Western Graduate&PostdoctoralStudies

Western University

Scholarship@Western

Electronic Thesis and Dissertation Repository

12-14-2012 12:00 AM

System of Terrain Analysis, Energy Estimation and Path Planning for Planetary Exploration by Robot Teams

David C. Michel
The University of Western Ontario

Supervisor
Kenneth McIsaac
The University of Western Ontario

Graduate Program in Electrical and Computer Engineering
A thesis submitted in partial fulfillment of the requirements for the degree in Doctor of Philosophy
© David C. Michel 2012

Follow this and additional works at: https://ir.lib.uwo.ca/etd

Part of the Robotics Commons

Recommended Citation

Michel, David C., "System of Terrain Analysis, Energy Estimation and Path Planning for Planetary Exploration by Robot Teams" (2012). *Electronic Thesis and Dissertation Repository*. 1078.
https://ir.lib.uwo.ca/etd/1078

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact wlswadmin@uwo.ca.

System of Terrain Analysis, Energy Estimation and Path Planning for Planetary Exploration by Robot Teams

(Spine title: Terrain Analysis, Energy Estimation and Path Planning System)

(Thesis format: Monograph)

by

David Michel

Graduate Program in Electrical & Computer Engineering

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

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

© David Michel 2012

THE UNIVERSITY OF WESTERN ONTARIO
SCHOOL OF GRADUATE AND POSTDOCTORAL STUDIES

CERTIFICATE OF EXAMINATION

Supervisor	Examiners
Dr. Kenneth McIsaac	Dr. Lyndon Brown
Supervisory Committee	Dr. Roy Eagleson
Dr. Ilia Polushin	Dr. Samuel Asokanthan
Dr. Jin Jiang	Dr. Piotr Jasiobedzki

The thesis by

David Michel

Entitled:

System of Terrain Analysis, Energy Estimation and Path Planning for Planetary Exploration by Robot Teams

is accepted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Date December 14th 2012

Dr. Peter Simpson
Chair of the Thesis Examination Board

Abstract

NASA's long term plans involve a return to manned moon missions, and eventually sending humans to mars. The focus of this project is the use of autonomous mobile robotics to enhance these endeavors. This research details the creation of a system of terrain classification, energy of traversal estimation and low cost path planning for teams of inexpensive and potentially expendable robots.

The first stage of this project was the creation of a model which estimates the energy requirements of the traversal of varying terrain types for a six wheel rocker-bogie rover. The wheel/soil interaction model uses Shibly's modified Bekker equations and incorporates a new simplified rocker-bogie model for estimating wheel loads. In all but a single trial the relative energy requirements for each soil type were correctly predicted by the model.

A path planner for complete coverage intended to minimize energy consumption was designed and tested. It accepts as input terrain maps detailing the energy consumption required to move to each adjacent location. Exploration is performed via a cost function which determines the robot's next move. This system was successfully tested for multiple robots by means of a shared exploration map. At peak efficiency, the energy consumed by our path planner was only 56% that used by the best case back and forth coverage pattern.

After performing a sensitivity analysis of Shibly's equations to determine which soil parameters most affected energy consumption, a neural network terrain classifier was designed and tested. The terrain classifier defines all traversable terrain as one of three soil types and then assigns an assumed set of soil parameters. The classifier performed well over all, but had some difficulty distinguishing large rocks from sand.

This work presents a system which successfully classifies terrain imagery into one of three soil types, assesses the energy requirements of terrain traversal for these soil types and plans efficient paths of complete coverage for the imaged area. While there are further efforts that can be made in all areas, the work achieves its stated goals.

Keywords:

Path-Planning, Energy Estimation, Soil Identification, Planetary Exploration

Table of Contents

Certificate of Examination ii

Abstract iii

List of Tables vi

List of Figures vii

Acknowledgements ix

1.0 Introduction 1

1.1 Motivation 1

1.2 Thesis Statement 2

1.3 Scope 2

1.4 Contributions 3

2.0 Background 4

2.1 Similar Work 4

2.2 Wheel/Soil and Rover Modeling 5

2.3 Complete Coverage by a Single Robot 6

2.4 General Exploration Approaches 11

2.5 Path and Coverage Planning for Multiple Robots 12

2.6 Terrain Representations 17

2.7 Traversability and Image-Based Terrain Type Classification 20

2.8 Non Traversability-Based Rough Terrain Traversal 25

2.9 Localization 28

3.0 Approach 30

3.1 Energy Requirements of Terrain Traversal 30

3.2 Assessment of Terrain 31

3.3 Path Planning Algorithm 32

3.4 System Architecture 32

4.0 Energy Consumption Model 34

4.1 Rocker-Bogie Model 34

4.2 Wheel/Soil Interaction Model 40

4.3 Testing Configuration 45

4.4 Results 47

5.0 Path Planner 53

5.1 Terrain Representation 53

5.2 Accessibility 53

5.3 Movement Cost 55
5.4 Calculation of D 57
5.5 Results 57
5.5.1 Completely Accessible Area 57
5.5.2 Map with Null Values 59
5.5.3 Multiple Rovers on Completely Connected Map 61
5.5.4 Multiple Rovers on Map with Null Values 63
6.0 Terrain Classification 67
6.1 Sensitivity Analysis 67
6.2 Terrain Classifier 73
6.2.1 Pre-processing 73
6.2.2 Wavelet Transform 74
6.2.3 Feature Extraction 75
6.2.4 Classification 75
6.3 Results 77
7.0 Conclusion 81
7.1 Future Work 81
7.2 Conclusions 81
Bibliography 83
Curriculum Vitae 95

List of Tables

Table 4.1 43
Table 4.2 43
Table 4.3 48
Table 5.1 59
Table 5.2 60
Table 5.3 63
Table 5.4 63
Table 5.5 64
Table 6.1 73
Table 6.2 76
Table 6.3 77

List of Figures

Figure 3.1 33
Figure 4.1 34
Figure 4.2 35
Figure 4.3 36
Figure 4.4 37
Figure 4.5 38
Figure 4.6 40
Figure 4.7 41
Figure 4.8 45
Figure 4.9 46
Figure 4.10 46
Figure 4.11 47
Figure 4.12 48
Figure 4.13 49
Figure 4.14 50
Figure 4.15 50
Figure 4.16 51
Figure 4.17 51
Figure 4.18 52
Figure 5.1 53
Figure 5.2 54
Figure 5.3 54
Figure 5.4 56
Figure 5.5 57
Figure 5.6 58
Figure 5.7 59
Figure 5.8 60
Figure 5.9 60
Figure 5.10 61
Figure 5.11 61
Figure 5.12 61
Figure 5.13 61
Figure 5.14 62
Figure 5.15 62
Figure 5.16 64
Figure 5.17 64
Figure 5.18 65
Figure 5.19 65
Figure 5.20 65
Figure 5.21 66
Figure 6.1 68
Figure 6.2 68
Figure 6.3 68
Figure 6.4 69

Figure 6.5 69
Figure 6.6 69
Figure 6.7 70
Figure 6.8 70
Figure 6.9 70
Figure 6.10 71
Figure 6.11 71
Figure 6.12 71
Figure 6.13 72
Figure 6.14 74
Figure 6.15 76
Figure 6.16 76
Figure 6.17 77
Figure 6.18 78
Figure 6.19 79

Acknowledgements

There are a great many people to whom I owe thanks for help on this project. Firstly, I would like to thank my advisor, Dr. Kenneth McIsaac, for giving me the opportunity to choose my own course of study, and for having the wisdom and patience to allow me to learn from my mistakes. I am also grateful for his gracious hospitality during the final testing process.

Additionally, I am indebted to Chris Vandelaar of University Machine Services, for the tremendous amount of work he contributed to the design and construction of the rocker-bogie testing platform. I would also like to thank my colleagues Bryan Godbolt and Duane Jacques for their advice and sober second opinions.

Thanks also to my good friends Andrew Waters and Daniel Karpenchuk for their continued friendship and loyalty.

Finally, I would like to thank my mother Jane, my brother Alexander and my sister Claire for their continual love, support and sacrifice. Without them, I know I would not be where I am today.

David Michel
November 2012
London, Ontario, Canada

1.0 Introduction

The era of autonomous planetary exploration by mobile robots was begun with the Sojourner rover's successful contribution to the Mars Pathfinder mission in 1997. Beginning in 2004, the Mars Exploration Rovers (MER) Spirit and Opportunity displayed greatly enhanced exploration capabilities, returned orders of magnitude more science data and are still in operation today. NASA has just finished work on its third generation of rover, the Mars Science Laboratory (MSL), which weighs 900 kg, and is approximately the same size as a small automobile. The MSL successfully landed on Mars in August 2012 and is currently operational. The work proposed herein is not intended to extend or replace the cutting edge and costly research and development undertaken by NASA in this area. Rather, the purpose of the project is to take a different approach to the problem. It is our hope that the system of terrain analysis, energy estimation and path planning for teams of less complex and potentially expendable rovers presented herein will allow for future use of robots for low level exploration tasks to be done more cost effectively.

1.1 Motivation

Large rovers are expensive to build, operate and to send to other planetary bodies. As NASA's long term plans involve a return to manned moon missions, and eventually sending humans to mars, the focus of this project is the use of autonomous mobile robotics to enhance these missions.

The creation of detailed terrain maps, location of mineral and water resources and low level geological surveying will all be useful to future manned missions to other planetary bodies. Important applications of this data include locating potential bases and colonies near valuable resources and a foreknowledge of the most interesting sites for intensive research. While this information is useful, its collection is a relatively mundane process which if conducted by humans would detract from the time available for more in depth study. The goal of this work is to allow teams of simple robots to perform these low level surveying tasks either in advance of human arrival, or concurrently with a manned mission, allowing astronauts more time to focus on research activities which would be impossible to automate.

Research in this area has already been conducted by Fong et al (discussed in detail in the literature review), who have made a strong case for the potential contribution of a system which would employ teams of rovers for detailed terrain mapping, prospecting, and opportunistic science. While promising, their work makes such limited use of autonomy that it fails to fill the described need; hand generated coverage plans and large human ground crews run contrary to the stated design goals. If a team of astronauts is to be emancipated from low level surveying, and freed to spend time on more important problems, the solution must be truly autonomous.

1.2 Thesis Statement

The purpose of this research is to create systems of terrain analysis, energy estimation and low cost path planning for teams of inexpensive and potentially expendable robots performing low level science and/or terrain assessments of wide areas of the Lunar or Martian surface.

1.3 Scope

The design of planetary rovers is a well-established and costly field. As such, a new physical design for a planetary rover would not only be redundant, but divert valuable resources from the true focus of this project. Consequently, the scope of this work is largely algorithmic. The major areas of research were as follows:

- Wheel/soil interaction as described by Bekker theory, and the modeling of rocker-bogie type rovers
- Coverage algorithms for single robots in conjunction with area partitioning
- Complete coverage algorithms for multiple robots
- Autonomous terrain assessment and terrain classifiers based on visual and topographic data

1.4 Contributions

This work dealt with a number of challenging problems and sought to tackle them with novel and untested approaches. The three main contributions of this work are as follows:

- The creation of a new model of the most common rover suspension (rocker-bogie) and an accompanying wheel/soil interaction model which combined allow for the estimation of relative energy costs of terrain traversal for different soil types.
- The creation of a new path planning algorithm which, given a cost map for terrain traversal, allows one or more robots to completely cover the map area whilst minimizing energy consumption.
- A sensitivity analysis of Shibly's modified Bekker equations.
- A system for terrain classification and estimation of Bekker properties based on colour images.

2.0 Background

This project drew from a number of different areas of study in the fields of mobile robotics and image processing. As such, a fairly broad range of work has been surveyed for this chapter.

2.1 Similar Work

Planetary resource prospecting and mapping by teams of autonomous robots is a relatively new idea. The most relevant research in this field has been conducted by Terrence Fong and his team at JPL. First proposed in 2006 and then implemented in a series of tests in 2007, Fong's system used two of NASA's K10 test rovers to conduct collaborative LiDAR and GPR mapping at Haughton Crater [1] [2] [3] [4] [5] [6]. Haughton Crater is a lunar analog site in the Canadian high arctic, and is frequently used for tests of this nature. The system used a full colour 60 cm/pixel resolution satellite map for a-priori traversability analysis and path planning, which was registered to UTM (Universal Transverse Mercator) by hand to allow for the precise use of differential GPS. Traversability analysis was implemented using the Morphin algorithm (discussed in more detail in the traversability section). While the generation of coverage plans was initially sought to be achieved autonomously using a path transform (discussed in more detail in the complete coverage section), coverage plans for the LiDAR scans ended up being created by hand, while GPR coverage plans employed Boustrophedon (back and forth) type paths. The proposal for the system also called for autonomously generated partial coverage plans in the event of time restrictions, but this was not implemented in the version tested at Haughton. Software control of both rovers was implemented using PLEXIL (PLan EXecution Interchange Language), a custom designed language for abstracting rover hardware and executing activity plans [7].

The rover responsible for LiDAR imaging was active for 9 days, generated 25 panoramas and traversed a total of 14 km. The rover carrying the GPR was active for 10 days, and traversed 32.3 km.

This system and these series of experiments had a heavy emphasis on human supervision; coverage plans were manually generated, and there was an extensive ground crew used to coordinate and monitor rover progress. There was also no real cooperation between rovers, as they were conducting different sets of experiments, and their coverage plans were independently generated.

Another system, called MISUS (Multi-rover Integrated Science Understanding System) was proposed by Estlin et al in 2005 [8]. MISUS attempts to achieve maximum science return from multiple rovers by rating science goals and conducting observations of high value targets. The system requires reasonably high computing overhead (P4 3.06 GHz, 1 GB RAM), and still does not function in anything approaching real-time. It also has yet to be implemented in the real world, and has only been tested in simulation.

2.2 Wheel/Soil and Rover Modeling

The modeling of wheeled vehicles on rough terrain has its origins in the field of terramechanics. The pressure-sinkage relationship for a wheeled vehicle is well established, and was developed by Bekker in the 1960's. Bekker was responsible for many early developments in the study of terramechanics, including equations for drawbar pull and various forms of soil resistance to motion [9]. Shibly *et al.* later developed linearizations for a number of Bekker's equations which were tested over a constrained range of soil parameters [10]. Shibly’s equations are drawn on heavily in this work.

Bekker theory has been used to model vehicle interactions with rough terrain on a number of prior occasions. Ben Amar and Bidaud created a simulation tool which combined terramechanics with vehicle parameters to determine whether or not the vehicle could traverse rough terrain in a safe configuration [11]. Grand *et al.* used Bekker theory to design a control system which optimized wheel torques to provide maximum tractive effort given varying soil conditions [12]. Patel *et al.* developed a rover analysis tool called RMPET (Rover Mobility Performance Evaluation Tool [13]. RMPET used Bekker theory to determine the sink, slip, drawbar pull and a number of other parameters of various rover designs on different soils. Bauer *et al.* developed a rover prototyping tool called RCAST [14], which was mainly concerned with determining optimum rover design parameters for performance on rough terrain. Ding *et al.* used Bekker theory to develop a tool similar to RMPET which determined the forces acting on the wheels of a specific 4-wheel rover [15].

The rocker-bogie rover design has been NASA's standard for the wheeled mobile exploration of Mars, and was used on Sojourner and the Mars Exploration Rovers [16]. Detailed kinematic modeling of this suspension system was first performed by Hacot [17], who developed a series of closed form equations to determine rover configuration based on joint angles and connection lengths. Hacot also developed equations for static force balances for the rocker-bogie design. These equations were later used by Hacot *et al.* to develop a simulation tool which attempted to predict the normal force on each wheel based on rover configuration and applied torque [18]. Gang and Yi took a different approach to modeling the rocker-bogie design [19]. They used a Denavit-Hartenberg based representation of the rover to develop transformation matrices to and from relevant coordinate frames, along with forward and inverse kinematic equations.

2.3 Complete Coverage by a Single Robot

The study of complete coverage of an area by a single robot is a field that has been thoroughly researched. Early work was done by Zelinsky et al with their development of the distance transform [20]. This technique worked by defining start and end points for the robot’s traverse, dividing the workspace into cells of equal size to that of the robot and assigning each cell a value based on its distance from the end point. The robot would cover the area by moving to the unexplored cells with the highest values, resulting in complete coverage once the end point cell had been explored. In the same paper, they also presented the concept of the path transform. The path transform works much the same way as the distance transform, except that there is additional cost associated with each cell depending on its proximity to obstacles. Calculating cell values in this way creates wall-following inward spiraling paths of complete coverage. These paths require less turning, and are therefore more efficient. More recently, Wirth and Pellenz have tried to improve on these methods with the concept of the exploration transform [21]. While intended more for path planning than exploration, it allows for the creation of "safer" or "faster" routes to goal points depending on the user's preference. This is done by adjusting the degree to which the proximity of obstacles affects the values of cells.

Pioneering work was also done by Choset [22] with his development of the Boustrophedon Cellular Decomposition. The area to be covered was divided into cells which were covered by a back and forth motion. Cells were opened and closed by the discovery of what Choset termed "critical points", which were found by sweeping lines from positive to negative infinity in the vertical or horizontal direction. Locations at which an obstacle was encountered were deemed to be critical points. The advantage of critical points is that they allow the area to be easily mapped in adjacency graph, and make global path planning a simple process. The disadvantage of this technique is that it will only work if the obstacles are convex. Garcia and de Santos [23] were able to improve this technique to deal with non-convex obstacles, and the corresponding mismatches in IN/OUT points they were found to generate. Acar et al also incorporated work on critical points into the field of autonomous land mine detection [24]. Knowledge of distribution patterns was incorporated to expedite the searching process. Once enough mines had been detected to determine a pattern, the Boustrophedon style search was aborted in favour of traversal to the next most likely mine location. Acar and Choset then further improved upon their technique for cellular decomposition based on critical points by designing an implementation that did not require pre-existing knowledge of the area to be covered [25].

A number of minor improvements have been developed to augment the Boustrophedon Cellular Decomposition. Huang [26] was able to improve performance by adjusting the inclination of the line sweeps based on cell dimensions to minimize the distance traveled. Yao [27] also attempted improvements along similar lines by adjusting the angle of the sweeps to accommodate for desired entry and exit points to each cell. This allowed for a substantial reduction in repeated coverage when transitioning from cell to cell.

Kang at el devised a system of complete coverage which used a method of cell decomposition virtually identical to BCD [28]. They also incorporated 12 movement templates to facilitate the most efficient exploration of each cell, and eliminate repeat coverage when transitioning between cells. A cost function was developed to select the most appropriate of the templates.

Acar and Choset also developed a method of complete coverage based on a hierarchical decomposition of the workspace [29]. They divided space into two categories: vast and cluttered. Vast spaces were defined as those that could not be completely covered by the robot's sensor range, while cluttered spaces were enclosures which the robot could completely cover with a single sweep. Vast spaces were covered as seen previously with BCD. Cluttered spaces were covered with paths created from Generalized Voronoi Diagrams (GVD). GVDs create paths which are always equidistant from two or more obstacles. They will be discussed in more detail in the path planning section.

Ge and Fua envisioned a system which treated explored space as a "spurious obstacle" [30]. It featured two exploration modes: normal and wrap. The normal exploration mode worked in the Boustrophedon (back and forth) manner. When no more free space was detected, the wrap mode was initiated, tracing the edge of the current spurious obstacle until unexplored space was detected. The robot would alternate between normal and wrap modes until there was no more unexplored space.

Pirzadeh and Snyder [31] developed a system which divided the terrain to be covered into cells equal in area the exploring robot's base. Cells were treated as four-connected, and cell traversal costs were initiated to zero (this assumes a flat indoor type environment). The algorithm worked by augmenting the current cell's cost by α, assigning a cost β to any direction which features an uninterrupted sequence of explored cells followed by an obstacle, assigning further costs δ and γ to adjacent cells for meeting criteria defining their re-traversal as unnecessary and then moving to the lowest cost adjacent cell. The costs were experimentally determined to be most effective when $\gamma>\delta>\beta>\alpha$ by a factor of 3.

Koenig et al presented work on the concept of greedy mapping [32]. Greedy mapping is a simple procedure, which always moves the exploring robot to the nearest unexplored area. While they acknowledged that this technique would never be optimal, they showed that the upper bound on completion time in graph like worlds was far from unacceptable.

Schmidt and Hofner [33] proposed a system which used a simple back and forth coverage plan for complete coverage by an automated cleaning robot. Robot coverage was defined by a polygon. As the robot moved, polygons were fused to create a record of the covered area. After initial traversal, a follow up coverage plan was generated to traverse any areas which had been missed by the initial sweep. This system was designed for indoor environments, and is incapable of taking into account different costs for terrain traversal.

De Carvalho et al [34] did related work in the area of autonomous cleaning robots. Their system used a rectangular-grid based representation of the environment, and required an a-priori map of the area to be covered. It was however able to deal with unforeseen obstacles by using wall-following to bypass them. A series of coverage templates were used to traverse the area as efficiently as possible.

Oh et al [35] extended the work from the previous paper. They used a triangular-cell based representation of the environment which was 12-connected. The additional complexity of their mapping scheme allowed for more complex movements and better coverage plans. The height of each triangular cell was the same as the diameter of the robot. This technique also eliminated the need for a pre-existing map by first using wall-following to define the boundary of the area to be covered. As with the original work, a series of coverage templates where then used to traverse the interior as efficiently as possible.

Lang and Bing-Yung [36] attempted to improve upon existing cleaning coverage algorithms by using fuzzy logic to vary speed and turning rate based on the robot’s proximity to obstacles. Wall following was used initially to define the boundaries of the workspace. During this initial phase, a series of horizontal tracks of equal width to the robot were created and later used as a path for complete coverage. In the event of unexpected obstacles, the robot reverts to wall-following and updates its coverage plan.

Zhang et al worked on further improving cleaning robots by attempting to have them localize themselves based on landmarks gleaned from limited sensor information [37]. Using only ultrasonic proximity sensors, they attempted to distinguish between various types of concave and convex obstacles, and to have the robots localize themselves on a pre-existing map using these landmarks.

Deng and Papadimitriou [38] worked on complete exploration by representing the environment in a graph form as a series of nodes connected by edges. They found that many graphs could be explored efficiently by traversing unknown edges until none remained, and then backtracking and repeating this process recursively. Unfortunately, there is a class of non-Eulerian or “deficient” graphs which cannot be explored in a reasonable time using this method. Albers and Henzinger [39] tackled this problem a decade later, and were able to produce the first exploration algorithm for this class of graph which took less than exponential time.

Dudek et al did work treating robotic exploration as graph construction [40]. Their work dealt with robots incapable of self-localization. They proved that under such a scenario, exploration was impossible unless the robot was capable of leaving and recovering markers at graph nodes.

Gabriely and Rimon attempted to plan paths of complete coverage using a construct called a spanning tree [41]. The algorithm works by dividing the workspace into cells of size 2D x 2D, where D is the diameter of the robot. Cells that are partially covered by obstacles are eliminated from consideration. A graph structure is then defined with nodes at the centre of each cell, and edges connecting the nodes of adjacent cells. Given a starting cell S, a path is then created connecting every node in the workspace. The robot is then instructed to circumnavigate this path in a counter-clockwise direction until it returns to its starting point. This paper also featured an on-line spanning tree creation algorithm which required no a-priori knowledge of the workspace, but substantially more computing power to implement.

Gonzalez at al created what they called the “backtracking spiral algorithm” [42]. It functioned by using wall following inward spiraling exploration patterns. Once the robot had reached the centre of its current spiral, it backtracked to the nearest free space and started a new one.

Jiao and Tang developed an exploration algorithm relying on visibility based area partitioning [43]. Traversable areas were labeled C_1, C_2…C_N, while obstacles were labeled O_1, O_2…O_N. Boundaries between visible areas that were not obstacles were termed gates and labeled G_1, G_2…G_N. A stack of all unvisited gates, GU, was created, and exploration continued until GU was empty.

Lee et al worked on using Peano curves to generate paths of complete coverage for two-dimensional spaces [44]. Their technique used self-organizing feature maps to iteratively generate a path of complete coverage over simple spaces. This process was responsive to changes in sensor range, generating shorter paths for robots with greater observational capabilities.

2.4 General Exploration Approaches

There have also been more general approaches to exploration which although not specifically designed for complete coverage, can be used to that end. The concept of the frontier was introduced by Yamauchi, and is fundamental to many algorithms [45]. Yamauchi's system divided maps into an evidence grid, where each cell contained the probability of its occupation by an obstacle. Cells were classified as open, unknown or occupied, and were updated whenever new sensor data was received. Any open cell adjacent to an unknown one was deemed a frontier edge cell, and adjacent frontier cells were considered frontier regions. Any frontier region over a threshold size was deemed a frontier, and the exploring robot always took the shortest path to the nearest frontier.

Makarenko et al used the frontier method to identify regions of interest for exploration [46]. However, instead of simply selecting the closest frontier, they used entropy-derived equations to attempt to calculate the expected information gain at each frontier. Frontiers were selected for exploration based on these calculations, their distance from the robot and the projected ability of the robot to localize itself in each area. Moorehead et al took a similar approach to exploration [47]. They sought to plan exploratory paths by rating the expected information gain for each cell from multiple sources of information. The total expected information gain was calculated as the weighted sum of the factors under consideration, and a greedy algorithm was used to select from adjacent cells to move to.

Liu at al presented a more complicated approach to information gain based exploration [48]. They used a 3-D triangular mesh map representation of the environment, and correlated information gain to visibility. Ray tracing from the robot's camera height was used to determine the area visible from each cell, and this information was combined with the calculated energy required for traversal to select the next cell to be explored. This energy estimate was not terramechanics-based; it assumed equivalent soil parameters and was simply proportional to changes elevation. Dijkstra's graph search algorithm was used for planning paths between cells [49].

Oriolo et al devised a structure called the Sensor-based Random Tree (SRT) for exploration [50]. The SRT is essentially a series of random walks. Each point in the structure contains a node, and a surrounding Local Safe Region; an estimation of the free space surrounding that point. The tree is extended by picking a direction and a distance randomly such that the distance is beyond a preset minimum, and the destination point is not within the LSR of another node. If the maximum number of iterations is exceeded and a viable destination point has not been found to extend the tree, the robot backtracks to the previous node and again attempts to extend the tree. Freda and Oriolo improved on this method the following year by biasing point selection towards frontier areas [51]. Espinoza et al also made a minor improvement to the SRT algorithm by modifying the manner in which the LSR was determined to more accurately account for the placement of obstacles [52].

2.5 Path and Coverage Planning for Multiple Robots

Path and coverage planning for multiple robots has been the focus of a great deal of study. Methods generally fall into one of two categories: some manner of centralized control involving area partitioning which allows each robot to explore its own section of the workspace, or a less centralized approach requiring substantial communication between robots to update their knowledge of explored space. Both methodologies have been covered in the following section.

Early research in this field was done by Singh and Fujimura [53]. Their research focused on exploration using teams of heterogeneous robots. Exploration would initially be conducted individually by team members. When a robot encountered an area it could not access, that area was assigned to a smaller robot. This process would continue until all the areas reachable by the smallest team member had been explored.

Vincent and Soille developed a fast algorithm to segment greyscale elevation maps based on the concept of watersheds [54]. By analyzing local minima and their surroundings, they were able to separate mapped areas into drainage basins using simulated immersion.

Hert and Lumelsky did early work on the centralized decomposition of a workspace into n polygons [55]. Their algorithm divided a polygon P into n separate polygons, all of a specified area with a specified point on their boundaries. This algorithm was only shown to be efficient when the polygon was convex and contained no holes. Bast and Hert were able to extend this work and create an algorithm which, although not optimal, was able to divide any arbitrary polygon [56]. Jager and Nebel devised a much simpler method of decomposing an area into polygons [57]. They simply overlaid a grid on the entire workspace, which led to a number of interconnected square regions, with polygonal areas bordering obstacles. Each robot had a stack of areas to explore which were assigned by attempting to minimize the diameter of the total assigned space. Schneider-Fontan and Mataric devised a system which divided the workspace into equally sized areas for each robot [58]. In the event of robot failure or disparities in progress, their algorithm would resize the areas to prevent any unit from becoming idle.

Solanas and Garcia used a K-Means clustering algorithm to evenly divide unexplored space between robots [59]. Their algorithm dynamically repartitioned unexplored space as new areas were discovered, and was shown to be more efficient than greedy mapping or techniques making assumptions about the utility of frontiers.

Wurm et al worked on segmentation of indoor environments based on typical interior layouts [60]. Their method used the understanding of building designs to assign individual robots completely explore rooms.

Yamauchi implemented his concept of exploration of frontiers on multiple robots by having team members share information [61]. This distributed system had each robot maintain its own global map and make its own decisions. As new information was obtained, it was shared with the group. However, since there was no coordination in the decision making progress, it was possible for multiple robots to choose to explore the same frontier. Similarly, Parker et al planned exploration paths for each robot independently, and simply varied their velocity profiles to avoid collisions [62].

Burgard et al created a system whereby robots were assigned frontiers based on the cost of reaching the location and the "utility" of that point [63][64][65]. Utility was calculated in a probabilistic manner based on the range of the robots sensors, and the proximity of obstacles. A greater expected information gain would constitute a greater utility. Conversely, Poemomo and Ying believed that any cost function based on expected information gain was inherently flawed and highly inaccurate [66]. They introduced a cost function for multi-robot exploration based on greedy mapping and the desire to maintain as great a distance as possible between each unit.

Latimer et al extended BCD to multiple robots [67]. They used teams of robots traveling in formation to expedite cell exploration. In the event that different teams encountered each other, they would merge and continue the exploration of that cell together.

Kong et al presented a multi-robot system similar to BCD [68]. However, instead of creating cells based on critical points, the workspace was initially divided into cells of a width twice that of the exploring robots. If obstacles were encountered which partitioned a cell, that cell was decomposed into two cells, one on either side of the obstacle. Any time a robot completed exploration of a cell, the global map would be updated, and it would move to the nearest unexplored area.

Hazon et al were able to extend the concept of spanning trees to multiple robots [69]. Their implementation featured the on-line creation of spanning trees by each robot. Information was coordinated such that each robot had knowledge of the connection points between their respective spanning trees. In the event one or more robots failed, their spanning trees would be incorporated into that of a still functional robot. This system guarantees robustness so long as a single robot is still operating. Agmon, Hazon and Kaminka also worked to improve the efficiency of the off-line creation of spanning trees for multiple robots [70]. Their approach created a single spanning tree for the entire workspace, with the exploring robots as close to evenly spaced as possible along the tree. This allowed for a substantial savings in exploration time if all the robots remained functional. Hazon and Kaminka were able to further improve the robustness of multi-robot spanning trees by introducing an algorithm which allowed backtracking in the event of failures [71]. Previous multi-robot spanning trees only allowed traversal in a single direction, the ability to traverse the tree in both directions allowed for a substantial improvement in worst-case completion time in the event of failures.

Franchi et al developed an implementation of the SRT method for multiple robots [72]. This was done by initially having each robot create its own SRT. When a robot reaches a point where it can no longer expand its own tree, it moves to a support role and helps expand the tree of the nearest team member. A feasibility check was used before the execution of team motions to prevent collisions. Sanchez at al simultaneously created a similar multi-robot SRT implementation [73]. Their system also had robot create its own tree then attempt to expand the trees of other robots once it had finished its own. They also analyzed trajectories for feasibility to prevent collisions between explorers.

One of the more popular techniques for assigning target points in multi-robot exploration problems is based on a market economy. Simmons et al developed a framework in which individual robots would construct bids for frontier locations based on expected information gain and travel costs [74]. A central executive received the bids and assigned tasks by attempting to maximize information gain and minimize overlap. Zlot et al developed a system whereby robots would create their own goal points, and attempt to maximize profit by trading goals with other robots [75]. Robots would bid on the list of goals generated by those within communications range, and a central executive would assign payment based on the user’s priorities. Gerkey and Mataric created an auction-based system called MURDOCH using a publish/subscribe communication architecture [76]. Robots only subscribed (receive messages) to tasks they were capable of completing. Bids consisted of a robot’s evaluation of the cost of these tasks, with the lowest cost always being selected. Zlot and Stenz added levels of abstraction to the market-based approach by introducing the concept of task trees [77]. Tasks were broken down hierarchically using AND/OR relationships. Robots were then permitted to bid on a high level task, or components of that task.

Rekleitis et al developed an auction-based system with the primary goal of eliminating idle time for robots [78]. Their methodology segmented the workspace into a single evenly sized slice for each robot to explore. In the event that part of the slice was unreachable, the unexplored portion would be auctioned off to robots capable of accessing it.

Another market based approach was developed by Kalra et al [79]. Their architecture, call Hoplites, was designed to deal with the problem of constrained exploration. It allowed for robots to maintain line of sight, and hence radio communication, during the entire exploration process, by imposing a severe financial penalty on goals which violated predefined constraints.

Fu et al introduced a path planning scheme which used four primitive behaviours to coordinate multiple robots [80]. A fuzzy controller chose between move to goal, avoid static obstacle, avoid dynamic obstacle and avoid robot behaviours. This system seems more useful for dynamic environment path planning than planetary exploration.

Mendez-Polanco and Munoz-Melendez created an exploration team composed of three robots and a central server [81]. The server would receive mapping information from the team members, decompose it to a topological map and assign robots to terminal vertices. This system was designed and implemented for indoor exploration.

Howard et al produced an incremental deployment algorithm for robots exploring an unknown area [82]. Robots were deployed one at a time, and intended to maintain line of sight while maximizing the area surveyed. The algorithm also allowed robots to switch roles in the event of irresolvable interference.

Sujan et al devised a team of heterogeneous robots, specifically intended for the exploration of cliff faces in extra-terrestrial environments [83]. “Reconbots” would scout cliff edges, while “anchorbots” would affix themselves to the precipice and lower a “cliffbot” on tethers to map the sheer edge.

Chibin et al made one of the more novel attempts at multi-robot coverage [84]. While they used Boustrophedon Cellular Decomposition to partition the workspace into cells. Exploration of each cell was undertaken using what they called an “ant colony algorithm”. This algorithm required the exploring robots to deposit pheromones to mark explored territory. The pheromones would repel other explorers and force them towards unexplored territory.

Thayer and Singh devised the Immunology-derived Distributed Autonomous Robotics Architecture (IDARA) for controlling large numbers (hundreds) of simple robots [85]. Modeled on the human immune system, IDARA was shown to be effective at controlling up to 1500 robots exploring unstructured environments.

Hougen et al developed and implemented a heterogeneous multi-robot exploration architecture based on two different robot types: “scouts” and “rangers” [86]. Their implementation used a small number of larger more sophisticated ranger robots to control and distribute a large number of smaller scouts which possessed sensor capabilities but were incapable of decision making.

2.6 Terrain Representations

The fundamental basis of any exploration strategy is a representation of the environment being examined. There is a wide array of methods available for terrain modeling, and they generally fall into three classes: metric, topological and hybrid. Metric maps usually involve segmentation of the environment into some kind of grid. They provide accurate position information, but become difficult to store for large areas. Topological maps are more abstract, and tend to represent relationships between landmarks. They do not provide accurate position information, but have low memory overhead, even for very large areas. Hybrid maps are an attempt to combine metric and topological; they often use topological maps for global relationships between metric maps of more interesting areas. Techniques of all three varieties will be discussed in the following section.

Bakambu et al devised a method of terrain modeling for the Canadian Space Agency (CSA) which used 2.5 D point data recovered from panoramic LiDAR scans [87][88]. Their technique stitched together numerous LiDAR scans, and then decimated the point data into an Irregular Triangular Mesh (ITM). This had the benefit of substantially reducing the required storage space by simply representing flat areas with a few triangles, and retaining detail in more uneven terrains. The ITM construct also allowed for simple path planning along triangle edges using Dijkstra’s graph search. Further work on this system was done by Rekleitis et al, who were able to eliminate as much as 93% of the collected point data and still create acceptable ITMs [89].

Hahnel et al attempted to create simplified 3D models from laser range finder scans fitting planes to surfaces with a low enough variance to meet their definition of flat [90]. They were able to reduce indoor environment models by a factor of approximately

25, but the process was time consuming (over an hour in one case), and did not work as well for outdoor environments.

Ye and Borenstein created 3D terrain maps using a 2D laser rangefinder [91]. They accomplished this by mounting the rangefinder on the front of their mobile robot at an inclination of -11°. As the robot moved forward, the scanner would cover terrain in a push broom fashion, and elevation maps would be built with assistance of their custom designed Certainty Assisted Spatial (CAS) filter.

NASA developed a stereo-vision based 3D environment rendering system for the Mars Exploration Rovers (MER) called the Ames Stereo Pipeline [92]. This software package used two cameras a fixed distance apart, and knowledge of their focal lengths to create 3D triangular mesh models of wedges of terrain extending away from the rover. These wedges were then stitched together to create complete models of the surrounding Martian environment. The models created varied in resolution depending on proximity to the rover, with the most detailed information being available at short distances [93]. Se and Jasiobedzki developed another stereo-vision based 3D modeling system for MacDonald Dettwiler and Associates (MDA) called instant Scene Modeler (iSM) [94]. It is has the additional capability of recovering motion between frames (visual odometry). Unfortunately, a single rendering takes approximately 5 minutes on a high end computing platform (P4 2.4 GHz).

The baseline distance (length between the focal points of the two cameras in a stereo pair) has been limited by mass and volume restrictions for planetary imaging. Short baseline distances have made accurate imaging of distant features an impossibility. Olson et al developed a wide baseline stereo imaging system which worked by extracting 3D information from rover imagery taken at two different positions [95]. They also worked on integrating this data with descent and satellite imagery to create a more comprehensive environment model.

Choset and Burdick made an important contribution to mapping with their development of the Generalized Voronoi Graph (GVG) [96]. The GVG can be constructed from sensor data; all that is required are distance measurements to obstacles. It is essentially a roadmap which is equidistant from any number of detected obstacles. The GVG can be used in three dimensions, and because of its structure the GVG reduces motion planning to a one dimensional graph search. In a companion paper, Choset and Burdick detailed the procedure for incremental construction of a GVG from sensor data [97]. Choset et al later developed an improved method for incremental construction of a GVG which would ensure a complete roadmap of the area under exploration [98]. Kalra et al later introduced an efficient algorithm for the dynamic reconstruction of GVGs as new information became available [99].

Castejon et al were able to implement GVGs for path planning on a real robot [100]. Their system used data from a laser range finder, and first assessed terrain for traversability based on slope and roughness. Once the terrain had been segmented, GVGs were used to plan paths in the traversable region. The algorithm was able to function in real-time, and was implemented on a 1.5 ton robot in an outdoor environment. Cheong et al improved upon the GVG by introducing the concept of the concave node [101]. Concave nodes were defined as GVG branching points in which both branches led to a dead end. This distinction informed an exploration strategy based on exploring dead ends first, and was able to noticeably increase exploration efficiency.

Murrieta-Cid et al created maps of outdoor scenes using landmarks [102]. Landmarks were defined as easily identifiable peaks in terrain surrounded by relatively flat land. These maps were compact, and designed to assist in mobile robot localization.

Simhon and Dudek created a hybrid mapping system which relied on local metric maps as "islands of reliability" [103]. The locations for the islands were chosen by an equation presented by the authors to calculate distinctiveness using sonar sensor inputs. Thrun and Bucken took a different approach to the hybrid map creation problem [104][105]. Instead of mapping different regions in either a metric or topological manner, they created complete metric and topological maps and overlayed them. The topological map was created by splitting the metric representation into what were termed "coherent regions". Coherent regions were defined as being separated by "critical lines" which were essentially narrow passages between wider open spaces (i.e. doorways). Tomatis et al created hybrid maps of indoor environments which modeled halls topologically and rooms metrically [106].

Guivant et al developed another hybrid mapping scheme [107]. They used distinctive features in the environment to define Local Triangular Regions (LTR). A landmark was located at each vertex of every LTR, and a local coordinate frame was defined for each region. The system functions well in three dimensions, and was designed to facilitate robot localization.

Lisien et al developed a hybrid form of environment map combining the GVG and landmark based mapping [108]. Their mapping system was called the Hierarchical Atlas and used a GVG to represent the entire workspace. In different regions of the GVG, maps of detected edges were used as landmarks to assist with localization.

2.7 Traversability and Image-Based Terrain Type Classification

Traversability is a key concept when considering paths of complete coverage on outdoor terrain. The physical design of the rover imposes limitations on which terrain types it is capable of crossing. Additionally, some terrains will present riskier and more energy intensive paths which should be minimized. Terrain classification has been extensively investigated. However, a large number of the efforts have been of a proprioceptive nature, requiring vibration, wheel torque and sinkage information from robots as they traverse terrain to determine soil properties. As we require this information a-priori, these methods do not present an acceptable solution. This section focuses on exteroceptive approaches.

Early work on image processing to identify ridges and valleys was done by Gauch and Pizer [109]. They used sharp drop-offs and increases in pixel intensity along with watersheds to accurately identify ridges and valleys in fingerprint and other biomedical related imagery. Their watershed method was later extended by Liu et al in an attempt to create an obstacle detection system for a lunar lander [110]. This method incorporated models of craters, slopes and block ejecta in order isolate obstacle regions unsuitable for landing. Traversability was not of any concern in this effort.

George et al attempted to segment images into traversable and obstacle regions by processing greyscale images [111]. This was done by analyzing a single picture for pixel luminance; obstacles tend to have a sharp contrast in intensity with flat ground. Additionally, 3D terrain reconstruction from a stereo camera pair was used to classify obstacles by identifying protrusions from the ground. This technique was found to be effective for detecting large obstacles in close proximity to the rover. It did not compare favorably with human segmentation of terrain.

Moorehead et al developed an algorithm for navigation and obstacle avoidance called Morphin, and successfully implemented it on the Nomad test rover in automated meteorite search conducted in Antarctica [112]. Morphin combined laser rangefinder and stereo imagery to create “goodness” maps of the surrounding terrain, based on perceived slope and roughness. Potential trajectories were then evaluated based on these values, and the path with the greatest goodness value was selected.

Seraji introduced and did early work on the traversability index [113]. This approach used a stereo vision system to assess terrain slope and roughness. Terrain slope was fuzzified as {LOW, MEDIUM, HIGH, VERY HIGH} by fitting a plane to each grid cell. Roughness was defined as {SMOOTH, ROUGH, BUMPY, ROCKY} by summing the deviations from the fitted plane in each cell and calculating the frequency of rocks. Predefined relationships between these two outputs were then used to classify terrain traversability as a member of the fuzzy set {POOR, LOW, MEDIUM, HIGH}. This index was used to govern rates of speed and turning rate to ensure safe rover navigation. Seraji and Bon augmented this work by producing a larger rule set using the traversability index to govern navigation behaviours such as goal seeking and obstacle avoidance [114]. Seraji and Howard worked to implement a scaled down version of this index on a mobile robot in a Mars-like test environment [115] [116]. In these experiments the fuzzy set defining slope was reduced to {FLAT, SLOPED, STEEP}, roughness was calculated only by size and distribution of rocks and defined as {SMOOTH, ROUGH, ROCKY}, and traversability was classified by the fuzzy set {LOW, MEDIUM, HIGH}. The paths generated in these tests were found to be traversable.

Additional testing of this system was performed by Seraji, Howard and Tunstel [117]. They used additional fuzzy rules to govern the degree to which the traversability index affected path planning for goal seeking operations. They assigned the traversability index a greater emphasis at large distances, and decreased this influence as the target was approached, as goal seeking behaviour was found to be substantially more effective at close range.

Yet another series of tests was later conducted by Seraji and Howard [118]. These trials confirmed the ability of the system to detect and avoid hazardous discontinuities. Also, a more exhaustive test of goal seeking behavior was performed with the system achieving an 80% success rate under 10 different test scenarios. The primary source of failure was found to be error in the dead reckoning calculations used to determine position.

A later implementation of this system by Seraji retained the reduced fuzzy sets for slope, roughness and traversability [119]. However, it added an assessment of hardness ({SOFT, MEDIUM, HARD}) to the calculation of the traversability index. No decisions were made as to the manner in which hardness would be calculated, but both force sensors and image processing techniques were proposed. Seraji and Howard subsequently incorporated hardness information by using a neural network for classification [120]. The neural network analyzed image texture, and was trained using images of gravel, sand and compacted soil. This system was found to be susceptible to changes in lighting conditions.

Further work by Seraji, Howard and Tunstel introduced the concept of discontinuity to the traversability index [121]. This was done to allow for the detection of sudden changes in elevation that could not be detected by the calculation of slope. The presence of ridges, valleys and ravines could result in catastrophic failure of the rover if undetected. Discontinuity was defined by the fuzzy set {SMALL, LARGE}, with LARGE discontinuity indicating impassable terrain.

Howard et al attempted to improve on the traversability index by using optimization techniques to incorporate human assessments of terrain [122]. A database of human expert classifications of 17 image pairs was used. One image in each pair was defined as having HIGH traversability, while the other was labeled LOW. Using these image sets to optimize output yielded an improvement in the agreement between human assessment and the final calculation of traversability index.

Ye and Borenstein developed their own traversability index [123]. Their system used the slope and roughness of robot-sized patches of terrain to assign a numerical value indicating the difficulty associated with traversal. These values were then used to create a Traversability Field Histogram. This histogram assigned obstacle densities to each 5 degrees of arc surrounding the robot. Navigation was performed by moving the robot in the direction of the lowest obstacle densities.

Gennery devised a technique similar to the traversability index which instead relied on laser rangefinder data to make terrain assessments [124]. Slope and roughness were calculated in a similar manner, and combined using a cost function. This cost function was used to create a grid-based representation of the terrain, with each cell containing a numerical value corresponding to its difficulty of traversal. Path planning was accomplished by selecting the route to a target point with the lowest total cost.

The Mars Exploration Rovers (MER) used a system called Grid-based Estimation of Surface Traversability Applied to Local Terrain (GESTALT) [125]. GESTALT modeled its environment as grid composed of patches equal in size to that of a rover wheel. Each cell contained two 8-bit values: goodness and certainty. Goodness was a measure of traversability determined by the presence of sudden changes in elevation and roughness estimates, while certainty was a measure of the reliability of these estimates.

Castejon et al used laser rangefinder data to create what they Traversable Region Models (TRM) [126]. Their approach analyzed the slope and roughness of 3D maps to isolate traversable areas. They then overlayed this information onto a visibility map (essentially a DEM that makes worst case assumptions about occluded areas) to obtain a binary traveraability map. A GVG was then created on this binary image for path planning purposes. Hata et al [127] used laser rangefinder data in combination with a neural network classifier to reduce 3D terrain data into a 2D navigation map which defined terrain as traversable, partly traversable or intraversable.

Manduchi et al attempted to isolate obstacles based on their reflectance properties when illuminated by light in the near infrared spectrum [128]. This research was intended primarily for unmanned military vehicles, and was able to use Laser reflectance data to isolate regions of green vegetation, non-green vegetation and soil. Jansen et al used colour information in a Gaussian Mixture Model (GMM) based approach to segment outdoor images into regions of sky, foliage, grass, sand and gravel [129]. Sung et al [130] used a wavelet transform to extract feature information and location-based weighting to similarly classify terrain, this method was the basis for our terrain classifier.

In another terrestrial-centered approach, Wolf et al used a Hidden Markov Model (HMM) to isolate flat stretches of road in outdoor environments [131].

Angelova et al created a texton-based series of five complementary terrain classifiers [132]. This approach used texture information to extract a 75 dimensional representative vector for a number of different terrain types. There was however a marked loss of efficiency at medium to long range, and the maximum effective distance of the classifiers was approximately 15 m. This classifier was later used to associate learned slip (measured whilst traversing previously identified terrain patches) with each of these terrain types [133].

Karlsen and Witus analyzed single photos of terrain to create binary images which consisted of “Go” and “NoGo” regions [134]. This data was then used with an offline clustering algorithm to create a set of exemplars; essentially image chips which typified certain terrains and were deemed to be either traversable or intraversable. Photos taken when the robot was in use were then segmented by classifying sections as one exemplar or another to binarize the image into Go and NoGo Regions. Karlsen and Witus later expanded on this approach by associating training images with Vehicle Terrain Interaction (VTI) parameters [135]. This method produced more complicated assessments of potential traversability. Using VTI data also had the added benefit of allowing the attachment of maximum velocities and turning rates for each exemplar. Testing the following year using 325 training images, and 24x24 pixel exemplars showed this system could estimate ground resistance with less than 10% error [136].

Iagnemma et al made early attemps to define traversability of different terrain types based on available towing force relative to wheel torque and sinkage [137]. Nine terrain classes were defined based on experimentally observed values for this relationship and used to define traversability as “good”, “medium” or “poor”. Later work by Brooks and Iagnemma described the traversability of terrain using a metric called the coefficient of traction [138]. Defined as the available towing force divided by the wheel load, it is a measure of how much a robot can pull relative to its own weight. Bekker equations were used along with experimentally obtained maximum and minimum soil parameter values in an attempt to establish upper and lower bounds on the coefficient of traction for different terrain types. Five different terrain classes were then created based on this estimated lower bound, and a Support Vector Machine (SVM) based approach was used in an attempt to visually separate terrain into each of these classes. The visual classification technique was similar to one Brooks and Iagnemma had used previously while trying to identify a-priori terrain which was substantively different from that contained within their robot's knowledge base [139]. This was done in order to avoid traversing potentially hazardous unfamiliar terrain. Further SVM-based visual terrain classification was attempted by Brooks and Iagnemma in an attempt to distinguish between rock, sand and beachgrass [140]. This approach yielded mixed results, as there were issues with robustness and colour classification.

Shirkhodaie, Amrani and Tunstel analyzed terrain photos and classified sub-regions as traversable or nontranversable [141]. Using calculations for image energy, contrast, variance and rock blob area, they implemented heuristic, neural network and fuzzy logic based terrain classifiers. The neural network and fuzzy logic based classifiers were highly successful (93% and 98% respectively) at classifying terrain into 1 of 9 different categories ranging from "very rocky" to "very sandy". Shirkhodaie et al later combined this fuzzy logic terrain classifier with a path planner, which searched images row by row for the most suitable and safest waypoints [142].

Kubota et al worked on using greyscale images of the lunar surface to assign a "degree of danger" to each pixel [143]. This approach determined risk by centering a window on each pixel and using the maximum brightness and pixel variance within that window to ascertain the degree of danger.

2.8 Non Traversability-Based Rough Terrain Traversal

Methods for traversing rough terrain that do not explicitly use the concept of traversability have also been studied in detail. Chatila et al worked to develop a laser rangefinder based path planning scheme for the French space agency (CNES) [144]. This technique had two different approaches depending on whether terrain had been classified as "flat" or "uneven". Flat terrain paths were acquired by binarizing the map into obstacles and free space, and generating a voronoi graph. The voronoi graph was then searched for a path to the goal. Uneven terrain was handled in a similar manner, except it was necessary to keep track of the system configuration (ground clearance, roll and pitch) at each point along potential paths to ensure that dangerous paths were vetoed. Lacroix et al improved this system by adding heuristic measures to improve path selection [145]. This was done by favorably weighting paths which were known to contain more reliable terrain information as well as those which would facilitate easier localization via feature mapping.

Path planning by estimation of vehicle terrain interactions received additional contributions from Cherif and Laugier [146]. They used a detailed kinematic description of their test rover to determine all safe configurations in the workspace. An A* search was then used to find the optimal path along these safe configurations to the target point.

Bonnafous, Lacroix and Simeon did similar work on path selection by estimation of robot configuration along potential paths [147]. However, they used their model to evaluate a number of potential arc paths and calculate their risk. The “interest” of a path was calculated by the distance between its end point and the robot’s target. The arc with the lowest risk/interest ratio was then selected.

Tunstel, Howard and Seraji, while not particularly concerned with path planning, developed a scheme to moderate rover speed over rough terrain based on the concept of safety [148]. They used measurements of pitch and roll combined with analysis of terrain roughness to implement a set of fuzzy rules which would increase or decrease speed depending on the perceived rover safety at any moment in time.

Valavanis et al implemented a matlab based fuzzy controller for rough terrain navigation [149]. Using filtered laser range finder data to estimate the distance to the nearest obstacle in each of three sectors (left, centre and right) comprising the forward 180° view of the robot, as well as heading error to the goal point, a fuzzy rule set was developed that output translational and rotational velocity.

Important early work was done by Stentz with his development of the D* algorithm [150]. D* can be used to plan optimal paths in partially known and changing environments, and is the basis of many modern autonomous navigation systems. D* works by considering each point in the navigation space to be a state. Each state contains an estimate of the lowest cost (distance in this case) route to the currently defined goal state. State data is continually modified by sensor information, and thus allows for replanning in the event of inaccuracies in initial information or changing conditions in the environment.

Yahja et al later implemented the D* algorithm using a framed quadtree environment map [151]. Instead of subdividing regions into grid cells of equal size, quadtrees map an environment by considering obstacle free regions as single cells. This is done by successively subdividing the area into four quadrants. The subdivision is continued until a quadrant contains no obstacles, or the smallest grid sized is reached. A framed quadtree surrounds each obstacle free area with cells of the smallest size to facilitate smoother navigation. This system was found to have memory and path length advantages over an ordinary grid representation, but performance suffered in highly cluttered environments. Yahja et al further extended this system to incorporate terrain costs [152]. By subdividing regions into areas of like cost and providing this information to the path planner, the system was able to generate more efficient traverses. It was found that this worked best with course information, since it was less of a strain on computing resources.

Singh et al developed an architecture which combined the global scheme using D* and framed quadtrees developed by Yahja et al with a local path planner [153]. The global and local modules provided either a vote between 0 and 1, or a veto for each prospective path. An arbiter module would then decide the final trajectory based on a predefined weighting of the two planners. Local path planning was done using the Morphin algorithm.

Saab and VanPutte performed rough terrain path planning by separating terrain into low and high cost regions [154]. This was done by defining minimum and maximum altitudes for low cost terrains and comparing these values to a DEM of the workspace. Any points above the maximum or below the minimum were considered high cost and bounded by polygons. A modified version of Dijkstra’s graph search was then used to find the shortest path to the goal point that did not intersect a high cost polygon.

Yenilmez and Temeltas developed a cost function for rough terrain path planning based on energy requirements [155]. Using assumptions regarding surface interaction between robot and terrain, they developed a differential equation for estimating the relative energy costs of traveling between any two adjacent points on a DEM. Their work was limited to adjacent points, and did not cover the creation of complete paths.

Dupuis et al developed a path planning scheme for the Canadian Space Agency based on their method of terrain modeling using Irregular Triangular Meshes [156]. By converting the ITM terrain model into an undirected weighted graph with edge weights determined by distance and slope, Dijkstra's shortest path search algorithm was then used to find optimal routes and waypoints. Rekleitis et al did further work on this system by adding a roughness measure to the calculation of edge weights [157]. In addition, they performed a number of semi-autonomous traverses (the operator selected the destination point), each in excess of 100 m. Rekleitis et al again added to this methodology by turning Dijkstra's undirected graph search into a directed A* search [158]. This substantially reduced the computing burden of the path planner by ensuring that the search was not grown outward in all directions, but only towards the target point.

Massari et al used a simulated annealing optimization approach to generate Bezier curve paths to a goal point on a DEM [159]. The generated paths were over short distances (5m x 5m simulated environment), but were shown to be safe and near optimal.

2.9 Localization

A great deal of time and effort has been devoted to developing localization techniques for mobile robots. However, this work is beyond the scope of the current project. Consequently, this section will detail the techniques most recently implemented on actual space missions (the Mars Exploration Rovers), as well as an emerging technology which not only has the potential to make existing localization methodologies obsolete, but which holds tremendous promise for the future of planetary exploration.

The Mars Exploration Rovers used a combination of techniques for position estimation. Initial localization of the Spirit rover was achieved eight days after landing (before the rover left the landing platform) using a combination of two-way Doppler radio positioning, descent and rover imagery as well as reconstruction of the entry profile [160]. After departure from their landing sights, the lack of a magnetic field on mars required that attitude be obtained by using a camera to locate the position of the sun in the sky [161]. Accelerometers, gyro readings and wheel odometry were used to update position information. Visual odometry was also used in high slip areas, but its use was limited because of the high computing cost.

On earth, centimeter level position information can be obtained using Carrier-phase Differential GPS (Global Positioning System). Since other planetary bodies in the solar system lack an array of orbiting satellites dedicated solely to localization and navigation, this would seem to be an unacceptable option. However, work on Self-Calibrating Pseudolite Arrays (SCPA) will make this a realistic option in the future. A pseudolite is considered to be any device which transmits GPS satellite-like signals, and was initially conceived of to augment existing GPS signals in obstructed work environments (ie deep open pit mines) [162]. LeMaster and Rock used an array of these devices, along with a mobile transceiver, to “self-survey” areas and create the first SCPA [163]. Self-calibration is a multi-step process. First pseudolites exchange signals to determine their relative ranges, these ranges are then combined to determine array geometry. An initial guess of carrier-cycle ambiguities is then made. Finally, motion of a mobile transceiver (such as a rover) is used to refine this estimate and survey the pseudolite locations with centimeter level accuracy.

Field tests were first conducted in 2000 using three stationary transceivers and one carried by a human subject to simulate the motion of a mobile robot over an area approximately 100m by 100m [164]. Using presurveyed pseudolite locations along with an equal number of known points in the workspace, CDGPS readings had an RMS error of 0.76 cm. LeMaster, Matsuoka and Rock conducted further field tests in 2003 using three stationary pseudolites and NASA’s K9 test rover as the platform for the mobile transceiver [165]. Operating over a smaller area (15m by 20m), but without any knowledge of the relative positions of the pseudolites, these tests yielded an RMS position error of 4.2 cm. Matsuoka, Rock and Bualat saw the obvious benefits of SCPA localization for planetary rovers. In 2004 they tested a scheme for the autonomous deployment of an SCPA using a single test rover in NASA’s Marscape (simulated Martian terrain) [166] [167]. In these tests, three pseudolites were “deployed” (placed by hand behind the rover) in a roughly triangular pattern. After calibration the RMS position error was found to be 9cm.

3.0 Approach

This chapter will provide a short introduction to the overall system design and the stages in which it was created. The project was conceived to be modular in nature, and had a number of clearly identifiable milestones along the path to completion.

3.1 Energy Requirements of Terrain Traversal

To date, Mars rovers have been almost completely reliant on the solar panel/rechargeable battery combination as a source power. The recently deployed MSL relies on radioisotope thermal decay (RTG) as a power source, but is still limited to 125 W. Given that Mars' average distance from the sun is approximately 1.5 times that of the earth and the limitations in amount of space available for solar panels and the generating capacity RTGs, power is clearly any rover's most critical resource. In fact, the Mars Exploration Rovers were only in operation for a small portion of each day, and were required to shut down all systems except for heaters every night to conserve power. Consequently, it seems obvious that the most optimal path planner for any Martian rover system would be one which was capable of finding the lowest energy paths.

The literature review surveyed a great deal of work done in the area of traversability and the creation of goodness maps, as well as a number of other methods for determining the ease of rover motion over varying terrains. Saab and VanPutte attempted to classify the costs of terrain traversal, however, these techniques were qualitative in nature [154]. Karlsen and Witus attempted to define maximum vehicle velocities and turning rates based on terrain type, but made no attempt to determine energy expenditures [134][135][136]. The only quantitative attempt made to analyze the real-world energy costs of terrain traversal was by Yenilmez and Temeltas, and their system could only estimate energy costs between adjacent map points [155]. To date, the best attempt at classifying terrain by energy cost for traversal was the work done by Brook and Iagnemma [138]. By classifying terrain based on the worst case coefficient of friction (wheel load divided by available towing force), their system has a hierarchical understanding of energy costs. However, it does not provide a basis for numerical comparison of different terrain types and does not account for the non-linear manner in which changes in ground angle affect wheel/soil interaction.

The first stage of this project was the creation of a model which estimates the energy requirements of the traversal of varying terrain types for a six wheel rocker-bogie rover. The model uses Shibly's modified Bekker equations [10] and takes into account the following variables:

1. **Robot weight distribution and dimensions**: our rocker-bogie model accepts a configuration file which details key dimensions and rover mass, allowing for its use in the modeling of any rover of this type.
2. **Terrain angle**: the model also accepts a DEM (Digital Elevation Model) of the terrain to be analyzed and uses this data not only for energy estimation, but also to determine which areas are traversable and which are not.
3. **Bekker soil properties**: the model requires six soil properties (discussed in detail in Chapter 4). The model accepts these data as input for every point on the DEM.

Our model provides a detailed estimation of the energy costs of traversing different terrain types. This allows for the creation of a 2D terrain representation similar in structure to a DEM, but representing energy costs instead of elevations.

3.2 Assessment of Terrain

Knowledge of the relative energy costs of traversing varying terrain types is useless if the robots are unable to accurately determine the properties of the environment they are surveying. The literature review surveyed a number of existing methodologies in this area. These attempts largely dealt with trying to determine traversability (or degrees thereof) not classifying soil types [111][113][115][116][120][122-127][134-136][141]. Those attempts which did distinguish between soils often relied on similar image types involving a road (desired terrain) surrounded by foliage, sand and gravel [129-132]. They also made no to attempt to estimate soil parameters. The classification work done by Brooks and Iagnemma [138-140] required the robot to first traverse the different visually identified soil types to allow them to be learned. This exteroceptive terrain classification was limited to the selection of either sand, rock or grass.

Our work defines traversable vs. intraversable areas using information gleaned from the DEM input. Estimations of soil parameters are done using a neural network classifier which defines traversable terrain as one of three deferent soil types, and then assigns soil properties based on existing knowledge of Martian soil properties and a sensitivity analysis of Shibly's equations, which was used to determine the factors most relevant to energy costs.

3.3 Path Planning Algorithm

Once a model of power costs for terrain traversal, and the ability to identify said terrains were available, a computationally efficient algorithm for complete coverage by single or multiple rovers whilst minimizing energy cost was developed. A large number of terrain segmentation techniques for multi-robot exploration were surveyed in the literature review. While many of them deal with shortening path lengths, and efficient exploration of areas, none of them seek to optimize power consumption.

Our algorithm uses the information provided by the first two stages of the project and a central processor and functions in a similar manner to the distance transform. While the entire set of paths could be planned a-priori, our algorithm is robust enough to react to changes in ground information by simply moving from point to point on the map until all areas have been visited.

3.4 System Architecture

Figure 3.1 illustrates the complete system design. Colour ground imagery is fed to the terrain classifier, which provides soil parameter estimates to the energy consumption model. The energy consumption model also accepts a DEM of the area to be covered and rover specifications as input. Energy consumption estimates are then provided to the path planner, which designates course information based on the number of available rovers.

The modular nature of this project has the advantage of allowing future improvements in any one stage to be easily integrated into the system as a whole. For example, were an improved terrain classifier to be developed elsewhere, our path planning algorithm could incorporate it to achieve better results.

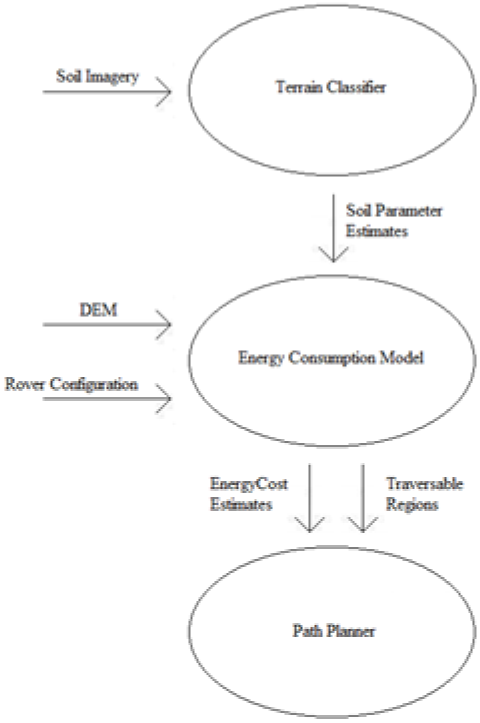


Figure 3.1: System Architecture

4.0 Energy Consumption Model

The energy consumption accepts grid-based terrain elevation data in the form of a Digital Elevation Model (DEM) along with rover and soil parameters. Our newly developed model of the most common rover suspension design (rocker-bogie) along with a terramechanics-based wheel-soil interaction model are then used to build a map of the estimated torque required by each wheel to move the rover to each adjacent terrain square. At a constant velocity there is an almost linear relationship between motor torque and energy consumption, this work therefor considers the two to be analogous.

4.1 Rocker-Bogie Model

The rocker-bogie rover features a passive suspension system with a number of significant advantages, the foremost of which is the ability to keep all six wheels in constant contact with uneven terrain. While there are conceivable scenarios where ground contact for all six wheels would not be possible, such terrain would almost certainly be intraversable. Shown in Figure 4.1, this class of rover uses a symmetrical design with six independently driven wheels. Each half of the vehicle consists of a rocker attached to a secondary rocker called a bogie. The wheel assemblies are connected to the body by a pivoting differential, which ensures that each assembly is always carrying half the rover's mass, and that the centre of mass (not including wheel assemblies) is always located halfway between the rocker-bogie/body connection points.

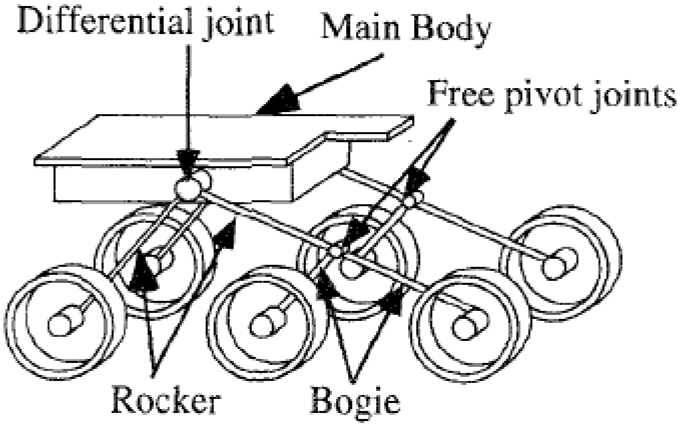


Figure 4.1: Rocker-Bogie Rover Design

Another point in favour of this design is the ease with which it can be modeled; the differential allows for each side of the His model is comprehensive; it models forces and moments in three dimensions. As our simplified wheel/soil interaction model does not include transverse forces, Hacot's approach provides unnecessary information and would add substantial computational overhead. We therefore developed a new model better suited to the energy consumption problem. Figure 4.2 illustrates our model for one side of a rocker-bogie rover. W is the mass of the rover neglecting wheels and suspension system, w_1, w_2 and w_3 are the vertical loads on each wheel and l_1, l_2, l_3 and l_4 represent fixed length suspension system linkages. ψ_1, ψ_2 and ψ_3 are angles which define the suspension system configuration; ψ_1 and ψ_3 are fixed while ψ_2 operates over a specified range. The distances between wheels along the operating angle of the rover side are represented by x_1 and x_2, while x_3 and x_4 represent the distances along the operating angle of the rover side between the rear wheel and the centre of mass and rear wheel and the bogie joint respectively.

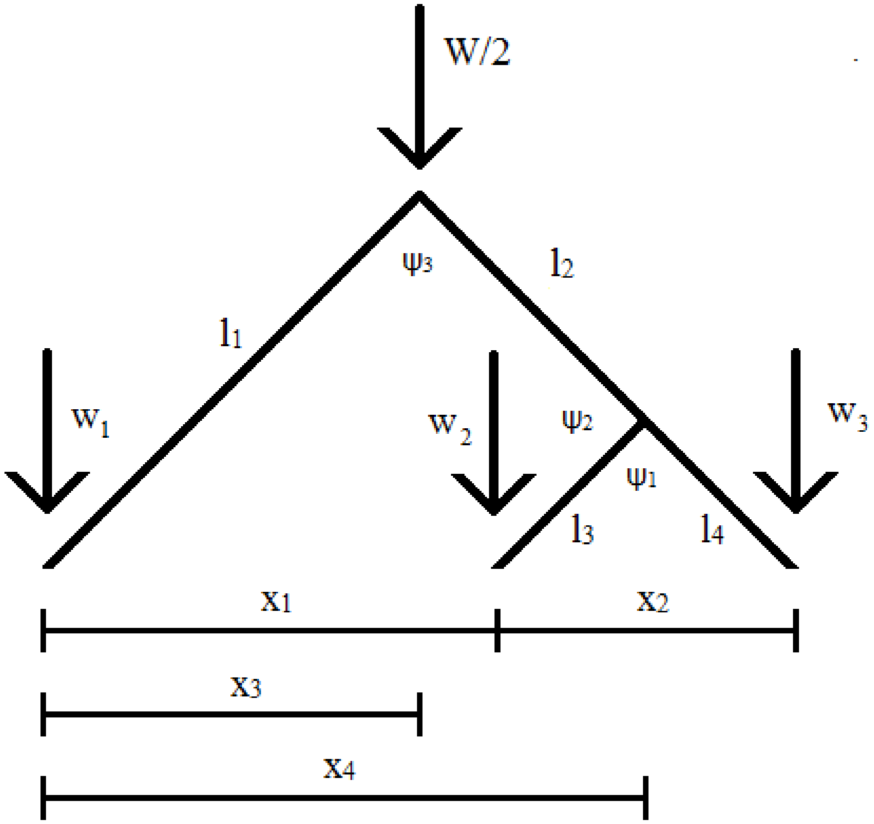


Figure 4.2: Single Side Rocker-Bogie Configuration

The concept of an operating angle for each side of the rover suspension system is of key importance to our model. Shown in Figure 4.3, the operating angle is used to determine the rover configuration at any given point and heading as well as determine whether or not rover pitch exceeds its safety threshold. Operating angles are dependent on the value of ψ_2. When $\psi_2 < \psi_{2flatground}$ the angle is defined as the angle between the front and rear wheels and the horizontal and when $\psi_2 \geq \psi_{2flatground}$ the angle is defined as the angle between the rear and middle wheels and the horizontal. $\psi_{2flatground}$ represents the value of ψ_2 when the rover is on an even surface. It is important to note that on uneven terrain, the operating angle for the left and right sides of the rover will almost certainly be different. The variable z_{plus} is used to determine the best fit rover configuration at all terrain points and headings. It represents the vertical distance between the wheel not used to define the operating angle of the rover side and the line defined by the operating angle itself.

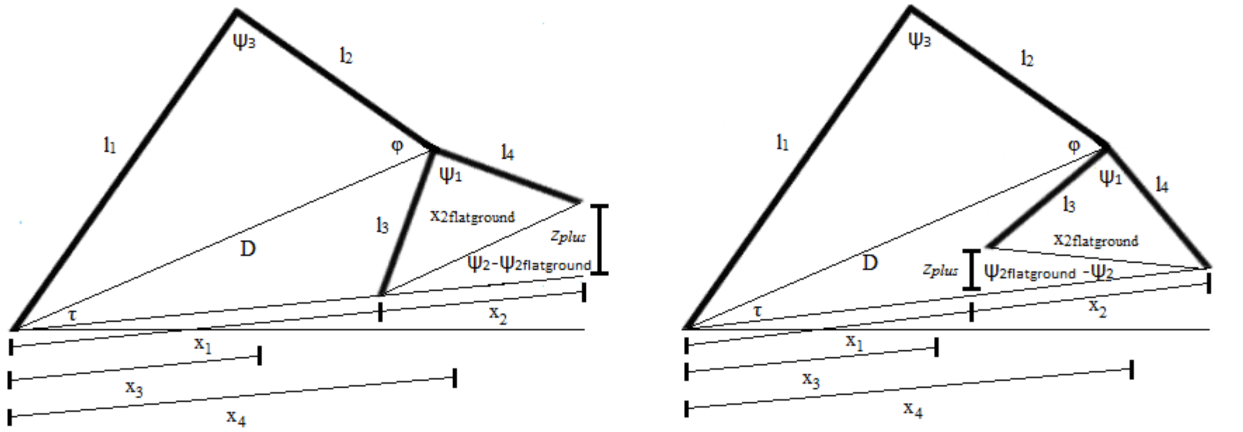


Figure 4.3: Relationship Between φ_2 and Operating Angle

Our model accepts as input a rover configuration file containing W, l_1, l_2, l_3, l_4, ψ_1, ψ_{2min}, ψ_{2max}, $\psi_{2flatground}$ and ψ_3 and first calculates the set of rover configurations for every possible combination of ψ_2 across its entire range at 1° increments. The set of possible configurations is then reduced by considering positions in which wheel positions differ by less than 0.01 m to be duplicates and eliminating all but one of each. Distance offsets from the rover centre of mass to each wheel are then calculated in the x and y direction for every remaining configuration at every heading.

Next, the DEM and z_{plus} are used to select the rover configuration which best maintains contact between each wheel and the terrain for each side at every position and heading. Once the rover configuration for each side at each point and heading has been determined, the rover width is then used to select the best fit configuration for each side for the centre of mass at every location and heading. With each rover side configuration for the centre of mass at every point and heading determined, the roll angle of the rover can be easily calculated using the provided DEM. Next, the vertical wheel loads w_1, w_2, w_3 are be calculated at every point and heading. The 2-D nature of the system means that achieving a solution requires solving a number of triangles; these are illustrated in Figures 4.4 and 4.5.

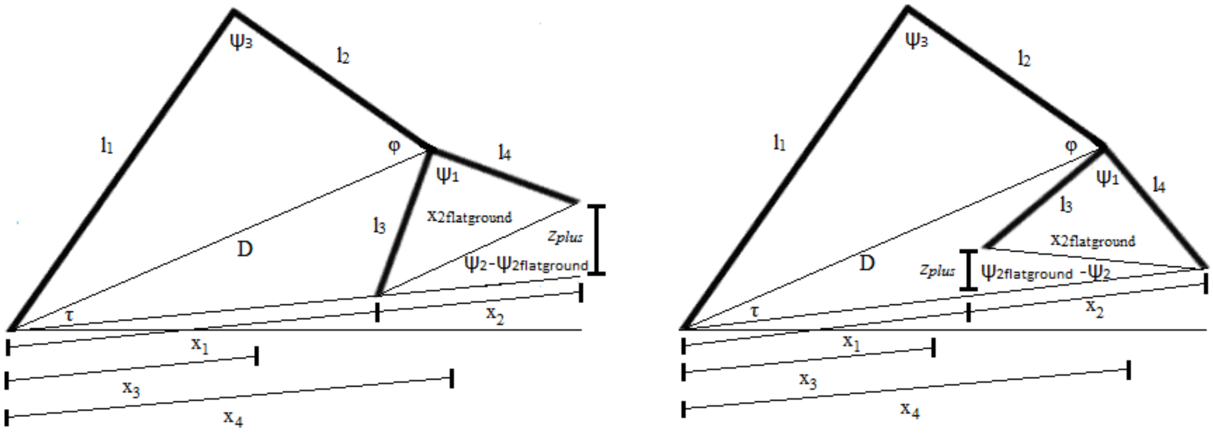


Figure 4.4: Intermediate Values for Rocker-Bogie Model

The intermediate lengths D and $x_{2flatground}$ along with angle φ can be found using equations 4.1-4.3:

$$D = \sqrt{{l_1}^2 + {l_2}^2 - 2l_1 l_2 cos\psi_3} \qquad (4.1)$$

$$\varphi = asin\left(\frac{l_1 sin\psi_3}{D}\right) \qquad (4.2)$$

$$x_{2flatground} = \sqrt{{l_3}^2 + {l_4}^2 - 2l_3 l_4 cos\psi_1} \qquad (4.3)$$

$x_{2flatground}$ then allows for the calculation of the configuration variable z_{plus} via equation 4.4:

$$z_{plus} = sin\left(abs(\psi_2 - \psi_{2flatground})\right) x_{2flatground} \qquad (4.4)$$

The lengths x_1 and x_2 can be determined using equations 4.5-4.7:

$$x_2 = \cos(\psi_{2flatground} - \psi_2)x_{2flatground} \qquad (4.5)$$

$$x_1 = \sqrt{{l_3}^2 + D^2 - 2l_3D\cos(\psi_2 - \varphi)} \quad \{\psi_2 \geq \psi_{2flatground}\} \qquad (4.6)$$

$$x_1 = \sqrt{D^2 + {l_4}^2 - 2Dl_4\cos(\psi_2 - \varphi + \psi_1)} - x_2 \quad \{\psi_2 < \psi_{2flatground}\} \qquad (4.7)$$

Equations 4.8 and 4.9 can then yield angle τ:

$$\tau = asin\left(\frac{l_3 sin(\psi_2 - \varphi)}{x_1}\right) \quad \{\psi_2 \geq \psi_{2flatground}\} \qquad (4.8)$$

$$\tau = asin\left(\frac{l_4 sin(\psi_2 - \varphi + \psi_1)}{x_1 + x_2}\right) \quad \{\varphi \leq \psi_2 < \psi_{2flatground}\} \qquad (4.9)$$

Lengths x_3 and x_4 can now be evaluated using equations 4.10 and 4.11:

$$x_3 = l_1 sin\left(\varphi + \psi_3 - \tau - \frac{\pi}{2}\right) \qquad (4.10)$$

$$x_4 = Dsin\left(\frac{\pi}{2} - \tau\right) \qquad (4.11)$$

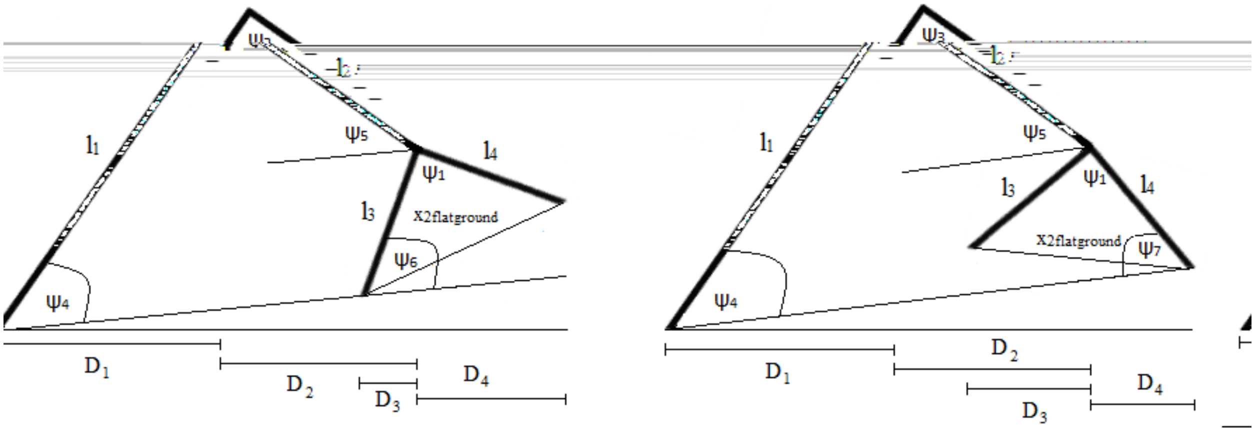


Figure 4.5: Additional Intermediate Values for Rocker-Bogie Model

Angles ψ_4, ψ_5, ψ_6 and ψ_7 are between the specified linkage member and the operating plane and are calculated using equations 4.12-4.18.

$$\psi_4 = \frac{\pi}{2} - asin\left(\frac{x_3}{l_1}\right) \qquad (4.12)$$

$$\psi_5 = \frac{\pi}{2} - asin\left(\frac{x_4 - x_3}{l_2}\right) \qquad (4.13)$$

$$\psi_6 = asin\left(\frac{l_4 sin\theta_1}{x_{2flatground}}\right) + \psi_2 - \psi_{2flatground} \quad \{\psi_2 \geq \psi_{2flatground}\} \qquad (4.14)$$

$$\psi_7 = \pi - \psi_2 - \psi_6 \quad \{\psi_2 \geq \psi_{2flatground}\} \qquad (4.15)$$

$$\psi_7 = asin\left(\frac{l_3 sin\psi_1}{x_{2flatground}}\right) + \psi_{2flatground} - \psi_2 \quad \{\psi_2 < \psi_{2flatground}\} \qquad (4.16)$$

$$\psi_6 = \pi - \psi_2 - \psi_7 \quad \{\psi_2 < \psi_{2flatground}\} \qquad (4.17)$$

The distances distributed D_1, D_2, D_3 and D_4 along the horizontal can then be determined with equations 4.18-4.21:

$$D_1 = l_1 cos(\psi_4 + OperatingAngle) \qquad (4.18)$$

$$D_2 = l_2 cos(\psi_5 - OperatingAngle) \qquad (4.19)$$

$$D_3 = l_3 cos(\psi_6 + OperatingAngle) \qquad (4.20)$$

$$D_4 = l_4 cos(\psi_7 - OperatingAngle) \qquad (4.21)$$

Finally, D_1, D_2, D_3 and D_4 allow for the vertical loads on each wheel to be evaluated by balancing moments using equations 4.23-4.25:

$$w_1 = \frac{W}{2}\left(\frac{D_2}{D_1+D_2}\right) \qquad (4.23)$$

$$w_2 = \frac{W}{2}\left(\frac{D_2}{D_1+D_2}\right)\left(\frac{D_4}{D_3+D_4}\right) \qquad (4.24)$$

$$w_3 = \frac{W}{2}\left(\frac{D_2}{D_1+D_2}\right)\left(\frac{D_3}{D_3+D_4}\right) \qquad (4.25)$$

In order to create a map of rover configurations at every point and heading, our system assumes an 8-connected grid, an example of which is shown in Figure 4.6. Terrain data is accepted in the form of a Digital Elevation Model (DEM). A DEM is a 2-D matrix in which the matrix values represent terrain height and matrix indices represent location on the x-y plane. Each DEM segment is assumed to be flat and will have an area equivalent to the product of the x and y grid spacings, with the x-y coordinate being the middle of that section.

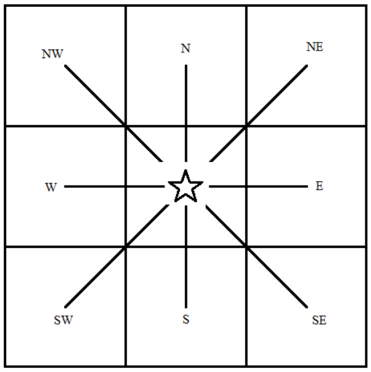


Figure 4.6: Possible Headings for 8-Connected DEM

Left and Right rover configuration for the centre of mass at every point and heading is determined by using the previously calculated wheel positions for each configuration and the assumption that all wheels are in constant contact with the ground. The absolute elevation of the non-operating angle defining wheel (z_{plus} plus the height of operating angle at the wheel's x-y location) is compared to the terrain elevation at that point for every configuration. The configuration for which these values are closest is then selected as the configuration at that point and heading. The end result is a left and right side configuration for every point at every heading. In the event that a given combination of rover side configurations exceeds preset roll and/or pitch thresholds or the best fit configuration does not match the terrain profile closely enough, that section of terrain is designated untraversable and rover configurations for each side will be set to null.

4.2 Wheel/Soil Interaction Model

The fundamental relationships and processes involving the interaction between soil and a rigid wheel were first analyzed and quantified by Bekker [9]. Figure 4.7 illustrates a number of basic terramechanics concepts.

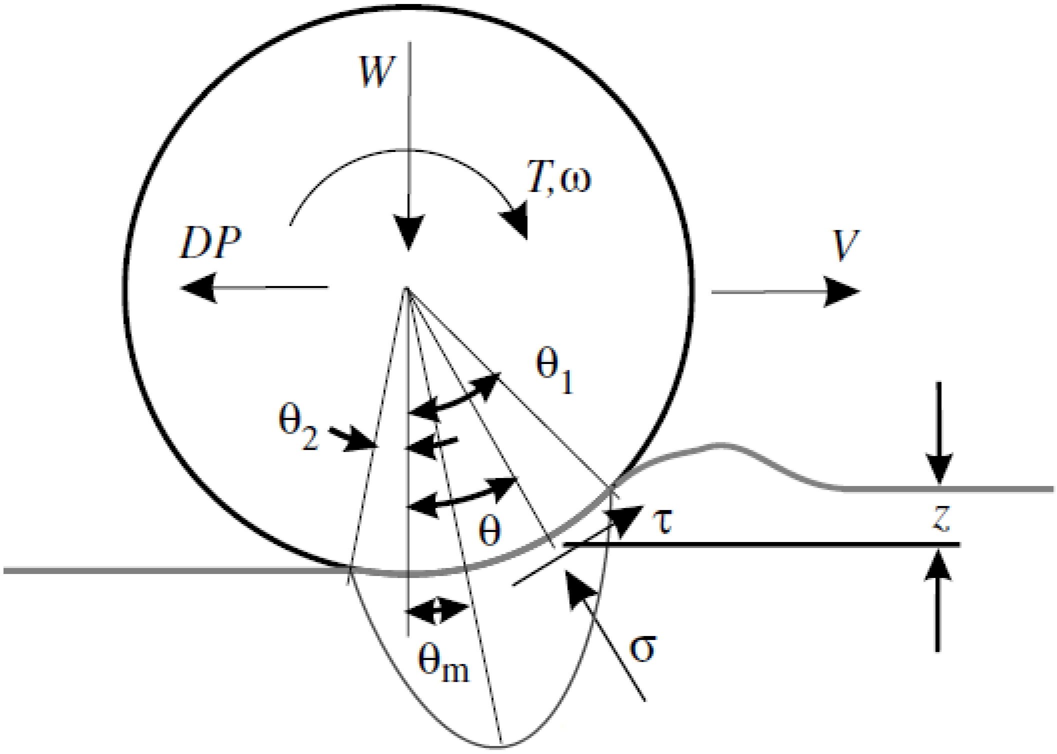


Figure 4.7: Basic Model of Wheel/Soil Interaction

W is the wheel load, T is the wheel torque and ω the angular velocity of the wheel. DP or Drawbar Pull is the towing force generated at the rear of the wheel; a DP of zero would result in a constant velocity if nothing was being towed. Wheel sinkage is represented by z, while τ and σ represent normal stress and shear stress acting at point θ along the wheel rim. θ_m, θ_1 and θ_2 are the location of maximum normal stress, the entry angle and the exit angle respectively.

Bekker's studies of wheel/soil interaction yielded the following relationships:

$$DP = rb\left(\int_{\theta_2}^{\theta_1} \tau(\theta) cos\theta d\theta - \int_{\theta_2}^{\theta_1} \sigma(\theta) sin\theta d\theta\right) \tag{4.26}$$

$$T = r^2 b \int_{\theta_2}^{\theta_1} \tau(\theta) d\theta \tag{4.27}$$

where r and b are the wheel radius and width, respectively. Due the complex nature of normal and shear stress equations, these equations cannot be solved analytically and a closed form solution is not possible.

Shibly was able to introduce a number of simplifications which approximated these equations over a range of tested values. His equations are as follows:

$$
\begin{aligned}
\frac{DP}{rb} = c\left[\frac{f}{f^2+1}\left(f sin\theta_1 - cos\theta_1 + e^{-f\theta_1}\right)\right] \\
&+ \frac{1}{\theta_m(\theta_1-\theta_m)}[(\theta_1 cos\theta_m - \theta_m cos\theta_1 - \theta_1 + \theta_m)tan\phi \\
&- (\theta_1 sin\theta_m - \theta_m sin\theta_1)]\sigma_m \\
&- \frac{1}{\theta_m(\theta_1-\theta_m)(f^2+1)^2}\left[(f^2-1)\left(\theta_m cos\theta_1 - \theta_1 cos\theta_m e^{-f(\theta_1-\theta_m)}\right)\right. \\
&+ 2f\left(\theta_m sin\theta_1 - \theta_1 sin\theta_m e^{-f(\theta_1-\theta_m)}\right) \\
&\left.+ (f^2-1)(\theta_1-\theta_m)e^{-f\theta_1}\right]\sigma_m tan\phi
\end{aligned}
\tag{4.28}
$$

$$
\begin{aligned}
\frac{T}{r^2 b\theta_1} = c\left(1 - \frac{1-e^{f\theta_1}}{f\theta_1}\right) \\
&+ \left[\frac{1}{2}\right. \\
&\left.- \frac{1}{f^2{\theta_1}^2\theta_m(\theta_1-\theta_m)}\left(\theta_m - \theta_1 e^{-f(\theta_1-\theta_m)} + (\theta_1-\theta_m)e^{f\theta_1}\right)\right]\sigma_m tan\phi
\end{aligned}
\tag{4.29}
$$

$$
f = \frac{r}{2k}[(1+i) - (1-i)cos\theta_1] \tag{4.30}
$$

Where c, ϕ, σ_m, i and k represent the soil cohesion, angle of internal shearing resistance, maximum normal stress, slip and modulus of shear deformation respectively.

The location of maximum normal stress (θ_m) can be assumed to be halfway between θ_1 and θ_2, and θ_2 can be assumed to be zero [9][10]. Both these assumptions were used by Shibly, and have been shown to be valid for low cohesion soils similar to those found on Mars. These assumptions allow θ_m to be calculated using the following:

$$
\theta_m = \frac{\theta_1}{2} = \frac{\frac{z}{sin\left(\frac{\pi}{2}-\theta\right)}}{r}\cdot\frac{\pi}{2} \tag{4.31}
$$

where θ is the ground angle and z is the wheel sinkage. Ground angles were calculated for every heading at every point by finding the slope to the adjacent point in all 8 directions.

The relationship between applied load and wheel sinkage was first defined by Bekker and is as follows:

$$z = \left(\frac{3w}{b(3-n)\left(\frac{k_c}{b}+k_\phi\right)\sqrt{2r}} \right)^{\left(\frac{2}{2n+1}\right)} \tag{4.32}$$

where n, k_c, k_ϕ and w are the exponent of sinkage, soil cohesive modulus of deformation, soil shear modulus of deformation and applied load respectively.

Table 4.1 shows the range of soil parameters over which Shibly tested his approximations for accuracy, while Table 4.2 shows sample parameter values for a number of soil types.

Minimum Value	**Parameter**	**Maximum Value**
0.47	n	1.20
20.00	ϕ(°)	60.00
0.00	c(kPa)	3.00
0.00	k_c(kPa)	140.00
520.00	k_ϕ(kN/m^3)	680.00
0.01	k(m)	0.04
0.00	i	1.00

Table 4.1: Range of Tested Parameter Values for Shibly's Equations

Terrain	**n**	**k_c (kN/m^{n+1})**	**k_ϕ (kN/m^{n+2})**	**c (kPa)**	**ϕ (deg)**
Dry Sand	1.1	0.99	1528.43	1.04	28
Sandy Loam	0.7	5.27	1515.04	1.72	29
Sandy Loam	0.2	2.56	43.12	1.38	38
Sandy Loam	0.9	52.53	1127.97	4.83	20
Sandy Loam	0.4	11.42	808.96	9.65	35
Sandy Loam	0.3	2.79	141.11	13.79	22
Sandy Loam	0.5	0.77	51.91	5.17	11
Clayey Soil	0.5	13.19	692.15	4.14	13
Clayey Soil	0.7	16.03	1262.53	2.07	10
Heavy Clay	0.13	12.7	1555.95	68.95	34
Heavy Clay	0.11	1.84	103.27	20.69	6
Lean Clay	0.2	16.43	1724.69	68.95	20
Lean Clay	0.15	1.52	119.61	13.79	11
LETE Sand	0.79	102	5301	1.3	31.1
Upland Sandy Loam	1.1	74.6	2080	3.3	33.7
Rubicon Sandy Loam	0.66	6.9	752	3.7	29.8
North Clayey Loam	0.73	41.6	2471	6.1	26.6
Grenville Loam	1.01	0.06	5880	3.1	29.8

Snow (USA)	1.6	4.37	196.72	1.03	19.7
Snow (USA)	1.6	2.49	245.9	0.62	23.2
Snow (Sweden)	1.44	10.55	66.08	6	20.7

Table 4.2: Sample Soil Parameters [168]

It is obvious from the sample values that certain soil parameters are difficult to measure, and vary considerably even among soils with similar appearances. It is also evident that most of the listed values of k_ϕ fall outside the range tested by Shibly. However, as the sensitivity analysis detailed in section 6.1 shows, this variable has little impact on the outcome of the model.

Our system assesses energy cost by determining the torque required at each wheel to move the rover centre of mass between adjacent points whilst maintaining a DP of zero (constant velocity). Setting DP to zero leaves two equations and two unknowns (torque and slip). Shibly's equation for DP cannot be solved explicitly for i. However, i must exist in the range [0,1]. Our approach solves Shibly's equation for every value between zero and one at 0.001 increments, and sets slip as the value which yields a DP closest to zero. This value for i is then used to determine the torque required to move each wheel to each adjacent point. The end product is eight mxnx6 matrices, where m and n are the dimensions of the original DEM. Each matrix represents a separate heading, and each 1x6 vector represents the torque required at each wheel to move the centre of mass of the rover to the adjacent point in the specified direction.

In the event that wheel sinkage is less than 0.01 m, our system will not use the terramechanics principles previously detailed. In these cases the soil is treated as non-deformable, and torque is calculated using an assumed coefficient of friction between rock and aluminum of 0.3 using the following:

$$T = 0.3rw\cos(GroundAngle) \qquad (4.33)$$

Where r is wheel radius and w is vertical wheel load.

Finally, the gravitational contribution is either added (downhill) or subtracted (uphill) for each wheel as follows:

$$G = wr\cos(GroundAngle) \qquad (4.34)$$

As motor torque is directly related to energy consumption, the only supplemental information required by our model is the torque/current curve of the motor in use.

4.3 Testing Configuration

In order to test our wheel/soil interaction model, the testing mechanism shown schematically in Figure 4.8 and photographed in Figure 4.9 was designed and constructed. It consists of one half of a rocker-bogie suspension system, with an aluminum frame and wheels fabricated from PVC piping. The half rocker bogie has a mass of approximately 11.8 kg, and is configured as follows:

l_1=1.129m	l_2=0.475m	l_3=l_4=0.507m
φ_1=100°	φ_3=120°	φ_2=70±20°
r=0.0778m	b=0.15m	

The suspension was powered by three gear motors operating at 115VAC with a maximum generated torque of 0.784 Nm.

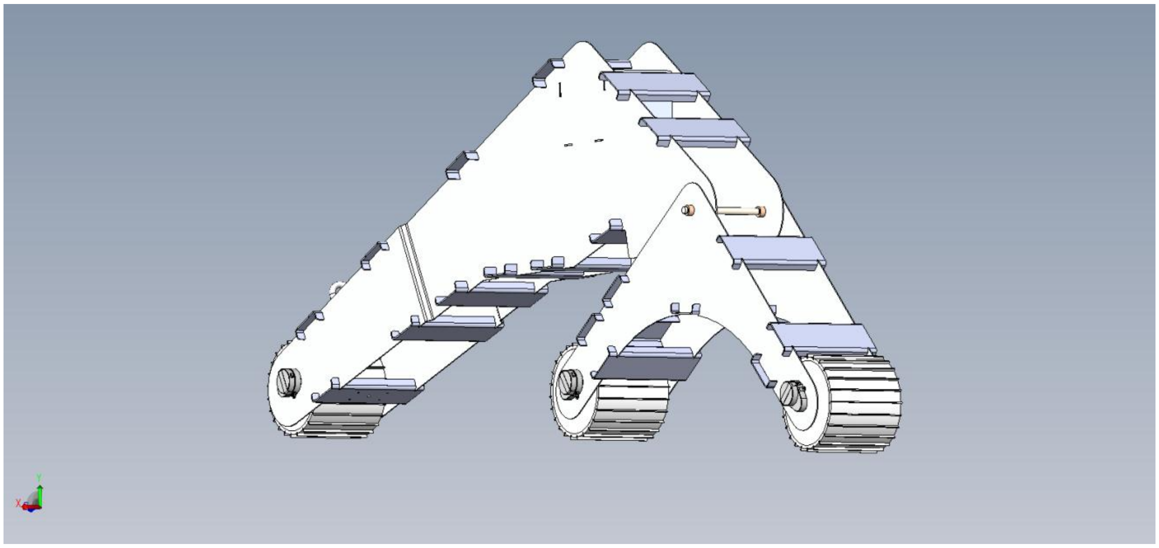


Figure 4.8: Half Rocker-Bogie Testing Platform Schematic

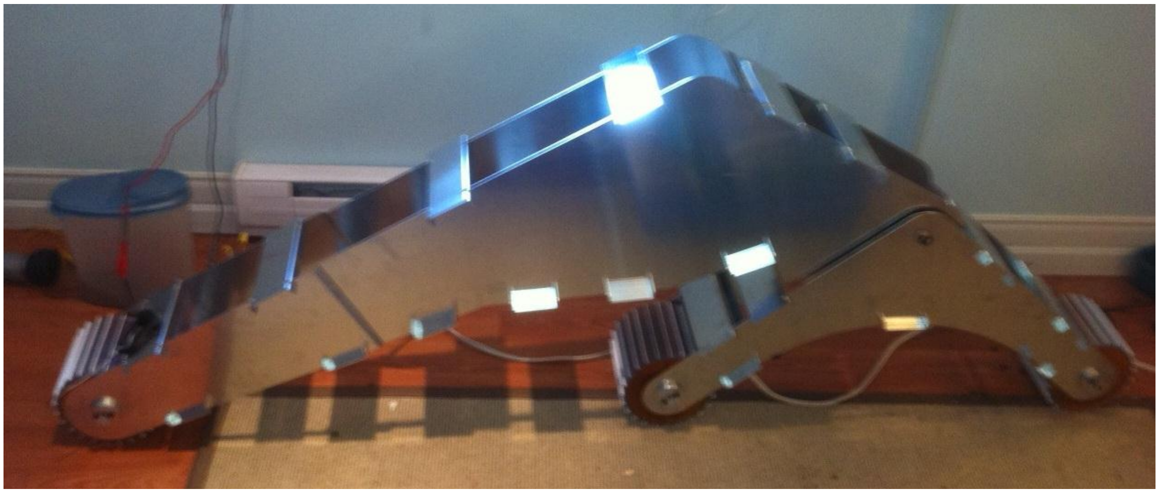


Figure 4.9: Half Rocker-Bogie Image

The testing platform shown in Figure 4.10 was specially designed to work with the half rocker-bogie to validate the wheel/soil interaction model. It consists of a 2x4 wooden frame with peg board at intervals corresponding to a range of potential wheel distances. The peg board was used to mount three platforms; one for each wheel. By altering the angle of the rear wheel platform, differing ground contact angles could be simulated. The rear wheel platform also contained a custom designed sandbox, Shown in Figure 4.11 which could be filled with soils of varying types.

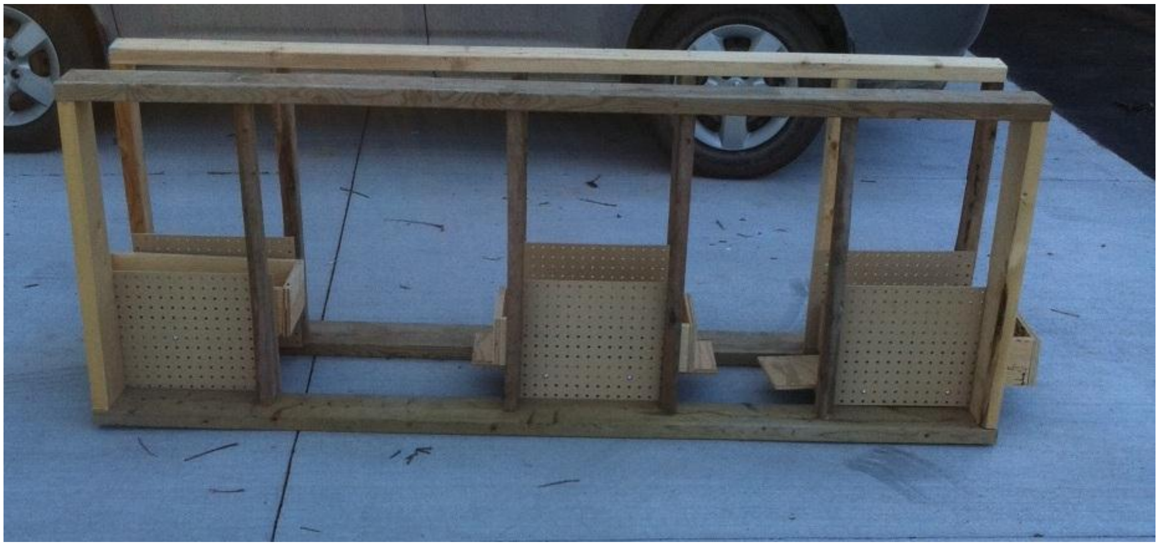


Figure 4.10: Testing Platform

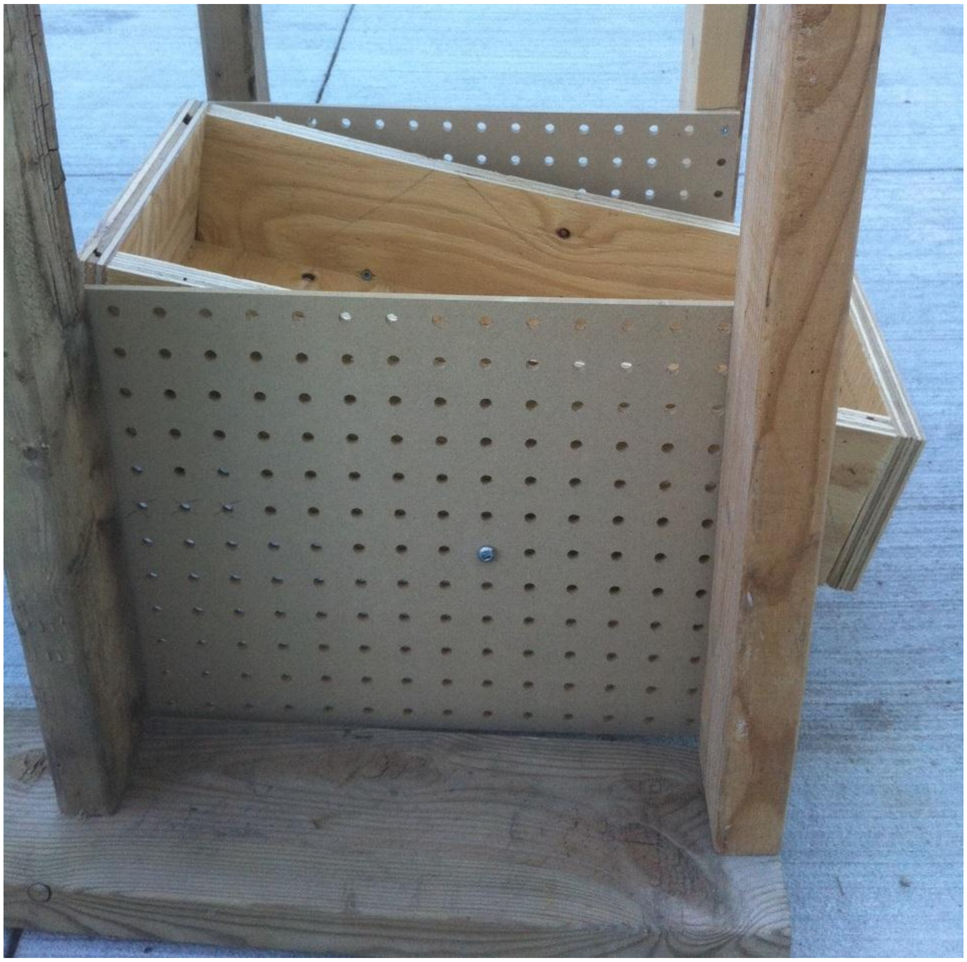


Figure 4.11: Adjustable Sand Box

4.4 Results

In order to determine the validity of our wheel soil interaction model, a series of trials were run using the half rocker-bogie suspension and the testing platform. A variety of soil types and ground contact angles were tested. This was done by applying a known torque to the rear wheel at a known ground angle and measuring the resulting Drawbar Pull. The front two wheels were left free to rotate and both had contact angles set constant at 0°. By measuring the sinkage for each test it was possible to eliminate the need to estimate n, k_c and k_φ. The Bekker soil parameters that were used are shown in Table 4.3. These parameters were determined a study of available information on terrestrial soils and some data analysis from our crusty soil sample (compaction over multiple test runs may have altered the parameters as the trials proceeded).

	Sandy	Crusty	Rocky
k (m)	0.02	0.025	0.01
c (kPa)	0.5	0.1	3
ϕ (°)	15	45	25

Table 4.3: Model Soil Parameters

Figure 4.12 shows the Drawbar Pull predicted by our model for zero slip over the range of contact angles and soil types tested. It shows a linear relationship between an increase in decline and generated Drawbar Pull. It also shows that generated Drawbar Pull is higher under identical conditions for crusty soil than for rocky and higher for rocky than for sandy.

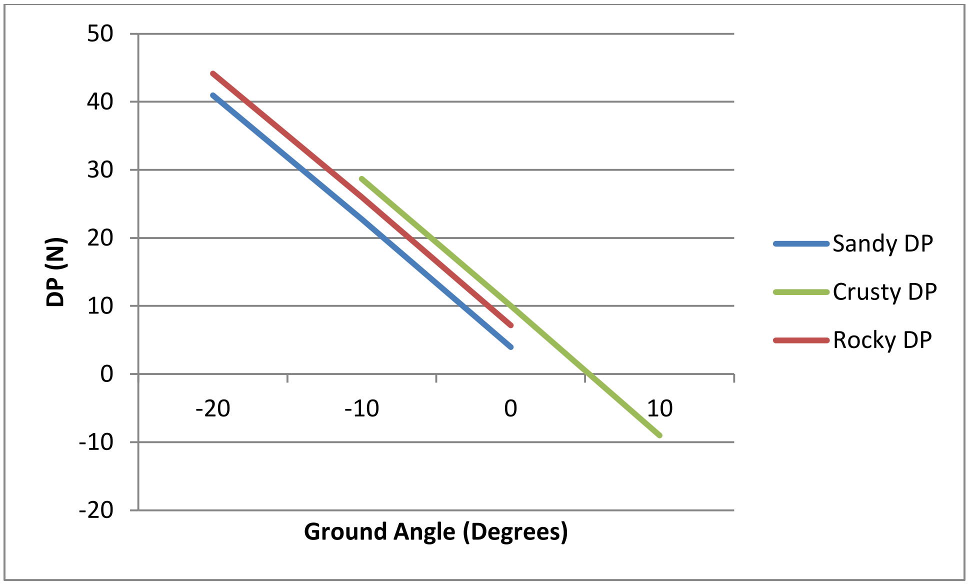


Figure 4.12: Predicted DP vs. Ground Angle for Zero Slip

Initial results were highly unexpected. It was observed that measured Drawbar Pull actually increased proportionately to the slope of the ground angle. This was the opposite of what our model (and basic physics) would indicate. It was discovered that these results were a consequence of the fact that the front two wheels were unpowered, not moving and at a ground angle of zero degrees. This resulted in a transfer of force back through the rigid frame in the opposite direction of the Drawbar Pull. The results shown in Figures 4.13-4.18 correct for this by adding this opposing force back to the Drawbar Pull using equation 4.35.

$$DP_{corrected}=DP_{measured}+F_{wheel}Sin(GroundAngle) \qquad (4.35)$$

Once this opposing force had been taken into account results were similar to the model's predictions. In all the test scenarios, measured Drawbar Pull was higher for crusty soil than both other types given the same conditions. In all but a single case (-10°) measured DP was higher for rocky soil than it was for sandy. Generated Drawbar Pull also decreased as ground slope increased in all cases.

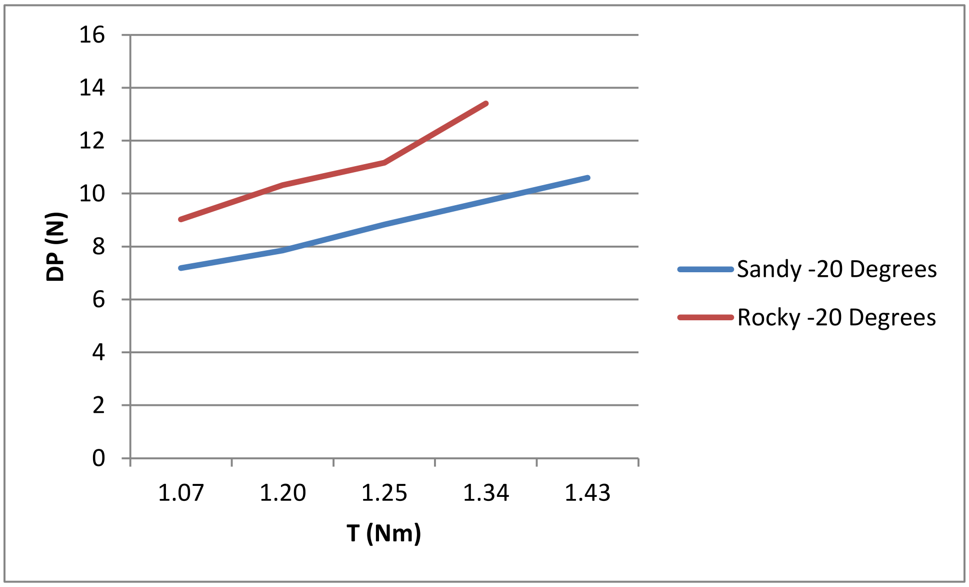


Figure 4.13: Corrected DP vs. T -20 Degrees

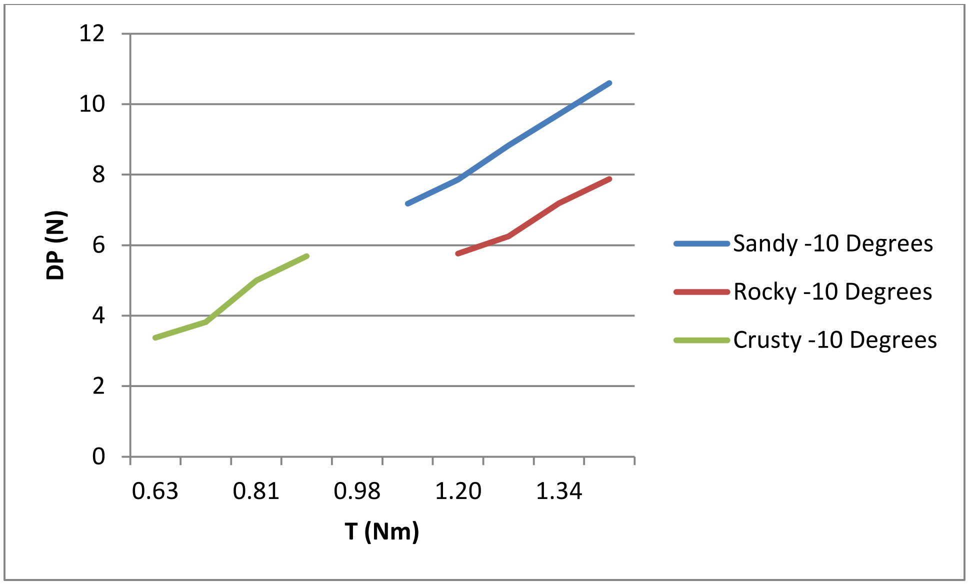


Figure 4.14: Corrected DP vs. T -10 Degrees

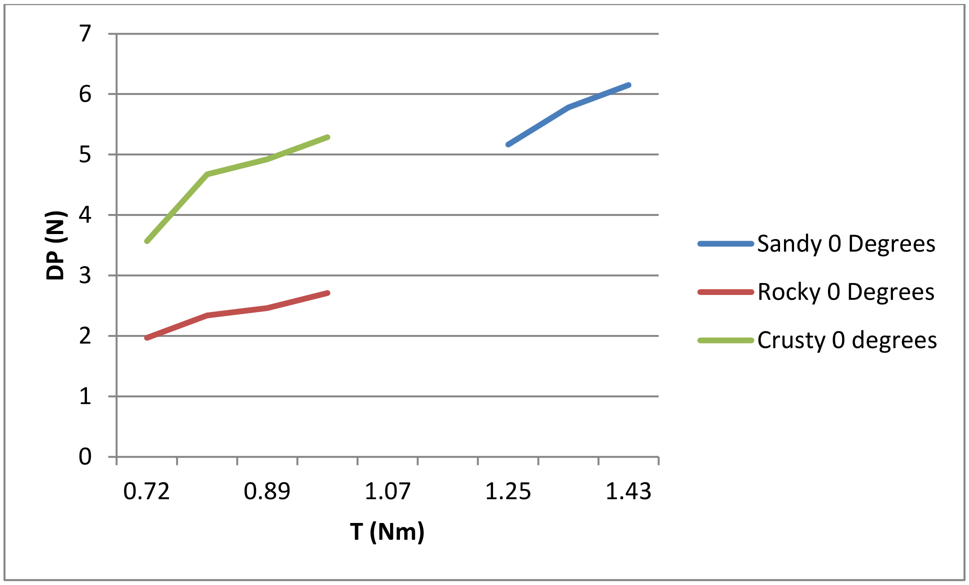


Figure 4.15: Corrected DP vs. T 0 Degrees

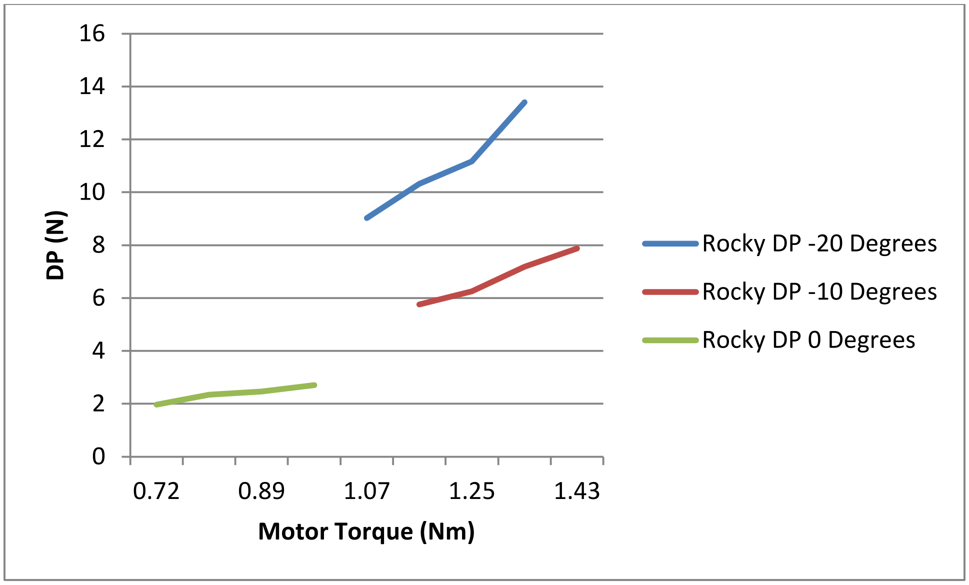


Figure 4.16: Corrected DP vs. Motor Torque: Rocky Soil

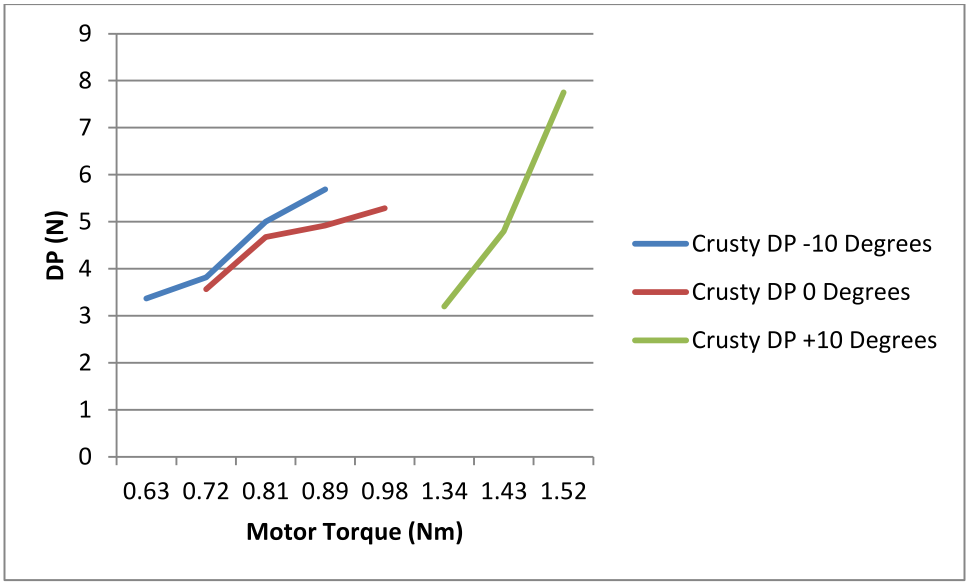


Figure 4.17: Corrected DP vs. Motor Torque: Crusty Soil

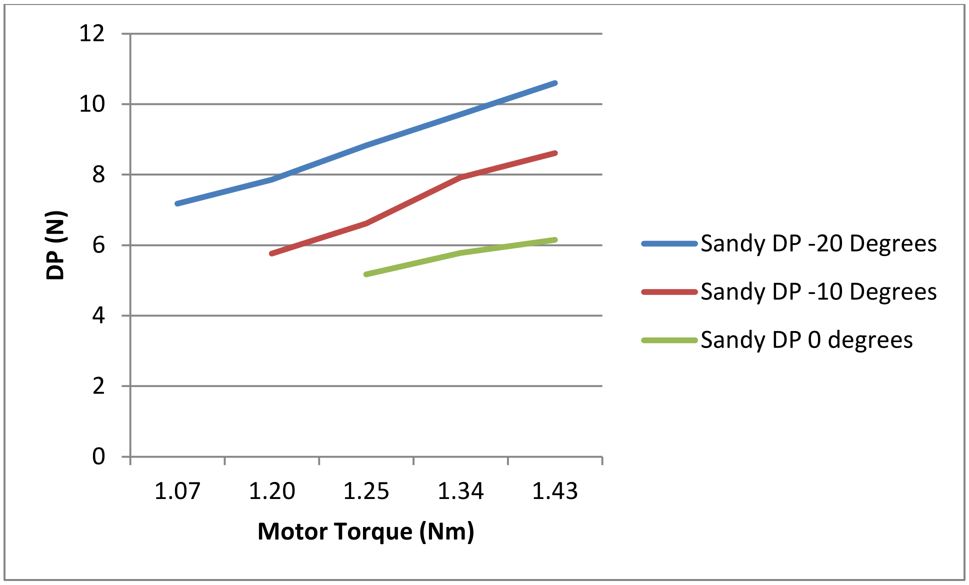


Figure 4.18 Corrected DP vs. Motor Torque: Sandy Soil

While the model fails to correctly predict numerical values for generated Drawbar Pull on inclined terrain, the predicted values are close to those measured on flat ground. The model also does a good job of estimating the relative relationships between different soil types and their energy requirements for traversal. This suggests that future tuning of the model, especially the gravitational contribution to Drawbar Pull, could lead to more accurate numerical results.

5.0 Path Planner

Path planning for complete coverage has traditionally focused on minimizing time or generating safe paths which avoid obstacles. The background section detailed an extensive search of existing literature; we were unable to find an existing approach to minimizing energy consumption for paths of complete coverage.

5.1 Terrain Representation

Our system receives a series of eight *mxn* matrices as input. These matrices correspond exactly to the DEM provided to the energy estimation model in terms of positional information. However, as shown in Figure 5.1, the values in each matrix represent the energy required to move the rover to one of eight possible adjacent cells. It should also be noted that values at *i=1*, *i=m*, *j=1* and *j=n* must be null in any direction that exits the predefined area.

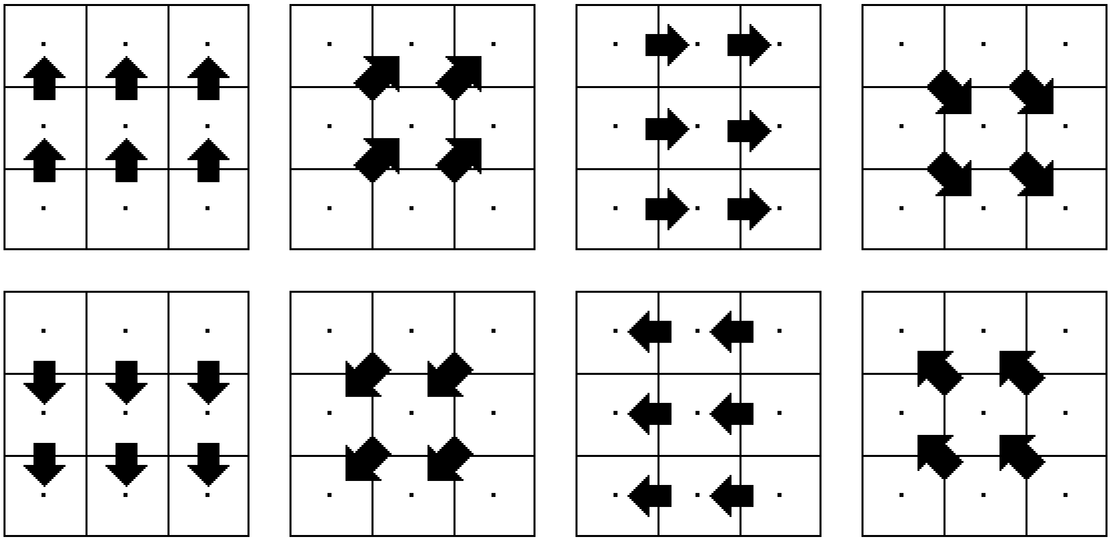


Figure 5.1: System Input

5.2 Accessibility

Outdoor environments often contain swaths of terrain which are inaccessible due to limitations in the physical design of the rover. These areas will have been predefined by our energy estimation model, and their associated map values set to null. As such, our complete coverage algorithm can only operate over the areas the rover is actually capable of reaching, and these must be determined before any further action can be taken. This is done using a variation of the two-pass region finding algorithm. A standard implementation of this algorithm passes through the grid from top left to bottom right. On the first pass, cells located to west, northwest, north and northeast of each cell are examined (see Figure 5.2).

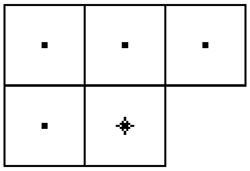


Figure 5.2: West, Northwest, North and Northeast Neighbours

The current cell is assigned the minimum region number of all its connected neighbours, and any of those connected neighbours with different region values are reassigned that same minimum value. A table monitoring connections between different numbered regions is then updated to indicate that all the different region numbers for the connected cells are in fact part of the same region. In the event that the current cell is not connected to any of the four analyzed neighbours, that cell is assigned a new region number. The second pass analyzes the region equivalence table and sets each cell's region number to the minimum equivalent region.

This procedure differs slightly in our implementation since it is possible for every one of a block of 5 examined cells to be part of the same region even if the current cell has only one connection (see Figure 5.3). As there are 7 potential connections in each 5-cell block that is analyzed, the number of different connection scenarios is equal to:

$$C(7,0)+C(7,1)+C(7,2)+C(7,3)+C(7,4)+C(7,5)+C(7,6)+C(7,7)=128$$

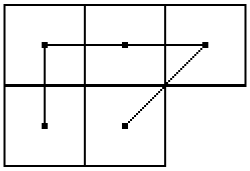


Figure 5.3: All Cells Connected With a Single Connection from Current Cell

Each of these 128 possibilities corresponds to a region ranging in size from a single cell (no connections from current cell to adjacent cells) to all five cells. Although the number of scenarios is substantially increased in our version, the algorithms function in an identical manner, and the second pass requires no changes in implementation.

5.3 Movement Cost

The path planning algorithm assesses each accessible adjacent square and assigns it a cost, the cell with lowest cost is selected as the next target. It was determined early on in the development process that multiple visits to the same cell would be permitted. This was an easy decision, as the stated goal was complete coverage whilst minimizing energy consumption. One can envision any number of scenarios in which a robot retracing its path might not only be more energy efficient, but necessary in order to exit an isolated region. This choice differentiates the problem from the classically studied "travelling salesman" dilemma, as that scenario does not permit repeat visits. Algorithm development was conducted largely by trial and error, with the following factors having been considered:

- The energy cost to move to the adjacent cell in question
- The density of unexplored space in the direction of the cell in question
- The lowest number of cells between the cell in question and unexplored space
- The number of times the cell in question had previously been visited

The first stage of algorithm development involved developing a reliable method for complete coverage without incorporating energy costs. Various combinations of the above factors were tested and the results analyzed, and the following trends were observed:

- Attempting to bias the direction of movement in favour of the density of unexplored space (unexplored cells/total accessible cells) in that direction led to more costly paths, and often prevented completion on highly disjointed terrain.
- Relying on a linear relationship between costs and the number of previous visits to a cell in certain scenarios (complex, highly disjointed maps) was insufficient to ensure complete coverage.

These observations, along with a significant amount of trial and error led to the development of equation 5.1

$$C = D + H^2 \qquad (5.1)$$

Where C is the movement cost to each adjacent cell, D is the number of cells between each cell and unexplored space and H is the number of times that cell has been visited.

The next step was to test a number of different energy components to determine the most effective way to minimize energy costs. This was done by randomly generating different energy cost maps with the fraction of null connections (adjacent cells which are unconnected) ranging from 0 to 70%. Figure 5.4 shows the results of these tests for energy components ranging from 0 (equation 5.1) to two times E, where E is the energy cost to move to each adjacent cell and Emax is the maximum movement cost in the map being explored.

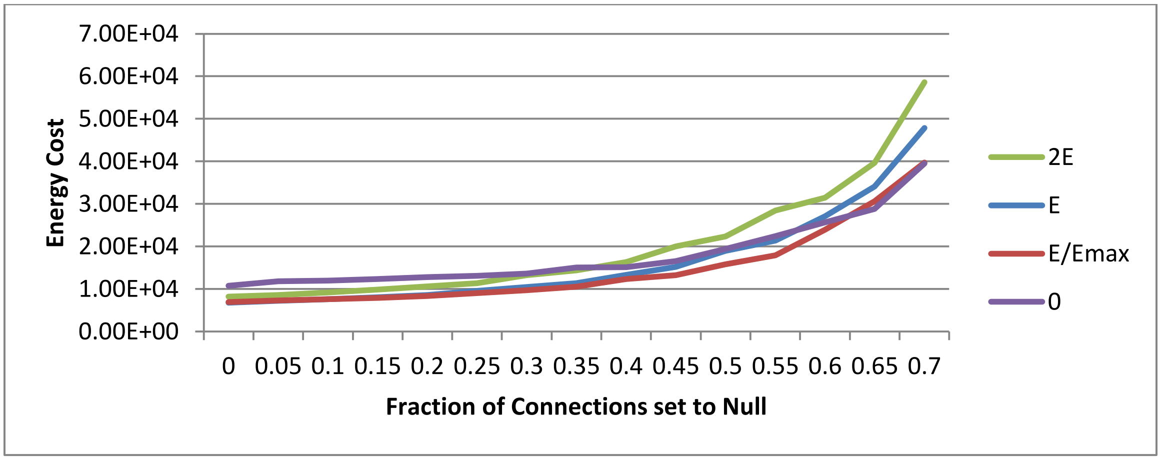


Figure 5.4: Energy Cost for Complete Coverage vs. Fraction of Null Connections

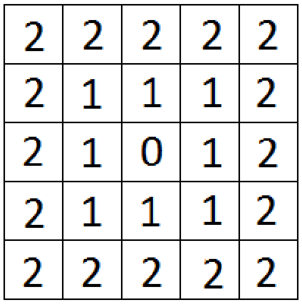
As maps became highly disjointed, the zero energy equation became increasing efficient. However, for the vast majority of cases, E/Emax yielded the most energy efficient paths of complete coverage. Thus, the final version of our path planning algorithm calculates movement cost to each accessible adjacent cell using Equation 5.2:

$$C = D + \frac{E}{MaxE} + H^2 \qquad (5.2)$$

This calculation is performed for each connected adjacent cell, and the one with the minimum cost is selected as the next destination. The entire path (or paths) can be calculated a-priori, or each movement can be determined in real-time to take into account improved knowledge of ground conditions.

5.4 Calculation of *D*

The value of *D* is calculated by analyzing the history map. If the cell under consideration is unexplored, *D* is assigned a value of zero. If its history value is greater than or equal to one, the surrounding eight cells are analyzed. If all of those have been visited, the surrounding sixteen cells are examined. Provided that the range of cells examined is not limited by the boundary of the entire area being covered, the number of cells examined is equal to *8D*, where *D* is the number of cells between the current cell and unexplored space. Figure 5.5 illustrates *D* values from the centre cell.



2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

Figure 5.5: Distance Values from Centre Cell

This approach can lead to misleading *D* values for regions with a large number of null values (ie the nearest unexplored region is inaccessible), but as shown in the results section, the effect is negligible.

5.5 Results

A series of simulations were run to determine the effectiveness of our methodology. Tests were run initially for a single rover over a completely connected area. A second series of tests was then run over an area with many null values. The completely connected and multiple null value scenarios were also used to validate complete coverage by multiple rovers. Multiple rover test cases were performed for two, three and four rovers with varying starting positions.

5.5.1 Completely Accessible Area

This test case was chosen because it is easily compared to a boustrophedon (back and forth) coverage pattern. Our initial test case was a 100x100 map with all energy costs in all directions set to one. We then introduced a normally distributed adjustment to each energy cost with a standard deviation ranging in value from 0.01 to 1.30 and compared the energy costs of a Boustrophedon coverage pattern against those generated by our own system. When performing our comparison the starting point was held constant at (25,25) and 16 potential Boustrophedon paths were analyzed (all potential back and forth routes from every corner plus the cost to reach that initial corner) with lowest cost path being chosen for comparison. Each test case was performed four times and the results were averaged.

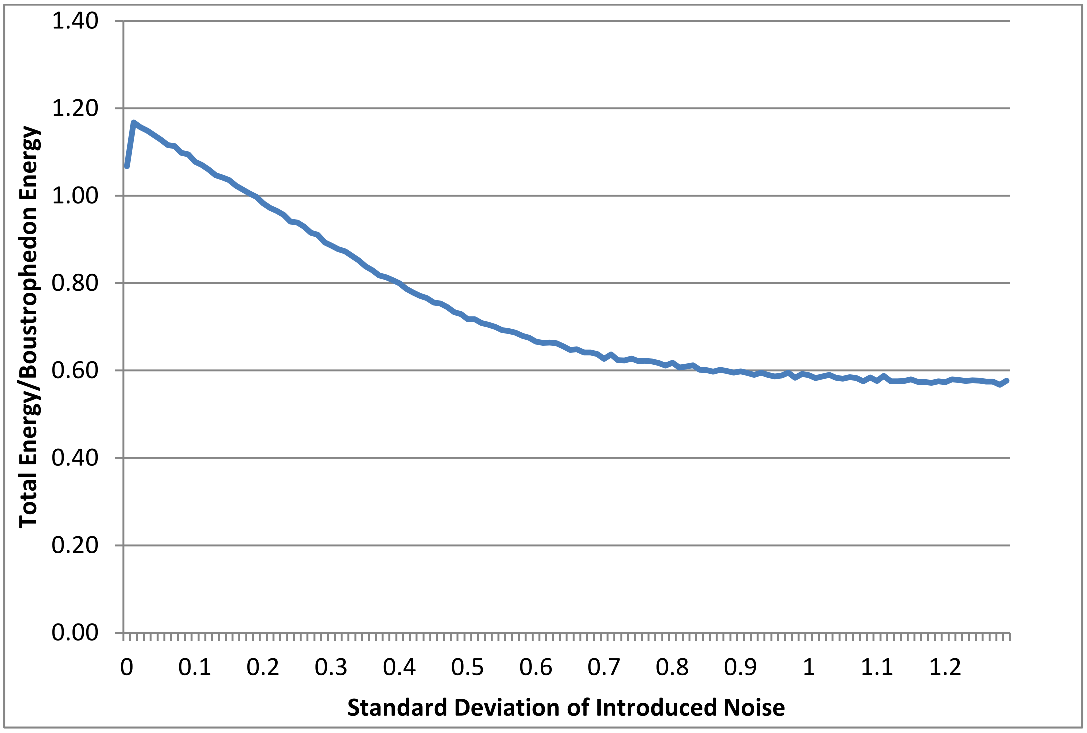


Figure 5.6: Energy Ratio vs. Introduced Noise

Figure 5.6 shows the energy consumption of our system over the energy consumption of the lowest cost Boustrophedon path as a function of the standard deviation of the introduced noise. When all energy values are equal, our system requires just under 1.07 times the energy of the cheapest Boustrophedon path. When we begin to introduce noise to the map, that figure initially jumps to approximately 1.17 and then steadily falls as the standard deviation is increased. Our system becomes more efficient when the standard deviation exceeds 0.2, and at its maximum efficiency requires only 0.56 times the energy of the best case boustrophedon path to achieve complete coverage.

The number of moves required by our system to achieve complete coverage was 10597 for the equivalent energy case. This jumped with the introduction of noise, but was steady, averaging approximately 11680.

5.5.2 Map with Null Values

In order to assess the effectiveness of our algorithm on a more realistic case – one where not every adjacent point is connected – a map generated by our energy consumption model with simulated terrain input was used. The accessible region of this map is illustrated by the white pixels in Figure 5.7. The map is a 400 by 400 square. Table 5.1 shows the number of null values for each direction map. It should be noted that when these values are calculated, any null value in a single direction results in a null value in the opposite direction from the appropriate adjacent location. For example, if North(2,1) was null, South(1,1) would also be set to null. The discrepancies in the number of null values in the corresponding directions are a result of the different number of border pixels in each direction.

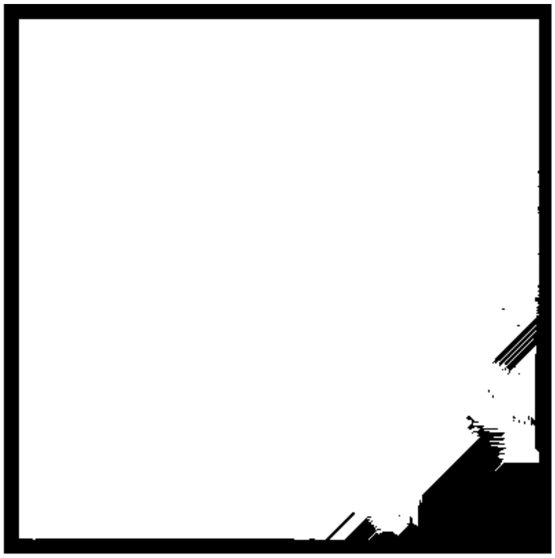


Figure 5.7: Accessible Area on Test Map

Direction	Number of Null Values
North	9579
Northeast	3135
East	14700
Southeast	18309
South	9538
Southwest	3199
West	14707
Northwest	18310

Table 5.1: Null Values for Each Direction Map

The accessible region for the tested data had an area of 136,285 points, and complete coverage by our algorithm was tested from 5 different starting points. Results are summarized in Table 5.2.

Starting Point	# Points Visited	Max # Number Visits	# with > 5 Visits	Total Energy Consumption
(25,25)	183275	20	1210	9592
(25,375)	199669	25	2587	11682
(375,25)	208685	19	4091	12726
(325,330)	189628	23	2200	10661
(200,200)	195454	31	2259	11194

Table 5.2: Results for Complete Coverage of Map with Null Values

Figures 5.8 through 5.12 show number of times each point was visited during each of the five tests, with the red point indicating rover starting location for that test. Figure 5.13 illustrates the number and location of input data null values. The data displayed in the maps makes evident that high numbers of repeat visits occur only when the density of null values are high.

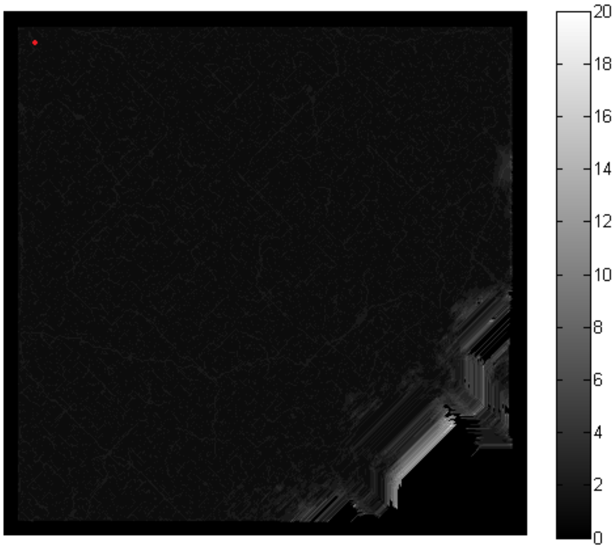


Figure 5.8: Test 1 Starting Point and Density of Visits

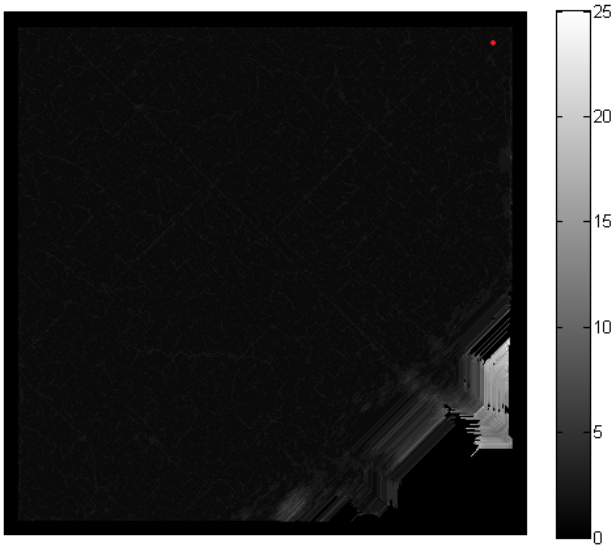


Figure 5.9: Test 2 Starting Point and Density of Visits

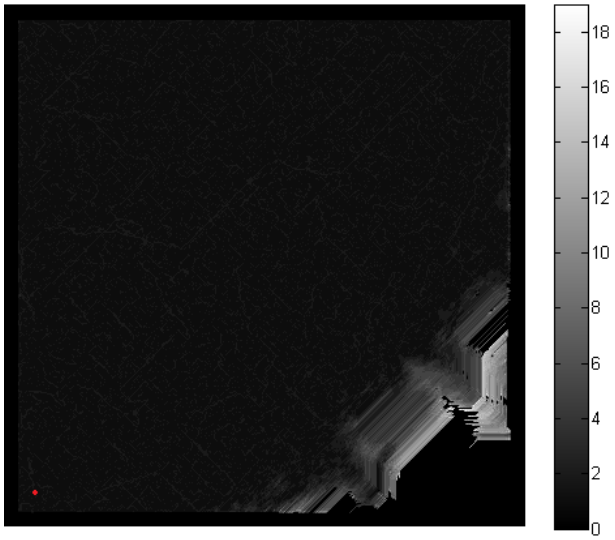


Figure 5.10: Test 3 Starting Point and Density of Visits

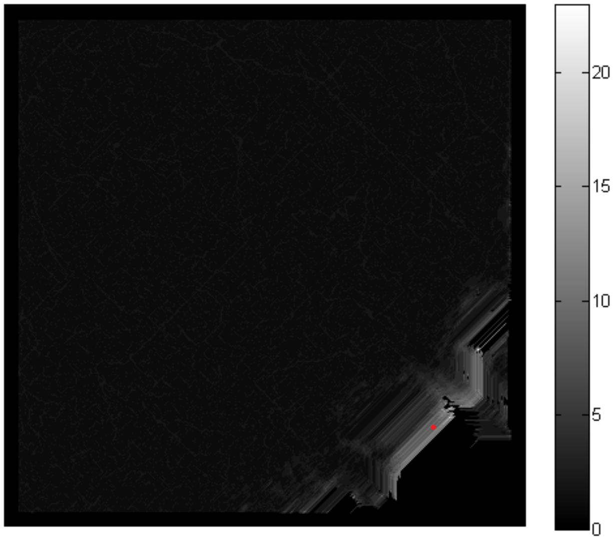


Figure 5.11: Test 4 Starting Point and Density of Visits

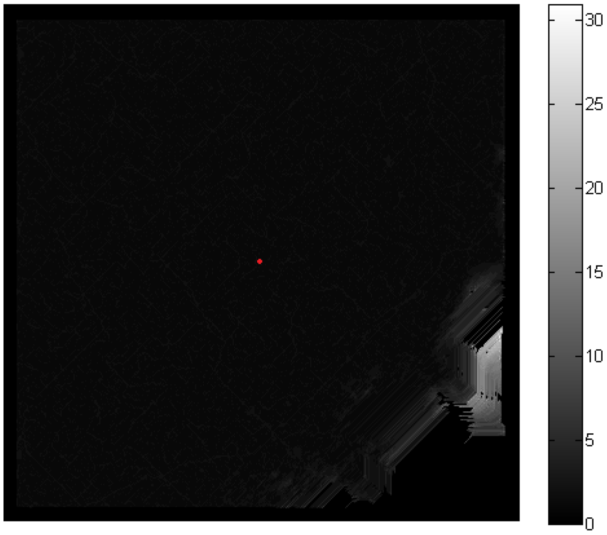


Figure 5.12: Test 5 Starting Point and Density of Visits

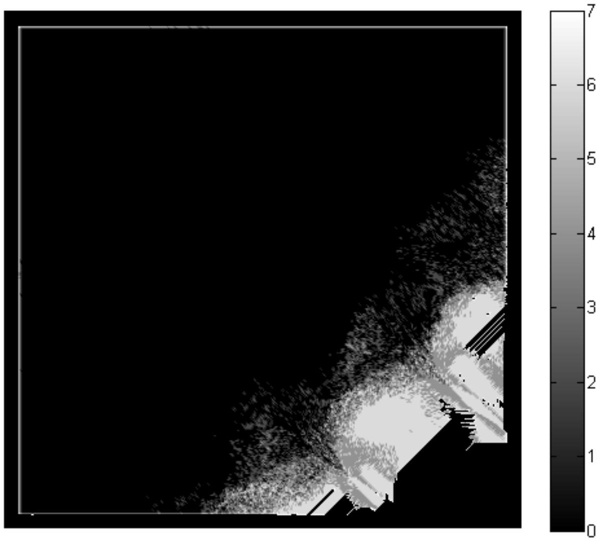


Figure 5.13: Input Map Density of Null Values

5.5.3 Multiple Rovers on Completely Connected Map

In a similar manner to our first set of tests, our initial test case was a 100x100 map with all energy costs in all directions set to one. We again introduced a normally distributed adjustment to each energy cost with a standard deviation ranging in value from 0.05 to 1.25 and compared the energy costs of the best case Boustrophedon coverage pattern against those generated by our own system. These tests were performed for groups of two, three and four rovers for both clustered and dispersed starting points. Clustered starting points were {(25,25), (25,26)} for two rovers, {(25,25), (25,26), (26,25)} for three rovers and {(25,25), (25,26), (26,25), (26,26)} for four rovers. Dispersed starting points were {(25,25), (25,75)} for two rovers, {(25,25), (25,75), (72,25)} for three rovers and {(25,25), (25,75), (72,25), (75,75)} for four rovers.

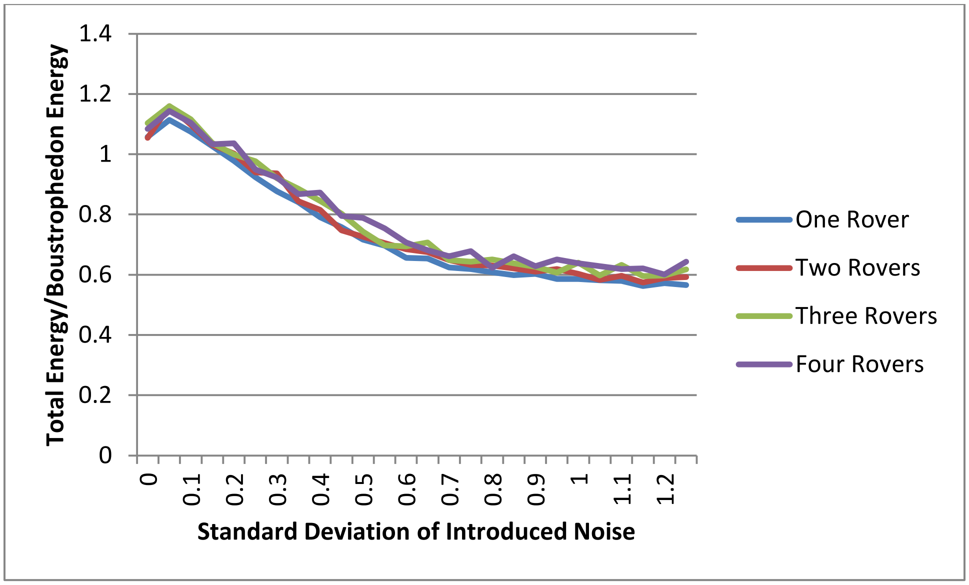


Figure 5.14: Ratio of Energy Consumption vs. Introduced Noise (Clustered)

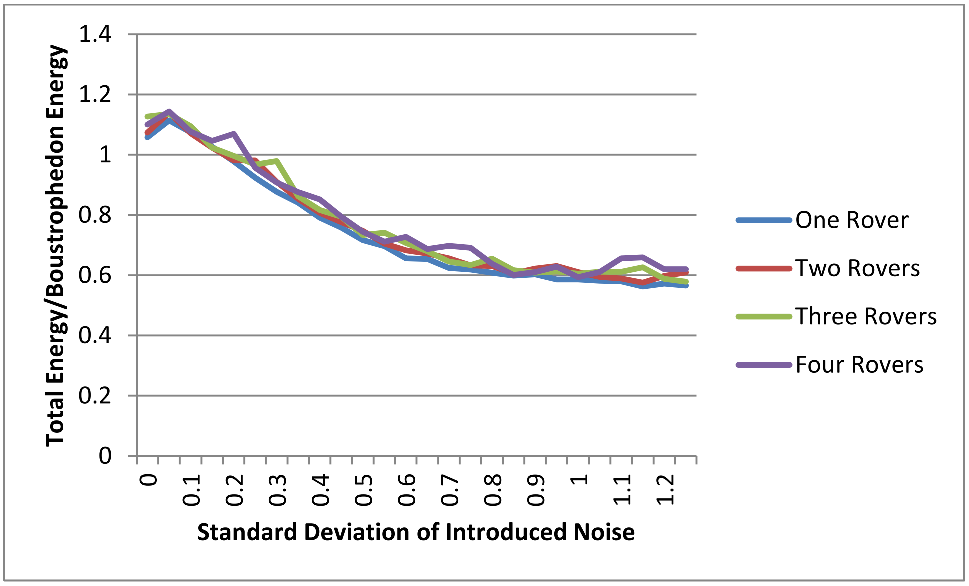


Figure 5.15: Ratio of Energy Consumption vs. Introduced Noise (Dispersed)

Figures 5.14 and 5.15 show the energy consumption of our system over the energy consumption of the lowest cost Boustrophedon path as a function of the standard deviation of the introduced noise for both clustered and dispersed starting points. Our system requires more energy than the best case Boustrophedon path when all energy values are equal, and this ratio increases with the number of rovers in use. When we begin to introduce noise to the map, the ratio initially increases, again with the maximum value increasing as more rovers are introduced. This ratio steadily falls as the standard deviation is increased. Our system once again becomes more efficient when the standard deviation exceeds roughly 0.2. Once the standard deviation of the introduced noise exceeds approximately 0.7, the ratio between energy consumed by our system and the lowest cost boustrophedon path stabilizes. There is a slight loss in efficiency for each additional rover used. However, even with four rovers in use, the energy consumption ratio still stabilizes around 0.60. The average number of total moves was 11,854 for two rovers, 12,063 for three rovers and 12,153 for four rovers.

5.5.4 Multiple Rovers on Map with Null Values

The same map shown in Figure 5.6 was used for this series of tests. Tests were run for groups of two, three and four rovers. A variety of clustered and dispersed starting points were selected. Results are summarized in Tables 5.3, 5.4 and 5.5. With the exception of the three rover case with a bottom left clustered starting location, the total number of moves and average energy consumption per move remained relatively stable.

Two Rovers					
Starting Points	**Total Moves**	**Max # Visits**	**# with > 5 Visits**	**Total Energy Consumption**	**Cost per Move**
(330,330),(330,331)	200638	20	3404	113560	0.565994478
(25,25),(25,26)	186616	15	1474	101070	0.541593433
(25,375),(25,376)	180226	13	979	93324	0.517816519
(375,25),(375,26)	183096	12	936	97979	0.535123651
(200,200),(200,201)	186814	17	2061	104250	0.558041689
(25,25),(25,375)	180816	17	1154	96377	0.533011459

Table 5.3: Results for Two Rovers on Map with Null Values

Three Rovers					
Starting Points	**Total Moves**	**Max # Visits**	**# with > 5 Visits**	**Total Energy Consumption**	**Cost per Move**
(330,330),(330,331),(331,330)	191691	18	2403	107280	0.559650688
(25,25),(25,26),(26,25)	186630	23	1534	100670	0.539409527
(25,375),(25,376),(26,375)	188157	20	2178	103660	0.550922899
(375,25),(375,26),(376,25)	238539	24	6971	169890	0.712210582
(200,200),(200,201),(201,200)	189891	14	2287	105520	0.55568721
(25,25),(25,375),(375,25)	188637	14	2419	102710	0.54448491

Table 5.4: Results for Three Rovers on Map with Null Values

Four Rovers					
Starting Points	**Total Moves**	**Max # Visits**	**# with > 5 Visits**	**Total Energy Consumption**	**Cost per Move**
(330,330),(330,331),(331,330),(331,331)	179248	11	234	93004	0.518856556
(25,25),(25,26),(26,25),(26,26)	199048	27	2144	114660	0.57604196
(25,375),(25,376),(26,375),(26,376)	194020	23	2225	109540	0.564580971
(375,25),(375,26),(376,25),(376,26)	184600	16	1076	99991	0.541663055
(200,200),(200,201),(201,200),(201,201)	188928	14	1836	103510	0.547880674
(25,25),(25,375),(375,25),(330,330)	183880	21	960	97630	0.530944094

Table 5.5: Results for Four Rovers on Map with Null Values

Figures 5.16 through 5.21 show number of times each point was visited during each test, with the red points indicating rover starting locations for that test. Figure 5.12 illustrates the number and location of input data null values. The data displayed in the maps makes evident that high numbers of repeat visits occur only when the density of null values are high.

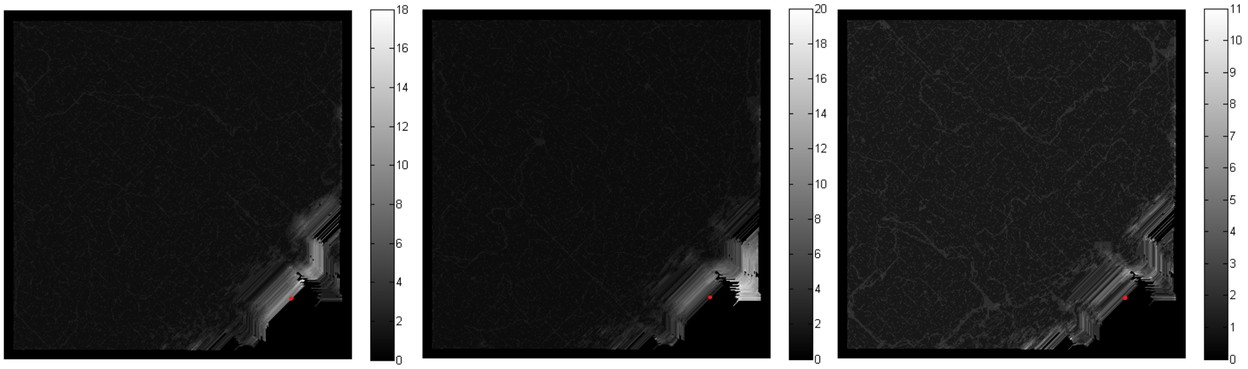


Figure 5.16: Density of Visits for Starting Points {(330,330), (330,331)}. {(330,330),(330,331),(331,330)} and {(330,330),(330,331),(331,330),(331,331)}

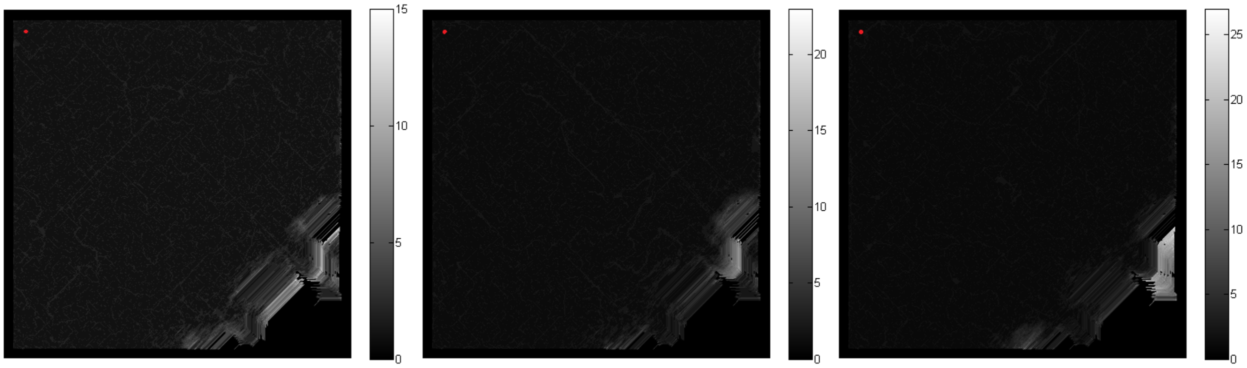


Figure 5.17: Density of Visits for Starting Points {(25,25), (25,26)}. {(25,25),(25,26),(26,25)} and {(25,25),(25,26),(26,25),(26,26)}

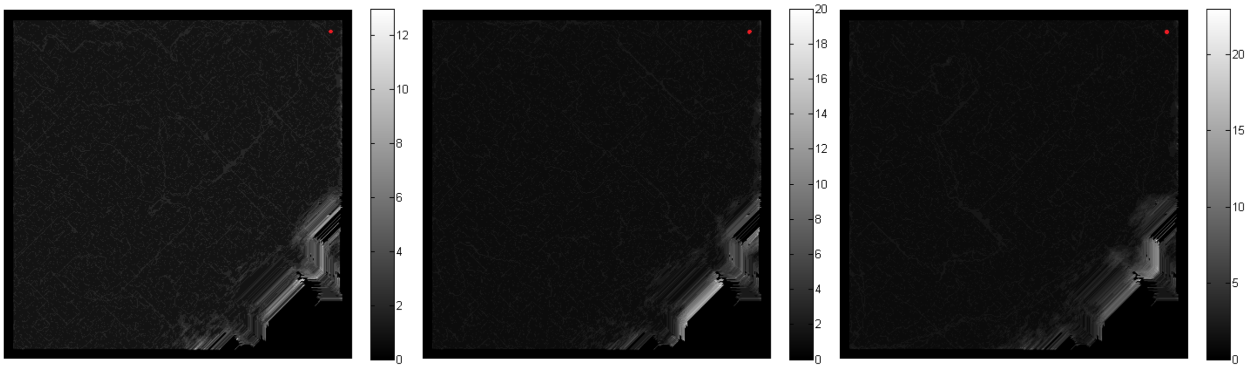


Figure 5.18: Density of Visits for Starting Points {(25,375), (25,376)}. {(25,375),(25,376),(26,375)} and {(25,375),(25,376),(26,375),(26,376)}

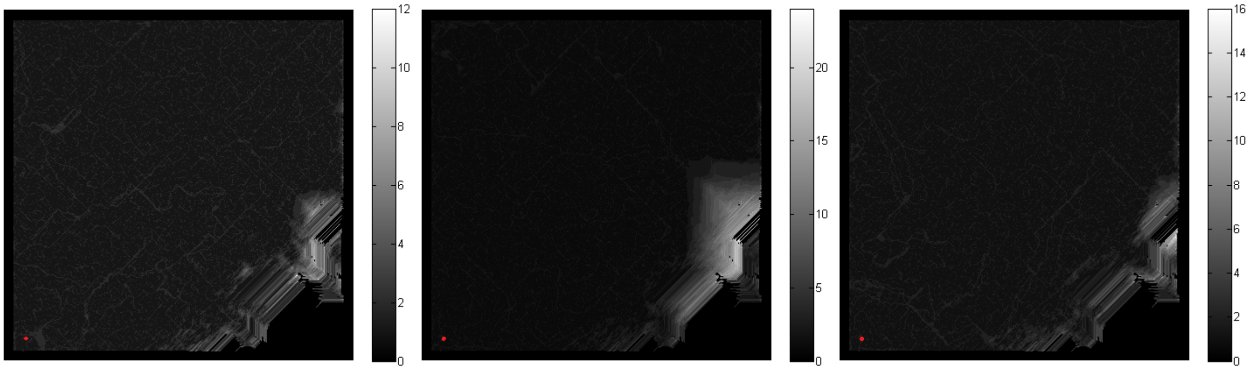


Figure 5.19: Density of Visits for Starting Points {(375,25), (375,26)}. {(375,25),(375,26),(376,25)} and {(375,25),(375,26),(376,25),(376,26)}

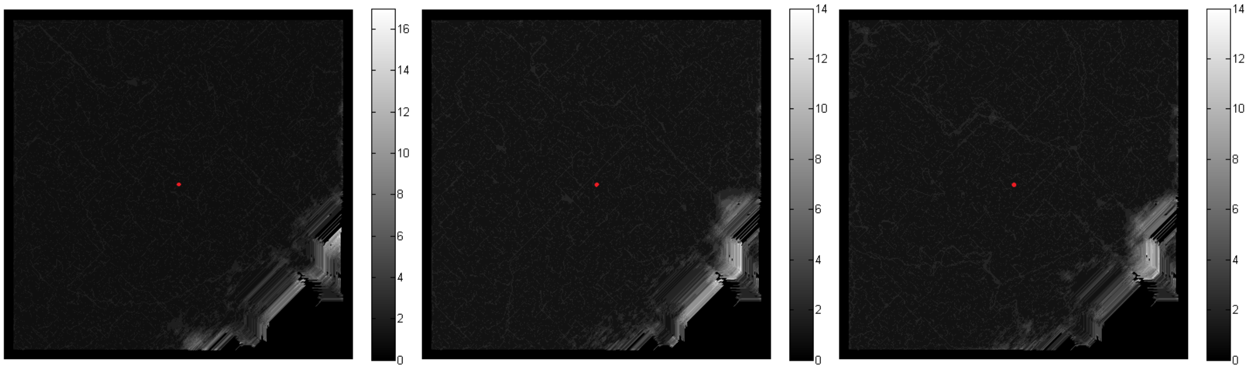


Figure 5.20: Density of Visits for Starting Points {(200,200), (200,201)}. {(200,200),(200,201),(201,200)} and {(200,200),(200,201),(201,200),(201,201)}

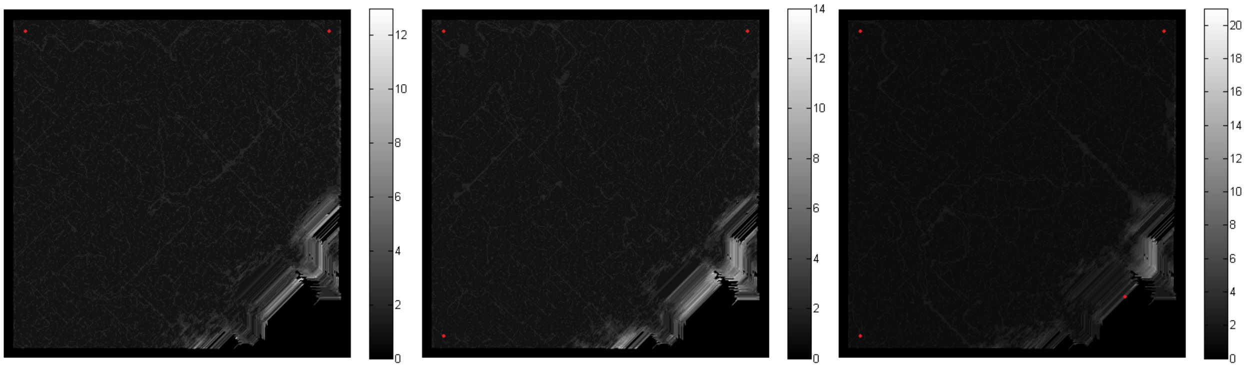


Figure 5.21: Density of Visits for Starting Points {(25,25), (25,375)}. {(25,25),(25,375),(375,25)} and {(25,25),(25,375),(375,25),(330,330)}

6.0 Terrain Classification

This chapter details the design of a neural network terrain classifier which classifies all traversable terrain as one of three soil types, and then assigns an assumed set of Bekker soil parameters based on existing knowledge of the Martian surface.

6.1 Sensitivity Analysis

In order to most effectively assign generic soil parameters, it was decided that an analysis of the relationships detailed by Shibly's equations would first be undertaken. This would allow for a better understanding of which parameters most greatly affect the energy costs of terrain traversal.

The surfaces in Figures 6.1-6.12 show torque plotted as a function of wheel sinkage and slip at the high and low end of acceptable values for soil cohesion and modulus of shear deformation for Shibly's equations. Surfaces were also plotted for high, intermediate and low value end values of the angle of internal shearing resistance. Wheel parameters where kept constant throughout, with width equal to 0.15 m, a radius of 0.2 m and a wheel load of 0.02 kN.

The most generally observable trend is that after a slight initial decrease, torque requirements increase exponentially as slip and sinkage increase. Modulus of shear deformation has a large impact on torque, with a higher k value allowing for much more wheel sinkage before T values start to increase exponentially. Increased soil cohesion results in a substantial increase in T values, especially when slip and sinkage are high. Increasing the angle of internal shearing resistance has a similar but far less pronounced effect.

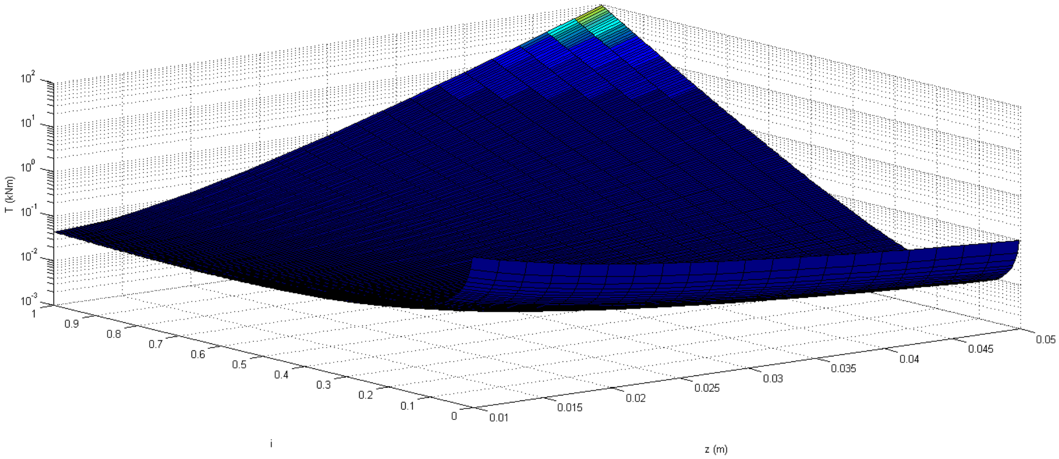


Figure 6.1: Wheel Torque as a Function of Slip and Sinkage, c=0.0 kPa phi=20° k=0.01 m w=0.02 kNm r=0.2 m b=0.15 m

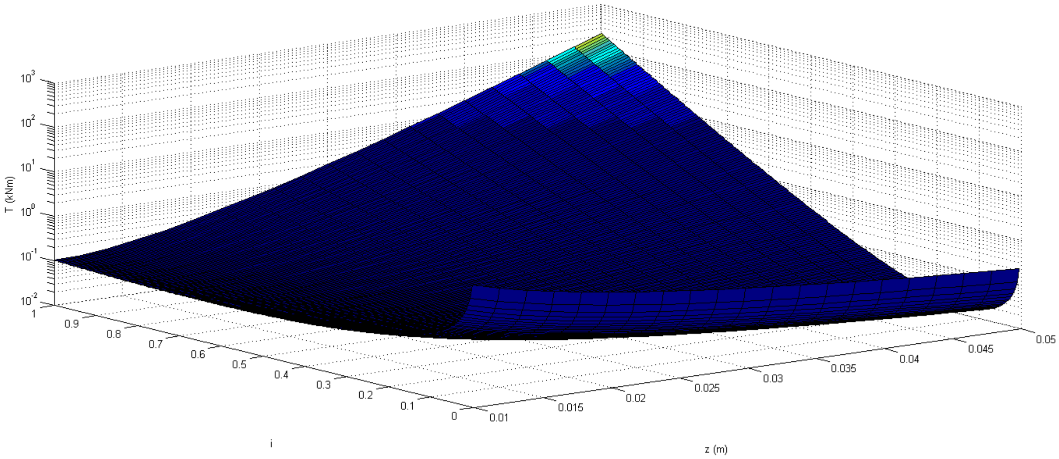


Figure 6.2: Wheel Torque as a Function of Slip and Sinkage, c=0.0 kPa phi=40° k=0.01 m

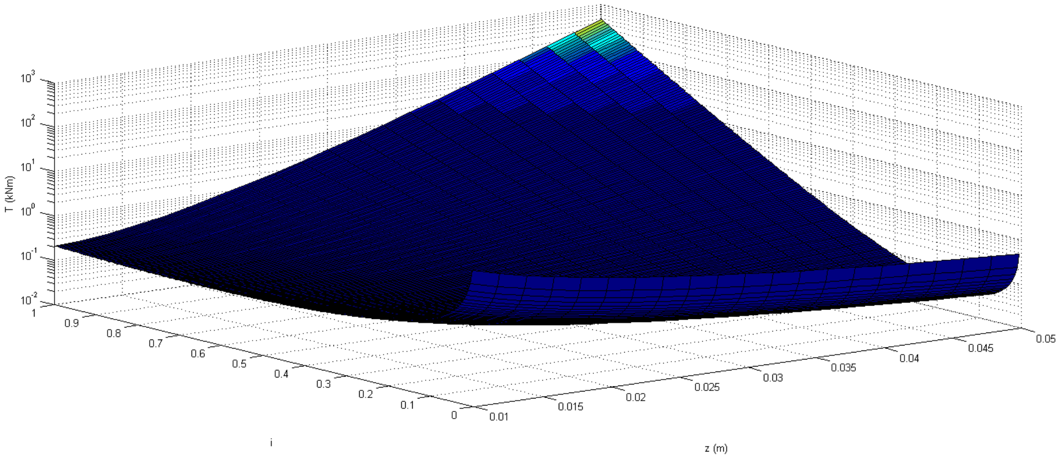


Figure 6.3: Wheel Torque as a Function of Slip and Sinkage, c=0.0 kPa phi=60° k=0.01 m

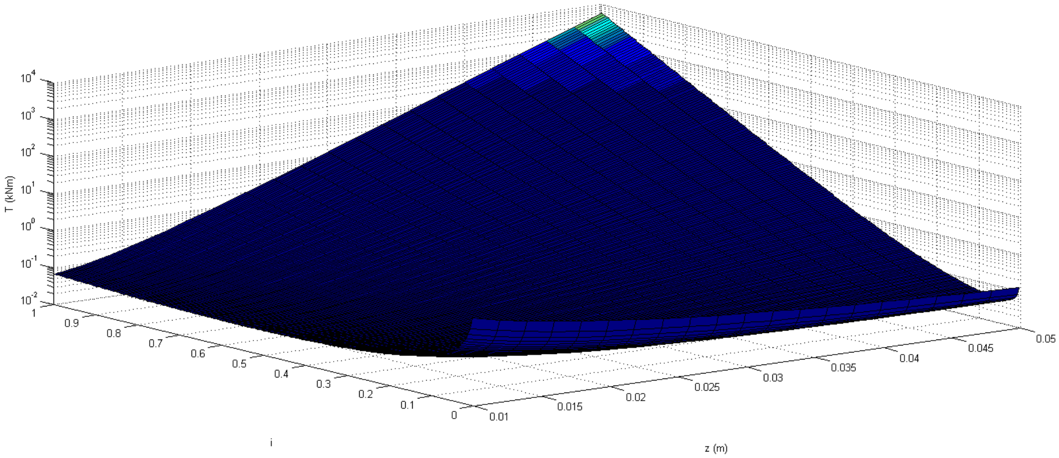


Figure 6.4: Wheel Torque as a Function of Slip and Sinkage, c=3.0 kPa phi=20° k=0.01 m

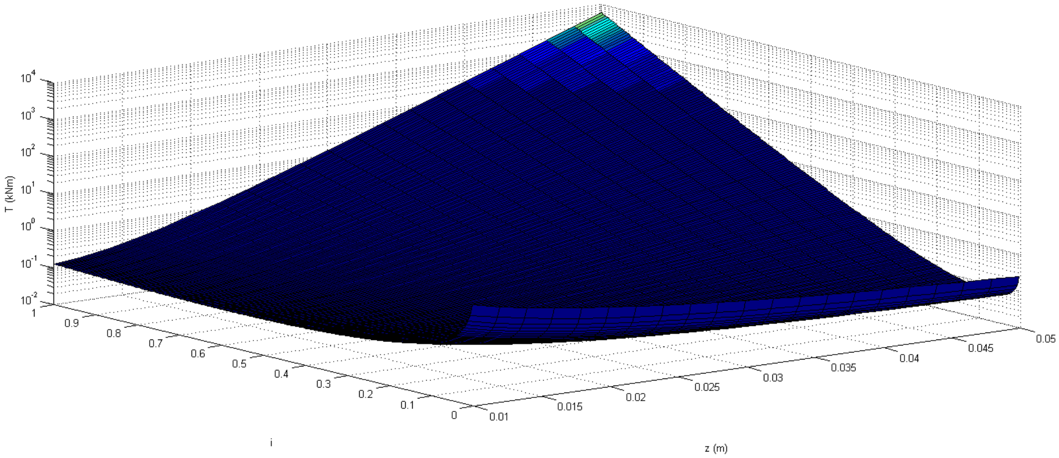


Figure 6.5: Wheel Torque as a Function of Slip and Sinkage, c=3.0 kPa phi=40° k=0.01 m

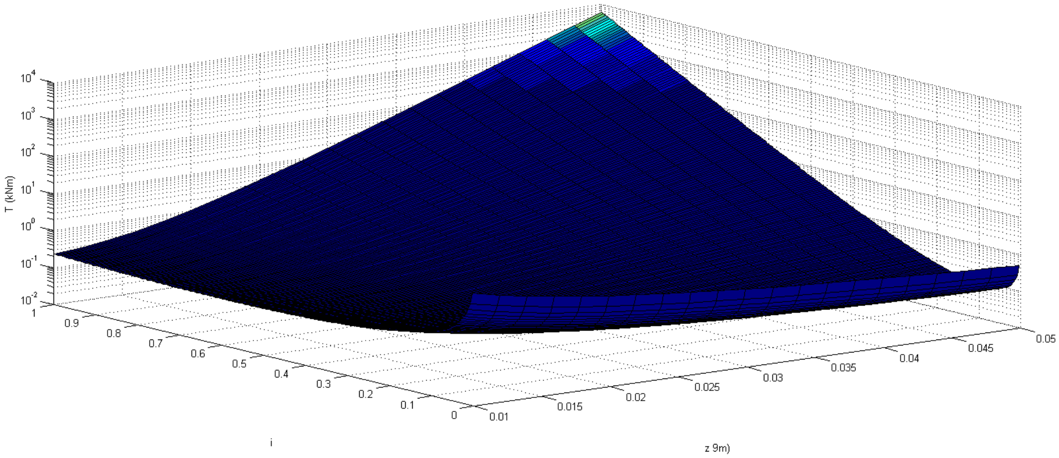


Figure 6.6: Wheel Torque as a Function of Slip and Sinkage, c=3.0 kPa phi=60° k=0.01 m

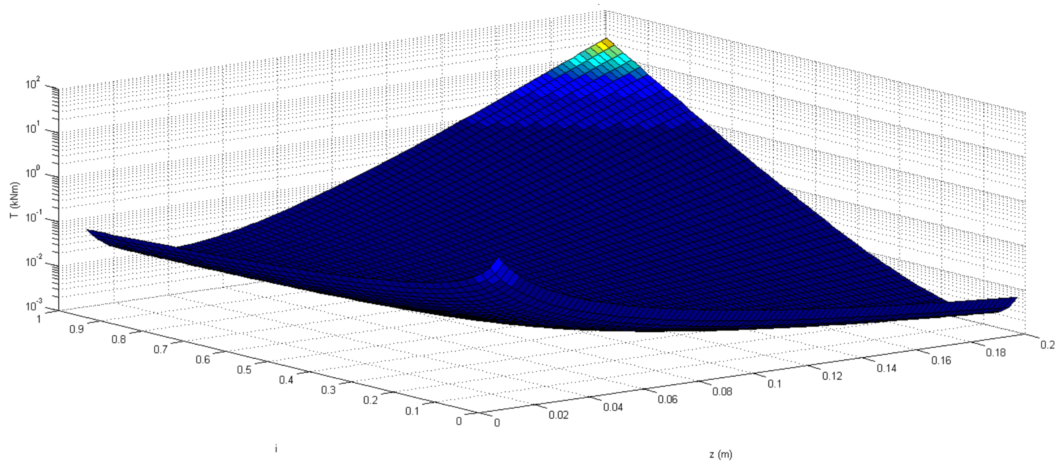


Figure 6.7: Wheel Torque as a Function of Slip and Sinkage, c=0.0 kPa phi=20° k=0.04 m

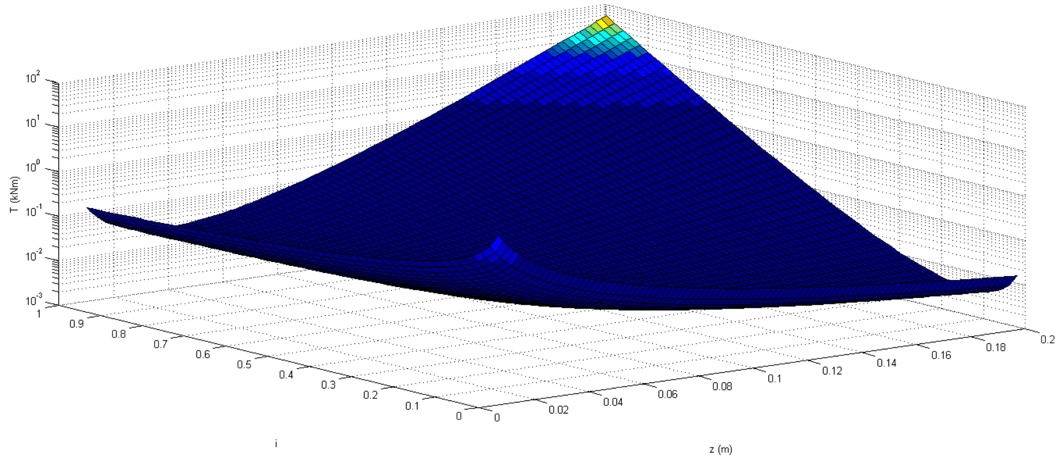


Figure 6.8: Wheel Torque as a Function of Slip and Sinkage, c=0.0 kPa phi=40° k=0.04 m

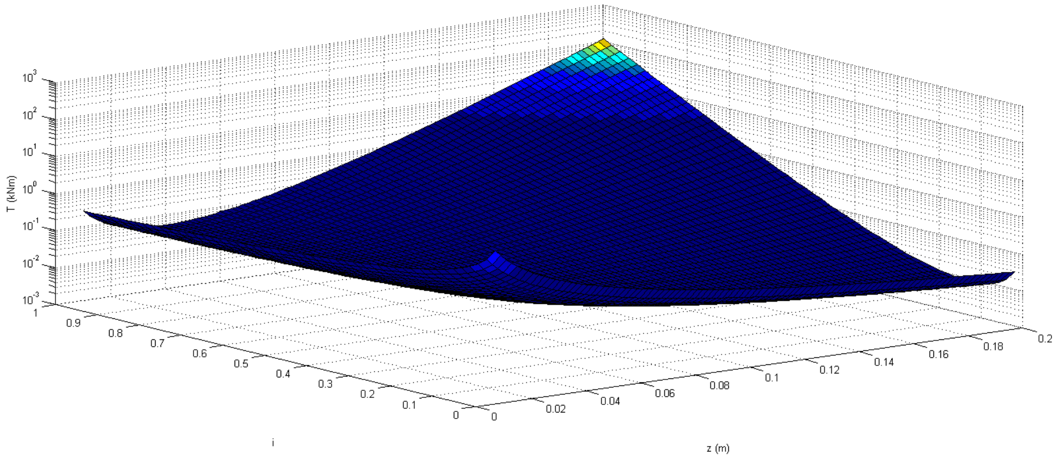


Figure 6.9: Wheel Torque as a Function of Slip and Sinkage, c=0.0 kPa phi=60° k=0.04 m

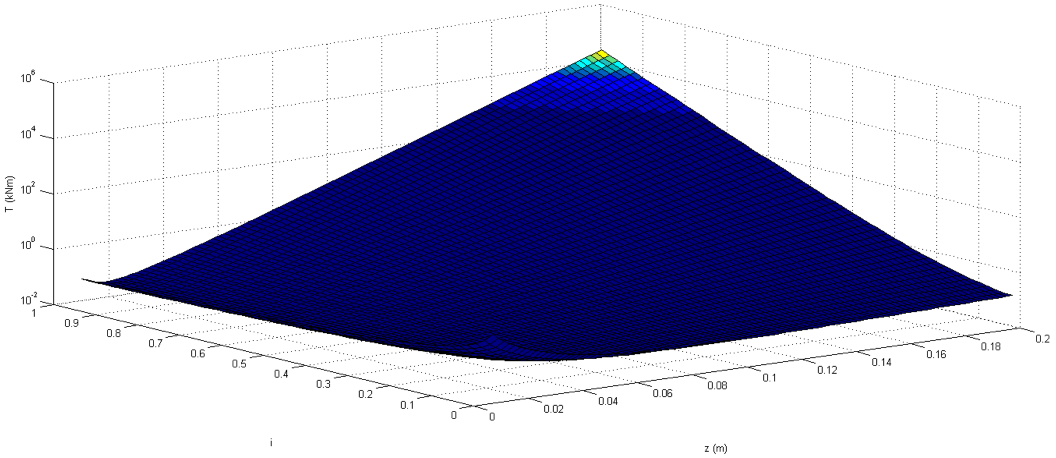


Figure 6.10: Wheel Torque as a Function of Slip and Sinkage, c=3.0 kPa phi=20° k=0.04 m

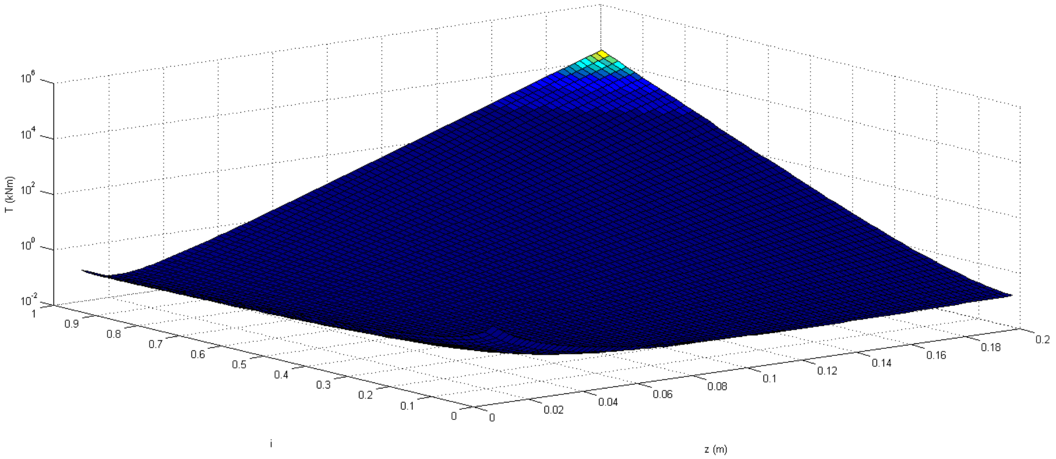


Figure 6.11: Wheel Torque as a Function of Slip and Sinkage, c=3.0 kPa phi=40° k=0.04 m

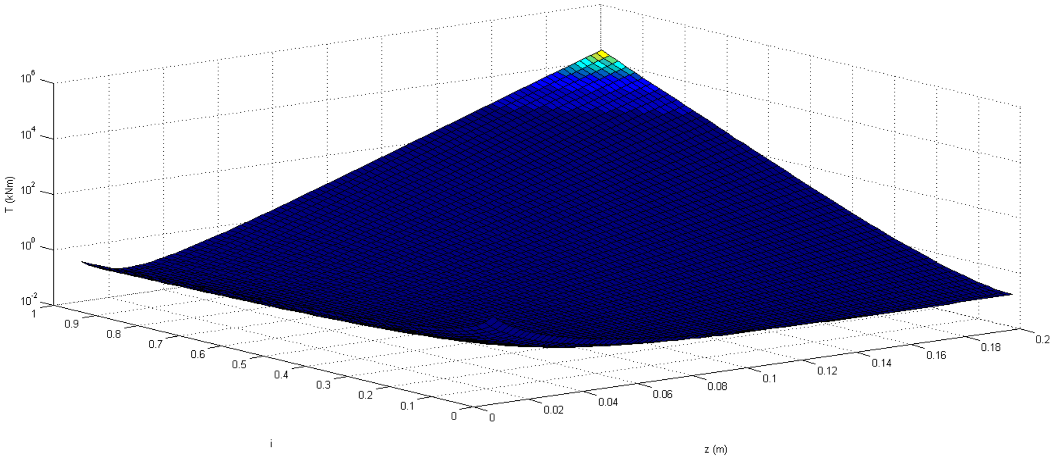
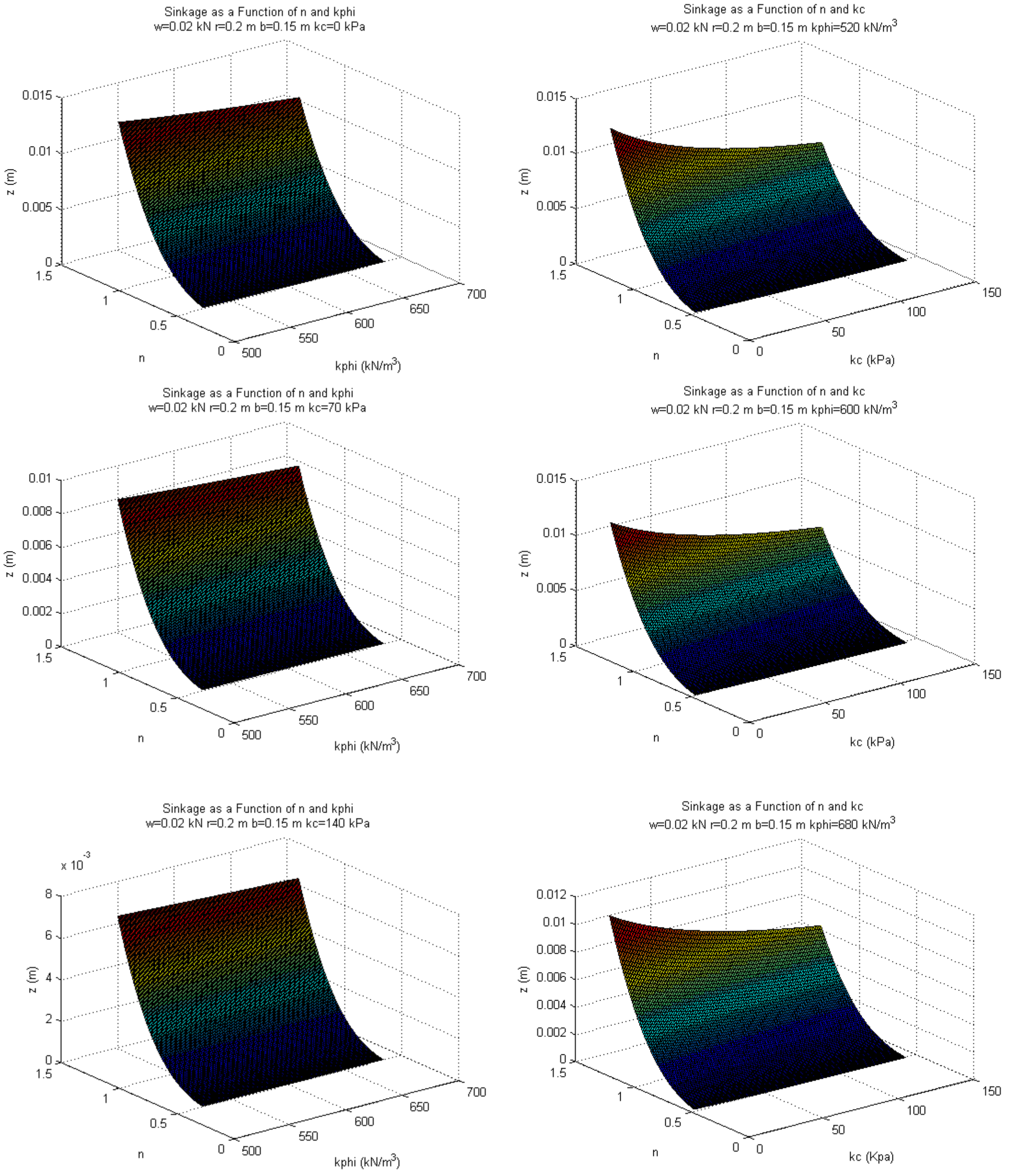


Figure 6.12: Wheel Torque as a Function of Slip and Sinkage, c=3.0 kPa phi=60° k=0.04 m

Figure 6.13 illustrates the relationship between sinkage, exponent of sinkage, soil cohesive modulus of deformation and soil shear modulus of deformation over the range of acceptable values defined by Shibly. From the plotted surfaces, it is clear that sinkage is largely determined by n, and that k_c and k_φ are relatively unimportant.


Sinkage as a Function of n and kphi
w=0.02 kN r=0.2 m b=0.15 m kc=0 kPa

Sinkage as a Function of n and kc
w=0.02 kN r=0.2 m b=0.15 m kphi=520 kN/m^3

Sinkage as a Function of n and kphi
w=0.02 kN r=0.2 m b=0.15 m kc=70 kPa

Sinkage as a Function of n and kc
w=0.02 kN r=0.2 m b=0.15 m kphi=600 kN/m^3

Sinkage as a Function of n and kphi
w=0.02 kN r=0.2 m b=0.15 m kc=140 kPa

Sinkage as a Function of n and kc
w=0.02 kN r=0.2 m b=0.15 m kphi=680 kN/m^3

Figure 6.13: Wheel Sinkage as a Function of n, k_c and k_φ

As can be seen in Table 4.2, the soil parameters defined by Bekker theory are difficult to measure, with the values varying considerably between highly similar samples. Consequently, it was determined that the most practical solution would be to base traversability assessments entirely on DEM data. The remaining terrain could then be lumped into one several generic soil types and assigned roughly representative Bekker parameters.

Measurements of Martian soil parameters have been taken by both the Viking landers in the 1970s and the Sojourner rover in the 1990s [169]. Additional information was also collected by the MER robots Spirit and Opportunity in the mid 2000s [170]. Combining the available Martian data with information provided by Shibly and Wong, the three soil types in table 6.1 were chosen and assigned the parameters shown.

	Sand	Crusty	Rocky
φ(°)	34	37	31
c (kPa)	1.2	0.8	3
n	1.1	1	0.7
k (m)	0.025	0.025	0.02

Table 6.1: Soil Types and Selected Properties

As the sensitivity analysis of the Bekker pressure/sinkage relationship in the previous section showed, k_c and k_φ have a substantially lower impact on sinkage than n. Given their limited effect, they were kept constant for all three soil types at k_c = 5 kN/m^{n+1} and k_φ = 680 kN/m^{n+2}.

6.2 Terrain Classifier

The procedure for classification of terrain imagery was similar to that used by Sung et al [130]. Colour images were pre-processed, a wavelet transform of each image was then taken, and neural network classifier was then used to label terrain patches was one of the three previously described soil types.

6.2.1 Pre-processing

Colour images were first converted from RGB to HSV in order to allow for the separation of luminance from colour information. This was done by normalizing the “V” portion of each pixel using equation 6.1.

$$V(i,j)_{new} = M_{desired} + \frac{\sigma_{desired}(V(i,j)-M_{current})}{\sigma_{current}} \tag{6.1}$$

$M_{desired}$ and $\sigma_{desired}$ are the sought after mean and standard deviation for the V values of the image and $M_{current}$ and $\sigma_{current}$ are the existing properties. While Sung et al [130] calculated values for $M_{desired}$ and $\sigma_{desired}$ which gave them the best results through trial and error, it was determined that results were acceptable so long as the same values were used for every image normalization. Consequently, M and σ were calculated for a representative image (0.4 and 0.085 respectively), and these values were used when processing every subsequent image.

6.2.2 Wavelet Transform

Once images had been normalized, a two level Daubechies wavelet transform was used to extract a feature vector from each image. Wavelet analysis has been proven effective at analyzing localized portions of larger signals [171] and was demonstrated by Sung et al to be an effective tool for the classification of segments of colour images [130]. Figure 6.14 shows an image of a two level Daubechies wavelet transform of typical terrain image with adjacent band details. B_1 and B_4 represent the horizontal sub-band images, B_2 and B_5 the vertical and B_3 and B_6 the diagonal.

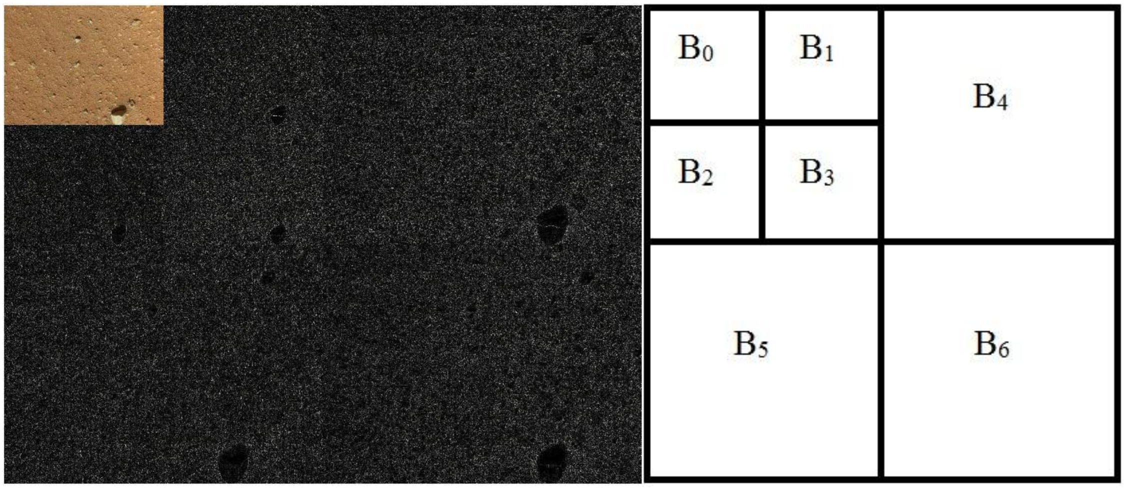


Figure 6.14: Two level Daubechies Wavelet Transform

6.2.3 Feature Extraction

In a similar manner to Sung, the feature vector was defined as the following:

$$F = [0.1B_0^{mean}\ B_4^{mean}\ B_5^{mean}\ B_0^{energy}\ B_1^{energy}\ B_2^{energy}\ B_4^{energy}\ B_5^{energy}]$$

Where B_i^{mean} is the mean value of each channel (H, S and V) for band i, and B_i^{energy} is the percentage of the total channel energy from all bands contained by band *i*. The calculation of B_i^{energy} is illustrated in equations 6.2 and 6.3.

$$E_{total} = \sum_{i=0}^{5} \sum_{j=0}^{N-1} (c_j^i)^2 \tag{6.2}$$

$$B_i^{energy} = \frac{100}{E_{total}} \sum_{j=0}^{N-1} (c_j^i)^2 \tag{6.3}$$

Where c_j^i represents the *jth* value of H, S and V for band *i*, and *N* represents the total number of values. The diagonal band information and first level horizontal and vertical means are omitted to minimize noise, while B_0 is scaled by 0.1 to reduce bias during classification. This means that feature vector *F* is a 1*x*24 vector.

6.2.4 Classification

Classification of imagery was performed by an Artificial Neural Network (ANN) created using the pattern recognition tool in Matlab's neural network toolbox. Matlab defines a pattern recognition network as having two layers. However, the selection of the number of neurons in the hidden layer is an inexact science at best. A number of different networks were tested using a total of 1050 16x16 pixel representative image chips (350 for each soil type) for training, validation and testing. Of the 1050 samples, 892 were used for training, 105 for validation and 53 for testing. Networks tested had the number of hidden neurons ranging from 16 to 40, with the best results being achieved with 20 hidden neurons. Figure 6.15 illustrates the network structure, while Figure 6.16 and Table 6.2 detail network performance during training.

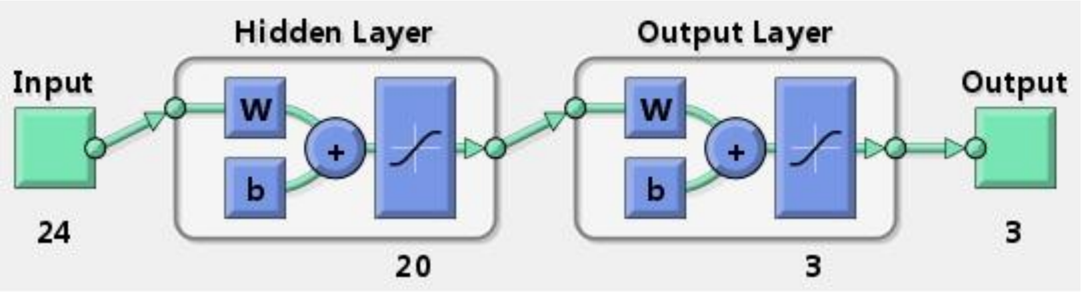


Figure 6.15: Neural Network Structure

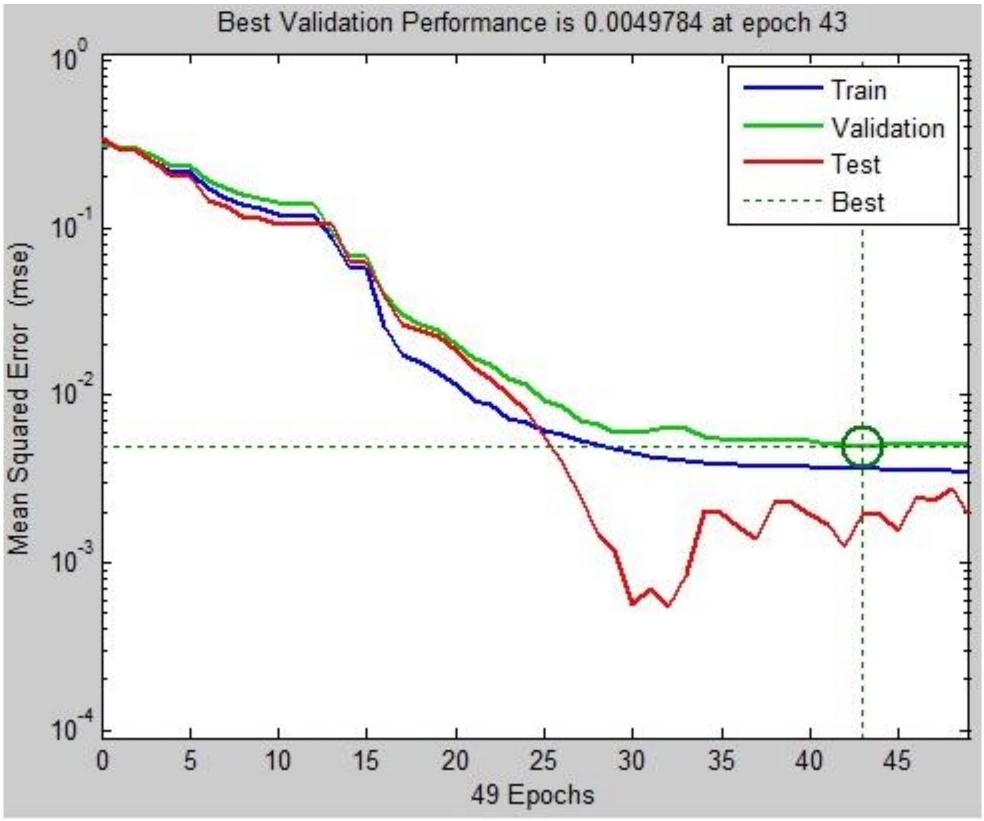


Figure 6.16: Neural Network Training Performance

	Samples	MSE	% Error
Training	892	0.00361	0.56
Validation	105	0.00498	0.95
Testing	53	0.00195	0

Table 6.2 Neural Network Performance

6.3 Results

The neural network classifier previously described was then used on a number of different images. Each image was first converted from RGB to HSV format, allowing its V value to be normalized in the same manner as the training, validation and testing sets for the neural network. Each image was then divided into 16x16 pixel sub-images. Each of these sub-images was then analyzed to determine its feature vector *F*, allowing the classifier to label each sub-image as one of the three previously defined terrain types.

In order to assess the effectiveness of the classifier, each image chip was hand annotated as one of the three soil types, and qualitatively assessed be at one of three distances (near, medium or far) in order so better grasp the outcome. Results are summarized in Table 6.3. Figures 6.17 through 6.19 display each test image adjacent to its classified version, with the third image representing the ground truth. The classified images are colour coded; red shaded terrain has been labeled gravel, green crusty soil and blue sand. The black areas to the right and bottom of some images are areas that were not processed as a result of the system analyzing 16x16 pixel blocks.

	Identified	Correct	% Correct
Gravel	963	812	84.3
Crusty Soil	1196	962	80.4
Sand	495	453	91.5
Total	2654	2227	83.9

Table 6.3: Summary of Classification Results

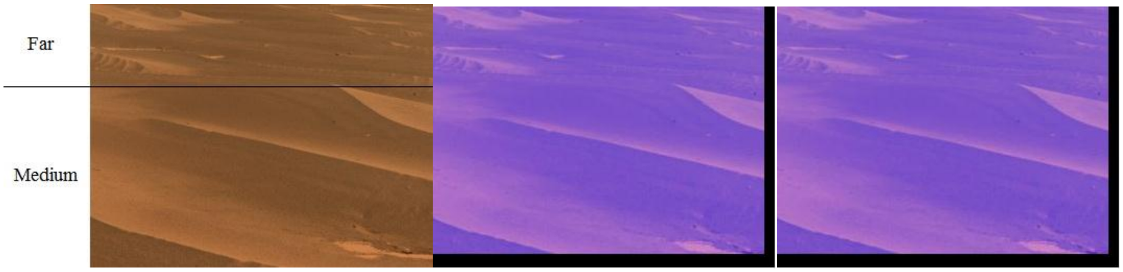


Figure 6.17: Soil Classification Test 1

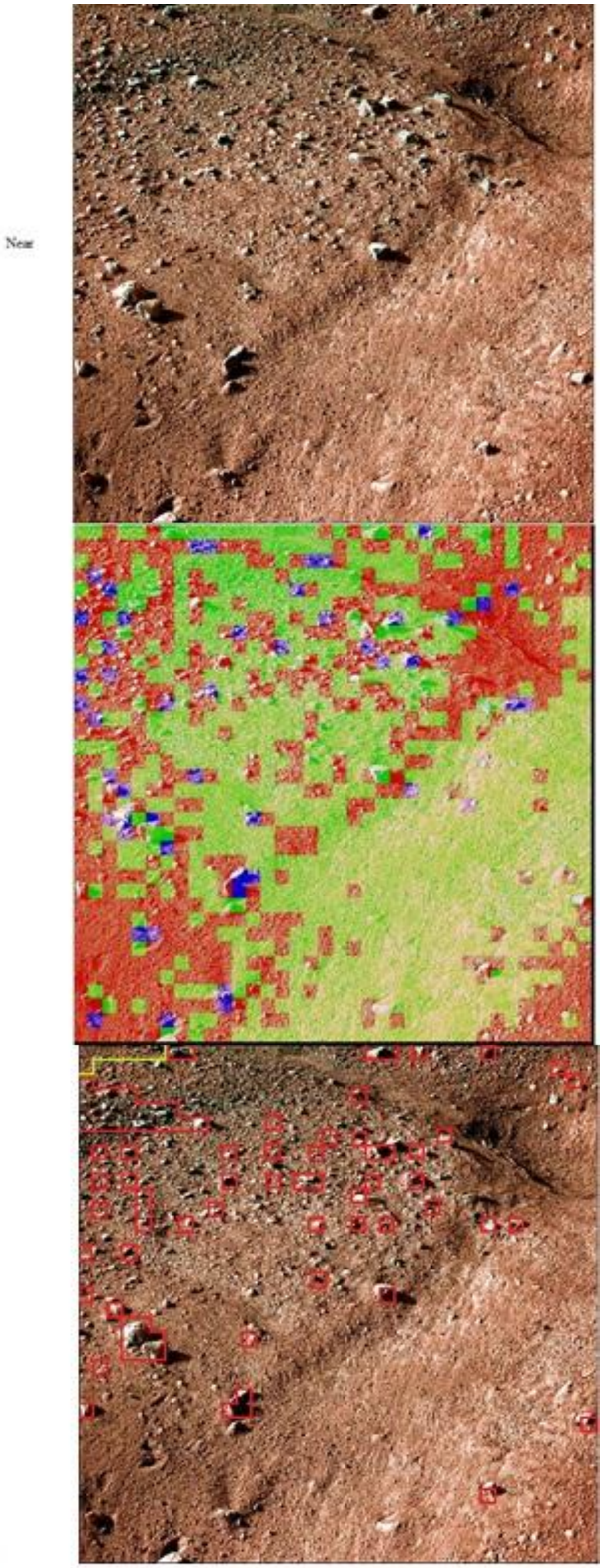


Figure 6.18: Soil Classification Test 2

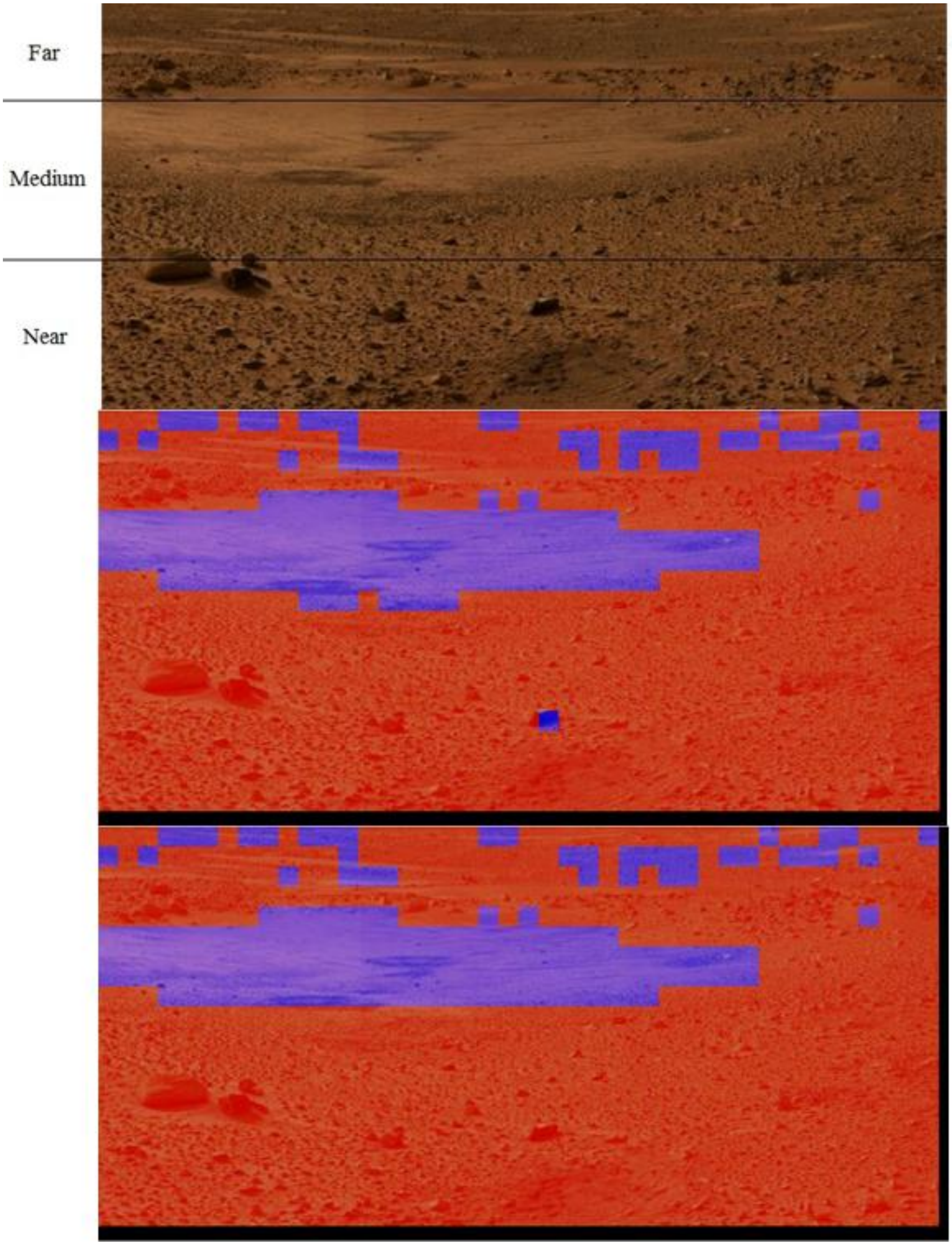


Figure 6.19: Soil Classification Test 3

From the data displayed in Table 6.3, it is clear that the neural network does a relatively good job of identifying the three different soil types. Analysis of the test images shows some difficulty distinguishing between large smooth rock formations and sand at short distances. Although the overall efficiency of the system is less than ideal, it is our belief that it could become viable by registering multiple images of the same terrain and then using a fuzzy classifier for the output produced for each image. This would also allow for a better classification of heterogeneous image chips.

7.0 Conclusion

This work details the design and testing of a system of terrain classification, relative energy estimation and low energy path planning for teams of lower complexity robots performing basic science, mapping and resource assessments of wide areas of the Lunar or Martian surface.

7.1 Future Work

Given additional time and resources there are a number of areas of this project that could be further advanced.

Additional testing of the half rocker-bogie could be used to tune the energy consumption model to yield more accurate numerical results. Additionally, tests could be run powering all three wheels to determine if differing ground contact angles negatively affect Drawbar Pull when the rover is in motion.

While the path planner was successfully able to plan paths of complete coverage in all test cases, it has not been validated mathematically. In future, efforts could be undertaken to prove that it is always successful if not optimal.

The terrain classifier also could be more rigorously tested. By having groups of lay people individually hand annotate grayscale terrain images as one of the three defined soil types, testing bias could be substantially reduced. This could potentially yield greater accuracy in numerical values obtained for the effectiveness of the neural network for the different soil types and ranges tested. There is also the potential to integrate elevation data into the classification process to further refine results, as well as incorporating multiple registered images of the same terrain to further augment results.

7.2 Conclusions

The staged development of a system for low level mapping, science and resource assaying by a team of rovers was described herein. While some of the required underlying technologies are immature (SCPA, LiDAR), all are being rapidly developed specifically for space exploration and should be available in the near future.

The main contributions made by this research are as follows:

- A simplified rocker-bogie model
- A wheel/soil interaction model based on Shibly's modified Bekker equations
- A complete coverage path planning scheme which minimizes energy consumption
- A sensitivity analysis of Shibly's modified Bekker equations
- A wavelet-based neural network terrain classifier

The system as a whole works by providing colour terrain images to the neural network classifier. The classifier then labels terrain patches as one of three predetermined soil types and assigns each patch a set of representative terrain parameters. The rocker-bogie and wheel/soil interaction models then work in concert to use these terrain parameters along with rover configuration to create energy consumption maps describing the cost of movement from every point in every direction. These maps are then provided to the path planner which plans low energy paths of complete coverage for however many rovers are in use.

This work has presented a system which successfully classifies a large percentage of terrain imagery into one of three soil types, assesses the energy requirements of terrain traversal for these soil types and plans efficient paths of complete coverage for the imaged area. While are further efforts that can be made in all areas, the work has largely achieved its stated goals.

Bibliography

[1] Fong et al, “Human-Robot Site Survey and Sampling for Space Exploration,” *AIAA Space 2006 Conference*, 2006.

[2] T. Fong, M. Deans, P. Lee and M. Bualat, “Simulated Lunar Robotic Survey at Terrestrial Analog Sites,” *Lunar and Planetary Science Conference*, Abstract 1487, Houston, TX, 2007.

[3] T. Fong et al, “Analog Lunar Robotic Site Survey at Haughton Crater,” *LEAG Workshop on Enabling Exploration: The Lunar Outpost and Beyond*, Houston, 2007.

[4] T. Fong et al, “Robotic Site Survey at Haughton Crater,” *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, 2008.

[5] D. Schreckenghost et al, “Human Supervision of Robotic Site Surveys,” *Conference on Human/Robotic Technology and the Vision for Space Exploration (STAIF)*, 2008.

[6] T. Fong et al, “Field Testing of Utility Robots for Lunar Surface Operations,” *AIAA Space 2008 Conference*, 2008.

[7] V. Verma, V. Baskaran, H. Utz, R. Harris and C. Fry, “Demonstration of Robust Execution on a NASA Lunar Rover Testbed,” *International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*, 2008.

[8] T. Estlin, D. Gaines, F. Fisher and R. Castano, “Coordinating Multiple Rovers with Interdependent Science Objectives,” *Fourth International Joint Conference on Autonomous Agents and Multi Agent Systems*, Utrecht, the Netherlands, July 2005.

[9] M.G. Bekker, *Introduction to Terrain-Vehicle Systems*, Ann Arbor, Michigan: University of Michigan Press, 1969.

[10] H. Shibly, K. Iagnemma and S. Dubowsky, "An Equivalent Soil Mechanics Formulation for Rigid Wheels in Deformable Terrain, with Application to Planetary Exploration Rovers," *Journal of Terramechanics*, vol. 42, no. 1, 2005.

[11] F. Ben Amar and P. Bidaud, "Dynamic Analysis of Off-Road Vehicles," in *Proc. 4th Int. Symp. on Experimental Robotics*, p. 363-371, 1995.

[12] C. Grand, F. Ben Amar, F. Plumet and P. Bidaud, "Simulation and Control of High Mobility Rovers for Rough Terrains Exploration," in *Proc. IARP Int. Workshop on Humanitarian Demining*, p. 51-56, Vienna, Austria, 2002.

[13] N. Patel, A. Ellery, E. Allouis, M. Sweeting and L. Richter, "Rover Mobility Performance Evaluation Tool (RMPET): A Systematic Tool for Rover Chassis Evaluation via Application of Bekker Theory," in *Proc. 8th ESA Workshop on Advanced Space Technologies for Robotics and Automation*, p. 251-258, Noordwijk, The Netherlands, 2004.

[14] R. Bauer, W. Leung and T. Barfoot, "Development of a Dynamic Simulation Tool for the Exomars Rover," in *Proc. of 8th Int. Symp. on Ariticial Intelligence, Robotics and Automation in Space*, Munich, Germany, 2005.

[15] L. Ding, K. Nagatani, K. Sato, A. Mora, K. Yoshida, H.Gao and Z. Deng, "Terramechanics-based High-fidelity Dynamics Simulation for Wheeled Mobile

Robot on Deformable Rough Terrain," in *Proc. IEEE Int. Conf. on Robotics and Automation*, p. 4922-4927, Anchorage, Alaska, 2010.

[16] R.A. Lindemann and C.J. Voorhees, "Mars Exploration Rover Mobility Assembly Design, Test and Performance," in *Proc. IEEE Int. Conf. on Systems, Man and Cybernetics*, vol. 1, p. 450-455, 2005.

[17] H. Hacot, *The Kinematic Analysis and Motion Control of a Planetary Rover,* Masters Thesis, Department of Mechanical Engineering, MIT, Cambridge, Massachusetts, 1998.

[18] H. Hacot, S. Dubowsky and P. Bidaud, "Analysis and Simulation of a Rocker-Bogie Exploration Rover," in *Proc. Twelfth CISM-IFToMM Symp.*, Paris, France, 1998.

[19] T.G. Gang and S.Y. Yi, "Kinematic Modeling of Mobile Robot with Rocker-Bogie Link Structure," in *Proc. ICMIT: Control Systems and Robotics*, 2005.

[20] A. Zelinsky, R. A. Jarvis, J. C. Byarne and S. Yuta, "Planning Paths of Complete Coverage of an Unstructured Environment by a Mobile Robot," *International Conference on Advanced Robotics (lCAR)*, pp. 533-538, 1993.

[21] S. Wirth and J. Pellenz, "Exploration Transform: A stable Exploring Algorithm for Robots in Rescue Environments," *IEEE International Workshop on Safety, Security and Rescue Robotics*, Rome, Italy, September 2007.

[22] H. Choset, "Coverage of Known Spaces: The Boustrophedon Cellular Decomposition," *Autonomous Robots*, vol. 9, pp. 247–253, 2000.

[23] E. Garcia and P. Gonzalez De Santos, "Mobile-Robot Navigation with Complete Coverage of Unstructured Environments," *Robotics and Autonomous Systems*, vol. 46, iss. 4, pp. 195-204, 2004.

[24] E. U. Acar, H. Choset, Y. Zhang and M. Schervish, "Path Planning for Robotic Demining: Robust Sensor-Based Coverage of Unstructured Environments and Probabilistic Methods," *The International Journal of Robotics Research*, vol. 22, no. 7-8, pp 441-466, 2003.

[25] E. Acar and H. Choset, "Robust Sensor-Based Coverage of Unstructured Environments," *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2001.

[26] W. H. Wang, "Optimal Line-Sweep-Based Decompositions for Coverage Algorithms," *IEEE International Conference on Robotics and Automation*, Seoul, Korea, 21-25 May 2001.

[27] Z. Yao, "Finding Efficient Robot Path for the Complete Coverage of a Known Space," *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 9-15 October, Beijing, China, 2006.

[28] J. W. Kang, S. J. Kim, M. J. Chung, H. Myung, J. H. Park and S. W. Bang, "Path Planning for Complete and Efficient Coverage Operation of Mobile Robots," *IEEE International Conference on Mechatronics and Automation*, Harbin, China, 5-8 August 2007.

[29] E. U. Acar, H. Choset and P. N. Atkar, "Complete Sensor-Based Coverage with Extended-Range Detectors: A Hierarchical Decomposition in Terms of Critical Points and Voronoi Diagrams," *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 1305–1311, October 2001.

[30] S. S. Ge and C. Fua, “Complete Multi-Robot Coverage of Unknown Environments with Minimum Repeated Coverage,” *IEEE International Conference on Robotics and Automation (ICRA)*, Barcelona, Spain, April 2005.

[31] A. Pirzadeh and W. Snyder, “A Unified Solution to Coverage and Search in Explored and Unexplored Terrains Using Indirect Control,” *IEEE International Conference on Robotics and Automation (ICRA)*, 1990.

[32] S. Koenig, C. Tovey and W. Halliburton, “Greedy Mapping of Terrain,” *IEEE International Conference on Robotics and Automation (ICRA)*, Seoul, Korea, 21-26 May, 2001.

[33] G. Schmidt and C. Hofner, “An Advanced Planning and Navigation Approach for Autonomous Cleaning Robot Operations,” *IEEE International Conference on Intelligent Robots and Systems*, Victoria, B.C., Canada October 1998.

[34] R. Neumann de Carvlho, H. A. Vidal, P. Vieira and M. I. Riberio, “Complete Coverage Path Planning and Guidance for Cleaning Robots,” *IEEE International Symposium on Industrial Electronics*, pp. 677-682, Guimaraes, Portugal, 7-11 July 1997.

[35] J. Oh, Y. Choi, J. Park and Y. F. Zheng, “Complete Coverage Navigation of Cleaning Robots Using Triangular-Cell-Based Map,” *IEEE Transactions on Industrial Electronics*, vol. 51, iss. 3, pp. 718-726, 2004.

[36] S. Y. T. Lang and C. Bing-Yung , “Coordination of Behaviours for Mobile Robot Floor Cleaning,” *IEEE International Conference on Intelligent Robots and Systems (IROS)*, vol. 2, pp. 1236-1241, 1998.

[37] H. Zhang, W. Wang and W. Zhao , “A Topological Area Coverage Algorithm for Indoor Vacuuming Robot,” *IEEE International Conference on Automation and Logistics*, Jinan, China, 18-21 August 2007.

[38] X. Deng and C. H. Papadimitriou, “Exploring an Unknown Graph,” *IEEE Symposium on Foundations of Computer Science*, pp. 356-361, Los Alamitos, CA, 1990.

[39] S. Albers and M. R. Henzinger, “Exploring Unknown Environments,” *SIAM Journal on Computing*, vol. 29, no. 4, pp. 1164-1188, 2000.

[40] G. Dudek, M. Jenkin, E. Milios and D. Wilkes, “Robotic Exploration as Graph Construction,” *IEEE Transactions on Robotics and Automation*, vol. 7, iss. 6, pp. 859-865, 1991.

[41] Y. Gabriely and E. Rimon, “Spanning-Tree Based Coverage of Continuous Areas by a Mobile Robot,” *Annals of Mathematics and Artificial Intelligence*, vol. 31, pp. 77–98, 2001.

[42] E. González, O. Álvarez, Y. Díaz, C. Parra and C. Bustacara, “BSA: A complete Coverage Algorithm,” *IEEE International Conference on Robotics and Automation (ICRA)*, Barcelona, Spain, April 2005.

[43] L. Jiao and Z. Tang, “A Visibility-Based Algorithm for Multi-Robot Boundry Coverage,” *International Journal of Advanced Robotic Systems*, vol. 5, iss. 1, pp. 63-68, 2008.

[44] W. K. Lee, W. Duch and G. S. Ng, “Robot Space Exploration Using Peano Paths Generated by Self-Organizing Maps,” *International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 1-6, 5-8 December 2006.

[45] B. Yamauchi, "A Frontier-Based Approach for Autonomous Exploration," *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp.146-151, 1997.

[46] A. A. Makarenko, S. B. Williams, F. Bourgault and H. F. Durrant-Whyte, "An Experiment in Integrated Exploration," *IEEE International Conference on Intelligent Robots and System (IROS)*, vol. 1, pp. 534-539, 30 September-5 October 2002.

[47] S. Moorehead, R. Simmons and W. Whittaker, "Autonomous Exploration Using Multiple Sources of Information," *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 3, pp. 3098-3103, 2001.

[48] L. Liu, T. G. Crowe and J. N. Bakambu, "Efficient Exploration Algorithms for Rough Terrain Modeling Using Triangular Mesh Maps," *IEEE Conference on Robotics, Automation and Mechatronics*, pp.1206-1211, 21-24 September 2008.

[49] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numerische Math*, vol. 1, pp. 269-271, 1959.

[50] G. Oriolo, M. Vendittelli, L. Freda and G. Troso, "The SRT Method: Randomized Strategies for Exploration," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4688-4694, 2004.

[51] L. Freda and G. Oriolo, "Frontier-Based Probabilistic Strategies for Sensor-Based Exploration," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3892–3898, 2005.

[52] J. Espinoza, A. Sanchez and M. Osorio, "Exploring Unknown Environments with Mobile Robots Using SRT-Radial," *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 2089-2094, 2007.

[53] K. Singh and K. Fujimura, "Map Making by Cooperating Mobile Robots," *International Conference on Robotics and Automation (ICRA)*, pp. 254-259, 1993.

[54] L.Vincent and P.Soille, "Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.13, no.6, pp. 583-598, 1991.

[55] S. Hert and V. Lumelsky, "Polygon Area Decomposition for Multiple-Robot Workspace Division," *Special Issue of International Journal of Computational Geometry & Applications on Applied Computational Geometry*, vol. 8, no. 4, pp. 437-466, 1998.

[56] H. Bast and S. Hert, "The Area Partitioning Problem," *Canadian Conference on Computational Geometry*, pp. 163-171, Fredericton, NB, 2000.

[57] M. Jäger and B. Nebel, "Dynamic Decentralized Area Partitioning for Cooperating Cleaning Robots," IEEE *International Conference on Robotics and Automation (ICRA)*, pp. 3577–3582, 2002.

[58] M. Schneider-Fontan and M. Mataric, "Territorial Multi-Robot Task Division," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 5, pp. 815–822, 1998.

[59] A. Solanas and M. A. Garcia, "Coordinated Multi-Robot Exploration Through Unsupervised Clustering of Unknown Space," *IEEE International Conference on Intelligent Robots and Systems (IROS)*, vol. 1, pp. 717-721, 28 September-2 October 2004.

[60] K. M. Wurm, C. Stachniss and W. Burgard, “Coordinated Multi-Robot Exploration Using a Segmentation of the Environment,” *IEEE International Conference on Intelligent Robots and Systems (IROS)*, Nice, France, 22-26 September 2008.

[61] B. Yamauchi, “Frontier-Based Exploration Using Multiple Robots,” *International Conference on Autonomous Agents*, pp. 47-52, 1998.

[62] L. E. Parker, K. Fregene, Y. Guo and R. Madhavan, “Distributed Heterogeneous Sensing for Outdoor Multi-Robot Localization, Mapping, and Path Planning,” Multi-Robot Systems: From Swarms to Intelligent Automata, pp. 21-30, Kluwer Academic Publishers, 2002.

[63] W. Burgard, M. Moors, D. Fox, R. Simmons and S. Thrun, “Collaborative Multi-Robot Exploration,” *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1, pp. 476-481, 2000.

[64] W. Burgard, M. Moors and F. Schneider, “Collaborative Exploration of Unknown Environments with Teams of Mobile Robots,” In Dagstuhl, Springer Verlag, 2002.

[65] W. Burgard, M. Moors, C. Stachniss and F. Schneider, “Coordinated Multi-Robot Exploration,” *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 376-386, June 2005.

[66] A. K. Poemomo and H. S. Ying, “New Cost Function for Multi-Robot Exploration,” *IEEE International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 1-6, 5-8 Dec. 2006.

[67] D. Latimer IV, H. Choset, S. Srinivasa and A. Hurst, “Towards Sensor Based Coverage with Robot Teams,” *IEEE International Conference on Robotics and Automation*, pp. 961-967, 2002.

[68] C. S. Kong, N. A. Peng and I. Rekleitis, “Distributed Coverage with Multi-Robot System,” *IEEE International Conference on Robotics and Automation (ICRA)*, Orlando, Florida, May 2006.

[69] N. Hazon, F. Mieli and G. A. Kaminka, “Towards Robust On-line Multi-Robot Coverage,” *IEEE International Conference on Robotics and Automation (ICRA)*, Orlando, FL, May 2006.

[70] N. Agmon, N. Hazon and G. A. Kaminka, “Constructing Spanning Trees for Efficient Multi-Robot Coverage,” *IEEE International Conference on Robotics and Automation (ICRA)*, Orlando, Florida, May 2006.

[71] N. Hazon and G. A. Kaminka, “On Redundancy, Efficiency and Robustness in Coverage for Multiple Robots,” *Robotics and Autonomous Systems*, vol. 56, iss. 12, pp. 1102-1114, 2008.

[72] A. Franchi, L. Freda, G. Oriolo and M. Vendittelli, “A Randomized Strategy for Cooperative Robot Exploration,” *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 768-774, 10-14 April 2007.

[73] A. Sanchez, A. Toriz and M. A. Osorio, “Exploring Unknown Environments with Multiple Robots,” *Mexican International Conference on Artificial Intelligence*, 2008.

[74] R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun and H. Younes, “Coordination for Multi-Robot Exploration and Mapping,” *National Conference on Artificial Intelligence*, Austin, TX, 2000.

[75] R. Zlot et al, “Multi-Robot Exploration Controlled by a Market Economy,” *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 3, pp. 3016–3023, May 2002.

[76] B. P. Gerkey and M. J. Mataric, “Sold!: Auction Methods for Multirobot Coordination,” *IEEE Transactions on Robotics and Automation*, vol. 18, iss. 5, pp. 758-768, 2002.

[77] R. Zlot and A. Stentz, “Complex Task Allocation For Multiple Robots,” *IEEE International Conference on Robotics and Automation Barcelona (ICRA),* Spain, April 2005.

[78] I. Rekleitis, A. P. New and H. Choset, “Distributed Coverage of Unknown/Unstructured Environments by Mobile Sensor Networks,” MRS Workshop, 2005.

[79] N. Kalra, D. Ferguson and A. Stentz, “A Generalized Framework for Solving Tightly-Coupled Multirobot Planning Problems,” *IEEE International Conference on Robotics and Automation (ICRA),* Rome, Italy, 10-14 April 2007.

[80] Y. Fu, H. Li and Y. Ma, “Path Planning of Cooperative Robotics and Robot Team,” *IEEE International Conference on Robotics and Biomimetics*, Kunming, China, 17-20 December 2006.

[81] J. A. Mendez-Polanco and A. Munoz-Melendez, “Collaborative Robots for Indoor Environment Exploration,” *International Conference on Control, Automation, Robotics and Vision*, Hanoi, Vietnam, 17–20 December 2008.

[82] A. Howard, M. Mataric and S. Sukhatme, “An Incremental Deployment Algorithm for Mobile Robot Teams,” *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 2849–2854, 2002.

[83] V. Sujan, S. Dubowsky, T. Huntsberger, H. Aghazarian, Y. Cheng and P. Schenker, “An Architecture for Distributed Environment Sensing with Application to Robotic Cliff Exploration,” *Autonomous Robots*, vol. 16, iss. 3, pp. 287–311, 2004.

[84] Z. Chibin, W. Xingsong and D. Yong, “Complete Coverage Path Planning Based on Ant Colony Algorithm,” *International Conference on Mechatronics and Machine Vision in Practice (M2VIP08)*, Auckland, New Zealand, 2-4 Dec 2008.

[85] S. M. Thayer and S. Singh, “Development of an Immunology-Based Multi-Robot Coordination Algorithm for Exploration and Mapping Domains,” *IEEE International Conference on Intelligent Robots and System (IROS)*, vol. 3, pp. 2735-2739, 30 September-5 October 2002.

[86] D. F. Hougen et al, “A Miniature Robotic System for Reconnaissance and Surveillance,” *IEEE International Conference on Robotics and Automation (ICRA)*, San Francisco, CA, April 2000.

[87] J. N. Bakambu, P. Allard, and E. Dupuis, “3D Reconstruction of Environments for Planetary Exploration,” *International Symposium on Robotics*, Tokyo, Japan, 2005.

[88] J. N. Bakambu, P. Allard, and E. Dupuis, “3D Terrain Modeling for Rover Localization and Navigation,” *Canadian Conference on Computer and Robot Vision (CRV 2006)*, pp. 61–69, Quebec, June 2006.

[89] I. Rekleitis, J.-L. Bedwani, S. Gemme and E. Dupuis, “Terrain Modelling for Planetary Exploration,” *Computer and Robot Vision (CRV),* pp. 243–249, Montreal, QB, May 2007.

[90] D. Hähnel, W. Burgard and S. Thurn, “Learning Compact 3D Models of Indoor and Outdoor Environments with a Mobile Robot,” *Robotics and Autonomous Systems*, vol. 44, iss. 1, pp. 15-27, 2003.

[91] C. Ye and J. Borenstein, “A New Terrain Mapping Method for Mobile Robots Obstacle Negotiation,” *SPIE - The International Society for Optical Engineering*, vol. 5083, pp. 52-62, 2003.

[92] L. Edwards et al, “Photo-Realistic Terrain Modeling and Visualization for Mars Exploration Rover Science Operations,” *IEEE International Conference on Systems, Man, and Cybernetics*, October 2005.

[93] J. Wright, A. Trebi-Ollennu, F. Hartman, B. Cooper, S. Maxwell, J. Yen and J. Morrison, “Terrain Modelling for In-situ Activity Planning and Rehearsal for the Mars Exploration Rovers,” *IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, pp. 1372–1377, 2005.

[94] S. Se and P. Jasiobedzki, “Stereo-Vision Based 3D Modeling for Unmanned Ground Vehicles,” *SPIE - The International Society for Optical Engineering*, vol. 6561, 2007.

[95] C. F. Olson, L. H. Matthies, J. R. Wright and R. Li, “Visual Terrain Mapping for Mars Exploration,” *IEEE Aerospace Conference*, vol. 2, pp. 762-771, 6-13 March 2004.

[96] H. Choset and J. Burdick, “Sensor Based Planning, Part I: The Generalized Voronoi Graph,” *IEEE International Conference on Robotics and Automation (ICRA),* Nagoya, Japan, 1995.

[97] H. Choset and J.W. Burdick, “Sensor Based Planning, Part II: Incremental Construction of the Generalized Voronoi Graph,” *IEEE International Conference on Robotics and Automation (ICRA),* pp. 1643–1649, 1995.

[98] H. Choset, S. Walker, K. Eiamsa-Ard and J. Burdick, “Sensor-Based Exploration: Incremental Construction of the Hierarchical Generalized Voronoi Graph,” *International Journal of Robotics Research*, vol. 19, iss. 2, pp. 126-128, 2000.

[99] N. Kalra, D. Ferguson and A. T. Stentz, “Incremental Reconstruction of Generalized Voronoi Diagrams on Grids,” *Robotics and Autonomous Systems*, vol. 57, iss. 2, pp. 123-128, 2009.

[100] C. Castejon, B. L. Boada and L. Moreno, “Topographical Analysis for Voronoi-Based Modelling,” Annual *Conference of the IEEE Industrial Electronics Society*, 2002.

[101] H. Cheong, S. Park and S.-K. Park, “Topological Map Building and Exploration Based on Concave Nodes*,” International Conference on Control, Automation and Systems*, Seoul, Korea, October 14-17, 2008.

[102] R. Murrieta-Cid, C. Parra, M. Devy, B. Tovar and C. Esteves, “Building Multi-Level Models: From Landscapes to Landmarks,” *IEEE International Conference on Robotics & Automation (ICRA),* Washington, DC, May 2002.

[103] S. Simhon and G. Dudeck, “A Global Topological Map Formed by Local Metric Maps,” *International Conference on Intelligent Robots and Systems (IROS)*, Victoria, Canada, 1998.

[104] S. Thrun and A. Bücken, “Integrating Grid-Based and Topological Maps for Mobile Robot Navigation,” *National Conference on Artificial Intelligence (AAAI-96)*, pp. 944-950, Portland, OR, August 1996.

[105] S. Thrun, “Learning Metric-Topological Maps for Indoor Mobile Robot Navigation,” *Artificial Intelligence*, vol. 99, iss. 1, pp 21-71, 1998.

[106] N. Tomatis, I. Nourbakhsh and R. Siegwar, “Simultaneous Localization and Map Building: A Global Topological Model with Local Metric Maps,” *IEEE International Conference on Intelligent Robots and Systems (IROS)*, vol. 1, pp. 421-426, 2001

[107] J. Guivant, E. Nebot, J. Nieto and F. Masson, “Navigation and Mapping in Large Unstructured Environments,” *International Journal of Robotics Research*, vol. 23, iss. 4-5, pp. 449-472, 2004.

[108] B. Lisien, D. Morales, D. Silver, G. Kantor, I. M. Rekleitis and H. Choset, “The Hierarchical Atlas,” *Transactions on Robotics*, vol. 21, no. 3, pp. 473–481, June 2005.

[109] J. M. Gauch and S. M. Pizer, “Multiresolution Analysis of Ridges and Valleys in Grey-Scale Images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 6, pp. 635-646, June 1993.

[110] F. Liu, J. Deng, P. Cui and H. Wu, “A Method for Lunar Obstacle Detection Based on Multi-scale Morphology Transformation,” *IEEE International Conference on Mechatronics and Automation*, Harbin, China, 5-8 August 2007.

[111] D. A. George, C. M. Privitera, T. T. Blackmon, E. Zbinde and L. W. Stark, “Segmentation of Stereo Terrain Images,” *SPIE - The International Society for Optical Engineering*, vol. 3959, 2000.

[112] S. Moorehead, R. Simmons, D. Apostolopoulos and W. L. Whittaker, “Autonomous Navigation Field Results of a Planetary Ananlog Robot in Antarctica,” *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, pp. 237 - 242, Noordwijk, Holland, June 1999.

[113] H. Seraji, “Traversability Index: a New Concept for Planetary Rovers,” IEEE *International Conference on Robotics and Automation (ICRA)*, vol. 3, pp. 2006-2013, Detroit, MI, 1999.

[114] H. Seraji and B. Bon, “Autonomous Navigation of Planetary Rovers: A Fuzzy Logic Approach,” JPL Internal Document, Pasadena, CA, Jet Propulsion Laboratory, 1998.

[115] A. Howard and H. Seraji, “Real-Time Assessment of Terrain Traversability for Autonomous Rover Navigation,” *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 58-63, November 2000.

[116] A. Howard and H. Seraji, “A Real-Time Autonomous Rover Navigation System,” *World Automation Congress*, Maui, HI, 2000.

[117] H. Seraji et al, “Safe Navigation on Hazardous Terrain,” *IEEE International Conference on Robotics and Automation (ICRA)*, Seoul, Korea, May 2001.

[118] H. Seraji and A. Howard, "Behavior-Based Robot Navigation on Challenging Terrain: A Fuzzy Logic Approach," *IEEE Transactions on Robotics and Automation*, vol. 18, iss. 3, pp. 308-321, June 2002.

[119] H. Seraji, "Fuzzy Traversability Index: A New Concept for Terrain-Based Navigation," *Journal of Robotic Systems*, vol. 17, no. 2, pp. 75-91, 2000.

[120] A. Howard and H. Seraji, "Vision-Based Terrain Characterization and Traversability Assessment," *Journal of Robotic Systems*, vol. 18, iss. 10, pp. 577-587, 2001.

[121] A. Howard, H. Seraji and E. Tunstel, "A Rule-Based Fuzzy Traversability Index for Mobile Robot Navigation," *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 3, pp. 3067-3071, 2001.

[122] A. Howard, E. Tunstel, D. Edwards and A. Carlson, "Enhancing Fuzzy Robot Navigation Systems by Mimicking Human Visual Perception of Natural Terrain Traversability," *Joint IFSA World Congress and NAFIPS International Conference*, pp. 7-12, Vancouver, B.C., Canada, July 2001.

[123] C. Ye and J. Borenstein, "A Method for Mobile Robot Navigation on Rough Terrain," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3863-3869, 2004.

[124] D. B. Gennery, "Traversability Analysis and Path Planning for a Planetary Rover," *Journal of Autonomous Robots*, vol. 6, pp. 131-146, 1999.

[125] S. Goldberg, M. Maimone and L. Matthies, "Stereo Vision and Rover Navigation Software for Planetary Exploration," *IEEE Aerospace Conference*, vol. 5, pp. 2025–2036, 2002.

[126] C. Castejon, B. L. Boada, D. Blanco and L. Moreno, "Traversable Region Modeling for Outdoor Navigation," *Journal of Intelligent and Robotic Systems*, vol. 43, pp. 175-216, 2005.

[127] A. Hata, D. Wolf, G. Pessin and F. Osorio, "Terrain Mapping and Classification in Outdoor Environments Using Neural Networks," *International Journal of u- and e- Service, Science and Technology*, vol. 2, no. 4, pp. 51-61, 2009.

[128] R. Manduchi, A. Castano, A. Talukder and L. Matthies, "Obstacle Detection and Terrain Classification for Autonomous Off-Road Navigation," *Autonomous Robots*, vol. 18, pp. 81-102, 2005.

[129] P. Jansen, W. Mark, J. Heuvel, F. Groen, "Colour based Off-Road Environment and Terrain Type Classification," *IEEE Conferene on Intelligent Transportation Systems*, pp. 216-221, September 2005.

[130] Gi-Yeul Sung, Dong-Min Kwak and Joon Lyou, "Neural Network Based Terrain Classification Using Wavelet Features," *Journal of Intelligent Robotic Systems*, vol. 59, no. 3-4, pp. 269-281, February 2010.

[131] D. F. Wolf, G. S. Sukhatme, D. Fox and W. Burgard, "Autonomous Terrain Mapping and Classification Using Hidden Markov Models*," IEEE International Conference on Robotics and Automation (ICRA)*, 2005.

[132] A. Angelova, L. Matthies, D. Helmick and P. Perona, "Slip Prediction Using Visual Information," *Robotics Science and Systems Conference*, 2006.

[133] A. Angelova, L. Matthies, D. Helmick and P. Perona, "Learning and Prediction of Slip for Visual Information," *Journal of Field Robotics*, vol. 24, no. 3, pp. 205-231, 2007.

[134] R. E. Karlsen and G. Witus, "Vision-Based Terrain Learning," *SPIE - The International Society for Optical Engineering*, vol. 6230, pp. 33-42, 2006.

[135] R. E. Karlsen and G. Witus, "Image Understanding for Robot Navigation," *Army Science Conference*, Orlando, FL, 2006.

[136] R. E. Karlsen and G. Witus, "Terrain Perception for Robot Navigation," *SPIE - The International Society for Optical Engineering*, vol. 6561, 2007.

[137] K. Iagnemma, S. Kang, C. Brooks and S. Dubowsky, "Multi-Sensor Terrain Estimation for Planetary Rovers," *Proceedings of the 7th Annual Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, Nara, Japan, 2003.

[138] C. Brooks and K. Iagnemma, "Self-Supervised Terrain Classification for Planetary Surface Exploration Rovers," *Journal of Field Robotics*, vol. 29, no. 3, pp. 445-468, 2012.

[139] C. Brooks and K. Iagnemma, "Visual Detection of Novel Terrain via Two-Class Classification," *Proceedings of the 24th Annual ACM Symposium on Applied Computing*, pp. 1145-1150, 2009.

[140] C. Brooks and K. Iagnemma, "Self-Supervised Classification for Planetary Rover Terrain Sensing," *Proceedings of the 2007 IEEE Aerospace Conference*, Big Sky, Montana, 2007.

[141] A. Shirkhodaie, R. Amrani and E. Tunstel, "Visual Terrain Mapping for Traversable Path Planning of Mobile Robots," *SPIE - The International Society for Optical Engineering*, vol. 5608, 2004.

[142] A. Shirkhodaie, R. Amrani, N. Chawla and T. Vicks, "Traversable Terrain Modeling and Performance Measurement of Mobile Robots," *Performance Metrics for Intelligent Systems (PerMIS)*, 24-26 August, 2004.

[143] T. Kubota, R. Ejiri, Y. Kunii and I. Nakatani, "Autonomous Behavior Planning Scheme for Exploration Rover," *IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT 2006)*, pp. 258-264, 17-20 July 200

[144] R. Chatila, S. Fleury, M. Herrb, S. Lacroix and C. Proust, "Autonomous Navigation in Natural Environments," Lecture Notes in Control and Information Sciences, vol. 200, *The 3rd International Symposium on Experimental Robotics*, 28-30 October 1993.

[145] S. Lacroix, R. Chatila, S. Fleury, M. Herrb and T. Simeon, "Autonomous Navigation in Outdoor Environment: Adaptive Approach and Experiment," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 426-432, San Diego, CA, 1994.

[146] M. Cherif and C. Laugier, "Dealing with Vehicle/Terrain Interactions when Planning the Motions of a Rover," *IEEE International Conference on Intelligent Robots and Systems (IROS)*, vol. 1, pp. 579-586, 12-16 September 1994.

[147] D. Bonnafous, S. Lacroix and T. Simeon, "Motion Generation for a Rover on Rough Terrains," *IEEE International Conference on Intelligent Robots and Systems (IROS)*, Maui, HI, 29 October-3 November, 2001.

[148] E. Tunstel, A. Howard and H. Seraji, "Fuzzy Rule-Based Reasoning for Rover Safety and Survivability," *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, pp. 1413-1420, 1994.

[149] K. P. Valavanis, L. Doitsidis, M. Long and R. R. Murphy, "A Case Study of Fuzzy-Logic-Based Robot Navigation," *IEEE Robotics & Automation Magazine*, vol. 13, iss. 3, pp. 93-107, September 2006.

[150] A. Stentz, "Optimal and Efficient Path Planning for Partially-Known Environments," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3310-3317, vol. 4, 8-13 May 1994.

[151] A. Yahja, A. Stentz, S. Singh and B. L. Brumitt, "Framed-Quadtree Path Planning for Mobile Robots Operating in Sparse Environments," *IEEE International Conference on Robotics and Automation (ICRA)*, 1998.

[152] A. Yahja, S. Singh and A. Stentz, "An Efficient On-line Path Planner for Outdoor Mobile Robots," *Robotic Autonomous Systems*, vol. 32, pp. 129-143, 2000.

[153] S. Singh et al, "Recent Progress in Local and Global Traversability for Planetary Rovers," *IEEE International Conference on Robotics and Automation (ICRA)*, 2000.

[154] Y. Saab and M. VanPutte, "Shortest Path Planning on Topographical Maps," *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, vol. 29, no. 1, January 1999.

[155] L. Yenilmez and H. Temeltas, "Long Range Navigation Using 3-D Topological Map For General Purposes," *IEEE Annual Conference of the Industrial Electronics Society (IECON '03)*, vol. 2, pp. 1097-1100, 2-6 November 2003.

[156] E. Dupuis, P. Allard, J. Bakambu, T. Lamarche and W.-H. Zhu, "Towards Autonomous Long-Range Navigation," *ESA Workshop on Advanced Technologies for Robotics & Automation (ASTRA)*, Noordwijk, The Netherlands, 2-4 November 2004.

[157] I. Rekleitis, J.-L. Bedwani and E. Dupuis, "Over-the-Horizon, Autonomous Navigation for Planetary Exploration," *IEEE International Conference on Intelligent Robots and Systems (IROS)*, San Diego, CA, 29 October-2 November 2007.

[158] I. Rekleitis, J.-L. Bedwani, E. Dupuis and P. Allard, "Path Planning for Planetary Exploration," Canadian *Conference on Computer and Robot Vision (CRV)*, pp. 61-68, 28-30 May 2008.

[159] M. Massari, E. Ceriani, L. Rigolin and F. Bernelli-Zazzera, "Optimal Path Planning for Planetary Exploration Rovers based on Artificial Vision System for Environment," *IEEE International Conference on Advanced Intelligent Mechatronics*, Monterey, CA, 24-28 July 2005. Reconstruction

[160] R. Li et al, "Spirit Rover Localization and Topographic Mapping at the Landing Site of Gusev Crater, Mars," *Journal of Geophysical Research*, vol. 111, E02S06, 2006.

[161] K. S. Ali, C. A. Vanelli, J. J. Biesiadecki, M. W. Maimone, Cheng Yang, A. M. San Martin and J. W. Alexander, "Attitude and Position Estimation on the Mars Exploration Rovers," *IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, pp. 20-27, 10-12 October 2005.

[162] J. Stone et al, "GPS Pseudolite Tranceivers and their Applications," *Institute of Navigation TM-1999*, San Diego, CA, January 1999.

[163] E. LeMaster and S. Rock, “Self-Calibration of Pseudolite Arrays Using Self-Differencing Tranceivers,” *Institute of Navigation GPS-99*, pp. 1549-1558, Nashville, TN, September 1999.

[164] E. LeMaster and S. Rock, “Field Test Results for a Self-Calibrating Pseudolite Array,” *Institute of Navigation GPS-2000*, Salt Lake City, UT, September 2000.

[165] E. LeMaster, M. Matsuoka and S. Rock, “Mars Navigation System Utilizes GPS,” *IEEE Aerospace and Electronic Systems Magazine*, April 2003.

[166] M. Matsuoka, S.M. Rock and M.G. Bualat, “Autonomous Deployment of a Self-Calibrating Pseudolite Array for Mars Rover Navigation,” *Position Location and Navigation Symposium (PLANS 2004)*, pp. 733-739, 26-29 April 2004.

[167] M. Matsuoka, S. M. Rock and M. G. Bualat, “Rover, Go Your Own Way Self-Calibrating Pseudolite Array,” *GPS World*, vol. 15, iss. 6, pp. 14-22, June 2004.

[168] J.Y. Wong, *Theory of Ground Vehicles Third Edition*, John Wiley & Sons Inc, 2001.

[169] G. H. Peters, W. Abbey, G. H. Bearmana, G. S. Mungasa, J. A. Smith, R. C. Anderson, S. Douglas and L. W. Beegle, “Mojave Mars simulant—Characterization of a new geologic Mars analog,” Icarus 197, 470-479, 2008.

[170] R. Sullivan, R. Anderson, J. Biesiadecki, T. Bond, and H. Stewart, “Cohesions, friction angles, and other physical properties of Martian regolith from Mars Exploration Rover wheel trenches and wheel scuffs,” *J. Geophys. Res.*, 116, E02006, 2011.

[171] M. Misiti, Y. Misiti, G. Oppenheim and J.M. Poggi, *Wavelet Toolbox User’s Guide*, MathWorks Inc.

Curriculum Vitae

Name:	David Michel
Place of birth:	Toronto, Ontario
Year of birth:	1980
Post-secondary education and degrees	1999-2004 B.Sc (Eng.) Engineering Systems & Computing The University of Guelph Guelph, Ontario, Canada 2005-2007 M.E.Sc Electrical & Computer Engineering University of Western Ontario London Ontario Canada
Honours and awards	University of Guelph Entrance Scholarship 1999 Western Engineering Scholarship 2005-2011
Related work experience	2005-2011 Teaching Assisstant Department of Electrical & Computer Engineering The University of Western Ontario London, Ontario, Canada