SOLAR BASED NAVIGATION FOR ROBOTIC EXPLORERS

Kimberly J. Shillcutt

CMU-RI-TR-00-25

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
in the Robotics Institute
of
Carnegie Mellon University

Abstract

This thesis introduces the application of solar position and shadowing information to robotic exploration. Power is a critical resource for robots with remote, long-term missions, so this research focuses on the power generation capabilities of robotic explorers during navigational tasks, in addition to power consumption. Solar power is primarily considered, with the possibility of wind power also contemplated. Information about the environment, including the solar ephemeris, terrain features, time of day, and surface location, is incorporated into a planning structure, allowing robots to accurately predict shadowing and thus potential costs and gains during navigational tasks. By evaluating its potential to generate and expend power, a robot can extend its lifetime and accomplishments. The primary tasks studied are coverage patterns, with a variety of plans developed for this research. The use of sun, terrain and temporal information also enables new capabilities of identifying and following sun-synchronous and sun-seeking paths.

Digital elevation maps are combined with an ephemeris algorithm to calculate the altitude and azimuth of the sun from surface locations, and to identify and map shadows. Solar navigation path simulators use this information to perform searches through two-dimensional space, while considering temporal changes. Step by step simulations of coverage patterns also incorporate time in addition to location. Evaluations of solar and wind power generation, power consumption, area coverage, area overlap, and time are generated for sets of coverage patterns, with on-board environmental information linked to the simulations. This research is implemented on the Nomad robot for the Robotic Antarctic Meteorite Search. Simulators have been developed for coverage pattern tests, as well as for sun-synchronous and sun-seeking path searches. Results of field work and simulations are reported and analyzed, with demonstrated improvements in efficiency, productivity and lifetime of robotic explorers, along with new solar navigation abilities.

Table of Contents

Abstract ..... **i**

List of Figures ..... **v**

List of Tables ..... **vii**

Acknowledgements ..... **viii**

I. Introduction **1**

A. Motivation 1

B. Goals 9

II. Background **11**

A. Solar Power 11

B. Navigational Planning 14

C. Mission Planning 19

III. Methodology **25**

A. Solar Ephemeris Algorithm 25

B. Solar Navigation 27

C. Coverage Pattern Simulation 32

D. Evaluation Categories 35

E. Task Selection 42

F. Implementation 45

IV. Experimentation **55**

A. Experimental Procedures 55

B. Field Work 57

C. Simulations 59

V. Results 65

A. Fieldwork 65

B. Simulations 76

C. Solar Navigation 103

VI. Conclusions 107

A. Contributions 110

B. Future Work 111

VII. Appendices 115

A. Navigational Planning Library 115

B. Solar Ephemeris Library 121

C. Meteorite Search Demonstration Statistics 133

VIII. Bibliography 137

List of Figures

1. Changing crater shadows at 80 S, Earth's summertime, with sun elevation 3
2. Potential lunar cold traps overlaid on Earth radar image.....5
3. Lunar polar sun elevation and horizon lines.....7
4. Sun angle calculations for lunar surface 26
5. Hypothetical sun-synchronous and sun-seeking paths 28
6. Shadow mapping.....29
7. Coverage patterns 33
8. Row ends for straight rows pattern 34
9. Wind values on January 25, 2000, at Elephant Moraine 39
10. Three methods for sorting and selecting plans 43
11. Aerial view of shadows in Antarctica 45
12. Nomad software architecture 48
13. Nomad in Antarctic configuration 56
14. Coverage pattern groundtracks, Patriot Hills.....58
15. Shadows at lunar south pole, 0UT, April 1, 2000 63
16. Shadows at Haughton Crater, 500UT, July 15, 2000.....64
17. Groundtracks of straight and spiral patterns, Elephant Moraine 66
18. Power generation per panel, Elephant Moraine straight rows pattern 67
19. Power generation per panel, Williams Field straight rows pattern 68
20. Power generation per panel, Elephant Moraine spiral pattern.....68
21. Actual vs. calibrated a priori solar power generation simulations.....69
22. Actual vs. desired pattern groundtracks, Elephant Moraine 74
23. Pattern error for spiral pattern, Elephant Moraine.....74
24. Straight rows pattern at Williams Field 76
25. Area covered and overlapped by range of pattern types.....77
26. Varying row length for sun-following patterns.....78
27. Energy consumed by range of pattern types 79
28. Solar energy generation 79

29. Variation in straight rows energy generation due to initial robot yaw 82
30. Area covered and overlapped as a function of pattern curvature.....84
31. Energy consumed as a function of pattern curvature 85
32. Solar energy generation as a function of curvature 86
33. Covered regions of Haughton Crater 86
34. Effect of terrain shadowing on solar power generation 87
35. Robot lifetimes for 23 pattern types - 80S, Earth, summertime 90
37. Area coverage after fixed mission time 92
38. Results from multiple target distributions.....94
39. Results from multiple runs, same targets, random timings.....95
40. Mission times for 23 pattern types - 60S, Earth, summertime 96
41. Solar energy as a function of pose uncertainty 100
42. Sample variable wind/sun direction results from 80S 101
43. Sun-synchronous path around feature in Haughton Crater.....104
44. Autonomously-generated path to a recharging location 106
A1. Alternating intertwined pattern 119
A2. Angular calculations 122
A3. Calculating angular terrain elevation 133

List of Tables

1. Solar power generation prediction accuracy 70
2. Mean and standard deviation of robot pose 71
3. Mean heading differences, straight rows patterns 72
4. Mean pattern error 73
5. Errors in predicted area calculations 75
6. Total solar energy generation comparison, Elephant Moraine 81
7. Improvement metrics for mission scenarios 97
A1. Basic search pattern characteristics 134
A2. Search pattern energy characteristics 135
A3. Search pattern timing characteristics 136

Acknowledgements

I would like to thank my advisor Red Whittaker and my thesis committee members Reid Simmons, Tony Stentz, and Pete Bonasso, for all their comments, suggestions and ideas over the past few years, helping me to focus and refine my research. I would also like to thank the members of and contributors to the Robotic Antarctic Meteorite Search project and Nomad (Dimi, Ben, Mike, Mike, Mike, Sib, Jim, Alex, Liam, Matt, Stewart, Nicolas, Vandi, Chris, Eric, Pascal, Bill, John, John, Lenny). Without them, my work would have been much less rich, and with far fewer field results. The RAMS project, along with the Haughton-Mars project, also allowed me to experience the most incredible adventures literally at the ends of the Earth. Beyond Nomad, I am grateful for the opportunities presented by the sun-synchronous project, and for the assistance and advice of Dave W. and Paul T. And for guiding me through all the details of life in the FRC, I definitely want to thank Dot Marsh. On a more official note, I should acknowledge the grant money provided by the NASA Graduate Student Researchers Program (NGT-5-50248), as well as the support of the Robotics Institute, Carnegie Mellon University, and the Zonta Foundation. And on a lighter note, thanks to the numerous other people at the RI for making my stay here enjoyable as well as productive, with foosball, FIRST, flying and free food being some of the highlights.

Chapter I

Introduction

A. Motivation

Humans continually seek out new places, searching for the unknown and the undiscovered. Explorers have survived inhospitable conditions and unproven equipment to pursue knowledge. Today, capable and adaptable robots can explore areas too dangerous or costly for humans to visit, such as planets and moons, ocean floors, polar regions, and other remote earthly locations. As an extension of the human presence, robotic explorers must provide as much information as possible about the region being explored, reacting intelligently to changing conditions and remaining alive and active for extended periods of time. One primary environmental variable, the interaction of the sun and terrain over time, creates both difficulties and opportunities for robotic explorers. On-board intelligence enabling the prediction of shadows and solar power generation provides the abilities needed for a new type of robotic navigation -- solar navigation -- which produces self-sufficient robots capable of prolonged operation in remote locations.

This chapter describes the driving factors behind this research. The critical need for power, along with the limited power sources available in remote regions, inspires the development of power generation prediction and *resource creation*. The information needed to effectively harness solar power, among other benefits, is contained in *environmental knowledge*. Then, the unrivaled ability of *robotic explorers* to take advantage of such knowledge is made evident. Finally, the extreme lighting conditions in *polar regions* lead to the development and discovery of specific navigational tasks and capabilities, along with unique benefits for exploring such regions.

RESOURCE CREATION

Reasoning about actions and their effect on resources is crucial in remote locations. On-board resources are restricted by size and weight allowances for any robot launched into space or placed into other remote areas, due to limited financial budgets and physical constraints. Successfully accomplishing a mission, therefore, requires careful use of the available resources. Robotic explorers must evaluate potential tasks not only in terms of the resources they will expend, but also of the resources they can gain. Knowledge of resource usage and production is thus of vital importance, potentially making the difference between success and failure of a mission.

The most critical exploration resource is power, since without power, a robot cannot perform tasks or record imagery or even communicate. Without nearby humans to easily refuel robots or replace batteries, power generation becomes crucial for extended missions. For the class of remote, exploring robots in the inner solar system, solar power is the mode of choice [17, 18]. The use of non-renewable batteries or fuel cells alone is not practical for long missions, due to the enormous weight and volume that would be needed to transport the required amount. Nuclear power is currently infeasible, due to cultural resistance and lack of technological readiness. Despite the drawbacks of solar power, such as inefficiencies due to material composition, dust storms or clouds, solar power is a prime power source in the inner solar system, ranging from the sun to the asteroid belt. By intelligently planning the collection of solar power during the course of its mission, a robot can minimize its power storage requirements, which directly translates to reduced weight, complexity, and cost.

Going beyond the use of planning to conserve and allocate resources, this thesis addresses the issue of generating new resources. Solar power, and even wind power, can be harnessed by robots as they explore their surroundings. By considering its own potential to generate and expend power, a robotic explorer can extend its lifetime and accomplishments.

ENVIRONMENTAL KNOWLEDGE

To generate solar power, several factors need to be considered: the visibility and strength of sunlight, terrain shadowing, the solar panel configuration, and the changing orientation of those

panels with respect to the sun. The motion of the sun over time in a particular location combined with terrain maps will indicate whether or not the sun will be visible at a given location, and at which angle the sunlight will be incident on the panels. With this knowledge, an intelligent planner can determine the amount of power which can be generated. The importance of solar and terrain knowledge, along with timing, can be seen in Figure 1, showing several snapshots of the shadows appearing within a hypothetical polar crater at 80 S on Earth. Twenty four hours of daylight are available during this summer period, but selecting the best location to explore requires knowledge of the extensive shadowing.

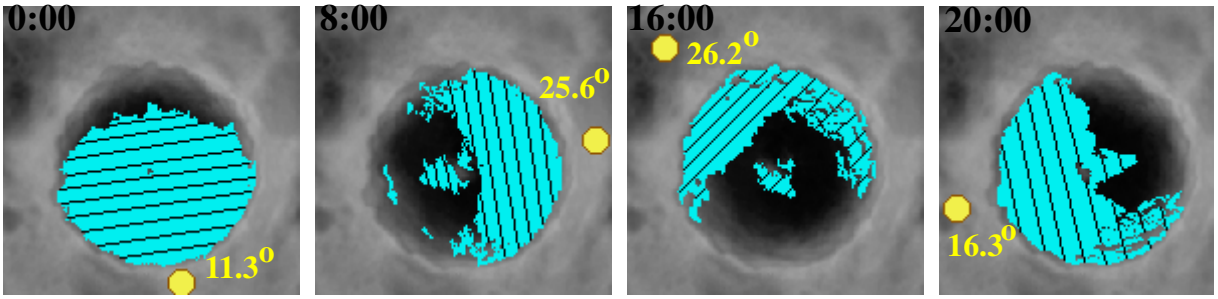


Figure 1: Changing crater shadows at 80 S, Earth's summertime, with sun elevation

A clear understanding of the changing relationships between the sun, the terrain, and the robot is needed for many navigational objectives. Comparison of actual solar power gain to predicted values can also indicate reduced lighting conditions due to clouds or other phenomenon, useful for model calibration. And upon gathering updated local terrain information, solar ephemeris knowledge can be applied to determine more precisely the existence of permanently shadowed valleys and craters in polar regions. Knowledge about the wind can also be useful for robotic explorers on Earth and Mars. Wind turbines can be utilized for additional power generation, and a predominant wind direction can be exploited to ease the wind resistance of a robot's profile, reducing power consumption.

ROBOTIC EXPLORERS

Robotic explorers are well-suited to take advantage of a changing environment, as they have the ability to travel to seek out the most beneficial conditions. With sufficient knowledge of the

terrain and the sun's movement, a robot can continually seek out sunlit valleys and ridges, generating power to achieve goals and remain alive as long as possible. The extension of this concept leads to a unique ability of sun-synchronous circumnavigation. In polar regions where the sun is continually above the horizon for long periods of time, a robot can follow the sun around the planetary pole or another feature, continually staying in sunlight and generating power. By finding and following such routes, a robotic explorer can operate in polar regions for an entire season or more, remaining active continually without the need to pause at nightfall and shut down non-critical systems until sunrise, greatly extending the typical lifetime of planetary robots. The range of such robots is also greatly extended, allowing exploration of an enormous region, leading to more discoveries than previously considered possible by robotic vehicles. Because of their ability to generate sufficient solar power and continually travel with the sun, robots will not need to depend on nuclear power sources, or deal with nighttime hibernation and heating issues. They can thus be of less complexity, leading to lower mass and lower cost robots.

Other tasks for robotic explorers include performing patterns to completely cover entire regions, and finding paths to designated sites. In order to examine an area thoroughly and identify any potential targets, coverage patterns are essential. This thesis addresses the question of the performance of different types of coverage patterns, with respect to criteria such as power generation, power consumption, area covered, area overlapped, and total time. In order to more closely examine target objects or to take imagery from designated angles, path finding is required. Both area coverage and path finding benefit from modeling the sun, terrain and time interactions, with which a robot can select plans which allow the best chance of survival and success. Paths can also be sought to find suitable locations and orientations for recharging batteries or downlinking information to Earth users, after an extended period in shadowed regions or away from Earth line-of-sight. Once robotic explorers have identified and mapped short-term or long-term solar-powered routes, these routes can be transferred to human-driven vehicles, after human bases have been established. For both robotic and human vehicles, onboard knowledge of terrain, lighting, and Earth visibility will be essential for any extended trips, providing information on possible locations for recharging batteries, optimal orientations for solar panels, and locations for communicating with Earth.

POLAR REGIONS

Polar regions are prime locations for robotic explorers, whether on the Earth, the moon, Mars, or other bodies. The underlying reason for this interest is the sun. For many of these bodies, the poles have the potential to harbor water ice and other volatiles, permanently shielded from the low-lying sun by craters and other terrain features [10, 32, 58]. For the moon, given the small angle between its spin axis and the ecliptic normal, the sun elevation varies from only about 1.6° above the horizon to 1.6° below at the poles [44]. With the sun always so low on the horizon, small rises in the terrain cause extended shadows, and some regions may be permanently shadowed year-long. In these regions, ice can collect over millions of years, and never sublimate. Figure 2 indicates some potential cold traps, shown in white, derived from Earth-based radar measurements of the terrain [58]. The Earth is always toward the top of this image, always facing the near-side of the moon, with terrain peaks and crater rims casting Earth-shadows toward the bottom of the image. The sun circles around the horizon and around the south pole, at low angles.

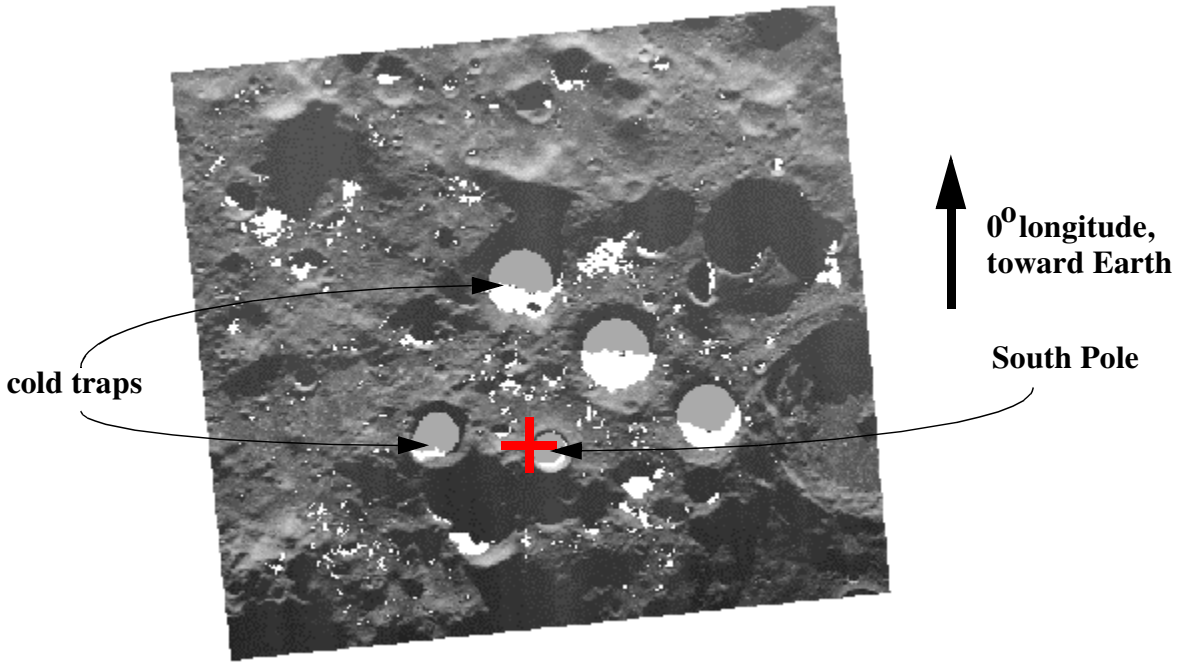


Figure 2: Potential lunar cold traps overlaid on Earth radar image

Ice at the lunar poles will enable scientists to understand more about the history of the solar system, providing facts to help determine the origin of the ice and to chronicle lunar surface

impacts. In addition to scientific knowledge, polar volatiles on the moon are extremely valuable commodities. Oxygen and hydrogen can be extracted for cryogenic rocket propellant, and water combined with surface regolith can be used in construction. Commercial mining operations on the moon could enable large-scale construction in space without depleting Earth resources and without the expenses involved in moving heavy equipment and materials from Earth to the moon. Mining water from the poles will also enable the production of drinking water and oxygen, reducing a habitat's dependence on resupplies from Earth or fully recyclable life support systems.

Sun and terrain information is needed by robotic explorers in order to decisively identify and find cold traps containing ice, and to navigate around the shadow-sensitive polar regions. In addition, this same information can be used in the selection of appropriate lunar base sites. One of the critical factors in determining the best location for a base is knowledge of the orbital ephemeris and of terrain features which may either assist in, or detract from, viewing the sun and Earth. Viewing angles to the Earth from the lunar poles remain within 6.7° of the horizon, similar to the low sun angles, resulting in frequent occlusions by even low-lying terrain features. Locations with both sun and Earth visibility are needed for most base sites, whether human or automated. Permanently shadowed, resource-rich areas which are adjacent to sunlit and earthlit peaks will be prime resource extraction sites. Sunlit areas will be needed to host power generation facilities, and earthlit areas will be needed to establish communication links with Earth. Power is required for lighting, oxygen production, food production and preparation, and research, while sunlight is important to support the morale of human inhabitants. Communication visibility with Earth will be needed on a regular, relatively frequent basis.

The low sun and Earth visibility angles are not a detriment to operations in polar locations, however, but a benefit. Just as the sun is barely above the horizon during the summer, the sun is barely below the horizon in the winter. Peaks high enough above surrounding features will provide sunlight for extended periods even in winter. A base located at such a site would be able to remain in sight of the sun nearly year-round. Figure 3 shows the south polar sun elevation from the moon for a three year period, with the horizon line marked. The line at 0° is the horizon for flat ground, while the line below that is for a postulated terrain high point, showing that even with the sun elevation below zero degrees, the sun will be visible above the horizon. For the sun to be

visible at a location even when the sun's elevation is -1.2°, that location needs to be at least an angular distance of 1.2° above all the surrounding terrain. A solitary peak, surrounded by flat ground all the way to the moon's horizon would only need to be 382 meters high for this to happen. In addition, the sun's disk subtends an angle of 0.5°, while the elevations given here are for the sun's center. This means that a partial sun can be seen at even lower elevations.

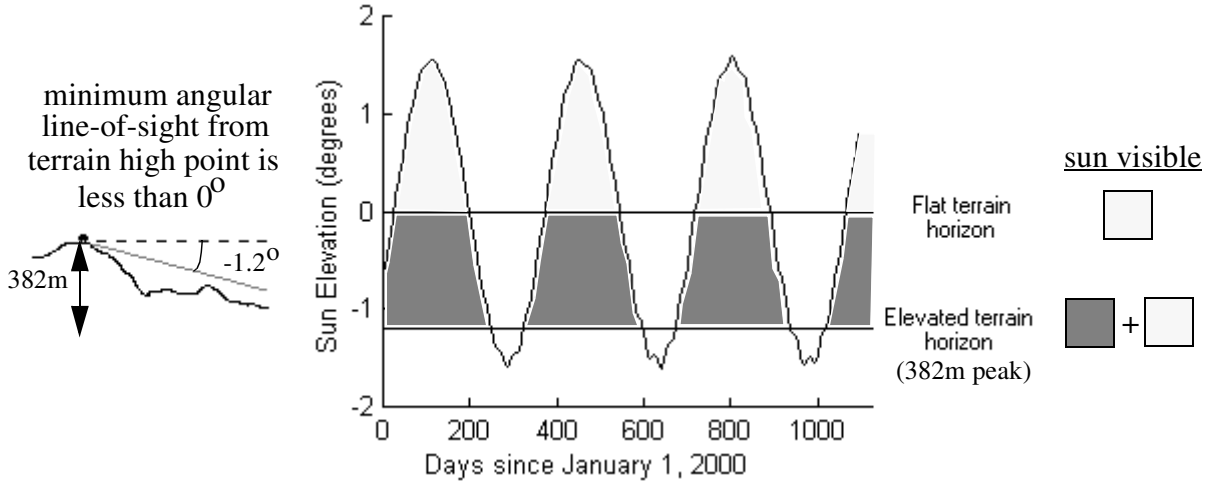


Figure 3: Lunar polar sun elevation and horizon lines

Since such elevated regions at the poles may experience illumination for most of the year, a single solar power production facility located there, rotating its solar collectors to match the sun's motion, would enable near-continuous energy collection. One step beyond that concept, identifying multiple terrain high points, or even level terrain, at various longitudes around a polar region will enable the installation of a permanent power generation infrastructure. These sites can be identified by the combination of terrain maps and solar ephemeris algorithms, with robots then being sent to automate the construction of solar arrays at each point. One or more arrays will be exposed to sunlight at all times so that when connected by conductors, the resulting ring can provide a continual source of power for all polar activities: resource extraction facilities, scientific outposts, astronomical observatories, and human bases.

Robotic explorers can trailblaze such sun-synchronous routes, identifying appropriate paths to link array sites for polar power grids. Since the moon rotates once a month, even having multiple solar arrays spaced around the lunar equator would enable continuous power generation. But an

equatorially-distributed solar power grid would require power distribution over 10,000 kilometers of territory. Assuming a perfectly spherical and flat moon, a polar solar grid would only have to be about 1.5° off the pole to have sun visibility during the lunar winter, resulting in a power grid circumference of about 340 km. High elevation sites nearer the pole will enable this distance to be decreased even more. Power distribution from this solar array grid would service polar bases, which could be located anywhere near the connecting power cables. Establishing outposts in these regions would drastically reduce their dependence on power storage or nuclear power production. Creating this network of power generation stations could also provide an additional power source for Earth or orbiting stations, by beaming the power via microwave transmissions.

Finally, thermal characteristics of the polar regions provide additional benefits for activities in these areas. Equatorial regions of the moon experience extreme variations in surface temperature as the sun rises and sets once a lunar day (254 K ± 140 K), while polar temperatures vary more slowly and less dramatically (220 K ± 10 K) [44]. The requirements on the thermal design of polar facilities would be far less demanding than those for the highly variable equatorial and mid-latitude regions. Equatorial and mid-latitude regions also experience full day and night cycles every lunar day, leading to very irregular solar power production schedules.

Though much of the previous discussion concentrated on Earth's moon, other planetary bodies also have intriguing polar regions. Scientists desire more in-depth studies of the Martian polar regions, where permafrost soil conditions may exist, and where there is the potential for finding signs of previous life. Radar data suggests that water may even exist on Mercury's poles [49]. Closer to home, the Antarctic continent provides not only an Earth-based polar analog for testing purposes, but also fertile ground for finding and studying meteorites, adding to knowledge of the solar system.

OVERVIEW

The motivation for this thesis research then is primarily concerned with power generation. Remote regions have a critical need for the limited resource of power, leading to a study of how to predict power generation, with the encoding of the sun, terrain and temporal information needed to harness solar power effectively. Expanding the capabilities of robotic explorers to take

advantage of the changing sun and shadows is another motivation, guiding the research into the evaluation of navigational tasks and the search for sun-synchronous and sun-seeking paths. Lastly, the unique characteristics and needs of polar regions introduce several motivations: enabling solar powered circumnavigation for continual exploration, enabling navigation to discover rich scientific and commercial resources, and identifying prime locations for polar power grid infrastructures and potential base sites.

B. Goals

The purpose of this thesis is to demonstrate how essential sun, terrain and temporal information is for solar-powered robotic exploration. This research strives for three goals, relating to three scales of navigation. At the largest scale, long range paths throughout a region are sought such that a robot can travel from one site to another while remaining continually in sunlight. Polar regions are considered in particular, as the sun can remain continually above the horizon for entire seasons. Low sun elevations also persist in such regions, causing long shadows highly sensitive to terrain elevations and the motion of the sun over time.

At an intermediate scale, coverage patterns are used to investigate in detail smaller regions. Improvements are sought in the efficiency, productivity, and lifetime of solar-powered robots performing coverage by considering an assortment of coverage pattern types and variations. Such robots are assumed to have fixed solar panels, and to be primarily powered by solar energy, with limited battery reserves. Shadowing and changing sun angles affect a robot's performance while following different pattern types. Solar power generation prediction must therefore be developed.

The third goal is enabling autonomous emergency recovery by finding short-range paths to locations with sun or Earth line-of-sight. Robots exploring in shadowed areas will need to find sunlit locations to recharge batteries when needed, and robots driving out of range of Earth communication will need to regain communications line-of-sight to relay data or receive new orders. Finding satisfactory locations without human interaction requires on-board knowledge of terrain and orbital parameters.

THESIS STATEMENT

This thesis asserts that environmental knowledge such as solar and terrain information can improve the performance of remote robotic explorers. On-board knowledge and power generation prediction enables simulation and evaluation of applicable navigational tasks in the current environment, while off-board knowledge enables time-intensive temporal and spatial predictions to be generated. This thesis addresses the question of how sun, terrain and temporal parameters can be used to increase the productivity, efficiency, lifetime and abilities of robots. Quantitative studies of the improvements are shown using field data and simulations.

Chapter II

Background

Solar power concepts, navigational planning, and mission planning architectures have all been studied to varying degrees. This section will describe previous work done in each of these areas, and compare the work to the new concepts introduced by this thesis research.

A. Solar Power

The research to date concerning solar power deals with several issues: material composition of solar panels, modeling of solar panel responsiveness, charting of the solar ephemeris for specific locations and times, and optimizing deployment angles for fixed solar panels. Composite mapping of sunlight on lunar terrain features has been done for some regions based on astronomical observations. Resource management for task ordering and resource allocation has also been studied, with some mention of solar power constraints.

Several studies have been made of the material composition of solar arrays. Landis [51] describes the evaluation of several types of material based on the environment of Mars. The effect of latitude on available solar power is mentioned, but more attention is paid to the effects of dust, light spectrum, temperature, and the soil. An experiment to test five types of solar array material near the Martian equator is described in [76], proposed to take place on the Mars 2001 Surveyor Lander. An earlier paper by Landis and Appelbaum [52] also discusses the efficiencies of solar array materials, and various projects which could utilize solar power on Mars. Panel efficiencies range from 14.5% to over 30% with a concentrator, but they are also dependent on

temperature. In addition, [52] provides graphs of the number of daylight hours as a function of time of year and latitude. However, these papers primarily concentrate on the physical aspects of solar arrays, and not the dynamic evaluation of power generated by a specific solar panel configuration.

The optimum angles for deployed solar arrays on Mars or the moon are the topic of two papers by Anthony Colozza. In [17], Colozza describes the deployment of a stationary tent-shaped array. The desired energy profile is flat, generating a constant amount of power. A tent angle of 60 degrees is found to be best, assuming the sun passes directly overhead, which is too much of a simplification for the needs of this thesis. In [18], a tracking array is studied, comparing the gain of generating more power with the costs of consumed power, mass, volume, implementation time and reliability of the array. It proves to be only slightly more profitable to use a tracking array than a fixed array with most of the solar panel materials evaluated, and considering all the costs involved with the structure of a moveable array. No specific latitude or range of sun angles was analyzed, leading to very generalized results. In both papers, more details about the composition and mechanical configuration of the arrays are described than the actual evaluation of solar power gain over time.

Three papers discuss the modeling of solar power input and output for proposed and actual Martian missions. McKissock et al. [61] discuss the power requirements for a forty day human mission in an equatorial location. The solar model used has equal amounts of sun and darkness, and the solar arrays considered are fixed. Ewell and Burger [27] describe the solar array power model for Mars Pathfinder. This models the power produced by the lander, based on the latitude of the landing site, the orientation of the lander, the time of year, and the atmospheric and temperature conditions. Shadowing by the rest of the lander is considered as it changes throughout the day, but the lander itself is not mobile. The authors state that the sunlight model used is “too crude an approximation...especially for missions near the poles.” So while the concept of modeling how much power can be produced is relevant to this thesis research, it is not easily applied to a mobile rover at an arbitrary latitude. A review of the power components of Mars Pathfinder is given in [81]. The model discussed in [27] provides some of the input for the power subsystem model, which takes the optical depth and intensity of the sunlight, the tilt and

shadowing of the solar array, and the amount of power used to predict the status and voltage of the lander battery. Again, the modeling concept is relevant to this thesis, and can be used to develop a more accurate solar power generation model, but does not consider a mobile rover, or tasks such as coverage patterns.

Studies of solar power in polar latitudes are even more limited. A study was made of the trade-offs between cost, complexity and power generation for a fixed versus tracking solar panel for the Mars Polar Lander [90]. The fixed panel was found to be more cost-effective, despite the large angular range of sun position at the lander's proposed site of 73° to 78° S latitude. Instead of panels tracking the sun, the lander was preprogrammed to touch down in an orientation which optimized the amount of sunlight incident on the panels and limited the amount of shadowing caused by the rest of the lander body. As a direct precursor to this thesis research, the appendix of a technical report detailing the design of a lunar polar rover [22] describes calculations of solar elevation for polar latitudes on the moon. Some description of terrain effects is also given.

More general lunar polar lighting conditions have been researched, concentrating on the possibility of cold traps, where no sunlight ever strikes the surface [10, 58]. Elevation maps of the moon's south pole have been generated using radar returns from Arecibo Observatory [59], and are used in simulations in this thesis. Lighting maps have been generated based on Clementine imagery from one season of observations [10]. These maps aid in identifying regions with sunlight for large percentages of the lunar day, but are restricted to available observational data.

Ephemeris algorithms are readily available for sun, moon, and planetary positions as seen from the Earth's center or various surface locations, and are packaged in many software programs and on many web pages. These programs do not generally enable ephemeris calculations as seen from the surface of other bodies, however, or include surface heights. Output from a more detailed ephemeris algorithm is available from JPL [47], which does calculate positional and other information for the sun and the Earth, as seen from various planetary surface locations. Currently, viewing locations from within 1km of the moon's poles are not available, however. Also, this data is not combined with terrain elevation maps to determine actual surface heights or shadowing. The ephemeris algorithm developed for this research, on the other hand, does include terrain elevations and terrain shadowing calculations, and has been incorporated with navigational

planning. This ephemeris provides values for any time and surface location on the Earth or moon with minimal computation time, and with an accuracy more than sufficient for robot motion (less than 10 arc seconds of angular error).

Bresina et al. [7] suggest the use of solar power resource information in the planning process. Shadowing, continual assessment of resources, and updating the utility of plans based on environmental and resource changes are all mentioned. The power profile described includes estimations of solar exposure, terrain tilt, and risks associated with uncertainty. These concepts are highly relevant to this thesis, though they are not implemented or developed in detail. More about [7] is discussed in the section on planning architectures, as there is more relevance to that aspect than to the development of solar algorithms.

This thesis work will introduce several new factors to the current body of research. Variable locations, instead of single fixed locations will be addressed, with solar power generation linked to robot motion. A solar ephemeris will be implemented for onboard rover use with an accuracy sufficient for robotic navigation, describing solar conditions for polar locations as easily as for equatorial locations, on Earth and on the moon, and providing data for any given date and time. This ephemeris will be utilized by navigational algorithms, providing evaluations of a wide variety of potential tasks. The ephemeris, combined with elevation maps, will enable mapping of sunlight on terrain features for any future or past times, decoupled from observational data.

B. Navigational Planning

Navigation in outdoor environments and planning of paths are each topics that have generated a multitude of methodologies and papers. Latombe [53] reviews many of the approaches. For the purposes of this thesis research, several related subjects provided useful background knowledge.

One subject is the formulation of maps to use in representing the environment and in planning paths. Maintaining complete and high resolution information about the robot's environment is difficult and resource-intensive, but multiple ways to consolidate the information have been developed. Brooks first suggested representing the obstacle-free space with geometrical cones

[8]. Other methods are the visibility graph, based on the vertices of obstacles [54, 73, 96], the Voronoi graph, based on the distance from obstacles [15, 73], and cellular decomposition, dividing the area into variously shaped regions [19, 73, 48]. A uniform grid cell is also a common representation [55], where the grid resolution can be chosen to reflect the size of obstacles or the robot [19]. This is the approach used in the implementation of this work, as the a priori digital terrain elevation maps to be incorporated are already grid-based. This approach also provides a simple way to attach additional information, indicating the presence of target objects or shadows in a specific grid cell, or indicating which grid cells have been covered by the robot, for example. Grid cells which are approximately the scale of the robot allow determining the presence of shadows which are also as small as the scale of the robot, enabling sufficient identification of shadows which effect solar power generation capabilities along the robot's path. To reduce the memory costs of using a grid-based approach, a map structure is developed which only adds grid blocks to the map incrementally as they are needed.

For all these types of maps, certain useful points can be selected, such as meetpoints where all nearest obstacles are equally distant, nodes situated between obstacles at defined distances, or lines of points separating obstacles. Maintaining only this point information further consolidates information about the environment, allowing a concise graph, such as a quadtree, to be built. This graph can be searched to pick a sequence of nodes to follow through the environment [55, 96]. The best paths between each node can be determined beforehand. For known environments, a complete map can be built ahead of time, and unknown environments can be mapped incrementally, using sonar, vision, or other sensors. This type of approach may be useful in searching through the three-dimensional shadow maps of location and time for the solar navigational paths developed in this thesis.

Algorithms for searching a graph or map are another useful subject. One of the primary considerations in finding a path is avoiding obstacles, so by limiting the path search to free space areas, collisions can be avoided. To guarantee that the path is wide enough for the robot, many maps are built with "expanded" obstacles, or configuration space [53, 77]. Aside from collisions, the length or cost of a path should be optimized as well. Several algorithms can search a map and produce a minimum cost path, such as A* [40, 42, 54, 73], D* for when the world is not

completely known or is changing [85, 86, 87], and distance wavefront propagation [19, 39, 96]. The cost function used in these approaches can be not only distance, but can also include a discomfort cost of being too near obstacles [94], a value describing terrain difficulty [83], the number of maneuvers required [40], visibility of a landmark, or amount of uncertainty [41]. Another way to avoid obstacles is to use a potential field technique, that provides a "repulsion" factor to steer the robot away from dangerous areas [93]. A common failing of this technique is the production of potential field minima which trap the robot, but there are several methods to overcome this failing, by either proper definition of the potential function, or search algorithm techniques to escape any minima [53, 84]. One such technique uses a genetic algorithm to effectively generate a random search when the robot is temporarily trapped in this way [77].

A non-optimal, but guaranteed path can be found by using a "Bug" algorithm. This algorithm guides the robot straight to the goal, and when an obstacle is encountered, the robot follows rules which guide it around the obstacle [54, 57, 60]. A different approach, utilizing qualitative information and a human-based idea, is to plan a route with multiple levels, similar to picking major highways to get from one city to another, then picking the local roads which lead to a particular location within the city [38]. Another multi-layered approach depends on the type of environment. In simple areas, the robot just reacts to the immediate environment. In more complex areas, a two dimensional map is utilized, and in the roughest terrains, a three dimensional map incorporating elevation is required to ensure the robot can safely travel from point to point [13, 14].

In most of this thesis research, the world in which the robot will travel is known with a fair amount of accuracy ahead of time. Obstacles are not considered as part of the planning, but are assumed to be avoided reactively by a separate system. Therefore, a simple planning search such as wavefront propagation is used in most of the path search algorithms developed here, with the obstacles to avoid being shadows cast by terrain features. Battery discharging and charging is also simulated in some path evaluations, and could be incorporated to allow travel through limited duration shadowed regions.

One specialized type of navigational path is coverage patterns, and many researchers have concentrated on this area as well. The most common type of coverage pattern is the straight rows

pattern, where the robot sweeps up and down parallel lines [11, 16, 45, 46]. Obstacles are handled in various ways. In [16], the search area is decomposed into cellular regions, each of which contains no obstacles. The robot performs the straight rows search in each region, using a calculated sequence of moves to optimally link the regions. In [11], when obstacles are sensed, the robot follows the obstacle boundary until the other side is reached, and then continues the search pattern. Similarly, rules are defined in [45] for where the robot should move when an obstacle or sub-inlet is detected, producing an optimal coverage pattern in terms of distance so that minimum overlap occurs. The floor cleaning robot of [46] has several options for turning around when obstacles are encountered or at the end of rows, based on the width of the rows and the capabilities of the robot. This thesis uses similar straight rows patterns, but without the use of boundary, obstacle or inlet algorithms since the regions to cover are assumed to be open, unbounded, and uncluttered. Obstacles are not a major concern, but are avoided reactively by a separate subsystem, and the pattern resumed immediately afterward.

Another type of approach is the contour following pattern [50, 95]. The coverage area is divided between multiple robots in [50], with each covering an area by following its contours. The extension into small corners is done by following a Voronoi diagram. [95] uses a distance wavefront propagation in a grid representation of the environment. The robot starts by moving through all grid cells with the smallest value, moving up to the next higher valued cells after that, and so on. This produces a contoured path surrounding the starting point.

Random searching may be more beneficial than an exhaustive search in some conditions, such as mine-sweeping, where minimizing the number of unseen targets in a limited area is a higher priority than maximizing the total number of targets found [33, 43]. Knowledge of the prior probabilities of target locations also can affect how efficient a given search pattern is. Animals, for instance, tend to follow heuristics more than exact rules when searching for food, as if they were incorporating the effect of uncertain target location probabilities [36].

For robotic exploration in new areas, however, which is emphasized by this thesis, maximizing the area seen is a more likely goal. The area to be searched is not a bounded area, but will in most cases be much larger than a single robot can expect to cover, such as an entire planetary polar region. For these cases, just seeing as much new area as possible is desirable.

Randomized searching within a bounded area or following boundary contours will not suffice when the area has no defined boundaries. One potential mission structure would be for a robot to travel along long-range paths, following the direction of the sun, while performing more detailed coverage patterns in interesting, or randomized, locations to gain a sampling of the region.

The types of coverage patterns studied in the literature is limited, so this work has developed a much larger range of options, where complete coverage is not always the ultimate goal, but the amount of coverage can be one parameter to evaluate, along with other parameters. The evaluation of pattern efficiency is addressed in [33], [43] and [46]. Hofner and Schmidt, in [46], evaluate their floor cleaner's patterns after completion, based on the total time required, the amount covered out of the specified area, the amount of overlap in covered area, and the floor space covered per hour. Similarly, this thesis includes the evaluation of pattern coverage, pattern overlap, pattern completion time, and also power consumption and generation. A goal of this research is in fact to evaluate a range of pattern types primarily for their power generation capabilities, which has not been an evaluation performed in prior work.

The coverage patterns and path searching methods discussed above can select complete paths or just waypoints for the robot to use as subgoals. Calculating a precise path to follow and maintaining that path is another subject. Straight line paths between multiple waypoints, with abrupt heading changes between segments, can be smoothed or modified to better enable robots to follow them [25, 39, 79]. In very rough terrain, each point along the path may need to be checked to guarantee that the robot can be safely positioned in that location [40, 41]. To maintain a given path, the robot needs to know where it is. Dead reckoning can provide a rough estimate, while a human operator or landmarks can provide a more accurate fix from time to time [23, 41, 46, 60, 95]. In fact, a range of human interaction can be utilized, from complete teleoperation to complete autonomy [93]. Accurate positioning information is assumed in this thesis, with path following maintained by comparison of the robot's actual position with its desired position.

The details of all the above techniques are described in the referenced papers, along with descriptions of their implementation on real robots in many cases. This review indicates that a large body of work in this area already exists which could be used and modified for this thesis. However, this research develops a slightly different focus in navigational planning: temporally-

changing, environmentally-based evaluations of navigational plans. This thesis introduces the new concept of evaluating coverage patterns based on the amount of solar, or even wind energy which can be generated, in addition to area coverage, area overlap, total time and power consumption. Maximum resource creation, as opposed to just minimal resource usage or resource constraint satisfaction, is the focus.

C. Mission Planning

Planning to optimize resources is a prevailing concept in robotics research [31]. Some research deals with specific ways to optimize a particular resource, and other research has developed more general architectures which include resource management and optimization. Architectures provide a framework for mission planning, deciding when to do which task. Many architectures have multiple layers, such as a functional level which controls the actual hardware, a planning level which decides what to do when, and perhaps an executive or sequencing level which converts a plan into specific actions.

One multi-layer architecture is the Remote Agent, or RA, developed by researchers at NASA's Ames Research Center. RA was originally designed for spacecraft such as Deep Space 1 [4, 26, 34, 67, 68, 71, 72]. The top layer of this architecture is the Planner/Scheduler and Mission Manager. Human operators give high level goals that need to be achieved in a certain amount of time, while the planner uses temporal reasoning to determine the sequence in which actions need to be taken. Expert subsystems can be used to provide information to the planner, such as the amount of resources that will be consumed during a certain action. The second layer is the Smart Executive, which can select among different ways to accomplish a task given to it by the planner, and requests that the actions be executed in the proper order. It communicates with an additional layer, Mode Identification and Reconfiguration. This is a more reactive troubleshooter which checks to ensure that errors do not occur, and determines what to do if they do occur. Any uncertainty is handled by assuming the worst case.

The spacecraft domain considered for RA has many differences from the mobile robot domain, such as a simpler environment. NASA currently plans to adapt Remote Agent for rovers

[7]. This adaptation considers resource utilization based on estimated profiles of the capacity and demand for the rover's resources. Maximum resource needs of discrete tasks are considered to guarantee that enough power or other resources are available for each task. Continuously assessing the capacity of resources and predicting the demand for them allows dynamic resource allocation and scheduling, optimizing their utilization. As seen from a software engineering perspective, abstract resources are managed as discrete objects, and resource locks are used to manage conflicting demands. The Smart Executive's choice of which task decomposition to pick can be based on a utility value, updated to reflect the environment, current resource availability, and the system state of the rover. Ways for handling environmental uncertainties and the allowable risk are also being studied.

Remote Agent can ensure that resources are not depleted ahead of time, but mainly manages discrete resources or tools which are already present, as opposed to ways to generate new resources such as solar power. Combining aspects of this thesis research with the Remote Agent is a definite possibility, however. For instance, the solar power evaluation algorithm developed here could be used as an expert subsystem in the top decision layer of RA, providing specific information about tasks to be accomplished. The current Remote Agent work only determines the concurrency or ordering of tasks, however, rather than deciding which tasks should even be performed. The resources and tasks considered by RA are also not specifically navigational and are more discrete than the continual evaluation of long-term missions such as coverage of an area. In addition to performing specific pre-developed tasks and satisfying resource constraints, this thesis introduces the ability for a robot to achieve new tasks, such as continually seeking the sun and circumnavigating polar regions, and finding locations based on the current environmental and temporal conditions which will enable it to recharge batteries or communicate with Earth.

Another agent architecture is 3T, which is being developed and implemented in work at NASA's Johnson Space Center [6]. 3T primarily deals with scheduling of resources, and can detect events such as when a resource reaches a certain threshold or is depleted. [26] suggests that this architecture might be effectively merged with the Remote Agent, combining the best properties of both, such as RA's resource management and 3T's user interactivity and adjustable level of autonomy. 3T also has three layers, the planner, the sequencer, and the skill manager.

The planner takes a high level goal, reasons about goals, resources and time constraints, and produces a partially ordered sequence of tasks. The sequencer takes each task and decomposes it into detailed steps necessary to accomplish the task. This layer monitors the state of the world, stopping or starting actions when required. The skill manager contains the actual interfaces with the hardware for each action.

3T has been combined with a Generic Scheduler/Rescheduler for use in crop planning for life support systems [56]. The scheduler reasons about time and dynamic changes in resource availability, including long-term dynamics that are part of a system such as planting crops. User interaction is allowed at various levels, such as manipulating the weights in an objective function used to select the best task, similar to what might be done in this thesis work when selecting the best coverage pattern plan. Replanning occurs when significant deviations from the plan are observed. In [9], a different type of planner replans only when new information about the environment is obtained. This planner uses D* to select the best paths for multiple robots traveling to multiple goals. In some cases, the planner reorders which robot should travel to which goal. Both these bodies of work address dynamic evaluations, but specific tasks or goals are the primary consideration, as opposed to long-term coverage patterns which have a continual affect on power generation and consumption.

Two other architectures of note are the Task Control Architecture [82], and the architecture developed at LAAS-CNRS in France [1, 13, 14, 24, 92]. The Task Control Architecture, or TCA, provides a framework for control and management of multiple processes, hierarchical task decomposition, sequencing and scheduling of actions, resource management, activity monitoring and error recovery, and interprocess communication. Algorithms for specific robots or tasks, such as path planners or real-time interfaces, are not included in TCA. However, the interprocess communication in this architecture is used in portions of this thesis implementation.

The LAAS architecture has three layers again: decision, execution and functional. The decision layer is primarily a temporal planner using the IxTeT software to roughly schedule tasks. A partial plan satisfying temporal constraints for accomplishing a given set of tasks is generated by this layer, considering both attributes of the environment and resources of the robot. Synthesizing original plans is not done, nor is selection at the top level from among multiple plans

and lists of tasks, as is needed in this thesis. The PRS software further refines the tasks by selecting from among a library of plans and procedures, such as elementary robot actions and computing tasks, which can be combined and enacted in order to accomplish a given goal. The version which least constrains the original partial plan is selected. The refinements deal with resolving resource conflicts, eliminating possibly inconsistent events or assertions, and adding subgoals that were not included in the original partial plan. This refining level can optimize paths and trajectories or other characteristics. Different navigational modes, for example, can be selected based on the current terrain [14].

The execution layer uses Kheops to produce executing code from a set of logical rules, providing a link between long scale planning and short term action. The functional layer controls the hardware to interact directly with the environment, using action modules generated by GenoM. Resource management is done at the planning level, dealing with the predictable part of a dynamic environment. Primarily access to hardware resources and amount of available power are the resources which are considered, but potential power generation is not included in the LAAS work. This thesis addresses the evaluation of multiple plans based on their usefulness in meeting various constraints or overall goals, rather than generating the steps which can be used to accomplish a single given plan.

Several other papers deal with the evaluation of tasks. Sukhatme and Bekey use a weighted evaluation of multiple criteria to pick the best configuration design for a physical rover [88], similar to using multiple task evaluation modules [78]. A “systematic, quantitative performance evaluation” for planetary rovers is described in [60]. This is an evaluation of different physical rover types for a single task instead of an evaluation of different task types for a single physical rover. Farritor describes several uses of modular evaluations to pick the best plan of action [28] or the best robot assembly [29]. Software modules include basic actions such as “move this leg forward,” or “use this sensor.” A plan or task sequence is produced based on possible robot actions that can fulfill a given task. The enormous number of plan possibilities is limited with initial heuristics, and then with a genetic algorithm. These plans are evaluated considering a range of constraints, such as power consumption, based on a detailed model of the robot and terrain interaction. However, though the power consumption and terrain models can be a useful

extension to this thesis work, power generation is not considered, and the navigational plans considered are more limited in scale than those considered here. [30]

[34] describes behavior control, where a planner arbitrates between behaviors to determine which behaviors will generate the desired action of the robot. Morignot and Hayes-Roth describe a motivational profile that is used to determine which task a robot should perform [65, 66]. The profile is a set of weights indicating the importance of five different attributes, including physiological needs such as power and the desire to learn and explore the environment. The profile weights can be adapted depending on the environment or human operator choice, and a linear combination of the weights determines which task is the best to perform. Similarly, this thesis can utilize varying evaluation weights to set which evaluation categories are the most important for a given mission.

Standard concepts described here of weighted task evaluation for mission planning and of multi-layer architectural frameworks have been used in the implementation of this work, to allow a concentration on new issues such as evaluating solar power generation. Power generation during task performance as well as coverage pattern tasks themselves are not considered in most prior work. While constraint-based planners which address other resources are available, this thesis combines evaluations and tasks that have not been considered together before. The complicated temporal and orbital mechanics equations which are required to predict the sun's position are combined with the temporal simulation of coverage patterns, providing the structure needed to predict solar power generation. In addition, this research addresses the benefits of having sun, Earth and terrain information onboard an exploring robot, where it can be accessed in realtime to generate new plans and tasks best suited to the conditions.

Chapter III

Methodology

To address the question of how robotic explorers can benefit from sun and terrain information, solar and other environmental attributes have been modeled in a manner compatible with robotic planning. Modeling primarily consists of calculating the orbital parameters of the Earth-sun-moon system, and incorporating terrain features to generate sun and Earth visibility. Robotic capabilities and performance, along with wind effects, are also modeled. Using such models in combination with coverage pattern and path search simulations enables simultaneous evaluation in a range of categories as the simulations are performed. A variety of new coverage pattern types has been developed for these simulations, and evaluation categories useful for selecting more desirable types include attributes such as power generation, power consumption, area coverage, or total time length. For path searches, some useful attributes are power generation and path length.

This chapter discusses the solar ephemeris model, and the application of that model to sun-synchronous and sun-finding path searches. The simulation of coverage patterns is then described, along with evaluation categories and methods for selecting the best pattern. Finally, the implementation of these algorithms in an actual system is detailed.

A. Solar Ephemeris Algorithm

A solar ephemeris algorithm was developed to find the position of the sun at a given time and date, as seen from a given surface location. This calculation has been implemented for both the Earth and the moon in this research. For the moon, the algorithm also finds the position of the

Earth, for communications purposes. The initial stage of the calculation is finding the position of the center of the sun and moon relative to the center of Earth. Many available software programs perform this calculation, one of which is used by this simulation to calculate the right ascension, declination, and distance of the centers of the sun and moon as seen from the center of the Earth, for a given time and date [5]. This information is then transformed to provide the altitude and azimuth of the sun as seen from a given surface location on either the Earth or the moon. The details of these transformations are quite complicated, particularly for the lunar surface, and are given in Appendix B, while Figure 4 demonstrates the basic steps. In the second step, the term "selenographic" refers to moon-centered coordinates, and the transformations involved to reach this coordinate system involve not only geometrical calculations, but also time-dependent calculations concerning the rotation of the moon.

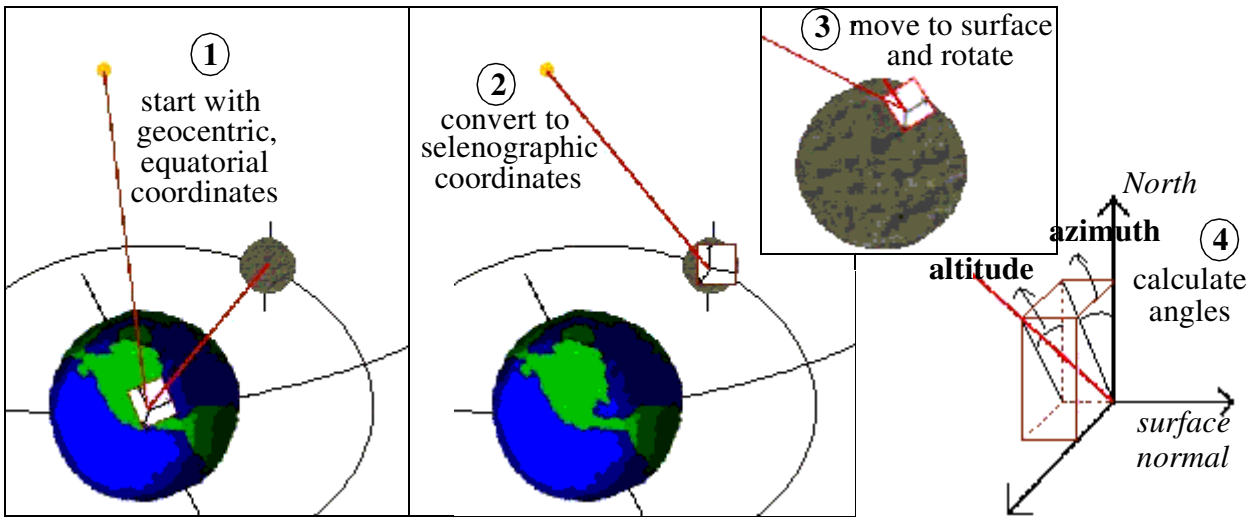


Figure 4: Sun angle calculations for lunar surface

Once the position of the sun and Earth are known, potential shadowing by terrain features must be determined. By comparing surrounding terrain elevations in the direction of the sun (or Earth) from a robot, any occlusions of the sunlight (or Earth line-of-sight) can be determined. Terrain information is obtained from digital elevation maps, which are read into an internal grid-based map when first initialized. The azimuth of the sun is converted into a line of map grid cells along a great circle of the spherical surface, passing through the robot's current position. The elevation angle from the robot's current position to the terrain elevation of each grid cell along

that line is then calculated incorporating the curvature of the sphere, and compared to the elevation of the sun. Note that the current terrain elevation of the robot's position is also included in the calculation. The first terrain feature to block the sun halts this ray-tracing algorithm; otherwise, the algorithm continues to the edge of the known map. Geometrical constraints can be used to quickly determine potential shadowing caused by the remaining body of the planetary sphere, assuming a constant terrain elevation extending past the edge of the internal map. Appendix A contains more details about the grid-based map and reading digital elevation maps, while Appendix B discusses the calculation of great circle lines and angular elevations.

One problem which may occur is that the above calculations assume the ground, and therefore the robot, is flat and tangent to the planetary sphere. For determining only the existence of sunlight, the sun's incident angle on the robot is not needed, but for predicting solar power generation, it is. Robots with moveable solar panels can just adjust panel angles when needed, but this research primarily considers fixed panel configurations. Terrain slopes will violate the flat ground assumption, and invalidate the calculation of the angle between the sun and a robot's solar panels. At runtime this is not a problem as the robot's sensors will give the correct pose, which is incorporated into the calculations. For prior predictions, several solutions are possible. One, a digital elevation map or other remote sensing data can be processed to estimate average terrain slopes, though this may prove inaccurate, particularly with low-resolution map data. Secondly, for regions in which the slope is roughly constant, such as on a hillside, and for which the entire pattern will be carried out in that region, the robot can determine the slope ahead of time with its sensors, and incorporate the value into the sun angle calculations. This is one benefit of having the solar ephemeris library on board the robot, as real-time data can be immediately incorporated into the calculations. Regions with more rapidly varying slopes can be seen as areas with a higher robot pose uncertainty. The results of pose uncertainty on solar power generation calculations have been studied, and are described in the simulations section of the results chapter.

B. Solar Navigation

Solar navigation refers to the concept of searching for or altering navigational plans based on the effect of the sun. One planning goal is finding paths for a robot to follow such that it is always

in the sun and able to generate power with its solar panels. With solar panels properly sized for the power consumption needs of the robot, traveling along sun-synchronous paths enables continual activity. Navigating around a feature such as a mountain while remaining in sunlight requires a robot to keep up with the motion of the sun as it progresses around the feature. For a polar feature, or even the planetary pole itself, a robot can circumnavigate a feature over the course of a planetary day and then immediately repeat a similar circuit, continually driving day after day, as the sun will not set during the summertime at high latitudes. Such a plan will allow the robot to thoroughly explore the region, traversing circumnavigational circuits of varying radii. A hypothetical example of this is shown in the left side of Figure 5. In lower latitude regions, a robot can navigate along shorter paths from sunrise to sunset.

As a second goal, when a robot is blocked from the sun, solar navigation can find the nearest sunlit spot for recharging, and determine for how long such a spot will remain lit. An example of this is shown in the right side of Figure 5, where the blue striped region indicates shadowed terrain. Of course this type of path search assumes that a sunlit location exists within range of the robot. Future research could address the need for the robot to remain aware of when, in location and time, it moves beyond a safe distance from sunlight. For non-Earth locations, the same type of path search can find the nearest locations within line of sight of Earth for communications.

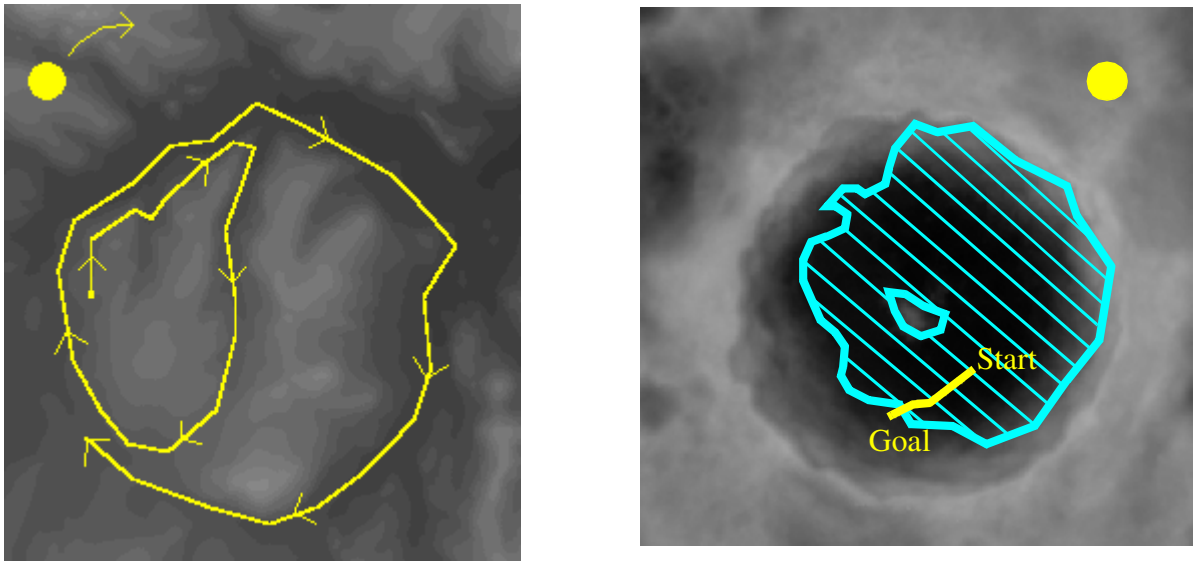


Figure 5: Hypothetical sun-synchronous and sun-seeking paths

One calculation useful to human operators is finding and displaying shadows for an entire region and for a range of times, showing the progression of shadows across the map. A program performing this calculation was developed, using functions from the solar ephemeris library. A digital elevation map is read in, and for a given date and time, the ray-tracing algorithm is performed for each grid cell in the map, calculating whether or not each cell is shadowed. The resulting map is printed out as an image, with shadowed grid cells indicated by solid blue, and sunlit cells retaining their elevation coding with a scaled red color. Partial blocking of the sun's disc can also be determined, indicating whether a cell is at least partially sunlit, generating solid green twilight regions (see Figure 6). A sequence of these shadow snapshots can be combined into an animation, graphically showing shadow movements across the region. These animations assist in locating potential starting points for long-range paths, and in selecting regions suitable for landing sites or primary investigation sites. Understanding the way humans can quickly identify favorable attributes using such animations could lead to better autonomous means of using the information as well.

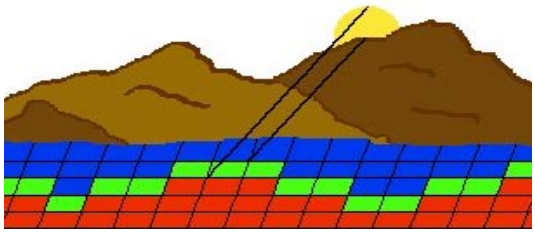


Figure 6: Shadow mapping

SUN-SYNCHRONOUS SOLAR NAVIGATION

The method implemented in this research for sun-synchronous planning, for either single path segments or circumnavigation, involves an 8-connected wavefront propagation search through the two-dimensional terrain space, starting from a given grid cell and time. The existence of shadows in the eight neighboring cells is determined for the starting time plus the amount of time required for the robot to reach those cells, given an average expected speed. Shadowing can be found by calculating at runtime the sun angles and terrain effects for each such cell, or by accessing the appropriate portion of a pre-computed database. Each cell which is sunlit is added to the search

queue, and its neighboring cells are examined for shadowing in turn, incrementing the time again by the expected travel time.

This comprehensive search becomes intractable when considering large or high-resolution terrain maps, however. One simple efficiency improvement which has been implemented is limiting the search by considering only grid cells in the direction of desired travel, and within a certain range of the desired center line of travel. This prevents any backtracking, however, or even pauses in a single cell, which might be necessary in complex terrain areas to maneuver around the changing shadows. Other possible efficiency improvements include generating initial paths with reduced map resolution, or considering only paths composed of shorter straight-line path segments.

An alternative approach is to use pre-computed databases of shadow maps as a three-dimensional search space. Such a database would consist of not only the 2-dimensional aspects of the region, but also a third dimension for time, identifying the existence of shadows for each point and at each time. This initial stage of solar navigation can be done offline, enabling detailed time and memory intensive calculations to be performed. The spatial and time dimensions can be appropriately discretized for the available amount of memory and computational ability, while considering the resolution requirements of the robotic planning. The resulting shape can be visualized as a hopefully connected region of sunlit areas, with blobs of shadows interspersed. This search space can be approached with a number of search algorithms which are more efficient than wavefront propagation, such as potential fields or hierarchical generalized Voronoi graphs [15], producing a viable path from one point to another. These searches must be restricted to only allow travel in one direction, and with a maximum rate of travel, in the temporal dimension, but they are more flexible in that variable robot speeds can be allowed throughout the path. Although such databases have been developed, search algorithms using them have not been implemented.

Difficulties in both methods include deciding with which grid cell to start, and at which time to begin the search. Based on the general current location of a robot, a range of starting times can be autonomously selected by determining the times when the sun is on the robot's side of a mountain, or when the robot's side of a valley is lit by the sun. Analysis of the results from a variety of starting points and times for a particular shadowed terrain region is discussed in the

Results chapter, with some heuristics and trends made evident. For multiple terrain features creating complex shadow movements, the selection of a starting time can be tricky, however. The graphical animations of shadows across the terrain, discussed earlier, can be used in such cases by human operators, who can pick out potential starting times and locations by visually observing the motion of the shadows over time.

Additional complications in designing a path search include considering battery power storage, which would allow a robot to traverse a certain number of shadowed grid cells before regaining the sun. Keeping track of the state of a simulated battery during simulations has been developed, and could potentially be used in the wavefront propagation method, where shadowed cells are allowed to be included in a path if they are traversed when the robot's battery state is high enough. Using a potential field method could be addressed in future work, where the repulsive forces assigned to shadowed regions could be designed such that a robot can actually pass through at least the edges of such regions, allowing a shorter overall path to be found with the small shortcoming of having to go through shadows occasionally.

SUN-SEEKING SOLAR NAVIGATION

A second type of solar navigation is sun-seeking: finding a path from a currently shadowed location to a sunlit location, specifically for a robot needing to find a place to recharge its batteries. A nearly identical application is finding a path to a location in view of Earth to enable a communications link. A different type of map database is generated for this planning challenge - a sunlight (or "Earthlight") endurance map. Two dimensional terrain map grid cells are augmented with a linked list of time periods. Each time period contains fields for the starting and stopping times, as well as the state of sunlight or shadow. In most cases, these periods will be few in number and fairly continuous, due to the regularity of the motion of the sun. A two-dimensional wavefront propagation search in this case seeks the closest grid cells which will be lit by the sun when the robot arrives. The time at which the robot is expected to arrive at a cell is compared to the cell's linked list to determine in which time period that arrival time falls, and whether that time period is sunlit or shadowed. If sunlit, the period stopping time is consulted to determine the amount of time left in that location before the sun is occluded. If the remaining

sunlight time is not sufficient for the robot's purposes, the search continues until a suitable location is found. Care must be taken in performing a sun-seeking search, however. Expanding a search outward only from the starting point is not complete because there may be pockets of light that suddenly appear behind the wavefront as time progresses. Sunlight may be reached quicker by going a shorter distance and then waiting for the sun. A complete, though more memory-intensive search must also consider moving to grid cells "behind" the wavefront, or staying in place in a single cell as time progresses. Another use of such endurance maps is determining when a robot's current location will be shadowed for an extended time period. During such times, the mission planner may choose to shut down non-critical equipment and hibernate until higher power levels can be restored.

One further application of this research is a visibility integration map. While not a navigational planning tool, this map can enable the discovery of important information. The cyclical motion of the Earth, sun and moon is fairly similar from year to year. In fact, the cycle of orbital parameters repeats exactly about once every 18 years [75]. By integrating the amount of sun visibility at each lunar location over a lunar day, over a full year, or over 18 years, time-averaged sun visibility maps can be created which describe the percentage of the time interval that each area was illuminated. Similar maps can be created for Earth visibility for communications. Combining such visibility integration maps for both sun and Earth visibility can enable the identification of sites with particular characteristics which make them suitable for human or robotic base operations. Regions with high visibility percentages imply a potential for high solar power generation and regular communication with Earth. Regions of zero sun visibility percentages near polar regions indicate the potential for cold traps harboring water ice, which can be utilized to support a human presence or in-situ rocket propellant production facilities.

C. Coverage Pattern Simulation

In addition to long range sun-synchronous paths, and short range sun-seeking paths, the intermediate range of coverage patterns also benefits from sun and shadow information. Four general types of coverage patterns have been developed and implemented for this work, with a number of different variations possible for each type. Starting angles, direction of travel, width

between rows, and length of rows are some of the options. A complete list is given in Appendix A. A subset of coverage patterns and initial parameters is selected ahead of time by a human user, or potentially by an autonomous selection algorithm, which are appropriate for the robot's mission and current conditions. For instance, the width between rows is usually set to the distance the robot's sensors can see, ensuring that the area between rows can be completely covered. The length and number of rows, or size of patterns, is generally set to fit the size of the desired region to cover, or to last a desired amount of time. Heuristics for selecting other parameters are described in the results chapter. The chosen subset of coverage patterns is simulated step by step, incrementing the time index as the simulation progresses to enable the incorporation of temporal effects, and the resulting changes in sun position and shadowing.

Each general pattern type has specific characteristics that may prove more or less desirable given the mission and the environmental situation. By simulating each type of pattern under the relevant solar and terrain conditions, the best pattern for the conditions can be found. This thesis concentrates on exploration in relatively uncluttered, unbounded environments, where obstacles are infrequent, and can be avoided reactively without much distortion of the path plan being followed. The effects of shadowing are then a primary distinguisher between pattern types, along with power consumption and area covered. Brief descriptions of the main pattern types are given below, and sample images of the simulations are shown in Figure 7.

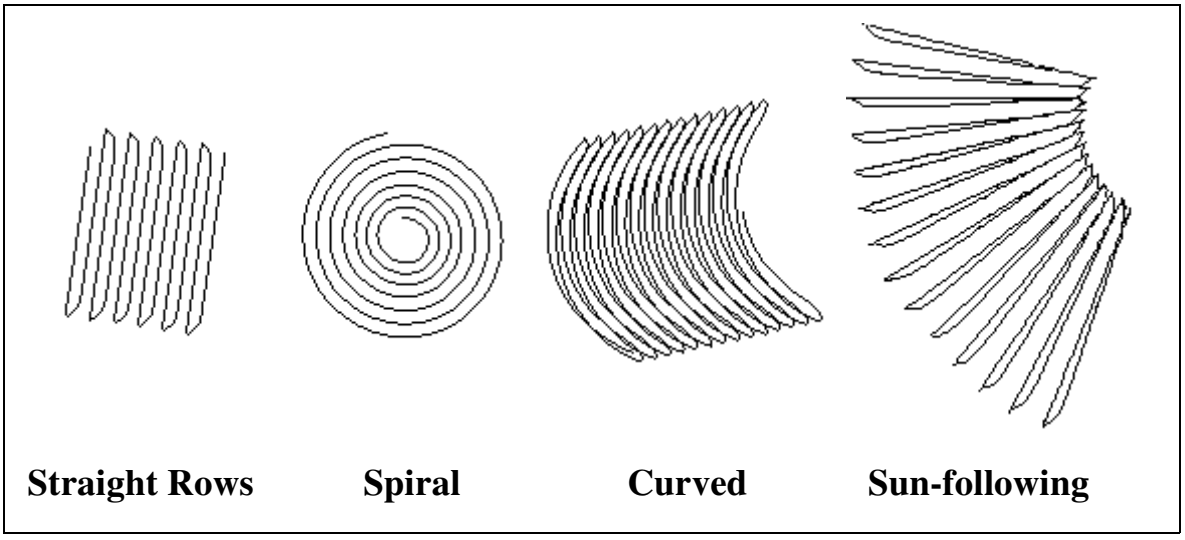


Figure 7: Coverage patterns

The first type of coverage pattern is a simple straight rows pattern, where a robot drives back and forth in parallel tracks. This is the most common method used for covering a region. At the ends of the rows, a robot can either perform a point turn and gradually merge back into the next row, or else turn in a circular arc. When the rows are narrow compared to a robot's minimum turning radius, the row end circles can bulge out to allow the robot to make a more gradual turn, as shown in Figure 8. All types of end turns are designed to take place just outside the area of desired coverage, as the turns and row merging prevent a robot's fixed sensors from seeing the entire portion around the turn. Evaluations of power consumption, power generation, and total completion time determine whether performing point turns instead of circular turns will improve the performance, given the desired width of the rows. The feature of this pattern which makes it so commonly used is that it covers an area very efficiently, with very little overlap. Similarly, the spiral pattern covers a circular area very efficiently. The main variation for that pattern is the direction of travel: clockwise or counterclockwise.

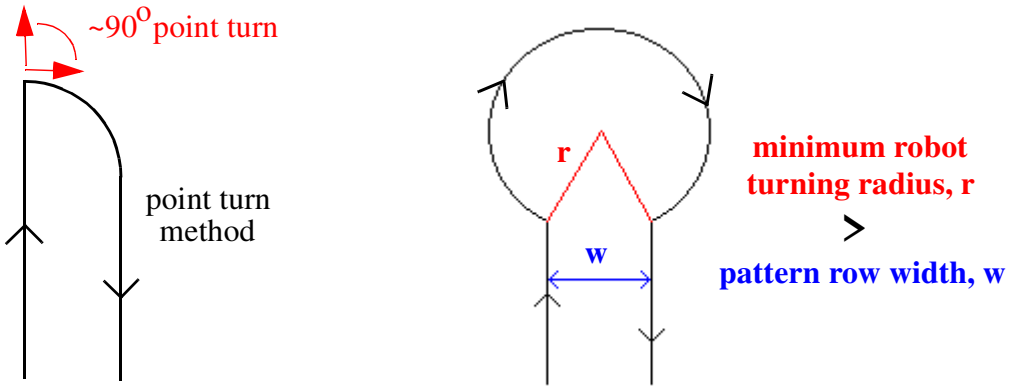


Figure 8: Row ends for straight rows pattern

Curved patterns are a variation of the straight rows pattern, where the rows are allowed to have any desired radius of curvature. While the straight rows and spiral patterns have minimal area overlap, the curved pattern can be defined with row lengths, curvatures and direction of travel that produce more or less overlap. By evaluating a range of curved pattern types, trends in the numerical evaluations of area coverage and overlap, as well as other evaluation categories, can be linked analytically to the row curvature and length.

For a polar location in the summer, the sun is always visible, circling around the horizon at a low elevation. A robot with vertical solar panels on its sides can maintain an optimal orientation with respect to the sun by turning just enough to keep up with the sun's rotation. A sun-following pattern is thus similar to a curved pattern, but with the row curvature defined by the sun's rotation. While the solar power generated will be maximized for this pattern type, other evaluations such as area covered and area overlapped will definitely be less optimal than for other pattern types.

D. Evaluation Categories

As coverage patterns are simulated, evaluations of various attributes are made simultaneously. The patterns are defined incrementally, simulating a robot moving along the pattern one (simulated) second at a time. A structural loop is initiated in which the pattern-following function is called once during every pass through the loop, until the loop is terminated by the completion of the pattern. This structure enables sun and shadowing information to be calculated for the time at which a robot would reach each point in a pattern, adding in temporal effects which are otherwise difficult to determine analytically. Evaluation functions are called once during every pass through the loop, keeping track of desired attributes throughout the performance of the task [78]. For instance, the evaluation function for the total time required to perform a task is a single step: incrementing the running time total by one second each pass through the loop. The other evaluation algorithms are more complicated, and include functions for solar energy generation, wind energy generation, energy consumption, area covered, and overlap of covered area, all of which will be described below.

The coverage pattern algorithms are not only used for simulating a robot's motion, but are also used in commanding a robot to follow a given pattern. During actual motion, both real telemetry and evaluation function results are recorded. As opposed to *a priori* simulations, where evaluation results are generated in a fraction of the actual task time, *concurrent* simulations evaluate the patterns as they are being performed. These evaluations are compared to actual telemetry, to aid in calibration of models, generation of statistical data, and comparison with ground truth. The sections below describe both the evaluation functions and the actual telemetry recording for each category, where relevant.

SOLAR ENERGY GENERATION

Solar energy generation is both simulated and recorded. Actual solar power generation by the field work robot's solar panels is recorded and compared to concurrent simulations based on solar panel configuration, actual robot pose, and the solar ephemeris [80]. A priori simulations are also run using simulated pose values. The primary values needed to calculate solar energy generation are the incident angle of the sun on a solar panel, and the amount of power which can be generated by that panel. Details such as reflection and atmospheric diffusion of light are not included in this simulation. Such details could improve the fidelity of the simulation, but are not required for the comparative results desired in this thesis, as they in general produce only scaling effects during the nearly constant environmental conditions present during a single coverage pattern enactment. Fixed robot solar panels are assumed, though portions of this evaluation algorithm can also be used to calculate the optimal orientation for movable panels.

Calculating the incident angle of the sun on a robot's solar panels requires knowing where the sun is relative to the robot's location. This position information, along with terrain shadowing information, is obtained from the solar ephemeris algorithm described earlier, producing the vector to the sun from the robot's surface position.

$$\overrightarrow{\text{Sun}}_x = \cos\left(\frac{\pi}{2} + \text{azimuth}\right)\cos(\text{altitude})$$

$$\overrightarrow{\text{Sun}}_y = \sin\left(\frac{\pi}{2} + \text{azimuth}\right)\cos(\text{altitude})$$

$$\overrightarrow{\text{Sun}}_z = \sin(\text{altitude})$$

The geometrical normals to each solar panel, relative to the robot, are then rotated based on the robot's current pose, producing the normals in world coordinates relative to the sun. The rotation matrix is shown below.

ρ = robot roll
γ = robot pitch
λ = robot yaw, 0 is +y direction

$$\overrightarrow{Rot}_{xx} = \cos\lambda\cos\rho$$
$$\overrightarrow{Rot}_{xy} = \cos\lambda\sin\rho\sin\gamma - \sin\lambda\cos\gamma$$
$$\overrightarrow{Rot}_{xz} = \cos\lambda\sin\rho\cos\gamma + \sin\lambda\sin\gamma$$

$$\overrightarrow{Rot}_{yx} = \sin\lambda\cos\rho$$
$$\overrightarrow{Rot}_{yy} = \sin\lambda\sin\rho\sin\gamma + \cos\lambda\cos\gamma$$
$$\overrightarrow{Rot}_{yz} = \sin\lambda\sin\rho\cos\gamma - \cos\lambda\sin\gamma$$

$$\overrightarrow{Rot}_{zx} = -\sin\rho$$
$$\overrightarrow{Rot}_{zy} = \cos\rho\sin\gamma$$
$$\overrightarrow{Rot}_{zz} = \cos\rho\cos\gamma$$

As an example, the normal vector for the panel on the field work robot's left side, which is 40 degrees from vertical, in robot coordinates, is:

$$\overrightarrow{Panel}_R = (-\cos(0.698),\ 0,\ \sin(0.698))$$

The rotated normal, in world coordinates is then:

$$\overrightarrow{Panel}_W = \overrightarrow{Rot} \times \overrightarrow{Panel}_R$$

The incident angle of the sun on the panel is just the dot product of the two vectors:

$$\theta = \overrightarrow{Sun} \bullet \overrightarrow{Panel}_W$$

The solar power then varies directly with the cosine of the angle between each panel's normal and the direction of the sun.

$$\text{power} = \cos\theta \times \text{max power potential/solar panel}$$

WIND ENERGY GENERATION

Wind speed and direction can be recorded during field tests, which are the primary variables needed in order to simulate wind power generation, as indicated by the equation below.

$$\text{power} = \frac{1}{2} \times e \times A \times d \times v^3 \times \cos\theta$$

where

e = efficiency,

A = turbine area,

d = air density,

v = wind speed, and

θ = relative angle between turbine normal and wind direction. Note that if this angle is greater than 90 degrees, the wind turbine will not generate any power.

Air density can also be recorded as telemetry, but varies so gradually that a constant can be assumed in many cases. Air density varies with humidity, temperature, and altitude, so for a given location, time of day and time of year, these values are relatively stable. For the other parameters, a particular turbine configuration must be used. The efficiency is assumed to be a constant 25% in most of the simulations performed for this thesis. Betz' law says that the maximum efficiency of converting wind kinetic energy to mechanical energy is 59%. Many standard wind turbines have averages slightly above 20%, but that number varies with the wind speed. In many cases, wind turbines are designed to have their maximum efficiency at the most frequently expected wind speeds. For locations where the average wind speeds are relatively constant, assuming a constant efficiency is valid for simulations. [21]

The primary question for validating the simulation of wind power generation is whether or not the wind speed and direction can be predicted accurately. The power varies as the cube of the wind speed, so this is a value which should be known quite well for an accurate prediction. Based on field data from Elephant Moraine in Antarctica, the wind direction and speed do appear to be predictable in this location. The wind nearly always blew from the south, and the wind speed varied with an approximately Gaussian distribution, for a given day. Data from over nine hours

on one particular day are shown in Figure 9. The recorded wind speeds at this location were 13.6 mph on average, and on nearly a third of the days the average was greater than 18 mph, promising high potential power generation. So for this location, at least, predicting wind power generation does seem to be feasible, when the expected winds for a given day can be estimated at the start of the day. The predictability is not as good as for solar power, but the potential power gain can be great, so even rough estimates can provide useful information.

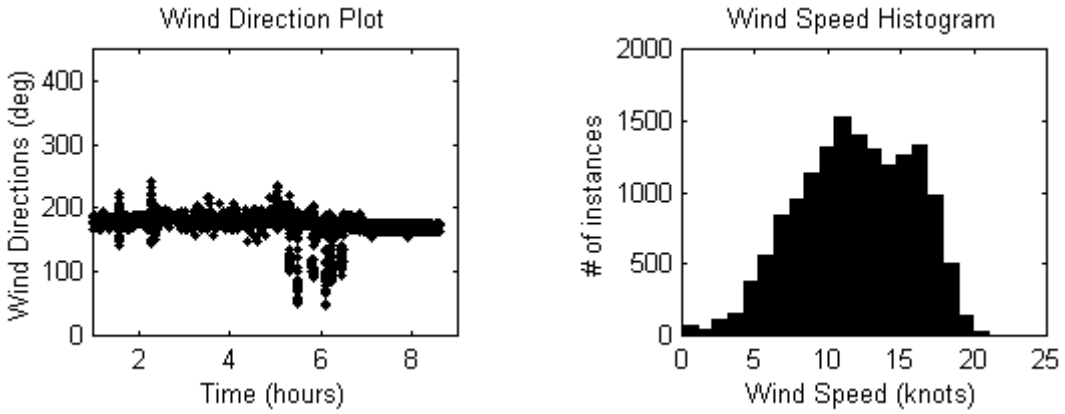


Figure 9: Wind values on January 25, 2000, at Elephant Moraine

Given the wind speed and direction then, either from actual or simulated data, the potential energy which can be generated is calculated. A wind simulator provides randomized values for wind speed and direction, producing a Gaussian distribution of values with given widths and given predominant values, and the remaining parameters in the power generation equation are based on a model of the desired wind turbine and the primary location.

ENERGY CONSUMPTION

The current evaluation function for energy consumption is based solely on statistical telemetry data from field tests. Recorded telemetry includes steering and wheel motor currents, and power source voltages. The currents are multiplied by their supplying voltage to produce the power consumed, which can be categorized and combined into locomotion power, steering power, or total power. Since the evaluation function is expected to be called once a second, the power can be converted to energy consumption by assuming the power draw is constant during this time

interval. While in general just the total energy consumption is recorded, maximum power draw can also be recorded to assist in understanding the power needs of a robot.

To develop the simulation of energy consumption used during task evaluations, recorded telemetry from 12 different days was used, from several locations, totaling approximately 9 hours. The data consisted of coverage pattern tasks, when the robot was continually moving with various turning radii, over varied terrain. First, the values were filtered, removing any voltage and current spikes. After multiplying the respective currents and voltages and combining them into locomotion and steering power, the resulting power values were still quite noisy, so the values were smoothed with a Gaussian filter, iterated five times. The filtered values were then split into several sets of categories for comparison: a point turn vs. non point turn set, changing turning radii vs. maintaining turning radius set, a set of eleven categories of turning radius, a set of three categories of robot pitch, and a set of three categories of robot roll. The variations between the categories in the set of eleven turning radii, and in the set of three robot roll values, were not significant compared to the standard deviations within each category, but some trends could still be discerned in the other sets.

Based on these trends, the energy consumption simulation takes information about a robot's pose, commanded turning radius, and previously commanded turning radius, and produces a total energy consumption value for that pass through the task loop. The baseline energy consumption value includes both locomotion and steering motors, and for the model based on the robot used in field work, that value is 355 J. If the commanded turning radius differs from the previously commanded radius, then an additional value is added, compensating for the steering motors' efforts to switch turning radii. If the commanded turn is a point turn, then another energy adjustment is made, reflecting the increased power needed to transform into the point turn configuration or to overcome any ground friction. For the field robot, changing the turning radius only consumes an extra 15 J, while performing a point turn requires an additional 88 J per second.

In addition to simulating robot location and heading, the pitch and roll can also be simulated, with a Gaussian distribution of a defined size around a defined primary pitch and roll (usually 0 degrees). If no pose variations are desired in the simulation, the width can be set to 0. In field tests, the observed pitch and roll did not vary considerably, so the entire range was split into only

three categories: high values above 1.5°, low values below -1.5°, and all values in between. The roll values did not produce significant changes in energy consumption, but the pitch values did, either increasing or decreasing the required energy consumption. A pitch value above 1.5° required an additional 60 J from the field robot, while a pitch below -1.5° required 60 J less than the baseline.

For simulations in this work, either the actual values consumed by the field robot are used, or a different set of power values scaled up or down is used in order to simulate a different power input/output relationship. For different robots, or on different planets with different gravitational factors, the power values will need to be different. In addition, these power consumption categories simplify the evaluation algorithm, and may not be appropriate for all terrain conditions or robot configurations. However, the model used here was developed to demonstrate the simulation approach and to understand some general variations between pattern types due to steering activity. Other robots and missions will of course need an appropriate model of their own. More complex models can be easily implemented and included in this structure, as long as a power consumption value can be calculated for the existing robot, steering configuration and terrain conditions during each discrete time interval of the simulation. An evaluation of the instantaneous or total energy consumption can then be produced.

AREA COVERAGE AND OVERLAP

The amount of area covered by a robot depends on the "footprint" of the sensor being used, and must be calculated based on the current position of the robot and the size and shape of the sensor footprint. If "covered area" refers to area physically driven over by a robot, then the footprint is just the size of the robot itself. Generally, covered area refers to area that has been observed by a sensor such as a camera. In this case, the footprint depends on the camera's field of view. The area coverage algorithm takes a robot's pose, whether simulated or actual, and calculates which grid cells in the internal grid-based map are seen by a camera with a given field of view, marking them as having been seen. This assumes perfect pose knowledge, to within the resolution of the internal map. To calculate the incremental area covered, the algorithm sums the

areas of all currently visible grid cells which have not previously been marked. This incremental area is added to a running total, which will be recorded at the end of the pattern.

The amount of area overlap refers to the area which has been seen more than once. This can be a measure of the inefficiency of the task, since the area may only need to be seen once. In other cases, some amount of overlap may be desirable, such as with imperfect sensors which may miss targets the first time through a region. To calculate the overlap, the same area coverage algorithm sums the areas of the currently visible grid cells which have already been marked. To avoid the problem of this evaluation function being re-called before moving to a new area, the time at which cells are marked are also recorded each time a cell is seen. If a cell has only recently been marked, then that cell is not included in the summing of overlapped area. The recency is defined to be approximately 5 seconds, which is based on the size of the robot's sensor footprint and the speed of the robot, such that the robot's footprint will not overlap with the footprint from 5 seconds ago for the expected row curvatures.

E. Task Selection

While some coverage patterns are nearly always better than others in certain evaluation categories, no one pattern is the best in all areas. Straight rows and spiral patterns cover the area more efficiently, but the sun-following pattern generates more solar power. Variable curvature patterns produce a range of values in all categories. Selecting the best pattern for a given mission is then not a simple task. The importance of each evaluation category must be considered when deciding which pattern's attributes are the most desired.

A representative set of coverage patterns can be chosen for evaluation, given an approximate area to cover, a starting time, and in some cases a starting location. These coverage patterns are then evaluated by performing simulations of each. During each simulation, evaluations are performed of the desired categories of interest, such as potential solar power generation, wind power generation, power consumption, area covered and overlapped, and total time. To select the best plan based on these evaluations, more information is needed about the available resources, and the desired goals. A single evaluation category can be the determining factor, such as the area

coverage. The pattern which covers the most area would then be selected. This method is used in a mission planning implementation, described in the next section, where the desired primary evaluation category can be selected by the user. A weighted combination of evaluation categories can also produce the rankings for each pattern, such as a combination of power generation and power consumption. The weights for each category can be determined by the user ahead of time, based on the desired characteristics of the mission. This method is used in some simulated tests. Alternatively, the weights could be adaptively altered over time to fit the current state of the robot, or the progression of its mission [78]. In other cases, a few primary evaluation category results can be tested to see if a plan passes certain cutoff criteria, while the plans surviving such tests can then be ranked by the remaining evaluation categories. This third method is also used in some simulated tests. All of these methods are demonstrated graphically in Figure 10. The method which should be used for a given situation depends on the mission criteria, and the evaluation categories of interest.

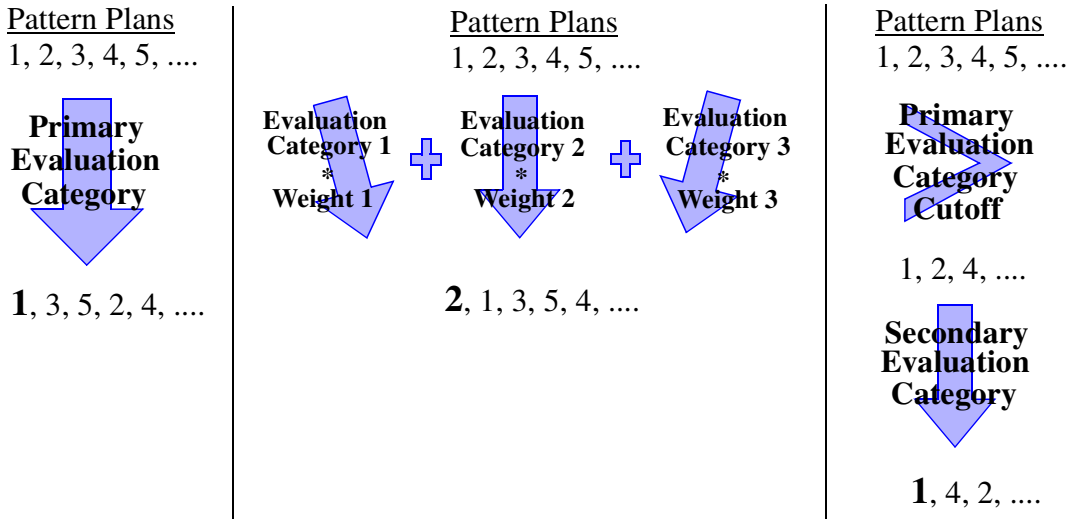


Figure 10: Three methods for sorting and selecting plans

Selecting the representative set of patterns to evaluate can be difficult. An exhaustive set of all possible types would be infeasible to simulate, particularly in a realtime implementation. The time to simulate a pattern is only a fraction of the time required to actually perform a pattern, but it is still a non-negligible amount of time. However, multiple simulations, with varying sun, terrain and temporal parameters, have been performed to gain an understanding of the different

pattern types and their results in a range of conditions. Results of these simulations are discussed in Chapter V, where a subset of pattern types can be identified which span the range of possible evaluation results. An alternative heuristic is to avoid simulating the patterns at all. This requires finding analytical equations which produce final evaluations of each category based solely on the pattern and other parameters. These equations can be used to quickly generate evaluation category totals without the time cost of simulating a complete pattern. A discussion of this approach, for variable curvature patterns, is given in Chapter V.

The validity of evaluating simulations to use in planning real actions can be questioned, especially when considering plans which occur over long periods of time and varying terrain, such as coverage pattern plans. Simulation accuracy is examined in the experimentation and results chapters. Even if the accuracy is fairly high, though, problems may occur when the planner trusts a simulation to make the critical determination if enough power can be generated to survive a mission. Uncertainty in terrain, and thus robot pose and shadows, can affect the amount of solar power generation. As power generation is a vital concern for a self-powered robotic explorer, underestimating this quantity can result in an abrupt end to its active life.

There are several alternatives to depending solely on simulated evaluations, however. First, the evaluations can indicate a range of possible values, directly incorporating the uncertainty. A pattern evaluation can indicate the expected power generation, but also the minimum and maximum values with a certain degree of confidence. The planner can take into account the minimum values, if so desired. A variation of this idea was tested as part of this research, where the chosen plan must have a desired percentage of excess power generation capability, as compared to the robot's power consumption. Second, the planner can proceed with a more risky plan, knowing that two options are available if power runs low. One, the robot can hibernate, shutting down all but the most vital systems, waiting until sunlight reappears to charge its batteries. Two, when energy resources drop below a certain level, the planning software can find the nearest sunlit spot as discussed earlier, where the robot can recharge itself and then resume its task. Although these options are not guaranteed to enable a robot's survival, they provide a greater chance of survival while a robot attempts plans with greater potential returns.

F. Implementation

Searching for meteorites in Antarctica is a time-consuming, repetitive activity in a cold and remote environment. Humans perform this difficult work every year, but a robot designed to explore this area can provide great scientific returns, assisting and expanding the potential of human teams. Exploring the Antarctic also provides information about geologic processes, such as the formation of moraines [79]. Meteorite searches are frequently conducted in and near moraines, where wind strips off the top layers of ice flows blocked by mountains or other obstructions, revealing concentrations of meteorites and terrestrial rocks [12]. A project to develop a robot for this work provided a suitable platform for the implementation of this thesis research, incorporating coverage patterns, navigational planning and a polar location.

The navigational and sensing tasks which are needed for meteorite hunting can benefit from knowledge concerning the sun and terrain. In the Antarctic summer, the sun is visible 24 hours a day, circling around the sky at a low angle above the horizon. Long shadows can be created by even small terrain features, constraining solar-powered robotic explorers. For example, Figure 11 shows an aerial view of a mountain range with a maximum elevation of 390 meters above the flat plain, causing shadows up to 1.6 kilometers in length. Steady winds are also common, continually coming from one primary direction. Thus, the sun and the wind are two potential power sources for a robotic explorer [78]. In addition, solar information can also be useful for understanding lighting conditions which affect image processing.

Figure 11: Aerial view of shadows in Antarctica

SOFTWARE ARCHITECTURE

Much of the software developed for this thesis was written to be used by a specific robot, called Nomad, so all new work had to be incorporated into the software architecture already in place. Nomad's locomotion is physically controlled by an onboard Realtime System, which communicates with the Network Data Delivery Service (NDDS) communications protocol [74]. Navigational commands sent to the robot must be in the format of a desired turning radius and speed. These commands can be sent directly to the Realtime System, but in order to utilize the already-existing obstacle avoidance software, such commands must be sent through an arbitration module. The Arbiter receives votes from various modules, communicating via Task Control Architecture (TCA) messages [89], as to which turning angles they prefer out of a specified set of angles. It then selects the best direction based on those votes, and sends that command to a controller module, called CTR, which converts the message to the NDDS protocol and passes it along to the Realtime System. Obstacle Avoidance is one of the modules which sends votes to arbiter. It communicates with sensors such as a laser and stereo cameras, which indicate the certainty of obstacles in a given range in front of the robot, and then calculates the turning angle which the robot must follow in order to avoid the obstacle. It can veto specific angles when they will cause a collision. [64] Any new navigational planning software must then generate commands as a series of turning angles, and send its own votes to Arbiter.

In addition to working with the existing architecture, new software needs to incorporate the task requirements for Nomad's new mission -- the search for meteorites in Antarctica. Human meteorite search teams typically employ patterned searches on Antarctic ice fields and moraines. The team members spread out along a line just within view of each other and proceed straight ahead, sweeping through the entire area [2]. The method for a robotic search employs the same idea, using coverage patterns to ensure that the robot's cameras see all potential meteorites. Planning software for this mission needs to communicate with new science sensors which search for and examine potential meteorites, communicate with the Realtime System to acquire telemetry data, communicate with users to be given specific search parameters, plan navigational tasks ahead of time without pausing or blocking current activities, and then enact navigational plans by sending turning angle votes one by one to Arbiter.

To meet these requirements, the new software was split into several pieces: planning, navigating, sensing, and communicating with users. The result was a three-level architecture, often used for robot control [35]. The Mission Planner is the highest level, controlling unit, which communicates with everything else, and is the link between the robot and user via a text-based User Interface. The Mission Planner performs all time-intensive planning and task simulations to determine the best course of action, decomposing each mission into specific commands for various other components. On a level below that, the Sensor Manager receives all science sensor commands from the Mission Planner, enacts the commands, and sends back status reports. The Target Acquisition module is a specialized sensor manager, with the single continual task of looking for target objects. The Mission Planner can ask it to either start target acquisition or stop. In this implementation, it utilizes a single sensor -- the high resolution camera -- and performs a segmentation algorithm on the images it receives once every few seconds. Both the Sensor Manager and the Target Acquisition modules enact sensor commands by requesting action from the level below them, which contains sensor controllers and the sensors themselves. Any data the sensor managing modules receive are put into a Database for storage. A Classifier module can then access the Database and analyze the new data to determine the probability that the rocks examined were meteorites, putting those probabilities back into the Database. The Mission Planner accesses the Database to alert the user if any high-probability meteorites are found.

Parallel to the Sensor Manager on the second level, the Navigation Manager receives all navigational commands from the Mission Planner. It translates the commands into navigational plans and then into turning angles for Arbiter, sending back status reports when tasks are done. The planning and simulation library, indicated in Figure 12, is used by both the Mission Planner and the Navigation Manager, as it contains the task knowledge needed both for planning and enacting tasks. This is the module which encapsulates all the coverage pattern and evaluation algorithms described earlier in this chapter. The level below the Navigation Manager and Arbiter is similar to the sensors level, containing the Realtime System which communicates directly with the robot and its driving and steering actuators. The Navigation Manager can also pass along telemetry data from the Realtime System in return. A Monitor module is somewhat outside this hierarchical structure, as are the Database and Classifier. The Monitor simultaneously listens to most of the other modules, recording data in several formats for later human processing and study.

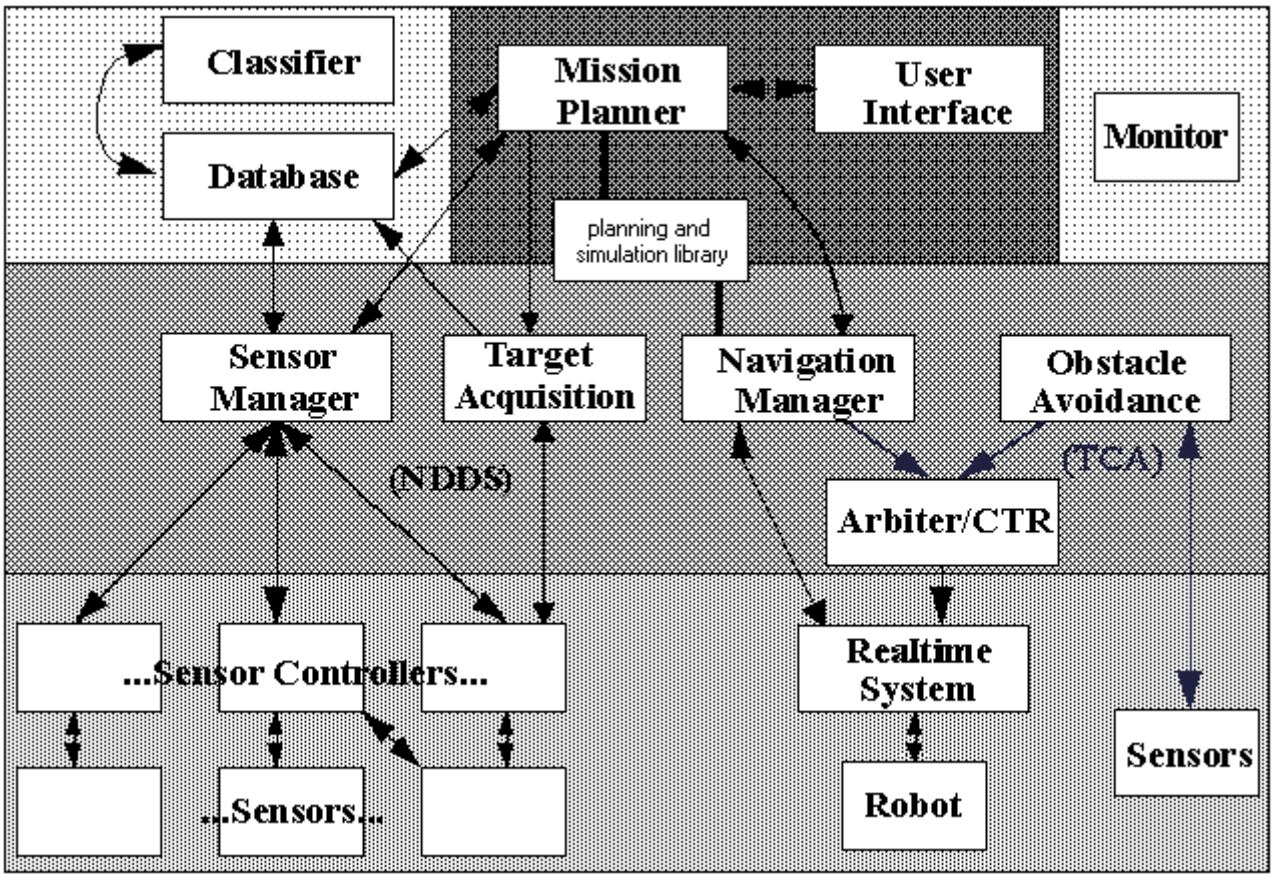


Figure 12: Nomad software architecture

The Sensor Manager, Target Acquisition, Database and Classifier modules, as well as the sensor controllers and sensors, will not be discussed further in detail, as these modules were not part of this thesis work, but developed mostly by others for the meteorite search project [3, 70]. Similarly, the Realtime System, Arbiter, CTR and Obstacle Avoidance modules will not be discussed, as they were primarily developed earlier in other research as well [64]. However, the implementations of the Mission Planner and Navigation Manager were done as part of this research, and will be described along with their interaction with the planning and simulation library, which is the core module from this thesis work.

The Mission Planner takes high level commands from the user and determines how best to perform them. These commands can be individual navigational tasks, individual sensor tasks, or a more complicated mission such as a meteorite search. The planner has an internal model of the

robot which is used as to simulate and evaluate navigational tasks ahead of time. Additional responsibilities of the Mission Planner include modifying the mode of operation to allow various levels of user teleoperation, processing notifications by the user of teleoperated or manual actions completed outside of the Mission Planner's knowledge, and performing several safe-guarding features. As one such feature, the planner monitors the current stage of activity, to enable it to decide if a user command or notification is currently allowed or not. For instance, commanding the robot's arm is not allowed if the robot is currently enacting a navigational task or if its wheels are not straight, for safety reasons concerning potential collisions of the arm with the robot's wheels. As another example, a user notification of a completed sensing task will be ignored during a search if the planner is not currently waiting for such a notification. If a search is interrupted by the user, the planner is able to save the state of the search so that it can be resumed at any later date. The Mission Planner also keeps track of the time in order to tell whether the robot appears stuck in an action, such as waiting for a sensor to complete its task, and thus needs to be interrupted. Reports on activities are regularly sent to the Monitor.

For search missions, the planner initiates navigational and sensor activities and, based on feedback from the rest of the system, such as current robot position and sensing task success, determines the sequence of steps to take in order to complete the mission. The Mission Planner evaluates the costs and benefits of performing different tasks, such as using different types of sensor on different targets, and determines the best course of action, given the goals of the mission. The planner can also consider sun, terrain and temporal parameters and how they affect the robot, and decide a course of action which will improve the overall performance, though these parameters were not used as much in the autonomous meteorite searches as in later simulations. Planning ahead is needed in many cases, so the Mission Planner runs concurrently with lower level modules such as the Sensor Manager and Navigation Manager. It can simultaneously evaluate potential sensing tasks, listen for status reports from other modules, and listen to the user in case an interruption in the mission is desired, all while a previously commanded navigational task is being carried out.

SEARCH DETAILS

As a specific example of the general architecture described above, the basic sequence of tasks that comprise an autonomous meteorite search mission are listed below. This sequence is a hardcoded control loop developed for this mission. Varying levels of teleoperation can be achieved by allowing the user to take control in steps 2A, 3Bii, 3D, 3E, 3G and/or 4, notifying the Mission Planner when complete. If any of the autonomous steps are not completed before a designated timeout occurs, the Mission Planner will indicate an alert, and revert to a more user controlled state. The length of a timeout period is set to five minutes for autonomous camera usage, ten minutes for autonomous arm usage, and is calculated to be twice the predicted length of any planned maneuvers. The steps that require cost and benefit evaluations (steps 1, 2C and 3B) will be explained in more detail afterwards.

1. Request coverage pattern from Navigation Manager, and start Target Acquisition.
2. While performing pattern, continually:
   A. Listen for new target notifications from Database, to add to internal list.
   B. Listen for any overriding user commands.
   C. Evaluate all targets in internal list to determine if any should be examined now.
3. If new target is found, or if step 2C is positive:
   A. Request pattern interruption and stop from Navigation Manager, and stop Target Acquisition.
   B. Evaluate all targets in list.
      i. Request costs and benefits for each target/sensor pair.
      ii. Calculate evaluations for each pair, and pick highest ranked pair.
      iii. If no pairs are feasible, skip to step 4.
   C. Check if the robot is within the chosen sensor's workspace for the chosen target.
   D. If not within workspace, maneuver.
      i. Plan maneuver to reach the workspace.
      ii. Request Navigation Manager to enact planned maneuver, and listen for completion.
      iii. Re-check that the robot is within the workspace. If not, repeat step 3D.
   E. Request Sensor Manager to take sensor measurement of target, and listen for completion.
   F. Process return status from Sensor Manager:
      i. If status is "redo sensor," and number of sensor attempts is less than limit, return to step 3C. If too many attempts, consider status to be "failed."

ii. If status is "failed," set evaluation cost such that sensor/target pair will be ignored. Return to step 3B.
iii. If status is "successful," continue to step 3G.

G. Determine new information gain value for sensor/target pair, depending on desired mode of operation:
   i. When Database indicates that Classifier is done analyzing sensor measurements, request new information gains from Database, or
   ii. Set information gain to 0 for that sensor/target pair, indicating that no new measurements should be taken.

H. Return to step 3B.

4. When all suitable target/sensor pairs have been examined, request resumption of coverage pattern from Navigation Manager, restart Target Acquisition, and return to step 2.

In step 1, the coverage pattern is selected. In the implementation tested in the field, this pattern selection is mostly done by the user. The user picks the type of coverage pattern, where and in which direction to start, and the dimensions of the pattern to perform. With prior knowledge of the field of view of the Target Acquisition camera, the Mission Planner then determines the necessary width between pattern rows to insure that no area is missed.

In an expanded version of this step, which has been tested in simulation, the Mission Planner is only given a starting position, not including orientation, and the total time allotted for the search. A large set of possible coverage patterns is then simulated, *a priori*, and evaluations of each pattern are calculated for the energy consumption, area covered, overlap of covered area, wind energy generation, and solar energy generation. The set of patterns is given to the Mission Planner ahead of time by the user, and in simulations the number of patterns has ranged from 23 to 86 different varieties. The Mission Planner picks the pattern which best meets the robot's requirements, based on one of the task selection methods described in the previous section E.

In steps 2C and 3B, all target and sensor pairs are evaluated, and the best pair is selected. The evaluations performed here are the information gain of using the sensor on the target, the distance of the target from the robot, the time cost to perform the sensor measurement, and whether or not the target is already within the sensor's workspace. The information gain value is used as an absolute cutoff: if the information gain is less than 0.05, the target/sensor pair is not considered

any further. This value was chosen based on the expected information gain outputs of the Classifier module. Otherwise, two types of target distance are considered in the next stages of the evaluation. First, the distance of the target to the side of the robot, in the local x direction, is calculated. If the target is further than half a row width away, in the direction the coverage pattern is progressing, the target should instead be examined on the next row, and not at the current time. These first two evaluation cutoffs are the only evaluations considered in step 2C, when the robot is simultaneously following the coverage pattern. If all the target/sensor pairs get thrown out of consideration by either the information gain cutoff or the local x distance cutoff, then the Mission Planner will just continue the search pattern until a target/sensor pair can pass those two evaluation tests. If no target/sensor pairs pass these two evaluations during step 3B, which should always occur after the robot has examined all targets in the immediate vicinity, then the Mission Planner will request the resumption of the pattern and restart Target Acquisition.

If a target/sensor pair does pass the two cutoff evaluations, then further evaluations are performed. The linear distance of the target away from the robot is considered, along with the time cost and sensor workspace. A standard time cost for using each sensor is obtained from the Sensor Manager. Then, the workspace of each sensor is checked, to see if a maneuver will have to be performed before examining the target, adding to the time cost. The total evaluation cost for the target/sensor pair is calculated as a weighted sum: cost = 1.0 * distance + 0.10 * time cost. These weights for the distance and time cost were chosen such that the distance is the primary discriminator between targets, and the time cost will generally only have an effect on separating different sensors being used on the same target. The standard outcome of this target/sensor pair ranking and selection is that each target will be examined by all sensors eventually, with the cheaper cost sensors being used first for all targets in the immediate area.

NAVIGATIONAL PLANNING

As mentioned earlier, the Navigation Manager utilizes the same planning and simulation library as the Mission Planner. This library is actually comprised of two parts: navigational planning and the solar ephemeris algorithm discussed in section A. The library creates plans such as coverage patterns, waypoint tasks, and maneuvers, based on initial parameters given to it by

whichever module calls it. The functions in the library can initialize the given type of plan, and then put the plan into a general structure which can be used by either a simulator or the actual robot. Once initialized, other functions in the library interpret the plan structure based on knowledge of the real or simulated robot's current location, and produce turning and speed commands designed to be enacted once a second. In one case, the Mission Planner's simulator calls the interpretation functions, takes the plan's commands, and converts them into new "robot" positions, moving through the plan as rapidly as possible. The simulator's internal "clock" is advanced just as rapidly, allowing evaluations of all costs and benefits to be made as the simulation progresses. The simulations can be set to use any starting date and time, and any starting location on Earth or the moon. In the other case, the Navigation Manager calls the interpretation functions every real-time second, passing them information about the robot's current position, and passing the resulting turning commands to Arbiter and then the robot itself.

In addition to the rapid simulation capabilities of the Mission Planner, a separate simulator module can be used in place of the Realtime System and robot. This simulator takes the place of the Arbiter and CTR modules, processing the commands from the Navigation Manager directly with a model of the robot's locomotion, and sending out telemetry messages similar to those of the real robot. The simulator processes the commands at the same rate that the real robot would, and is primarily useful for testing the entire loop of planning, commanding and monitoring.

Coverage pattern plans are generated in terms of path segments, either straight lines, circular arcs, or point turns. All the patterns can be broken down into a combination of these path segments. The circular arcs can be of any curvature and any length, and straight lines are just a special case with an infinite radius of curvature. Whichever type of path segment the robot is following, a single path following function can take that path segment, regardless of the pattern type, and determine from its parameters and the robot's current position where the robot should be driving next. Patterns are simulated or enacted one path segment at a time, with the pattern being built up piece by piece. Patterns can also be interrupted, saved and resumed at any time, by recording information about the current path segment and the pattern parameters as a whole.

When the Navigation Manager actually enacts a pattern, deviations of the robot from the pattern are to be expected for various reasons, such as inertia and mechanical sluggishness of the

robot, irregular terrain, avoidance of obstacles, and pattern interruptions due to target investigations. If the robot is not where it should be, the navigational library functions can still determine where it should be next. The closest path point to the robot is found, and if that closest point is further than a certain distance, the robot is commanded to just drive directly toward that goal point. If the robot is closer to the path or even exactly on the path, then a new goal point is found a lookahead distance away on the path. This goal point is updated every second, pulling the robot along the path. This pure-pursuit method of path following is commonly used [20], but the proper lookahead distances depend upon the responsiveness of the vehicle being used, and the curvature of the path being followed. Proper setting of the lookaheads allow the robot to merge back into the path as quickly as possible, but without oscillations about the path. The lookahead distances used with Nomad were set experimentally based on many field tests. These tests also resulted in several modifications to the pure-pursuit method for situations when the robot is switching between path segments [79]. The primary modification requires the path following algorithm to switch to the lookahead distance of the next path segment before the robot actually reaches the end of the current path segment. The timing of that switch is based on the size of the lookahead distances themselves.

The turning radius with which the robot must drive to reach the desired goal is compared to the set of turning radii allowed by Arbiter, and the closest allowable radius is given a vote of 1.0. Nearby turning radii are given lessor, but non-zero votes, with larger votes being given to the closer radii values. The votes for the entire radii set are sent to the Arbiter module every second. If the Arbiter is told to veto the most desirable radius by the Obstacle Avoidance module, then nearby radii are still given higher priority due to their non-zero votes. This will cause the robot to deviate from the pattern until the obstacle is no longer a factor, but as described above, the robot can quickly recover from such deviations.

Waypoint and maneuver plans are also implemented in the navigational library. For waypoints, the robot's current position is compared to the goal point, and the required turning radius to reach that goal is calculated in much the same way as for path segment goal points. Maneuver plans are more complicated, and are discussed in Appendix A, along with further details about the navigational planning library.

Chapter IV

Experimentation

This chapter describes the performance of field tests and simulations leading to the development and validation of this research. First, the procedures for generating and recording field data are given, along with a description of the robotic platform itself. Second, the types of field work performed are described, and finally, the types of simulations produced are specified.

A. Experimental Procedures

Field tests and simulations both generated data for this thesis. Field tests with the robot Nomad were performed in Pittsburgh at a slag heap, and in three locations in Antarctica: near Patriot Hills in the Chilean sector, at Williams Field near the U.S. McMurdo base, and at a remote site named Elephant Moraine, 160 miles northwest of McMurdo. Various types of data were recorded by the Monitor at these locations, including differential GPS position, compass heading, roll and pitch, motor currents, power supply voltages, wind direction and speed, and timestamps. Once solar panels were attached to the sides of Nomad, current readings from the two panels were recorded. Data from the concurrent simulation of solar power generation was also recorded, along with other evaluation categories such as area covered and area overlap, and post-processed for calibration and analysis. The totals for each category were recorded as well as incremental values each step along the way. Additional reports from the Mission Planner are recorded in separate files, such as the start and stop times of sensor and navigational tasks, along with defining parameters for each task. The distance deviation of the robot from a coverage pattern was also recorded once a second. Additional code was written to process these recorded data

files, producing reports on the amount of time spent in various tasks during a search, statistics on the positional error, number of sensor failures, and other similar reports. An overview of these report results is given in Appendix C.

Nomad is a four-wheeled, 1600 pound robot designed by researchers at Carnegie Mellon University as a prototype planetary explorer. This gasoline-powered vehicle has a footprint of about 1.5 x 1.5 meters when stowed for transport. The wheels can be deployed in the field, for a larger, sturdier configuration and more stable locomotion (see Figure 13). The robot's fuel tank can last over 12 hours when idle, but must be refilled more frequently during active testing.

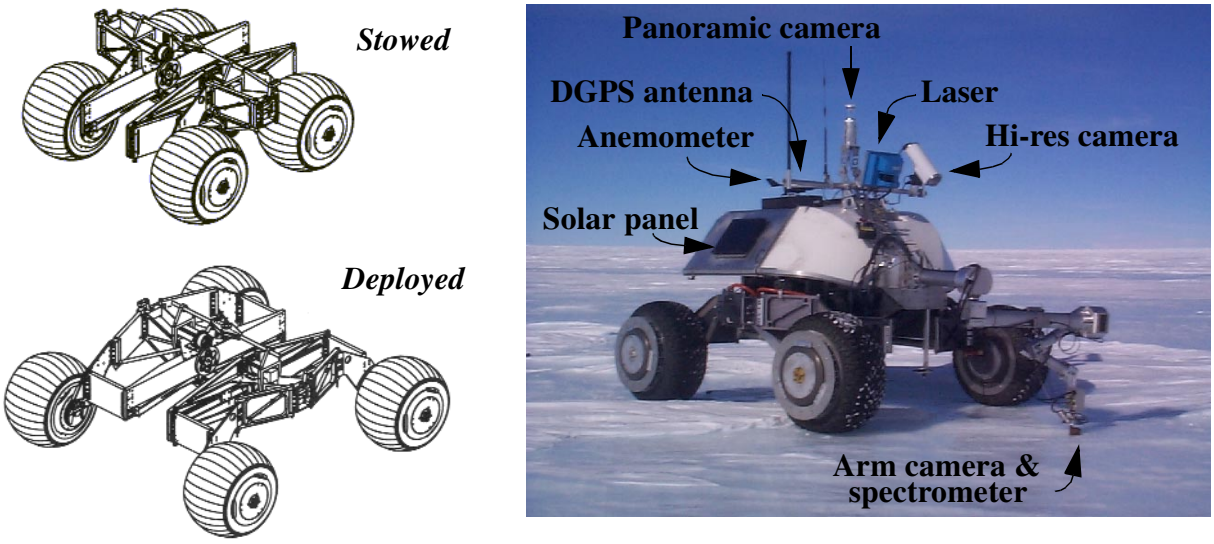


Figure 13: Nomad in Antarctic configuration

The sensors on board Nomad have changed over time as the robot's mission has changed. In the summer of 1997, Nomad and its autonomous navigation system was tested extensively in Chile's Atacama Desert. At this site, Nomad primarily used differential GPS, stereo cameras, a panoramic camera, and various communications equipment such as wireless ethernet. Nomad was then retrofitted for cold weather conditions to prepare for Antarctic expeditions in 1998 - 2000. The mission here was searching for meteorites on ice fields. In the first years, various sensors were tested for their utility in this search. Stereo cameras were found to not work well in the featureless expanses of blue ice fields. A laser was added for obstacle avoidance, and a high resolution camera was added for detecting target rocks and taking close-up images. An arm was added in the summer of 1999, with an attached camera and spectrometer for observing rocks.

Two small solar panels were attached to the sides of Nomad, to gather data for this thesis and to evaluate the potential for a future solar-powered robot, though they do not power any systems. The panels are rated for 10 Watts, and are approximately 0.11 square meters each. The solar panel model used in simulations is based on the configuration of these physical panels, including both their power output and their orientation relative to the robot. Additional configurations are used in some simulations, to vary the potential amount of power generation. An anemometer also records wind speed and direction, along with other atmospheric data gathered by the multi-purpose weather station, but no wind turbine is in place to actually harness wind power. For most simulations, a turbine with one meter radius blades was assumed, to match the size of Nomad.

B. Field Work

Preliminary navigational testing was performed in Pittsburgh in the summer of 1998. Two types of coverage patterns as well as path following algorithms were initially developed at this time, and integrated into the existing navigational software. These coverage patterns were tested in November 1998, in the Patriot Hills region of Antarctica. Seven separate patterns were performed at this time, primarily to test the path following accuracy of Nomad in its winterized condition, and on the snow and ice terrain of Antarctica. In most cases, the patterns were followed accurately, with little difference from Pittsburgh tests. One test evaluated the ability of the robot to purposefully deviate from a pattern, such as to investigate a target, and then resume the pattern. Images of the robot's groundtrack during three of the tests are given in Figure 14, and further images are shown in [79]. The deviation shown in the leftmost image below is much greater than any deviations which occurred in actual meteorite searches the following season, since the row width is usually set such that any targets a sensor detects will be no more than half a row width away from the current row. These preliminary experiments did lead to improvements in the coverage pattern generation software, such as the lookahead distance modifications described in the previous chapter. Other improvements included setting appropriate experimentally-based lookahead distances for particular path segment curvatures, and developing the bulging row end turn for narrow row width patterns, as shown in Figure 8 of chapter 3. Sun and terrain information was not utilized at this time.

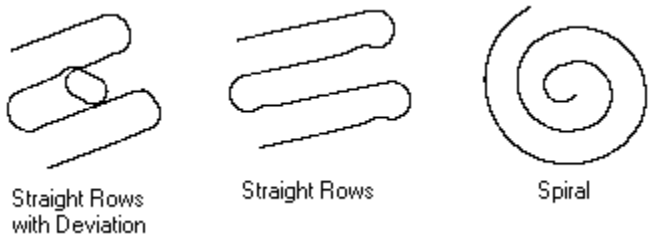


Figure 14: Coverage pattern groundtracks, Patriot Hills

Further Pittsburgh testing was performed for 32 days during the months from August to November of 1999. Much of this testing involved verifying the interactions of the planning software with science sensors and navigational software. The Mission Planner's ability to guide a meteorite search was tested, and potential sensor errors to which the planner would need to respond were identified. Some sensor status reports indicated that redoing the command might produce different, better results, such as when temporary mechanical glitches occurred which were unlikely to be repeated. Other sensor status reports indicated that the target object could not be found, either due to lighting conditions or the target being a false positive. Distinguishing between status reports allowed the Mission Planner to either redo a command or continue with the next step. Modeling of the robot's response to steering commands allowed development and testing of the maneuvering capability. Initial testing of Nomad's solar panels also occurred, with data gathered on the performance of the panels in Pittsburgh environmental conditions.

Nomad went to Antarctica again in the 1999-2000 season, this time to the region near McMurdo, the main U.S. base. Eight days of test data were taken at Williams Field, in December and early January. Several coverage patterns were performed while gathering solar power generation and wind speed and direction data, along with basic pose information. This data was taken to validate the solar simulation, and to obtain information on the effect of pose uncertainty due to terrain. This pose uncertainty can result in unexpected variations in the amount of solar power generation, as compared to a simulation of the same task performed ahead of time. Similar pose data is available from Pittsburgh and Elephant Moraine locations, though in all cases, the terrain is fairly smooth, with only minor variations in pitch and roll.

Seventeen days of test data were taken at Elephant Moraine in January, including ten meteorite search demonstrations. Some of the non-search tests included coverage patterns performed while collecting solar and wind data. A straight rows pattern and a spiral pattern were both performed at the same location, starting at approximately the same time on two different days, to evaluate the differences between the patterns with basically the same solar parameters. Sun-following and curved patterns were never demonstrated in the field, as these pattern types were not implemented until after Nomad returned from the 2000 Antarctic season.

The meteorite search demonstrations were all performed using the straight rows pattern. Multiple stops were made during the patterns to evaluate targets, lasting 4 minutes and 11 seconds on average. Statistics on each search are given in Appendix C. These searches were performed with minimal intervention from human team members. Allowed intervention included only monitoring the placement of the spectrometer due to a frozen contact sensor, and recovering from sensor errors or timeouts which triggered user teleoperation modes in the Mission Planner. In most cases, the searches were performed seamlessly, with the Mission Planner overseeing the entire sequence of events. The three meteorites which were encountered during these searches were all given higher probabilities of being a meteorite by the Classifier than any of the terrestrial rocks. Several other meteorites were also identified during non-search tests.

C. Simulations

Field tests were not possible for evaluating all aspects of this thesis, so simulations were performed to gather more data. In addition to guiding searches, the Mission Planner has the ability to either perform or simulate individual navigational tasks. These simulations provide final evaluations for each category (solar power generation, power consumption, area covered, etc.), as well as record incremental evaluations along with other telemetry. A standalone version of this simulation capability was developed, such that no inter-module communication is needed.

Simulations of previously enacted field tests, using just the initial pattern parameters, are used in offline performance critiques, and new simulations have been studied individually and in composite missions, showing the increase in robotic ability that occurs when solar conditions are

included in planning decisions. The simulations are based on models of Nomad and its solar panel configuration in most cases, with a different robot configuration, including solar panel size and power consumption values, modeled in other cases. Six types of simulations were performed to evaluate aspects of onboard usage of sun and terrain knowledge during coverage patterns.

1. The amount of variation between pattern types was examined by evaluating multiple variations of all four types of patterns for a range of latitudes from the pole to the equator, during summer and wintertime, and for the Earth and moon. Evaluation categories of solar power generation, power consumption, area covered, and area overlap were considered, for patterns lasting a designated amount of time. Evaluations of this large set of pattern types was also used to determine some heuristics for selecting a smaller, but still representative set of patterns to evaluate for a given set of conditions. Considering every conceivable variation of all pattern types is not feasible for onboard planning. Given the goals of a mission, however, a smaller set should be identifiable which will still contain the more preferred patterns.

2. Curved patterns were examined in greater detail, by varying the curvature and direction of travel for a smaller set of locations. The resulting evaluations were examined to determine any clear relationships between the evaluation results and the pattern curvature and location. Such relationships are exploited as heuristics for shortening the amount of time needed to evaluate such patterns. Finding analytical descriptions of evaluation categories as a function of curvature allows evaluation without complete simulation.

3. A set of patterns covering designated areas were simulated, with the addition of a terrain map causing shadows throughout the region. Various starting points, headings and times were allowed. The amount of solar power generation was evaluated and compared to the amount of power which would have been generated without the terrain features. Analysis of the results from different starting conditions provides assistance in automating starting point selection.

   Determining appropriate starting locations and times is a complex task, considering all the potential choices. Utilizing previously generated shadow map animations is one method for selecting promising starting points. Humans can identify such locations without too much difficulty, but autonomous selection is harder due to the third temporal dimension. This set of

simulations provides a heuristic for reducing the number of potential starting points to consider. Other potential methods for autonomous selection are discussed in Chapter 6.

4. A more complicated mission scenario involved a robot which derives power from solar panels and a battery, and requires recharging stops whenever its battery reach a critical level. Previous simulations assumed an unlimited power supply. The power requirements of the robot were scaled appropriately, and a model of two 1.25 square meter solar panels replaced the model of the smaller panels installed on Nomad. Various sizes of batteries were assumed, ranging from 1.0 to 2.5 amp-hours. A simplistic model of the power system was assumed, where the robot was either generating more power with the solar panels than it was consuming, or the robot required more power from the battery. The amount of power consumption was based on the simulation of robot pose and turning activity. A baseline locomotion and steering power of 150 W was assumed, with additional power required for driving uphill or for performing point turns, for example. If an excess amount of power was being generated, then the battery was charged with a rate of 0.5 times the capacity. If excess power was needed, the battery was discharged with a rate of 1.0 times the capacity.

   Three missions scenarios were simulated. In one, the robot drove for a certain total distance. In the other two, the robot drove that total distance, but random targets were included once every 100 meters or so for which the robot had to stop and "investigate" for about 5 minutes. One version of this required the robot to just stop in its current orientation. The other required the robot to point turn to the heading which enabled it to generate the most power for that 5 minutes. All four types of patterns were tried, for a range of latitudes and locations. The state of the battery was monitored, and when it dropped below 20% capacity, the robot stopped, point turned to the optimal location, and recharged its battery to at least 99%. Terrain shadowing was not present in these simulations, as a flat, benign surface was assumed. After recharging, the pattern was resumed. The total mission times required for each pattern type, the amount of time to reach the first recharging period, the time spent during each recharging period, and the number of recharging periods required were recorded.

5. The effect of pose uncertainty on power generation and consumption was studied by evaluating a set of straight rows patterns. A randomized Gaussian distribution of roll and

pitch values was incorporated into the simulation, with several different magnitudes of allowable pose variation. This set of simulations addresses the short-term pose variations due to driving over small rocks, dips or varying slopes, and considers whether such unpredictable variations create a measurable difference between a priori simulations and actual patterns. Yaw uncertainty was not considered for two reasons. One, the magnitude of its effect would be similar to the magnitude of roll and pitch uncertainty effects. Two, the absence of yaw uncertainty allowed more control of the shape and size of the pattern simulations.

6. Trade-offs between solar and wind power generation were studied by simulating straight rows patterns with varying relationships between the starting sun azimuth, wind direction, and row direction. Several different wind speeds were considered, as well as different latitudes. The simulation enables selecting the best row direction in order to generate the most total power, whether that direction favors solar power or wind power, or a combination of the two.

Off-board usage of sun, terrain and temporal information was also developed, with the creation of shadow maps and path search algorithms. Two detailed terrain maps were used in the creation of animated shadow maps: a digital elevation map of the south pole region of the moon [59], and a digital elevation map of Haughton Crater in the Canadian Arctic. Shadows covering the lunar map were simulated for a series of 30 days both in the summer, when the sun is at its highest elevation at the south pole, and in the winter. The lunar “day” lasts about 30 Earth days, and so a “snapshot” of the shadows was taken once each Earth day, for 30 days. The resulting series of images have been put into animations showing the progression of shadows across the surface. These animations have been used to determine the feasibility of sun-synchronous robot missions in the south polar region, and to identify prime exploration regions for a potential mission. Similarly, shadows in the Haughton Crater map were simulated for a series of 24 hours, with one snapshot for each hour.

Single snapshots are shown in Figures 15 and 16, to demonstrate the type of images generated. The blue pixels indicate completely shadowed terrain, and the grey pixels are sunlit terrain. Black indicates unknown terrain for the moon, and creeks or dry creekbeds for Haughton Crater. The approximate position of the sun relative to the map is also shown. The discrete terrain elevations in the Haughton Crater map do create shadowing artifacts at the edges of

elevation differences. Higher resolution terrain maps will reduce the number of such artifacts, but the map used in Figure 16 already has a relatively high resolution, with elevation steps of only 10 meters. False shadows could be removed by considering a smoothed range of pixel values instead of individual pixels, assuming that the slopes will always be gradual. However, this method will not be able to distinguish between actually smooth hillsides and abrupt dropoffs, based on the limited map information itself. Considering the safety of the robot, leaving in the artifacts may be preferred, since avoiding false shadows is better than ignoring real ones.

As discussed in the section on solar navigation in Chapter 3, these shadow maps can be combined and used for a variety of purposes: human path or location selection, autonomous sun-synchronous and sun-seeking path planning, and identification of promising lunar base sites. Autonomous path searches using both a pre-generated map database and the solar ephemeris algorithm directly are described in the next chapter.

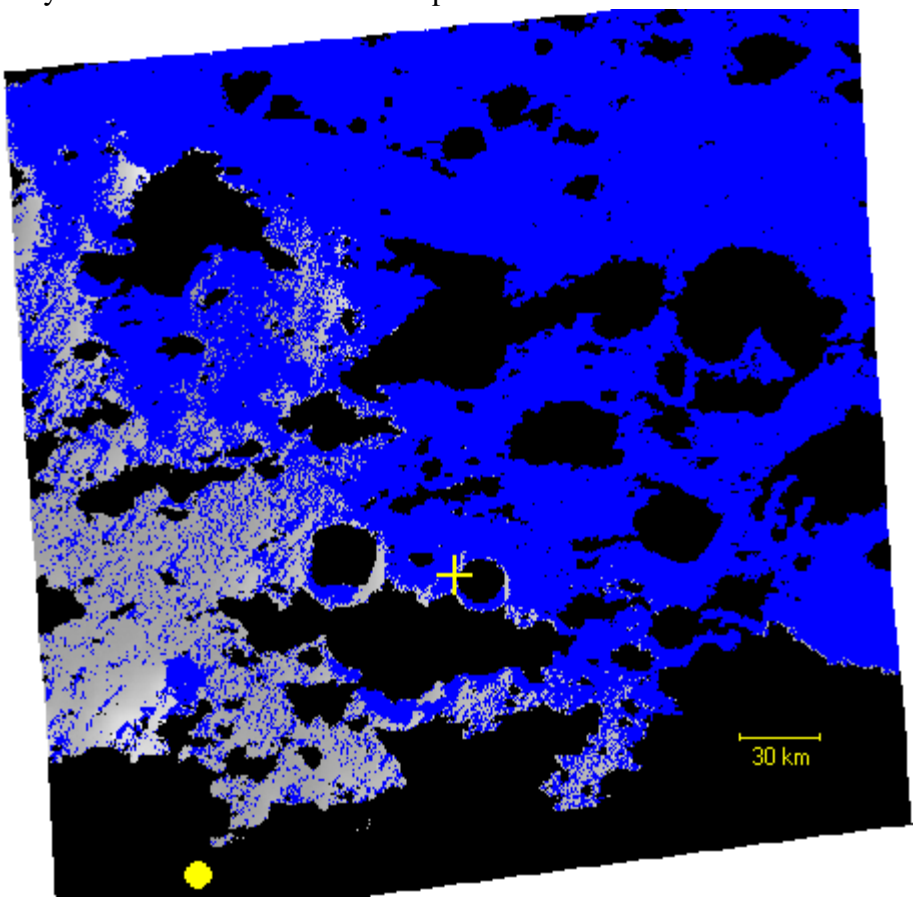


Figure 15: Shadows at lunar south pole, 0UT, April 1, 2000

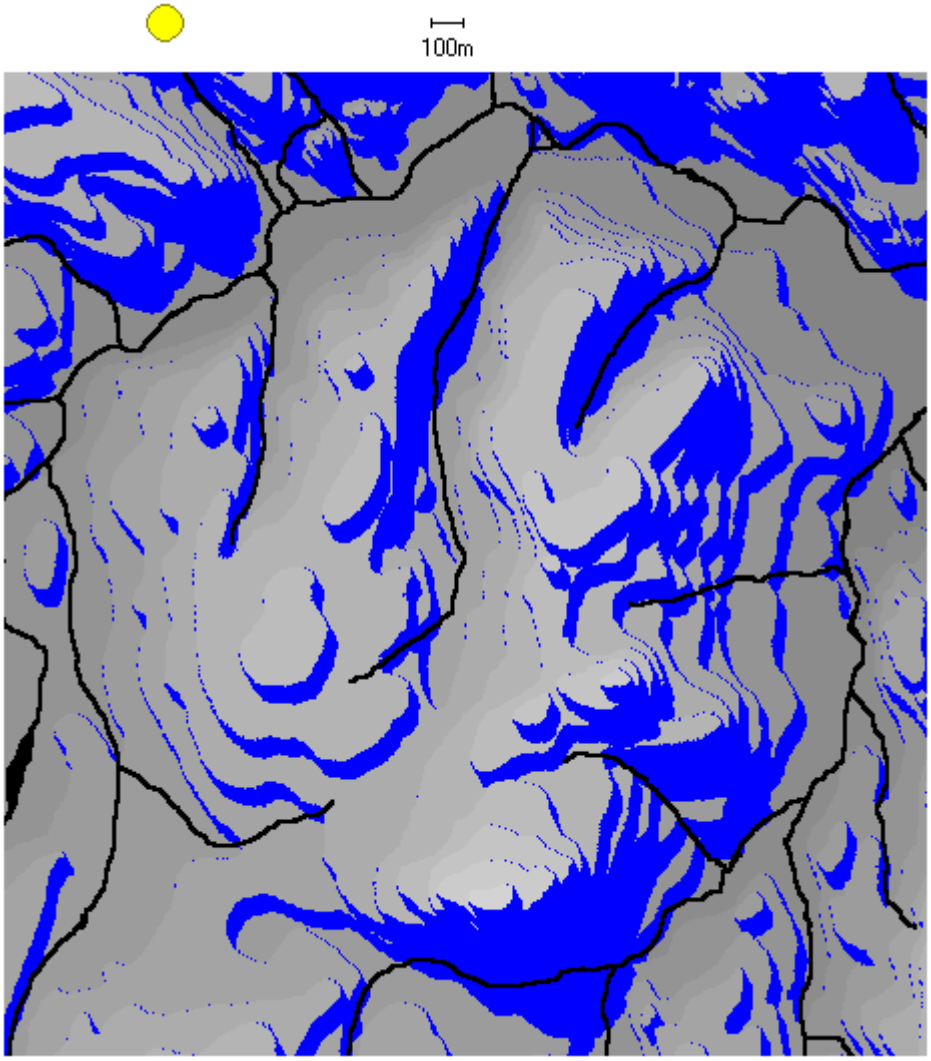


Figure 16: Shadows at Haughton Crater, 500UT, July 15, 2000

Chapter V

Results

This chapter presents the results of field tests and simulations performed for this thesis. The fieldwork demonstrates the robot's path and pattern following accuracy and the comparison between actual and simulated solar power generation. The simulations demonstrate pattern comparisons and selection, analytical heuristics for shortening simulation time, the effects of terrain features on pattern selection, quantitative improvements in mission productivity and efficiency, the effect of uncertainty on predictions, and a method for pattern selection based on multiple variables. Autonomous solar navigation results are also shown.

A. Fieldwork

A set of one straight rows pattern and one spiral pattern were performed in each of three locations: Pittsburgh, Williams Field, and Elephant Moraine. These patterns demonstrate the path following ability of the Nomad robot. The patterns in Elephant Moraine were more carefully designed to take place in exactly the same location, and at approximately the same time of day, so the solar aspects are nearly identical for comparison of solar power generation. The actual groundtracks of these patterns are shown overlapping in Figure 17.

SOLAR POWER ACCURACY

The solar based selection of plans depends heavily on the accuracy of solar power generation predictions. This section shows that the predictions are accurate, once the solar panel models are

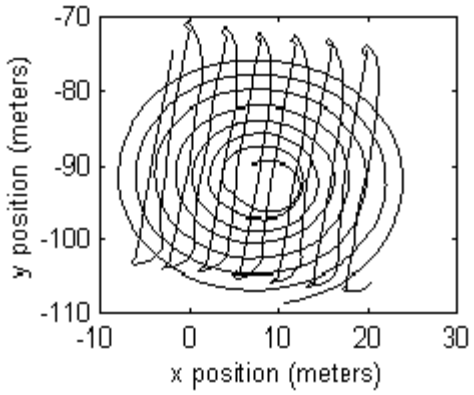


Figure 17: Groundtracks of straight and spiral patterns, Elephant Moraine

properly calibrated. There are two types of simulations, as discussed earlier: concurrent and a priori simulations. The concurrent simulations aid in the calibration of the models, while the a priori simulations are used in the selection of the best patterns.

The solar power generation model used in the simulations does not take into account all the details involved. The overall magnitude of the actual power differs due to inaccurate modeling of the panel configuration, including the wiring connection to the digital-analog converter, inaccurate modeling of the sunlight intensity and panel response, and lack of modeling of reflected or diffuse light. The sunlight intensity in the wavelengths to which the panels are responsive is apparently stronger in Antarctica, for example, but this was not modeled. However, even the uncalibrated model used for this research has proved to be sufficient in most cases, particularly for differentiating between different coverage pattern tasks, which is a primary goal of this work. This simpler model is more computationally efficient, reducing the overall planning time, while calibration improves the accuracy considerably without much additional computation.

Figure 18 shows the actual power being generated by each of the robot's solar panels, as well as the concurrent simulation of power generation, for the straight rows pattern at Elephant Moraine. A calibrated version of the concurrent simulation is also shown, having been scaled and shifted to match the actual power. This calibration is performed by using only two small sets of points -- one during the panel's peak power generation and one during the panel's minimum power generation. These sets can be obtained quickly and autonomously at the start of a day by

turning the robot such that one panel faces the sun, and then the other panel faces the sun. The point sets are averaged and used to scale the simulation, producing calibration values which can then be used for accurate a priori simulation evaluations.

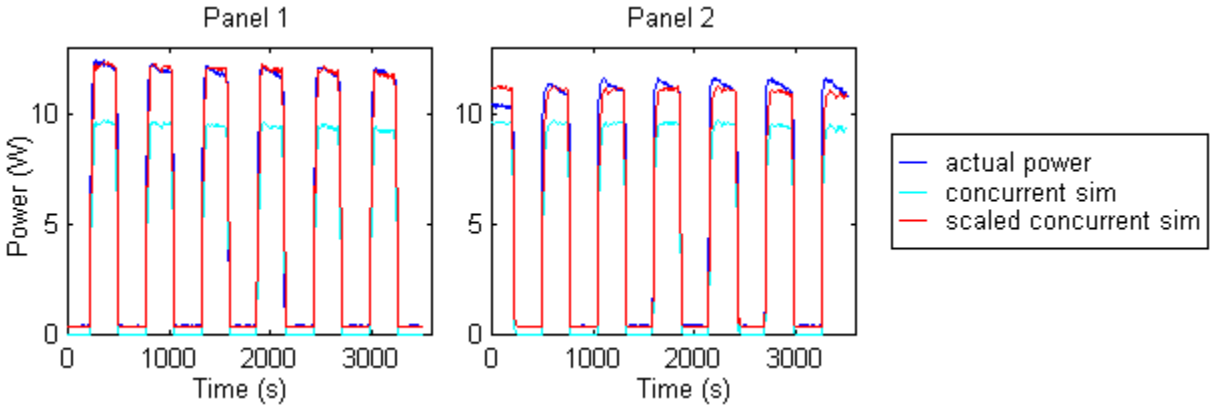


Figure 18: Power generation per panel, Elephant Moraine straight rows pattern

As long as environmental conditions such as cloud cover, and terrain conditions such as reflectivity, persist for the entire day, or at least the duration of desired patterns, then calibration at the start of the planning process will be sufficient. The moon, therefore, is an excellent location for this calibration method. Additional calibrations will only be needed as the robot travels to regions with different reflectivity values. More detailed modeling can prevent the need for as much calibration, but at higher computational costs. Actual data was only obtained for short time periods, and in single terrain types, as the typical coverage patterns did not last longer than an hour or two. Environmental conditions during this time length did remain relatively constant in most cases. As a note, the sun moved about 14.5° across the sky in the hour the Elephant Moraine straight rows pattern took to complete. The pattern followed the heading which was optimal for the starting time, but by the time the pattern finished, the maximum power generation was still 96.8% of optimal, due to the dependence of power generation on the cosine of the incident angle. Thus, the maximum power values did not noticeably drop off during the pattern.

Figure 19 shows the same panel values for a straight rows pattern at Williams Field, from a cloudy day. The peak power generation in this case is greatly reduced, while the minimum power is only slightly less. This is due to reflection of the light on the white snow and ice, as well as

diffusion of the light by the clouds. The same calibration procedure can be carried out, however, producing accurate scaled simulations.

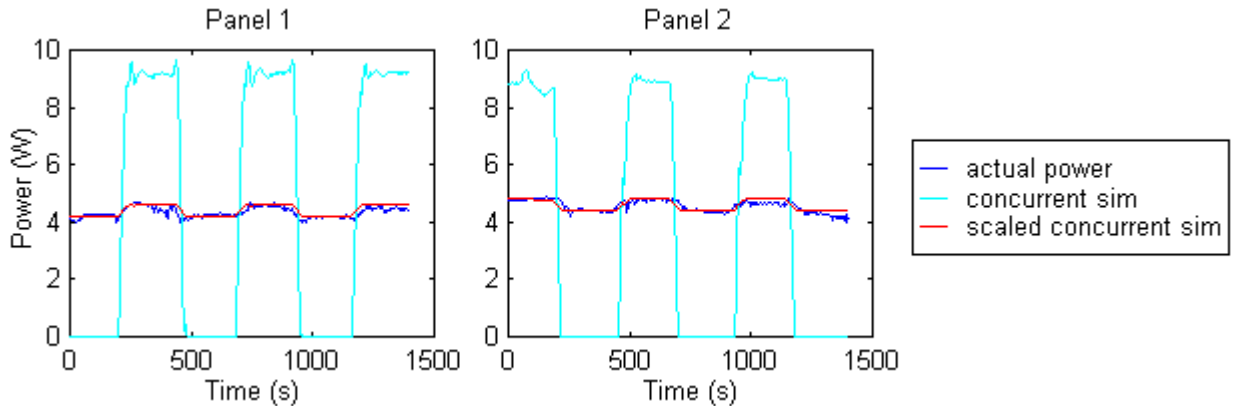


Figure 19: Power generation per panel, Williams Field straight rows pattern

Figure 20 shows the calibration for a spiral pattern, with similarly promising results. Notice that for straight rows patterns, the square wave effect of the power generation is due to the robot, and its panels, switching sides for each row. For the spiral pattern, the change in power generation is more gradual for each panel, following a sinusoidal curve with peak values occurring when the panels face the sun directly. The minimum values are flattened since the power drops off to near-zero when the panel is more than 90 degrees offset from the sun. The only power generation occurring at these times is due to surface reflection of the light, since the sun itself is blocked by the body of the robot. For spiral patterns, this orientation occurs for approximately half of the spiral loop.

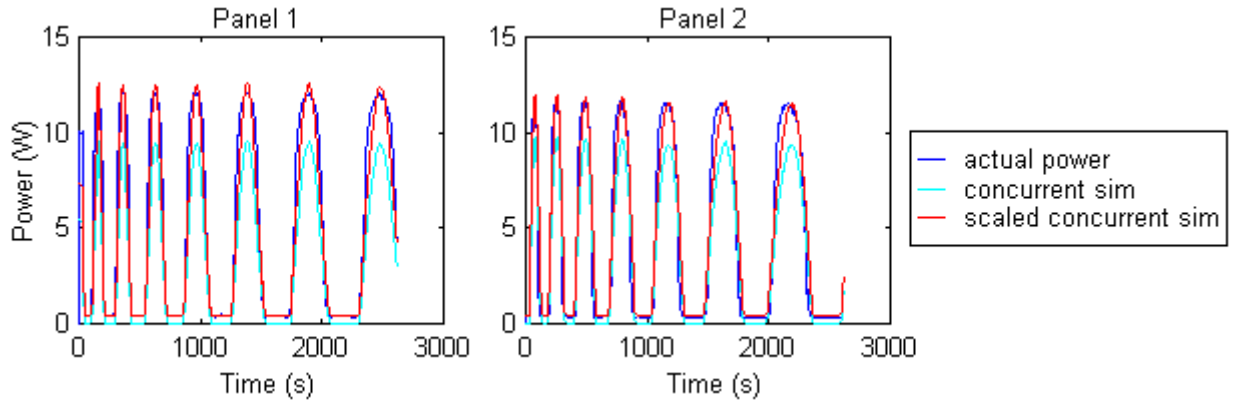


Figure 20: Power generation per panel, Elephant Moraine spiral pattern

The results of applying the calibration values to a priori simulations are shown in Figure 21. These plots show the cumulative solar energy being generated from both panels during the patterns. The total times for the simulations differ from the actual patterns, due to inconsistencies in the robot's speed, particularly during turns. The actual values for the patterns in Antarctica (blue) match the calibrated a priori simulations (red) quite well. Note that the results of the different coverage pattern types in Elephant Moraine are clearly distinguishable, allowing the pattern generating the most energy to be accurately identified from the a priori simulations. Numerical statistics on the accuracy of the calibrated predictions are given in Table 1.

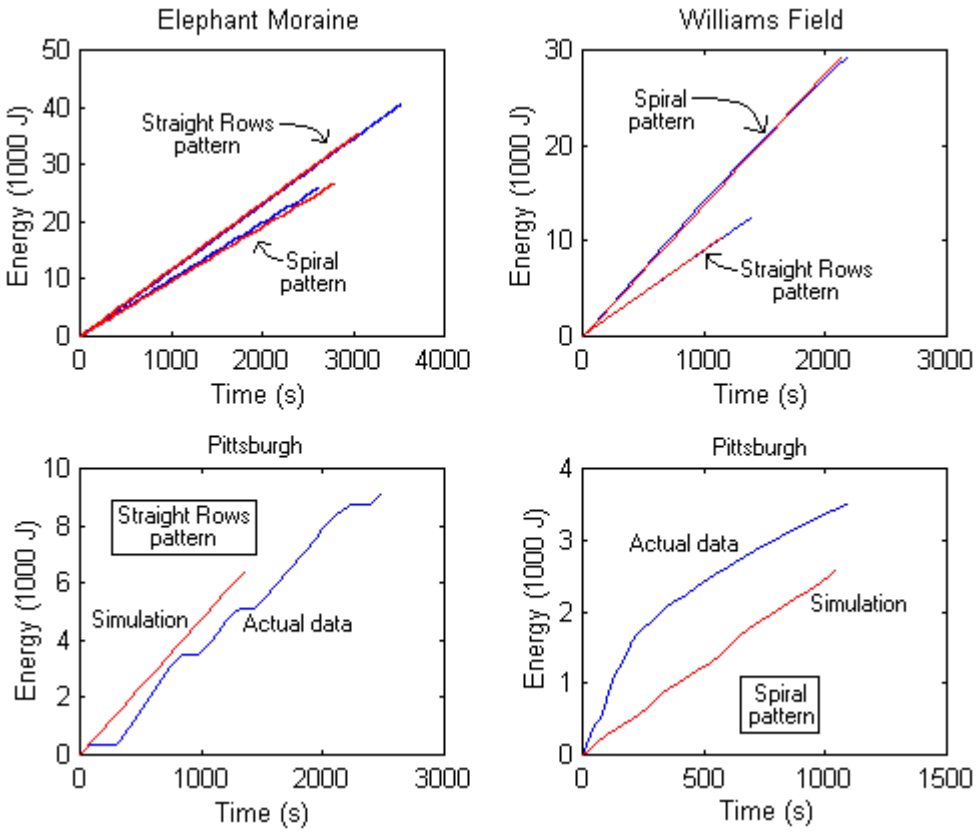


Figure 21: Actual vs. calibrated a priori solar power generation simulations

The straight rows pattern in Elephant Moraine generates more power overall than the spiral pattern, while at Williams Field, the opposite is true. This is due to the fact that the day the Williams Field straight rows pattern was performed, the sky was completely overcast, reducing the power generation capabilities considerably. For the spiral pattern at Williams Field, and both

patterns at Elephant Moraine, the sky was clear. Thus for comparison between pattern types, the Elephant Moraine trials are the best ones to consider.

Location	Straight Rows		Spiral	
	mean error (J)	mean error (% of final value)	mean error (J)	mean error (% of final value)
Pittsburgh	687.3631	17.96%	95.4069	2.78%
Williams Field	19.6269	0.19%	262.7916	1.62%
Elephant Moraine	194.7242	0.65%	323.1652	1.25%

Table 1: Solar power generation prediction accuracy

The Pittsburgh pattern results are not quite as good as the Antarctic patterns. The problem with the spiral pattern is that clouds moved in right after the first spiral loop was completed, greatly diminishing the solar power generation capabilities. The calibration values were taken from the cloudy portion, as can be seen from the matching slope in the latter portion. In the initial portion, for about the first 250 seconds, a considerably greater amount of power was being generated. For the straight rows pattern, clouds were moving in and out of the area, causing several periods of diminished power generation capability. The calibration attempts to match the average slope, but with limited success. This confirms that this modeling technique and solar power generation prediction is better suited to locations with more constant atmospheric conditions. Locations with more atmospheric and reflectivity variations would benefit from higher fidelity modeling of solar power generation.

PATH ACCURACY

Studies of how accurately the robot follows coverage patterns and paths in Antarctica help in understanding the uncertainties involved with planning. These uncertainties might arise from terrain effects such as rocks and slopes, obstacles, or just inaccuracies in following the path. The power generation capability of a robot depends on the pose of the robot, if solar panels or wind

turbines are used. Therefore, the uncertainties in predicting the robot's position over a given path will determine the uncertainties in predicting the power gain from solar or wind power.

The mean and standard deviation of the pitch and roll values of the robot throughout the duration of the patterns was calculated to give an idea of how much uncertainty might be due to the terrain. These numbers are given in Table 2. The terrain in each of the three locations was fairly mild, with the standard deviation less than two degrees in all cases. The mean values are generally less than one degree as well, indicating that the slopes are mild. The effect of pose variations on prediction of solar power generation is studied in one of the simulation categories described later, showing that such mild variations cause little deviation between actual and predicted solar power.

Location	Straight Rows		Spiral	
	mean pitch (deg)	mean roll (deg)	mean pitch (deg)	mean roll (deg)
Pittsburgh	– 2.26 ± 1.54	– 0.08 ± 0.93	– 0.95 ± 0.99	– 0.21 ± 1.30
Williams Field	–0.70±1.68	– 0.57 ± 1.15	– 0.88 ± 1.70	0.51 ± 1.80
Elephant Moraine	0.03 ± 0.80	0.22 ± 1.33	0.04 ± 0.97	0.26 ± 1.34

Table 2: Mean and standard deviation of robot pose

The variation in yaw is more difficult to evaluate. For a given moment in time, the actual yaw could be compared with the predicted yaw from an a priori simulation, but the timing of the actual and simulated patterns differs enough, due to robot speed modeling inaccuracies, to make such instantaneous comparisons meaningless. If the robot is not at the expected point in the pattern for a given time, then its yaw should not be expected to match the predicted yaw for that given time. This was not an issue for roll and pitch calculations, since the simulation assumes those values are always zero. To demonstrate the timing differences, the straight rows pattern at Elephant Moraine was simulated with a robot speed of 0.15 m/s, which is the commanded speed during the pattern. However, the actual speed of the robot varied from about 0.12 to 0.18 m/s. The average speed was close to 0.15 m/s, but the point turns at each row end were also performed slower than modeled speeds, causing an increasing difference between the actual and simulated patterns.

The other option is to compare the actual yaw with what the yaw should be at a given time during the pattern enactment, but this value is not recorded moment by moment, and generally differs from a priori predictions anyway since it is based on instantaneous actual robot positions and incorporates path deviations. The desired yaw is defined based on the position of the robot with respect to the desired path, and is such that the robot will either continue driving along the path, or drive back toward the desired path when off course. For the spiral pattern, the yaw error is especially hard to measure, since the robot is continually changing its yaw as it spirals around. In this case, the yaw variation may not be as relevant, as long as it is not too large, remaining close to the preceding or upcoming heading values. For the straight rows pattern, the yaw during the straight part of the row can be compared to the desired straight row heading. The difference between the mean actual heading and mean simulated heading of the robot is shown in Table 3 for the three straight rows patterns. The values are divided into the two directions of travel: the initial direction and the return row direction. The simulated heading is basically constant, with slight variations due to the discretization of the grid-based map used. The standard deviation of the actual yaw from the simulated heading is also given, for the course of the pattern.

Location	Mean Actual Yaw - Mean Simulated Yaw ± Standard Deviation (deg)	
	initial row direction	return row direction
Pittsburgh	– 1.33 ± 6.99	– 7.50 ± 3.74
Williams Field	– 0.66 ± 2.52	– 7.51 ± 2.66
Elephant Moraine	1.32 ± 1.19	– 5.34 ± 1.69

Table 3: Mean heading differences, straight rows patterns

The variation in yaw is generally low, and has little effect on the total solar power generation. A deviation of 8 degrees from an optimal heading only causes a power dropoff of less than 1%, due to the dependence of power on the cosine of the incident sunlight angle. The standard deviation of the heading error is primarily a result of path deviations. A few targets were stopped for in the Pittsburgh pattern, causing path deviations during the maneuvering. The fact that the

mean errors are non-zero is caused in part by direction-dependent inaccuracies in the robot's compass, as will be described more below. This inaccuracy causes a bias in the heading error. The row direction was generally chosen to diminish this instrumental error for the initial row heading, which explains why the return row direction has higher mean errors.

The combination of yaw, pitch and roll errors, along with timing and solar panel modeling errors, can also be quantified by looking at the cumulative solar energy generation, and how the actual energy varies from predicted values. That type of error was shown in the previous section. An alternate measure of the quantity of yaw deviation is the distance error between the actual and desired robot positions, since those distance deviations are what cause the heading to deviate from expected values. Statistics on that error are given and described below.

Location	Straight Rows	Spiral
Pittsburgh	0.4573 m	0.3711 m
Williams Field	0.4024 m	0.4053 m
Elephant Moraine	0.2891 m	0.6355 m

Table 4: Mean pattern error

The mean deviation of the robot from the desired path is shown in Table 4 for the three sets of patterns. The errors do not include the portions of straight rows patterns where the robot has just point turned and is merging back into the pattern, as that maneuver is designed to be performed outside the designated coverage area. The errors are mostly consistent, and not due to oscillations about the path. The robot tends to stay a relatively constant distance away from the path on one side of the pattern or row, as shown in Figure 22. The light grey indicates the desired path, black indicates the actual path, and medium gray shows the error between the two. The consistency of the path errors is due to the fact that they are primarily caused by compass inaccuracies, where the direction sensed by the robot is offset from the actual direction. This inaccurate, but constant compass offset is what causes the constant path offset to one side. Given a lookahead distance of 5 meters, and an angular deviation of 4.6 degrees, the resultant path error will be 40cm, for example. Smaller lookaheads will result in smaller path errors, for a given angular deviation.

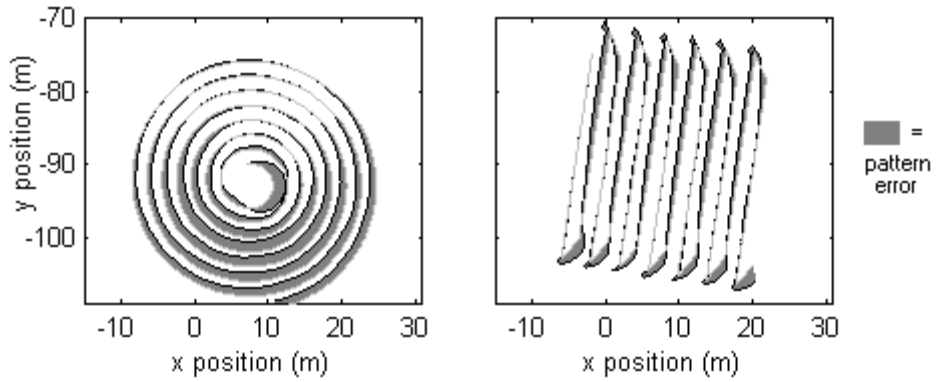


Figure 22: Actual vs. desired pattern groundtracks, Elephant Moraine

The compass inaccuracy also depends on the absolute heading of the robot, causing additional systematic effects. For example, the compass can be calibrated for accurate readings when the robot travels in one direction, but then when the robot starts heading in the opposite direction, the compass errors are larger. In other words, the required compass offset to account for magnetic variations is not constant, but varies with heading, and the compass calibration procedure did not sufficiently compensate for that fact. For the Elephant Moraine location, the spiral pattern shows that the compass error was greatest when the robot was traveling in the west-northwest direction. The straight rows pattern was performed with no travel in that direction, so the errors were less. This heading bias also explains why the error shown in Figure 23 has a systematic effect, with the peak error occurring once during a spiral loop.

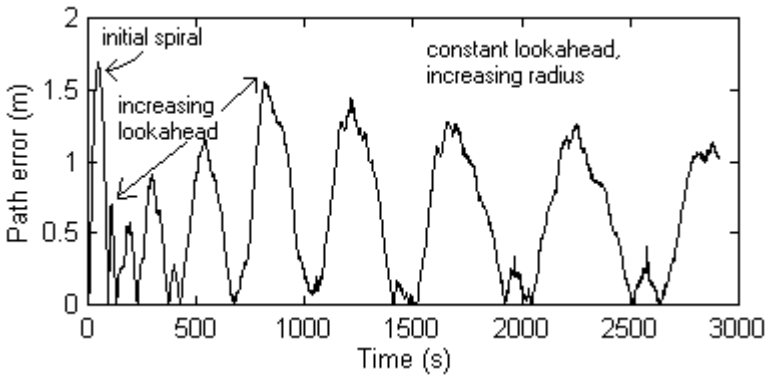


Figure 23: Pattern error for spiral pattern, Elephant Moraine

As mentioned earlier, smaller lookahead distances will result in smaller path errors, which is one reason why the peak error increases for the first loops of the spiral pattern. The lookahead distance is gradually increased for each successive spiral, from 1 meter until reaching a maximum value of 5 meters. The following spirals have a decreasing peak error, as the radius of the spiral increases. Larger radii, for a fixed lookahead distance, cause the angular offset to have less of an effect on the distance offset. The initial large deviation is due to the limited minimum turning radius of the robot.

The pattern errors discussed here can be reduced by using a better heading indicator, or by adding more feedback into the control loop. The consistency of path following directly affects the amount of area covered and area overlap, so for accurate predictions, pattern errors should be reduced. The percent error in predicted area covered and area overlap, as compared to the actual area values for the three pattern sets, are given in Table 5. With the current configuration, the width of rows for meteorite search demonstrations was selected such that an overlap of 50cm purposefully occurred, creating more undesirable overlap, but ensuring that more of the area would be seen. The compass offset was also set such that the error was either minimized for one direction, or equalized between opposing rows.

Location	Straight Rows Errors			Spiral Errors		
	actual area covered	area covered	area overlap	actual area covered	area covered	area overlap
Pittsburgh	661 m^2	4.3%	12.0%	461 m^2	5.6%	5.1%
Williams Field	374.25 m^2	1.8%	12.9%	695.5 m^2	3.7%	5.8%
Elephant Moraine	1070 m^2	3.4%	8.0%	880 m^2	5.4%	1.0%

Table 5: Errors in predicted area calculations

The area overlap predictions are not as accurate as the area coverage predictions in general. The compass error tends to cause pattern rows to be pinched together, and this is particularly noticeable for the Williams Field pattern, as shown in Figure 24. Due to the built-in overlapping in the requested row width, the area is still well covered despite the row deviations. Thus the area coverage error compared to the simulated prediction is small. However, more area overlap occurs in the actual pattern than predicted in the simulation, causing that error to be larger.

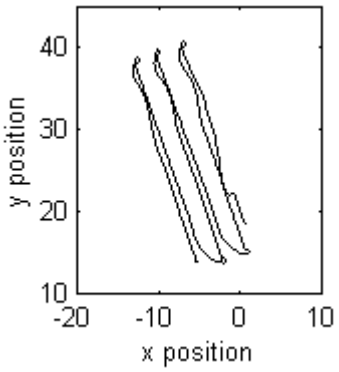


Figure 24: Straight rows pattern at Williams Field

B. Simulations

Field work results were not available for all locations, times, terrain features and coverage patterns of interest. Therefore, a range of simulations was also performed with additional patterns, in different locations, and at different times. General pattern attributes are studied first, leading to heuristics for pattern set reduction, and demonstrating how pattern selection can be done on-board a robot. Analytical evaluation of curved patterns is described next, followed by an examination of terrain shadowing effects. Numerical results of specific mission scenarios are given. The effect of pose uncertainty on power generation predictions is discussed, and finally, pattern selection is shown with the multiple, potentially conflicting evaluations of wind and solar power generation.

GENERAL PATTERN ATTRIBUTES

A set of 86 patterns was selected for simulation to gain an understanding of the differences between pattern types. Sixteen variations of the straight rows pattern were selected, including those with point turns and those with half circle turns at the row ends, with row lengths varying from 30 to 200 meters, and with initial robot yaws pointing toward the sun and pointing 90 degrees away from the sun. Two variations of the spiral pattern were selected: traveling clockwise and traveling counterclockwise. Four variations of the sun-following pattern were selected, with row lengths varying from 30 to 200 meters. 64 variations of the curved pattern were selected, with row lengths from 30 to 200 meters, travel directions to the left or to the right,

and radii of curvature ranging from -200 meters to +200 meters, with negative curvatures indicating a leftward curvature. This set of patterns was evaluated for latitudes from 0 to 90 degrees S on the Earth, during the peak of summer and during the peak of winter. Latitudes from 0 to 90 degrees S on the moon were used during the lunar summer. The small change in sun elevation between lunar summer and winter, about 3 degrees, did not make much difference in the pattern evaluations, so only one set was performed. All the patterns were simulated for 2000 seconds, or about 33 minutes. The sun's position does not change much during this time, which allows the differences between patterns to be seen for more constant sun parameters. Longer patterns will reveal more time-related differences, as are shown in later simulations.

The following simulation results are only a sampling of pattern evaluations. Specific pattern parameters and environmental conditions must still be considered in any given case, which is why the simulation structure and evaluation algorithms have been developed for on-board robotic use. However, these results demonstrate some general characteristics which can be used to understand which pattern types are more beneficial to consider for a given set of conditions, as well as provide insight into the differences between pattern types and the range of possibilities.

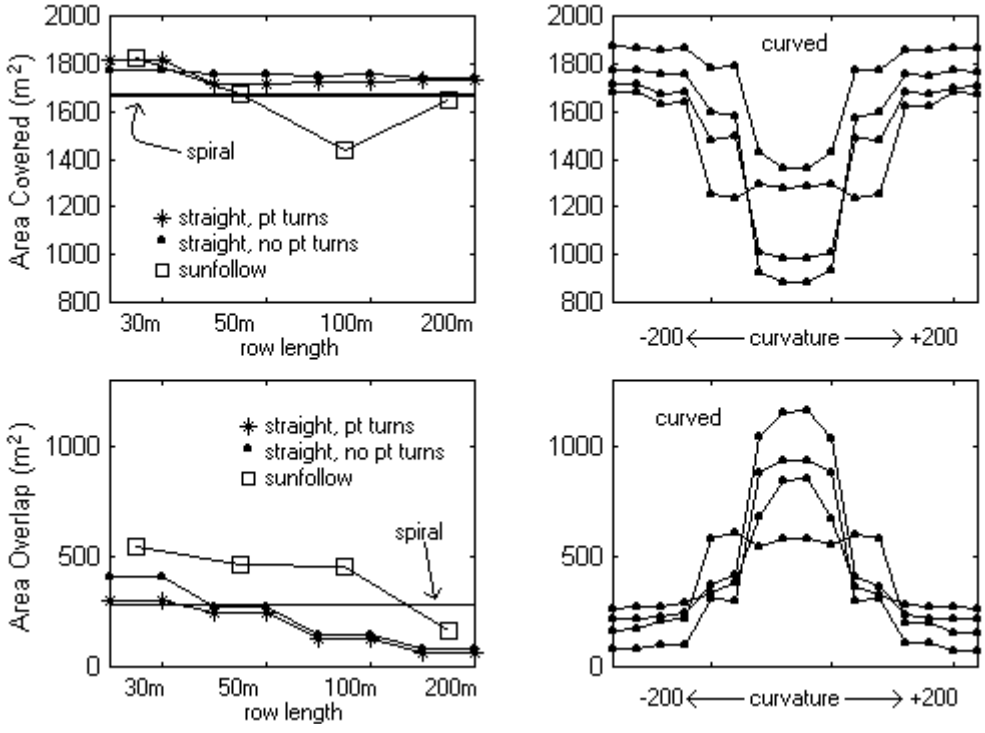


Figure 25: Area covered and overlapped by range of pattern types

Area coverage and area overlap evaluations did not vary much for different latitudes or seasons, as should be expected since location does not affect pattern shape in general. The only differences occurred in the sun-following pattern, since the shape of that pattern does change depending on the speed and motion of the sun. Even so, these evaluations did not change a great amount due to latitude. The differences are more noticeable between pattern types, as shown in Figure 25, for Earth's summertime. Spiral, straight rows and sun-following patterns are shown on the left as a function of row length, while curved patterns are shown on the right as a function of curvature. Note that spiral patterns are independent of row length.

The area coverage for sun-following patterns generally decreases as the row length increases, for a fixed row width, as longer rows mean that adjacent rows will curve back on themselves more, as shown in Figure 26. The 200m row length case has an increase in coverage, since for the limited total pattern time, that pattern did not have as much time to curve back on itself, resulting in most of the pattern adding to area coverage and not area overlap.

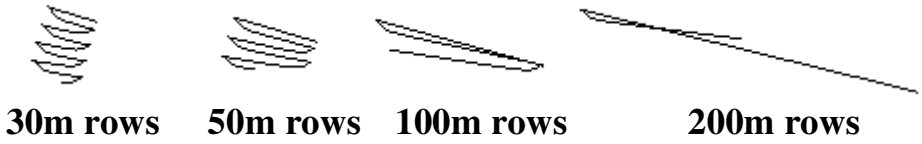


Figure 26: Varying row length for sun-following patterns

Power consumption also did not vary much with latitude, but primarily with the number of point turns and frequency of turning radii changes. The patterns with longer row lengths spend more time driving, and perform fewer point turns, so they consume less power. The straight rows patterns require changing the turning radii less frequently, so they also consume less power. Graphs of power consumption from Earth's summertime are shown in Figure 27, with curved patterns on the right as before.

For consistency, the starting time for each location, latitude and season was selected such that one hour later the sun would be at the peak elevation for the day. The starting yaw was then selected such that one of the side solar panels was optimally oriented for that starting time, pointing towards the sun. Note that in these tests, the panel configuration remained the same, and did not change based on the expected sun elevations for a particular latitude. This situation would

be the case if a single robot were to traverse a range of latitudes, but have either fixed or limited motion solar panels. All the patterns for a given location and season were evaluated for the same starting conditions, however, though the sun-following pattern automatically adjusts its starting yaw to the optimal heading.

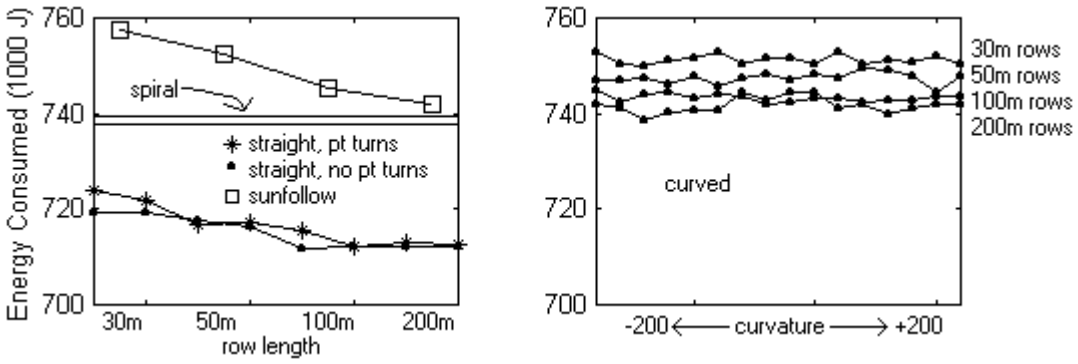


Figure 27: Energy consumed by range of pattern types

Sample graphs of the total sun energy generation are shown in Figure 28. The left side shows the values for all the different pattern types as a function of row length, for 70S on Earth during the summertime. The right side shows the values for the four sun-following pattern variations, for 0S to 80S during Earth's summertime.

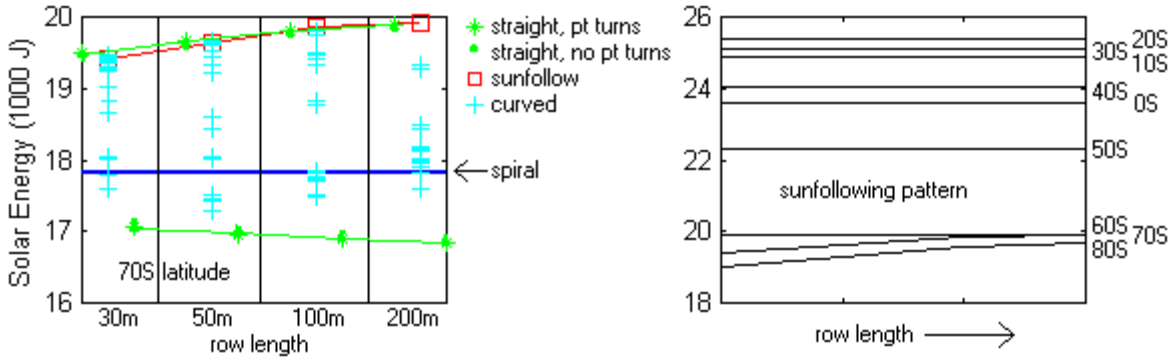


Figure 28: Solar energy generation

There are three points concerning the 70S latitude figure on the left. One, the two green lines for the straight rows patterns are for two different initial robot headings, 0 and 90 degrees relative to the sun's initial azimuth. The reason the 90 degree heading is evaluated at all is because if longer patterns were being considered, the sun would move enough during that time for the 90 degree heading to be more optimal. For this short 30 minute pattern, the 0 degree robot heading is

nearly optimal for solar power generation for the entire pattern duration, causing those patterns to generate basically the same amount of energy as the sun-following patterns.

Second, the range of plus signs for each row length in the plot is for the varying curvatures of the curved patterns. Certain curvature values are very similar to the value used by the sun-following pattern, causing those patterns to generate nearly as much energy. For instance, for the 50m row lengths, a curvature of -300m actually generates slightly more energy than a sun-following pattern of 50m row length. This is due to the fact that the sun-following pattern is designed such that sharper point turns are used than for the curved patterns, leading to more time being spent in the non-optimal row end turning and merging into the next row. This difference between 300m radius curved patterns and sun-following patterns, and in fact between straight rows with and without point turns, causes the slight, mostly negligible, differences that are just enough to push one pattern's values above another's. The current time spent in non-optimal headings is about two minutes per point turn, so altering the required point turns to calculate a better angle for merging into the next row, and not just the best angle for the current sun position, would reduce that time.

Third, the increasing value of the sun-following pattern as a function of row length is due to fewer rows being traversed in the allocated pattern time. Fewer rows means fewer row end turns, which means the robot is less often put into a non-optimal heading during the merging back into the next row. This effect is more pronounced for higher latitudes, as shown in the figure on the right, where the low sun angles, in combination with the solar panel configuration, causes the power generation to be more sensitive to relative solar azimuth changes.

One other point needs to be made concerning the figure on the right showing sun-following results for a range of latitudes. The total energy values first start to increase with latitude, beginning at the equator, reach a peak at 20S, and then begin to decrease. The reason for this is the angle of the solar panels attached to Nomad is actually 40 degrees from vertical. For a more horizontal sun, such as at higher latitudes, only one panel generates power at a time. For lower latitudes, with higher sun angles, even the panel facing away from the sun generates some power since the sun is shining down more from above. The maximum sun elevation is the highest at 20S latitude for this date, reaching a peak elevation of 88.3 degrees.

As an example of how to use these pattern evaluations, consider the case of 60S latitude, 0 longitude, on July 15, 2000, in the middle of winter. The sun reaches a maximum elevation of 8.86 degrees at 1200 hours Universal Time on this date. The set of 86 patterns was simulated for these conditions, with the patterns starting at 1100 hours and lasting a little over half an hour. Considering each evaluation separately, pattern 22 (sun-following, 200m rows) produced the most solar energy, pattern 72 (curved, 30m rows, 300m curvature) covered the most area, pattern 16 (straight rows, 200m rows) overlapped the least area, and pattern 9 (straight rows, 100m rows, no point turns) consumed the least energy. However, a combination of these evaluations can also be considered. A cutoff can be used for the amount of energy consumed as a percentage of solar energy generated. All patterns which produce a percentage less than a certain amount can then be ranked according to the other evaluations. Consider an equally weighted evaluation function of area covered and area overlapped, for all those patterns passing an initial cutoff test: rank = 1.0 * area covered - 1.0 * area overlapped. The highest ranked pattern is then pattern 14, a straight rows pattern of 200m rows, with point turns.

Other types of weighted rankings, cutoffs, evaluation categories and pattern types can be considered, as the situation warrants. For instance, instead of considering a fixed pattern time, a fixed pattern length can be used, and the total time used as an evaluation category. As an example, a straight rows pattern of 3 10m rows was simulated both with and without point turns. The pattern with point turns resulted in a 13.5% savings in total time to complete the pattern. Different starting times can also be simulated for comparison, to find the best time to start a pattern. Table 6 shows the percentages of maximum solar energy generation for a set of 3 simulated patterns lasting about 2 hours at Elephant Moraine, starting at different times on January 15th, 2000. Other simulation comparisons are described in [80].

Time of Day	sun-following pattern	straight rows pattern	spiral pattern	sun elevation (degrees)
1200 UT	100.0%	94.2%	69.5%	8.6 to 7.4
0 UT	100.0%	96.2%	85.0%	33.6 to 35.1

Table 6: Total solar energy generation comparison, Elephant Moraine

The other purpose for evaluating this large set of patterns is to determine general characteristics to aid in selecting a subset of patterns which is still representative of the range of evaluation values. Evaluating 86 patterns, or more, for every environmental situation can be time-consuming and unnecessary. Without an analytical means of evaluating all possibilities, choosing the best patterns to simulate can be tricky. A few characteristics have emerged from testing, however, which can be used to limit the choices.

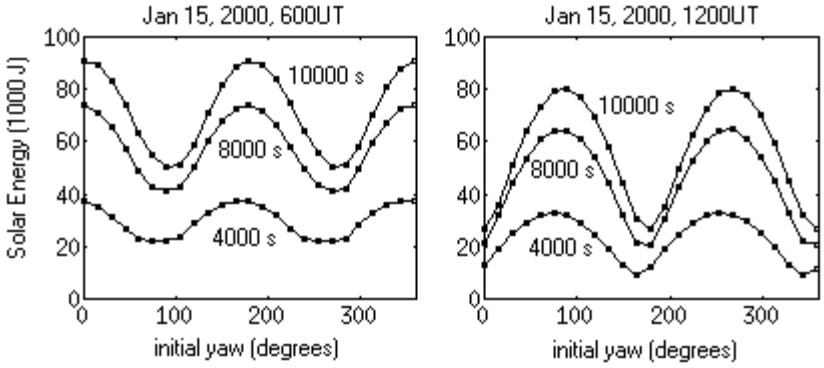


Figure 29: Variation in straight rows energy generation due to initial robot yaw

For the straight rows pattern, the initial robot heading, the row length, and the use of point turns can be varied. The row length can be set to a single value based on the area to be covered for a given mission. The use, or non-use, of point turns will then double the set of patterns considered. The initial heading, however, could potentially range over 360 degrees. As shown in Figure 29, varying headings produce a sinusoidal variation in energy generation. The different lines are for different starting times and total pattern times. Longer patterns might be expected to have less of a sinusoidal outcome, as the changing sun azimuth begins to smooth out the differences between starting yaws. The patterns considered here are frequently only performed for periods of time considerably less than a day, however, as larger, more irregularly shaped regions will need to be covered by a combination of several smaller patterns. For the 10,000 second, or 2 hours and 45 minute, patterns, the solar azimuth only changes by about 41 degrees, leading to a power dropoff of only 25% from optimal by the time the robot reaches the end of the pattern. Most of the pattern is performed with even smaller power dropoffs. Thus, the sinusoidal results still persist for that longer pattern. Other evaluations like area covered, area overlap, and energy consumption (given benign terrain) will not vary. Therefore, a set of headings ranging over only 90 degrees can span the entire range of possible evaluation results, for a given starting

time. If it is possible to determine a maximum or minimum result, then just two simulations, with headings 90 degrees apart, need to be performed to span the entire range.

For spiral patterns, only two variations are possible, given a specific row width and initial robot yaw: clockwise or counterclockwise. These two variations do not result in greatly varying evaluations, so a single spiral pattern can be sufficient. Of course, the initial heading can also be altered, but since the pattern continually spirals around, the heading does not have a great effect on the final results. The row width for both spiral and straight rows patterns should be based on the field of view of the sensors used for coverage, and fixed ahead of time. For the sun-following pattern, the primary variation is row length, as the initial heading is automatically calculated. If a given size area is to be covered, then the length can be set, and only a single sun-following simulation performed. Variations in row width can also be simulated, as this pattern generally has a greater amount of area overlap.

For curved patterns, sharp curvatures in combination with long row lengths produce generally undesirable results. These patterns turn in on themselves, causing large amounts of area overlap and covering very little new area. Therefore, if long rows are to be used, then more gradual curvatures should also be used whenever area coverage is a desirable pattern trait. Another reason for only testing curved patterns with larger curvature radii is that they mimic the curvature of the sun-following pattern, generating more solar energy. Curved patterns with large enough curvature radii mimic the straight rows pattern, however, leading to a maximum identifiably unique curvature. A discrete set of curved patterns can thus be identified for simulation, but an alternate heuristic for curved patterns is discussed below.

PATTERN CURVATURE

A set of 74 curved patterns were simulated for a range of locations and for two different row lengths, examining the evaluation results for a more continuous range of curvatures. Plots of these evaluations have then been fitted with polynomial functions of the curvature, showing that analytical expressions can be developed for this pattern type, eliminating the need to simulate a large set of complete patterns. This is a more formal method for generating and evaluating

patterns, with the possibility of exploring intermediate versions of patterns in a continuous progression from one pattern to the next.

Plots of the evaluation results are given below, along with an example function. Higher order equations can always be found to fit the data more accurately, though in many cases, lower order equations are sufficient even when considering sparse data points. In addition to using these equations to predict the individual evaluation totals for intermediate curvatures, a set of equations for each evaluation category can be combined in an optimization function. Evaluations can be weighted appropriately for the desired mission attributes, as before, and the optimal curvature calculated using the set of equations. Instead of considering all the results for a large set of patterns, and just selecting the best pattern out of a discrete set, a single optimization calculation can be performed to find the optimal pattern curvature considering a variety of evaluation categories. This thesis research did not include performing such optimizations, but the purpose of this work is to demonstrate that analytical equations are possible for a range of evaluations of curved coverage patterns. Further work could also address the issue of incorporating other variables, such as row length, into a single equation.

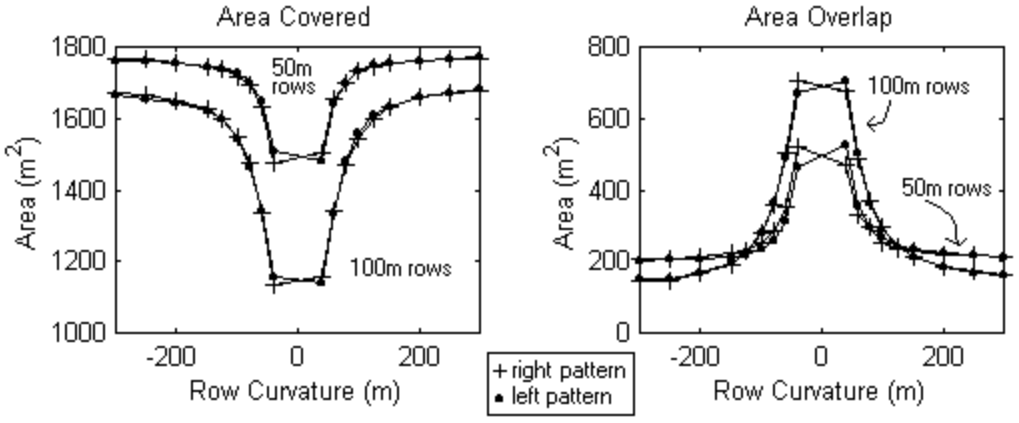


Figure 30: Area covered and overlapped as a function of pattern curvature

Area covered and area overlap as a function of curvature radius are shown in Figure 30. Negative curvatures are to the left, positive curvatures are to the right, and straight is technically an infinite curvature, though in practice, curvatures greater than 400m are basically straight for the row lengths considered. Simulations with 50 meter rows and 100 meter rows were both tried,

and with the direction of pattern travel to the left and to the right. The direction of travel has only a minor effect, especially with larger radii of curvature. The row length has primarily a scaling effect on the results, though with some change of function shape.

The following is an example of a polynomial equation fitting the data points for the area covered by patterns with 100m long rows, traveling to the right, where x is the pattern curvature.

$$\text{Area covered} = -878394.65x^{-2} + 86.71x^{-1} + 1654.97$$

The maximum error resulting from using this equation, compared to evaluation results from complete pattern simulation, is less than 5.8% for the range of curvatures considered. If a 4th order polynomial is used instead, then the error is reduced to less than 0.9%. For the area overlap, a 4th order polynomial produces errors of less than 4.7%.

Figure 31 shows the energy consumed by curved patterns as a function of curvature. The left side shows the same four sets of simulations as in Figure 30, with 50-100m row lengths, and traveling to the left or to the right. The right side of the figure shows just the data from the 100m row patterns, traveling to the left, with curves for a 4th order and a 6th order polynomial fitted to that data. The errors for the 4th order polynomial are slightly more than 2.0%, and for the 6th order polynomial are less than 0.3%.

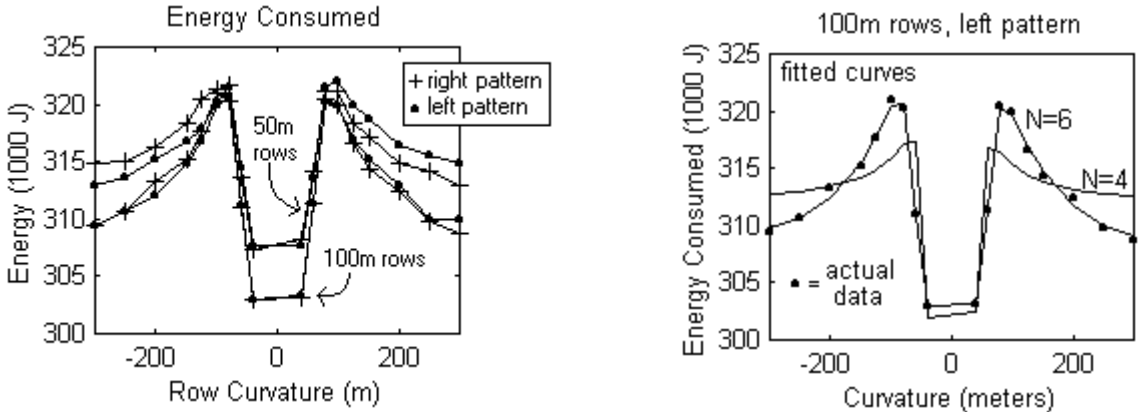


Figure 31: Energy consumed as a function of pattern curvature

All of the above evaluations are independent of location and latitude. The solar energy generation, however, depends on both. Figure 32 shows the energy generated for 4 different

latitudes, from 70 S to 85 S, for both the Earth and moon. Latitude primarily has a scaling effect on the total energy generated. Data points from complete simulations are fitted with 4th order polynomial approximations, for each latitude. The errors for these approximations are all less than 2.4%, and the mean error for the moon locations is about 0.33%.

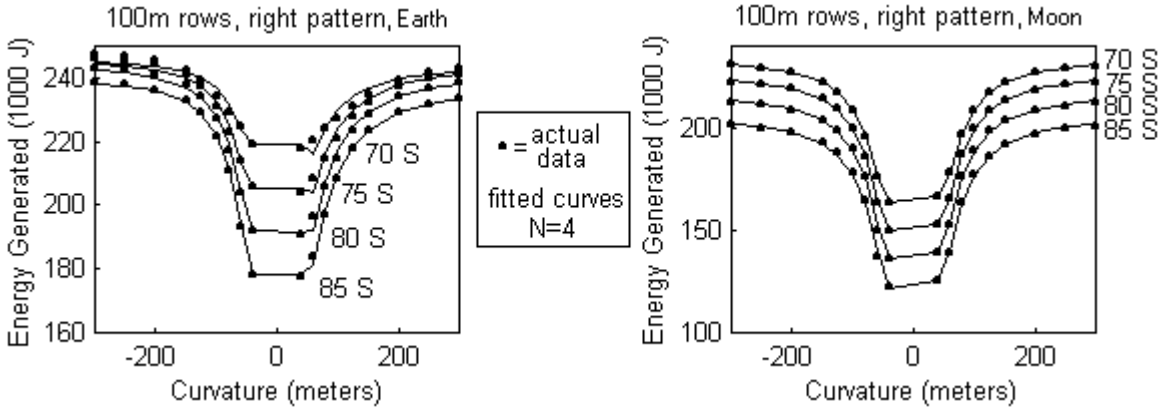


Figure 32: Solar energy generation as a function of curvature

TERRAIN EFFECTS

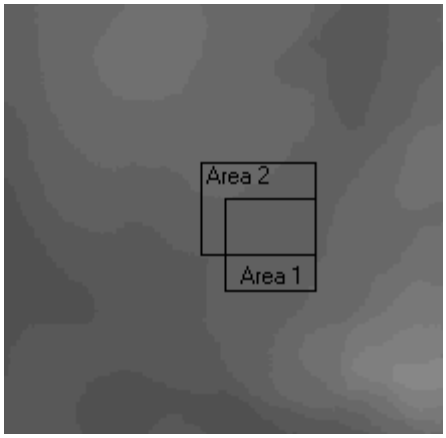


Figure 33: Covered regions of Haughton Crater

For the previous simulations, shadowing caused by the terrain has not been considered. This section describes the evaluation of coverage patterns performed in a more hilly area, with shadows to limit solar power generation. Initial results with a cratered terrain are given in [80]. The following simulations were performed in a portion of Haughton Crater, in the Canadian Arctic. Straight rows patterns, with a variety of starting points and times, were simulated. The

same patterns were also simulated without any terrain shadowing, to understand how much effect the shadowing has on solar power generation. Figure 33 shows the two regions in which patterns were performed, with the rectangular boxes indicating the covered areas. Each of the four corners, for each area, was used as a starting point, with patterns running either north-south or east-west. Four different times of day on July 15, 2001, were used as starting times for each starting location, resulting in 32 pattern plans for each area.

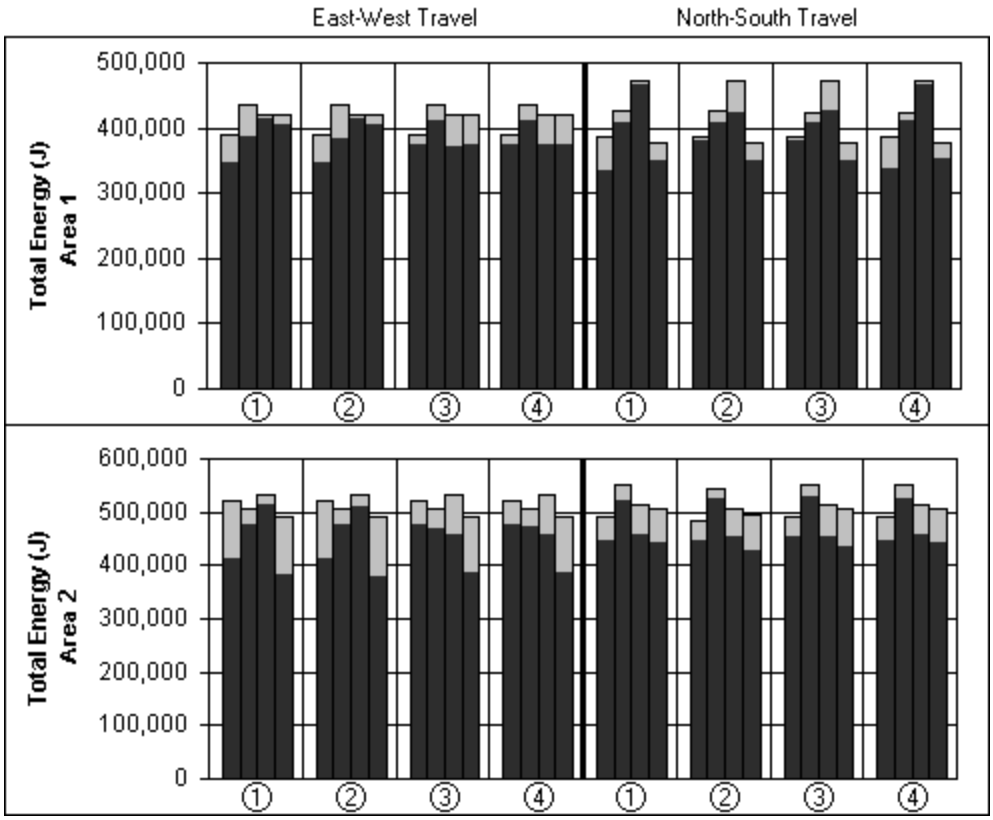


Figure 34: Effect of terrain shadowing on solar power generation

Figure 34 shows the energy generated during the approximately 16 hours of each of the plans. The dark grey indicates the amount of energy which would be generated considering the terrain shadowing, while the addition light grey on the top of each bar indicates the extra energy that would have been generated without shadows. The plans are grouped by location, with the circled numbers indicating the starting corners, clockwise around the regions starting with the northwest. The first set of four corners is for the plans with rows traveling in a east-west orientation, while the second set is for the north-south orientation. The four bars within each grouping are for the

four times of day: 0UT, 600UT, 1200UT, 1800UT. The regions are more shadowed during the first eight hours of the day, due to the hills to the north, with the shadows moving primarily from north to south and back to north, and secondarily from east to west. The hills to the southeast do not produce as many shadows because the sun is higher when it is in shining from that direction.

For such nearly day-long patterns, at one point during the plan the sun will be at an optimal orientation with respect to the primary row direction. For Earth locations, such an orientation continues to be relatively optimal for an hour or so. Timing the pattern such that this optimal time occurs when the robot is unshadowed improves the total energy generation. Timing the pattern such that less shadowing occurs overall also improves the total energy. For example, in Area 2, with East-West travel, starting corner 1 shows marked differences between starting times. With no terrain shadowing considered, starting at 0UT or at 1200UT only causes a total energy generation difference of 1.6%. When terrain shadowing is added, that difference increases to 19.6%. The length of time the robot will be in shadow when starting at 0UT is nearly 4 hours out of a total pattern time of 20.5 hours. Starting at 1200UT results in shadowing for only 30 minutes, 17.4% less time than at 0UT. Considering all 32 plans, the increase in total energy generation when selecting the best plan is over 39% more than the worst plan, in both regions.

With no terrain shadowing, the starting corner makes practically no difference in the total amount of solar energy generated, with the only differences arising from the row direction (north-south or east-west) and the starting time. With terrain shadowing, the side on which the robot starts also makes a difference in the total energy generated. For instance, with east-west rows, the two northern corners 1 and 2 produce basically the same results, while the two southern corners 3 and 4 produce another set of results. For patterns traveling north-south, the eastern and western corners are paired. This is because the time spent traveling along a single row is negligible with respect to the motion of the sun, while the progression across the width of the area is more affected by the sun's motion. Thus, when attempting to select a starting location, the choice of locations to evaluate can be reduced by half. So although the starting locations, headings and times are all interdependent when terrain shadowing is considered, a reduced set of pattern plans can be evaluated through the planning simulator developed for this thesis.

A robot will take from 22 to 36 minutes to drive to a different corner for these areas at the simulated robot speed of 15 cm/s. Doing so before starting the search pattern may result in a considerably larger final energy production, so assuming the robot's other mission constraints such as completion time do not prohibit such an action, the mission planner can use the pattern evaluation information to pick a much preferred starting corner and time. Aside from just desiring to produce more power or to have a higher safety margin due to uncertainty, choosing an alternate starting location and time may be the only possible option. The power consumption needs of the robot may require higher power generation for some robot power configurations or for more highly shadowed regions. Instantaneous power generation may also be more of a concern than total energy generation, depending on the capacity of a robot's storage batteries. It is conceivable that a single starting location and time may be the only possible way that a robot can cover a certain area. Simulation of the options as demonstrated here will determine if any of the plans considered are possible considering the robot's needs.

MISSION SCENARIOS

Quantitative improvements in robot lifetime, efficiency and productivity can be calculated for specific robot missions based on coverage patterns. This section describes the results from simulating two basic types of missions. For one mission, a robot must drive for a certain distance, covering as much ground as it can during that traverse. This distance results in five and a half hours of driving, for the defined robot speed, not including any stops for recharging or target investigations. In most actual cases, a solar-powered robot would also have batteries on board, and so the robot performing this mission is assumed to have a battery of size from 1.0 to 2.5 amp-hours. Larger solar panels and a reduced-scale power consumption model, compared to Nomad, are used in order to simulate a truly solar powered robot. For this mission, when the robot's battery reaches a critical level, it must stop driving and recharge the battery, losing productive time during the middle of its mission.

The power generation and consumption values were chosen such that even when driving at an optimal angle with respect to the sun, the robot still consumed more power than it generated. This was done so that battery lifetimes could be examined for the chosen mission lengths. For actual

missions, a robot's battery size and solar panel size would be selected such that power consumption costs of moderate driving would not drain the battery, but could be supported solely by solar power. However, using such a model would require week-long or more missions to be simulated in order to gain any statistical results on robot lifetime. Such actual missions will still be limited by weight and power constraints, and will still benefit from having larger battery safety margins. These results, then, still reflect the potential benefits and improvements for longer-term missions, at a scaled-down level.

The lifetime of a robot is defined as the amount of time the robot can continue performing its mission before having to stop and recharge its batteries for the first time. This is the total length of time such a robot could be active if it did not have rechargeable battery reserves. Figure 35 shows the differences in lifetime for a robot if it were to choose different types of coverage patterns to pursue its mission, for an 80S latitude on Earth on January 15, 2000. Similar simulations were run for latitudes from 60S to 90S on both the Earth and the moon, producing similar results. The set of plans simulated include four varieties of straight rows, shown in dark gray, two varieties of spiral in diagonal lines, one sun-following pattern in light grey, and sixteen types of curved patterns in dots. The sun-following pattern clearly has the longest lifetime, though this is not always the case. In some cases, the additional power consumption required by the point turns in a sun-following pattern is enough to offset the additional solar power generated as compared to some of the straight rows and curved patterns, which can spend less time performing point turns.

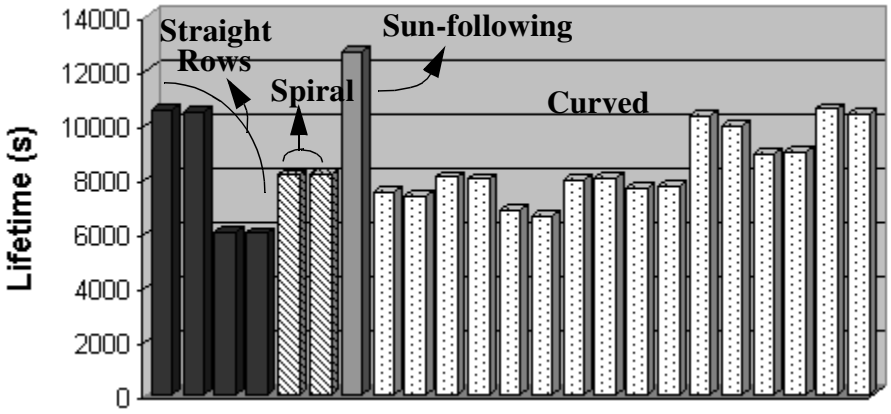


Figure 35: Robot lifetimes for 23 pattern types - 80S, Earth, summertime

Even though the sun-following pattern may not cover as much area initially, a robot whose power consumption is just barely matched to its power generation capabilities may be able to continually survive only as long as it maintains its orientation to the sun. Such a robot may be able to follow a sun-following pattern much longer than it could follow a straight rows pattern, for example, and eventually cover a greater amount of area, at the expense of more area overlap and more total mission time. For the case in Figure 35, 80S on Earth, the sun-following pattern does eventually pass the first straight rows pattern plan in terms of area coverage, even for the same amount of mission time, due to the multiple recharging stops needed for the straight rows pattern. Two recharging periods are needed while performing straight rows, while only one is needed by the sun-following pattern to complete the designated mission. At the end of the straight rows pattern's first recharging stop, the sun-following pattern's area coverage is still less, but by the time the straight rows pattern must recharge the second time, the sun-following pattern coverage begins to pass that of the straight rows. At a mission time of 6 hours, 49 minutes and 49 seconds, the area coverage of the two plans is equal. At the final sun-following mission time of 7 hours, 8 minutes and 29 seconds, its area coverage is 664.25 m^2 larger than the area coverage of the straight rows pattern at the same time.

Figure 36 shows the area covered by each of the pattern plans at the time their first recharging stop is required, which is a different time for each pattern. The longer lifetime enabled by the sun-following pattern results in that plan still covering a large amount of area by the time the robot must first stop and recharge.

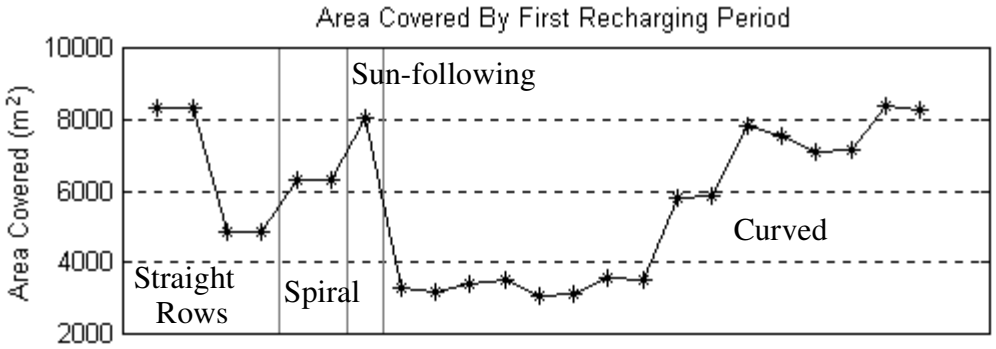


Figure 36: Area coverage during robot "lifetime" for each simulated pattern

Figure 37 shows the area covered by each of the plans after a fixed time, equal to the shortest mission plan found, which is the sun-following plan. By this time, the sun-following pattern's coverage has passed that of most of the other patterns, with the exception of two of the straight rows patterns. For longer missions, the sun-following pattern might also pass those patterns. Determining which pattern will allow the mission to cover the most area then depends on the total mission time length. As another example, the 3rd and 4th straight rows patterns are started with a heading 90 degrees different than the first 2 straight rows patterns. For shorter mission times, the first 2 patterns perform better, but for longer mission times, the second 2 patterns perform better, due to the sun traveling farther across the sky.

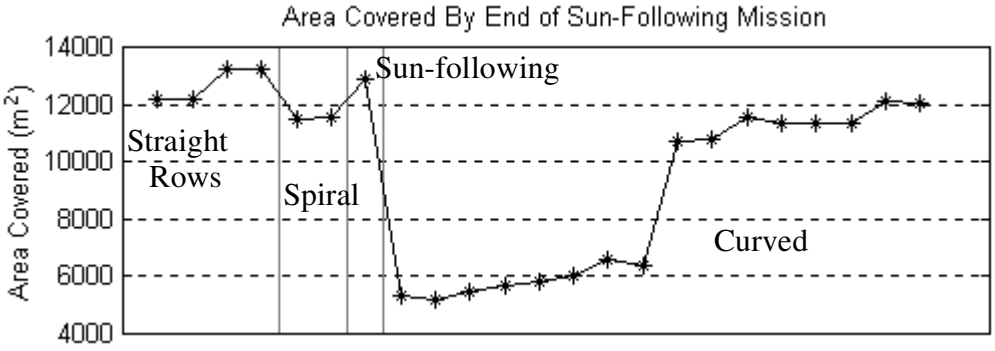


Figure 37: Area coverage after fixed mission time

The second basic type of mission is for a robot to again drive for a certain distance, but this time with targets randomly interspersed throughout the region. Whenever a target is encountered, the robot stops to investigate. The time for investigation is generated randomly, with a gaussian distribution centered around five minutes. Two variations of this missions were simulated. In one, the robot stopped right where it was to investigate the target. In the second case, the robot point turned to reach the optimum solar power generating orientation before beginning the investigation, and then point turned back to the original heading before continuing to drive. The proper orientation to turn to is determined using the solar ephemeris library. For some sensors, the time required for investigation can be extensive, so for a solar-powered robot, being able to generate power at the highest rate possible during target examinations can be essential.

Two additional measures can be introduced here: efficiency and productivity. Efficiency relates to the total time the robot takes to complete the mission. Different pattern types are more efficient than others, allowing the mission to be finished in a shorter amount of time. The total driving time and distance driven is fixed for all the patterns, with the only differences occurring due to varying solar power generation, which leads to variations in the amount of time needed for recharging. The improvement in efficiency is the percent of time saved for a particular pattern type as compared to the slowest pattern type in the set of 23 simulated patterns. As the worst possible pattern type is not even included in this set, these percentages underestimate the benefit of intelligent pattern selection. Productivity is the percent of time spent by the robot doing productive work - driving and examining targets - out of the total mission time. For missions without target stops, productive work is just driving. The improvement in productivity then is the difference between a particular pattern's productivity percentage and the worst pattern's productivity percentage. Again, this measure will underestimate the potential productivity improvements since only a limited pattern selection is simulated.

As mentioned earlier, the mission definition in these simulations is driving for a designated length of time, not counting time spent investigating targets or recharging. An alternative mission might be to cover a designated amount of area, or to investigate a specified number of targets. For general exploration of a new area, defining a number of targets to see is not suitable, since the number of targets in the area may be completely unknown. Searching an area and finding no targets at all would be equally informative. Covering a specific amount of area is more useful, although the amount of area within the scope of a robotic mission will generally be a very small percentage of the total unknown area. For example, an Antarctic-exploring robot will most likely never be able to cover the entire continent. In such cases, complete coverage is not a goal or even an option. Covering the most area during a given time can be one criterion of mission efficiency, but the simulations performed here are evaluated more for their power generation abilities, as shown by their ability to spend more time working and less time recharging batteries. The patterns may cover less new area, and produce more area overlap, but this can be useful in new locations, due to imperfect sensors and the need to take multiple data readings. Driving over previously covered ground or re-investigating a target can be just as useful as exploring new territory, as long as the robot continues to be productive and does not have to pause to recharge.

The targets in these simulations are randomly distributed through the entire region, with a designated frequency. Several different target distributions are used, and the results generally vary for the different distributions, as shown in Figure 38, though several large scale similarities are observed, such as the fact that the sun-following pattern tends to always require less recharging. Different patterns result in different amounts of coverage and overlap, with different numbers of targets being investigated. The variations indicate that such randomized simulations cannot accurately predict which pattern is best for a specific location with specific targets. For best results, knowledge of the target locations should be known ahead of time in order to evaluate which pattern is best in a given situation.

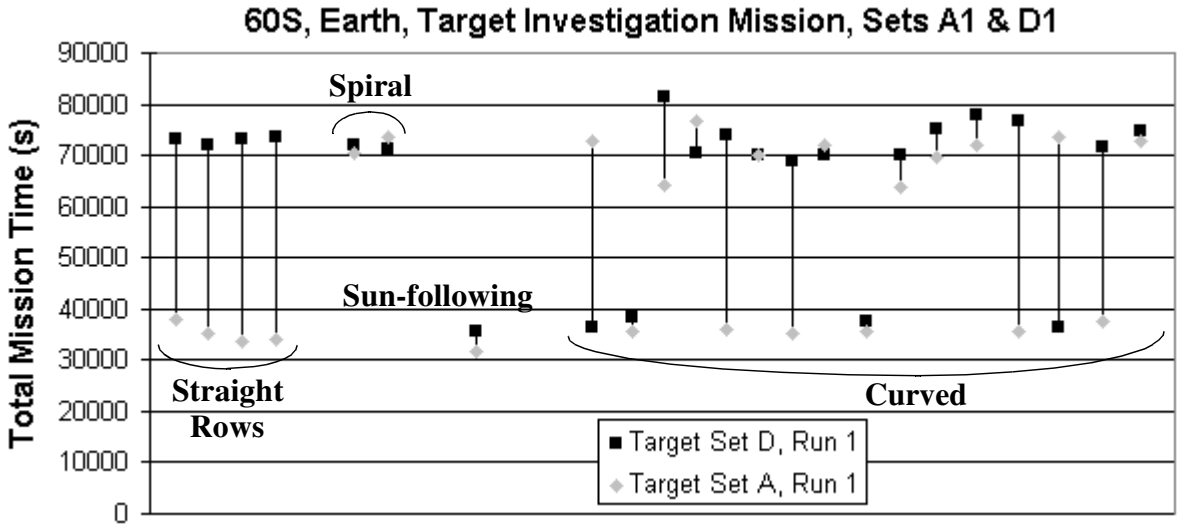


Figure 38: Results from multiple target distributions

A solution to this problem would be to perform a survey mission first, such as the previous mission scenarios where the robot never stops to investigate targets, but just drives steadily. During such a mission, target locations can be identified and used in simulations of the next mission stage where the robot will stop to investigate any targets. The initial survey mission can be accurately simulated without prior target distribution knowledge and the best pattern chosen for that mission. Then, with the gathered information, the best pattern can be found for the more detailed second-stage mission. Other missions may involve stopping, not for specific targets, but at pre-defined intervals to take sample readings. This defined "target" distribution will enable valid simulations and predictions to be made as well, without prior survey knowledge.

Multiple simulations were performed in this research with the same target distributions, but with different randomized investigation times, and those multiple runs produced very similar results, as seen in Figure 39. Three different runs are indicated in this figure, with mostly overlapping results with insignificant differences compared to pattern differences, with the exception of curved pattern 21. One of those runs, indicated by the black square, needed one less recharging period, while the random timing differences lengthened the other two runs (the light grey circle hides the dark gray diamond) just enough to require more recharging. This indicates that primarily the number of targets and their locations have a large effect on pattern evaluations, while smaller timing differences for the same set of targets have a more minor effect. The average difference in total target investigation times between target set A and target set D, for all 23 pattern plans, is 6320 seconds, with a average difference in number of targets of 13. However, the average difference between runs 1, 2 and 3 of target set D is only 711 seconds, with the same number of targets for each plan. In fact, the timing difference for pattern 21 shown below is less than 20 seconds, showing that the two runs which needed extra recharging were just barely long enough to drain the battery below the critical level. An option in this case would be to have the mission planner know how much longer the mission will be lasting, and if it is almost complete, then the battery level could be allowed to drop slightly below the critical level.

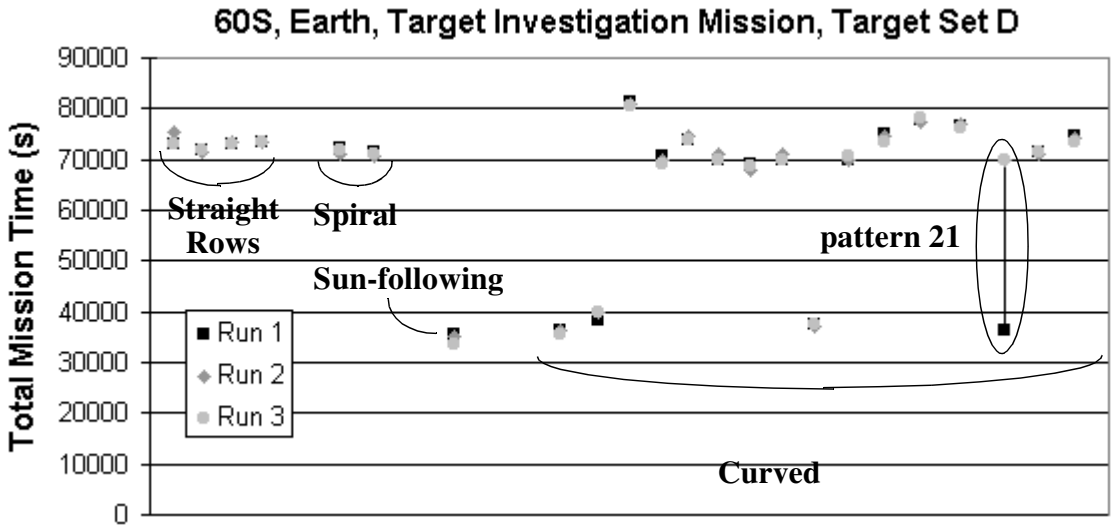


Figure 39: Results from multiple runs, same targets, random timings

Further results from this series of mission simulations are performed to demonstrate the magnitude of improvements possible with consideration of sun and shadow effects. Having this simulation capability on-board a robot, where new information can be incorporated as it is obtained, will allow simulations and evaluations to be made for the specific situations a robotic explorer encounters. The following performance improvements are examples of what might be expected for similar missions when solar information is part of the evaluation process.

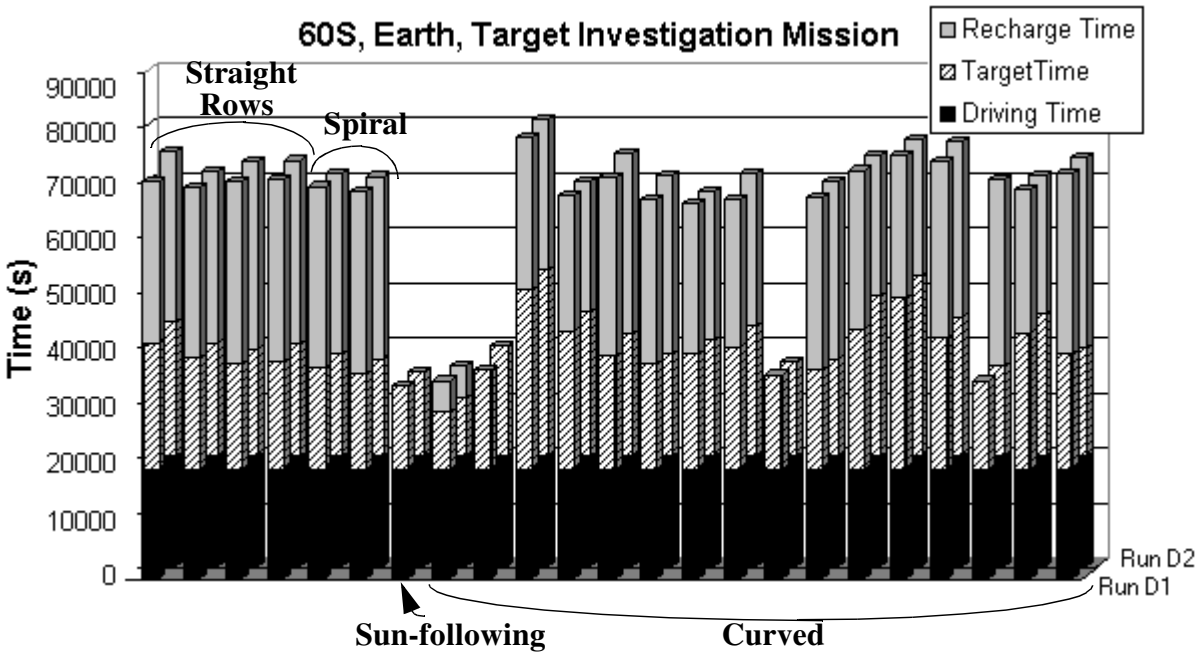


Figure 40: Mission times for 23 pattern types - 60S, Earth, summertime

Figure 40 shows the total time results for two of the target investigation missions, with point turns to orient to optimal power generation poses during target stops, for 60S latitude on Earth, on January 15, 2000. The pattern types are in the same order as in Figure 35, and the two sets of results are for two runs with the same target distribution, but different randomized target investigation times. This selection of results most clearly demonstrates the differences in mission efficiency for different pattern types. These missions were started one hour before the peak sun elevation, but at 60S, the sun is not above the horizon 24 hours a day as it is at higher latitudes. For run D2, three of these plans, sun-following and two curved patterns, did not need to spend any time recharging the robot's battery, as the time spent examining targets was sufficient to keep the battery above the critical level. For run D1, an additional curved pattern did not need recharging.

For both runs, the first curved pattern plan did need to recharge, but managed to do so in a short amount of time. The remaining plans had the misfortune of needing to recharge the robot's battery after the sun had set. The robot had to wait the entire night until the sun rose again in order to finish recharging, greatly extending the total mission time.

	Lifetime	**Efficiency**	**Productivity**
60S, Earth	23+%	0% - 18.4%	0% - 14.3%
80S, Earth	112+%	30.5% - 30.8%	24.0% - 30.5%
89S, Earth	143+%	30.5% - 30.9%	24.1% - 30.5%
60S, Moon	123+%	29.0% - 30.9%	24.1% - 29.0%
80S, Moon	161+%	35.6% - 47.6%	22.7% - 47.6%
89S, Moon	148%-153%	25.3% - 42.4%	16.9% - 27.0%
60S, targets, Earth	XX	52.8% - 61.3%	47.4%-66.9%
60S, targets, Moon	XX	25.7% - 27.8%	0%
80S, targets, Earth	XX	30.5% - 41.4%	17.0% - 17.1%
80S, targets, Moon	XX	41.2% - 57.3%	22.4% - 37.8%
89S, targets, Earth	XX	20.8% - 27.0%	16.4% - 16.6%
89S, targets, Moon	XX	41.7% - 50.8%	40.1% - 51.6%

Table 7: Improvement metrics for mission scenarios

Lifetime, efficiency and productivity results from all these simulations are shown in Table . The numbers are the maximum percentage improvements based on the set of 23 patterns simulated for each mission scenario, comparing the worst and best patterns from the set for each defined metric. The first six rows are from the mission simulations with no targets. The range of percentages is due to the range of battery sizes considered. In some cases, a maximum lifetime improvement cannot be calculated for larger battery sizes because at least one of the pattern plans enabled the mission to be completed before any recharging was needed, limiting the maximum lifetime to the total mission time. The efficiency and productivity improvements are zero for 60S on Earth with large batteries, since all the pattern plans in this case completed the mission without any recharging periods. The last three rows are from the mission simulations with target stops.

Only one size of battery was simulated. The range of values comes from multiple runs and target distributions. Missions with and without point turn target orientations turned out to produce relatively similar results for the simulated target investigation times used, as the power gain from maintaining a more optimal orientation was a close match to the extra power consumed by performing the point turns. Lifetime improvements are not meaningful in these cases, since some patterns enabled the entire mission to be completed before other patterns required a single recharging period.

The productivity and efficiency improvements for the Earth decrease with higher latitudes because for lower latitudes, some patterns result in overnight recharging periods, causing greater differences between the best and worst pattern plans. At higher latitudes, the patterns are more equalized since the sun is always above the horizon. For the Moon, the productivity improvements are greater for higher latitudes since at lower latitudes fewer recharging periods are needed overall, and the different patterns produce more similar results for the defined mission lengths. The solar panel configuration used in the simulations is more closely matched to the sun angles at the lower latitudes, as opposed to the extremely low angles at high lunar latitudes. The motion of the sun is also much slower on the Moon than on the Earth, and so longer missions would be needed to highlight the pattern differences. The Moon simulations at 60S have 0% productivity improvements since none of these patterns required recharging. In this case, the efficiency improvement is due only to differences in time spent investigating targets, which varies primarily due to some patterns covering more ground than others, and stopping for more targets.

POSE AND POWER UNCERTAINTY

The pose of a robot, including its roll, pitch and yaw, determines the orientation of any attached solar panels to the sunlight, and therefore the amount of solar power which can be generated. Any unexpected changes in the robot's pose will cause the actual power generated to differ from predicted power. Power consumption predictions can also be affected, with steeper slopes producing higher power consumption. Unexpected pose changes may be frequent as the robot travels over uneven terrain and deviates around obstacles. To examine the robustness of solar simulations and predictions, pose variations were added into pattern simulations, and the

results compared with the same patterns without the pose variations. Most of the field work performed for this thesis took place on benign, mostly flat ground, with few obstacles. Predicting results for activity in this type of terrain does not have to deal with uncertainty to a large extent. However, for confirmation, actual field data was also removed of all pitch and roll variations and replayed, allowing the resulting evaluations to be compared with the original evaluations.

Pose variations were introduced by calculating random values for pitch and roll, in a gaussian distribution. The central peak of the distributions was set at 0, but with varying magnitudes of uncertainty allowed, from 1 to 8 degrees. This addresses any sporadic pose changes due to one or more of the robot's wheels driving over rocks, dips, or slope variations, which can be expected to occur in most terrain regions. For comparison, the field tests in Pittsburgh had pitch and roll variations of -5.16 to +3.44 degrees, tests at Williams field had variations of -4.01 to +2.86 degrees, and tests at Elephant Moraine had variations of -3.44 to +5.16 degrees.

Sets of 25 straight rows patterns were simulated for four latitudes from 70S to 85S, both on Earth and moon, with each pattern taking almost a (simulated) hour. Five different magnitudes of randomized pose uncertainty -- a full width at half maximum of 0, 1, 2, 5, and 8 degrees -- were simulated five times each. The resulting evaluations of power consumption and solar power generation were examined to determine the variation between different runs with a single uncertainty magnitude, and between runs with and without uncertainty. The power consumption did not show significant variation between runs with and without uncertainty at these levels of pose uncertainty, with a maximum variation of about 0.1% between mean values. The maximum variation between multiple pattern runs of the same uncertainty magnitude was almost 0.4%.

Solar power generation shows more variation. Figure 41 plots the total solar energy generated for each of the 25 simulated patterns at each location and latitude as a function of pose uncertainty magnitude. The data points for the five different randomized pose patterns within a single magnitude category merge together at the scale of this plot, showing that the variation between runs with the same uncertainty magnitude is fairly small. For each latitude, however, total energy values decline by as much as 1.23% as the pose uncertainty increases. Greater pose variations cause greater deviations from optimal panel angles, though for cases such as 70S on Earth, larger

pose variations result in some of those poses being more optimal for the given sun elevation, causing the final energy generation to be slightly higher than with smaller pose variations.

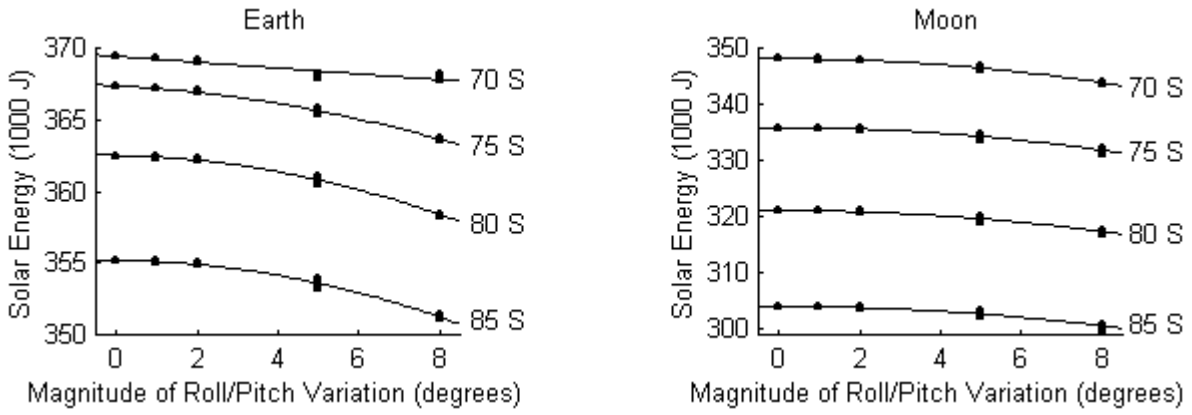


Figure 41: Solar energy as a function of pose uncertainty

In any of these cases, the variations due to uncertainty are minimal, and have a negligible effect on the validity of simulation predictions. These pose uncertainty magnitudes are similar to what would be expected in most robot-accessible locations. The position data from the three sets of patterns actually performed in Pittsburgh, Williams Field and Elephant Moraine were replayed through the evaluation simulations without the pitch and roll values. The results showed final solar power generation evaluations differed by less than 1.3%. For comparison, the power evaluation difference between the straight rows and spiral patterns performed in Elephant Moraine was 48.85%, greatly overwhelming any variation caused by pose uncertainty.

MULTIPLE VARIABLE SELECTION

The wind is another potential source of power generation for Earth, Mars, and possibly other planetoids with atmospheres. For a robot with both solar panels and a wind turbine, selecting a pattern orientation which will generate the most total power can be complicated. The orientation can either favor the sun direction or favor the wind direction. These simulations demonstrate how the evaluation method developed in this research can still enable the selection of the best pattern orientation. By simulating a set of straight rows patterns with a range of initial headings, and calculating the potential wind and solar power generation, the optimal heading can be identified.

This type of calculation seems quite straightforward, and actually is straightforward -- as long as models of the sun and wind have been developed and are available to the robot's planner.

Five different relative orientations between the primary wind direction and initial sun azimuth are simulated, with an initial robot heading also varying over a range of values. Straight rows patterns lasting from 3000 to 10000 seconds are considered, and the wind speed and direction is varied randomly with a gaussian distribution. The central wind speed peak, or mean wind speed, is one of four speeds for each simulation, from 5 knots to 20 knots. A wind turbine placed on top of a robot, with 1m radius blades so as to be scaled to the size of Nomad, is modeled. Larger solar panels are also assumed, but with the same orientation as the actual panels on Nomad. Due to the alternating direction of the robot along successive rows, and the mirrored effect of the wind coming from the left or right side of the robot, only a range of 90 degrees of robot/wind angles needs to be considered, and due to the sinusoidal nature of solar energy generation evaluations, as described earlier, only 90 degrees worth of sun/robot angles need to be considered.

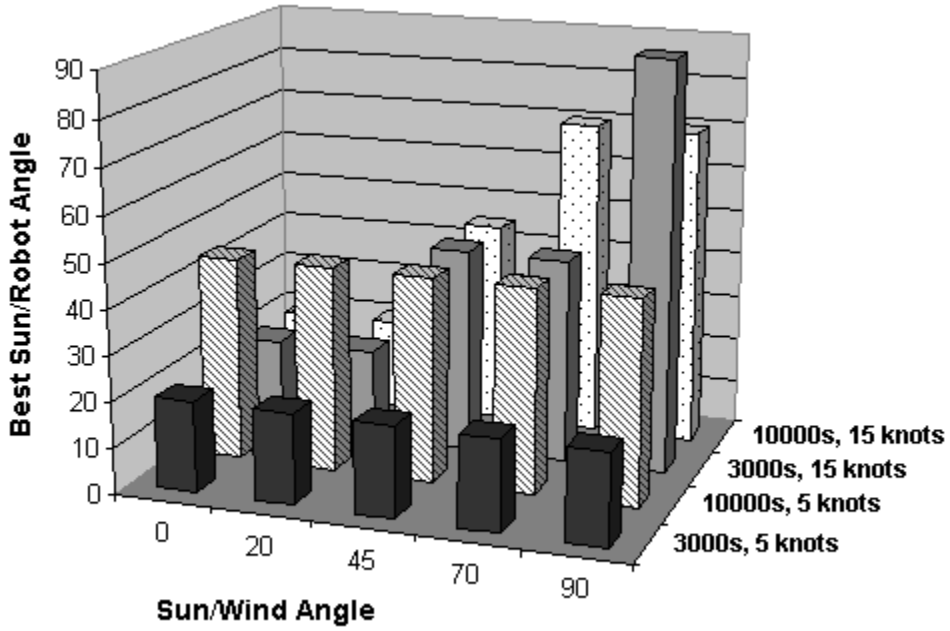


Figure 42: Sample variable wind/sun direction results from 80S

Figure 42 shows some of the results of these simulations for 80S latitude on Earth. Five different initial angles between the sun and wind direction are shown along the bottom y-axis, in degrees. The z-axis indicates the best initial angle between the sun and the robot, which will

generate the most total combined solar and wind energy. The x-axis indicates four different cases, varying both the total pattern length and the mean wind speed.

Two main points can be made about this chart, both reflecting common sense. One, for low wind speeds, the best starting angle for the robot depends primarily on direction of the sun and the length of the pattern. In these low wind conditions, the wind power which can be generated is small enough that the best angle depends solely on solar power generation. The pattern length is important because this determines how far the sun will have moved about the horizon by the time the pattern is finished. The heading of the straight portions of the pattern is only optimal at one point, with respect to the solar panels and the position of the sun. The pattern which generates the most power will be oriented such that this optimal heading occurs near the middle of the pattern, with the power generation falling off sinusoidally on either side. The second point is that for higher wind speeds, the best starting angle for the robot depends primarily on the direction of the wind. The wind power generation outweighs any consideration of solar power generation at some point, causing the simulations to prefer robot headings aligned with the wind.

For a numerical example, consider the case of a 3000s long pattern, with 15 knots winds blowing at a 90 degree angle from the initial sun azimuth. Selecting the preferred initial robot heading out of the set of simulated patterns, or 90 degrees, will generate a total energy of 618.4 kJ, an 161% improvement over selecting the worst heading out of this set.

The simulations performed here do not model all the relevant issues for wind power generation. Placing a wind turbine on top of a robot will introduce considerable wind resistance, which will increase the power consumption of the robot's locomotion system when driving into the wind. It may prove that while driving into the wind enables more wind power generation, the additional power consumption required due to wind resistance will negate the benefits. Simulations to determine whether or not this will happen have not been performed as part of this thesis. However, the evaluation structure used here to determine the best pattern orientation is still valid, assuming the appropriate physical modeling for power consumption is provided. The simulations can evaluate solar power generation, wind power generation, and power consumption, and determine the starting angle which maximizes a function of power generation minus power consumption. Even without wind power generation, these simulations could

evaluate the trade-offs between solar power generation and power consumption due to wind resistance, and select the best angle at which to drive. This thesis does not address the issues of wind resistance modeling or whether wind turbines onboard mobile robots can be productive, but instead addresses the development of a planning structure that incorporates environmental knowledge. As more knowledge becomes available and modeled, a robot with such a planning structure will be able to make more accurate and beneficial decisions.

C. Solar Navigation

Solar and terrain information has been incorporated in individual and composite shadow maps. Autonomous use of these maps is a new research field which will produce many more results in the future. Some structures for using these maps have already been developed in this research work, and will be shown here.

One type of solar navigation is sun-synchronous navigation -- finding a continually sunlit path around a terrain feature or planetary pole. An example of this is shown in Figure 43, which is a sunlit path autonomously identified for a region of Haughton Crater, for July 15th, 2001. The terrain elevation is indicated by the grey intensity value, the black lines are creeks or dry creek beds, and the yellow line is the identified path. The current implementation does not take into account the navigability of terrain, such as the steepness of slopes, and ignores the creeks, treating them as unquestionably safe ground. This path was constrained to be at least 940 meters away from the center point, but no more than 1440 meters away, and was considered to be complete when the path reached the same angular position as the starting point, relative to the central point.

The robot's speed was set at 0.1m/s, and this entire path would take approximately 18 hours. The initial position of the sun is shown, and it travels clockwise about the horizon. The starting time of 0:00 Universal Time corresponds to 8pm local time. A full 24 hours is not needed to avoid the shadows completely, since the sun elevation is high enough in most cases to not block an entire valley. The remaining time in the 24-hour day could be spent returning to the exact starting point, waiting and recharging batteries and downlinking telemetry, or possibly could be consumed by navigational difficulties during the traverse. The waypoints marked on the image

indicate equally spaced one-hour time increments, with the exception of the first interval, which is for a half hour.

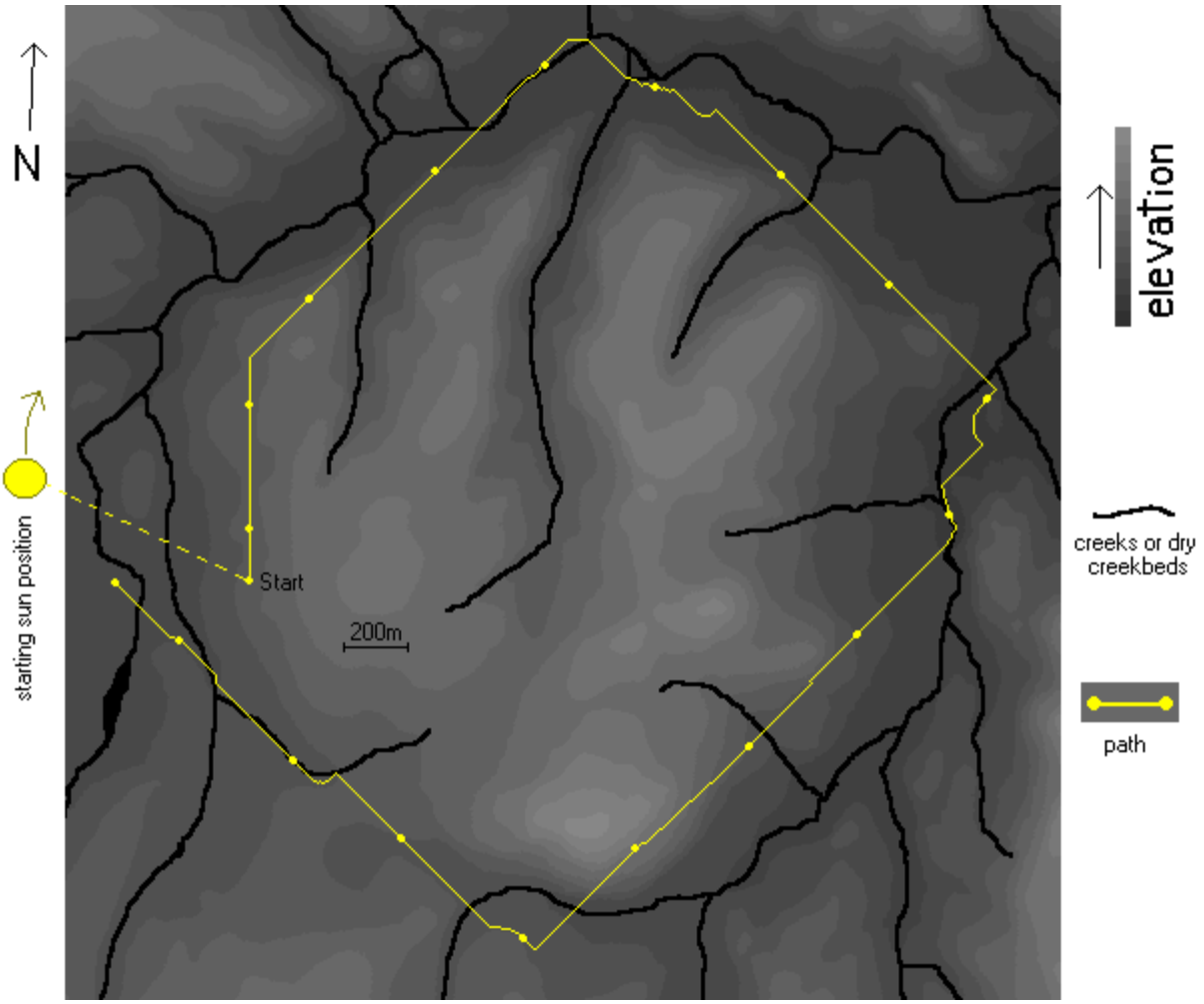


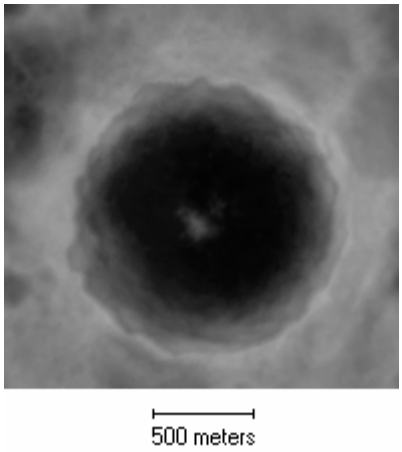
Figure 43: Sun-synchronous path around feature in Haughton Crater

The preponderance of diagonal path lines is due to the search being 8-connected, and being breadth-first. This type of search finds the first possible path, with the fewest number of steps, and a diagonal sequence of steps is the shortest way to travel around the edges of a feature, bounded by a minimum and maximum radius. The cost measure considered here is just a constant cost for each transition from one grid cell to another, whether it is a diagonal step or not, with shadowed cells removed from consideration. The least-cost A* search would produce the same results with this simple cost function, but the path can be improved by considering the actual path distance between cells, and the cost of turning to change the robot's heading. This would smooth

out the corners more, though determining the required robot heading change will require knowledge of more than just a single step between cells. With an 8-connected grid, each step would indicate heading changes of only multiples of 45 degrees. Since this type of path will most likely only be used to generate a sequence of waypoints spread farther apart, and not step by step motion of the robot, the simpler, less computationally-expensive cost function can be used, with smoothing of corners between waypoints performed locally as the robot drives between them.

Adding in terrain constraints would further change the path shape, with too steep areas placed off-limits. Calculating slopes automatically from elevation data is an area for future research, with more thorough consideration of resolution choices and trade-offs. With this search, however, the situations where shadows have to be avoided are made obvious, resulting in the only deviations from the diagonal path lines. For such areas, the number of waypoints would need to be increased, to guide the robot more carefully around the shadows.

A second type of solar navigation is sun-seeking navigation -- finding a sunlit location for recharging a robot's batteries. In Figure 44, two autonomously-generated paths are shown which take a robot from a shadowed location in the crater to a sunlit point either near the crater wall or the central peak, which will remain sunlit for a three-hour period, or long enough to recharge the robot's batteries. The snapshot image on the right shows the position of shadows, colored blue, at the starting time of the path. The left image shows the intensity-based elevation map of the crater at a smaller scale, which was "placed" on Earth, centered at 80S latitude on January 15, 2000. The range of elevations is from -40 meters to nearly 500 meters high, and the horizontal scale is shown in the figure. The movement of shadows while the robot follows the path is included in the temporal planning algorithm. Traversability of the crater wall is not included in this simulation, but could be added to the model. The resulting path found in such an enhanced model would most likely be different, perhaps longer, and perhaps take longer to find, but could still be generated using this same planning structure.



Elevation map of crater

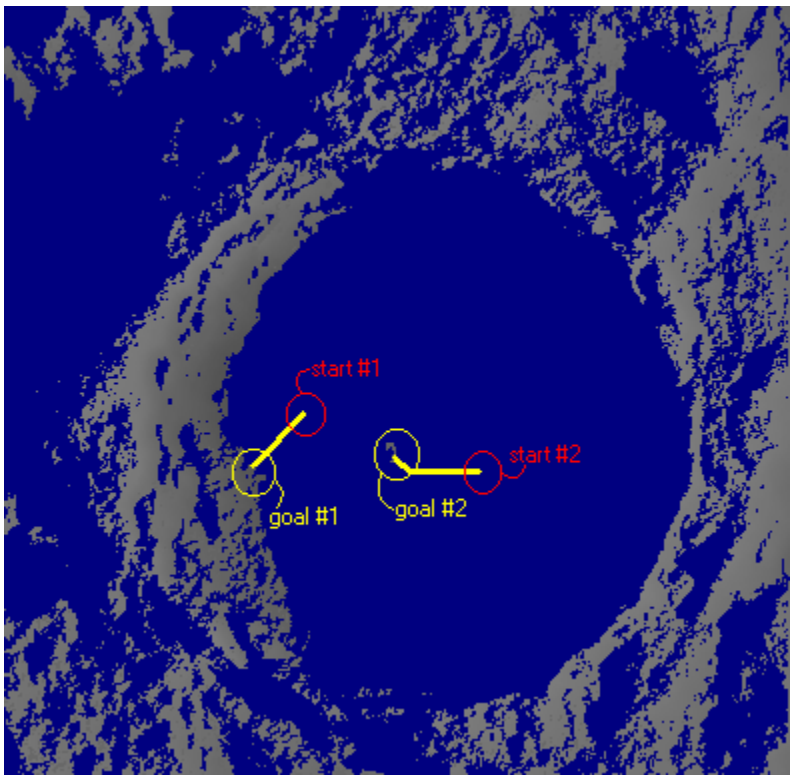


Figure 44: Autonomously-generated path to a recharging location

Chapter VI

Conclusions

This thesis has shown that sun, terrain and temporal information is profitable for a robotic explorer, and has shown multiple ways to use such information. On-board information enables intelligent navigation, including real-time sensing of changing conditions. Off-board information enables finding long-term or long-range solar navigation plans and the generation of databases of information which can be used by either robotic explorers or human users. By the introduction of a new type of robotic navigation, this research opens up a new area of work, which will lead to many future advances. The following items review the results developed for this thesis.

- Sun-synchronous path finding

  Finding a path which circumnavigates a terrain feature while remaining continually in sunlight is a difficult problem, but one whose solution has been shown in this research for the first time. Autonomous path identification is done with a search algorithm considering both the passage of shadows over time and the progression of a robot around a designated feature. It can be performed off-board or on-board, depending on time-constraints and desired terrain and path resolution. This thesis has opened the door to further refinements and research in this new area of solar navigation, pointers to which will be given in the future work section.

- Sun-seeking path finding

  For some exploration missions, a robot will find itself working in a location blocked from the sun and/or the Earth. In these cases, the algorithm developed in this research enables a robot to find the nearest location for battery recharging or Earth communication. Off-board development of sunlight and Earthlight endurance maps provides useful databases for a

robotic explorer, indicating not only which locations are lit by the sun or Earth, but for how much longer they will remain lit. In addition, such databases indicate when no sunlight will be available in a robot's current location for an extended time period, allowing it to shut down non-critical systems and hibernate when necessary.

- Coverage pattern selection

  A variety of coverage patterns has been simulated and evaluated for given locations and times, producing estimates of solar power generation, power consumption, area covered and overlap, total time, and for some locations, wind power generation. The pattern producing the best evaluation results out of a pre-determined set of pattern types can be identified, where pattern ranking can be done with multiple methods. A sun-following coverage pattern has been developed, providing near-optimal solar power generation capabilities for polar regions, while more traditional straight rows and spiral patterns cover the area more efficiently.

- Evaluation heuristics

  Selection of patterns to evaluate is simplified by understanding some basic characteristics of coverage patterns. Evaluation of a variety of pattern types in this research led to heuristics which reduce the needed number of patterns to evaluate on-board the robot. Examples are the double mirroring of straight rows pattern results over a 360 degree range of initial headings, and the creation of analytical evaluation equations based on pattern curvature.

- Terrain shadowing

  By simulating a range of starting locations, headings and times to cover a designated area which is partially shadowed over time, the pattern plan which generates the most solar energy can be identified. From just a limited selection of starting conditions for a single pattern type, a total energy increase of 39% was demonstrated. These time-dependent and terrain-dependent evaluations could conceivably identify the only feasible pattern plan which could cover a highly-shadowed region, constrained by the power requirements of a robot.

- Quantitative improvements

  Simulated mission scenarios, using solar powered robots with limited battery storage, were

evaluated to gain a numerical understanding of the order of magnitude of possible improvements when including solar information in the selection of pattern plans. Robot lifetimes improved by up to 161% for the best pattern as compared to the worst plan in the simulated set. Similarly, robot efficiency improved by over 58%, and robot productivity improved by over 50%, in the best cases.

- Uncertainty

  The effect of pose uncertainty on pattern evaluations was examined, specifically for solar power generation. The resulting variations, for the magnitude of roll and pitch variations expected in most robot exploration work areas, were not highly significant, especially as compared to the differences between pattern types. One benefit of the ability to use solar and terrain information in on-board planning is that consistent terrain slope can be sensed once a robot has arrived at a designated site, and incorporated into the evaluation simulations without having prior high-resolution terrain slope modeling.

- Multiple-parameter evaluation

  Multiple parameters may sometimes conflict, such as when solar power generation and wind power generation are maximized for different pattern orientations. Another possible conflict could be between power generation and power consumption when dealing with highly sloped terrain or high wind resistance. The simulation algorithms developed in this research can identify the best compromise between evaluations.

- Model calibration

  Another benefit of on-board solar information is the ability to use it in calibrating solar panel models. Field work results showed that a simple calibration using the comparison of two simulated data points to the corresponding actual data points can be used to calibrate the power generation model, producing sufficiently accurate results for robot navigation. This procedure adapts the solar panel model to changing lighting conditions, as well as surface reflection and other atmospheric effects, as long as these effects remain constant for the duration of the navigational tasks performed.

A. Contributions

Several key points of this research have contributed to the advancement of robotic exploration and outdoor, remote navigation:

- *New robotic navigational abilities.* New robotic capabilities of finding and following sun-seeking paths and sun-synchronous paths are enabled by this work. A solar ephemeris and terrain algorithm generates shadow map databases, including the temporal dimension, for autonomous path searches or for human examination. Sun-synchronous polar paths allow robots to keep up with the sun, empowering uninterrupted activity and extending useful lifetimes. Sun-seeking paths allow robots to find for themselves vital sunlit locations when power sources are low, or to locate regions within sight of Earth when communication is lost.

- *Environmental, time-dependent modeling of solar power generation.* Sun, terrain and temporal effects are modeled and evaluated during navigational tasks. Solar power generation becomes a new constraint to evaluate, in addition to the more common consideration of power consumption. Additional or more precise environmental and robot configuration models can easily be added to the current structure. The difficulty of integrating temporally changing aspects such as terrain shadowing with complex paths such as coverage patterns is addressed by step-by-step simulation of plans. The presence of this information on-board a robot allows it to adapt to new conditions, such as recalibrating solar panel models based on atmospheric effects, and incorporating previously unknown terrain slopes into task simulations.

- *Solar ephemeris algorithm applied to terrain elevation maps and robotic navigation.* This algorithm accurately determines the location of the sun and Earth from any Earth or lunar surface location, combining the information with a terrain map to display the shadows which would exist for a given location, date and time. This analytical planning tool is used in real-time predictive evaluations of navigational tasks to be performed by a robotic explorer. It can also be compared to observational data to confirm assumed terrain elevations.

- *Detailed analysis of coverage patterns.* Basic characteristics and properties of a variety of new coverage patterns were identified, producing some analytical descriptions. Analysis of

these characteristics reduces the number of potential patterns for evaluation, generating a smaller set which still spans the range of possibilities.

B. Future Work

This research work can be expanded in the future in multiple areas. These areas include environmental and robot modeling, autonomous selection of plan starting locations and times, and improvements and additions to path search algorithms. Research in these areas can build on the work described here, improving the results and increasing the benefits for robotic explorers. The application of this work to additional classes of robotic explorers can be studied as well, as many similar tasks could utilize these results. One example is agricultural or lawn-care robots attempting to perform coverage of fields in optimal ways, and whose power systems could be augmented by solar panels. Underwater robots may not utilize solar power, but the characteristics of coverage patterns described in this thesis could be applied to oceanic mapping and discovery missions. Any environment with predictable time-dependent and/or location-linked characteristics can be modeled and described in the planning structure developed here, allowing tasks to be simulated one step at a time through the changing conditions, producing ranked evaluations of a robot's potential tasks. More discussion of specific research topics for future work are detailed below.

Environmental and robot modeling is one area for additional research. Power consumption evaluation was not analytically developed in this thesis. The structure for evaluating a robot's power needs step by step through a complete task simulation is complete, requiring only an accurate model of the robot's response to commands, in conjunction with the current conditions. Statistical estimates of Nomad's reaction to commands and to the current pitch were used in this thesis, but more accurate modeling of a robot's reactions can be beneficial. Traversability information can also be incorporated, such as higher power consumption needs when driving through loose gravel as opposed to hard-packed ground. Such information could be added to a robot's internal map, identifying terrain types which can be linked to specific power consumption needs. Related work in this area has been done, so one option is linking such work with this thesis research. Another beneficial environmental model is one for wind resistance. Based on a robot's

profile, heading, and wind speed, resistance factors can be added into the power consumption model. This work would be especially useful for a robot with poor streamlining in one or more directions, such as a robot with a wind turbine or with a protruding solar panel.

An understanding of lighting conditions can also be used by image processing algorithms. Several of the experimentally-set parameters in the meteorite search color image segmentation algorithm required alterations from day to day, such as the cutoff threshold which distinguished between dark and light regions. These alterations were required due to variations in the lighting conditions, as tests showed that the cloudiness of the sky greatly affected the color values of snow and ice. Information on the current sky conditions can be obtained by monitoring solar panel values, and comparing the actual data to modeled results, as described earlier. Knowledge of the current time and location can also be used to calculate the expected position of the sun, to determine expected shadowing caused by the robot itself. Such information could be used to adaptively change and fine-tune the segmentation parameters to obtain more accurate results. Solar information could also potentially be used in determining the appropriate maximum sensor range to successfully identify target objects.

As mentioned earlier, the solar model developed in this research calculates the position of the sun and Earth from surface locations on the Earth or moon. A solar ephemeris can be developed for other planets, such as Mars and Mercury, to allow simulations to be performed for robots exploring those planets. Any locations in the inner solar system receive enough solar flux for solar power to be a viable power source. Most inner solar system bodies of any size rotate slowly enough to have extended sunlit regions in which robots can explore and benefit from methodical planning. Some asteroids are an exception, as their rotation and orbital parameters may be rapid and complicated enough to cause modeling difficulties. Mars, however, is the most promising candidate for immediate future work, as robotic missions are already being sent to that planet.

Another area for future research is the autonomous selection of plan starting points and times. A range of starting points and times have been used in the simulations for this thesis, with heuristics and approximations being used to limit the number of choices. Steps to limit the possibilities for coverage patterns in shadowed regions have also been made. However, more precise and autonomous methods could be found. Identifying larger regions which are sunlit for a

discrete time interval is a likely approach, and can be matched to the type of pattern desired. For instance, for straight rows patterns, strips of pixels can be examined, while for spiral patterns, concentric rings are better. The time period for which the strips should be tested for sunlight can be related to the time required to search the strip size. Reducing a search area to such regions can assist in identifying on which side of a region to start a pattern, and at what time of day. Alternately, using image processing techniques, sunlit regions can be "grown," and fitted to similar pattern shapes. A third approach is to consider the three-dimensional space generated by a database containing lighting information related to location and time. With appropriate discretization of time and position -- for example, such that pattern row lengths consist of a single block of space -- three-dimensional planes could be identified with a slope related to pattern traversal time and orientation. Planar regions more continually lit by the sun can be selected.

Another higher level of planning can be developed which identifies desirable regions to be searched, out of the entire range of the robot. Such regions could be identified based on the likelihood of containing valuable or new information relevant to the robot's mission. A selection of regions to search can also be prioritized based on power considerations or other constraints and benefits. Other researchers are currently working on some of these concepts.

One final area for future research is the expansion of path search algorithms. The search algorithms developed in this thesis consider primarily the existence of sunlight. Traversability also needs to be incorporated, such as the ability of a robot to traverse creeks or to climb steep slopes. This requires identifying and encoding impassable regions in the robot's internal map. This could be done either ahead of time based on elevation maps or other prior information, or by enabling a robot to recognize dangerous terrain and replanning the path from that point. Search improvements include allowing temporary pauses and backtracking, changes of a robot's speed, and short traverses through shadowed regions based on a robot's backup battery power. For robotic missions which expect to need sun-seeking searches, an additional ability needs to be developed for a robot to maintain a "lock" on the nearest sunlit location, in conjunction with the robot's remaining battery power, such that the robot can be assured that a sunlit location will be reachable when needed, and can alert the planner when it needs to start toward that location.

Chapter VII

Appendices

A. Navigational Planning Library

A navigational planning library was developed for this thesis, including a grid-based map structure with supporting functions, multiple coverage pattern generators, a waypoint planner, and a maneuver planner. The code was written in C, and primarily for the Linux platform, though some of the code was also used on a Unix platform. Details of the functions, algorithms and structures are given below.

GRID-BASED MAP

The grid-based map structure is used by all other components of the library. It is designed to not take up much more memory than is needed, by starting out with a central block of grid cells around the robot, and only adding additional blocks as they are needed. The grid cell resolution is set at runtime, by whichever programs are using the library, and is typically 0.5 meters per cell. The block size can also be easily altered, but is currently hard-coded into the library as 50 cells by 50 cells. The map structure includes the address of the central block, the size of the cells, and a few fields describing terrain elevation map parameters. The block structure includes the coordinates of the bottom left corner, a two-dimensional array of the cells contained in the block, and an array for the addresses of the 8-connected adjacent blocks. The grid cell structure includes the row and column of the cell within the block, the elevation of the terrain at that location, the address of the parent block, 2 characters which can be set bit by bit to describe the contents and

state of the cell (e.g. sunlit, shadowed, dangerous, covered by robot, etc.), and several other fields related to when the cell was last accessed.

When a map is initialized, the central point of the region -- such as the location of a DGPS base station -- is set as the center of the central map block, and the grid cell and neighboring block arrays are initialized. All coordinates in the map are described relative to this center point. The actual latitude and longitude of this center point are also needed for the sun position calculations, but are not referred to in the map structure itself. As the robot or a simulation progresses past the bounds of the central block, additional blocks are added one at a time, only as they are needed, and are linked to the map in an 8-connected manner.

After the map is initialized, an option is to read in a digital elevation map (DEM) containing elevation information. The center of the DEM is assumed to equal the center of the initialized map. The DEM must be in the ppm/pgm ascii image format, which contains an entry for each row and column of the image area, encoding the elevation as a single integer for black and white pgm images, or as the red integer value for 3-color ppm images. Other required information about the DEM includes the location (Earth or moon), the vertical elevation scaling (integer unit/ meter), the minimum elevation of the map (meters), the horizontal grid cell scaling (meters/cell), the projection type (flat world or stereographic), and the angular offset of the y-axis from north (radians, usually 0.0). As the DEM is read in, map blocks are created and linked as needed.

A variety of functions are available for those programs using the grid-based map structure:

init_grid_block initializes a new block structure and all its interior cells, sets the adjacent blocks to NULL, and returns a pointer to the block's address

init_map initializes a new map structure and its central block

linkAdjBlock finds the neighboring block in a given direction, if it exists, and ensures appropriate links are in place

add_grid_block adds and links a new block to a given block, in a given direction

GPS_grid_block adds a new block that contains the desired coordinates relative to the map center, filling in all needed blocks along the way

save_map saves the map to an image file for later viewing, marking things such as shadows and coverage

find_pixel_addr finds the address of the grid cell structure for the given coordinates, returning NULL if that location has not yet been added to the map

find_pixel_GPS returns the coordinates for a given grid cell address

find_adj_pixels returns an array of grid cell addresses for all cells adjacent (8-connected) to the given cell, with an option to either add new blocks if needed, or to set those not-yet-mapped cells to NULL

free_map frees the map memory

mark_obstacle marks a given grid cell as an obstacle, setting the appropriate bit in a cell field

mark_map marks all the grid cells which can be seen from the given current robot position as "covered", based on the field of view of the robot's sensor used for coverage

COVERAGE PATTERN PARAMETERS

The coverage pattern generators build on the grid-based map, using the grid cell structure to identify the current robot position, desired path points, and goal points. Four types of coverage patterns have been implemented to date: straight rows, spiral, sun-following and curved. Each pattern type is composed of a combination of path segments, either straight lines, circle arcs of variable radius, or point turns. Straight rows, for instance, are made up of straight lines with either points turns or half-circle arcs with a diameter equal to the row width for the ends. Instead of continually increasing the radius of a spiral, the pattern is simplified by using half circle path segments, with each segment increasing in radius from the one before just enough to produce the desired width between the concentric rows. This spiral is not technically a perfect spiral, but compensates for the limitations in the a robot's physical ability to pursue specific turning radii, which is reflected by the navigational software only allowing discrete steering angles. Curved patterns maintain the same curvature for both "upward" and "downward" rows, but opposite rows in the sun-following pattern have opposite curvatures, such that the appropriate solar panel will be normal to the sun's azimuth. The initial robot heading for this pattern is also based on the starting sun azimuth, with a point turn sometimes required to reach that heading.

The patterns are not generated in their entirety at the start, but are instead initialized by defining only the first two path segments to follow. As the robot progresses, when one path segment is finished, the next segment is started, and only then is the following segment calculated. In this way, only a minimum of information needs to be stored. The parameters which define the patterns provide all the information needed to generate new path segments one by one. A generalized pattern plan structure defines these parameters, as well as additional information used in reporting the status back to the user, and contains the following fields:

- a path segment structure for the current path segment (described below)
- coordinates for the current robot position, closest path point, and plan starting point
- the current row number, and the total number of rows in the pattern
- a flag indicating if point turns are to be used
- a flag indicating the direction of the straight rows/curved patterns (left or right), or the current half of the spiral pattern (top or bottom)
- a structure defining the vertices of the convex polygonal area to be covered, for the straight rows pattern
- the coordinates of the spiral's top and bottom half centers
- a general parameter indicating the primary heading of the straight rows pattern, the maximum radius of the spiral pattern, or the curvature of the curved pattern
- the length of pattern rows
- the width between each row
- the minimum turning radius allowed by the robot
- the minimum turning radius allowed by the obstacle avoidance sensor
- the maximum robot speed allowed
- several fields for waypoint and maneuver plans

The path segment structure contains these fields:

- the curvature of the segment
- the coordinates of the starting and end points
- the coordinates of the center of a circular path segment
- the coordinates of the current pure-pursuit goal waypoint

- a flag indicating direction (clockwise or counterclockwise)
- the lookahead distance to use for this segment
- a flag indicating the type of path segment
- a point to the next path segment structure

Addition pattern types can be added to the library fairly easily, as long as they are composed of the same types of path segments. For instance, a pattern like the one shown in Figure A1 could be developed. However, a new, somewhat complicated function would need to be defined for the calculation of successive path segments based solely on the pattern information contained in the pattern plan structure, indicating, for instance, what distance and direction the next row should be from the current row, and when it is time to move on to the next group of rows.

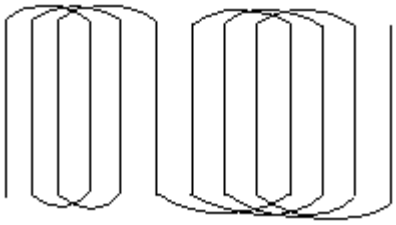


Figure A1: Alternating intertwined pattern

Separate functions are used for the initialization and path segment calculations for each pattern type. These functions can be called by the Mission Planner when a pattern is to be simulated, or by the Navigation Manager when a pattern is to be enacted. A general path segment following function is used for all patterns, taking only the path segment structure and determining from those values where the robot should drive. Generally, a desired robot speed and turning radius are determined, which can be sent to Arbiter. Point turns are not allowed by Arbiter, however, so those commands are sent directly to the Realtime System.

WAYPOINT AND MANEUVER PLANNING

Waypoint planning is approached simply in this implementation. The goal point is converted from global coordinates to relative-to-robot coordinates, and then the proper steering angle to head toward that goal is calculated once a second. Obstacles are not included in planning, but are

assumed to be detected and avoided by the obstacle avoidance module, which will veto any of the waypoint planner's steering angle choices which would hit an obstacle. The goal is considered reached when the robot comes within a given, user-changeable, distance from that goal.

Maneuver plans are processed slightly differently than coverage patterns and waypoints, requiring one extra procedure to first generate a plan. Given the start and goal positions of the robot, including orientation, an A* search is performed through the space of possible robot positions and orientations. The result is a sequence of turning radii commands needed for each one-second step along the way. An alternate type of goal can be selected where the robot does not need to be at any particular location, but only pointed toward the goal at a certain distance away. This case is used when the goal is a target object, and the robot only needs to be within a sensor's workspace of the target. The calculated sequence of steps to reach the goal state is then saved in a text file, which can then can be read back by either a simulator or the Navigation Manager. The turning radii are processed one by one, every second, without any feedback concerning the robot's position. This method was implemented in order to most easily interface with the rest of the already developed system. Due to the lack of responsiveness, only short maneuvers are usually performed, and the robot's position is checked at the end of the entire maneuver. At this point, if the robot is not where expected, another maneuver can be planned.

The maneuver planner uses a limited model of the robot's response to a set of commands: backing up, driving straight, turning left with a 4 meter turning radius, turning right with a 4 meter turning radius, and stopping. The previous state of the robot is also incorporated into the model. Each response is defined for a command lasting one second. This rough model has turned out, in practice, to be sufficient for planning short maneuvers, and the limited set of commands greatly improves the planning speed. The planner first uses a heuristic to determine if the robot will need to back up initially, since this saves time in the ensuing A* search. After any required backing up, the search applies the model to the robot's current (simulated) state and generates five new potential states for the robot, as if each of the modelled actions were performed. Those states are all put into a queue. Assuming none of those actions puts the robot into a goal state, the search takes the queue item which is closest to the goal and applies the model again to that state.

Identical states are never re-added, and the progression of states is recorded so that, at the end of the search, the path followed can be backtracked and saved as the maneuver plan.

After a search is completed, additional fine-tuning steps are found for sensor workspace maneuvering. The A* search just finds the first possible goal state, but more steps are needed in order to move the robot into the middle of the workspace and not just barely on the edge. An additional five steps using the model are processed, and the sequence of steps which moves the robot closest to the middle of the workspace is added to the end of the previously generated plan. When the arm sensor is to be used after the enactment of a maneuver, a command is sent by the Navigation Manager directly to the Realtime system to straighten the robot's wheels, as in their turned configuration, they can interfere with the deployment of the arm.

B. Solar Ephemeris Library

The solar ephemeris library calculates the altitude and azimuth of the sun, and optionally the Earth, as seen from a surface location on the Earth or the moon. To generate these values, a series of geometrical coordinate transformations are applied to initial values giving the right ascension and declination of the sun and moon from the center of the Earth. These initial values are obtained, given a Julian day and time, from commonly available astronomical algorithms, such as found in [62] and [5], with a more than adequate accuracy for surface navigational planning. The coordinate transformations then applied are detailed below.

EARTH LOCATION TRANSFORMATIONS

1. Start with the geocentric, equatorial-plane angular coordinates and distance of the sun:

   sun_{RA}, sun_{Dec}, sun_{Dist}

2. Given the time (Jdate = Julian Day, fractional days since the year -4712, not related to Julian calendar) and surface longitude (surf_{lon}), find the longitudinal rotation angle (H) needed to rotate coordinates to align with robot's surface position (time is needed because the sun's right ascension, sun_{RA}, has a time-based longitudinal reference).

- First, find the sidereal time (GMST) in degrees:

$$T = \frac{\text{Jdate} - 2451545.0}{36525.0}$$

$$GMST = 280.46061837 + 360.98565 \times (\text{Jdate} - 2451545.0) + 0.00038793T^2 - T^3/38710000.0$$

$$GMST = GMST - (\text{int})\left(\frac{GMST}{360}\right) \times 360$$

- Convert from degrees to radians ($GMST = GMST \times \pi \div 180$), and find hour angle H (note that surface longitude is positive to the west):

$$H = GMST - \text{surf}_{lon} - \text{sun}_{RA}$$

3. Using the longitudinal rotation angle H and the surface latitude (surf_{lat}), rotate the original coordinates to align with the robot's surface position and project the sun vector onto the surface plane. The *azimuth* is the angle that the sun vector projection makes with respect to the longitude line pointing north (see Figure A2):

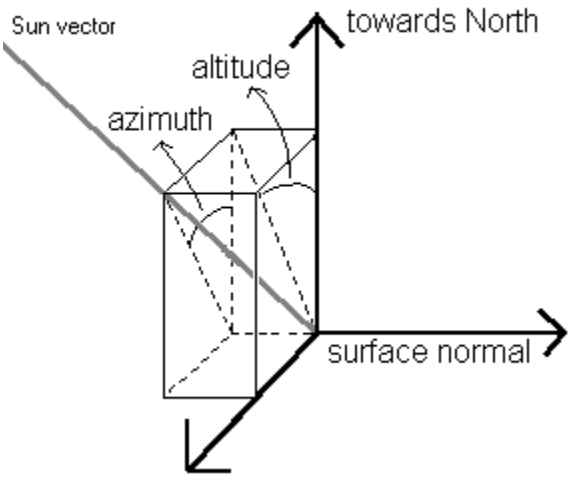


Figure A2: Angular calculations

$$\text{azimuth} = \text{atan}2(-\sin H \cos(\text{sun}_{Dec}), \cos(\text{surf}_{lat})\sin(\text{sun}_{Dec}) - \sin(\text{surf}_{lat})\cos(\text{sun}_{Dec})\cos H)$$

- This azimuth is positive in the clockwise direction, but the robot uses a different coordinate system where positive angles are in the counterclockwise direction, so azimuth = –azimuth.

4. Projecting the sun vector onto the north longitude-surface normal plane, the *altitude* is the angle that projection makes with respect to the longitude line pointing north:

$$\text{altitude} = \text{asin}(\sin(\text{surf}_{lat})\sin(\text{sun}_{Dec}) + \cos(\text{surf}_{lat})\cos(\text{sun}_{Dec})\cos H)$$

MOON LOCATION TRANSFORMATIONS

1. Start with the geocentric, equatorial-plane angular coordinates and distances of the sun and moon from the Earth

$$\text{sun}_{RA},\ \text{sun}_{Dec},\ \text{ESun}_{Dist},\ \text{moon}_{RA},\ \text{moon}_{Dec},\ \text{EMoon}_{Dist}$$

2. Rotate coordinates to geocentric, ecliptical-plane angular coordinates (the ecliptic is the plane of the Earth's orbit about the sun).
   - The difference in tilt between the equatorial and ecliptic planes, EEQ_ECL, changes gradually over time. At $\text{EE}_{Date} = 2449717.5$, EEQ_ECL = 0.40910416 radians, while it changes at a rate of $\text{EE}_{Change} = 0.1\times10^{-7}$ radians per day:

$$\text{curr}_{EEQ} = \text{EEQ\_ECL} - \text{EE}_{Change} \times (\text{Jdate} - \text{EE}_{Date})$$

```
/*************************************************************/
Spherical function:
  Converts one set of spherical coordinates into another set of spherical coordinates,
  based on the angle between their two x-y planes.
inputs = ε, Δ, RA
outputs = lat, lon
ε is the angle between the two planes at the ascending node of plane 2 on plane 1.
Δ and RA are the latitude and longitude of plane 1 (the lower plane) in radians.
lat and lon are the latitude and longitude of plane 2 in radians (the unknowns).
```

$$a = \sin(\Delta)\cos(\varepsilon) - \cos(\Delta)\sin(\varepsilon)\sin(RA)$$
$$b = \cos(RA)\cos(\Delta)$$
$$c = \sin(\Delta)\sin(\varepsilon) + \cos(\Delta)\cos(\varepsilon)\sin(RA)$$

$$\text{if}\left(\Delta = \frac{\pi}{2} \,\|\, \Delta = -\frac{\pi}{2}\right) \quad \text{lon} = \frac{\pi}{2} \quad \text{(polar points have indeterminate longitude)}$$

$$\text{else} \qquad \text{lon} = \operatorname{atan2}(c, b)$$

$$\text{lat} = \operatorname{asin}(a)$$

/***/

$$(\text{sunc}_{lat}, \lambda_0) = \text{Spherical}(\text{curr}_{EEQ}, \text{sun}_{Dec}, \text{sun}_{RA}) \quad \text{(coordinates of the sun)}$$

$$(\beta, \lambda) = \text{Spherical}(\text{curr}_{EEQ}, \text{moon}_{Dec}, \text{moon}_{RA}) \quad \text{(coordinates of the moon)}$$

3. Move origin of coordinate system to center of sun (heliocentric, ecliptical coordinates)
   - Coordinates of the moon, using equations from [62], p. 376.

$$\lambda_H = \lambda_0 + \pi + \frac{\text{EMoon}_{Dist}}{\text{ESun}_{Dist}} \times \cos\beta \times \sin(\lambda_0 - \lambda)$$

$$\beta_H = \frac{\text{EMoon}_{Dist}}{\text{ESun}_{Dist}} \times \beta$$

4. Rotate coordinates to selenographic angular coordinates (moon-equatorial coordinates centered at the center of the moon). This requires some sun and Earth libration calculations as seen from the moon, described in [62], in order to determine the lunar coordinate system's longitudinal and latitudinal reference points.

$$(\text{Earth}_{lon}, \text{Earth}_{lat}, \text{Sun}_{lon}, \text{Sun}_{lat}) = \text{findLibrations}(\text{Jdate}, \lambda, \beta, \lambda_H, \beta_H)$$

/***/

findLibrations:

$\text{inputs} = \text{Jdate}, \lambda, \beta, \lambda_H, \beta_H$

$\text{outputs} = \text{Earth}_{lon}, \text{Earth}_{lat}, \text{Sun}_{lon}, \text{Sun}_{lat}$

First, approximate $\Delta\Phi$ to accuracy of 0.5 seconds of arc:

$$T = \frac{\text{Jdate} - 2451545.0}{36525.0}$$

$$\Omega = \frac{125.04452 - 1934.136261T}{180\pi}$$

$$L = \frac{280.4665 + 36000.7698T}{180\pi}$$

$$L' = \frac{218.3165 + 481267.8813T}{180\pi}$$

$$\Delta\Phi = \frac{-17.20 \times \sin(\Omega) - 1.32 \times \sin(2.0L) - 0.23 \times \sin(2.0L') + 0.21 \times \sin(2.0\Omega)}{648000\pi}$$

Second, find a more accurate Ω, and the variables I and F:

$$\Omega = \frac{125.0445479 - 1934.1362891T + 0.0020754T^2 + \frac{T^3}{467441.0} - \frac{T^4}{60616000.0}}{180\pi}$$

$$I = 0.026920$$

$$F = \frac{93.2720950 + 483202.0175233T - 0.0036539T^2 - \frac{T^3}{3526000.0} + \frac{T^4}{863310000.0}}{180\pi}$$

Now, use Equation 53.1, on page 372 of [62], for Earth:

$$W = \lambda - \Delta\Phi - \Omega$$

$$A = \operatorname{atan2}(\sin W \times \cos\beta \times \cos I - \sin\beta \times \sin I, \cos W \times \cos\beta)$$

$$l' = A - F$$

$$b' = \operatorname{asin}(-\sin W \times \cos\beta \times \sin I - \sin\beta \times \cos I)$$

And for the sun:

$$W = \lambda_H - \Delta\Phi - \Omega$$

$$A = \operatorname{atan2}(\sin W \times \cos\beta_H \times \cos I - \sin\beta_H \times \sin I, \cos W \times \cos\beta_H)$$

$$l_0' = A - F$$

$$b_0' = \operatorname{asin}(-\sin W \times \cos\beta_H \times \sin I - \sin\beta_H \times \cos I)$$

Find the values D, M, M', E, K_1, K_2:

$$D = \frac{297.8501921 + 445267.1114034T - 0.0018819T^2 + \frac{T^3}{545868.0} - \frac{T^4}{113065000.0}}{180\pi}$$

$$M = \frac{357.5291092 + 35999.0502909T - 0.0001536T^2 + \frac{T^3}{24490000.0}}{180\pi}$$

$$M' = \frac{134.9633964 + 477198.8675055T + 0.0087414T^2 + \frac{T^3}{69699.0} - \frac{T^4}{14712000.0}}{180\pi}$$

$$E = 1.0 - 0.002516T - 0.0000074T^2$$

$$K_1 = \frac{119.75 + 131.849T}{180\pi}$$

$$K_2 = \frac{72.56 + 20.186T}{180\pi}$$

From page 373 of [62]:

$$\begin{aligned}\rho = {} & -0.02752 \times \cos M' - 0.02245 \times \sin F + \\ & 0.00684 \times \cos(M' - 2.0F) - 0.00293 \times \cos(2.0F) - \\ & 0.00085 \times \cos(2.0F - 2.0D) - 0.00054 \times \cos(M' - 2.0D) - \\ & 0.00020 \times \sin(M' + F) - 0.00020 \times \cos(M' + 2.0F) - \\ & 0.00020 \times \cos(M' - F) + 0.00014 \times \cos(M' + 2.0F - 2.0D)\end{aligned}$$

$$\begin{aligned}\sigma = {} & -0.02816 \times \sin M' + 0.02244 \times \cos F - \\ & 0.00682 \times \sin(M' - 2.0F) - 0.00279 \times \sin(2.0F) - \\ & 0.00083 \times \sin(2.0F - 2.0D) + 0.00069 \times \sin(M' - 2.0D) + \\ & 0.00040 \times \cos(M' + F) - 0.00025 \times \sin(2.0M') - \\ & 0.00023 \times \sin(M' + 2.0F) + 0.00020 \times \cos(M' - F) + \\ & 0.00019 \times \sin(M' - F) + 0.00013 \times \sin(M' + 2.0F - 2.0D) - \\ & 0.00010 \times \cos(M' - 3.0F)\end{aligned}$$

$$\begin{aligned}\tau = {} & 0.02520E \times \sin M + 0.00473 \times \sin(2.0M' - 2.0F) - \\ & 0.00467 \times \sin M' + 0.00396 \times \sin K_1 + \\ & 0.00276 \times \sin(2.0M' - 2.0D) + 0.00196 \times \sin\Omega - \\ & 0.00183 \times \cos(M' - F) + 0.00115 \times \sin(M' - 2.0D) - \\ & 0.00096 \times \sin(M' - D) + 0.00046 \times \sin(2.0F - 2.0D) - \\ & 0.00039 \times \sin(M' - F) - 0.00032 \times \sin(M' - M - D) + \\ & 0.00027 \times \sin(2.0M' - M - 2.0D) + 0.00023 \times \sin K_2 - \\ & 0.00014 \times \sin(2.0D) + 0.00014 \times \cos(2.0M' - 2.0F) - \\ & 0.00012 \times \sin(M' - 2.0F) - 0.00012 \times \sin(2.0M') + \\ & 0.00011 \times \sin(2.0M' - 2.0M - 2.0D)\end{aligned}$$

For the Earth:

$$l'' = -\tau + (\rho \times \cos A + \sigma \times \sin A) \times \tan(b')$$

$$b'' = \sigma \times \cos A - \rho \times \sin A$$

For the sun:

$$l_0'' = -\tau + (\rho \times \cos A + \sigma \times \sin A) \times \tan(b_0')$$

$$b_0'' = \sigma \times \cos A - \rho \times \sin A$$

Now, convert from degrees to radians, and find longitude and latitude:

$$l'' = \frac{l''}{180\pi} \qquad b'' = \frac{b''}{180\pi}$$

$$l_0'' = \frac{l_0''}{180\pi} \qquad b_0'' = \frac{b_0''}{180\pi}$$

$$\text{Earth}_{lon} = l' + l'' \qquad \text{Earth}_{lat} = b' + b''$$

$$\text{Sun}_{lon} = l_0' + l_0'' \qquad \text{Sun}_{lat} = b_0' + b_0''$$

/**/

- Selenographic longitude is positive to the Moon's east, so we need to change the sign to switch to our standard where west longitude is positive. Also, make sure the longitudes are between 0 and 2π.

$$\text{Sun}_{lon} = -\text{Sun}_{lon}$$

5. At this point, coordinates could be rotated to align with robot's surface position as with Earth calculations, but due to prior work and for simplification, coordinates are first transformed to Cartesian coordinates.

/**/

Cartesian:

Converts spherical coordinates into cartesian coordinates.

inputs = lat, lon, Dist

outputs = $\overrightarrow{\text{vector}}$

$$\overrightarrow{\text{vector}}_x = \text{Dist} \times \cos(\text{lat})\cos(\text{lon})$$

$$\overrightarrow{\text{vector}}_y = \text{Dist} \times \cos(\text{lat})\sin(\text{lon})$$

$$\overrightarrow{\text{vector}}_z = \text{Dist} \times \sin(\text{lat})$$

/**/

- First, find distances from moon to sun and from moon to Earth:

$\overrightarrow{\text{EarthSun}} = \text{Cartesian}(\text{sun}_{Dec}, \text{sun}_{RA}, \text{ESun}_{Dist})$

$\overrightarrow{\text{EarthMoon}} = \text{Cartesian}(\text{moon}_{Dec}, \text{moon}_{RA}, \text{EMoon}_{Dist})$

$\text{Moon}_{Dist} = \left|\overrightarrow{\text{EarthMoon}}\right| \quad (= \text{EMoon}_{Dist})$

$\text{Sun}_{Dist} = \left|\overrightarrow{\text{EarthSun}} - \overrightarrow{\text{EarthMoon}}\right|$

- Now find vectors from moon center to sun, and from moon center to Earth:

$\overrightarrow{\text{Sun}} = \text{Cartesian}(\text{Sun}_{lat}, \text{Sun}_{lon}, \text{Sun}_{Dist})$

$\overrightarrow{\text{Earth}} = \text{Cartesian}(\text{Earth}_{lat}, \text{Earth}_{lon}, \text{Moon}_{Dist})$

6. Find Cartesian coordinates of vector from moon's center to robot's surface position (surf_{lat}, surf_{lon}), and from moon's center to moon's north pole.

- Vector from moon center to moon surface:

$\overrightarrow{\text{Moon}} = \text{Cartesian}(\text{surf}_{lat}, \text{surf}_{lon}, \text{Moon}_{\text{radius}})$

- Vector from moon's center to moon's North pole:

$\overrightarrow{\text{Pole}} = (0.0, 0.0, \text{Moon}_{\text{radius}})$

7. Move origin of coordinate system to robot's surface position, maintaining vectors from robot to sun, robot to Earth, and robot to lunar north pole.

$\overrightarrow{\text{MoonSun}} = \overrightarrow{\text{Sun}} - \overrightarrow{\text{Moon}}$

$\overrightarrow{\text{MoonEarth}} = \overrightarrow{\text{Earth}} - \overrightarrow{\text{Moon}}$

$\overrightarrow{\text{MoonPole}} = \overrightarrow{\text{Pole}} - \overrightarrow{\text{Moon}}$

8. Rotate coordinates to a plane tangent to surface, with one axis toward the lunar north pole.

   - Project $\overrightarrow{\text{MoonSun}}$ and $\overrightarrow{\text{MoonEarth}}$ onto plane tangent to moon at surface point (normal vector to that plane is just $\overrightarrow{\text{Moon}}$):

$$\overrightarrow{\text{temp}} = \frac{\overrightarrow{\text{MoonSun}} \bullet \overrightarrow{\text{Moon}}}{\overrightarrow{\text{Moon}} \bullet \overrightarrow{\text{Moon}}} \times \overrightarrow{\text{Moon}}$$

$$\overrightarrow{\text{MoonSunProject}} = \overrightarrow{\text{MoonSun}} - \overrightarrow{\text{temp}}$$

$$\overrightarrow{\text{temp}} = \frac{\overrightarrow{\text{MoonEarth}} \bullet \overrightarrow{\text{Moon}}}{\overrightarrow{\text{Moon}} \bullet \overrightarrow{\text{Moon}}} \times \overrightarrow{\text{Moon}}$$

$$\overrightarrow{\text{MoonEarthProject}} = \overrightarrow{\text{MoonEarth}} - \overrightarrow{\text{temp}}$$

   - Project $\overrightarrow{\text{MoonPole}}$ onto surface plane:

$$\overrightarrow{\text{temp}} = \frac{\overrightarrow{\text{MoonPole}} \bullet \overrightarrow{\text{Moon}}}{\overrightarrow{\text{Moon}} \bullet \overrightarrow{\text{Moon}}} \times \overrightarrow{\text{Moon}}$$

$$\overrightarrow{\text{MoonPoleProject}} = \overrightarrow{\text{MoonPole}} - \overrightarrow{\text{temp}}$$

   - But beware of surface point being a polar point:

   if $(\left|\overrightarrow{\text{MoonPoleProject}}\right| < 0.0001)$, surface location is very close to a pole:

   if $(\left|\overrightarrow{\text{MoonPole}}_z - 2.0\text{Moon}_{\text{radius}}\right| < 0.01)$, we're at the south pole, or

   if $(\overrightarrow{\text{MoonPole}}_z < 0.01)$, we're at the north pole. In either case:

$$\overrightarrow{\text{MoonPoleProject}} = (1.0, 0.0, 0.0)$$

9. Projecting the sun and Earth vectors in a similar manner as for Earth locations (see Figure A2), find the *azimuth* and *altitude.*

   - $\pi/2$ - the angle between the $\overrightarrow{\text{Moon}}$ vector and the $\overrightarrow{\text{MoonSun}}$ / $\overrightarrow{\text{MoonEarth}}$ vectors is the elevation of the sun / Earth:

$$\text{altitude} = \frac{\pi}{2} - \text{acos}\left(\frac{\overrightarrow{\text{Moon}} \bullet \overrightarrow{\text{MoonSun}}}{\left|\overrightarrow{\text{Moon}}\right| \times \left|\overrightarrow{\text{MoonSun}}\right|}\right)$$

$$\text{earthElevation} = \frac{\pi}{2} - \text{acos}\left(\frac{\overrightarrow{\text{Moon}} \bullet \overrightarrow{\text{MoonEarth}}}{\left|\overrightarrow{\text{Moon}}\right| \times \left|\overrightarrow{\text{MoonEarth}}\right|}\right)$$

- Angle between $\overrightarrow{\text{MoonPoleProject}}$ and $\overrightarrow{\text{MoonSunProject}}$ is the azimuth of the sun:

$$\text{value} = \frac{\overrightarrow{\text{MoonPoleProject}} \bullet \overrightarrow{\text{MoonSunProject}}}{\left|\overrightarrow{\text{MoonPoleProject}}\right| \times \left|\overrightarrow{\text{MoonSunProject}}\right|}$$

$$\text{azimuth} = \text{acos}(\text{value})$$

Due to computational effects (e.g. floating point errors), there's a chance that value will be out of the input range for the acos function. So, if $(\text{value} > 1.0)$, set $\text{azimuth} = 0.0$, and if $(\text{value} < -1.0)$, set $\text{azimuth} = \pi$.

- Similarly for Earth:

$$\text{Earth}_{\text{azimuth}} = \text{acos}\left(\frac{\overrightarrow{\text{MoonPoleProject}} \bullet \overrightarrow{\text{MoonEarthProject}}}{\left|\overrightarrow{\text{MoonPoleProject}}\right| \times \left|\overrightarrow{\text{MoonEarthProject}}\right|}\right)$$

10. But the azimuth angles don't ever get above 180 degrees or below 0, so we need to fix that (note that this part could be simplified by not using Cartesian coordinates to start with).
    - Rotate vectors so that $\overrightarrow{\text{MoonPoleProject}}$ is along z-axis ([0,0,1]). First, rotate about the x-axis by an angle γ such that the rotated $\overrightarrow{\text{MoonPoleProject}}$ has no y-component:

$$\gamma = \text{atan2}(\overrightarrow{\text{MoonPoleProject}}_y, \overrightarrow{\text{MoonPoleProject}}_z)$$

$$\overrightarrow{\text{MPP}}_x = \overrightarrow{\text{MoonPoleProject}}_x$$

$$\overrightarrow{\text{MPP}}_y = 0.0$$

$$\overrightarrow{\text{MPP}}_z = \sin\gamma \times \overrightarrow{\text{MoonPoleProject}}_y + \cos\gamma \times \overrightarrow{\text{MoonPoleProject}}_z$$

- Second, rotate about the y-axis by an angle β such that the rotated vector has no x-component, and the z-component is positive:

$$\beta = \text{atan2}(-\overrightarrow{MPP}_x, \overrightarrow{MPP}_z)$$

$$\text{if } (\cos\beta \times \overrightarrow{MPP}_z < \sin\beta \times \overrightarrow{MPP}_x),\ \beta = \text{atan2}(\overrightarrow{MPP}_x, -\overrightarrow{MPP}_z)$$

$$\overrightarrow{newMPP} = (0.0, 0.0, -\sin\beta \times \overrightarrow{MPP}_x + \cos\beta \times \overrightarrow{MPP}_z)$$

- Now find the angle α to rotate about the z-axis based on the surface location's longitude

$$\text{if } (surf_{lon} < \pi), \text{ rotate counterclockwise about z: } \alpha = -surf_{lon}$$

$$\text{otherwise, rotate clockwise about z: } \alpha = 2\pi - surf_{lon}$$

- Now rotate $\overrightarrow{MoonSunProject}$ and $\overrightarrow{MoonEarthProject}$ by γ, β, and α:

$$\begin{aligned}\overrightarrow{newMSP}_x = {} & \cos\alpha\cos\beta \times \overrightarrow{MoonSunProject}_x + \\ & (\cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma) \times \overrightarrow{MoonSunProject}_y + \\ & (\cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma) \times \overrightarrow{MoonSunProject}_z\end{aligned}$$

$$\begin{aligned}\overrightarrow{newMSP}_y = {} & \sin\alpha\cos\beta \times \overrightarrow{MoonSunProject}_x + \\ & (\sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma) \times \overrightarrow{MoonSunProject}_y + \\ & (\sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma) \times \overrightarrow{MoonSunProject}_z\end{aligned}$$

$$\begin{aligned}\overrightarrow{newMSP}_z = {} & -\sin\beta \times \overrightarrow{MoonSunProject}_x + \\ & \cos\beta\sin\gamma \times \overrightarrow{MoonSunProject}_y + \\ & \cos\beta\cos\gamma \times \overrightarrow{MoonSunProject}_z\end{aligned}$$

$$\begin{aligned}\overrightarrow{newMEP}_x = {} & \cos\alpha\cos\beta \times \overrightarrow{MoonEarthProject}_x + \\ & (\cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma) \times \overrightarrow{MoonEarthProject}_y + \\ & (\cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma) \times \overrightarrow{MoonEarthProject}_z\end{aligned}$$

$$\begin{aligned}\overrightarrow{newMEP}_y = {} & \sin\alpha\cos\beta \times \overrightarrow{MoonEarthProject}_x + \\ & (\sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma) \times \overrightarrow{MoonEarthProject}_y + \\ & (\sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma) \times \overrightarrow{MoonEarthProject}_z\end{aligned}$$

$$\overrightarrow{\text{newMEP}}_z = -\sin\beta \times \overrightarrow{\text{MoonEarthProject}}_x + \cos\beta\sin\gamma \times \overrightarrow{\text{MoonEarthProject}}_y + \cos\beta\cos\gamma \times \overrightarrow{\text{MoonEarthProject}}_z$$

- Check if the rotated $\overrightarrow{\text{newMSP}}$ or $\overrightarrow{\text{newMEP}}$ has a positive or negative y to compensate for the (northern or southern) hemisphere of the robot's location, and then make sure the azimuths are in the correct range (0 to 2π):

if $(\text{newMSP}_y < 0)$, subtract azimuth from 2π:

$$\text{azimuth} = 2\pi - \text{azimuth}$$

if $(\text{newMEP}_y < 0)$, subtract azimuth from 2π:

$$\text{Earth}_{\text{azimuth}} = 2\pi - \text{Earth}_{\text{azimuth}}$$

Finding the altitude and azimuth of the sun is not all that is needed for calculating the presence of shadows or the potential for solar power generation, as terrain features can occlude the sun. Once the solar azimuth is known, the line of map grid cells along the azimuth line from the robot can be found, to determine what features along that line might block the sun. Finding the azimuth line depends on the type of terrain map projection. Since the surface is spherical, the azimuth line must follow a great circle of the sphere to follow the rays of the sun. For small areas, with a flat-world projection, the line can be approximated by a simple diagonal. For a stereographic projection, the distance of any given cell from a specified great circle can be calculated, and all cells within a certain distance of the great circle can then be examined for terrain blocking.

Terrain blocking is determined by comparing the angular elevation of a terrain feature with the altitude of the sun. The curvature of the surface must be taken into account when calculating the terrain elevation. When performing the search of all grid cells along the great circle line, the search extends from the robot's position to either the first terrain feature that blocks the sun, or else to the edge of the known map. Past that edge, the basic algorithm assumes that the world just drops off to nothing at the edge of the map, so particularly for smaller maps, some geometrical constraints are needed to approximate whether or not the ground further away, or the body of the

sphere itself, will occlude the sun, if it hasn't already been determined to be occluded. If the angular elevation of the highest known terrain feature is already greater than zero, then no further calculations are performed, using the (possibly false) assumption that no higher terrain features exist to block the sun. Otherwise, the terrain elevation corresponding to that highest angular elevation at the edge of the map, is extended until the ray from the robot's elevation to that terrain elevation is tangent to the surface (see Figure A3). If that tangential location occurs before the edge of the map, then the original terrain blocking result was sufficient. If that location occurs beyond the edge of the map, as shown in the figure, then the angular elevation of that extended terrain elevation will be higher, and is the new maximum terrain elevation which must be compared to the sun's altitude for any shadowing. This approximation assumes that no taller, but unknown, feature (in angular elevation) exists, as well as assumes the ground doesn't drop off drastically beyond the known map.

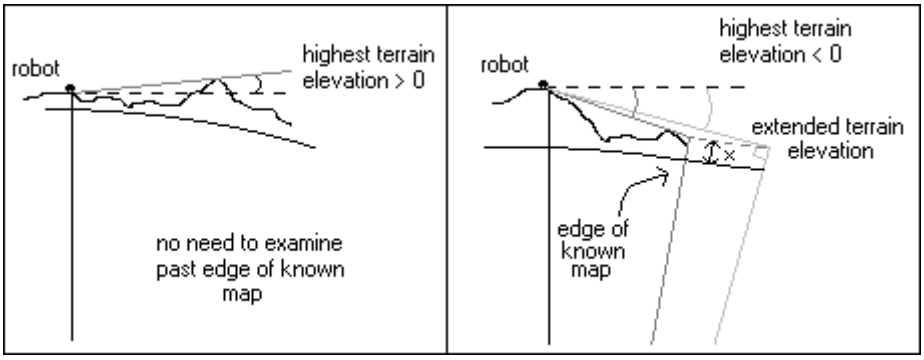


Figure A3: Calculating angular terrain elevation

C. Meteorite Search Demonstration Statistics

Table A1 lists the basic characteristics of each of the official meteorite search demonstrations performed in Antarctica in January, 2000. The total time spent during the search and the total area covered, based on the sensor's field of view, are given in the 2nd two columns. The mean wind speed and direction are given next. During the last two demos, the wind speed was considerably lower, making the wind direction readings less reliable. Then, the range of the sun's azimuth, from start to finish of the search, is given, followed by the primary heading of the straight rows pattern in the last column. The altitude of the sun in this location ranged from about 4 to 35

degrees in the latter half of January, but these searches were primarily performed during the higher range of these altitudes, during the "daytime."

Demo	Total Time (s)	Area Covered (m^2)	Mean Wind Speed (knots)	Mean Wind Direction (degrees)	Sun Azimuth (degrees)	Pattern Orientation (degrees)
0	4,035	928.25	15.5	189	349 - 329	72
1	6,881	709.75	18.3	191	346 - 314	72
2	9,468	632.00	17.2	174	16 -332	69
3	4,934	598.75	12.7	192	294 - 262	336
4	3,358	399.50	16.0	171	56 - 40	172
5	2,065	172.25	15.1	187	17 - 6	259
6	974	107.25	14.1	183	5 - 0	254
7	3,454	285.25	10.6	196	344 -328	104
8	14,163	844.00	8.6	154	70 - 343	188
9	8,646	506.75	2.9	150	354 - 315	184

Table A1: Basic search pattern characteristics

Table A2 lists the energy characteristics of the search demonstrations. The total energy generated by the small, test solar panels is listed first. Then, the expected, uncalibrated solar energy generation from concurrent simulations is listed for two cases. For one, the actual pitch and roll of the robot is considered in the solar panel orientation calculations. For the other, the pitch and roll are assumed to be zero at all times. The differences between the cases are slight, as can be seen. The mean actual pose values are shown in the next two columns, with the standard deviation also indicated. Again, these variations are fairly small for this Antarctic location. In the last column, the total energy consumed is given, after some smoothing and filtering is performed on the raw wheel and steering motor currents.

Demo	Energy Generated (J)	Simulated Energy Generated		Mean ± std of Pose		Energy Consumed (J)
		With Pitch/Roll (J)	Without Pitch/Roll (J)	Pitch (degrees)	Roll (degrees)	
0	49,162	34,411	34,196	0.42 ± 1.34	0.45 ± 0.48	1,357,338
1	81,534	57,012	56,412	0.87 ± 1.26	0.67 ± 0.46	1,905,850
2	113,201	87,441	86,995	0.38 ± 1.64	0.40 ± 0.65	2,432,547
3	58,699	45,385	45,574	0.57 ± 1.13	0.74 ± 1.75	1,126,376
4	43,020	27,375	27,498	0.76 ± 0.91	0.79 ± 1.09	1,135,780
5	27,150	21,675	21,662	– 0.61 ± 1.10	0.61 ± 0.44	681,341
6	12,253	9,835	9,820	– 0.38 ± 1.28	0.70 ± 0.41	233,152
7	42,878	34,492	34,423	– 0.55 ± 2.07	0.24 ± 0.58	891,314
8	192,830	148,191	149,228	– 0.43 ± 1.10	0.44 ± 1.45	3,346,412
9	98,878	60,491	59,568	0.49 ± 0.82	1.55 ± 1.21	2,266,750

Table A2: Search pattern energy characteristics

Table A3 lists the timing characteristics of each meteorite search. The mean time spent examining a single target is given in the first set of columns. The “camera” and “arm” columns indicate the mean time spent using just that sensor. The “both” column shows the mean time spent on a single target using both sensors, and includes any maneuvering time. However, for all targets in the first demonstration, and for several targets in the other demonstrations, the arm was not used. The second set of columns shows the percent of the total search time spent either performing the search pattern, using the camera, or using the arm. The remainder of the time was spent in maneuvers, user pauses, resuming the pattern, or network delays of some sort.

Demo	Mean Time/Target (s)			Percent of Time/Category (%)		
	Camera	Arm	Both	Pattern	Camera	Arm
0	31.2	-	31.2	94.2	3.9	-
1	42.8	505.0	179.9	52.4	10.0	29.3
2	72.4	354.9	324.9	36.9	11.5	37.5
3	55.3	463.5	220.2	59.1	6.7	18.8
4	42.7	209.5	125.5	58.0	7.6	12.5
5	107.5	636.0	487.5	24.3	10.4	30.8
6	94.0	444.0	566.0	36.0	9.7	45.6
7	96.3	417.7	535.3	35.3	8.4	36.3
8	90.6	335.8	415.0	15.3	15.3	49.8
9	93.7	270.0	358.3	13.6	19.5	46.8

Table A3: Search pattern timing characteristics

Chapter VIII

Bibliography

1 Alami, R., R. Chatila, S. Fleury, M. Ghallab, F. Ingrand, "An Architecture for Autonomy," *International Journal of Robotics Research*, Vol. 17, No. 4, April 1998, pp. 315-337.

2 Antarctic Search for Meteorites (ANSMET) web site, http://www.cwru.edu/affil/ansmet/index.html

3 Apostolopoulos, D., M. D. Wagner, B. Shamah, L. Pedersen, K. Shillcutt, and W. L. Whittaker, "Technology and Field Demonstration of Robotic Search for Antarctic Meteorites," *International Journal of Robotics Research*, December 2000.

4 Bernard, Douglas E. and Barney Pell, "Designed for Autonomy: Remote Agent for the New Millennium Program," *Proceedings of the Fourth International Symposium on Artificial Intelligence, Robotics, and Automation for Space, i-SAIRAS*, Tokyo, Japan, 1997.

5 Bester, M., *SatTrack* (computer program), http://www.bester.com/sattrack.htm

6 Bonasso, R. P., R. J. Firby, E. Gat, D. Kortenkamp, D. Miller and M. Slack, "Experiences with an Architecture for Intelligent, Reactive Agents," *Journal of Experimental and Theoretical Artificial Intelligence*, Vol. 9, No. 2, 1997.

7 Bresina, John, Gregory A. Dorais, Keith Golden, David E. Smith and Richard Washington, "Autonomous Rovers for Human Exploration of Mars," *Proceedings of the First International Conference of the Mars Society*, August 1998.

8 Brooks, Rodney A., "Solving the Find-Path Problem by Good Representation of Free Space," *Proceedings of the Second Annual Conference on Artificial Intelligence*, Pittsburgh, PA, August 1982, pp. 381-386.

9 Brumitt, Barry L. and Anthony Stentz, "Dynamic Mission Planning for Multiple Mobile Robots," *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, Minneapolis, MN, April 1996.

10 Bussey, D. Ben J., Paul D. Spudis, Mark S. Robinson, "Illumination conditions at the lunar south pole," *Geophysical Research Letters*, Vol. 26, No. 9, May 1999, pp.1187-1190.

11 Cao, Zuo Liang, Yuyu Huang and Ernest L. Hall, "Region Filling Operations with Random Obstacle Avoidance for Mobile Robots," *Journal of Robotic Systems*, Vol. 5, No. 2, 1988, pp. 87-102.

12 Cassidy, W. A., R. P. Harvey, J. W. Schutt, G. Delisle and K. Yanai, "The Meteorite Collection Sites of Antarctica," *Meteoritics,* Vol. 27, 1992, pp. 490-525.

13 Chatila, Raja, Simeon Lacroix, Thierry Simeon and Matthieu Herrb, "Planetary Exploration by a Mobile Robot: Mission Teleprogramming and Autonomous Navigation," *Autonomous Robots*, Vol. 2, 1995, pp. 333-344.

14 Chatila, Raja and Simon Lacroix, "A Case Study in Machine Intelligence: Adaptive Autonomous Space Rovers," *Proceedings of the International Conference on Field and Service Robotics*, Canberra, Australia, Dec. 8-10, 1997.

15 Choset, Howie and Joel Burdick, "Sensor Based Motion Planning: The Hierarchical Generalized Voronoi Graph," *Workshop on Algorithmic Foundations of Robotics*, 1996.

16 Choset, Howie and Philippe Pignon, "Coverage Path Planning: The Boustrophedon Cellular Decomposition," *International Conference on Field and Service Robotics*, Canberra, Australia, 1997.

17 Colozza, Anthony J., "Design and Optimization of a Self-Deploying PV Tent Array," *NASA CR 187119*, 1991.

18 Colozza, Anthony J., "Design and Optimization of a Self-Deploying Single Axis Tracking PV Array," *NASA CR 189132*, 1992.

19 Conte, G. and R. Zulli, "Hierarchical Path Planning in a Multi-Robot Environment with a Simple Navigation Function," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 25, No. 4, April 1995, pp. 651-654.

20 Coulter, R. C., "Implementation of the Pure Pursuit Path Tracking Algorithm", *CMU Technical Report CMU-RI-TR-92-01*, Robotics Institute, Carnegie Mellon University, January 1992.

21 Danish Wind Turbine Manufacturers Association web site, http://www.windpower.dk/

22 Deans, Matthew, Stewart Moorehead, Ben Shamah, Kimberly Shillcutt, and William "Red" Whittaker, "A Concept for Robotic Lunar South Pole Exploration," *CMU*

Technical Report CMU-RI-TR-97-22, Robotics Institute, Carnegie Mellon University, May 1998.

23 Deans, M. C. and M. Hebert, "Invariant Filtering for Simultaneous Localization and Mapping," *Proceedings of the International Conference on Robotics and Automation*, San Francisco, CA, 2000.

24 de Medeiros, Adelardo A. D., Raja Chatila and Sara Fleury, "Specification and Validation of a Control Architecture for Autonomous Mobile Robots," *International Conference on Intelligent Robots and Systems, IROS '96*, Osaka, Japan, Nov. 4-8, 1996.

25 Diken, Hamza, "Energy Efficient Sinusoidal Path Planning of Robot Manipulators," *Mechanism and Machine Theory*, Vol. 29, No. 6, 1994, pp. 785-792.

26 Dorais, Gregory A., R. Peter Bonasso, David Kortenkamp, Barney Pell and Debra Schreckenghost, "Adjustable Autonomy for Human-Centered Autonomous Systems on Mars," *Proceedings of the First International Conference of the Mars Society*, August, 1998.

27 Ewell, Richard C. and Dale R. Burger, "Solar Array Model Corrections From Mars Pathfinder Lander Data," *26th IEEE Photovoltaic Specialists Conference*, Anaheim, California, Sept 29 - Oct 3, 1997.

28 Farritor, Shane and Steven Dubowsky, "A Self-Planning Methodology for Planetary Robotic Explorers," *ICAR '97*, Monterey, CA, July 7-9, 1997, pp. 499-504.

29 Farritor, Shane, Steven Dubowsky, Nathan Rutman and Jeffrey Cole, "A Systems-Level Modular Design Approach to Field Robotics," *IEEE International Conference on Robotics and Automation*, Minneapolis, MN, 1996, Vol. 4, pp. 2890 - 2895.

30 Farritor, Shane, Herve Hacot, and Steven Dubowsky, "Physics-Based Planning for Planetary Exploration," *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 1998, pp.278-283.

31 Feldman, J. A. and R. F. Sproull, "Decision Theory and Artificial Intelligence II: The Hungry Monkey," *Cognitive Science*, Vol. 1, 1977, pp. 158-192.

32 Feldman, W. C., S. Maurice, A. B. Binder, B. L. Barraclough, R. C. Elphic, D. J. Lawrence, "Fluxes of Fast and Epithermal Neutrons from Lunar Prospector: Evidence for Water Ice at the Lunar Poles," *Science*, September 1998.

33 Gage, Douglas W., "Randomized search strategies with imperfect sensors," *Proceedings of SPIE Mobile Robots VIII*, Boston, Sept. 9-10, 1993, Vol. 2058, pp. 270-279.

34 Gat, Erann and Barney Pell, "Abstract Resource Management in an Unconstrained Plan Execution System," *Proceedings of IEEE Aerospace Conference*, Snomass, CO, 1998.

35 Gat, E., "On Three-Layer Architectures", *AI and Mobile Robots*, D. Kortenkamp, P. Bonasso and R. Murphy eds., MIT/AAAI Press, Cambridge, MA, 1998.

36 Gelenbe, Erol, Nestor Schmajuk, John Staddon and John Reif, "Autonomous search by robots and animals: A survey," *Robotics and Autonomous Systems*, Vol. 22, 1997, pp. 23-34.

37 Gennery, D., "Traversability Analysis and Path Planning for a Planetary Rover," *Autonomous Robots*, Vol.6, No.2, 1999, pp. 131-146.

38 Goel, Ashok K., Todd J. Callantine, Murali Shanker and B. Chandrasekaran, "Representation, Organization, and Use of Topographic Models of Physical Spaces for Route Planning," IEEE, 1991, pp. 308-314.

39 Guldner, Jurgen, Vadim I. Utkin and Rudolf Bauer, "A three-layered hierarchical path control system for mobile robots: Algorithms and experiments," *Robotics and Autonomous Systems*, Vol. 14, 1995, pp. 133-147.

40 Hait, Alain and Thierry Simeon, "Motion planning on rough terrain for an articulated vehicle in presence of uncertainties," *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS '96*, Osaka, Japan, Nov. 4-8, 1996, pp. 1126-1133.

41 Hait, A., T. Simeon and M. Taix, "A Landmark-based Motion Planner for Rough Terrain Navigation," *Proceedings of the 5th International Symposium on Experimental Robotics, ISR '97*, Barcelona, Spain, June 15-18, 1997, pp. 173-183.

42 Hart, P.E., N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, SSC-4(2), 1968, pp.100-107.

43 Healey, Anthony J., Scott McMillan, David Jenkins and Robert B. McGhee, "BUGS: Basic UXO Gathering System," *Proceedings of Autonomous Vehicles in Mine Countermeasures Symposium*, April 4-7, 1995, Monterey, CA. pp. 8-32 - 8-38.

44 Heiken, G. H., D. T. Vaniman, B. M. French, eds., *Lunar Sourcebook: a user's guide to the moon*, Cambridge University Press, 1991.

45 Hert, Susan, Sanjay Tiwari and Vladimir Lumelsky, "A Terrain-Covering Algorithm for an AUV," *Autonomous Robots*, Vol. 3, 1996, pp. 91-119.

46 Hofner, Christian and Gunther Schmidt, "Path planning and guidance techniques for an autonomous mobile cleaning robot," *Robotics and Autonomous Systems*, Vol. 14, 1995, pp.199-212.

47 *Horizons* (web-based software), http://ssd.jpl.nasa.gov/horizons.html, Jet Propulsion Laboratory.

48 Hou, Edwin S. H. and Dan Zheng, "Mobile Robot Path Planning Based on Hierarchical Hexagonal Decomposition and Artificial Potential Fields," *Journal of Robotic Systems*, Vol. 11, No. 7, 1994, pp.605-614.

49 Ingersoll, A. P., T. Svitek, B. C. Murray, "Stability of Polar Frosts in Spherical Bowl-Shaped Craters on the Moon, Mercury, and Mars," *Icarus*, Vol. 100, 1992, pp. 40-47.

50 Kurabayashi, Daisuke, Jun Ota, Tamio Arai and Eiichi Yoshida, "Cooperative Sweeping by Multiple Mobile Robots," *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, April 1996, pp. 1744-1749.

51 Landis, Geoffrey A., "Solar Cell Selection For Mars," *Second World Conference on Photovoltaic Solar Energy Conversion*, Vienna, Austria, July 6-10, 1998.

52 Landis, Geoffrey A. and Joseph Appelbaum, "Photovoltaic Power Options for Mars," *Space Power*, Vol. 10, No. 2, 1991, pp. 225-237.

53 Latombe, J. C., *Robot Motion Planning*, Kluwer Academic Publishers, Norwell, MA, 1991.

54 Laubach, Sharon L., Joel Burdick and Larry Matthies, "An Autonomous Path Planner Implemented on the Rocky7 Prototype Microrover," *Proceedings of the 1998 International Conference on Robotics and Automation*, Leuven, Belgium, May 16-20, 1998.

55 Lee, Jang Gyu and Hakyoung Chung, "Global Path Planning For Mobile Robot With Grid-Type World Model," *Robotics and Computer-Integrated Manufacturing*, Vol. 11, No. 1, 1994, pp. 13-21.

56 Leon, V. Jorge, David Kortenkamp and Debra Schreckenghost, "A Planning, Scheduling and Control Architecture for Advanced Life Support Systems," *NASA Workshop on Planning and Scheduling for Space*, October, 1997.

57 Lumelsky, Vladimir J. and Alexander A. Stepanov, "Path-Planning Strategies for a Point Mobile Automaton Moving Amidst Unknown Obstacles of Arbitrary Shape," *Algorithmica*, Vol. 2, 1987, pp. 403-430.

58 Margot, J. L., D. B. Campbell, M. A. Slade, "Topography of the Lunar Poles from Radar Interferometry: A Survey of Cold Trap Locations," *Science*, Vol. 284, No. 5420, June 4, 1999.

59 Margot, J. L., D. B. Campbell, M. A. Slade, "Digital Elevation Models of the Moon from Earth-Based Radar Interferometry," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 38, No. 2, March 1, 2000, p. 2.

60 Matthies, Larry, Erann Gat, Reid Harrison, Brian Wilcox, Richard Volpe and Todd Litwin, "Mars Microrover Navigation: Performance Evaluation and Enhancement," *Autonomous Robots*, Vol. 2, No. 4, 1995, pp. 291-312.

61 McKissock, Barbara I., Lisa L. Kohout and Paul C. Schmitz, "A Solar Power System for an Early Mars Expedition," *American Institute of Chemical Engineers Summer National Meeting* (also *NASA TM 103219*), San Diego, CA, Aug 19-22, 1990.

62 Meeus, Jean, *Astronomical Algorithms*, Willmann-Bell: Richmond, VA, 1991.

63 Miles, David W. and Stephen M. Rock, "Real-Time Dynamic Trajectory Optimization," *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Reston, VA, July 1996, Paper No. 96-3741.

64 Moorehead, S., R. Simmons, D. Apostolopoulos, and W.L. Whittaker, "Autonomous Navigation Field Results of a Planetary Analog Robot in Antarctica," *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Noordwijk, the Netherlands, June 1-3, 1999.

65 Morignot, Philippe and Barbara Hayes-Roth, "Why Does an Agent Act?" *AAAI Spring Symposium on Representing Mental States and Mechanisms*, Stanford, CA, March 1995.

66 Morignot, Philippe and Barbara Hayes-Roth, "Adaptable Motivational Profiles for Autonomous Agents," *Knowledge Systems Laboratory Report No. KSL 95-01*, Stanford University, 1995.

67 Muscettola, Nicola, Paul Morris, Barney Pell and Ben Smith, "Issues in Temporal Reasoning for Autonomous Control Systems," *Proceedings of the Second International Conference on Autonomous Agents*, Minneapolis, MI, 1998.

68 Muscettola, Nicola, P. Pandurang Nayak, Barney Pell and Brian C. Williams, "Remote Agent: To Boldly Go Where No AI System Has Gone Before," *Artificial Intelligence*, Vol. 103, No. 1/2, August 1998.

69 Neff, J. M. and W. T. Fowler, "Minimum-Fuel Rescue Trajectories for the Extravehicular Excursion Unit," *The Journal of the Astronautical Sciences*, Vol. 39, No. 1, Jan.-Mar. 1991, pp. 21-45.

70 Pedersen, L., D. Apostolopoulos, W.L. Whittaker, W. Cassidy, P. Lee, and T. Roush, "Robotic Rock Classification Using Visible Light Reflectance Spectroscopy:

Preliminary results from the Robotic Antarctic Meteorite Search program," *Lunar and Planetary Science Conference XXX*, 1999.

71 Pell, Barney, Edward B. Gamble, Erann Gat, Ron Keesing, James Kurien, William Millar, P. Pandurang Nayak, Christian Plaunt and Brian C. Williams, "A Hybrid Procedural/ Deductive Executive For Autonomous Spacecraft," *Proceedings of the Second International Conference on Autonomous Agents*, Minneapolis, MI, 1998.

72 Pell, Barney, Douglas E. Bernard, Steve A. Chien, Erann Gat, Nicola Muscettola, P. Pandurang Nayak, Michael D. Wagner and Brian C. Williams, "An Autonomous Spacecraft Agent Prototype," *Proceedings of the First International Conference on Autonomous Agents*, Marina del Rey, CA 1997.

73 Rao, Nageswara S. V., "Robot Navigation in Unknown Generalized Polygonal Terrains Using Vision Sensors," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 25, No. 6, June 1995, pp. 947-962.

74 Real-Time Innovations, Inc., web site, http://www.rti.com/products/ndds/ndds.html.

75 Roy, A.E., *Orbital Motion*, Wiley & Sons, New York, 1978.

76 Scheiman, David A., Cosmo Baraona, Dave Wilt, Phillip Jenkins and Geoffrey A. Landis, "Mars Array Technology Experiment (MATE) for the 2001 Lander," *Second World Conference on Photovoltaic Solar Energy Conversion*, Vienna, Austria, July 6-10, 1998.

77 Sheu, Chi-Haur and Kuu-Young Young, "A Heuristic Approach to Robot Path Planning Based on Task Requirements Using a Genetic Algorithm," *Journal of Intelligent and Robotic Systems*, Vol. 16, 1996, pp. 65-88.

78 Shillcutt, Kimberly and William Whittaker, "Modular Optimization for Robotic Explorers," *Integrated Planning for Autonomous Agent Architectures, AAAI Fall Symposium*, Orlando, FL, October 23-25, 1998.

79 Shillcutt, Kimberly, Dimitrios Apostolopoulos, and William Whittaker, "Patterned Search Planning and Testing for the Robotic Antarctic Meteorite Search," *International Topical Meeting on Robotics and Remote Systems for the Nuclear Industry*, American Nuclear Society, Pittsburgh, PA, April 25-29, 1999.

80 Shillcutt, Kimberly, and William Whittaker, "Solar Power Expert for Remote Robotic Explorers," *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Noordwijk, the Netherlands, June 1-3, 1999.

81 Shirbacheh, Mike, "Power and Pyro Sybsystems for Mars Pathfinder," *Proceedings of the 32nd Intersociety Energy Conversion Engineering Conference*, Honolulu, Hawaii, July 27 - Aug 1, 1997, pp. 2231-2236.

82 Simmons, Reid, "An Architecture for Coordinating Planning, Sensing, and Action," *Proceedings of the DARPA Workshop on Innovative Approaches to Planning, Scheduling and Control*, San Mateo, CA, 1990, pp. 292-297.

83 Singh, S., R. Simmons, M.F. Smith, III, A. Stentz, V. Verma, A. Yahja, and K. Schwehr, "Recent Progress in Local and Global Traversability for Planetary Rovers," *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, CA, April, 2000.

84 Slack, Marc G., "Navigation Templates: Mediating Qualitative Guidance and Quantitative Control in Mobile Robots," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 23, No. 2, 1993, pp.445-451.

85 Stentz, Anthony, "Optimal and Efficient Path Planning for Partially Known Environments," *Intelligent Unmanned Ground Vehicles: Autonomous Navigation Research at Carnegie Mellon*, M. Hebert, C. Thorpe and T. Stentz, eds., Kluwer Academic Publishers, Norwell, MA, 1997, pp. 203-219.

86 Stentz, Anthony, "Optimal and Efficient Path Planning for Unknown and Dynamic Environments," *Carnegie Mellon Robotics Institute Technical Report CMU-RI-TR-93-20*, 1993.

87 Stentz, Anthony, "The D* Algorithm for Real-Time Planning of Optimal Traverses," *Carnegie Mellon Robotics Institute Technical Report CMU-RI-TR-94-37*, 1994.

88 Sukhatme, Gaurav S. and George A. Bekey, "Multicriteria Evaluation of a Planetary Rover," *Proceedings of the Planetary Rover Technology and Systems Workshop, IEEE International Conference on Robotics and Automation*, Minneapolis MN, April 22-28, 1996.

89 Task Control Architecture, web site, http://www.cs.cmu.edu/afs/cs.cmu.edu/project/TCA/release/tca.orig.html, Carnegie Mellon University.

90 Thurman, Sam W., Project Engineer/Flight Operations Manager, Mars Surveyor '98, personal communication, October 1998.

91 Tschauner, Johann F. A. and Peter R. Hempel, "On Minimum-Fuel Rendezvous-Techniques," *AIAA 2nd Aerospace Sciences Meeting*, New York, Jan. 25-27, 1965.

92 Vidal, Thierry, Malik Ghallab and Rachid Alami, "Incremental Mission Allocation to a Large Team of Robots," *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, April 1996, pp. 1620-1625.

93 Wilcox, Brian, Larry Matthies, Donald Gennery, Brian Cooper, Tam Nguyen, Todd Litwin, Andrew Mishkin and Henry Stone, "Robotic Vehicles for Planetary

Exploration," *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, Nice, France, May 1992.

94 Zelinsky, Alexander, "Using Path Transforms to Guide the Search for Findpath in 2D," *International Journal of Robotics Research*, Vol 13, No. 4, August 1994, pp. 315-325.

95 Zelinsky, A., R. A. Jarvis, J. C. Byrne and S. Yuta, "Planning Paths of Complete Coverage of an Unstructured Environment by a Mobile Robot," *International Conference on Advanced Robotics, ICAR'93*, November 1993, Tokyo, Japan, pp.533-538.

96 Zelinsky, Alexander, "A Mobile Robot Navigation Exploration Algorithm," *IEEE Transactions of Robotics and Automation*, Vol. 8, No. 6, Dec 1992, pp.707-717.