiling of the Mars Lab (shown in Figure 3.8b) and an upwards facing camera and IMU mounted on board the robot. The triangulated translational solution has an accuracy of ± 10 cm. The MAMMOTH rover's localization system is described in Section 3.2.

Table 7.1 – The path following experiment initial and goal states for each trial.

	x_B^I [m]	y_B^I [m]	z_B^I [m]	ψ [°]	q_{H_1} [°]	q_{H_2} [°]	q_{H_3} [°]	q_{H_4} [°]
\underline{x}_{init}	4.5	4.0	0.7	0	0	0	0	0
\underline{x}_{goal}	16.5	4.0	0.7	0	0	0	0	0

**Figure 7.1** – The two environments used in the plan following experiment. Obstacles within the environment are cardboard boxes, rocks and boundary walls.

The path following experiments are run on two separate environments: *Mars Lab Boxes* and *Mars Lab Corridors*. The Mars Lab Boxes environment is a long thin wedge of the Mars Lab that is on a flat surface and is littered with box obstacles. The environment forces the robot to modify its footprint and possibly raise legs off the ground to get from one end of the space to the other. Each of the cardboard boxes in the environment is given an identifying number as shown in Figure 7.1a and 7.1b. The Mars Lab Corridors environment is the same as the one described in Section 6.5.1 and shown in Figures 7.1c and 7.1d. To traverse this environment, the rover must lift at least one leg over the ‘S’-shaped obstacle in the centre of the space. The three sections of the ‘S’ are denoted as ‘Base’, ‘Middle’ and ‘Top’. The environment is used to showcase how the planner accommodates for clambering manoeuvres using feedback sampling.

In selected trials sampling constraints are applied to various hip joint rotations or the body yaw rotation. These trials serve to highlight how the planner may be easily modified to

accommodate for such cases as broken actuators or sensor pointing constraints. These trials are also performed to highlight how constraining the dimensionality of the explored configuration space may be beneficial in finding lower cost paths. To apply these constraints the sampling range of the associated constrained sub-space is reduced. In the case of an equality constraint the sampling range is $\pm 0.1^\circ$ about the desired constraint value. Table 7.2 presents each of the successful path following trials executed by the MAMMOTH rover and the type of constraint(s) applied during the trial.

Table 7.2 – Setup details for each of the successful path traversals executed with the physical MAMMOTH rover.

Trial	Env.	n	$n^\#$	β [$^\circ$]	Constraint
10	Boxes	5000	5000	6	None
29	Boxes	5000	5000	23	None
47	Boxes	5000	5000	23	None
56	Boxes	10000	5000	23	None
55	Boxes	10000	5000	14	$q_{H_1} = 0^\circ$
32	Boxes	10000	5000	14	$q_{H_1} = 0^\circ$ and $q_{H_3} = 0^\circ$
49	Boxes	10000	5000	14	$q_{H_1} = 0^\circ$ and $q_{H_4} = 0^\circ$
38	Boxes	10000	5000	23	$\psi = 0$
61	Corridors	5000	5000	14	$q_{H_1} = 0^\circ$ and $q_{H_4} = 0^\circ$
66	Corridors	10000	5000	14	$q_{H_1} = 0^\circ$ and $q_{H_4} = 0^\circ$

7.2 Results

A total of 66 trials are attempted, 10 of which are successful in traversing the pre-planned path completely without hitting any obstacles. Out of the remaining 56 trials, seven trials are ‘partially successful,’ in which the corner of an obstacle is hit but the robot’s traversal is not impeded. A set of 21 trials are stopped due to a collision with an obstacle. The obstacle collisions are caused by a noisy localization solution, while three are caused by obstacles not being included in the map. Tip-over stability failures occurred five times. These are caused by the use of too small a stability margin within the planner, or the introduction of an incline due to thigh linear actuator positions having different stroke values. Nine trials fail due to software faults, while the remaining two trials fail due to low battery charge on the emergency stop remote control system.

For each trial a planned path is generated by HBFMT*, a simulation of the trial that uses the path following controller is run and then the physical traverse is attempted. Results from each of these steps are compared to validate the performance of the physical system in following a planned path. To validate the cost function used in planning with the physical system the root mean squared error (RMSE) and normalized root mean squared error (NRMSE) between each sub-space in each trial is found between the physical traverses and planned paths. The RMSE and NRMSE are calculated as:

$$RMSE(\hat{J}_i) = \sqrt{E[(\hat{c}_i - c_i)^2]}, \quad NRMSE(\hat{w}_i) = \frac{\sqrt{E[(\hat{c}_i - c_i)^2]}}{\hat{c}_{i_{max}} - \hat{c}_{i_{min}}}, \quad (7.1)$$

where \hat{c}_i is the vector of accumulated mechanical work costs for a sub-space i in the physical trial, while \underline{c}_i is the vector of accumulated mechanical work costs for sub-space i in the planned path. The $E[\cdot]$ operator takes the mean value. The RMSE and NRMSE results for each trial are presented in Figure 7.4. Given that the traverse duration may be different between the plan, simulation and physical traverse, a normalized traverse duration measure $\tau \in [0, 1]$ is used to compare the three. The τ parameter can be thought of as the proportion of traverse executed so far.

Time-lapse images taken from trials 56 and 61 are provided in Figure 7.2 and Figure 7.3 respectively. Additionally, videos of each successful traverse may be viewed at <https://goo.gl/5MC6Ib>. To demonstrate that the proposed planner can produce plans that can be followed by a real world reconfigurable rover, two separate trials (trials 56 and 61) from the two environments are presented in detail. Plots providing a detailed comparison between the physical trials, simulated trials and planned paths are provided at the end of the chapter.

7.2.1 Trial 56: Unexpected Emergent Behaviour

Trial 56 uses HBFMT* to plan a path through the Mars Lab Boxes environment. Initial settings for the planner include a sub-space batch size $n^\# = 5000$, a full space batch size $n = 10,000$, a minimum stability margin $\beta = 23^\circ$ and no motion constraints. Additionally, the tunable HBFMT* parameter values are set to $r_{single} = 2$ m, $r_{tunnel} = 0.2$ kJ and $\ell = 0.2$. Upon completion, the planner found a path with a mechanical work cost of 4.65 kJ with a computation time of 206 s. In its sub-space search two separate solutions are found. The results of the simulated MAMMOTH plan result in a mechanical work cost of 4.71 kJ, while the physical experiment resulted in a mechanical work cost of 5.07 kJ.

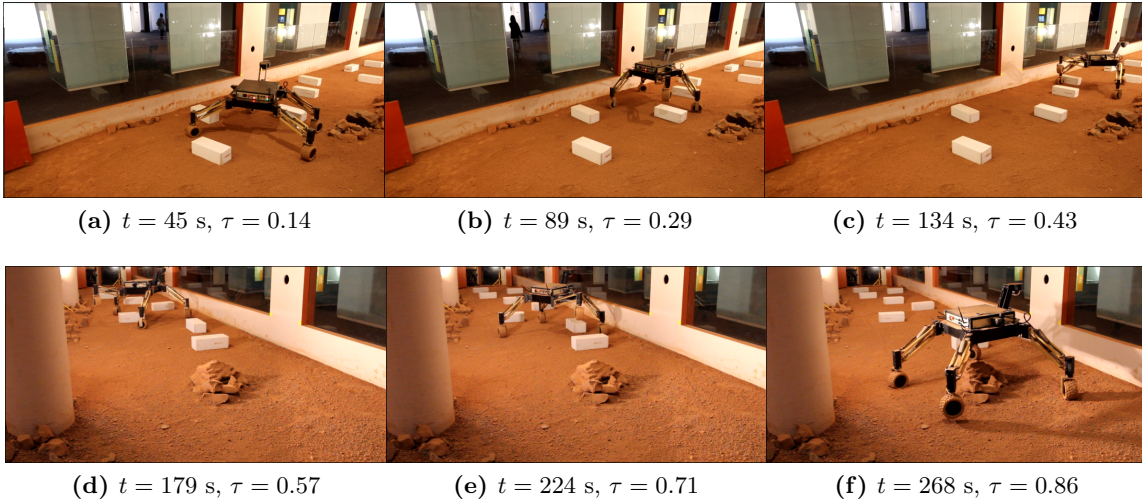


Figure 7.2 – Timelapse of Trial 56 with a time duration of 313 s. Images (a-c) show the traverse from a point of view close to the initial state, while images (d-f) show the traverse from a point of view close to the goal state.

The trial 56 traverse is now described; a video of the trial can be found at <https://youtu.be/fZ7X2Nz81hA>, while a time-lapse is shown in Figure 7.2. The rover begins in the common initial state and begins its traverse by driving directly forward. At 0:12 it begins to transition into a stable tripod state with legs 1,2 and 4 as the contact points of the tripod in preparation for an impending leg 3 lift. At the same time the rover re-orient its yaw to prevent leg 1 from hitting box 3. At 0:40 leg 3 begins to raise off the ground, initially to clear box 3. Excessive steering motion is observed at 1:24 due to noisy localization in this region of the yard. From 1:52 to 2:32 the rover yaws by over 90° to get leg 2 out of the path of box 5 and prepare leg 4 to avoid box 6. From 2:32 to 3:27 the rover continues to yaw by 180° so that it is almost facing backwards allowing leg 4 to drive past box 7 on the box's wall side. From 3:27 onwards the rover drives directly towards the xy position of \underline{x}_{goal} as it is already in a satisfactory configuration to traverse over the last rock obstacle. As the rover traverses over Home Plate it lowers leg 3. Lastly, it performs a yaw manoeuvre of -90° to reach the final state. It is observed that leg 1's steering joint slightly scuffs the wall during this manoeuvre. This minor collision may be attributed to an incorrect wall location within the map.

A notable aspect of this path is how the robot raises a single leg for the majority of the traverse. This is an example of an emergent behaviour that was not expected and was unintuitive to the operator. It is observed in Figure 7.5 that there is a smaller mechanical work cost accumulated by the hip joints than in trials 10, 29 and 47, the other unconstrained Mars Lab Boxes trials. The Trial 56 plan trades hip motion for increased yaw motion compared to trials 10, 29 and 47. It is also observed from this plot that the q_{U_3} contribution to the total mechanical work cost is almost four times the cost of any of the other thigh joints in Trial 56. This is due to the leg raise and lower manoeuvres performed by leg 3 and the relatively small displacements performed by the other legs' thighs.

In Figure 7.7a the x_B^I and y_B^I position of the rover in the physical traverse is compared with the planned path and simulated traverse. It is noted that the reported position in the physical experiment is more jagged than that of the plan and simulation. This may be attributed to noise in the localization solution provided by the fiducial localization system. It is also observed that in the physical trial the robot starts moving slightly behind the initial position used in the simulated trial.

In Figure 7.7b the physical hip motions are seen to track the simulated trial, however an offset is present in part due to the physical trial's initial position offset. The effectiveness of hip tracking is also noted in Figure 7.7c where the mechanical work cost of each of the hips have similar trends between the plan, simulation and physical trial. The hip tracking performance is reiterated in Figure 7.4 where RMSE values for each hip cost were below 50 J and NRMSE values below 0.2. The thigh joints do not have as effective tracking performance as shown in Figures 7.7b and 7.7c. This is attributed to the limited velocity control on each of the linear actuators. The NRMSE values for the hip actuators are generally multiple times higher than the hip NRMSE values for this reason, as shown in Figure 7.4a. For the physical trials the linear actuators are driven at a constant speed of 11 mm/s to make it easier to synchronize all thighs. The lack of low-speed actuator control results in discrepancies between the physical traverse and planned paths.

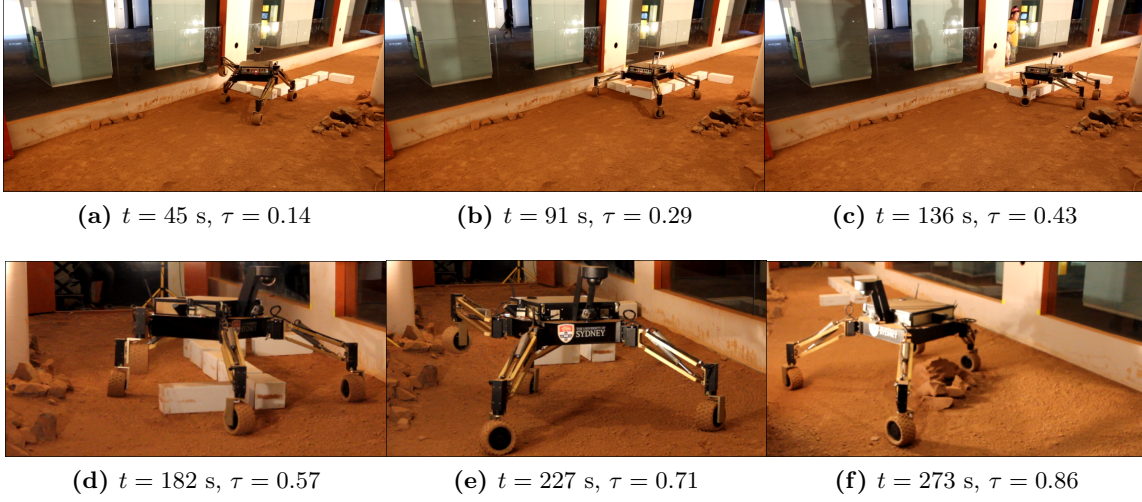


Figure 7.3 – Timelapse of the Trial 61 traverse with a time duration of 318 s. Images (a-c) show the traverse from a point of view close to the initial state, while images (d-f) show the traverse from a point of view close to the goal state.

7.2.2 Trial 61: Clambering Example

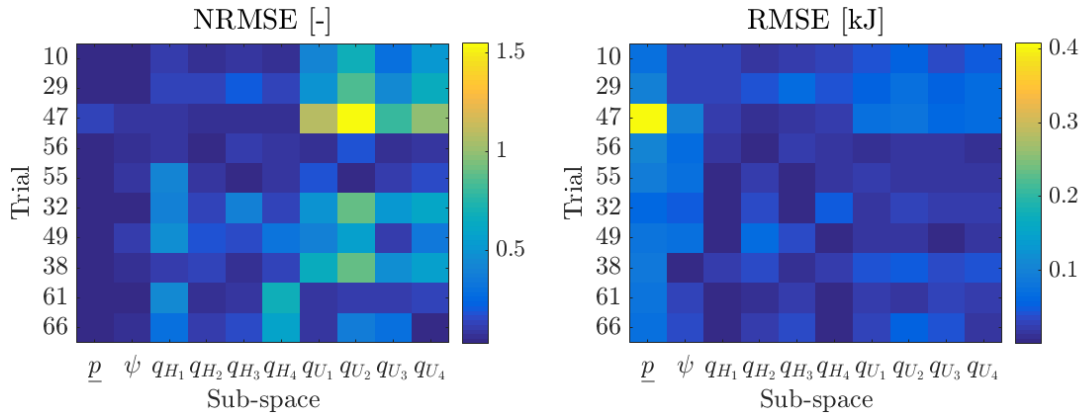
Trial 61 uses HBFMT* to plan a path through the Mars Lab Corridors environment. The traverse uses a batch of $n^\# = 5000$ samples for subspace exploration, a batch of $n = 10,000$ samples for the full space exploration and a tip-over stability constraint of $\beta = 14.3^\circ$. Two angular constraints were applied, with $q_{H_1} = 0^\circ$ and $q_{H_4} = 0^\circ$. HBFMT* was used with the parameter values: $\ell = 0.2$, $r_{tunnel} = 0.2$ kJ and $r_{sing} = 2.0$ m. The plan was generated in 136.5 s with a mechanical work cost of 4.4 kJ. Two alternative paths were found through the sub-space. The simulation result used 4.2 kJ while the physical trial used 4.39 kJ in mechanical work cost.

A video of Trial 61 may be viewed at https://youtu.be/2vwEwn_SwGE. Like in trial 56 the rover begins moving slightly behind the desired initial state. Its first set of motions move the rover to the desired initial state by driving the whole platform forward and to the right, while also slightly raising the platform. The rover then begins to drive through open space towards the ‘S’-shaped obstacle while also performing a slight counter-clockwise yaw. At 0:14 the rover begins to rotate hips 2 and 3 in preparation for a tripod configuration with leg 1 lifted. At 0:39 leg 1 is lifted so that the rover can move it over the base of the ‘S’-shaped obstacle. At 1:15 leg 1 is being moved over the obstacle while soon after hip 2 begins to rotate back towards $q_{H_2} = 0^\circ$. At 1:28 leg 1 begins to lower and a static instability is approached as hip 2 continues to be rotated. At 1:39 leg 1 has not lowered fast enough and the robot becomes unstable and falls onto leg 1. While falling, leg 1 continues to lower, however it does not reach the same level as the other thighs leaving the robot with a slight pitch and roll. At 2:10 the rover begins to yaw in a counter-clockwise direction so that legs 2 and 3 will pass around the base of the ‘S’-shaped obstacle. While this is happening, hips 2 and 3 rotate in a positive direction to form a tripod stance with leg 1, preparing for a lift of leg 3. At 2:29 leg 3 is lifted so that it does not hit the top of the obstacle. At 2:41 leg 4

passes through the narrow passageway, and at 2:54 the platform yaws positively to bring leg 2 through the narrow passageway. At 3:29 the platform begins to yaw in the clockwise direction to prepare for the traverse over the Home Plate obstacle. At 4:32 the platform begins to pass over the final rock obstacle straddling the rock initially with legs 1 and 2. By 4:52 the rock obstacle has been traversed and there is a yaw motion back to the goal state.

This traverse demonstrates the robot's ability to clamber over an obstacle. Additionally, it demonstrates the ability to impose actuator constraints within the planner. The clambering manoeuvre is planned for by using the feedback sampling method described in Section 6.4.1. Given that the two front hips are constrained the robot must rely on the two back hips for reconfiguration and ensuring that static stability is maintained. The first leg raise manoeuvre lifts leg 1 over the 'S'-shaped obstacle. To do this it is critical that the hip 2 swing around to a negative value to ensure static stability. It is noted that thigh 1 does not lower fast enough when leg 1 is being lowered back into ground contact to prevent an unstable state from being reached and the rover tipping onto leg 1. This is caused by the path following controller not issuing a thigh actuation value until it is too late. The lack of controllability of the thigh actuation controller is the root cause of this problem. Even so, the rover's traverse is not detrimentally affected by this tip-over with the robot simply tipping into a leg 1 contact state.

The RMSE values for mechanical work cost in Trial 61 shown in Figure 7.4a demonstrate that the planned paths and physical trials return relatively similar results. The highest NRSME values are seen in the two constrained sub-spaces q_{H_1} and q_{H_4} . This behaviour is due to the small maximum range in the denominator when calculating NRMSE for a constrained sub-space.



(a) The normalized root mean squared error (NRMSE) for each sub-space in each trial. (b) The root mean squared error (RMSE) for each sub-space in each trial.

Figure 7.4 – Comparison between the physical paths followed and the planned paths.

7.2.3 Planner Validation

This section analyzes how well the planned paths and physical traversals agree with each other. In doing so the planner's ability to find a path for a physical reconfigurable system is validated. Furthermore, the cost function used by the planner is compared with the actual energy consumption of the rover during a traversal.

Figure 7.4 shows the RMSE and NRMSE between the mechanical work costs for each sub-space for planned paths and physically followed paths in each trial. Figure 7.5 presents the mechanical work costs calculated using Eq. (6.2) from each of the physical trials and planned paths.

In Figure 7.5, the percentages of total cost attributed to each sub-space are visualized in each bar division. Every odd bar with a 'P' at the beginning of its trial name is associated with a physical trial, while every even bar that has a 'S' at the beginning of its trial name is associated with a planned path. It is observed in each of these plots that the mechanical work due to translation of robot is the motion that dominates total cost, with over 50% of the cost in each trial. The next most dominant mobility mode is the body yaw ψ , except for trial 4, which had its yaw constrained to $\psi = 0^\circ$. There is a general tendency across all trials, except for Trial 61, for the translation mechanical work in the physical trials to be greater than the planned paths. This may be attributed to noise in the physical localization solution, as shown in Figure 7.7a.

Agreement between mechanical work cost in the simulated traversals and the physical traversals has been demonstrated, however the actual energy expenditure is not given by this cost

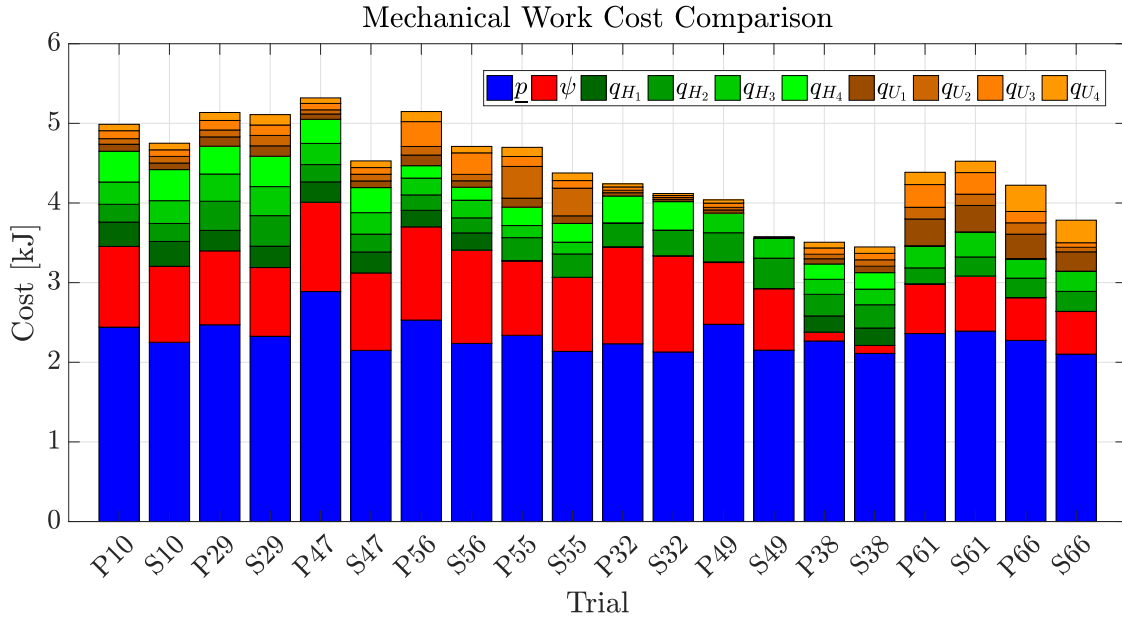


Figure 7.5 – Successful traversal costs that are proportional to the translational and angular displacements of the MAMMOTH rover. Physical and simulated trials are compared side by side with 'P' denoting a physical trial and 'S' denoting a planned path.

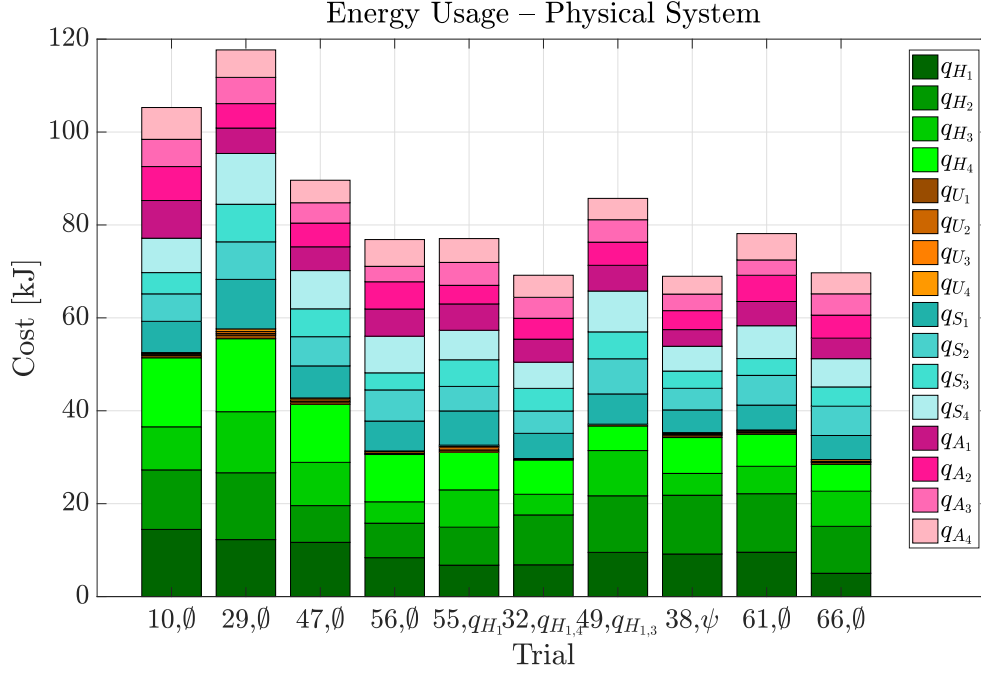


Figure 7.6 – Energy cost calculated from current draw and voltage of each of the MAMMOTH rover’s actuators.

function. The rover’s true energy expenditure is calculated as

$$Q = \sum_{i=0}^m \sum_{t_j=1}^{t_f} I_i(t_j) V_i(t_j) (t_j - t_{j-1}) \quad (7.2)$$

where Q is the total traversal energy expenditure in Joules, $I_i(t_j)$ is the current drawn by actuator i at time t_j , $V_i(t_j)$ is the operating voltage of actuator i at time t_j and m is the number of actuators. Figure 7.6 presents the actual energy expenditure by the MAMMOTH rover during each successful traverse.

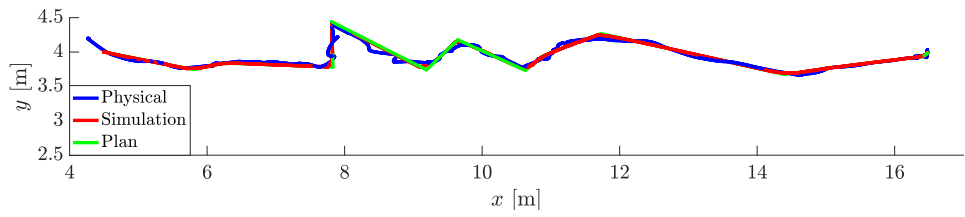
There is an obvious discrepancy between the actual energy costs and the mechanical work costs, with the actual energy costs being an order of magnitude larger. The first discrepancy is seen with the introduction of the ankle actuators q_S . These actuators are responsible for changing the ankle angle of each of the wheels and are not considered in the mechanical work cost function as their displacement is a function of the velocity of the platform, which is not considered by the planner. The energy required to move these actuators is dependent on the loading on a leg as well as the coefficient of friction between the wheel contact ellipse and the ground. The ankle joint angle is a function of the velocity of the body frame as well as the angular velocities of the associated leg joints as described in Eq. (4.21). An item of future work would be to make the planner kinodynamic and incorporate the velocity state of the vehicle. From Figure 7.6 it is observed that the total ankle actuator energy cost is similar to that of the total hip joint energy cost and larger than the total wheel drive energy cost. It is noted that in most trials that there were many jerky ankle actuator

motions that were a direct result of joint actuator saturation and sudden stopping. Energy cost for wheel steering would be reduced by introducing smooth joint velocity transitions as well as incorporation of the steering cost into the planner.

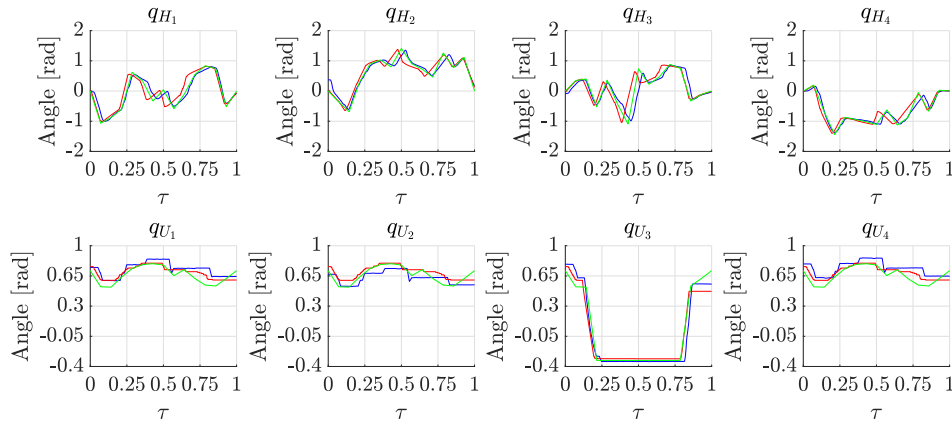
From Figure 7.6 it can be seen that the energy cost accrued by the hip actuators is significant and in the majority of trials the most costly sub-spaces. The energy required for the hip actuators does not reflect costs only proportional to displacement, as is obvious in the bar for Trial 49 where q_{H_1} and q_{H_4} were constrained yet still require significant amounts of energy to hold their respective positions over the traverse. The energy required for holding position is not considered in the planner's cost function and is an obvious contributor to total energy cost. It is also noted that the energy cost due to the thigh joints \underline{q}_U is marginal compared to the other actuated joints. This is due to the non back-drivable linear thigh actuators that do not require power to hold a position. Given this hardware design feature the energy cost of operating the linear actuators is due to motion of the actuators and not due to holding forces. For future statically stable reconfigurable systems it may be desirable to use non back-drivable actuators to ensure that energy is not wasted in holding a configuration.

7.3 Summary

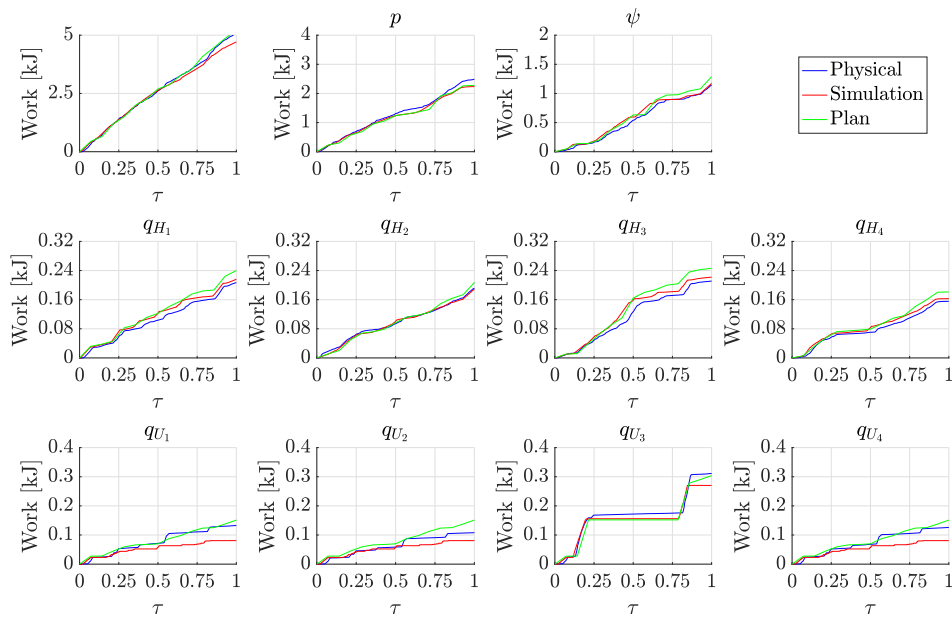
In summary, this chapter demonstrates the use of HBFMT* to plan a path for a physical RWMR. Planned paths are successfully followed by the MAMMOTH rover. It is shown that during these successful traverses that the executed path is generally in agreement with the planned path although with error introduced by noisy localization and poor low-speed control of the linear actuators. Leveraging of redundant actuation is demonstrated by introducing actuator constraints on hips and the platform's yaw. Disagreement between mechanical work cost function and the actual energy usage is noted and motivates future work in incorporating a high fidelity energy cost function into the planner as well as reliance on non-backdrivable actuators to minimize energy usage. The successful traverses presented are amongst the first examples of an FMT* variant being used to plan for a physical robotic system, and to the best of the author's knowledge the first for a WMR, let alone an RWMR.



(a) Body position.

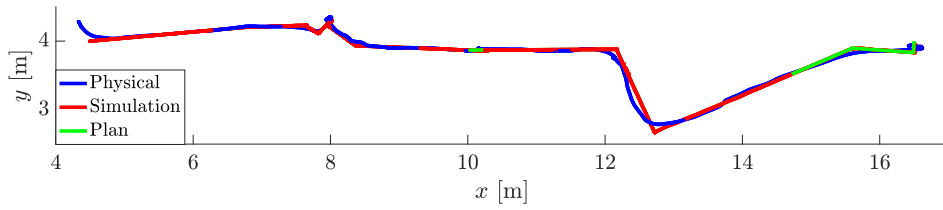


(b) Actuator position.

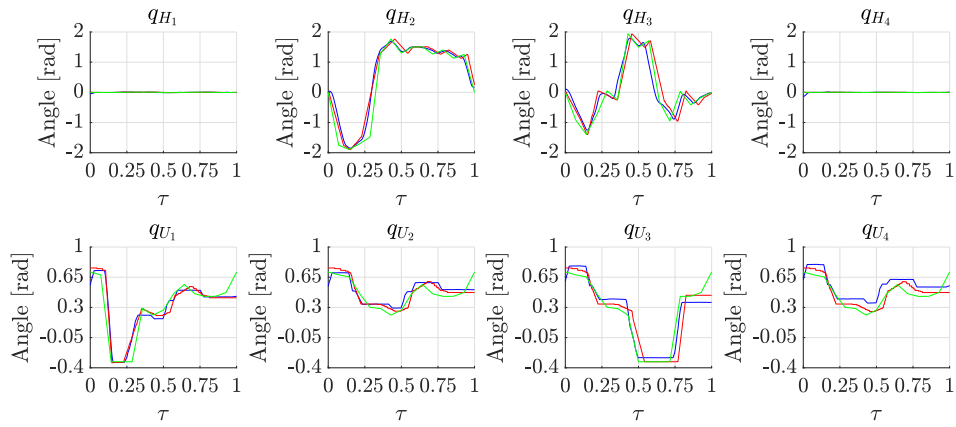


(c) Mechanical work costs.

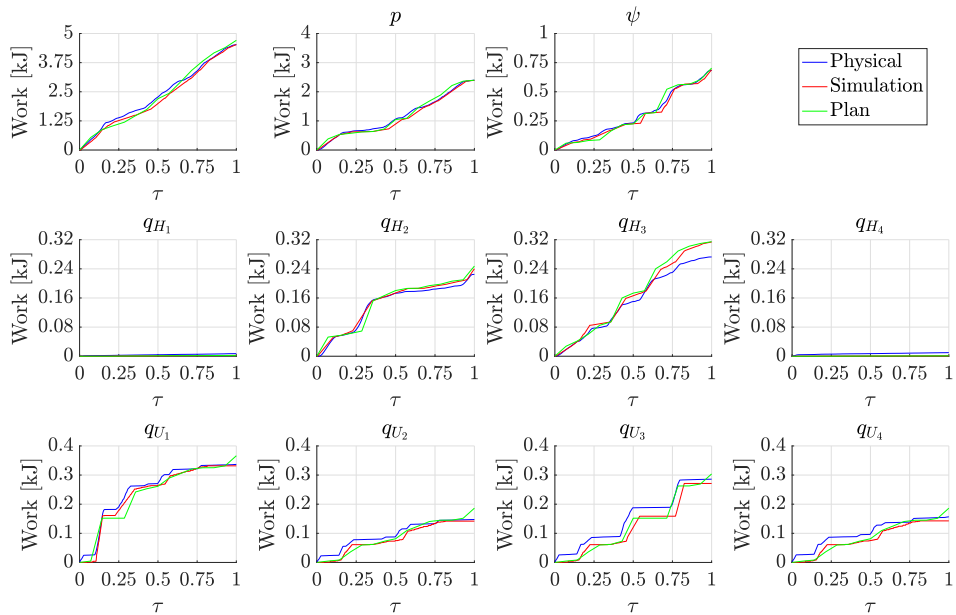
Figure 7.7 – Trial 56 plots comparing workspace position, actuator position and mechanical work costs between the physical trials (blue), simulated trials (red) and planned paths (green).



(a) Body position.



(b) Actuator position.



(c) Mechanical work costs.

Figure 7.8 – Trial 61 plots comparing workspace position, actuator position and mechanical work costs between the physical trials (blue), simulated trials (red) and planned paths (green).

Chapter 8

Conclusion

The aim of this thesis is to develop an autonomous planner used by a real-world RWMR system to find paths over unstructured terrain that leverage its many degrees of freedom. This aim has been realized, which is an important step in making the case for the use of RWMRs in extraterrestrial surface operations. With the capabilities demonstrated in this thesis, a Martian or lunar RWMR would be able to successfully traverse known obstacle-rich environments enabling access to previously inaccessible scientifically interesting sites. Additionally, such a planning system could be employed by terrestrial RWMRs that operate in unstructured environments, such as those used in forestry, mining and search and rescue.

A significant result of this work is the generation of planned paths that utilize versatile and fluid RWMR locomotion, which to a human operator may be unintuitive. By structuring the planning task in a hierarchical manner the intuitive sub-dimensional structure of the problem is quickly explored to then allow for a more focussed search of high-dimensional spaces to uncover these unintuitive behaviours. This work demonstrates some of the first physical planning trials for a robot that uses its reconfigurability for locomotion as well as some of the first physical realizations of hierarchical planning for reconfigurable robotic platforms.

8.1 Principal Contributions

The principal contributions of this thesis include the design of the MAMMOTH rover, a reconfigurable wheel-on-leg system. The system is designed to maximize limb reach so as to provide many locomotion and geometric morphing options for terrain traversal.

This thesis draws on the generalized kinematic modelling methodology presented in [65]. The thesis expands on this work by applying the modelling methodology to actively articulated RWMRs. Manipulation of the kinematic model is performed to assemble rowing and inchworming manoeuvres, and to perform an actively articulated suspension technique. It is straightforward to see how an RWMR may use these locomotion modes to get out of difficult situations like wheels embedding in soft soil or articulating over step obstacles. Additionally, this motion modelling enables RWMR model predictive planning: relationships

between each of the vehicle’s actuated joints and a way to quantify static stability provide a mathematical model to aid in sampling valid states and connecting them.

The proposed HBFMT* planner is amongst the first hierarchical planners to use continuous domain sampling-based planning at all hierarchical levels, thereby making the planner both simpler to implement and immune to the configuration space discretization pitfalls of grid-based planners. The planner takes advantage of the FMT* with biased sampling asymptotic optimality guarantees presented in [60] to retain AO. Bidirectional search is utilized to speed up computation time at the low-level of the hierarchy and to find multiple candidate paths in the sub-dimensional state space explored at the high level. Numerical results demonstrate that HBFMT* returns more feasible lower cost paths in a shorter amount of time compared with other AO planners such as the FMT* planners, BIT*, RRT* and Informed-RRT*.

This thesis puts a heavy emphasis on experimentally validating modelling and planning techniques with robotic hardware. The demonstration of FMT* sampling-based planning working with real robotic hardware is, to the best of the author’s knowledge, the first performed with a physical WMR. It is also one of the first works to demonstrate the use of asymptotically optimal sampling-based planning for global full body planning of a physical RWMR. Moreover, the planner validation experiments demonstrate novel capabilities such as clambering via feedback sampling and accounting for ‘broken’ actuators or sensor pointing constraints by applying motion constraints to sampling routines.

8.2 Future Work

This work has attempted to tie together state-of-the-art planning theory with a real world system to both validate the theory and to expose shortcomings in the application of the theory to the physical robot. Naturally, future work focusses on addressing these shortcomings.

Regarding future RWMR designs, it is observed in Chapter 7 that the energy cost of the non back-drivable linear actuators in each of the MAMMOTH rover thigh joints was minimal compared to the other joints that had to expend energy to hold their positions. For future RWMR designs, it would be highly desirable to use similar non back-drivable actuators at the hip joints, which were observed to consume the majority of energy during each traverse even when their motion was constrained.

Experimental validation of inchworming, rowing and actively articulated suspension highlighted some of the possible locomotion modes available to RWMRs. It would be desirable to expand on this and perform these mobility modes on different Martian and lunar analogues with varying degrees of slip and slope to gain a better understanding of traverse efficiency. Additionally, actively articulated suspension could be developed to better control tractive forces so as to reduce the likelihood of wheel slippage.

The prospects for actively articulated suspension becomes even more promising with an increased number of leg joints. Such a mechanism would allow a wheel to remain at a single point of contact while the legs and/or body are being moved. Such an RWMR design is presented in [26]. By exploiting the null space of the leg, wheel motion could be removed

completely from the body motion in a rowing type mobility mode. Increased kinematic redundancy within a leg would also allow for a greater ability to follow terrain profiles and to reconfigure geometry for reducing the amount of torque required to perform a traverse. As done in the manipulator arm literature, manipulability metrics described in [137] could be used to characterize how ‘suspendable’ the rover is. Such a metric could be used within a motion controller or planner to aid in selecting appropriate body motions.

Regarding sampling-based planners, one of the disadvantages of using FMT* is that it is not an anytime planner. FMT* relies on connecting a single batch of samples and only returns a solution after all of the samples between the initial and goal nodes have been processed. Anytime variants of FMT* have been proposed with MPLB [102] and more loosely with BIT* [40]. In environments where a close to straight line solution is not available, these anytime variants perform worse than FMT*. Extra computational effort is required by expanding a wavefront on every planner iteration and re-wiring the existing tree. The parallelization of FMT* could address this issue, where individual FMT* wavefronts are expanded one after the other on separate processors so that even though more computation may be required than the single-core FMT*, the resulting anytime planner returns as good or better quality solutions within an equivalent computation time.

The use of sub-dimensional decomposition to bias sampling has obvious performance effects, as demonstrated in the numerical experiments from Section 6.5 and the work presented in both [85] and [94]. It is also evident in [101] and [128] that learning sub-dimensional decomposition results in increased planning efficiency. The HBFMT* algorithm is flexible in that more levels may be added into the planning hierarchy besides just planning in the workspace and full state space. A future step in the development of this algorithm would be to integrate a learned intermediate sub-dimensional space. This would most likely take the form of planning over sets of motion primitives or task space regions like those described in [15] and [48].

A lesson learned from the plan following experiments is that a high-fidelity energy model of the robot is required to better capture the predicted energy expenditure of the rover. Usage of a fast high fidelity dynamic model formulation like that presented in [112] to evaluate candidate traversals between waypoints may help address this problem. Similarly, the non-holonomic ankle joints were not accounted for in the HBFMT* planner state space formulation. The planning examples presented in this work considered the geometric planning problem and it would be an obvious extension of this work to formulate a kinodynamic version of the planning problem like the one presented in [108].

Bibliography

- [1] E. Allouis and Y. Gao, *Planetary Robotic System Design*. John Wiley & Sons, 2016.
- [2] O. Arslan and P. Tsiotras, “Use of relaxation methods in sampling-based algorithms for optimal motion planning,” in *Proc. of IEEE ICRA*, 2013, pp. 2421–2428.
- [3] R. E. Arvidson, S. W. Squyres, R. C. Anderson, J. F. Bell *et al.*, “Overview of the Spirit Mars Exploration Rover mission to Gusev Crater: landing site to Backstay Rock in the Columbia Hills,” *J. Geophys. Res.*, vol. 111, no. 2, 2006.
- [4] R. E. Arvidson, K. D. Iagnemma, M. Maimone, A. A. Fraeman *et al.*, “Mars Science Laboratory Curiosity rover megaripple crossings up to sol 710 in Gale crater,” *J. Field Robot.*, vol. 34, no. 3, pp. 495–518, 2017.
- [5] R. Arvidson, J. Bell, P. Bellutta, N. Cabrol *et al.*, “Spirit Mars Rover Mission: Overview and selected results from the northern Home Plate Winter Haven to the side of Scamander crater,” *J Geophys. Res.-Planet*, vol. 115, no. E7, 2010.
- [6] R. Arvidson, S. Squyres, J. Bell, J. Catalano *et al.*, “Ancient aqueous environments at Endeavour crater, Mars,” *Science*, vol. 343, no. 6169, p. 1248097, 2014.
- [7] ASUS, “ASUS Xtion Pro Live,” 2017, [Accessed: 2017-Jul-26]. [Online]. Available: https://www.asus.com/3D-Sensor/Xtion_PRO_LIVE/specifications/
- [8] *ATmega640/V-1280/V-1281/V-2560/V-2561/V*, Atmel, Feb. 2014, rev. 2549-02/2014.
- [9] N. Ayanian and V. Kumar, “Decentralized feedback controllers for multiagent teams in environments with obstacles,” *IEEE Trans. Robot.*, vol. 26, no. 5, pp. 878–887, 2010.
- [10] M. Bajracharya, M. Maimone, and D. Helmick, “Autonomy for Mars rovers: Past, present, and future,” *Computer*, vol. 41, no. 12, pp. 44–50, 2008.
- [11] D. Bapna, E. Rollins, J. Murphy, E. Maimone *et al.*, “The Atacama desert trek: Outcomes,” in *Proc. of IEEE ICRA*, 1998, pp. 597–604.
- [12] P. W. Bartlett, D. Wettergreen, and W. Whittaker, “Design of the Scarab Rover for Mobility & Drilling in the Lunar Cold Traps,” in *Proc. of i-SAIRAS*, 2008.

- [13] BeagleBoard, “Beaglebone black,” 2017, [Accessed: 2017-Jul-26]. [Online]. Available: <https://beagleboard.org/black>
- [14] D. Belter, P. Labecki, and P. Skrzypczynski, “Adaptive motion planning for autonomous rough terrain traversal with a walking robot,” *J. Field Robot.*, vol. 33, no. 3, pp. 337–370, 2015.
- [15] D. Berenson, S. Srinivasa, and J. J. Kuffner, “Task space regions: A framework for pose-constrained manipulation planning,” *Int. J. Robot. Res.*, vol. 30, no. 12, pp. 1435–1460, 2011.
- [16] D. B. Bickler, “Articulated suspension system,” U.S. Patent 4 840 394, 1989. [Online]. Available: <https://www.google.com/patents/US4840394>
- [17] J. J. Biesiadecki, P. C. Leger, and M. W. Maimone, “Tradeoffs between directed and autonomous driving on the Mars Exploration Rovers,” *Int. J. Robot. Res.*, vol. 26, pp. 91–104, 2007.
- [18] J. Biesiadecki, M. W. Maimone, and J. Morrison, “The Athena SDM rover: A testbed for Mars rover mobility,” in *Proc. of i-SAIRAS*, 2001.
- [19] R. Bohlin and L. E. Kavraki, “Path planning using lazy PRM,” in *Proc. of IEEE ICRA*, 2000, pp. 521–528.
- [20] O. Brock and L. Kavraki, “Decomposition-based motion planning: A framework for real-time motion planning in high-dimensional configuration spaces,” in *Proc. of IEEE ICRA*, 2001, pp. 1469–1474.
- [21] M. Brunner, T. Fiolka, D. Schulz, and C. M. Schlick, “Design and comparative evaluation of an iterative contact point estimation method for static stability estimation of mobile actively reconfigurable robots,” *Robot. Auton. Syst.*, vol. 63, pp. 89–107, 2015.
- [22] G. Campion, G. Bastin, and B. Dandrea-Novet, “Structural properties and classification of kinematic and dynamic models of wheeled mobile robots,” *IEEE Trans. Robot. Autom.*, vol. 12, no. 1, pp. 47–62, 1996.
- [23] J. Carsten, A. Rankin, D. Ferguson, and A. Stentz, “Global planning on the Mars Exploration Rovers: software integration and surface testing,” *J. Field Robot.*, vol. 26, no. 4, pp. 337–357, 2009.
- [24] C. P. Connette, C. Parlitz, M. Hagele, and A. Verl, “Singularity avoidance for over-actuated, pseudo-omnidirectional, wheeled mobile robots,” in *Proc. of IEEE ICRA*. IEEE, 2009, pp. 4124–4130.
- [25] F. Cordes, “SherpaTT,” [Accessed: 2017-May-05]. [Online]. Available: <http://robotik.dfki-bremen.de/de/forschung/robotersysteme/sherpatt.html>
- [26] F. Cordes and A. Babu, “SherpaTT: A versatile hybrid wheeled-leg rover,” in *Proc. of i-SAIRAS*, 2016.

- [27] F. Cordes, A. Dettmann, and F. Kirchner, “Locomotion modes for a hybrid wheeled-leg planetary rover,” in *Proc. of IEEE ROBIO*. IEEE, 2011, pp. 2586–2592.
- [28] P. Corke, *Robotics, Vision and Control Fundamental Algorithms in MATLAB*, 1st ed. Berlin: Springer, 2011.
- [29] N. C. Costes, J. E. Farmer, and E. B. George, “Mobility performance of the lunar roving vehicle: Terrestrial studies: Apollo 15 results,” NASA, NASA TR-R-401, Tech. Rep., 1972.
- [30] R. C. Coulter, “Implementation of the pure pursuit path tracking algorithm,” CMU, CMU-RI-TR-92-01, Tech. Rep., 1992.
- [31] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
- [32] A. Dobson and K. E. Bekris, “Sparse roadmap spanners for asymptotically near-optimal motion planning,” *Int. J. Robot. Res.*, vol. 33, no. 1, pp. 18–47, 2014.
- [33] J. M. Esposito and J. N. Wright, “Matrix completion as a post-processing technique for probabilistic roadmaps,” in *Proc. of WAFR*, 2016.
- [34] D. Ferguson and A. Stentz, “Using interpolation to improve path planning: The Field D* algorithm,” *J. Field Robot.*, vol. 23, no. 2, pp. 79–101, 2006.
- [35] —, “Anytime RRTs,” in *Proc. of IEEE/RSJ IROS*, 2006, pp. 5369–5375.
- [36] FLIR, “Blackfly 2.3 MP Color GigE PoE (Sony IMX136),” [Accessed: 2017-07-26]. [Online]. Available: <https://www.ptgrey.com/blackfly-23-mp-color-gige-vision-poe-sony-imx136-camera>
- [37] Freetronics, “Ethermega,” 2016, [Accessed: 2017-Jul-26]. [Online]. Available: <https://www.freetronics.com.au/products/ethermega-arduino-mega-2560-compatible-with-onboard-ethernet#.WXg80SF97RY>
- [38] Q. Fu and V. Krovi, “Articulated wheeled robots: Exploiting reconfigurability and redundancy,” in *Proc. of ASME DSCC*, 2008, pp. 653–660.
- [39] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, “Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic,” in *Proc. of IEEE/RSJ IROS*, 2014, pp. 2997–3004.
- [40] —, “Batch Informed Trees (BIT*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs,” in *Proc. of IEEE ICRA*, 2015, pp. 3067–3074.
- [41] A. Ghasempoor and N. Sepehri, “A measure of machine stability for moving base manipulators,” in *Proc. of IEEE ICRA*, 1995, pp. 2249–2254.

- [42] B. Gipson, M. Moll, and L. E. Kavraki, "Resolution independent density estimation for motion planning in high-dimensional spaces," in *Proc. of IEEE ICRA*, 2013, pp. 2437–2443.
- [43] C. Grand, "Stability and traction optimization of a reconfigurable wheel-legged robot," *Int. J. Robot. Res.*, vol. 23, no. 10-11, pp. 1041–1058, 2004.
- [44] C. Grand, F. Benamar, and F. Plumet, "Motion kinematics analysis of wheeled-legged rover over 3D surface with posture adaptation," *Mech. Mach. Theory*, vol. 45, no. 3, pp. 477–495, 2010.
- [45] J. P. Grotzinger, D. Y. Sumner, L. Kah, K. Stack *et al.*, "A habitable fluvio-lacustrine environment at Yellowknife Bay, Gale Crater, Mars," *Science*, vol. 343, no. 6169, pp. 1 242 777–1–1 242 777–14, 2014.
- [46] P. E. Hart, N. Nil, and B. Raphael, "Formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cyb.*, vol. SSC-4, no. 2, pp. 100–107, 1968.
- [47] R. S. Hartenberg and J. Denavit, *Kinematic synthesis of linkages*. McGraw-Hill, 1964.
- [48] K. Hauser and J. C. Latombe, "Multi-modal motion planning in non-expansive spaces," *Int. J. Robot. Res.*, vol. 29, no. 7, pp. 897–915, 2010.
- [49] K. Hauser, T. Bretl, J.-C. Latombe, K. Harada *et al.*, "Motion planning for legged robots on varied terrain," *Int. J. Robot. Res.*, vol. 27, no. 11-12, pp. 1325–1349, 2008.
- [50] K. Hauser, "Lazy collision checking in asymptotically-optimal motion planning," in *Proc. of IEEE ICRA*, 2015, pp. 2951–2957.
- [51] S. Hayati, R. Volpe, P. Backes, J. Balaram *et al.*, "The Rocky 7 rover: A Mars sciencecraft prototype," in *Proc. of IEEE ICRA*, 1997.
- [52] G. C. Haynes, D. Stager, A. Stentz, J. M. Vande Weghe *et al.*, "Developing a robust disaster response robot: CHIMP and the Robotics Challenge," *J. Field Robot.*, vol. 34, no. 2, pp. 281–304, 2017.
- [53] P. Hebert, M. Bajracharya, J. Ma, N. Hudson *et al.*, "Mobile manipulation and mobility as manipulation - Design and algorithms of RoboSimian," *J. Field Robot.*, vol. 32, no. 2, pp. 255–274, 2015.
- [54] M. Heverly, J. Matthews, J. Lin, D. Fuller *et al.*, "Traverse performance characterization for the Mars Science Laboratory rover," *J. Field Robot.*, vol. 30, no. 6, pp. 835–846, 2013.
- [55] J. Hidalgo and F. Cordes, "Kinematics modeling of a hybrid wheeled-leg planetary rover," in *Proc. of i-SAIRAS*, 2012.
- [56] D. Hsu, "Randomized single-query motion planning in expansive spaces," Ph.D. dissertation, Stanford University, 2000.

- [57] K. Iagnemma, A. Rzepniewski, S. Dubowsky, P. Pirjanian *et al.*, “Mobile robot kinematic reconfigurability for rough-terrain,” in *Proc. of SPIE*, 2000, pp. 413–420.
- [58] InspiredEnergy, “NL2024 & RH2024 specification summary,” Inspired Energy, Jan. 2016, rev. 3.2. [Online]. Available: <http://www.inspired-energy.com/products.html>
- [59] Intel, “Intel NUC Kit D54250WYKH,” 2013, [Accessed: 2017-Jul-26]. [Online]. Available: <http://ark.intel.com/products/81164/Intel-NUC-Kit-D54250WYKH>
- [60] L. Janson, E. Schmerling, A. Clark, and M. Pavone, “Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions,” *Int. J. Robot. Res.*, vol. 34, no. 7, pp. 883–921, 2015.
- [61] M. Jordan and A. Perez, “Optimal bidirectional rapidly-exploring random trees,” MIT CSAIL, MIT-CSAIL-TR-2013-021, Tech. Rep., 2013.
- [62] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.
- [63] S. Karumanchi, K. Edelberg, I. Baldwin, J. Nash *et al.*, “Team RoboSimian: Semi-autonomous mobile manipulation at the 2015 DARPA Robotics Challenge finals,” *J. Field Robot.*, vol. 34, no. 2, pp. 305–332, 2016.
- [64] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, 1996.
- [65] A. Kelly and N. Seegmiller, “Recursive kinematic propagation for wheeled mobile robots,” *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 288–313, 2015.
- [66] J. Kober, J. A. Bagnell, and J. Peters, “Reinforcement learning in robotics: A survey,” *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1238–1274, 2013.
- [67] Kontron, “KTQM77/mITX,” 2013, [Accessed: 2017-Jul-26]. [Online]. Available: <https://www.kontron.com/products/boards-and-standard-form-factors/motherboards/mini-itx/ktqm77-mitx.html>
- [68] E. Krotkov, D. Hackett, L. Jackel, M. Perschbacher *et al.*, “The DARPA Robotics Challenge finals: Results and perspectives,” *J. Field Robot.*, vol. 34, no. 2, pp. 229–239, 2016.
- [69] J. J. Kuffner and S. M. LaValle, “RRT-Connect: An efficient approach to single-query path planning,” in *Proc. of IEEE ICRA*, 2000, pp. 995–1001.
- [70] S. M. LaValle, *Planning Algorithms*, 1st ed. Cambridge University Press, 2006.
- [71] S. M. LaValle and J. J. Kuffner, “Randomized kinodynamic planning,” *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, 2001.
- [72] J. Lim, I. Shim, O. Sim, H. Joe *et al.*, “Robotic software system for the disaster circumstances: System of team KAIST in the DARPA Robotics Challenge finals,” in *Proc. of IEEE-RAS ICHR*, 2015, pp. 1161–1166.

- [73] M. Maimone, J. Biesiadecki, E. Tunstel, Y. Cheng *et al.*, “Surface navigation and mobility intelligence on the Mars Exploration Rovers,” in *Intelligence for Space Robotics*, 2006, pp. 45–69.
- [74] M. Maimone, A. Johnson, Y. Cheng, R. Willson *et al.*, “Autonomous navigation results from the Mars Exploration Rover (MER) mission,” *Spr. Tra. Adv. Robot.*, vol. 21, pp. 3–13, 2006.
- [75] M. Malenkov, “Self-propelled automatic chassis of Lunokhod-1: History of creation in episodes,” *Front. of Mech. Eng.*, vol. 11, no. 1, pp. 60–86, 2016.
- [76] R. B. McGhee and A. A. Frank, “On the stability properties of quadruped creeping gaits,” *Math. Biosci.*, vol. 3, pp. 331–351, 1968.
- [77] D. A. Messuri and C. A. Klein, “Automatic body regulation for maintaining stability of a legged vehicle,” *IEEE Trans. Robot. Autom.*, vol. RA-1, no. 3, pp. 132–141, 1985.
- [78] T. Moore and D. Stouch, “A generalized extended kalman filter implementation for the robot operating system,” in *Intelligent Autonomous Systems 13*. Springer, 2016, pp. 335–348.
- [79] P. F. Muir and C. P. Neuman, “Kinematic modeling of wheeled mobile robots,” *J. Robot. Syst.*, vol. 4, no. 2, pp. 281–340, 1986.
- [80] NASA, “Spirit embedded in soft soil on mars as engineers devise methods to ‘free spirit’,” 2009, [Accessed: 16-Mar-2017]. [Online]. Available: https://www.nasa.gov/multimedia/imagegallery/image_feature_1506.html
- [81] M. Otte and N. Correll, “C-Forest: Parallel shortest path planning with superlinear speedup,” *IEEE Trans. Robot.*, vol. 29, no. 3, pp. 798–806, 2013.
- [82] J. Pan, S. Chitta, and D. Manocha, “FCL: A general purpose library for collision and proximity queries,” in *Proc. of IEEE ICRA*, 2012, pp. 3859–3866.
- [83] E. G. Papadopoulos and D. A. Rey, “A new measure of tipover stability margin for mobile manipulators,” in *Proc. of IEEE ICRA*, vol. 33, no. 1, 1996, pp. 3111–3116.
- [84] M. Pivtoraiko, R. A. Knepper, and A. Kelly, “Differentially constrained mobile robot motion planning in state lattices,” *J. Field Robot.*, vol. 26, no. 3, pp. 308–333, 2009.
- [85] E. Plaku, L. E. Kavraki, and M. Y. Vardi, “Motion planning with dynamics by a synergistic combination of layers of planning,” *IEEE Trans. Robot.*, vol. 26, no. 3, pp. 469–482, 2010.
- [86] F. Pokorný, M. Hawasly, and S. Ramamoorthy, “Topological trajectory classification with filtrations of simplicial complexes and persistent homology,” *Int. J. Robot. Res.*, vol. 35, no. 1-3, pp. 204–223, 2016.
- [87] Pololu, “Pololu dual vnh5019 motor driver shield for arduino,” 2017, [Accessed: 26-Jul-2017]. [Online]. Available: <https://www.pololu.com/product/2507>

- [88] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, “Comparing ICP variants on real-world data sets: Open-source library and experimental protocol,” *Auton. Robot.*, vol. 34, no. 3, pp. 133–148, 2013.
- [89] M. Quigley, K. Conley, B. Gerkey, J. Faust *et al.*, “ROS: An open-source Robot Operating System,” in *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009, p. 5.
- [90] W. Reid, A. H. Göktoğan, and S. Sukkarieh, “Moving MAMMOTH: Stable motion for a reconfigurable wheel-on-leg rover,” in *Proc. of ARAA ACRA*, 2014.
- [91] W. Reid, R. Fitch, A. H. Göktoğan, and S. Sukkarieh, “Motion planning for reconfigurable mobile robots using hierarchical fast marching trees,” in *Proc. of WAFR*, 2016.
- [92] W. Reid, F. J. Pérez-Grau, A. H. Göktoğan, and S. Sukkarieh, “Actively articulated suspension for a wheel-on-leg rover operating on a Martian analog surface,” in *Proc. of IEEE ICRA*, 2016, pp. 5596–5602.
- [93] J. H. Reif, “Complexity of the mover’s problem and generalizations,” in *Proc. of IEEE FOCS*, 1979, pp. 421–427.
- [94] M. Rickert, O. Brock, and A. Knoll, “Balancing exploration and exploitation in motion planning,” *IEEE Trans. Robot.*, vol. 30, no. 6, pp. 2812–2817, 2014.
- [95] Robotis, “Dynamixel Pro H54-100-S500-R,” 2010, [Accessed: 2017-Jul-05]. [Online]. Available: http://en.robotis.com/index/product.php?cate_code=101111&bbs_no=31#product_title
- [96] —, “Dynamixel Pro L54-50-S290-R,” 2010, [Accessed: 2017-Jul-05]. [Online]. Available: http://en.robotis.com/index/product.php?cate_code=10111112&bbs_no=36#product_title
- [97] —, “Robotis Dynamixel SDK,” 2017, [Accessed: 2017-Jul-26]. [Online]. Available: <https://github.com/ROBOTIS-GIT/DynamixelSDK>
- [98] T. M. Roehr, F. Cordes, and F. Kirchner, “Reconfigurable Integrated Multirobot Exploration System (RIMRES): Heterogeneous modular reconfigurable robots for space exploration,” *J. Field Robot.*, vol. 31, no. 7, pp. 31–34, 2014.
- [99] E. Rollins, J. Luntz, A. Foessel, B. Shamah *et al.*, “Nomad: A demonstration of the transforming chassis,” in *Proc. of IEEE ICRA*, 1998, pp. 611–617.
- [100] E. Rosten and T. Drummond, “Machine learning for high speed corner detection,” in *Proc. of Computer Vision–ECCV*, 2006, pp. 430–443.
- [101] J. Rowekamper, G. D. Tipaldi, and W. Burgard, “Learning to guide random tree planners in high dimensional spaces,” in *Proc. of IEEE/RSJ IROS*, 2013, pp. 1752–1757.

- [102] O. Salzman and D. Halperin, “Asymptotically-optimal motion planning using lower bounds on cost,” in *Proc. of IEEE ICRA*, 2015, pp. 4167–4172.
- [103] O. Salzman, D. Shaharabani, P. K. Agarwal, and D. Halperin, “Sparsification of motion-planning roadmaps by edge contraction,” *Int. J. Robot. Res.*, vol. 33, no. 14, pp. 4098–4105, 2013.
- [104] G. Sanchez and J.-C. Latombe, “A single-query bi-directional probabilistic roadmap planner with lazy collision checking,” in *Proc. of ISRR*, 2003, pp. 403–417.
- [105] B. W. Satzinger, J. I. Reid, M. Bajracharya, P. Hebert *et al.*, “More solutions means more problems: Resolving kinematic redundancy in robot locomotion on complex terrain,” in *Proc. of IEEE/RSJ IROS*, 2014, pp. 4861–4867.
- [106] B. W. Satzinger, C. Lau, M. Byl, and K. Byl, “Tractable locomotion planning for RoboSimian,” *Int. J. Robot. Res.*, vol. 34, no. 13, pp. 1541–1558, 2015.
- [107] P. S. Schenker, E. T. Baumgartner, P. G. Backes, H. Aghazarian *et al.*, “FIDO: A Field Integrated Design & Operations rover for Mars surface exploration,” in *Proc. of i-SAIRAS*, 2001.
- [108] E. Schmerling, L. Janson, and M. Pavone, “Optimal sampling-based motion planning under differential constraints: The driftless case,” in *Proc. of IEEE ICRA*, 2015, pp. 2368–2375.
- [109] J. T. Schwartz, “On the “piano movers” problem. II. General techniques for computing topological properties of real algebraic manifolds,” *Adv. Appl. Math.*, vol. 4, no. 3, pp. 298–351, 1983.
- [110] M. Schwarz, T. Rodehutsors, D. Droschel, M. Beul *et al.*, “NimbRo Rescue: Solving disaster-response tasks with the mobile manipulation robot Momaro,” *J. Field Robot.*, vol. 34, no. 2, pp. 400–425, 2017.
- [111] N. Seegmiller, “Dynamic model formulation and calibration for wheeled mobile robots,” Ph.D. dissertation, Carnegie Mellon University, 2014.
- [112] N. Seegmiller and A. Kelly, “High-fidelity yet fast dynamic models of wheeled mobile robots,” *IEEE Trans. Robot.*, vol. 32, no. 3, pp. 614–625, 2016.
- [113] J. A. Sethian, “A fast marching level set method for monotonically advancing fronts,” in *Proc. of PNAS*, 1996, pp. 1591–1595.
- [114] D. Shirley and J. Matijevic, “Mars Pathfinder microrover,” *Auton. Robot.*, vol. 2, pp. 283–289, 1995.
- [115] P. H. Smith, L. Tamppari, R. Arvidson, D. Bass *et al.*, “H₂O at the Phoenix landing site,” *Science*, vol. 325, no. 5936, pp. 58–61, 2009.
- [116] S. W. Squyres, R. E. Arvidson, J. F. Bell, J. Brückner *et al.*, “The Spirit rover’s Athena science investigation at Gusev Crater, Mars,” *Science*, vol. 305, no. 5685, pp. 794–799, 2004.

- [117] S. Squyres, A. Knoll, R. Arvidson, B. Clark *et al.*, “Two years at Meridiani Planum: Results from the Opportunity rover,” *Science*, vol. 313, no. 5792, pp. 1403–1407, 2006.
- [118] J. A. Starek, E. Schmerling, L. Janson, and M. Pavone, “Bidirectional fast marching trees: An optimal sampling-based algorithm for bidirectional motion planning,” in *Proc. of WAFR*, 2014.
- [119] A. Stentz, H. Herman, A. Kelly, E. Meyhofer *et al.*, “CHIMP, the CMU Highly Intelligent Mobile Platform,” *J. Field Robot.*, vol. 32, no. 2, pp. 209–228, 2015.
- [120] I. Sucan, M. Moll, and L. E. Kavraki, “The open motion planning library,” *IEEE Robot. Autom. Mag.*, pp. 72–82, 2012.
- [121] V. SunSpiral, D. Wheeler, D. Chavez-Clemente, and D. Mittman, “Development and field testing of the FootFall planning system for the ATHLETE robots,” *J. Field Robot.*, vol. 22, no. 3, pp. 483–505, 2012.
- [122] M. Tarokh and G. J. McDermott, “Kinematics modeling and analyses of articulated rovers,” *IEEE Trans. Robot.*, vol. 21, no. 4, pp. 539–553, 2005.
- [123] M. Tarokh, H. D. Ho, and A. Bouloubasis, “Systematic kinematics analysis and balance control of high mobility rovers over rough terrain,” *Robot. Auton. Syst.*, vol. 61, pp. 13–24, 2013.
- [124] D. R. Thompson, D. S. Wettergreen, and F. J. C. Peralta, “Autonomous science during large-scale robotic survey,” *J. Field Robot.*, vol. 28, no. 4, pp. 542–564, 2011.
- [125] A. Tompkins, S. Potiris, and A. H. Göktoğan, “Development of a low-cost vision-based localisation system for the Experimental Mars Rover,” in *Proc. of ASSC*, 2013, pp. 283–294.
- [126] B. Trease, R. Arvidson, R. Lindemann, K. Bennett *et al.*, “Dynamic modeling and soil mechanics for path planning of the Mars Exploration Rovers,” in *Proc. of ASME IDETC/CIE*, 2011.
- [127] VectorNav, “VN-200 GPS/INS,” [Accessed: 2017-07-26]. [Online]. Available: http://www.vectornav.com/docs/default-source/documentation/vn-200-documentation/PB-12-0003.pdf?sfvrsn=749ee6b9_13
- [128] P. Vernaza and D. D. Lee, “Learning and exploiting low dimensional structure for efficient holonomic motion planning in high dimensional spaces,” *Int. J. Robot. Res.*, vol. 31, no. 14, pp. 1739–1760, 2012.
- [129] G. Wagner, M. Kang, and H. Choset, “Probabilistic path planning for multiple robots with subdimensional expansion,” in *Proc. of IEEE ICRA*, 2012, pp. 2886–2892.
- [130] M. Wagner, S. Heys, D. Wettergreen, J. Teza *et al.*, “Design and control of a passively steered, dual axle vehicle,” in *Proc. of i-SAIRAS*, 2005.

- [131] M. Wermelinger, R. Diethelm, P. Kruesi, R. Siegwart *et al.*, “Navigation planning for legged robots in challenging terrain,” in *Proc. of IEEE/RSJ IROS*, 2016, pp. 1184–1189.
- [132] D. Wettergreen, D. Bapna, M. Maimone, and G. Thomas, “Developing Nomad for robotic exploration of the Atacama desert,” *Robot. Auton. Syst.*, vol. 26, no. 2-3, pp. 127–148, 1999.
- [133] D. Wettergreen, D. Jonak, D. Kohanbash, S. Moreland *et al.*, “Field experiments in mobility and navigation with a lunar rover prototype,” in *Field and Service Robotics*, 2010, vol. 62, pp. 489–498.
- [134] B. H. Wilcox, T. Litwin, J. Biesiadecki, J. Matthews *et al.*, “ATHLETE: A cargo handling and manipulation robot for the Moon,” *J. Field Robot.*, vol. 24, no. 5, pp. 421–434, 2007.
- [135] WIZNet, “W5100 datasheet,” Atmel, Jul. 2011, rev. 1.1.6. [Online]. Available: http://www.wiznet.io/wp-content/uploads/wiznethome/Chip/W5100/Document/W5100_Datasheet_v1.2.7.pdf
- [136] D. Yi, M. A. Goodrich, and K. D. Seppi, “Homotopy-aware RRT*: Toward human-robot topological path-planning,” in *Proc. of ACM/IEEE HRI*, 2016, pp. 279–286.
- [137] T. Yoshikawa, “Manipulability of Robotic Mechanisms,” *The International Journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985. [Online]. Available: <http://ijr.sagepub.com/content/4/2/3.full.pdf>
- [138] C. Zhong and H. Liu, “A region-specific hybrid sampling method for optimal path planning,” *Int. J. Adv. Robot. Syst.*, vol. 13, 2016.

Appendix A

Kinematics

A.1 Recursive Kinematic Propagation Examples

This section provides examples of how RKP may be applied. It also demonstrates the kinematic model formulation for a single MAMMOTH rover leg.

A.1.1 MAMMOTH Rover Position Kinematics

To illustrate the usage of Eq. (4.2), the position and rotation of a MAMMOTH rover leg contact frame, C_i , is expressed with respect to the rover's body frame, I . In this case the kinematic branch's root frame is I ($j = 1$) and the leaf frame is C_i ($n = 11$). The relationships between each of these frames are shown in Figures 4.1 and 4.2, while the values of adjacent translation vectors and rotation matrices are given in Table 4.1. Using Eq. (4.2) recursively until only translations and rotations between adjacent frames are present results in:

$$\begin{aligned}
 \underline{r}_{C_i}^I &= \underline{r}_{11}^1 = \underline{r}_2^1 + R_2^1(\underline{r}_{11}^2) \\
 &= \underline{r}_2^1 + R_2^1(\underline{r}_3^2 + R_3^2 \underline{r}_{11}^3) \\
 &= \dots \\
 &= \underline{r}_2^1 + R_2^1 \underline{r}_3^2 + R_2^1 R_3^2 \underline{r}_4^3 + \dots + R_2^1 R_3^2 R_4^3 R_5^4 R_6^5 R_7^6 R_8^7 R_9^8 R_{10}^9 \underline{r}_{11}^{10} \\
 &= \underline{r}_{B_a}^I + R_{B_a}^I \underline{r}_{B_b}^{B_a} + R_{B_a}^I R_{B_b}^{B_a} \underline{r}_{B_c}^{B_b} + \dots + R_{B_a}^I R_{B_b}^{B_a} R_{B_c}^{B_b} R_B^{B_c} R_{H_i}^B R_{U_i}^{H_i} R_{Lo_i}^{U_i} R_{S_i}^{Lo_i} R_{A_i}^{S_i} \underline{r}_{C_i}^{A_i}.
 \end{aligned} \tag{A.1}$$

A.1.2 MAMMOTH Rover Leg Relative Positions and Rotations Using DH Parameters

To illustrate the use of the DH parameters, expressions for position vectors relating coordinate frames within the MAMMOTH rover leg are formulated by applying Eq. (4.6) and values from Table 4.1.

$$\begin{aligned}
\underline{r}_{H_i}^B &= \begin{bmatrix} L_i \cos \chi_i \\ L_i \sin \chi_i \\ -L_2 \end{bmatrix}, & \underline{r}_{U_i}^{H_i} &= \begin{bmatrix} L_3 \cos q_{H_i} \\ L_3 \sin q_{H_i} \\ 0 \end{bmatrix}, & \underline{r}_{Lo_i}^{U_i} &= \begin{bmatrix} L_4 \cos q_{U_i} \\ L_4 \sin q_{U_i} \\ 0 \end{bmatrix}, \\
\underline{r}_{S_i}^{Lo_i} &= \begin{bmatrix} L_5 \cos q_{U_i} \\ -L_5 \sin q_{U_i} \\ 0 \end{bmatrix}, & \underline{r}_{A_i}^{S_i} &= \begin{bmatrix} 0 \\ 0 \\ L_6 \end{bmatrix}, & \underline{r}_{C_i}^{A_i} &= \begin{bmatrix} -L_7 \cos(\delta_i - 90^\circ) \\ -L_7 \sin(\delta_i - 90^\circ) \\ 0 \end{bmatrix}.
\end{aligned} \tag{A.2}$$

Similarly, Eq. (4.7) is applied to formulate the rotation matrices from Eq. (A.1):

$$\begin{aligned}
R_{H_i}^B &= \begin{bmatrix} c\chi_i & -s\chi_i & 0 \\ s\chi_i & c\chi_i & 0 \\ 0 & 0 & 1 \end{bmatrix}, & R_{U_i}^{H_i} &= \begin{bmatrix} cq_{H_i} & 0 & -sq_{H_i} \\ sq_{H_i} & 0 & cq_{H_i} \\ 0 & -1 & 0 \end{bmatrix}, & R_{Lo_i}^{U_i} &= \begin{bmatrix} cq_{U_i} & -sq_{U_i} & 0 \\ sq_{U_i} & cq_{U_i} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \\
R_{S_i}^{Lo_i} &= \begin{bmatrix} cq_{U_i} & 0 & sq_{U_i} \\ -sq_{U_i} & 0 & cq_{U_i} \\ 0 & -1 & 0 \end{bmatrix}, & R_{A_i}^{S_i} &= \begin{bmatrix} c(q_{S_i} + 90^\circ) & 0 & s(q_{S_i} + 90^\circ) \\ s(q_{S_i} + 90^\circ) & 0 & -c(q_{S_i} + 90^\circ) \\ 0 & 1 & 0 \end{bmatrix}.
\end{aligned} \tag{A.3}$$

where c and s are short-hand for cos and sin respectively. With each of these expressions and the position vectors and rotation matrices given in the Inertial to Body Frame Transformation section of Table 4.1, it is now possible to solve for Eq. (A.1).

A.1.3 Jacobian Formulation Example

To illustrate the application of the Jacobian formulation technique from Eqs. (4.15) and (4.14), the Jacobian that is used to express the velocity of the C_i frame relative to the I frame for a single MAMMOTH rover leg is now found. Eq. (4.15) is recursively applied to find $J_{C_i}^I$:

$$\begin{aligned}
J_{C_i}^I &= J_{11}^1 \\
&= \left[\begin{bmatrix} I_{3 \times 3} & -[{}^1\underline{r}_{11}^2]^\times \\ 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix} \underline{d}_2^1 \mid \begin{bmatrix} R_2^1 & 0_{3 \times 3} \\ 0_{3 \times 3} & R_2^1 \end{bmatrix} J_{11}^2 \right] \\
&= \left[\begin{bmatrix} I_{3 \times 3} & -[{}^1\underline{r}_{11}^2]^\times \\ 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix} \underline{d}_2^1 \mid \begin{bmatrix} R_2^1 & -R_2^1 [{}^2\underline{r}_{11}^3]^\times \\ 0_{3 \times 3} & R_2^1 \end{bmatrix} \underline{d}_3^2 \mid \begin{bmatrix} R_2^1 R_3^2 & 0_{3 \times 3} \\ 0_{3 \times 3} & R_2^1 R_3^2 \end{bmatrix} J_{11}^3 \right] \\
&= \dots \\
&= \left[\begin{bmatrix} I_{3 \times 3} & -[{}^1\underline{r}_{11}^2]^\times \\ 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix} \underline{d}_2^1 \mid \begin{bmatrix} R_2^1 & -R_2^1 [{}^2\underline{r}_{11}^3]^\times \\ 0_{3 \times 3} & R_2^1 \end{bmatrix} \underline{d}_3^2 \mid \begin{bmatrix} R_2^1 R_3^2 & -R_2^1 R_3^2 [{}^3\underline{r}_{11}^4]^\times \\ 0_{3 \times 3} & R_2^1 R_3^2 \end{bmatrix} \underline{d}_4^3 \mid \dots \right. \\
&\quad \left. \begin{bmatrix} R_2^1 R_3^2 R_4^3 R_5^4 R_6^5 R_7^6 R_8^7 R_9^8 & -R_2^1 R_3^2 R_4^3 R_5^4 R_6^5 R_7^6 R_8^7 R_9^8 [{}^9\underline{r}_{11}^{10}]^\times \\ 0_{3 \times 3} & R_2^1 R_3^2 R_4^3 R_5^4 R_6^5 R_7^6 R_8^7 R_9^8 \end{bmatrix} \underline{d}_{10}^9 \right. \\
&\quad \left. \begin{bmatrix} R_2^1 R_3^2 R_4^3 R_5^4 R_6^5 R_7^6 R_8^7 R_9^8 R_{10}^9 & 0_{3 \times 3} \\ 0_{3 \times 3} & R_2^1 R_3^2 R_4^3 R_5^4 R_6^5 R_7^6 R_8^7 R_9^8 R_{10}^9 \end{bmatrix} \underline{d}_{11}^{10} \right] \\
&= \left[\begin{bmatrix} I_{3 \times 3} & -[{}^I\underline{r}_{C_i}^{B_a}]^\times \\ 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix} \underline{d}_{B_a}^I \mid \begin{bmatrix} R_{B_a}^I & -R_{B_a}^I [{}^{B_a}\underline{r}_{C_i}^{B_b}]^\times \\ 0_{3 \times 3} & R_{B_a}^I \end{bmatrix} \underline{d}_{B_b}^{B_a} \right]
\end{aligned}$$

$$\left[\begin{array}{cc} R_{B_b}^I & -R_{B_b}^I \begin{bmatrix} B_b r_{C_i}^{B_c} \end{bmatrix}^\times \\ 0_{3 \times 3} & R_{B_b}^I \end{array} \right] \underline{d}_{B_c}^{B_b} \left| \dots \right| \left[\begin{array}{cc} R_{S_i}^I & -R_{S_i}^I \begin{bmatrix} S_i r_{C_i}^{A_i} \end{bmatrix}^\times \\ 0_{3 \times 3} & R_{S_i}^I \end{array} \right] \underline{d}_{A_i}^{S_i} \left| \begin{array}{cc} R_{A_i}^I & 0_{3 \times 3} \\ 0_{3 \times 3} & R_{A_i}^I \end{array} \right] \underline{d}_{C_i}^{A_i}. \quad (\text{A.4})$$

The recursion stops given that $J_n^{n-1} = J_{11}^{10} = J_{C_i}^{A_i} = \underline{d}_{C_i}^{A_i}$ has been reached. The position vectors from Eq. (A.4) may be solved for using Eq. (4.2) and Eq. (4.4):

$$\begin{aligned} {}^9 \underline{r}_{11}^{10} &= R_{10}^9 \underline{r}_{11}^{10}, \\ {}^8 \underline{r}_{11}^9 &= R_9^8 (\underline{r}_{10}^9 + {}^9 \underline{r}_{11}^{10}), \\ &\dots \\ {}^2 \underline{r}_{11}^3 &= R_3^2 (\underline{r}_4^3 + {}^3 \underline{r}_{11}^4), \\ {}^1 \underline{r}_{11}^2 &= R_2^1 (\underline{r}_3^2 + {}^2 \underline{r}_{11}^3). \end{aligned} \quad (\text{A.5})$$

The \underline{d} vectors from Eq. (A.4) may also be solved for using Eq. (4.17):

$$\begin{aligned} \underline{d}_6^5 &= [-L_1 \sin \chi_i \quad L_1 \cos \chi_i \quad 0 \quad 0 \quad 0 \quad 1]^T, \\ \underline{d}_7^6 &= [-L_3 \sin q_{H_i} \quad -L_3 \cos q_{H_i} \quad 0 \quad 0 \quad 0 \quad 1]^T, \\ \underline{d}_8^7 &= [-L_4 \sin q_{U_i} \quad L_4 \cos q_{U_i} \quad 0 \quad 0 \quad 0 \quad 1]^T, \\ \underline{d}_9^8 &= [L_5 \sin q_{U_i} \quad L_5 \cos q_{U_i} \quad 0 \quad 0 \quad 0 \quad 1]^T, \\ \underline{d}_{10}^9 &= [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1]^T, \\ \underline{d}_{11}^{10} &= [L_7 \sin(\delta_i - 90^\circ) \quad -L_7 \cos(\delta_i - 90^\circ) \quad 0 \quad 0 \quad 0 \quad 1]^T. \end{aligned} \quad (\text{A.6})$$

The 6×10 $J_{C_i}^I$ can now be expressed using kinematic constants available in Table 4.1. The Jacobian may be reduce to the body Jacobian P and the articulated Jacobian $Q_{C_i}^I$:

$$\begin{bmatrix} \underline{v}_{C_i}^I \\ \underline{\omega}_{C_i}^I \end{bmatrix} = J_{C_i}^I \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{\chi}_i \\ \dot{q}_{H_i} \\ \dot{q}_{U_i} \\ \dot{q}_{Lo_i} \\ \dot{q}_{A_i} \\ \dot{q}_{S_i} \\ \dot{q}_{\delta_i} \end{bmatrix} = P \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + Q_{C_i}^I \begin{bmatrix} \dot{q}_{H_i} \\ \dot{q}_{U_i} \\ \dot{q}_{S_i} \\ \dot{q}_{\delta_i} \end{bmatrix} = P \underline{\dot{p}} + Q_{C_i}^I \underline{\dot{\gamma}}, \quad (\text{A.7})$$

where $\dot{\chi}_i = 0$, $\dot{q}_{A_i} = 0$ and $\dot{q}_{U_i} = -\dot{q}_{Lo_i}$.

A.2 Body Rotation Rates

Rotation rates of the robot body can be represented in two separate ways. The Euler angle rates $(\dot{\phi}, \dot{\theta}, \dot{\psi})$ may be used, or the angular rate vector $\omega_B^I = [\omega_x \ \omega_y \ \omega_z]$ may also be used. The expression for Ξ is the angular velocity transformation from I to B , and is a sub-matrix of J_B^I . Specifically, Ξ is the bottom quarter of J_B^I . The recursive expression for Jacobian formulation from Eq. (4.15) may be used along with adjacent frame constants from Table 4.1 to generate Ξ :

$$\begin{aligned}
 \Xi_B^I &= \Xi_B^{B_b} = \Xi_5^2 \\
 &= [I_{3 \times 3} \underline{d}_3^2(4:6) \mid R_3^2 \Xi_5^3] \\
 &= [I_{3 \times 3} \underline{d}_3^2(4:6) \mid R_3^2 \underline{d}_4^3(4:6) \mid R_3^2 R_4^3 I_{3 \times 3} \underline{d}_5^4(4:6)] \\
 &= \left[I_{3 \times 3} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \mid \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \mid \begin{bmatrix} c\theta & 0 & s\theta \\ -s\phi s\theta & c\phi & -s\phi c\theta \\ c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right] \\
 &= \begin{bmatrix} 1 & 0 & s\theta \\ 0 & c\phi & -s\phi c\theta \\ 0 & s\phi & c\phi c\theta \end{bmatrix}. \tag{A.8}
 \end{aligned}$$

A.3 Centre of Mass Calculation

The position of the MAMMOTH rover's centre of mass, r_{\bullet}^I , may be calculated using the weighted sum of all of the position vectors that describe the centre of mass positions of each of the rover's structural bodies. The weights of this sum are the masses of each body. The body position vectors are approximated as:

$$r_{body, \bullet}^I = r_B^I \tag{A.9}$$

$$r_{thigh, i, \bullet}^I = r_{U_i}^I + R_{Lo_i}^I \begin{bmatrix} \frac{d}{2} \\ 0 \\ 0 \end{bmatrix} \tag{A.10}$$

$$r_{ankle, i, \bullet}^I = r_{S_i}^I + R_{S_i}^I \begin{bmatrix} 0 \\ 0 \\ \frac{(L_6 - L_7)}{2} \end{bmatrix} \tag{A.11}$$

$$r_{wheel, i, \bullet}^I = r_{A_i}^I. \tag{A.12}$$

These position vectors are arranged into the position matrix Υ :

$$\Upsilon = [r_{body, \bullet}^I \ r_{thigh1, \bullet}^I \ r_{ankle1, \bullet}^I \ r_{wheel1, \bullet}^I \ \dots \ r_{thigh4, \bullet}^I \ r_{ankle4, \bullet}^I \ r_{wheel4, \bullet}^I] \tag{A.13}$$

The structural component masses are approximated as $m_{body} = 15$ kg, $m_{thigh, i} = 10$ kg, $m_{ankle, i} = 2.5$ kg and $m_{wheel, i} = 2.5$ kg. The mass vector, \underline{m} , is composed of the masses of each body:

$$\underline{m} = [m_{body} \ m_{thigh1} \ m_{ankle1} \ m_{wheel1} \ \dots \ m_{thigh4} \ m_{ankle4} \ m_{wheel4}]. \tag{A.14}$$

The position of the centre of mass may be calculated as:

$$\mathbf{r}_{\bullet}^I = \frac{\sum_{j=1}^N \underline{m}(j) \Upsilon(:, j)}{\sum_{j=1}^N \underline{m}(j)}. \quad (\text{A.15})$$

A.4 Euler Angles

The Euler Angle representation used in this thesis follows the standard ‘XYZ’ or roll-pitch-yaw rotation convention. These rotation angles are also commonly referred to as the Cardan or Tait-Bryan angles [28]. This roll, pitch and yaw typically refers to the transformation from the inertial to the body frame as summarised in Table 4.1. In this thesis, the transformation from the body frame to the inertial frame is typically used and is an expression for this transformation is derived here. This is done by first rotating about the z_{B_a} axis by ψ , then rotating about the y_{B_d} axis by θ and finally rotating about the x_{B_c} axis by ϕ . There are no remaining rotations, just translation from frame B_a to I . The rotation matrix that describes this complete rotation is given in the following expression where c and s are used for sin and cos for brevity:

$$\begin{aligned} R_B^I &= R_{B_a}^I R_{B_b}^{B_a} R_{B_c}^{B_b} R_B^{B_c} = I_{3 \times 3} R_{B_b}^{B_a} R_{B_c}^{B_b} R_B^{B_c} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix} \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} c(\psi)c(\theta) & -c(\theta)s(\psi) & s(\theta) \\ c(\phi)s(\psi) + c(\psi)s(\phi)s(\theta) & c(\phi)c(\psi) - s(\phi)s(\psi)s(\theta) & -c(\theta)s(\phi) \\ s(\phi)s(\psi) - c(\phi)c(\psi)s(\theta) & c(\psi)s(\phi) + c(\phi)s(\psi)s(\theta) & c(\phi)c(\theta) \end{bmatrix} \end{aligned} \quad (\text{A.16})$$

A singularity will occur using this convention when the pitch angle θ is 90° or -90° . The ‘XYZ’ convention has been selected given that it is highly unlikely that a ground-based rover will ever reach this singularity pitch angle during regular operations.

A.5 Thigh Joint Kinematics

The kinematic relationship between the upper thigh joint and the stroke length of the linear actuator must be considered in order to control the thigh joint. As shown in Figures A.1, the leg is a parallelogram with the linear actuator attached to freely rotating joints near diagonal corners of the parallelogram.

The relationship between leg i ’s linear actuator stroke length, q_{Th_i} , and the upper thigh joint angle, q_{U_i} , is derived using the law of cosines:

$$\begin{aligned} a^2 + b^2 - 2ab \cos(q_{U_i} + 90^\circ) &= (c + q_{Th_i})^2 \\ q_{U_i} &= -\arccos\left(\frac{(q_{Th_{i,z}} + c)^2 - a^2 - b^2}{2ab}\right) - 90^\circ \end{aligned} \quad (\text{A.17})$$

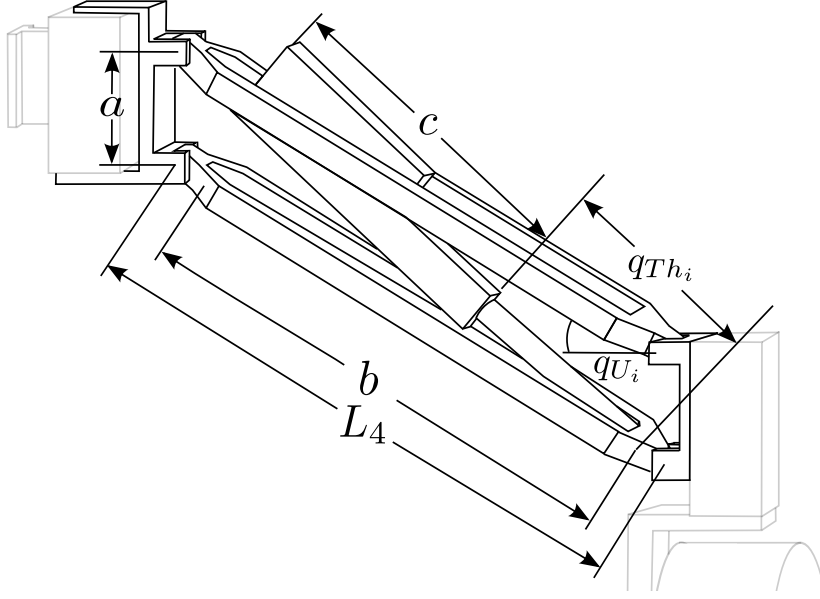


Figure A.1 – A labelled view of the MAMMOTH rover leg i thigh.

where a , c and b are shown in Figure A.1 and values for these thigh kinematic constants are provided in Table 4.1. The expression for q_{Th_i} as a function of q_{U_i} may be found by re-arranging Eq. (A.17):

$$q_{Th_i} = \sqrt{a^2 + b^2 - 2ab \cos(q_{U_i} + 90^\circ)} - c, \quad (\text{A.18})$$

The velocity of the thigh joint may be found by taking the derivative of the q_{Th_i} with respect to time:

$$\dot{q}_{Th_i} = \dot{q}_{U_i} \frac{2ab \sin(q_{U_i} + 90^\circ)}{\sqrt{a^2 + b^2 - 2ab \cos(q_{U_i} + 90^\circ)}}, \quad (\text{A.19})$$

which may be re-arranged to find the upper thigh velocity:

$$\dot{q}_{U_i} = \dot{q}_{Th_i} \frac{\sqrt{(a^2 + b^2 - 2ab \cos(q_{U_i} + 90^\circ))}}{2ab \sin(q_{U_i} + 90^\circ) \dot{q}_{U_i}}. \quad (\text{A.20})$$

To localize the MAMMOTH rover and drive to specific body orientations, it may be necessary to solve for the thigh configuration that corresponds to the rover pose, hip configuration and terrain profile.

The expression for q_{U_i} is formulated as:

$$q_{U_i} = \kappa_i + \text{asin} \left[\left(\frac{r_{U_i,z}^I}{\cos \varepsilon_i} + L_5 \sin \kappa_i - (L_6 + L_7) \cos \kappa_i \right) / L_4 \right] \quad (\text{A.21})$$

where κ_i is the angle between the x_{U_i} -axis and a plane parallel to the $x_I y_I$ -plane. Similarly,

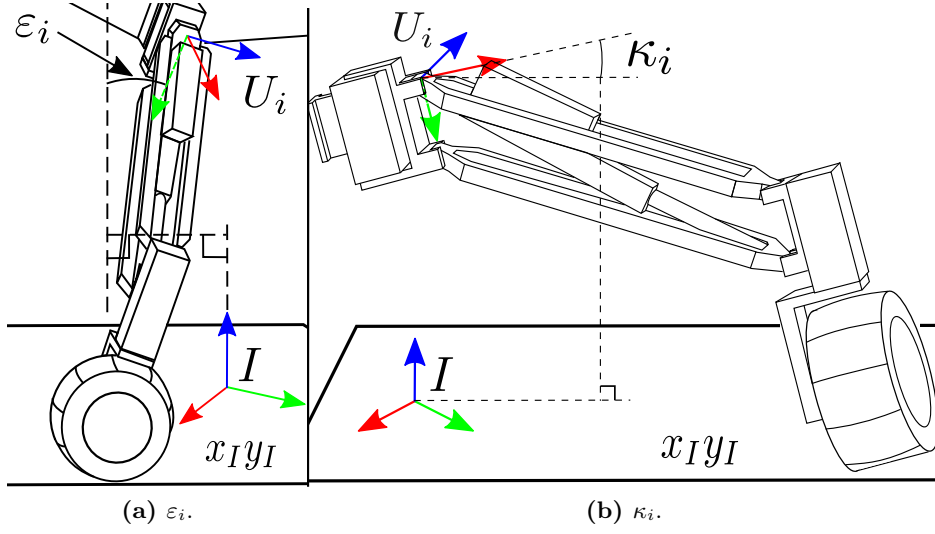


Figure A.2 – Views of the MAMMOTH leg showing the location of the κ_i and the ε_i used in the inverse kinematic calculations to find q_{U_i} .

ε_i is the angle between the z_{U_i} -axis and a plane parallel to the $x_I y_I$ -plane. These angles are both shown in Figure A.2 and are functions of the rotation matrix $R_{U_i}^I$:

$$\varepsilon_i = \text{asin}(\underline{n} R_{U_i}^I \underline{m}^T), \kappa_i = \text{asin}(\underline{n} R_{U_i}^I \underline{n}^T), \quad (\text{A.22})$$

where $\underline{n} = [0 \ 0 \ 1]$ and $\underline{m} = [1 \ 0 \ 0]$. By solving for $r_{U_i,z}^I$ and $R_{U_i}^I$ using Eqs. (4.2) and (4.3) and substituting these expressions into Eq. (A.21) and (A.22), we may develop an expression for q_{U_i} that is a function of the body pose \underline{p} , the hip joint position q_{H_i} and the terrain height at the contact point $r_{C_i,z}^I$. This analytical result has been omitted for brevity.