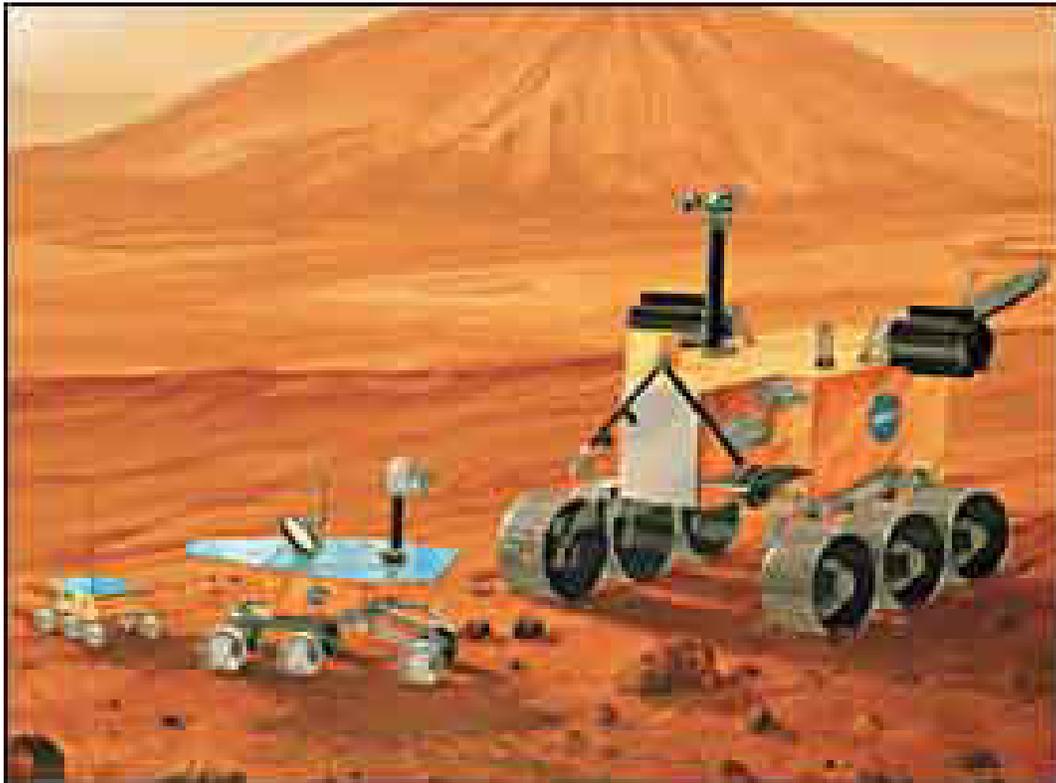


MIT OpenCourseWare
<http://ocw.mit.edu>

16.89J / ESD.352J Space Systems Engineering
Spring 2007

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.

Rapid Modeling of Mars Robotic Explorers



16.89 Design Document

Spring 2003

Massachusetts
Institute of
Technology

This work was completed as part of the 16.89 Space Systems Engineering course, a semester-long graduate design course in the Department of Aeronautics and Astronautics at the Massachusetts Institute of Technology.

Copyright 2003 Massachusetts Institute of Technology (MIT)

No part of this report may be reproduced, stored in a retrieval system or transmitted in any form or by any means without the written permission of MIT.

This document is available in Adobe PDF and Microsoft Word formats.

Contributors

Instruments

Christopher Roberts
Julie Wertz

Acquisition

Ian Garrick-Bethell
Erisa Hines

Environment

Stephanie Chiesi
Jessica Marquez

Rover

Mark Hilstad
Julien Lamamy

Power

Kalina Galabova
Roshanak Nilchiani

Communications

Babak Cohanin
Tsoline Mikaelian

Autonomy

Edward Fong
Barry Willhite

Faculty

Professor David Miller
Colonel John Keesee
Joseph Parrish (Payload Systems, Inc.)

Acknowledgements

The contributors to this report would like to acknowledge the following people for their contributions, advice, information and guidance.

Charles Whetsel, JPL
Allen Chen, JPL

Forward

The Rapid Modeling of Mars Robotic Explorers project is the result of a one-semester course, 16.89 Space Systems Engineering, held in the spring of 2003 in the Department of Aeronautics and Astronautics at the Massachusetts Institute of Technology. This graduate design course places an emphasis on systems engineering as an approach to design.

The objective of 16.89 is to further develop systems knowledge by applying systems engineering techniques to a particular design problem. The systems engineering method focuses on design based on the mission needs. The particular project described in this document resulted in a tool for conducting system level trades for Mars rovers, based on user-specified payloads and other mission attributes.

During the course of the semester, the class had several milestones marking the development of the design tool: a Trades Analysis and Requirements Review, a Preliminary Design Review, and a Critical Design Review. The class also produced two documents: a Trades and Requirements Document and this final Design Document. The class was divided into subsystem teams, allowing the members of each team to focus on the design options and algorithms for that specific subsystem. This report and the project code are the final products of the course. This design document includes descriptions of the subsystem modules and a user's manual for the code.

Table of Contents

Forward	5
Table of Contents	6
Figures and Tables.....	9
1.0 Executive Summary	12
1.1 Motivation and Objectives	12
1.2 Project Requirements and Assumptions.....	12
1.3 Design Process.....	12
1.4 Subsystems	13
1.5 System Validation.....	13
1.6 Conclusions	15
2.0 Project Overview	16
2.1 Introduction.....	16
2.1.1 Project Background	16
2.1.2 Motivation.....	16
2.1.3 Objectives.....	16
2.2 Requirements	17
2.2.1 Project Scope	17
2.2.2 Project Requirements	17
2.3 Project Overview	18
2.3.1 System Architecture	18
2.3.1.1 Science Vector	18
2.3.1.2 Design Vector	19
2.3.1.3 Cost Module.....	19
2.3.1.4 Complexity Module	20
2.3.2 System Design Flow	22
3.0 Project Subsystems.....	25
3.1 Instruments.....	25
3.1.1 Responsibilities	25
3.1.2 Instruments Database	25
3.1.3 Sources	26
3.1.4 Assumptions.....	26
3.1.5 Validation.....	27
3.1.6 Current Instruments.....	27
3.1.6.1 Contact Suite Instruments	28
3.1.6.2 Remote Sensing Instruments	29
3.1.6.3 Analytic Laboratory Instruments:.....	29
3.1.7 Expandability.....	30
3.1.8 References	30
3.1.9 Appendix	32
3.2 Acquisition.....	35
3.2.1 Responsibilities	35
3.2.2 Assumptions.....	35
3.2.3 Background.....	35
3.2.4 Program Flow	37
3.2.5 Validation.....	38
3.2.6 Expandability.....	38
3.2.7 References	39
3.3 Environment.....	40
3.3.1 Responsibilities.....	40

3.3.2	Assumptions.....	40
3.3.3	Background.....	40
3.3.3.1	Solar Irradiance.....	40
3.3.3.2	Hours of Sunlight per Sol.....	43
3.3.3.3	Temperature.....	44
3.3.3.4	Rock Density Determination.....	45
3.3.3.5	Other Outputs of the Environment Module.....	46
3.3.4	Program Flow.....	46
3.3.5	Validation.....	47
3.3.6	Expandability.....	50
3.3.7	References.....	50
3.4	Rover.....	52
3.4.1	Responsibilities.....	52
3.4.2	Assumptions - General.....	53
3.4.2.1	Scope and level of detail.....	53
3.4.2.2	Materials.....	53
3.4.2.3	Maximum acceleration.....	53
3.4.3	Assumptions – Mobility.....	53
3.4.3.1	Mobility dimensions.....	53
3.4.3.2	Mobility structural design.....	54
3.4.3.3	Mobility performance.....	55
3.4.4	Assumptions - Structure.....	56
3.4.4.1	WEB geometry.....	56
3.4.4.2	WEB size.....	57
3.4.4.3	WEB structural design.....	58
3.4.5	Assumptions - Thermal.....	59
3.4.6	Assumptions – Arm and Mast.....	59
3.4.7	Default Values.....	60
3.4.8	Background.....	60
3.4.8.1	Structural calculations – suspension system.....	60
3.4.8.2	Structural calculations – mast.....	62
3.4.8.3	Structural calculations - arm.....	62
3.4.8.4	Structural calculations – plate bending.....	63
3.4.8.5	Structural calculations – plate buckling.....	64
3.4.8.6	Structural calculations – plate tension.....	64
3.4.8.7	Thermal calculations.....	65
3.4.9	Program Flow.....	65
3.4.9.1	Thermal module algorithm.....	66
3.4.10	Validation.....	68
3.4.10.1	Mobility – rover maximum speed.....	68
3.4.10.2	Mobility – drive power.....	69
3.4.10.3	WEB and mobility mass.....	70
3.4.10.4	Thermal.....	70
3.4.11	Expandability.....	71
3.4.11.1	Mobility.....	71
3.4.11.2	Thermal.....	72
3.4.12	References.....	73
3.5	Power.....	75
3.5.1	Responsibilities.....	75
3.5.2	Assumptions.....	75
3.5.3	Background.....	76
3.5.3.1	Solar cells.....	76
3.5.3.2	Batteries.....	77

3.5.3.3	Solar cell and battery calculation	78
3.5.3.4	Nuclear Power.....	81
3.5.4	Program Flow	83
3.5.4.1	Solar Module.....	84
3.5.4.2	RTG Module.....	85
3.5.5	Validation.....	86
3.5.5.1	Solar Module.....	86
3.5.5.2	RTG Module.....	88
3.5.6	Expandability.....	89
3.5.6.1	Solar Module.....	89
3.5.6.2	RTG Module.....	90
3.5.7	References.....	90
3.5.8	Appendix	92
3.6	Communications.....	94
3.6.1	Responsibilities	94
3.6.2	Assumptions.....	94
3.6.3	Background.....	95
3.6.4	Program Flow	95
3.6.5	Validation.....	96
3.6.5.1	Link Budget	96
3.6.5.2	Communication Delay	98
3.6.6	Expandability.....	103
3.6.7	References.....	103
3.7	Autonomy.....	104
3.7.1	Responsibilities	104
3.7.2	Assumptions.....	104
3.7.3	Background.....	105
3.7.4	Program Flow	105
3.7.5	Validation.....	106
3.7.6	Expandability.....	108
3.7.7	References.....	109
4.0	System Integration and Validation.....	111
4.1	System Validation.....	111
4.1.1	Past Mission Representation.....	111
4.1.2	Single Axis Trades.....	113
4.2	Trade Space Results	122
4.2.1	MER Scenario Trade Space.....	122
4.2.1.1	Number of Samples Over Mass Metric.....	122
4.2.1.2	Number of Samples Over Cost Metric.....	126
4.2.1.3	Sensitivity Analysis.....	128
4.2.2	MSL Scenario Trade Space	131
4.2.3	Complexity	133
4.3	Conclusions	134
4.3.1	Technology Investments.....	134
4.3.2	Future Work.....	134
Appendix.....		137
A	Project Code	137
B	User Manual.....	138
B.1	Overview.....	138
B.2	Creating a Trade Space.....	138
B.2.1	Selecting the Science Vector	139
B.2.2	Selecting the Design Vector	140
B.2.3	Creating Designs	142

B.2.4	Preparing Designs for Analysis	143
B.3	Analyzing Designs	144
B.3.1	Viewing Subsystem Properties.....	144
B.3.2	Selecting a Design	145
B.3.3	Visualizing Results	146
B.4	Command Line Interface.....	148
B.5	Validation.....	148
B.6	Expandability	149
C	Contributors Resumes	150
D	Project References.....	162

Figures and Tables

Figure 1.5.1:	MER scenario trade space; Number of samples versus rover mass.....	14
Table 1.5.1:	Design Vectors and results for rover designs on the Pareto front	14
Figure 2.3.1:	Comparison of cost estimates	20
Table 2.3.1:	Complexity Levels	21
Table 2.3.2:	Complexity Level assignments for design elements	21
Table 2.3.3:	Example Mission	22
Figure 2.3.2:	N-Squared diagram for the rapid modeling design tool	22
Figure 2.3.3:	Project flow model for the program.....	23
Figure 2.3.4:	Rover design model details	24
Table 3.1.1:	Sample entry in Instrument database.....	26
Table 3.1.2:	Comparison between database mass values and Whetsel estimates.....	27
Table 3.1.3:	Instruments currently included in the Instruments module.....	28
Table 3.2.1:	Acquisition Tools in the Database and Some Selected Properties	36
Figure 3.2.1:	Flow of the Acquisition module from the user GUI to the outputs given to other modules	37
Table 3.3.1:	Representative latitudes and average albedo for each latitude band.....	41
Figure 3.3.1:	Average albedo over latitudes.....	41
Figure 3.3.2:	Solar Zenith Angle as a Function of Areocentric Longitude and Hour Angle across Latitude Bands	42
Figure 3.3.3:	Average solar irradiance across low latitude bands.....	43
Figure 3.3.4:	Hours of sunlight on Mars as a function of latitude and areocentric longitude.....	44
Figure 3.3.5:	Examples of good fits of the GCM to observational data. The black solid lines show measured temperature profiles from Mars Global Surveyor while the red dashed lines show the GCM predictions.....	44
Figure 3.3.6:	Cumulative number of rocks with diameter > D per m ² versus rock diameter in (m) [5]	45
Figure 3.3.7:	Solar irradiance flow diagram	47
Figure 3.3.8:	Watt-Hours per square meter on a horizontal surface on Mars per sol. Three latitudes are shown: 0°, 60° N, and -60° S. Optical depth = 0.5.....	48
Figure 3.3.9:	Total Watt-Hours on a horizontal surface on Mars as a function of latitude [9].	49
Figure 3.3.10:	(a) Data produced by the Environment module for cumulative fractional area covered with rocks larger than rock size (in meters). Rock coverage percentages range from 5% to 50%. (b) Published data from JPL of cumulative fractional area covered with rocks larger than diameter (in meters). Past Mars surface mission data also shown.	50
Figure 3.4.1:	Suspension structural model (based on [MER]).....	55
Figure 3.4.2:	Velocity as a function of wheel diameter.....	56
Figure 3.4.3:	WEB geometry.....	57

Figure 3.4.4: Wall with Z-spar structure type.....	58
Figure 3.4.5: Thermal model for the Warm Electronics Box (WEB).....	59
Table 3.4.1: Mobility constants	60
Table 3.4.2: Structure constants	60
Figure 3.4.6: Program flow for the Rover module and sub-modules. M , P , L and T stand for mass, power, size and thickness, respectively.	66
Figure 3.4.7: Comparison of maximum velocity models.....	68
Figure 3.4.8: Driving Cycle	68
Figure 3.4.9: Comparison of average velocities.....	69
Table 3.4.3: Actual and modeled drive power per wheel, in Watts.	70
Figure 3.4.10: Heater and heat pipe power as a function of steady-state ambient temperature.	71
Figure 3.4.11: Response factor of forced vibrations.....	71
Figure 3.5.1: The amount of solar energy available on Mars.....	76
Table 3.5.1: Secondary battery types available	78
Figure 3.5.2: Power profiling for Mars surface rovers.....	79
Figure 3.5.3: Power system comparison.....	81
Figure 3.5.4: Pu-238 power supply.....	82
Table 3.5.2: RTG options for a Mars surface rover mission.....	83
Figure 3.5.5: Process flow of the Power module.....	84
Figure 3.5.6: Process flow of the solar module.....	85
Figure 3.5.7: Mission lifetime versus solar array mass.....	86
Figure 3.5.8: Mission lifetime versus power system cost versus solar array mass	87
Figure 3.5.9: Mission lifetime versus total rover mass versus solar array mass.....	87
Figure 3.5.10: Mission lifetime versus total rover mass versus secondary battery mass	88
Figure 3.5.11: Number of samples versus mission lifetime	89
Figure 3.5.12: Number of samples versus power system mass	89
Figure 3.6.1: Communications subsystem program flow.....	95
Table 3.6.1: Validation using FireSat benchmark.....	97
Table 3.6.2: Link budget results.....	97
Figure 3.6.2: Communication duration	99
Figure 3.6.3: Total communication delay for a command cycle	100
Figure 3.6.4: Total communication delay for a command cycle	101
Figure 3.6.5: Odyssey data volume from MER per sol. The data rate is 128 kbps. [4].....	102
Figure 3.6.6: Uplink duration	102
Figure 3.7.1: Sample MER mission scenario.....	106
Figure 3.7.2: Autonomy module validation	107
Table 3.7.1: Module validation for A1	107
Table 3.7.2: Module validation for A3	108
Table 4.1.1: Design Vector values based on MER information.....	111
Table 4.1.2: Science Vector inputs based on MER information	112
Table 4.1.3: Code output parameters used for validation against MER.....	113
Figure 4.1.1 Lifetime vs. number of samples using navigational autonomy levels as the varied parameter.....	114
Figure 4.1.2 Lifetime vs. number of samples using acquisition autonomy as the varied parameter.	115
Figure 4.1.3 Lifetime vs. number of samples with communication system as the varied parameter.	116
Figure 4.1.4 Lifetime vs. mass of solar array	117
Figure 4.1.5 Wheel diameter vs. total rover mass.....	118
Figure 4.1.6 Wheel diameter vs. number of obstacles.....	119
Figure 4.1.7 Wheel diameter vs. path planning time.....	120
Figure 4.1.8 Wheel diameter vs. average rover traversal speed	121
Table 4.2.1: MER-like scenario Science and Design Vectors.....	122

Figure 4.2.1: MER-like rover trade space; Number of samples versus rover mass	123
Table 4.2.2: Design Vectors for rover designs on the Pareto front.....	124
Figure 4.2.2: MER-like trade space; Isometric lines were used to compare designs with similar productivity ratios.....	125
Table 4.2.3: Isoperformance line 5:7 Design Vectors.....	125
Table 4.2.4: Isoperformance line 3:7 Design Vectors.....	126
Table 4.2.5: Isoperformance line 2:7 Design Vectors.....	126
Figure 4.2.3: MER-like rover trade space; Number of samples vs. total cost.....	127
Table 4.2.6: Cost optimal Design Vector for rover labeled 'M'	128
Figure 4.2.4: A MER scenario trade space with varying landing dates.....	129
Table 4.2.7: Values in the Design Vector for.....	130
Figure 4.2.5.....	130
Figure 4.2.5: The MER-like trade space showing the traveled distance between sites versus the rover wheel diameter	131
Table 4.2.8: Values for MSL trade space calculation; Values not shown were set to MER default values	131
Table 4.2.9: Design Vector parameters for comparing designs 1-4.....	132
Figure 4.2.6: Total Number of samples vs. Cost for an estimated MSL trade space.	132
Figure 4.2.7: Rover total mass vs. Number of samples for an estimated MSL trade space.	133
Figure B.1 The main graphical user interface (GUI) at startup. Called by typing rover4mars at the MATLAB command line.....	139
Figure B.2 The Science Vector selection interface is dynamically generated based on the contents of the lookup tables for the given subsystems.....	140
Figure B.3 The Design Vector selection interface. Default values span MER and MSL mission parameters.....	141
Figure B.4 The status bar and status text reflect the current progress of the trade space calculations.....	142
Figure B.5 An example of feedback after a trade space calculation has completed.....	143
Figure B.6 Once the user presses [Analyze designs], the user interface is fully enabled and plots are ready to be generated	144
Figure B.7 For the Acquisition and Instruments subsystems, the Index slider determines which instrument or tool is displayed.....	145
Figure B.8 a) The design number, selection zoom box, and [Graphical selection] button. b) The graphical selection utility allows the user to click on a point design to select it. If the selection zoom is greater than zero, then clicking on a point opens a second selection window that is zoomed in by the specified factor.....	146
Figure B.9 a) A 2-D plot and a default plot pull-down menu. b) A 3-D plot.....	147
Figure B.10. a) A rendering of the currently selected rover design. b) When the rendering is exported to a figure, the solar panel is made translucent.....	147
Figure B.11 The user is informed when [Create designs] is attempted without valid SCIENCE and DESIGNS structures in the workspace.....	148
Figure B.12 The user is informed when [Analyze designs] is attempted without a valid ROVERS structure in the workspace.....	149
Figure B.13 The user is informed of an error. In this example, the function master attempts to access a non-existent entity called I_am_not_a_function_or_variable.....	149

1.0 Executive Summary

1.1 Motivation and Objectives

The Rapid Modeling of Mars Robotic Explorers project is motivated by the need to evaluate and compare rover architectures and designs for future unmanned Mars missions. The software tool developed by this project can compare costs and benefits over a large space of rover architectures by varying mission science and design parameters. Future missions, including the 2009 Mars Science Laboratory (MSL), can benefit from the trade space analysis provided by this tool.

1.2 Project Requirements and Assumptions

The project requirements were determined based on suggestions from personnel involved in Mars mission design at the Jet Propulsion Laboratory. The software tool is required to reliably and efficiently explore system-level architectural and design trades for a Mars rover. The tool must be easy to use and computationally efficient. The tool must be modular in implementation, allowing for expandability and adaptability to new technologies, and upgradeability as higher fidelity subsystem models become available. The tool should create a database of rover designs that can be examined and compared using a graphical user interface.

Because the project duration was limited to a single semester, several assumptions were made to limit the scope, simplifying and streamlining the modeling process. The tool creates a trade space of designs for single rover missions, and considers only the design drivers and constraints that are related to surface operations. This simplification eliminates from consideration design drivers related to launch, cruise, entry, descent, and landing, and other technical issues not specifically associated with surface system operation. Allowable landing sites do not include polar regions, and active landers and multiple rover (cooperative) architectures are excluded from consideration.

1.3 Design Process

A Science Vector and a Design Vector define each rover design in the trade space. The Science Vector defines constants for a given trade space, such as the science and navigation payload and site-specific information. Each parameter in the Design Vector is given a set of allowable values, determining the size and nature of the trade space. The Design Vector parameters are: mission lifetime, wheel diameter, computational capability, power system, communication system, and levels of autonomy. Each of these parameters has either a range of possible values or a choice of configurations that correspond to design choices in the mission and rover design. For each new rover design, a single value for each parameter in the Design Vector is chosen. This instantiation of the Design Vector is held constant, and a point design is created in the trade space. Allowable values for the mission lifetime can range from 30 to 1,000 sols. Wheel diameter can range from 0.05 to 1 meter. Computational capability is specified as a fraction or multiple of the computational ability of a RAD-6000 flight computer. Solar arrays and radioisotope thermoelectric generators are the two options for power source. Communication systems include direct-to-earth, low orbiter relay, high orbiter relay, and hybrid combinations of these three systems. Autonomy capability can be specified separately for long-distance traversal, short-distance traversal, autonomous acquisition, night navigation, and instrument night

processing. Traversal autonomy levels are specified as Mars Exploration Rovers (MER)-equivalent (A1) or highly autonomous (A3).

Information from the Science and Design Vectors is used by subsystems to calculate rover attributes. The subsystems were individually validated to demonstrate the model fidelity at the subsystem level. The subsystem modules were then integrated and a system level validation was performed to show that the tool functions as a whole and information passed between subsystem modules is accurate and consistent.

The tool creates and saves a database of rovers for a given Science Vector and range of Design Vectors. This database of rovers can be analyzed and visualized in the graphical user interface environment, producing visibility into interesting trends and trade-offs between different design variables.

1.4 Subsystems

The project personnel were organized into seven subsystem design teams: Acquisition, Autonomy, Communications, Environment, Instruments, Power, and Rover. An N-squared diagram showing subsystem inputs and outputs was used to identify and reduce the number of feedback loops between subsystem modules. The design teams were responsible for researching, developing, and validating models for the individual subsystems, and these models were combined to produce the system-level design algorithm. An eighth team, Systems, made up of members of the different subsystem teams, was tasked with project oversight, source code integration, and additional duties during design review presentations. A new group of students was assigned to the Systems team after each design review.

In addition, models were developed for complexity and cost, and a graphical user interface and design visualization and analysis tools were created.

1.5 System Validation

System validation for the rover design tool was performed by comparison of a rover design based on MER Science and Design Vectors to the actual MER mission specifications, and by verifying that known single axis trends behave as expected. The data show that for a rover design based on MER, the distance traversed per sol, top speed, and actual speed are similar to the estimated MER values. Roving time and thinking time are also similar to those of MER. Deviations in power, size, and mass of the rover can in part be attributed to the lack of margin included in the tool, and adding a system-level mass margin should reduce the discrepancy.

A trade space centered about the MER design produced four non-dominated rover architectures, as shown in Figure 1.5.1. The Design Vectors used to produce these designs are shown in Table 1.5.1. All four designs have a direct to Earth (DTE) communication system, high-capability (A3) short distance navigation autonomy, and autonomous acquisition. Designs 1 and 3 have low-capability (A1) long distance navigation autonomy and a single RAD-6000-equivalent computer, which results in a lower mass system than designs 2 and 4, which use A3 long distance autonomy but require additional computer hardware. As expected, nighttime data processing penalizes solar powered rovers due to more stringent battery requirements. The best RTG designs have nighttime processing, resulting in a higher science return.

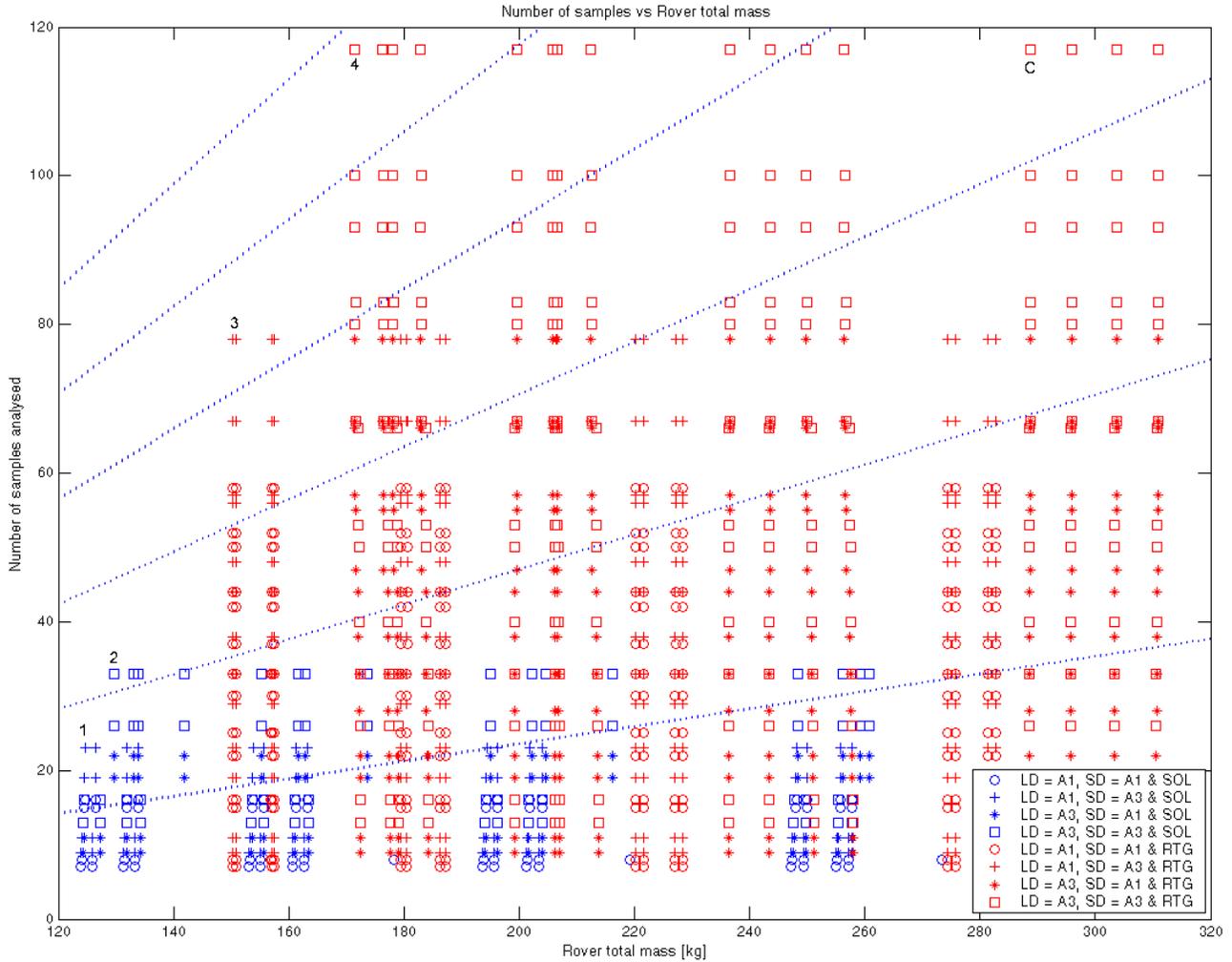


Figure 1.5.1: MER scenario trade space; Number of samples versus rover mass

Table 1.5.1: Design Vectors and results for rover designs on the Pareto front

	Design #1	Design #2	Design #3	Design #4
Mass [kg]	~124.5	~129.5	~150.5	~171.5
Samples	23	33	78	117
Lifetime [sols]	200	200	700	700
Wheel Diam. [m]	0.25	0.25	0.25	0.25
Power	Solar	Solar	RTG	RTG
Communication	DTE	DTE	DTE	DTE
Long Dist. Aut.	A1	A3	A1	A3
Short Dist. Aut.	A3	A3	A3	A3
Acquisition Aut.	Yes	Yes	Yes	Yes
Night Processing	No	No	Yes	Yes
Computers	1	2	1	2

1.6 Conclusions

The Rapid Modeling of Mars Robotic Explorers software tool has the potential to make a valuable contribution to the understanding of system level trades during the design and selection of rover architectures for Mars surface missions. Regardless of the limiting assumptions imposed by the scope of the project, the tool has many capabilities. The rover designs feature various levels of autonomy, hybrid communication systems, different instrument suites and different options for power source, landing site, and environment. The tool is highly modular and easily upgradeable. It allows for updates to the code as new technologies and strategies are developed. Although the tool can be very useful in its current state, a number of improvements can be made in terms of memory usage, computational speed, accuracy of results and a variety of other expansion options.

Revision 1 of the tool is capable of producing a trade space for landed single rover missions. Recommendations for a revision 2 of the tool include many upgrades and corrections. Modeling of the launch, cruise, entry, descent, and landing stages may be accomplished by adding modules to the current tool, and by modifying existing modules. Expanding current models to incorporate polar landing sites widens the scope of the tool to cover the entire Martian surface. Fundamentally different architectures, such as 4-wheel rovers, active landers, and multiple rover systems, can be included in the trade space analysis with some additional modeling. Currently the payload is fixed in the Science Vector for any given trade space, but the payload could also be made a design parameter that could be iterated upon and analyzed in the trade space; however, this modification would dramatically increase the number of permutations through which the rover design algorithm must iterate, with a corresponding increase in required computational resources. Model fidelity can also be increased at the cost of computational resources. Regardless of these recommendations, the Rapid Modeling of Mars Robotic Explorers software tool provides an efficient and effective means to evaluate and compare rover architectures for the design of future Mars missions.

2.0 Project Overview

2.1 Introduction

The Mars Exploration Program, managed by the Jet Propulsion Laboratory (JPL), is a science-driven program that seeks to understand whether Mars ever was or could be a habitable world. To answer this question, scientists are trying to understand how geologic, climatic, and other processes interact, shaping the Martian environment over time. Engineers are working on designing and validating advanced technologies that will enable us to explore Mars in ways we never have before, with the goals of higher-resolution images, precision landings, longer-ranging surface mobility and even the return to Earth of Martian soil and rock samples for studies in terrestrial laboratories.

2.1.1 Project Background

There is a wide variety of existing and developing technologies available for designing near-term Martian surface exploration vehicles. A tool to compare the benefits and costs of different rover designs incorporating these different technologies will facilitate the conceptual design and architecture selection of future rover missions. Such a tool should incorporate multiple options for subsystem components, including future technologies and components planned for future exploration missions. The tool should also be easily expandable to allow for new technologies to be added as they are conceived and created. In this way the tool will continue to be useful in comparing potential surface exploration rover missions.

2.1.2 Motivation

The motivation for this project is the need for an efficient approach to comparing potential Mars rover architectures and high-level designs early in mission planning. The tool creates a trade space of feasible rover designs specified by combinations of varying design parameters. Example parameters include different science instruments and acquisition tools, landing sites, levels of autonomy, types of communication systems, power source options and mission lifetime. The trade space created can then be analyzed to show the costs and benefits of different architectures and technologies. Such analysis can help in mission design and planning, as well as in technology funding and research funding. The trade space can also be analyzed to see how changing one or more design factors influences the cost and capabilities of the design. For example, this analysis could be used to show the value in funding the development of higher autonomy algorithms or in using a nuclear power source.

2.1.3 Objectives

The main objective of this project is to create a tool for JPL that can be used for creating system level designs of near future Mars surface exploration missions. This tool generates many different rover designs based on specified design parameters and compares the results to see trends in cost and benefits among the different designs generated.

In order to be useful the tool must be validated. Demonstrating that the produced designs are consistent with existing Mars rover designs is a means for validating the tool. The main data source for this validation comes from the MER rovers scheduled to launch in 2003. While some

preliminary data is available from the MSL rover design, it is hoped that this tool will actually be used to aid in the design process for the MSL mission.

Single axis trends in the rover design space are also used to validate the tool. Analyzing the single axis trends of different design parameters shows how the rover designs change when the value of a single design parameter is varied. This validates the tool by showing that the rover designs change as expected with respect to the design parameter being varied.

2.2 Requirements

2.2.1 Project Scope

In order to complete the project in one semester it was necessary to limit the scope of the tool that would be produced. Early in the project it was decided that this tool would only address the environmental constraints that relate to Mars surface operations. This limitation means that design considerations and constraints related to the launch, entry, descent and landing mission phases are not included in the existing tool. Some discrepancies in rover designs that the tool produces compared to actual rover designs may be attributed to the exclusion of these mission phases.

The project scope was also limited to modeling only single rover missions. Some extrapolation to multiple rover missions is possible. This can be done by using multiple copies of a single rover design or choosing multiple rover designs in the trade space to combine into one mission. This is not currently implemented in the tool, but could be done by hand using trade space results. It should be noted, however, that not all costs or benefits scale in the same manner between different rover designs, or even for multiple rovers of the same design. This needs to be understood and accounted for before multiple rover missions can be accurately analyzed from the trade space.

Another modeling constraint imposed on the tool is that it does not consider an active lander. An active lander could drastically change the characteristics of a system and the cost and benefits of the rover mission. Active landers have many possible functions including a communications system relay, a processing center or even a power source. Limits on time and other resources place an active lander outside of the scope of this project.

2.2.2 Project Requirements

Requirements for this project focused on the usability, maintainability and expandability of the tool itself. This meant having requirements that focused on the development of the code for the tool. Examples of some of the requirements established for the tool's generation include:

- Algorithms will be implemented using MATLAB
- The code will be kept as modular as possible to make it easy to read, revise, maintain and hand over to a different user who is not part of the tool creation process
- Module headers will include references for the modeling performed in that module
- Module headers will include a complete list of inputs to that module and what module they come from
- Module headers will include a complete list of outputs of that module and what modules need those outputs

-
- All constants used in a module, either scientific constants or constants from particular equipment or technologies, will be listed immediately after the module header so that changes due to equipment or technology changes can be made easily
 - All coding will use meaningful variable names to aid readability of the code
 - All modules will contain descriptive comments so that a reader can understand the flow of the module and integrated tool

2.3 Project Overview

2.3.1 System Architecture

At the beginning of the course, students were split into teams of two and assigned to the following subsystem groups: Instruments, Acquisition, Environment, Rover, Power, Communications, and Autonomy. Throughout the course of the project, students took turns being part of the systems team. The systems team worked on system integration, testing and validation of the tool.

Each subsystem team produced a modular function implementing a model of the subsystem. The algorithms in the subsystem modules take inputs from other modules and from the Science and Design Vectors, which define the scope and properties of a particular trade space. The Science Vector defines the landing site characteristics and the payload, consisting of science and navigation instruments and acquisition tools. The Science Vector is held constant for all rover designs in a particular trade space. The Design Vector uniquely defines individual rover designs. Each rover design is created based on a particular value for each of several design parameters such as wheel diameter, type of power system, and capability for autonomous operation. The user controls the scope of the trade space by defining the sets of allowable values for the parameters in the Design Vector. The Constants module contains constants needed by multiple modules for module-level design. The Constants module addresses low-level design relationships, while the Science and Design Vectors address high-level architectural choices. Additional modules such as Cost and Complexity are used for analysis of rover designs.

A brief description of the Science Vector, Design Vector, and Cost and Complexity modules is given here, while the subsystem modules are described in detail in separate chapters in Section 3.

2.3.1.1 *Science Vector*

The user determines values for the parameters in the Science Vector. These values are held constant for all the designs created in a particular trade space. The default values selected for the Science Vector correspond to the MER mission. The parameters in the Science Vector are:

- Science instruments
- Acquisition tools
- Navigation instruments
- Number of samples per instrument per site
- Average separation between sites
- Average site diameter
- Percentage of rock coverage expected at the landing site
- Latitude type of the landing site
- Hemisphere of landing site

-
- Areocentric longitude of Mars at landing
 - Ratio of traversable obstacle height to wheel diameter

2.3.1.2 *Design Vector*

The Design Vector is used to uniquely define a rover design. The parameters in the Design Vector were chosen because they constrain the rover design and are meaningful for rover design analysis. The user determines the scope of the trade space by choosing the allowable values for each parameter in the Design Vector. The tool iterates over all possible combinations of the values of the Design Vector parameters to create a trade space of rover designs. The parameters in the Design Vector are:

- Mission lifetime
- Type of power system
- Type of communications system
- Long distance navigation autonomy level
- Short distance navigation autonomy level
- Localization autonomy level
- Acquisition autonomy level
- Rover wheel diameter
- Presence of an active lander (currently this option can only be set to “no”)

2.3.1.3 *Cost Module*

The cost module consists of two main parts - subsystem costing and operations cost. The subsystem cost is the sum of the costs of the individual subsystems. The Communications, Power, Acquisition, and Instruments modules directly output costs, while the Rover subsystem cost is estimated within the cost module. Unfortunately, no cost models or relationships exist for the development and testing of autonomy. Therefore, the cost of increasing autonomy was left out of the cost module. It should be noted that this lack of autonomy cost might skew results towards reporting that more autonomy is preferable since these architectures should be more productive with no additional cost.

The Communications subsystem team used Wertz and Larson's *Space Mission Analysis and Design (SMAD)* [4] and NASA references for their cost model, which is a function of the usage of the Deep Space Network. Acquisition reported actual costs suggested from their references when available and estimated costs when these actual costs were not available. Instruments used a blanket assumption to cost each instrument, based on information and rules of thumb determined during science meetings at JPL. The Power subsystem used several references. While SMAD power-cost relationships are available, these relationships focus on low Earth orbiting satellites and were therefore only used to a limited extent. Additional relationships were found in AIAA papers and through discussions with Joe Parrish, president of Payload Systems, Inc. The Power subsystem costs include costs of solar and RTG power sources. Finally, the cost of the Rover subsystem is correlated directly to the mass of the rover.

Originally, four data points were used to test costing relationships. These data points were for Pathfinder, two different grass-roots preliminary estimates for MSL, and an estimate for the Mars Geological Rover (MGR). Allen Chen, of JPL, provided the estimates for Pathfinder and MSL. The estimates for MGR were taken from two presentations found on the JPL web site. All of these estimates should be considered preliminary and their use should be limited. Using these four data points, a cost relationship from SMAD was tested. Due to the fact that this relationship is applicable to low Earth orbiting satellites, and not Mars rovers, it was not surprising that the cost

estimates given by the SMAD function were far off from the test data point cost estimates. Since no known relationship is available for costing a Mars rover based on the mass of the rover, the cost team attempted to fit a cost function to the four data points available. An exponential function fit the four points well.

Late in the project a fifth data point became available. This data point is a third grass-roots estimate for MSL, provided by Allen Chen. This fifth data point showed that the exponential costing relationship derived previously was good only for the four previous points, and did not accurately predict the cost of all rovers. While another exponential relationship may have been able to be fit to these five points, this relationship would again change significantly if a sixth data point were to be added. It was decided that there are not enough data points available to derive a function relating the cost of a rover to its mass with accurate statistical relevance. While it is known to be highly inaccurate and meant for satellite design costing, a very different type of system, the SMAD relationship, shown in Equation 1, was chosen to relate the cost of the rover to the mass of the rover, since this relationship is a known cost relationship. This model should be examined in any future use of this tool. The comparison of the actual data points, the SMAD estimates, and the exponential function estimates can be seen in Figure 2.3.1. It should be noted that the SMAD cost relation estimates costs in 2000\$k.

$$C = 781 + 26.1m^{1.26} \tag{2.3.1}$$

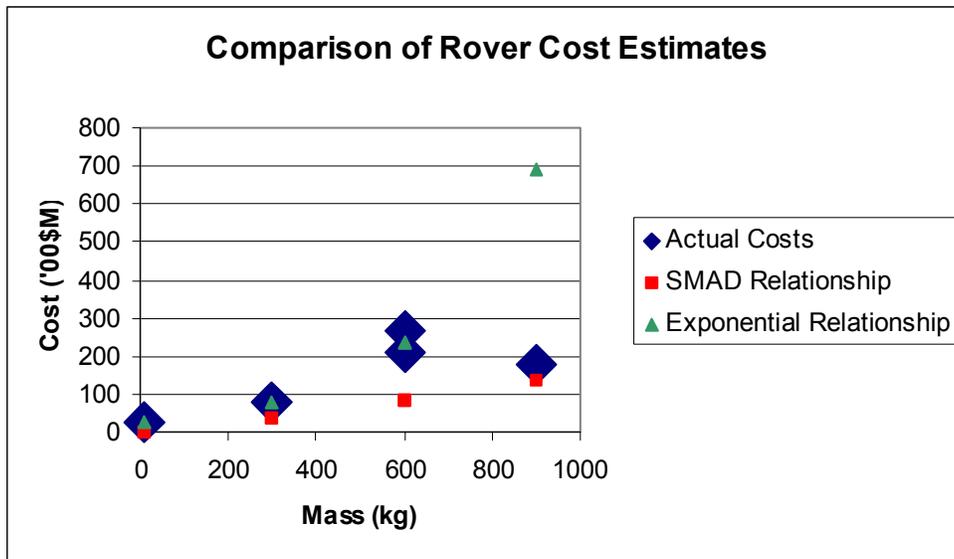


Figure 2.3.1: Comparison of cost estimates

The final cost modeled in the cost module is the operations cost. The operations costs were modeled as a function of lifetime. Operational costs for the rovers modeled in the tool were linearly scaled based on Pathfinder lifetime operational costs. The Pathfinder operational costs were taken from the Goddard Space Flight Center website. This simple model should again be examined in detail before any further analysis is conducted using the cost module.

2.3.1.4 Complexity Module

The purpose of modeling complexity is to provide the user with a relative comparison of the technology requirements for different rover designs. While it is not expected that the user will

always be able to gain conclusive information on which missions are “too complex”, the user will be able to understand how adding technology effects mission productivity.

In order to rank rover design complexity, each design is given a calculated complexity ranking between 1 and 4, with the 2003 MER mission having a technology complexity of 1, and with a highly complex mission having a value of 4. This complexity number is the average of complexity values for individual design elements, such as the Power or the Communications subsystems, which are each assigned a rank of 1, 2, 3 or 4. These complexity numbers represent technology development stages for planetary exploration technology, and a description of each level is shown in Table 2.3.1. The complexity levels are similar in concept to the NASA Technology Readiness Levels (TRLs), with specificity to planetary exploration. Making the levels specific to planetary exploration allows separating proven non-planetary surface technologies from similar technologies not yet used for planetary exploration (e.g. RTGs). The use of four levels was preferred to the full nine NASA TRLs because information on the readiness of most design elements was not detailed enough to make assignments to more than three or four general categories.

Table 2.3.1: Complexity Levels

Level	Description
1	Flight proven on a lander/rover, or will be flown in 2003
2	Validated on a lander/rover in a relevant environment
3	Analytically demonstrated or lab tested on a lander/rover
4	Technology concept formulated

After defining the complexity levels, assignments were made for each of the different options for each design element. For example, the Power subsystem has the option of solar power or nuclear power. The solar option is assigned level 1, since it is a well proven technology, while the nuclear option is assigned level 3, since nuclear power sources for rovers are only slightly more advanced than that defined for level 4. The lifetime design element is another example. Missions over 90 sols have not yet been achieved, so they are ranked level 2. The lifetime definition can become more granulated, with more levels assigned; once more information is available on how the complexity increases with certain intervals of mission lifetime. A summary of the assignments is presented in Table 2.3.2.

Table 2.3.2: Complexity Level assignments for design elements

Design Element	Option	Level	Option	Level
Power	Solar	1	RTG	3
Telecom	DTE	1	Other	2
Autonomous Long Distance Navigation	A1	1	A3	3
Autonomous Short Distance Navigation	A1	1	A3	3
Night Navigation (not currently modeled)	No	1	Yes	4
Autonomous Sample Collection	A1	1	A3	4
Night Instrument Processing	No	1	Yes	2
Lifetime	< 90 sols	1	> 90 sols	2

For each rover design produced in the trade space, these levels are all averaged to obtain a total complexity value for that particular rover design. An example mission with various design

options that average to a total complexity of 2.3 is shown in Table 2.3.3. Again, the values for each rover design option are assigned so that the 2003 MER mission design serves as a technology baseline with a total complexity equal to 1.

Table 2.3.3: Example Mission

Design Option	Level
RTG Power	3
DTE Telecomm	1
A1 Long Dist. Navigation	1
A3 Short Dist Navigation	3
Autonomous Sample Collection, Yes	4
Night Instrument Processing, Yes	2
300 sol Lifetime	2

→ Total complexity = average = 2.3

The complexity module easily accommodates changes in the complexity level rankings for design elements, or the addition of completely new design elements. An example of changing the level rankings would be assigning increasingly higher complexity levels for increasingly longer mission lifetimes. The complexity rankings could be changed to incorporate a political complexity value as well. Such a rating could be used to demonstrate how difficult it would be to ensure a nuclear power source is acceptable to the public.

2.3.2 System Design Flow

Once the inputs and outputs the different subsystem modules required were determined, an N-Squared diagram was created. The N-Squared diagram, Figure 2.3.2, shows the order in which the modules are executed within the tool.

Design Vector				X	X	X	X	X	X	X	X
	Science Vector	X	X		X	X	X	X		X	X
		Instr.	X	X		X			X	X	
			Acquisition	X		X		X	X	X	
				Rover Init.	X	X		X	X	X	
					Environ.	X		X			
						Auto. Init.	X	X	X	X	
							Comm. 2	X	X		
								Rover	X	X	
								X	Power	X	
										Auto.	X
											Comm. 3

Figure 2.3.2: N-Squared diagram for the rapid modeling design tool

Modules are executed along the diagonal. Any boxes above the diagonal that are checked refer to inputs received from a module that has already been executed. Boxes checked below the diagonal refer to loops between modules. The N-Squared diagram was manipulated to minimize the number of loops, in order to create a simpler and more efficient program.

As can be seen in Figure 2.3.2, there is one major iterative loop between the Rover and Power subsystems. This loop is due to the Power module needing to know the drive power of the rover in order to determine the size of the power system, while the Rover module needs to know the size of the power system in order to determine the power required to drive the rover.

The N-Squared diagram also shows how the Communications subsystem is split into three separate modules. The Communications 2 module is executed first to determine the size of the communications system before the Rover-Power loop executes. The Communications 1 module is executed within the Autonomy module to determine the communications delay for sequences of actions. The Communications 3 module is used last to determine the cost of the communications system.

Figure 2.3.3 shows the project flow model for the rapid modeling tool and how the trade space is generated. The trade space contains a collection of rover point designs, developed by iterating over combinations of the allowable values of the parameters in the Design Vector. For a particular iteration step, the parameter values in the Design Vector are held constant, and these values and those in the fixed Science Vector are passed as inputs to the rover design tool. Assuming a properly chosen set of parameters in the Design Vector and an accurate rover design algorithm, there is a one to one mapping between the Design Vector and a particular point design in the trade space.

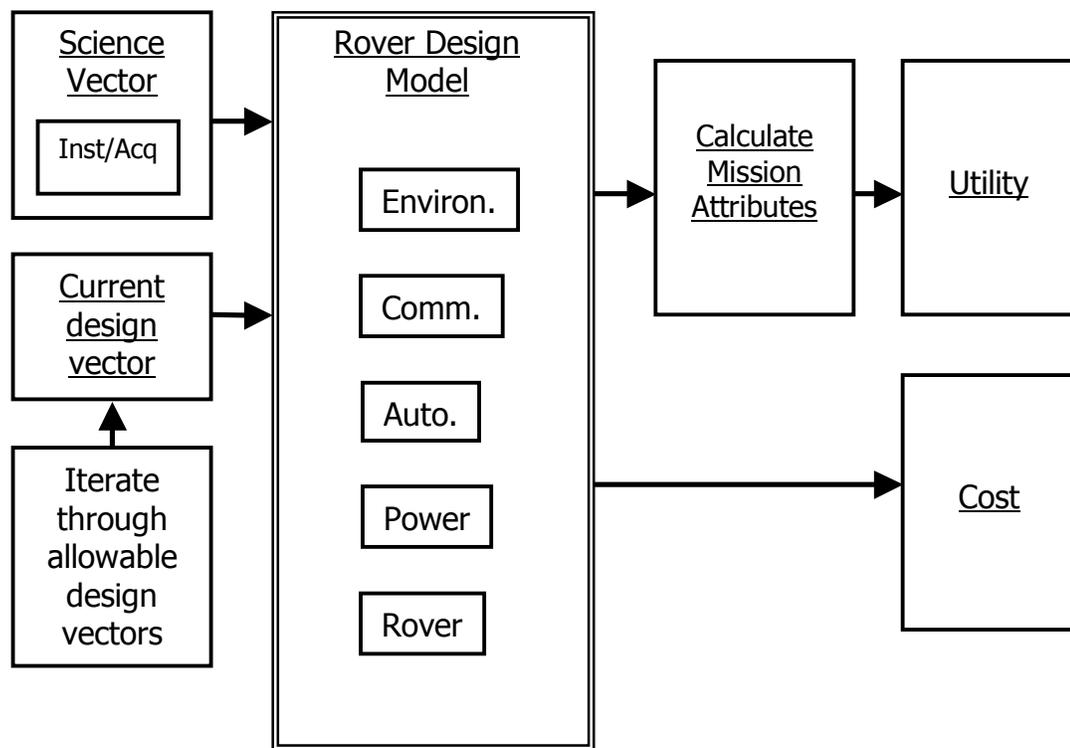


Figure 2.3.3: Project flow model for the program

The master code of the rover design model contains a set of nested loops, where each loop represents a different parameter in the Design Vector. In this way, all possible combinations of values for the Design Vector parameters are used, resulting in the evaluation of all possible rover configurations.

Figure 2.3.4 shows the execution sequence of the modules within the rover design model. The outermost loop is for the variable representing the wheel diameter of the rover. Before continuing to the other loops for the other parameters in the Design Vector, two modules are executed: the Rover Initialization module and the Environment module. These modules are executed here because the only Design Vector parameter required for their evaluation is the wheel diameter, and executing these modules in the outermost loop reduces computation time.

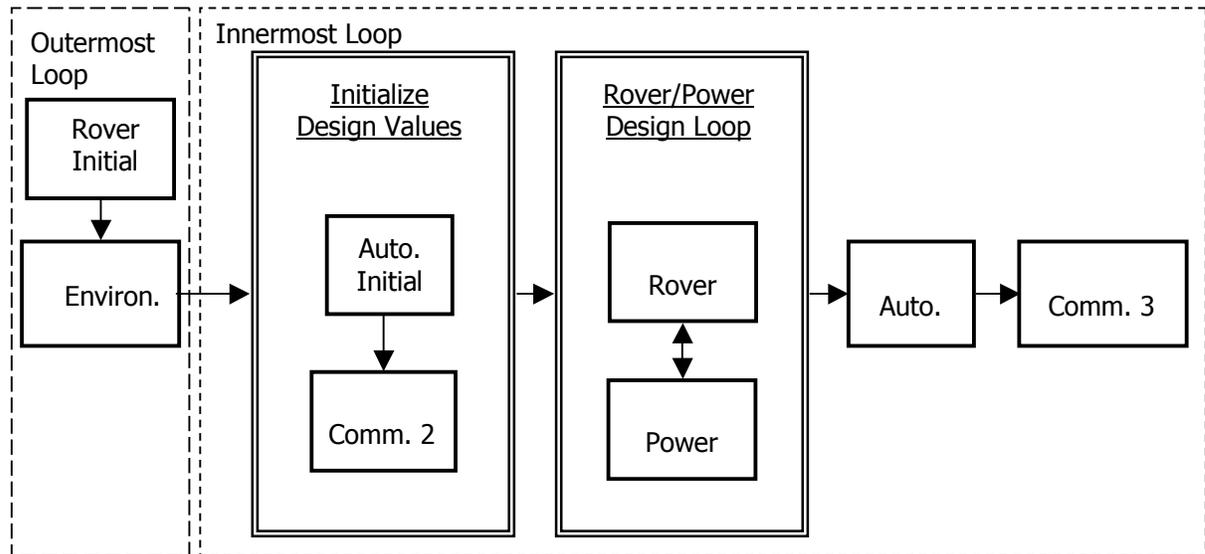


Figure 2.3.4: Rover design model details

The rest of the subsystem modules are executed within the innermost loop of the master code since they will differ for every unique set of values in the Design Vector. More initialization modules are executed first to determine system sizing before entering the Rover/Power module iterative loop. Once this iterative loop has finished, the Autonomy module is executed to determine its values based on the actual rover and power designs. Finally the Communications 3 module is called to determine the cost of the communications system, including Deep Space Network (DSN) operation costs and equipment costs.

3.0 Project Subsystems

3.1 Instruments

3.1.1 Responsibilities

During the design process, the decision was made to incorporate the scientific payload of the rover into the design through the Science Vector, not the Design Vector. In other words, the user chooses a particular suite of scientific instruments and the tool then designs rovers around that constant payload suite. Therefore, the purpose of the Instruments module is to give the user of the tool a reasonable set of instruments to choose from, to be modular enough to allow the user to add additional instruments if desired, and to provide information on the chosen instruments to the other modules of the tool. The set of information (mass, power, etc.) provided about each instrument directly imposes requirements on other subsystems.

3.1.2 Instruments Database

In the first step of the Instruments module the user chooses a set of unique science instruments from the GUI. The user also specifies where each instrument shall be located. For example, a user may choose to put a spectrometer in the contact suite or in the analytic laboratory or both, but each spectrometer is listed as a unique instrument. Next, the module retrieves data about each instrument that the user has chosen from the instruments database. Finally, this information is passed on to the other subsystem modules that have requested it. Therefore, the main component to the Instruments module is the database of information regarding each instrument.

The set of information required for each instrument in the database was determined from the requirements of other subsystems and modules. The mass, dimensions, temperature requirements, and location of each instrument are output to the Rover subsystem in order to size the mobility system. The day and night power requirements of each instrument are sent to the Power subsystem. The amount of data recorded per sample for each instrument is sent to the Communications subsystem. The time required to examine each sample and whether or not the instruments can be used while the rover is moving is sent to the Autonomy subsystem. Finally, an estimate of the cost of each instrument is output to the Cost module. A sample entry from the instrument database can be seen in Table 3.1.1, and the full database can be found in the Appendix.

Table 3.1.1: Sample entry in Instrument database.

Instrument Name	Mass	Power (day)	Power (night)	Size (X)	Size (Y)	Size (Z)	Cost
	kg	W	W	m	m	m	\$M
CONTACT SUITE							
Alpha Particle X-ray Spectrometer	1.02	0.34	0	2.60	1.35	0.60	10
	Sample Size (max)	Sample Size (min)	Max number samples	Data per sample	Time per sample	Temp. (min)	Temp. (max)
	m	m	-	bits	s	K	K
Alpha Particle X-ray Spectrometer	N/A	N/A	N/A	256000	43200	173.15	318.15

3.1.3 Sources

The main source of information for populating the instrument database was the Mars Science Laboratory (MSL) instrument database. This database has been put together by members of the MSL science team in preparation for choosing the final payload for the MSL mission, as well as for the purpose of evaluating sample mission scenarios. The Instruments team was given the MSL instruments database by one of the MSL systems engineers, Allen Chen, of JPL. However, it should be noted that the version the Instruments team used was required to be ITAR cleared, and therefore may have been missing some information that is considered known to the MSL community.

While the MSL instruments database was exceptionally useful and provided a very large amount of information, it did not provide all the information needed for most instruments. Therefore, the Instruments team researched other sources of information. Additional information for several instruments was found on the Internet. In most cases, this information was found on the instrument manufacturer websites. When information was gathered from sources other than the MSL instrument database it was noted both in the comments of the instrument module code as well as in the Excel database of instrument information.

3.1.4 Assumptions

Occasionally it was not possible to find all the information required for an individual instrument through either the MSL database or the Internet. When this was case, assumptions were made to estimate the data required. If the missing data for the instrument was temperature requirements, sample maximum and/or minimum dimensions, or the maximum number of samples the instrument can handle, the information was estimated based on the other instruments in the same location (remote sensing, contact suite, or analytic laboratory). If the instrument was missing the time required to analyze a sample or the amount of data recorded per sample, the information was estimated based on instruments with similar functions. In addition, in some cases default information was filled in if the specific information for that

instrument was not known. For example, one default was to place no limit on the number of samples that an instrument could process or the size of the sample obtained. Finally, the cost for all instruments was set at a default estimate of \$10M. This estimate came from comments made in several MSL meetings attended by the Instruments team. This is a very rough estimate and there is no distinction made, based on cost, from one instrument to another. This assumption should be examined further in any future work regarding this tool.

3.1.5 Validation

Once the database was populated, the information in the database was checked against rules of thumb provided by Charles Whetsel. It was noted that all the mass numbers from the database were consistently lower than those numbers provided by Whetsel. The comparison between the rule of thumb numbers and the database numbers can be seen in Table 3.1.2. Note that all mass numbers from the database are lower than those from Whetsel. However, if the database numbers are rounded up to the nearest 5 kg, they match exactly with those provided by Whetsel, with the exception of only one instrument. Therefore it was assumed, and later confirmed by Whetsel, that when giving mass estimates Whetsel was simply being conservative. There were two major reasons for this conservatism. The first is that the mass numbers given in the database account for only the instrument itself, and do not account for extra structure, cabling, etc. that is required to support the instrument. Secondly, the estimates given in the database were, in several cases, obtained from the instrument developers. Therefore, it is assumed that these estimates are optimistic in nature. Due to these two issues placing concern on the validity of the mass numbers in the database, it was decided to add an extra factor to the mass numbers reported to the other subsystems from the Instruments module. A factor of two was used to assure conservatism in the estimates provided. This constant factor is set at the beginning of the module code and can be easily adjusted in the future if more information becomes available.

Table 3.1.2: Comparison between database mass values and Whetsel estimates.

	Remote Sensing		Contact Payload Arm + Instruments (kg)	Analytic Lab		
	Imager (kg)	Spectrometer (kg)		Chemistry (kg)	Mineralogy (kg)	Oxidation (kg)
Whetsel Estimate	5	10	10	10	5	5
Database Estimate	0.35	3.19	9.231	7.1875	1.725	1.5
Difference	4.65	6.81	0.769	2.8125	3.275	3.5
Database rounded up to nearest 5 kg	5	5	10	10	5	5
Comments	Stereo Panoramic Imager	IR Spectrometer	Includes arm that can accommodate a scoop	Pyrolysis Oven, Mass. Spect., XRD, XRF (average)	Raman, Mossbauer (average)	OEI

3.1.6 Current Instruments

The NASA Mars scientific exploration strategy is to “follow the water,” with specific goals related to the search for ancient or extant exo-biology, the study of the climate, and examination of the Martian geology (Parrish, 2003). Many scientific instruments are necessary to achieve these goals. The Instruments module currently includes 15 such instruments. Some of these instruments are stand-alone, while others are combination packages of many instruments. Fundamentally these instruments are designed to obtain the measurements necessary to achieve

- **Microscopic Color Imager, 30 micron resolution**

The Microscopic Color Imager enables the characterization of specimen surface properties, the determination of geological structure and layering, as well as the possibility for direct exo-biologic discovery.

3.1.6.2 *Remote Sensing Instruments*

Remote sensing instruments work by sensing the environment surrounding the rover. It is not necessary to physically obtain specimens for the purposes of these instruments. However, the range of these instruments is often limited to the nearby surroundings of the rover due to power and field of view/range limitations.

- **Point Infra-Red Spectrometer**

The Point Infra-Red Spectrometer is used to determine the chemical composition of a specimen based on the spectrograph of samples illuminated with Infrared radiation.

- **Stereo Panorama Camera**

The Stereoscopic Panoramic Camera acquires spatial data for rover navigation. Its scientific uses include the characterization of the Martian topography. This can be used to aid the statistical presence of certain rock sizes and shapes, as well as the frequency of certain types of rocks that can be identified visually (igneous, sedimentary, etc.)

- **Laser Induced Breakdown Spectrometer (LIBS)**

The Laser Induced Breakdown Spectrometer is used to determine the chemical composition of a specimen. The LIBS laser pulses upon the surface of a specimen. This induces the creation of ionized gasses, which can then be spectroscopically analyzed.

3.1.6.3 *Analytic Laboratory Instruments:*

The analytic laboratory instruments are typically the most sophisticated instruments included on the rover package. Many of these instruments have smaller and less-capable counterparts that may be placed on the contact suite. The analytic laboratory instruments require an acquisition package to obtain and prepare a specimen for analysis.

- **Microscopic Imager, High magnification**

The Microscopic Color Imager enables the characterization of specimen surface properties, the determination of geological structure and layering, as well as the possibility for direct exo-biologic discovery.

- **Moessbauer Spectrometer**

The Mossbauer Spectrometer employs Gamma Rays to stimulate the surface of a specimen. The resulting spectrogram enables the chemical composition of the specimen to be determined.

- **Oxidation Effects Instrument**

The Oxidation Effects Instrument enables the chemical and electro-magnetic properties of a specimen to be determined. This is achieved by analyzing the spectroscopic signature of the ionization states, which are induced by electrostatic charges or a chemical reagent.

- **Pyrolysis Oven with Amino Acid Detector**

The Amino Acid Detector instrument enables the chemical detection and characterization of amino acids by heating and annealing a specimen in the Pyrolysis Oven, and then analyzing the evolved spectroscopic signatures.

- **X-Ray Florescence (XRF)**

The X-Ray Florescence instrument is a specific type of LIBS where the laser employed is an X-Ray laser. It is used to determine the chemical composition of a specimen. The X-Ray laser pulses upon the surface of a specimen. This induces the creation of ionized gasses, which can then be spectroscopically analyzed.

- **Gas Chronograph Mass Spectrometer + Laser Diode**

The Gas Chronograph Mass Spectrometer analyzes chemical composition by illuminating a specimen while tuning a laser diode to various frequencies and measuring the absorption of those frequencies.

- **Gas Chronograph Mass Spectrometer + Evolved Gas Analysis**

The Evolved Gas Analysis / Gas Chronograph Mass Spectrometer measures chemical composition by heating and annealing samples and then analyzing the chemical spectrograph of the evolved gas.

- **Raman Spectrometer**

This Raman Spectrometer characterizes the backscatter of a laser illumination to determine both qualitative and quantitative information about a specimen. The spectral signature of the specimen enables the quantification of the elements and compounds present, while the nature of the backscatter enables qualitative features, such as shape and texture, to be determined.

3.1.7 Expandability

The data from the instrument database was entered into the Instruments module through a Matlab subroutine. The information for each instrument is stored in a structure. The code creates a list of instruments based on the user input and then outputs the resulting array structure for the other modules to call. Therefore, the process of introducing a new instrument is very simple. The user simply needs to open the Instruments module code, copy the last instrument data set and paste below the last instrument. The data for the previous instrument can then be replaced by the data for the new instrument. This process can be repeated as many times as necessary. In addition, all database entries are clearly labeled, such that if more up to date information about a given instrument were to be made available, the code could be updated easily and quickly.

3.1.8 References

1. Ludwinski, Jan. *Mars Exploration Rover (MER) Project: Mission Plan*. JPL Document. JPL D-19659. April 24, 2003.
2. *Mimos II* [online]. Available at: <http://www.tu-darmstadt.de/fb/phys/ikp/mb/mimos2/mimos2.htm>. April, 2003.
3. Parrish, Joe, President, Payload Systems Inc. Lecture to MIT 16.89 Graduate Space Systems Engineering Class. Cambridge, MA. February, 2003.

-
4. Tamppari, Leslie. *PSIG_Instruments_summary_10Mar03.xls*. JPL Internal Document, Preliminary. March, 2003.
 5. USGS. *Astrogeology Research Program* [online]. Available at: http://astrogeology.usgs.gov?Projects?MER-AthenaMI/microscopic_imager.html. April, 2003.
 6. XRF Cooperation. *Products* [online]. Available at: <http://www.xrfcorp.com/products/>. April, 2003.

3.1.9 Appendix

Instrument Database

Instrument Name	Mass	Power (day)	Power (night)	Size (X)	Size (Y)	Size (Z)	Cost	Sample Size (max)	Sample Size (min)	Max number samples	Data per sample	Time per sample	Temp. (min)	Temp. (max)
	kg	W	W	m	m	m	\$M	m	m	-	bits	s	K	K
CONTACT SUITE														
Alpha Particle X-ray Spectrometer	1.02	0.34	0	2.60	1.35	0.60	10	N/A	N/A	N/A	256000	43200	173.15	318.15
Mössbauer Spectrometer	1.21	3.4	0	2.64	1.35	0.57	10	N/A	N/A	N/A	12000	43200	168.15	323.15
Raman Spectrometer - In-situ Remote Sensing	3	10	2	0.38	0.45	0.25	10	N/A	N/A	N/A	40000000	7200	148.15	338.15
Microscopic Color Imager, 30um resolution	0.22	4.8	0	0.12	0.13	0.08	10	N/A	N/A	N/A	424000 ³	600	153.15	153.15
REMOTE SENSING														
Point IR Spectrometer	3.1	6	5	0.22	0.16	0.10	10	N/A	N/A	0	256000	3600	241.15	323.15
Stereo Panorama Camera, 4 color + Calibration target	0.35	2.4	2.7	0.14	0.13	0.11	10	N/A	N/A	0	48000000	6600 ⁴	153.15	343.15
LIBS	4	5	0	0.35	0.25	0.45	10	N/A	N/A	0	18000000	120	203.15	313.15

Instrument Name	Mass	Power (day)	Power (night)	Size (X)	Size (Y)	Size (Z)	Cost	Sample Size (min)	Sample Size (max)	Max number samples	Data per sample	Time per sample	Temp. (min)	Temp. (max)
	kg	W	W	m	m	m	\$M	m	m	-	bits	s	K	K
ANALYTICAL LAB														
Microscopic Imager VIS, high magnification - 1 um pixel, 6-color	3.0	5	0	0.13	0.13	0.28	10	[TBD] (0)	[TBD] (0)	N/A	424000 ³	600	153.15	343.15
Moessbauer Spectrometer [6]	0.45	1 ¹	0	0.06 ¹	0.04 ¹	0.03 ¹	10	N/A	N/A	N/A	1280000 ⁵	43200 ⁵	168.15	323.15
Oxidation Effects Instrument [7]	1.5	1	0	0.06	0.07	0.02	10	<100 microns	<100 microns	N/A	1000000	86400	228.15	333.15
Pyrolysis oven integrated w/ GC/MS, amino acid detector	8.0	15	0	0.42	0.27	0.215	10	0.001	0.000	N/A	1000000	1800	228.15	333.15
XRF	1.5	3	0	0.254 ²	0.09 ²	0.03 ²	10	N/A	N/A	N/A	27000 ²	1200	228.15	333.15
Mass Spectrometers: GCMS + LD-TOF Integrated Instrument Pkg	18	30	0	0.77	0.46	0.62	10	0.001	0.000	N/A	16000000	3600	223.15	238.15
Mass Spectrometer: GCMS	14	30	0	0.3	0.25	0.3	10	0.001	0.000	150	16000000	3600	223.15	238.15
Raman - Analytical Lab	3	10	2	0.38	0.45	0.25	10	N/A	N/A	N/A	40000000	7200	148.15	338.15

Database Key

*Stability: 1 = Sample can be processed while rover is moving; 0 = Rover must remain stationary for the duration of instrument operations

Black Text: Information based on JPL MSL Instrument Database

Blue Text: Information from JPL notes other than MSL Instrument Database

Red Text: Information not readily available, but based on similar instruments (see code comments)

References

- 1: From <http://www.tu-darmstadt.de/fb/phys/ikp/mb/mimos2/mimos2.htm>
- 2: From <http://www.xrfcorp.com/products/crfpb.html>
- 3: From http://astrogeology.usgs.gov/Projects?MER-AthenaMI/microscopic_imager.html
- 4: From MER Mission Plan pg. 131
- 5: From MER Mission Plan pg. 132

3.2 Acquisition

3.2.1 Responsibilities

The acquisition module provides technical information about current acquisition tools that can be used for Mars sample collection. The information, such as power requirements, mass, and dimensions of each tool, is contained in a Matlab file look-up table. This table is used to generate outputs to other modules. Eight sample acquisition tools and a component for processing samples, the Sample Processing and Handling Hardware (SPaH), are currently modeled.

3.2.2 Assumptions

The information for the tools was compiled from modern planetary sample acquisition tools that are flying on missions or are in advanced development. Future technologies for sample collection, such as deep drilling machines, are not modeled because few complete designs exist for such hardware. Acquisition tools are considered to be any hardware that prepares or collects samples from the environment. For example, a rock abrasion tool that exposes unweathered rock surfaces is considered an acquisition tool, even though it does not actually acquire a sample. The SPaH system is also considered an acquisition tool because it prepares samples for the rover instruments. The Autonomy module models cameras that aid in the acquisition of samples, such as a panoramic camera, or "pancam". Cameras that collect mainly scientific information, such as the head of an IR spectrometer, are considered instruments, and modeled by the Instruments module. Anytime a robotic arm is required to support an acquisition payload, it is modeled by the Rover module, and sized appropriately for the rover.

Additionally, the Acquisition module assumes that the user is knowledgeable about the use and characteristics of surface science instruments and the associated acquisition tools that they require. For example, if the user selects a mass spectrometer as an instrument, it is likely that some kind of rock or regolith sampling tool will be required. However, the user is free to choose only contact acquisition tools, such as a rock abrasion tool and a magnet, without returning an error flag. One problem with implementing automatic tool selection is that some instruments analyze a variety of sample types, and some acquisition tools are multifunctional. For example, a mass spectrometer can accept samples of rock, soil and the atmosphere, and a subsurface mole can support *in situ* Raman spectroscopy as well as soil collection for a mass spectrometer.

3.2.3 Background

A variety of sources were used to collect characteristics of each tool, including industry, journal articles, NASA contacts, and DLR-Germany. By including only fully developed tools in the database and going directly to manufacturers for information, generalizations and guesses for critical design parameters were usually avoided. For the SPaH, which processes and delivers samples to analytic lab instruments, a scaling relationship was established based on a design from the JPL Sample Processing and Distribution (SPaD) study [3].

Table 3.2.1: Acquisition Tools in the Database and Some Selected Properties

Tool	Manufacturer / Developer	Status	Main Source of Information	Size LxWxH (cm)	Mass (kg)	Avg. Power (W)
Rock Abrasion Tool	Honeybee Robotics	Flying on MER	PSIG document – JPL [13]	13.7 x 7.5 x 8.	.77	10
Rock Corer	Honeybee Robotics	Tested on FIDO rover	Journal of Geophys Res. [7]	29.8 x 14.51 x 96.4	2.7	30
Pluto Mole	DRL	Flying on Beagle-2	Lutz Richter, DLR [14]	36.5 x 2 x 2	.86	3
MUM Mole	NASA Ames	In development	LPSC XXXIII 1201 PDF [16]	50 x 4 x 4	3.5	10
1 meter drill	Honeybee Robotics	Tested in lab	Honeybee Robotics doc [8]	60 x 60 x 125	15	100
10 meter drill	Honeybee Robotics	Tested in lab	Honeybee Robotics doc [8]	77.7 x 58.3 x 210.5	99	100
Magnets	-	Flying on MER	MER webpage [11]	1.5 x 10 x 10	.056	n/a
Soil Scoop	-	-	JPL – Allen Chen [3]	10 x 6.0 x 4.0	.5	20
Sample Processing Hardware	JPL SPAD-study results	Concept design for '09 rover mission	JPL – Allen Chen [3]	.30 .30 .15	10	25

The eight tools contained in the database are shown in Table 3.2.1 with their manufacturers, current development status, main source of information, and some technical details. Both the Rock Abrasion Tool (RAT) and the Beagle-2 Pluto mole will be flying on Mars missions in 2003. Also included is a larger mole being developed at NASA Ames for the Mars 2009 mission, the MUM mole. The MUM mole, being larger, will be capable of delivering 5cc samples of soil compared to the Pluto mole's sample volume of 1cc [16]. The two subsurface drills included are modeled after Honeybee Robotics' one and ten-meter depth drills. The rock corer is also manufactured by Honeybee, and was tested in 2002 on the FIDO rover [7]. While a large amount of information was available for most of these tools, estimates were occasionally required for some missing design parameters. In these cases the estimates were based on similar tools, or values from a rover design summary document obtained from Charles Whetsel, JPL. Examples of tools that required some of these estimates are the scoop and the magnet. When multiple sources gave conflicting values, the most conservative value was always chosen. All sources of information are commented in the code next to the value.

Some design values were difficult to obtain from any document and had no analogous system available for comparison. In these cases, conservative estimates were made from basic principles. For example, the data rate required for the operation of the tools is estimated to be 1000 bits/s. This data rate should be adequate, since most tools have some level of autonomy. Another estimate is the number of command cycles required to obtain a sample once the rover is close to the sample of interest, which is based on MER level autonomy. The commands per sample range from 2 for the rock abrasion tool to 4 for the mole.

Sample processing hardware for the rover is modeled such that it is only required when a payload includes any analytic lab instruments, such as a mass spectrometer. Since no former planetary surface science mission has required autonomous sample processing of different sample types for multiple analytic instruments, there was not an advanced design to model the SPaH in the tool after. While the

Mars Polar Lander mission carried a soil sifter for its Thermal and Evolved Gas Analyzer (TEGA) instrument, the sifter was built on top of the instrument, rather than serving as a multifunctional and independent hardware package [17]. The Mars Beagle-2 lander carries a mass spectrometer capable of handling rock and soil samples, but little special preparation is required prior to heating the samples for study [1]. To accommodate a more advanced set of processing demands, the SPaH hardware in this code is modeled after a study conducted at JPL to design hardware that could provide diverse sample processing. The JPL SPaD study involves a tiered carousel type system capable of processing up to five different types of samples for up to four or more analytic lab instruments on the 2009 Mars rover mission [12]. In order to accommodate fewer analytic lab instruments, the Acquisition module scales the size, mass, power and cost of the SPaH hardware by $k/4$, where k is the number of analytic lab instruments that have been chosen by the user.

3.2.4 Program Flow

The rover design tool Graphical User Interface (GUI) is the only place that a user will interact with the acquisition tools. The user makes choices of acquisition tools at the same time as choosing science instruments for the rover payload. The user does not have the option to choose or not choose processing equipment because it is based on the number of analytical lab instruments that are chosen.

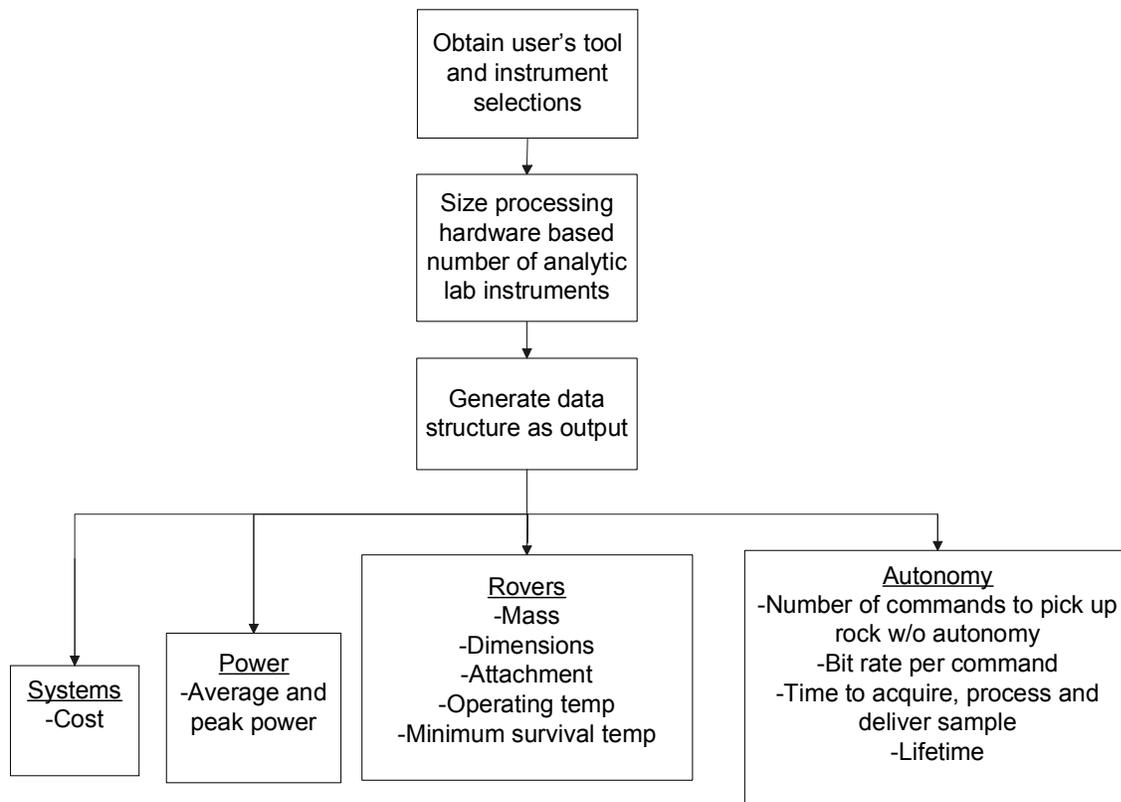


Figure 3.2.1: Flow of the Acquisition module from the user GUI to the outputs given to other modules

Once the user has made their acquisition choices and submitted the Science Vector, the tool accesses the acquisition.m file. Acquisition.m takes the user selections and locates the associated acquisition tool information from the create_acquisition_look_up_table.m file. This file is a list of structures, one per tool,

with all of the design parameters, such as power requirements and mass, which will be output to other modules for the rover design calculations.

Acquisition.m then looks at the 'location' parameter of each tool that has been selected by the user. The 'location' parameter places the instruments and acquisition tools somewhere on the rover such as on a mast, on a robotic arm, or on the rover body. If the location of an instrument is not 'arm' or 'mast', then it is assumed that the instrument is an analytical lab instrument. The Acquisition module then counts the number of analytical lab instruments that have been chosen. If no analytic instruments have been chosen, the Acquisition module outputs an empty array for the processing hardware. If the number of analytic lab instruments chosen is more than three, the full size processing hardware, based on SPaD, is assigned as the sample processing hardware output. If between one and three instruments are selected, the Acquisition module scales the processing hardware values based on the full size SPaD.

The relevant sample processing hardware values and acquisition tool values are then delivered in a structure that is accessible to all other modules, although most modules only use some of the data (e.g. the Rovers module uses the mass of the tool while Autonomy uses the data processing requirements).

3.2.5 Validation

Since the values obtained from each of the tools were usually from existing designs, the data is largely self-validated and contains no margins.

3.2.6 Expandability

There are a couple of ways that the Acquisition module could be improved and updated in the future. The module is limited by the information currently available about acquisition tools. Thus, the first improvement is keeping the module updated as new information and tools become available. For example, there might be data from Honeybee in a few years about their new Inchworm Deep Drilling System (IDDS), an autonomous un-tethered drill for subsurface sample collection.

The second place for improvement is regarding the sample processing and handling (SPaH) hardware. The model, currently based on JPL's SPAD study, is an estimate of overall mass and power values for a system that would 'include' whatever capabilities were desired. There are currently no individual SPaH capabilities modeled like sifters or crushers. If an accurate model of the Mars Polar Lander was desired, for example, it would be difficult to account for hardware such as the modular ovens used to heat samples.

Because the code does not model individual hardware components, the module assumes that the hardware necessary to meet the processing requirements for each of the selected analytical lab instruments is included within the SPaH output parameters. If enough information becomes available about individual processing components, then a SPaH system could be modeled that serves the specific instrument payload.

As mentioned in the background section, there is some ambiguity in the GUI that assumes the user is familiar enough with the instrument and acquisition choices that he or she can make reasonable choices that complement each other. It might be useful to have a window within the Science Vector that gives a short description of each acquisition tool (and instrument) to help users who may be less familiar with the scientific specifics or capabilities.

3.2.7 References

1. "Beagle2 Technology: System Overview." Beagle2 – a lander for the planet mars. <http://www.beagle2.com/technology/overview.htm>. Accessed May 14, 2003.
2. "Chemistry: Periodic Table: Magnesium: Physical Properties." Web Elements, 2003 www.webelements.com/webelements/elements/text/Mg/phys.html. Accessed May 14, 2003.
3. Chen, Allen. JPL. E-mail contact, April 8, 2003.
4. "Concepts and Approaches to Mars Exploration." Lunar and Planetary Science, 6105.pdf.
5. <http://www.lpi.usra.edu/meetings/robomars/pdf/6105.pdf>. Accessed May 14, 2003.
6. "Engineering the Future." Honeybee Robotics, Ltd. October 5, 2001. centauri.larc.nasa.gov/robot/hress01/RFI_responses/chapman.pdf. Accessed May 14, 2003.
7. "FIDO Science Payload Simulating the Athena Science Payload." Journal of Geophysical Research, Vol 107, NO. E11, 8006. October 4, 2002. robotics.jpl.nasa.gov/people/terry/papers/Haldemann_JGR.pdf. Accessed May 14, 2003.
8. Honeybee Robotics. Results from field/lab tests document of 1- and 10-m drill. Received April 4, 2003.
9. "Honeybee Robotics Rock Abrasion Tool." Honeybee Robotics, Ltd. 2001. www.honeybeerobotics.com/rat.htm. Accessed May 14, 2003.
10. "Magnetic Properties Experiment Folder: Introductory Remarks." ntserv.fys.ku.dk/mars/folder/folder.htm. Accessed May 14, 2003.
11. "Mars Exploration Rover Mission: Technology; Magnets." NASA web page. Last updated October 22, 2002. mars.jpl.nasa.gov/mer/technology/magnets.html. Accessed May 14, 2003.
12. "Mars In-Situ Sample Preparation and Distribution (SPAD)." Sample Preparation Team presentation, Mars Program Office. 4/12/2002; pg 4.
13. PSIG Instruments Summary document. JPL. March 10, 2003.
14. Richter, Lutz. Scientist, DLR-Germany. E-mail contact, April 12, 2003.
15. "Rock Abrasion Tool." PDF. athena.cornell.edu/pdf/tb_rat.pdf. Accessed May 14, 2003.
16. "The Mars Underground Mole (MUM)." Lunar and Planetary Science XXXIV (2003). 1201.pdf. www.lpi.usra.edu/meetings/lpsc2003/pdf/1201.pdf. Accessed May 14, 2003.
17. "Thermal and Evolved Gas Analyzer." University of Arizona Lunar and Planetary Laboratory, sKoBo. <http://grs.lpl.arizona.edu/tega/>. Accessed May 14, 2003.

3.3 Environment

3.3.1 Responsibilities

The Environment module is responsible for providing information to the other subsystem modules regarding the operating environment on the Martian surface where the rover is located. Environmental data required by other subsystems includes solar irradiance, sunlight hours per day, surface temperature ranges, and obstacle occurrence. This data is used by the Rover, Communications, Power, and Autonomy subsystems to further model the rover design.

3.3.2 Assumptions

Limits on time and other resources forced certain assumptions to be made when creating the Environment module. Latitude ranges were set in broad bands due to landing accuracy expected and for keeping the tool as generic as possible. Longitude ranges are not specified for the landing site. This greatly dilutes the temperature and rock coverage data due to averaging of data.

Other assumptions were made based on limited data or lack of data completely. Polar regions are not considered, as there is little to no data available to incorporate into the tool. Rock coverage data is estimated based on data from previous missions and reporting only. This limits the accuracy of the tool to estimate rock densities in high rock coverage zones, as this data is extrapolated from missions that have occurred in low rock and crater density areas. Data was not available for incorporating local slope estimation or crater avoidance modeling in the module.

There were also several assumptions made in calculating solar irradiance on the surface of Mars. The solar irradiance was calculated only for horizontal surfaces. Also, an average albedo was assumed for the latitude ranges used in the Science Vector. The optical depth of the atmosphere used in the calculations was $\tau = 0.5$, which corresponds to a clear day and does not account for any sort of dust storms. It is also assumed that the highest solar irradiance for a particular sol happens at high noon ($\omega = 0$). In order to perform the calculations, representative latitudes were used for each of the latitude ranges specified in the Science Vector and these representative values were only considered by season.

3.3.3 Background

3.3.3.1 Solar Irradiance

Equations for solar irradiance for horizontal surfaces on Mars were taken from Appelbaum et al. Global irradiance G_h is determined by equation (3.3.1), in which $f(z, \tau, al)$ is the normalized net solar flux function. The results of this function, which take into consideration the general circulation model (GCM) for Mars, are presented in tabular form in Appelbaum et al. The normalized net solar flux function is dependent on parameters: optical depth (τ), solar zenith angle (z), and surface albedo (al).

$$G_h = G_{ob} \cos(z) \frac{f(z, \tau, al)}{1 - al} \quad (3.3.1)$$

G_{ob} is the instantaneous beam irradiance of Mars – solar radiation on top of the atmosphere, and it is governed by equation (3.3.2), also taken from Appelbaum et al. 1993.

$$G_{ob} = 590 \frac{[1 + e \cos(L_s - 248)]^2}{(1 - e^2)^2} \quad (3.3.2)$$

The two parameters that affect direct beam solar radiation are e , Mars' orbital eccentricity, which is 0.093377, and L_s , areocentric longitude. Areocentric longitude is a measure of where Mars is on its orbit around the Sun. 0° of longitude refers to Northern Hemisphere vernal equinox, while 90° is summer solstice, and so forth. 248° , in equation (3.3.2), is Mars perihelion. 590 W/m^2 is the mean beam irradiance.

The values used for optical depth and albedo for the normalized net solar flux is as follows. Optical depth was assumed to be $\tau = 0.5$ because, unless dust storms are present, this value is typical (Appelbaum et al. 1993). Albedo was calculated for the particular latitude (bands) of proposed landing site. Appelbaum also presents in tabular form the values of albedos as a function of longitude (East-West) and latitude (in ten degree step sizes). Average albedos were used for each of the latitude bands (Table 3.3.1, Figure 3.3.1). Note that there are very small differences between the albedos for latitude bands. Furthermore, the net solar fluxes were only tabulated for $al = 0.1$ and 0.4 , hence linear interpolation, as suggested by the article, was used to figure out the intermediate values.

Table 3.3.1: Representative latitudes and average albedo for each latitude band

Latitude Band	Representative Latitude	Average Albedo Used
Equatorial (-5° to 5°)	0°	0.2277
Low South (-5° to -40°)	-20°	0.2277
Low North (5° to 40°)	20°	0.2277
High South (-40° to -80°)	-60°	0.2167
High North (40° to 80°)	60°	0.2534

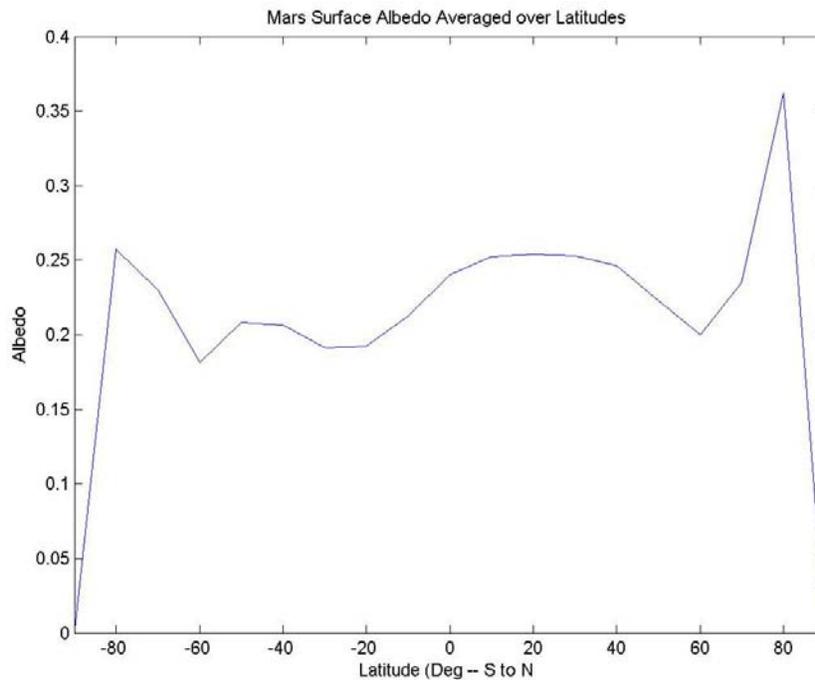


Figure 3.3.1: Average albedo over latitudes

Latitude (ϕ) was also used to calculate the appropriate solar zenith angles for the particular time of year, which were then used to calculate only the appropriate G_h values for a particular rover design's lifetime. Representative latitudes for each latitude band were used to calculate z (Table 3.3.1).

$$\cos(z) = \sin(\phi)\sin(\delta) + \cos(\phi)\cos(\delta)\cos(\omega) \quad (3.3.3)$$

L_s , areocentric longitude, is directly coupled with the declination number (δ). Traditionally, δ is just the angle which Mars' spin axis it tilted by, which causes seasons. However, in equation (3.3.3), $-24.936^\circ \leq \delta \leq 24.936^\circ$ and it is representative of the time of year, much like the areocentric longitude.

$$\sin(\delta) = \sin(24.936^\circ)\sin(L_s) \quad (3.3.4)$$

Thus, with latitude and declination known, hour angle, ω , is the only missing variable. Hour angle, as specified by Appelbaum, is a representation of the time of day about high noon (positive angle is after noon, negative angle is before noon). Assuming that high noon is the time of day where the solar irradiance is at its highest for the sol, ω is set to 0. Thus, at high noon the *smallest* solar zenith angles will be seen ($z = 0^\circ$ is sun directly overhead) with the *highest* amount of solar irradiance (Figure 3.3.2).

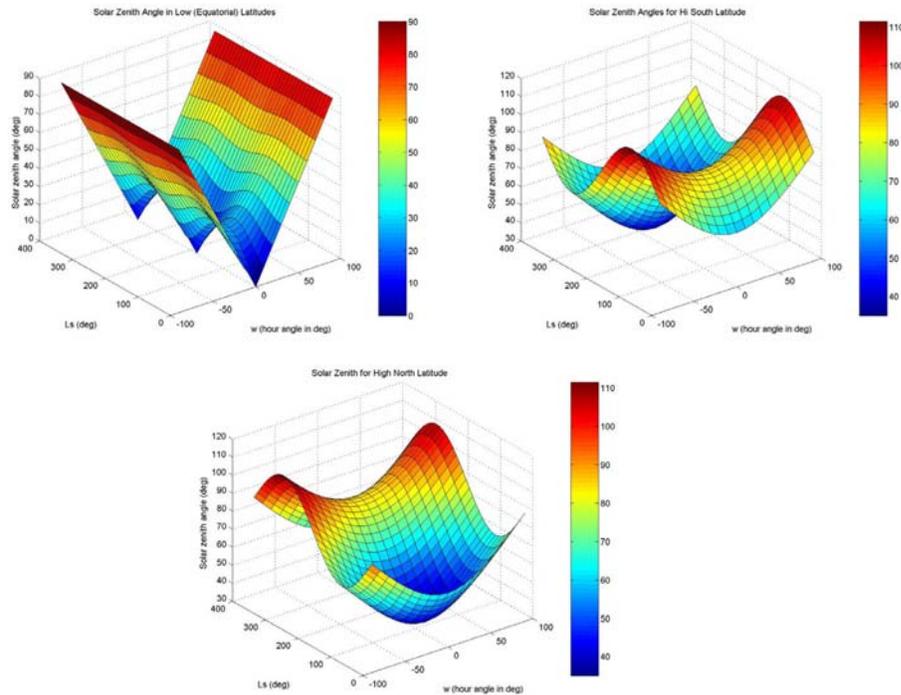


Figure 3.3.2: Solar Zenith Angle as a Function of Areocentric Longitude and Hour Angle across Latitude Bands

It is important to note at this time the effect that design lifetime had on the general organization of the Environment module. Landing date (represented in L_s landing) is used to figure out the season in which the rover's mission would start. If the lifetime is not longer than the number of sols remaining for that season, the time of year considered is simply that season as a whole (not part). Thus, time of year (appropriate L_s to consider within equation (3.3.3)) was broken down into seasons. If the lifetime encompassed any part of the season, the entire season is considered. Once the mission lifetime is greater than a Martian year, all four seasons are considered.

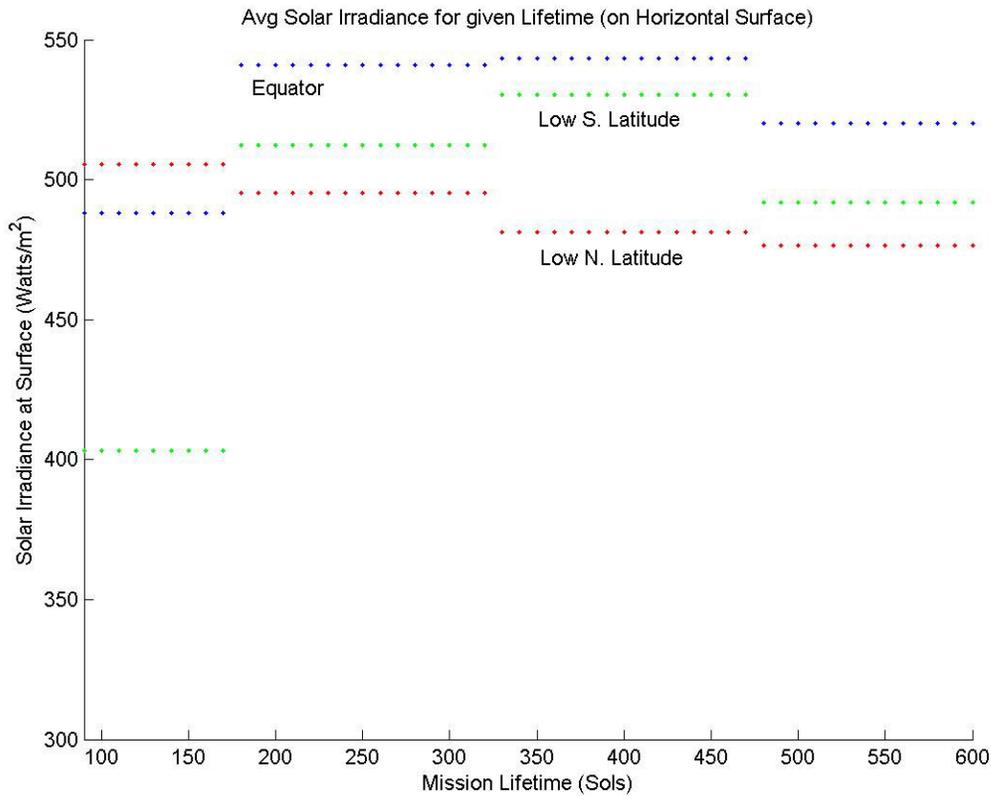


Figure 3.3.3: Average solar irradiance across low latitude bands

The Environment module outputs average, maximum, and minimum values of solar irradiance (Watts/meter²). Average solar energy is the mean of all possible G_n values for the appropriate solar zenith angles and seasons (Figure 3.3.3). Maximum solar energy is the maximum possible over the seasons that are being considered. Minimum is the smallest high solar energy for a particular day within the seasons – otherwise, the minimum solar energy experienced is zero, which obviously happens at night.

3.3.3.2 Hours of Sunlight per Sol

Following a similar structure as the solar irradiance coding, in particular the seasons and representative latitudes portion, the number of hours of sunlight on the surface of Mars were calculated using another Appelbaum et al. 1995 reference.

$$T = \frac{2}{15} \cos^{-1}(-\tan(\delta)\tan(\phi)) \quad (3.3.5)$$

This is a direct and unambiguous equation that only requires declination (δ) and latitude (ϕ). Figure 3.3.4 shows a plotting of equation (3.3.5) across all latitudes. Notice that the plots are symmetric and hence the average length of a sol over a year will be the same at any latitude.

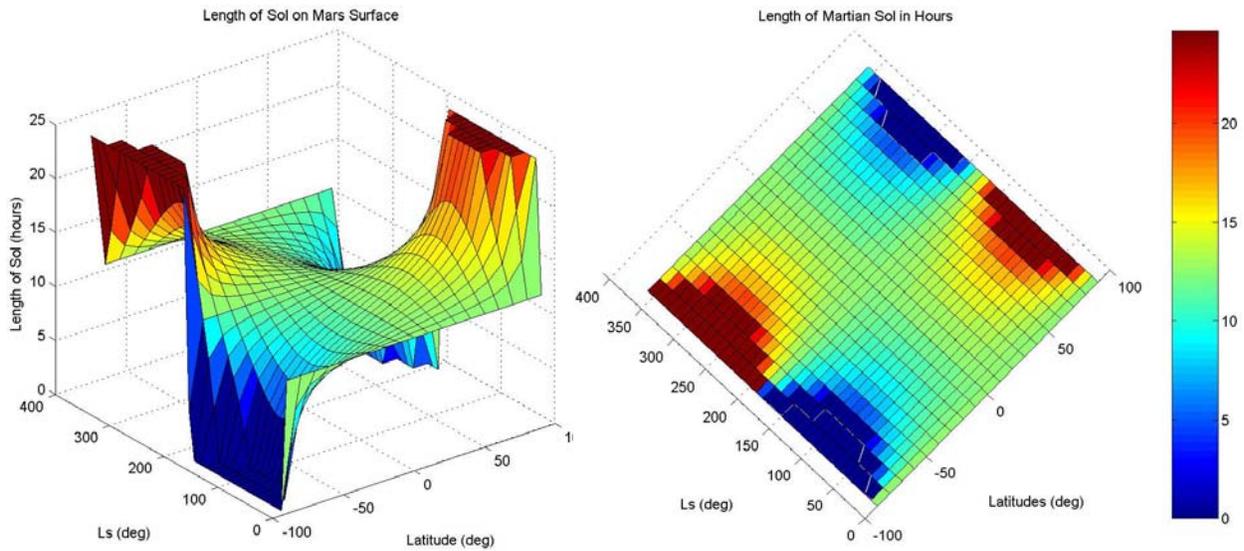


Figure 3.3.4: Hours of sunlight on Mars as a function of latitude and areocentric longitude

3.3.3.3 Temperature

Temperature data utilized in the Environment module comes from a General Circulation Model (GCM) of Mars. GCM's are numerical models of planetary environmental conditions that are tuned to observational data available. The Martian GCM used for the temperature data here has been tuned to and does a good job of representing the observations of Voyager 1 and 2. The model has been further validated against all available observational data and been shown to be fairly accurate.

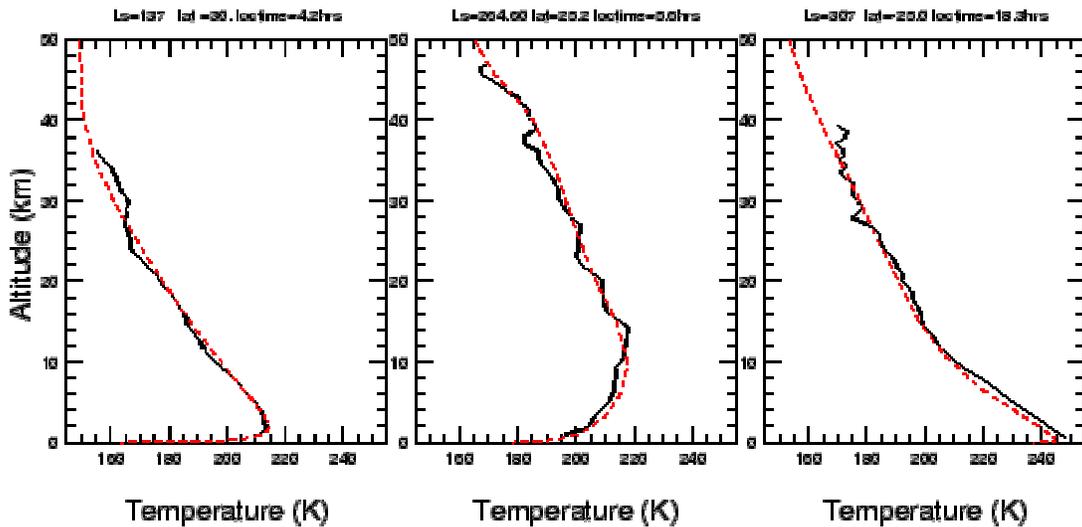


Figure 3.3.5: Examples of good fits of the GCM to observational data. The black solid lines show measured temperature profiles from Mars Global Surveyor while the red dashed lines show the GCM predictions.

3.3.3.4 Rock Density Determination

Plotting size-frequency data for rocks from the two Viking landing sites on a log-log scale shows a curve that can be fit with simple exponential functions to describe Martian rock populations.

$$N(D) = Le^{(-sD)} \quad (3.3.6)$$

$$F(D) = ke^{(-qD)} \quad (3.3.7)$$

In these equations D represents the diameter of the rock in meters. Equation 3.3.6 represents the cumulative number of rocks with a diameter greater than D per square meter. Equation 3.3.7 represents the cumulative fractional area covered by rocks with a diameter greater than D . L represents the total number of rocks of all sizes per square meter, k represents the fraction of surface area covered by rocks of all sizes (total rock coverage), and s and q are exponents based on L and k . The actual data and the exponential curve fits are shown in Figure 3.3.6.

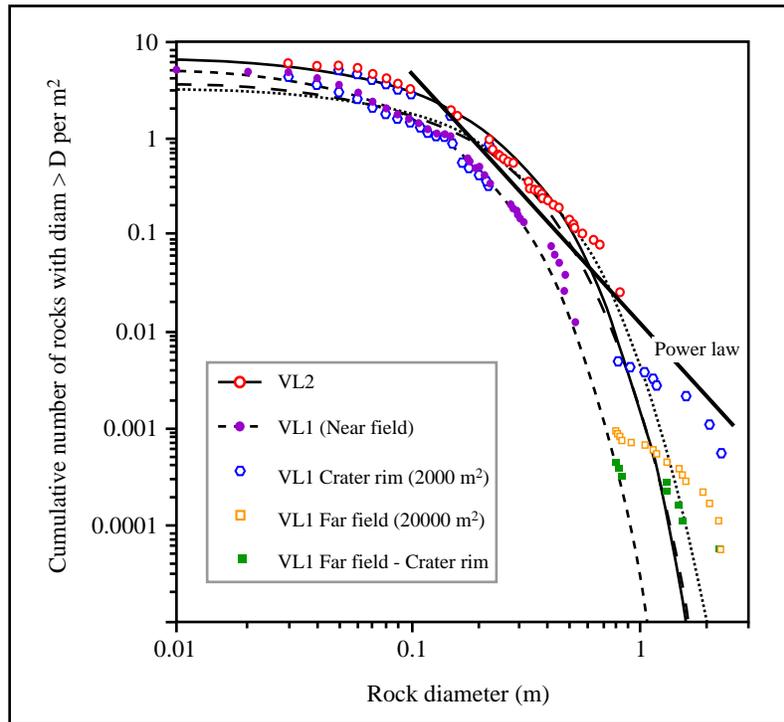


Image by MIT OpenCourseWare.

Figure 3.3.6: Cumulative number of rocks with diameter > D per m² versus rock diameter in (m) [5]

Golombek et al. have published these results with a greater insight into the equations for cumulative fractional area that includes defining the exponential q as:

$$q(k) = 1.79 + \frac{0.152}{k} \quad (3.3.8)$$

These findings have also been shown to agree with data collected from the Mars Pathfinder landing site.

The Autonomy module requires rock density information to calculate the mean free path of the rover and determine driving distances and times for different levels of autonomy. For these calculations it is necessary to use the cumulative number of rocks expected for rocks that are greater than the size of the largest rock that is traversable by the rover.

The largest traversable rock size is determined by multiplying the wheel diameter by an obstacle clearance factor. The default setting for the obstacle clearance factor is one for a one to one ratio of

wheel diameter to largest traversable rock, though the rover may in fact be able to drive over rocks up to 1.5 times its wheel diameter.

It was also necessary to determine the values for L and s to be used in equation 3.3.6 for the Environment module. The work published by Golombek et al. showed that there is no analytic way to go from the cumulative area relationship to a cumulative number relationship. Due to this the equations published for cumulative fractional area could not be used and integrated for the needs of the model in the tool. Numerical data collected from the previous Mars landing sites, as well as sample rock fields on Earth, was used to determine equations for L and s based on the total rock coverage in the landing area.

$$s = 2.28 + \frac{0.055}{k} \quad (3.3.9)$$

$$L = -4.28k^2 + 11.54k + 1.36 \quad (3.3.10)$$

The cumulative number of rocks per square meter equation was then used and numerically integrated in small diameter bins to determine the number of rocks greater than the largest traversable rock that the rover would encounter per square meter. The Autonomy module then uses this information to determine the mean free path of the rover.

3.3.3.5 Other Outputs of the Environment Module

There are two more variables that are passed within the Environment subsystem module, the soil bearing strength and gravitational acceleration of Mars. These are not calculated but rather are constants. The soil bearing strength used is an estimation provided by Charles Whetsel, while the gravitational acceleration is the commonly accepted average – 3.7 m/s².

3.3.4 Program Flow

Figure 3.3.7 shows the how the section calculating solar irradiance was coded.

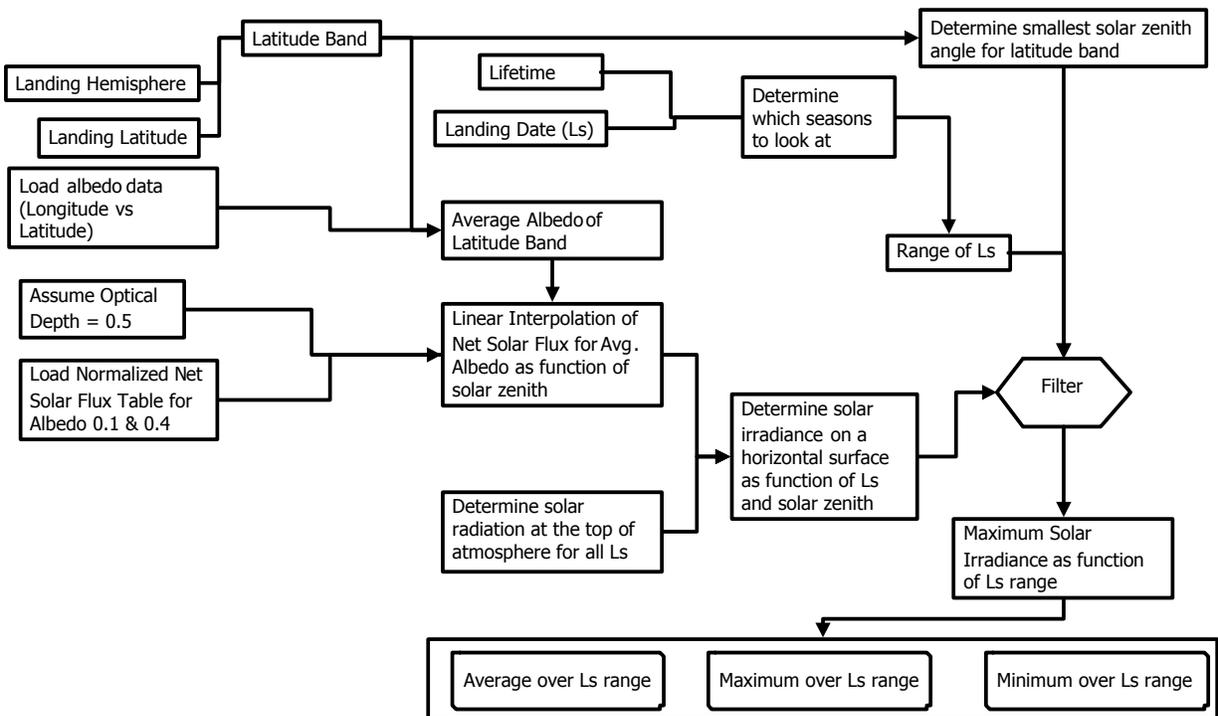


Figure 3.3.7: Solar irradiance flow diagram

Following the solar irradiance calculations, the module then calculates the average day, average night, maximum day and minimum night temperatures based on the landing site location data and specified mission lifetime for a given iteration of the Design Vector. The module reads a different text file containing the temperature data depending on the latitude and mission start date of the mission. The temperatures in the data files are then averaged based on the mission lifetime, if necessary.

After the temperature calculations, the rock density calculations are performed. Equations 3.3.9 and 3.3.10 are calculated based on the Science Vector value for rock coverage and the Design Vector values for wheel diameter and the obstacle factor. Equation 3.3.6 is then evaluated in small diameter bins and discretized to determine the number of rocks greater than the maximum traversable rock size per square meter expected. The average obstacle size and average obstacle area are calculated from this data, as well as the average separation between obstacles. This data is then written to the module to be used by other subsystems.

3.3.5 Validation

The model used to calculate hours of sunlight per sol was not validated because it was not necessary to prove that the equations used are correct.

In order to validate the solar irradiance model, a graph was created using the developed Matlab code that plotted Watt-hours/sol as a function of Areocentric longitude (Figure 3.3.8). This was done in an effort to duplicate a graph supplied by Whetsel. Whetsel's graph (Figure 3.3.9) shows daily total solar Watt-hours per square meter impinging on a horizontal surface at an optical depth of $\tau = 0.5$. In Figure 3.3.8, three latitudes were plotted, 0° , 60° , and -60° . In comparison with Figure 3.3.9, the graph outputted by the coded solar irradiance model overestimates the total amount of Watt-hours per sol. It

is believed that the code does work properly however, since it replicates the wave function rather closely (notice the peculiar shape of 0°). Possible explanations for this discrepancy are that Whetsel's document does not comment on which albedos were used for creating the graph and, over the last eight to ten years, it is likely that improvements have been made on the general circulation models of Mars which are at the basis of the solar irradiance models presented by Appelbaum.

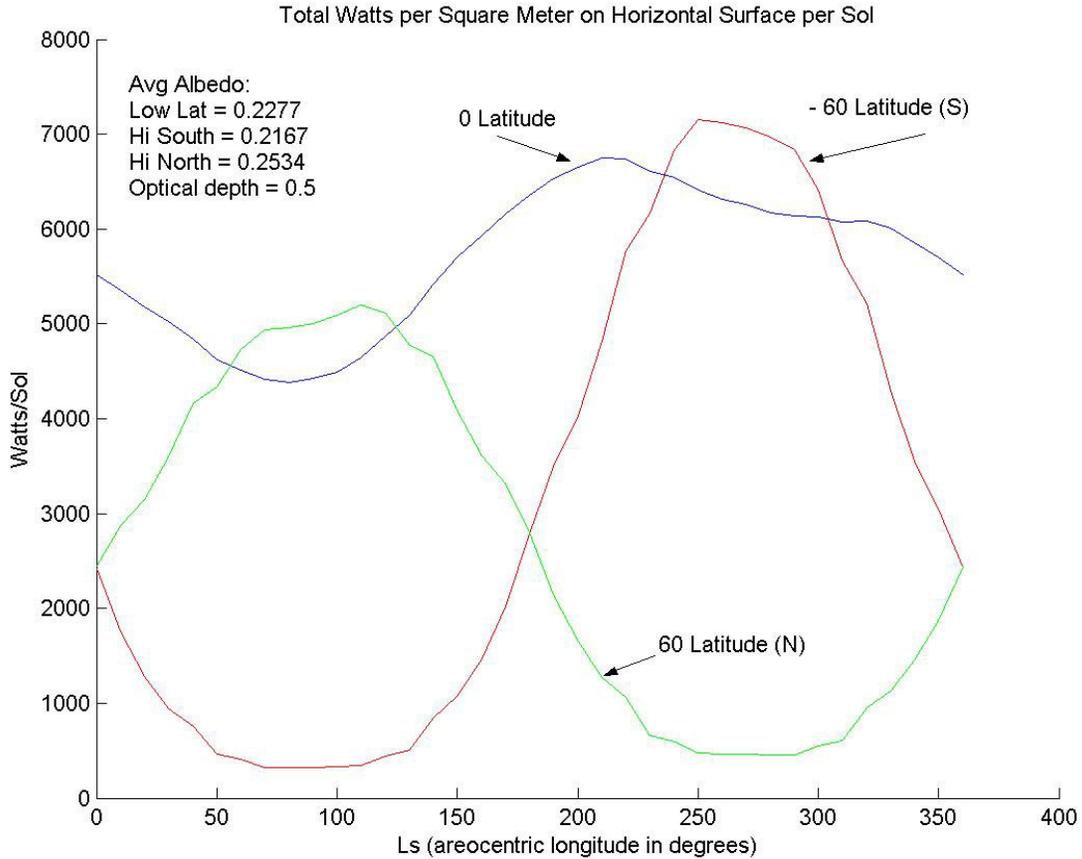


Figure 3.3.8: Watt-Hours per square meter on a horizontal surface on Mars per sol. Three latitudes are shown: 0° , 60° N, and -60° S. Optical depth = 0.5.

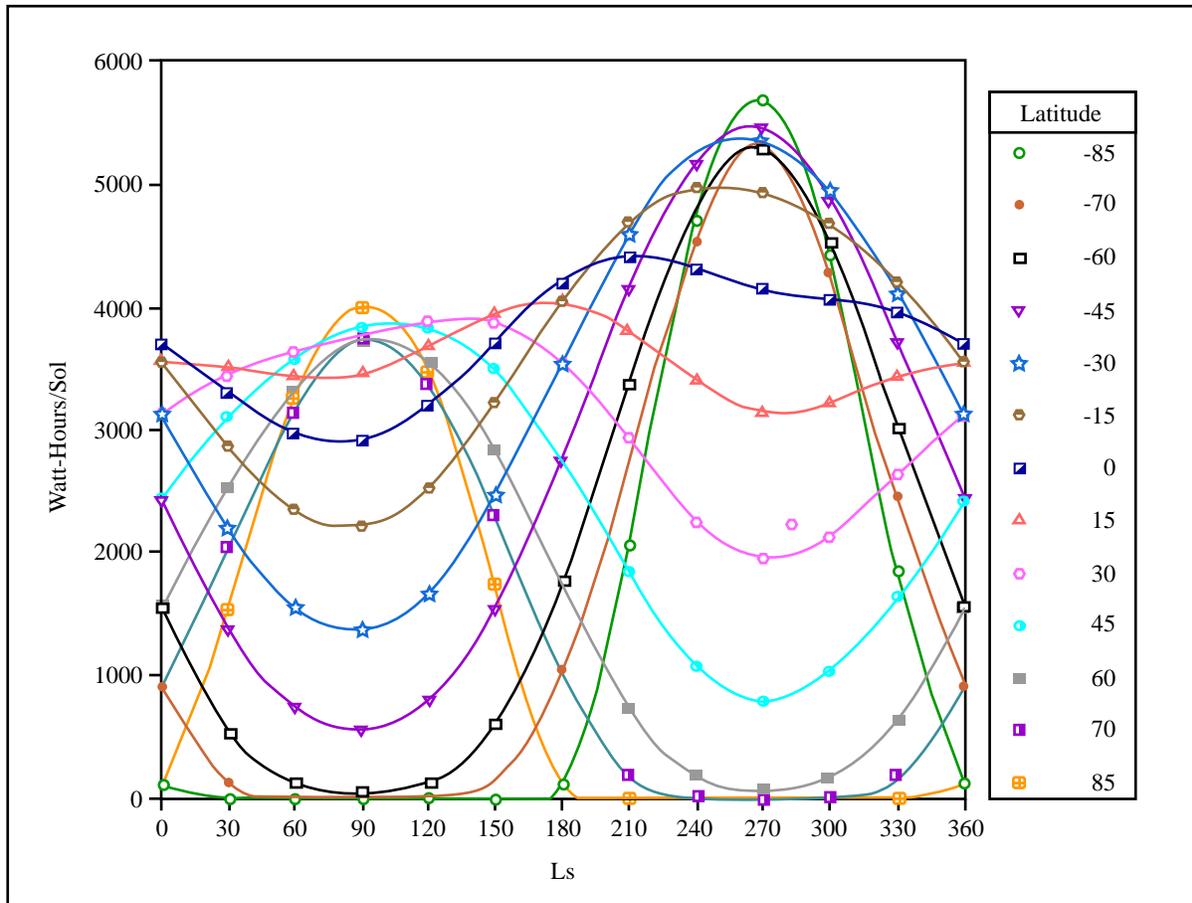


Image by MIT OpenCourseWare.

Figure 3.3.9: Total Watt-Hours on a horizontal surface on Mars as a function of latitude [9].

The data used for Mars surface temperatures was not separately validated as the global circulation model it was taken from was already validated.

The equations from the Environment module reproduced the curve fit equations and graphs published by Golombek et al. to validate the rock density determination model. Figure 3.3.10 (a) shows the curves developed by the equations used in the model, while Figure 3.3.10 (b) is taken from a presentation supplied by JPL [6]. Numerical exploration of the graphs shows them to be equivalent.

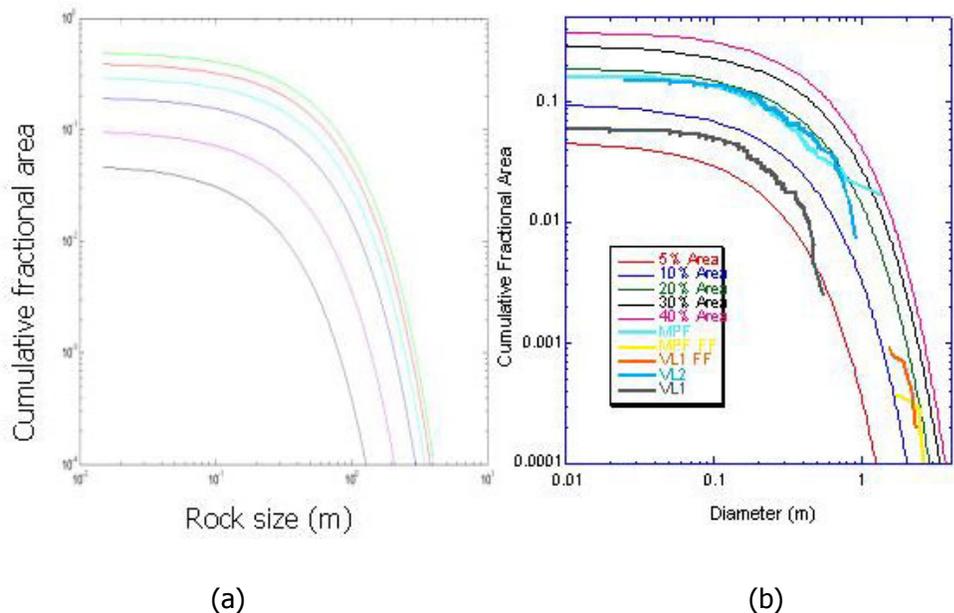


Figure 3.3.10: (a) Data produced by the Environment module for cumulative fractional area covered with rocks larger than rock size (in meters). Rock coverage percentages range from 5% to 50%. (b) Published data from JPL of cumulative fractional area covered with rocks larger than diameter (in meters). Past Mars surface mission data also shown.

3.3.6 Expandability

There are many improvements that can be done to make the Environment module more complete and a more accurate representation of the Martian surface. First, the latitude bands should be refined to include smaller latitude ranges (e.g. every 10 degrees). Also, the seasonal approach of averages, maximums, and minimums should be refined into smaller separations (e.g. by 30° Ls instead of 90°).

Another capability that might be necessary to include in the future is to be able to calculate solar irradiances for different optical depths. Appelbaum does supply the appropriate information – the normalized net solar fluxes; however, the current code does not include other optical depths. This capability would be advantageous to have in order to see the effects of dust storms on designs that use solar arrays as their main source of energy.

Modeling the surface of Mars for crater avoidance is also something not considered by the Environment module due to lack of information. This includes determination of local slope and rock coverage and distribution information in crate zones.

Further expansion of the Environment module is also necessary if other mission phases, such as entry and descent, are to be included in the modeling tool. This will require adding capabilities such as landing obstacle avoidance modeling and modeling for areas not necessarily at ground level.

3.3.7 References

1. Appelbaum, J., G. A. Landis, and I. Sherman "Solar Radiation on Mars – Update 1991." *Solar Energy*. 50(1): 35 – 51. 1993.

-
-
2. Applebaum, J., I. Sherman, and G. A. Landis "Solar Radiation on Mars: Stationary Photovoltaic Array." Journal of Propulsion and Power. 11(3): 554 – 561. May/June 1995.
 3. Allison, M. "Accurate analytic representations of solar time and seasons on Mars with applications to the Pathfinder/Surveyor missions." Geophysical Research Letters, 24(16): 1967 – 1970. Aug. 15, 1997.
 4. Golombek, Haldermann, Forsberg-Taylor, DiMaggio, Schoeder, Jakosky, Mellon, and Matijevic "Rock Size-Frequency Distributions on Mars: At the Pathfinder Landing Site, and in Boulder Fields, Thermal Inertia of Rock Populations, and Rock Shape and Burial and Implications for Mars Exploration Rover Landing Safety and Operations." Submitted to MER Special Issue of Journal of Geophysical Research, Planets. December 19, 2002.
 5. Golombek and Rapp. "Size-Frequency Distributions of Rocks on Mars and Earth Analog Sites: Implications for Future Landed Missions." Journal of Geophysical Research, Vol. 102, No. E2, Pages 4117-4129. February 25, 1997
 6. Models of the Martian Surface: Crater and Rock Hazard Modeling for Mars Landing. Presentation by Doug Bernard and Matt Golombek, JPL, Mars Program EDL Workshop, August 2002
 7. Website reference: <http://www.giss.nasa.gov/data/mars/time/>
 8. Website reference: <http://www-mars.lmd.jussieu.fr/mars.html>
 9. Whetsel, Charles. Notes on Surface System Design Process and Sizing Relationships for 16.89 class, Spring 2003.

3.4 Rover

3.4.1 Responsibilities

The primary purpose of the Rover module is to model the structure, mobility, and thermal components of the rover. The design and sizing of these three components, along with the Power subsystem, determine many of the raw capabilities and limitations of the rover hardware. In addition, the Rover module models the mast and robot arm. The five subsystems contained within the Rover module can be summarized as follows:

- Structure (`structure.m`) – the structure module sizes the warm electronics box (WEB), the main body of the rover, to meet packaging and strength requirements. The walls of the WEB are sized to withstand bending and buckling loads during launch and reentry, and the size of the WEB is determined by the volume of the instruments, electronics, and other hardware that must fit inside it, as well as by the need to support a solar array or RTG on top of it.
- Mobility (`mobility.m`, `structure_find_dimensions.m`) – the mobility module sizes the rocker-bogie suspension system, including wheels, motors, and linkages. The total mass of the rover directly affects the design of the mobility system through structural considerations (bending loads) and through actuator sizing.
- Thermal (`thermal.m`, `temp_convection_radiation.m`, `heat_convection_radiation.m`) – the thermal module determines the mass and power required to provide heating and cooling to the instruments, electronics, and other hardware inside the WEB. Each instrument, acquisition tool, and electronics package has individual thermal requirements, generally characterized by maximum temperature, minimum operating temperature, and minimum survival temperature. These temperature limits are considered with respect to ambient average and ambient extreme temperatures to size the components of the thermal subsystem. The heating and cooling components of the rover thermal system must be able to maintain each instrument within its allowable temperature range at all times, including at daytime high and low ambient temperature extremes. Primary outputs of the thermal module include the mass, size, and power requirements of the heaters, heat pipes, and radioisotope heater units required to maintain the payload within the allowable temperature range.
- Mast (`mast.m`) – the mast is a vertical appendage mounted on top of the WEB, onto which navigation and science instruments are attached. The algorithm currently allows for only one mast. The mast module uses a very low fidelity model and is essentially a placeholder in the Rev 1 Code.
- Arm (`arm.m`) – the arm is a jointed appendage mounted in front of the WEB, onto which acquisition tools and science instruments are attached. The algorithm currently allows for only one arm. The arm module uses a very low fidelity model and is essentially a placeholder in the Rev 1 Code.

The outputs of each of the Rover sub-modules are used by other Rover sub-modules, and some of these outputs are returned by the Rover module. Principle outputs of the Rover module include the total rover size and mass, various component sizes and masses, heater power requirements, and mobility constraints such as maximum speed and obstacle clearance height.

3.4.2 Assumptions - General

Default values for all the parameters that can be changed by the user are detailed in the default values subsection at the end of the assumptions section.

3.4.2.1 *Scope and level of detail*

The scope of the Rover module is limited to high level design, which is consistent with the levels of detail and accuracy in the overall rover design model. Therefore some details of the rover design are captured in safety factors rather than being modeled directly. For example, the mass of the cables used to connect the hardware components of different subsystems is estimated to be a fixed percentage of the total mass. Some design constraints like packaging and manufacturability are not taken into account in the model.

3.4.2.2 *Materials*

Information on the materials used to build the rover is required for the structural design of the WEB and suspension. The user can change these materials and their properties by modifying the `rover.m` file. The materials currently used are:

- Aluminum 2219-T851 for all metallic plates [JRW]
- Aluminum 6061-T6 for all beams and columns [JRW]
- Aerogel for insulation as defined in [HWS]
- 5052 H39 - 0.001P Hexagonal Honeycomb used for lightweight core in sandwich structures [HEX]

3.4.2.3 *Maximum acceleration*

The maximum acceleration experienced by the rover is an important consideration for the structural design. The landing deceleration on Mars is assumed to be greater than the launch acceleration on Earth. The maximum acceleration is set to 20g (refer to Default values), an assumed value for a soft landing deceleration, which is based on a hard landing load of 50g [JMa].

3.4.3 Assumptions – Mobility

The purpose of the mobility module is to determine the mass and power required for the rocker/bogie suspension system, which includes wheels, motors, and linkages. The size of the mobility system is determined primarily by the wheel diameter and the total mass of the rover. The mobility system must be strong enough to withstand static and dynamic loads likely to be encountered during operations on the Martian surface. The maximum speed of the rover is calculated based on a curve-fit to existing rover designs.

3.4.3.1 *Mobility dimensions*

Currently, only the rocker-bogie suspension system (patented by the Jet Propulsion Laboratory) has been considered. This design was chosen because it is used on Sojourner and MER, and because other six-wheeled designs will probably provide a similar performance. For example, the suspension system developed by Rover Science and Technology Company, selected by ESA for the ExoMars rover, has six wheels like the rocker-bogie [EXO]. Hence Sojourner, MER and ExoMars rovers can be used for design or as validation points. The main consequence of using this suspension system is the sizing of the rover

footprint. For a rocker-bogie suspension, the footprint dimensions are directly determined by the wheel diameter (see Eq. 3.4.1) [CWh].

$$L_{Wheelbase} = C_{user} D_{Wheel}, \quad L_{Track} = L_{Wheelbase}, \quad L_{Rover} = L_{Wheelbase} + D_{Wheel} \quad (3.4.1)$$

Where L is length, D is diameter, and C_{user} is a constant set by the user, with a default value of 3.7 [CWh]. Based on the Sojourner and MER designs, the height of the suspension system is estimated to one and a half times the wheel diameter. The wheel diameter is a parameter in the Design Vector, so the rover footprint can be easily calculated given the Design Vector.

The wheel diameter also drives the wheel width through sinking and turning ability constraints. The wheel must be wide enough to maintain the wheel floatation pressure below the Martian soil bearing strength, and narrow enough to permit turning the wheel. In Eq. 3.4.2, the left hand side is the floatation pressure, also known as the nominal ground pressure [AEI].

$$\frac{2W_{Rover}}{N_{Wheels} D_{Wheel} b_{Wheel}} \leq S_{Soil} \quad (3.4.2)$$

Where W_{rover} is the total weight of the rover on Mars, N_{wheels} is the number of wheels, D_{wheel} is the wheel diameter, b_{wheel} is the wheel width, and S_{soil} is the soil bearing strength. This equation can be solved to give a lower bound to the range of values for the wheel width as a function of wheel diameter and rover mass; however, this equation is not currently used due to uncertainty in the accuracy of the equation based on comparison with the Sojourner and MER designs.

An upper bound for the wheel aspect ratio (the ratio of wheel width to wheel diameter) comes from the turning ability constraint, which says that for the rover to be able to turn, the wheel aspect ratio must be smaller than $C=0.6$ (see Eq. 3.4.3) [CWh].

$$\frac{b_{Wheel}}{D_{Wheel}} \leq C \quad (3.4.3)$$

In the mobility module, the wheel width is set at the upper bound $b_{Wheel} = 0.6D_{Wheel}$ because of uncertainties in the Martian soil bearing strength.

3.4.3.2 Mobility structural design

In order to support the weight of the rover, the geometry of the suspension system must be considered during structural design. The assumed suspension geometry is shown in Figure 3.4.1, which is based on the MER design. The elements of length L_1 and L_2 are modeled as simply supported beams with center point loads (see Eq. 3.4.4). The actual point loads are the nominal point loads multiplied by the factor of safety, f_s , which is set to five by default.

$$\begin{cases} L_2 = 0.4L_{Rover} \\ L_1 = L_{Rover} - (L_2 / 2 + D_{Wheel}) \end{cases} \quad (3.4.4)$$

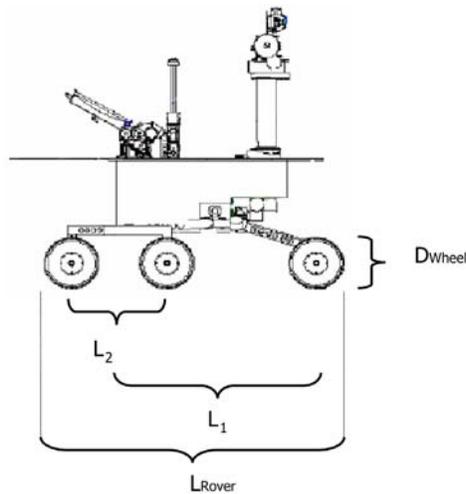


Figure 3.4.1: Suspension structural model (based on [MER])

3.4.3.3 *Mobility performance*

The rover mobility performance can be characterized by two quantities: the ability to drive over rocks, and the maximum speed. For a rocker-bogie suspension, the traversable obstacle height is directly related to the wheel size; the rover can traverse obstacles of height equal to 1.5 times the wheel diameter [CWh]. The ability of the rover to handle slopes is not yet modeled because it requires detailed knowledge of the rover mass distribution.

The main drivers for the maximum speed of the rover are the available power, rover mass, and structural design. Different drivers predominate according to the range of speeds for which the rover is designed. Existing rovers move slowly enough to be considered 'quasi-static', in which case speed scales with power and weight (personal communication with D. Bickler, JPL). The driving power required is an output from the Rover module, so the total mass of the rover is estimated while designing the mobility system. The maximum speed is assumed to be a function of the wheel diameter. This assumption allows the velocity to be determined independently from the other rover properties, which greatly simplifies the rover design algorithm. The maximum speed is determined in the function `rover_init`, which executes before the Autonomy module, which requires maximum speed as an input. A function relating maximum speed and wheel size was generated, based on a curve fit to data points from Sojourner, MER and the current design for MSL. The curve fit, shown in Figure 3.4.2, implies that the velocity saturates for wheel diameters larger than 1 meter, an assumption made based on qualitative structural considerations. In this range of wheel diameters the rover is not considered 'quasi-static' but 'dynamic'; the limiting factors in the average speed of the rover are then the structural design of the rover and the efficiency of the navigation algorithms. The saturation reproduces very conservatively the impact of these limitations. The validity of the curve fit is discussed in more detail in the validation section.

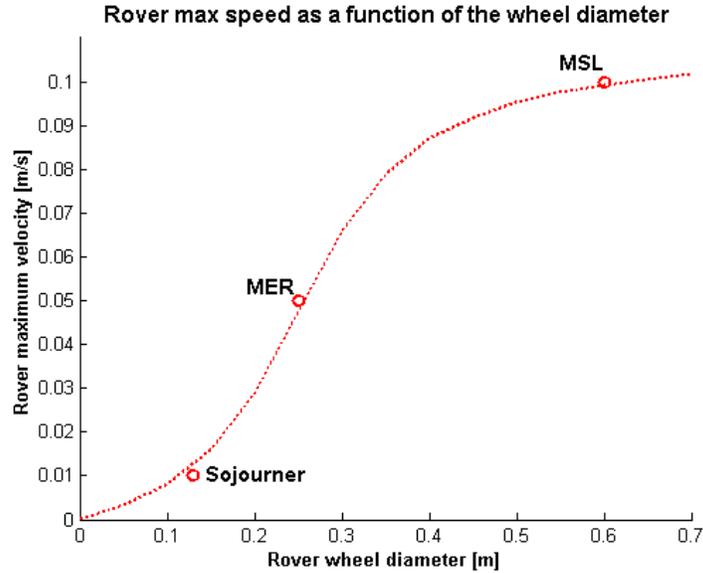


Figure 3.4.2: Velocity as a function of wheel diameter

Based on [CWh] guidelines, the driving power is scaled with the maximum velocity and the rover total mass. There is one motor for each of the six wheels and the required power for each motor is sized to provide a thrust equal to half the weight of the rover on Mars (Eq. 3.4.5).

$$P_{Drive} = N_{Wheels} \frac{M_{Rover} g_{Mars}}{2} \frac{V_{Max}}{\eta_{Gear}} \quad (3.4.5)$$

Where P_{Drive} is the overall power required to drive the wheel motors, g_{Mars} is the gravitational acceleration on Mars, V_{Max} is the rover's maximum speed, and η_{Gear} is the efficiency of the mechanical transmission and gearing. This formula is compared to empirical data in the Validation section. Motors are sized based on MER and Sojourner values.

3.4.4 Assumptions - Structure

The purpose of the structure module is to determine the size and mass of the main body of the rover, referred to as the warm electronics box (WEB). The function of the WEB is to contain and support the rover equipment. The size and mass are therefore driven by the sizes and masses of the hardware components that are contained inside or attached to the WEB. The process followed is to first size the WEB so that all the necessary elements fit inside and then to design its structural properties so that it can support them. The following assumptions are made for each of these steps.

3.4.4.1 WEB geometry

The shape of the WEB is a regular parallelepiped, as used for the Sojourner rover and shown in Figure 3.4.3. The assumption of a simple shape greatly simplifies the structural design algorithm. The sides of the WEB are referred to as top and bottom plates and walls.

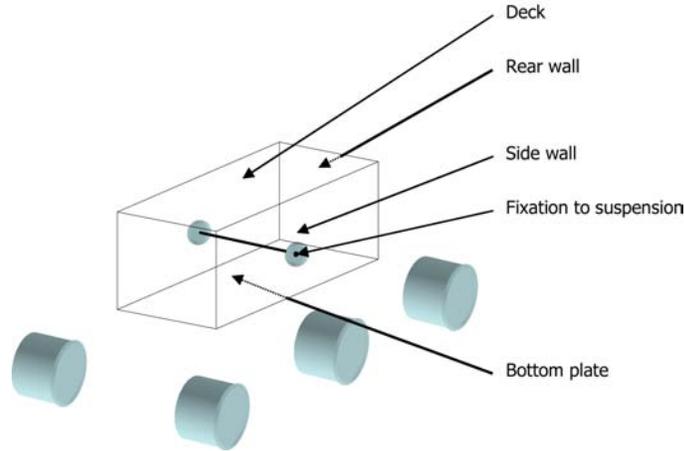


Figure 3.4.3: WEB geometry

3.4.4.2 WEB size

When sizing the WEB, the objective is to minimize its volume to save mass so that it actually fits inside the rover footprint. The dimensions of the rover footprint are driven entirely by the wheel diameter. The WEB size is also constrained by the volume of hardware that must fit inside and by the dimensions of the solar arrays that it must support. Therefore sizing is done in three steps: first the WEB is dimensioned so that it can contain all the required hardware; second, it is compared to the solar array constraint and expanded if necessary; third, it is compared to the footprint constraint.

Algorithms to arrange a set of volumes in order to maximize the compactness are very complex and computationally expensive. Therefore, the problem of sizing the WEB is simplified through the following assumptions. The area of the WEB (of its bottom and top plates) and its height are calculated independently. The area is estimated according to the equipment present inside the WEB, which is assumed to be lying on the bottom plate (no shelving of hardware is allowed). It is then assumed that all the pieces of hardware fit in a rectangular area a little bigger than the sum of all the individual hardware areas. The WEB area is set equal to this summed area times a factor defined by the user which accounts for imperfect arrangement of the pieces into a rectangular area. In the same manner, the height of the WEB is set equal to the largest height of all hardware pieces times the same factor (refer to Default Values).

The WEB top area must be large enough to support the solar panel, if one is present in the design. If the ratio of the solar panel area to the WEB top area is less than a threshold set by the user, then the design is valid. However if the area ratio is greater than the threshold, the size of the WEB is increased in order to reduce the area ratio to the threshold value. In this manner the WEB is sized to support the solar panel and made to be large enough to contain the required equipment.

The WEB is assumed to have the same aspect ratio (width divided by length) as the rover footprint. Since the size of the footprint is based entirely on the wheel diameter, then given a particular WEB size, it is possible to check if the WEB actually fits inside the rover footprint. For the design to be valid, the following system of equations must be verified (refer also to Default values).

$$\begin{cases} W_{WEB} \leq W_{Rover} - C_1 \times D_{Wheel} \\ L_{WEB} \leq L_{Rover} - C_2 \times D_{Wheel} \end{cases} \quad (3.4.6)$$

W and L represent the WEB width and length, respectively. C_1 and C_2 are parameters set by the user. If either of these equations is not satisfied, the rover design is invalid.

3.4.4.3 WEB structural design

The top plate, bottom plate and walls are modeled separately. For all plates and beams, the boundary condition is that all edges are clamped. Both the bottom and top plates have to support normal loads and are designed for bending. The thickness of these plates is sized in order to meet a maximum deflection requirement set by the user (refer to Default Values). The top plate is assumed to be a simple metallic plate, but the user can choose the option of a sandwich structure for the bottom plate.

The connection between the WEB and the suspension is assumed to be located at the center of the WEB side walls; it is at this location that the weight of the WEB and the attached equipment is transferred to the suspension system. In addition to the loads from the deck and bottom plate equipment, the walls also have to support the hardware attached to them, such as arms or cameras. As a first approximation, the upper half of the walls is assumed to be under compression between the load of the deck plus the wall-attached hardware and the ground reaction. It is therefore designed for buckling. The lower half is assumed to be under tension between the ground reaction and the bottom plate equipment. The walls have three components: a structural element (carries the loads), aerogel for insulation, and fiberglass sheets for containment [HWS]. The user can change the total thickness between the two fiberglass sheets. There are also three types of structural elements that one can choose from: a simple plate, an 'H' shape column, or a Z-spar shape (see Figure 3.4.4). The Z-spar case is chosen by default because it was used on Sojourner [HWS]. No information was found on the wall design of the MER rovers. The thickness of the structural element is calculated to satisfy both buckling and tensile stress requirements. The calculations capture the axial loads but not the torques that the wall-attached hardware applies to the walls.

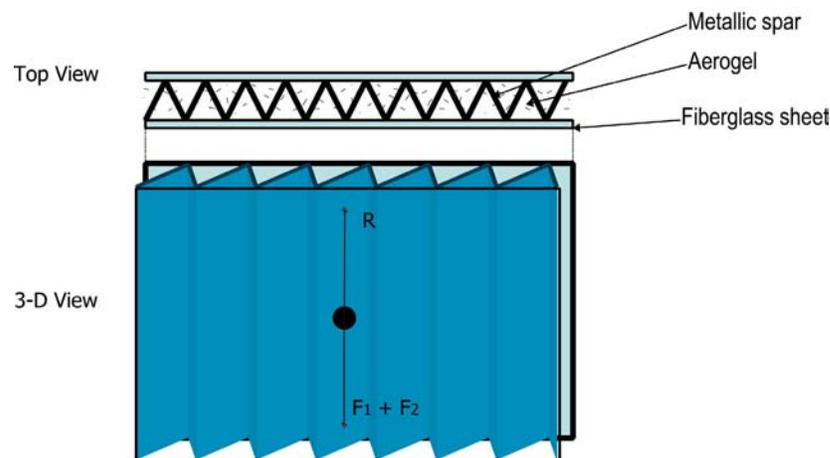


Figure 3.4.4: Wall with Z-spar structure type

R is the ground reaction force, F_1 is the weight of the deck equipment plus the WEB top plate and the equipment attached to the walls, and F_2 is the weight of the equipment inside the WEB plus the bottom plate.

The mass of attachments, bolts and other fixations is estimated to be a fixed fraction of the total calculated mass. The mass of the mobility system differential, which is located within the WEB, is currently ignored.

3.4.5 Assumptions - Thermal

The thermal requirements of each component of the rover payload (e.g. instruments, acquisition tools, electronics packages, etc.) are generally characterized by three quantities:

1. The *maximum temperature* is the highest temperature that the component may reach without sustaining damage.
2. The *minimum operating temperature* is the lowest temperature that the component can reach while in a powered-on, operational state, without sustaining damage.
3. The *survival temperature* is the absolute minimum temperature at which the component can be maintained without sustaining damage.

The scope of the thermal module does not include temperature regulation for instruments or other equipment located outside of the WEB. Heat generated from the dissipation of unused solar or RTG power is currently not included in the model. It is assumed that the maximum temperature is reached while the component is powered on, with the daytime high ambient temperature as an external influence. The minimum operating temperature is assumed to occur during average nighttime ambient temperatures. It is assumed that the minimum survival temperature is reached while the ambient temperature is at the nighttime low.

The thermal model is shown in Figure 3.4.5, where T_1 , T_2 , T_3 , and T_4 are the instrument temperature, interior wall temperature, exterior wall temperature, and ambient temperature, respectively. The heat emanating from the instruments is represented by q . In steady state, the heat being transferred from any point n to point $n+1$ is equal to q .

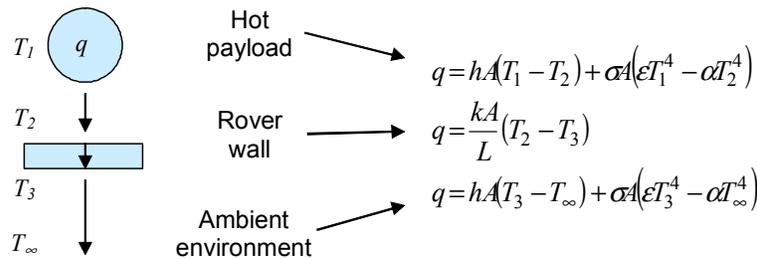


Figure 3.4.5: Thermal model for the Warm Electronics Box (WEB).

The emissivities of all payload components are assumed to be equal, and the absorptivities of all payload components are also assumed to be equal.

The convection coefficient used in the module is $k=0.5 \text{ W}/(\text{m}^2 \text{ K})$. This is a guess based on the low end of the convection coefficient scale reported by Incropera and DeWitt [IDF].

3.4.6 Assumptions – Arm and Mast

The arm and mast modules are low-fidelity placeholders. The mast is modeled as a beam undergoing bending due to the weight of a tip mass. The arm is modeled as multiple segments undergoing bending due to the weight of a tip mass. The model assumes that a 1 kg motor is used to raise the mast and arm motors are not modeled. Power requirements for raising the mast and manipulating the arm are not determined.

3.4.7 Default Values

These constants are used in the design of the mobility and structure systems. The user can change the default values of the constants by modifying the source code, in which the constants are grouped at the top of each file. The constants used to size and design the mobility system are listed in Table 3.4.1. The constants used to size and design the WEB are listed in Table 3.4.2.

Table 3.4.1: Mobility constants

Constant	Value	Units	Description
M_{Cables} / M_{Rover}	0.07		Mass fraction of cabling with respect to rover mass
$L_{Wheelbase} / D_{Wheel}$	3.7		Ratio of wheelbase to wheel diameter
$L_{Track} / L_{Wheelbase}$	1.0		Ratio of track to wheelbase
b_{Wheel} / D_{Wheel}	0.6		Ratio of wheel width to wheel diameter
$H_{Suspension} / D_{Wheel}$	1.5		Ratio of suspension height to wheel diameter
Δ_{Max}	2	mm	Maximum deflection of suspension beams
F_S	5		Factor of safety
$H_{Obstacle} / D_{Wheel}$	1.5		Ratio of traversable obstacle height to wheel diameter
η_{Gear}	0.97		Gear efficiency

Table 3.4.2: Structure constants

Constant	Value	Units	Description
$A_{WEB} / A_{WEB,0}$	1.1		Margin on WEB top area
$H_{WEB} / H_{WEB,0}$	1.1		Margin on WEB height
A_{panel} / A_{WEB}	< 6		Maximum ratio of solar panel area to WEB top area
$(L_{Rover} - L_{WEB}) / D_{Wheel}$	> 1		Check if suspension is long enough to fit the WEB
$(W_{Rover} - W_{WEB}) / D_{Wheel}$	> 2		Check if suspension is wide enough to fit the WEB
W_{Wall}	0.0254	m	Total wall thickness
W_{Sheer} / W_{Wall}	0.1		Ratio of fiberglass thickness to total wall thickness
g_{Max}	20×9.81	m/s ²	Maximum acceleration experienced by rover
Δ_{Max} / L_{Plate}	0.005		Max strain for plates
Δ_{Max} / L_{Beam}	0.005		Max strain for beams
$M_{WEB} / M_{WEB,0}$	1.1		Margin on WEB mass to account for bolts

3.4.8 Background

The WEB structure is designed using plate theory, and the suspension, mast, and arm are designed using Bernoulli-Euler beam theory.

3.4.8.1 Structural calculations – suspension system

The design procedure for the suspension system is as follows:

1. Assume rocker/bogie suspension geometry of the form used by Sojourner and MER, and shown in Figure 3.4.1.
2. Choose equations relating L_2 and L_1 to L . Assume the following relations, which produce relative dimensions similar to those shown in Figure 3.4.1.

$$\begin{aligned} L_2 &= 0.4L \\ L_1 &= L - (L_2 / 2 + d) \end{aligned} \tag{3.4.7}$$

3. The main body of the rover is attached at approximately the center point of the beam labeled L_1 , and that beam is attached to the center point of the beam labeled L_2 . Both these beams can be approximated as pin supported at each end, with a point load at the center. The following equation describes the deflection, δ , of a pin-supported beam with a center point load. Other quantities in the equation are moment of inertia I , factor of safety f_s , force N , length of the beam L_x , and modulus of elasticity E . The factor of safety is set to a default value of five.

$$I = f_s \frac{NL_x^3}{48E\delta} \quad (3.4.8)$$

4. Given a maximum allowable deflection, a known force, and known material properties, the moment of inertia of the suspension beam can be calculated. For the beams of length L_1 , the force is

$$N_1 = \frac{mg}{2} \quad (3.4.9)$$

5. Because the attachment between the mobility system and the WEB occurs at the midpoint of the beams of length L_1 , the force on the beams of length L_2 is one half that of the force on the beams of length L_1 .

$$N_2 = \frac{mg}{4} \quad (3.4.10)$$

6. Given the inertia of each beam, and assuming a square cross-section and known ratios of beam height to width and wall thickness to width, the beam dimensions can be determined by solving the following set of equations for the beam width b .

$$\begin{aligned} I &= \frac{1}{12}bh^3 - \frac{1}{12}(b-t)(h-t)^3 \\ &= \frac{1}{12}(3bth^2 - 3bht^2 + bt^3 + th^3 - 3h^2t^2 + 3ht^3 - t^4) \end{aligned} \quad (3.4.11)$$

The ratios currently assumed are $h=b$ and $t=b/2$ (i.e. a solid bar, though the algorithm has been tested with smaller values as well), as these relationships applied to a MER-sized rover returns a suspension system that appears similar to that of MER.

7. The mass of the differential (the part of the mobility system that is located inside the WEB) is assumed to be 4.4 kg. When information on the MER or MSL suspension system becomes available, this number can be replaced with a higher-fidelity estimate.
8. The mass of each wheel is determined by the following equation, where D is the wheel diameter, b is the wheel width, and T is the thickness of the wheel.

$$m_{wheel} = \rho \left(\frac{\pi}{4} D^2 T + \pi D b T \right) \quad (3.4.12)$$

9. The mass of each suspension system beam is determined by multiplying the cross-sectional area of the beam by its length, and then multiplying by a factor of two to account for unmodeled components such as standup motors, hinges, latches, etc.

-
10. The mass of each drive and steering motor is assumed to be linearly related to the mass of the rover. When motor mass data for MER and MSL become available, the fidelity of this model can be improved.

$$m_{motor} = 0.05 \frac{m_{rover}}{100} \quad (3.4.13)$$

11. The total mass of the mobility system is the sum of the four beam masses (two of length L_1 and two of length L_2), six wheel masses, ten motor masses (six drive and four steering motors) and the mass of the differential.
- 12.

3.4.8.2 Structural calculations – mast

The mast design algorithm is a low-fidelity placeholder. The design procedure for the mast is as follows:

1. Determine the mass of the instruments at the top of the mast.
2. Using the Bernoulli-Euler relation for a simply-supported beam, and given a max deflection δ , the moment of inertia, I , of the mast can be determined. The force N , is equal to the gravitational acceleration on Mars times the mass of the instruments at the end of the mast, and E is the modulus of elasticity. The factor of safety, f_s , is set to 5.

$$I = f_s \frac{NL^3}{E\delta} \quad (3.4.14)$$

3. Assume a square cross-section mast. Given a ratio of wall thickness to width equal to 0.5, the dimensions of the mast are determined by

$$\begin{aligned} I &= \frac{1}{12}bh^3 - \frac{1}{12}(b-t)(h-t)^3 \\ &= \frac{1}{12}(3bth^2 - 3bht^2 + bt^3 + th^3 - 3h^2t^2 + 3ht^3 - t^4) \end{aligned} \quad (3.4.15)$$

4. The total mass of the mast is equal to the mass of the beam, which can be determined from the dimensions and the material properties, plus the mass of the motor used to raise the mast, which is assumed to be 1 kg.

3.4.8.3 Structural calculations - arm

The arm design algorithm is a low-fidelity placeholder. The design procedure for the arm is as follows:

1. Determine the mass of the instruments and tools at the end of the arm.
2. Select the number of elbow-type joints. The default number of elbow joints is one.
3. Using the Bernoulli-Euler relation for a simply-supported beam, and given a max deflection, δ , the moment of inertia I of the outermost arm segment can be determined. The force N is equal to the gravitational acceleration on Mars times the mass of the instruments at the end of the arm, and E is the modulus of elasticity. The factor of safety, f_s , is set to 5.

$$I = f_s \frac{NL^3}{E\delta} \quad (3.4.16)$$

4. Assume a square cross-section arm. Given a ratio of wall thickness to width equal to 0.5, the dimensions of the mast are determined by

$$\begin{aligned} I &= \frac{1}{12}bh^3 - \frac{1}{12}(b-t)(h-t)^3 \\ &= \frac{1}{12}(3bth^2 - 3bht^2 + bt^3 + th^3 - 3h^2t^2 + 3ht^3 - t^4) \end{aligned} \quad (3.4.17)$$

5. The mass of the arm segment can be determined from the dimensions and the material properties.
6. For the next outermost arm segment, the tip mass is the mass of the instruments and tools plus the mass of the outermost arm segment. The same equations are used to calculate the dimensions. This process repeats for all arm segments, where at each step the tip mass is the sum of the instrument and tool mass and the masses of all previously calculated arm segments.
7. The total mass of the arm is the sum of the masses of the arm segments, plus the masses of each of the arm motors, which are currently assumed to be zero.

3.4.8.4 Structural calculations – plate bending

As explained in the assumptions section, the plates are designed to meet a maximum deflection requirement. This condition determines the plate's flexural rigidity and thickness. This method applies to both uniform and sandwich plates, but the equations are different. First for a rectangular uniform plate with built-in edges uniformly loaded, the maximum deflection is given by [ST-59]:

$$\Delta_{Max} = C q a^4 D \quad (3.4.18)$$

Where C depends on the plate's aspect ratio (see table 35 of [ST-59]), q is the load per unit area, a is the length of the small side of the plate and D is the flexural rigidity. Given Δ_{Max} , D is then known and the following equation is used to find the plate thickness, t (see Equation 3 of [ST-59]).

$$D_{unif} = \frac{Et}{12(1-\nu^2)} \quad (3.4.19)$$

Where E and ν are the Young modulus and the Poisson coefficient of the plate respectively and t is the thickness. If the plate is a sandwich structure, the core thickness is first estimated and then the maximum deflection condition is applied to find the skin thickness. The core thickness is determined by following a weight-minimization expression [HEX].

$$t_c = \sqrt{\frac{w_c \beta q a^2}{2w_s F_s}} \quad (3.4.20)$$

Where w_c and w_s are the densities of the core and skin respectively, F_s is the allowable facing stress and β depends on the aspect ratio and is conservatively set to 0.12. The relation between the maximum deflection and the flexural rigidity then is represented by [HEX]:

$$\Delta_{Max} = \frac{16qa^4}{\pi^6 D} C_1 \quad (3.4.21)$$

Where C_1 depends on the aspect ratio but is conservatively set to 2.

Hence, the given maximum deflection determines flexural rigidity, which is related to the plate's total thickness t by the following equation [HEX].

$$D_{sandwich} = \frac{E}{12(1-\nu^2)} \left(t^3 - t_c^3 \left(1 - \frac{E_{Core}}{E_{skin}} \right) \right) \quad (3.4.22)$$

Where t_c is the core thickness and t the total plate thickness

Finally the skin thickness is just half of the difference between the total and core thicknesses. Knowing the plate's thicknesses, dimensions and material densities, the mass is easily deduced.

3.4.8.5 Structural calculations – plate buckling

The same kind of method is used for the design of a plate under compression. The plate's thickness is sized so that the actual compressive load is less than the critical load given in [ST-40] section 64.

$$\begin{cases} N_{Crit} = \frac{4\pi^2}{3} Da^2 \left(\frac{3}{a^4} + \frac{3}{b^4} + \frac{2}{a^2b^2} \right) \\ N_{Crit} \geq f_s N_{Plate} \end{cases} \quad (3.4.23)$$

Where a is the plate side in the direction of the compression, b is the other side, N_{Wall} is the actual compressive load and D is the flexural rigidity as formerly defined. The second equation represents the design condition where f_s is a factor of safety. Hence, the combined equations give a value for D and consequently for the thickness.

3.4.8.6 Structural calculations – plate tension

This time the plate is designed to resist a tension load. The plate must be thick enough so that the actual tensile strength acting on it is less than the ultimate tensile strength.

$$t = \frac{f_s T_{plate}}{b T_{ult}} \quad (3.4.24)$$

Where t is the designed thickness, T_{plate} is the tension load, T_{ult} is the material ultimate stress and b is the length of the plate perpendicular to the tension direction.

3.4.8.7 Thermal calculations

Thermal transfer occurs by three primary methods: conduction, convection, and radiation. All three of these heat transfer methods are included in the WEB thermal model.

Heat transfer through a flat object (such as a wall) by conduction is proportional to the difference between the surface temperatures at each side of the object, and can be modeled by (Eq. 3.4.25), where Q is the heat transfer rate, L is the thickness of the material, A is the area of the material, and the temperature-independent conduction coefficient k depends on the properties of the material.

$$Q = \frac{kA}{L}(T - T_0) \quad (3.4.25)$$

Heat transfer by convection is proportional to the difference between the surface temperature of the object and the ambient temperature. Convective heat transfer can be modeled by the following equation, where Q is the heat transfer rate, A is the area of the material, and the convection coefficient h depends on environmental factors such as wind speed and the density and chemical composition of the convecting medium.

$$Q = hA(T - T_0) \quad (3.4.26)$$

Heat transfer by radiation depends on the difference between the fourth powers of the surface temperature of the object and the ambient temperature. The material may have different emission and absorption properties, which are characterized by the emissivity coefficient ε and the absorptivity coefficient α . Radiative heat transfer can be modeled by the following equation, where Q is the heat transfer rate, σ is the Stephan-Boltzmann constant, and A is the area of the material.

$$Q = \sigma A(\varepsilon T^4 - \alpha T_0^4) \quad (3.4.27)$$

3.4.9 Program Flow

Figure 3.4.6 illustrates the general program flow of the functions handled by the Rover module. The Rover module is split into two distinct pieces in the functions `rover_init` and `rover`. The only inputs required by the initialization function `rover_init` are the wheel diameter, which is part of the Design Vector, and the payload, which is part of the Science Vector. This function determines the footprint and maximum speed of the rover, which are required by the Environment and Autonomy modules, respectively. Outputs from the Environment and Autonomy modules are then passed as inputs to `rover`, in which the majority of the design is performed. Note that in Figure 3.4.6, `rover_init` is shown below the mobility module for drawing convenience.

The responsibilities of the `rover` function are split between design of the WEB, thermal system, mobility system, and the arm and mast, and are performed by the modules `structure`, `thermal` (detailed in a separate section), `mobility`, `arm`, and `mast`, respectively. A loop between the structural and thermal designs of the WEB ensures that the thermal regulation system fits inside the WEB. The loop is terminated when the masses of two consecutive rover designs differ by less than 1 kg.

Most outputs of the Rover module are copied directly from the outputs of one of the Rover system sub-modules. The total size and mass of the rover are calculated directly in `rover.m`.

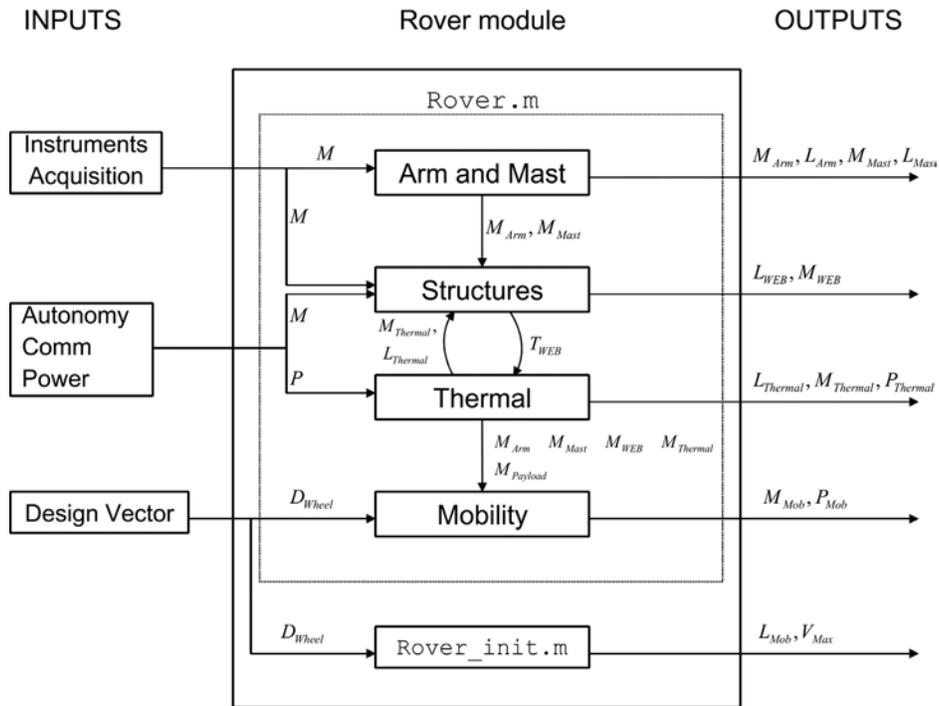


Figure 3.4.6: Program flow for the Rover module and sub-modules. M , P , L and T stand for mass, power, size and thickness, respectively.

3.4.9.1 Thermal module algorithm

The thermal module considers four different temperature pair cases, and uses these cases to calculate the thermal requirements of the vehicle. The four cases and the quantity calculated from each of them are:

1. Ambient $T_{day,max}$ and max payload temperature limit \rightarrow mass of heat pipes and RHUs
2. Ambient $T_{day,avg}$ and min operating temperature limit \rightarrow daytime average heater power
3. Ambient $T_{night,avg}$ and min survival temperature limit \rightarrow nighttime average heater power
4. Ambient $T_{night,min}$ and min survival temperature limit \rightarrow nighttime maximum heater power

More specifically, the four cases can be described in terms of heat transfer rates, Q , as follows:

1. $T_{day,max}$ and max payload temperature
 - $Q_{passive,max} = Q_{day,max} - Q_{payload,day}$
 - $Q_{passive,max} > 0$ is RHUs
 - $Q_{passive,max} < 0$ is heat pipes
2. $T_{day,avg}$ and min operating temperature
 - $Q_{heater,day} = Q_{day,avg} - Q_{payload,day} - Q_{passive,day,avg}$
3. $T_{night,avg}$ and min survival temperature
 - $Q_{heater,night} = Q_{night,avg} - Q_{payload,night} - Q_{passive,night,avg}$
4. $T_{night,min}$ and min survival temperature
 - $Q_{heater,max} = Q_{night,min} - Q_{payload,night} - Q_{passive,night,min}$

The algorithm used to determine the heat transfer rate for each case is as follows. For each case, perform the following iteration to determine the total power, Q (depending on case, one of $Q_{day,max}$, $Q_{day,avg}$, $Q_{night,avg}$, $Q_{night,min}$).

1. Make an initial guess for Q .
2. Given the conditions of the current case (e.g. day or night), calculate the payload waste heat and the ambient temperature, T_i . Given the current guess for Q , determine the temperature on the outer surface of the rover structure, T_3 , taking into account convective and radiative heat transfer:

$$Q = hA(T_3 - T_i) + \sigma A(\epsilon T_3^4 - \alpha T_i^4) \quad (3.4.28)$$

This equation cannot be easily solved for T_3 , so an iterative technique is used wherein guesses are made for T_3 until the correct value of Q is returned (to a specified tolerance).

3. Given a guess for Q and the corresponding T_3 , the temperature on the inner surface of the WEB structure, T_2 is determined, taking into account conductive heat transfer.

$$\begin{aligned} Q &= \sum \left(\frac{kA}{L} (T_2 - T_3) \right) \\ &= (T_2 - T_3) \sum \frac{kA}{L} \end{aligned} \quad (3.4.29)$$

The summation allows for different walls in the rover WEB structure to have different thicknesses. The assumption is made that the internal and external temperatures are the same for all surfaces. The temperature T_2 can be determined analytically.

4. Given the current guess for Q and the surface temperature of the WEB, T_2 , determine the temperature of the payload, T_1 , taking into account convective and radiative heat transfer inside the WEB:

$$Q = hA(T_1 - T_2) + \sigma A(\epsilon T_1^4 - \alpha T_2^4) \quad (3.4.30)$$

This equation cannot be easily solved for T_1 , so an iterative technique is used wherein guesses are made for T_1 until the correct value of Q is returned (to a specified tolerance).

5. Check the calculated value of T_1 against the case value (i.e. the limit or average value, as appropriate). If the calculated T_1 and the case T_1 differ by more than some tolerance, adjust the guess for Q and return to step 2.

Perform these operations for all four cases to get the RHU power, heat pipe power, average daytime heater power, average nighttime heater power, and max nighttime heater power. The mass of each thermal system component can be determined from the power requirements by applying scaling factors.

The following files contain functions used by the thermal module algorithm:

- `thermal.m` – manages the algorithm outlined in this discussion. Returns the mass and power required for heating and cooling the rover’s warm electronics box.
- `heat_convection_radiation.m` – determines a heat transfer rate, given all other parameters for convective and radiative heat transfer.
- `temp_convection_radiation.m` – determines an object surface temperature, given all other parameters for convective and radiative heat transfer.

3.4.10 Validation

3.4.10.1 Mobility – rover maximum speed

The main assumptions that need to be validated for the mobility system concern the speed and power models. The maximum speed of the rover is defined as a function of the wheel diameter by a curve fit to existing data points, with saturation for large wheel diameters. This curve fit is shown in Figure 3.4.7. Because very few data points exist, the curve fit is somewhat arbitrary; however, no better approach to modeling the maximum speed has been determined. Given this uncertainty, it is of interest to assess how robust the overall rover model is to uncertainties in the maximum speed calculations.

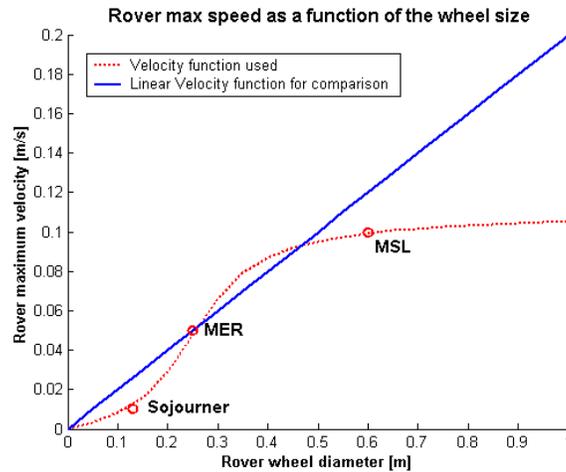


Figure 3.4.7: Comparison of maximum velocity models

The average speed is defined over a driving cycle that includes actual driving time (at the maximum speed) and standing still time during which the navigation sensors and algorithms are working (see Figure 3.4.8).

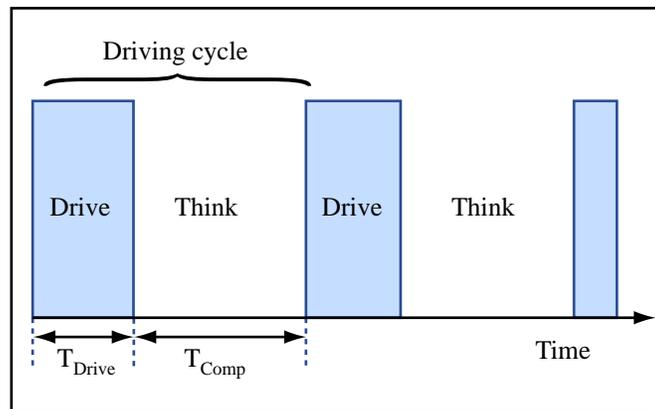


Image by MIT OpenCourseWare.

Figure 3.4.8: Driving Cycle

Assuming that the rover is moving forward by half a rover length at each cycle, and that it takes 45 seconds for the navigation instruments to compute an area of 0.09 m^2 [CWh], the expressions for driving and thinking times are:

$$\begin{cases} T_{Drive} = \frac{1}{2} \frac{L_{Rover}}{V_{Max}} \\ T_{Comp} = \frac{45}{0.09} \frac{1}{2} L_{Rover} W_{Rover} \end{cases} \quad (3.4.31)$$

Where L_{Rover} and W_{Rover} are the length and width of the rover, respectively. From these quantities, the average speed as a function of maximum speed is:

$$V_{avg} = \frac{V_{Max}}{1 + \frac{45}{0.09} W_{Rover} V_{Max}} \quad (3.4.32)$$

As the maximum speed increases, the average speed reaches a limit equal to $0.09/45 \times W_{Rover}$ m/s. As a consequence, average velocity is not very sensitive to uncertainties in maximum velocity. This insensitivity is illustrated in the following example.

The chosen speed relation is compared to a more simple linear law that fits the MER data point (see Figure 3.4.7). For a diameter of 1 m, the linear velocity law returns a maximum speed twice as large as that returned by the curve fit equation. Inaccuracies in the model for maximum speed have little effect on the average speed, as shown in Figure 3.4.9. The largest difference in average velocity estimates occurs for a 0.05m diameter wheel, and the difference is negligible (0.2 mm/s). Therefore, even if the maximum velocity model is inaccurate, the effects of this inaccuracy on the rover model are negligible.

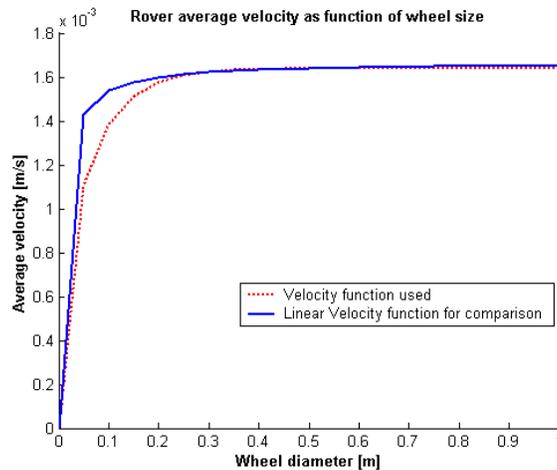


Figure 3.4.9: Comparison of average velocities

3.4.10.2 Mobility – drive power

The driving power model is compared to existing rover power requirements in Table 3.4.3.

Table 3.4.3: Actual and modeled drive power per wheel, in Watts.

Rover	Actual ¹	Modeled
Marsokhod 75	70	76
MER	17	18
ExoMars	12.5	14
Sojourner	1.7	0.2

The power equation used does not account for soil interactions such as slip, or for mechanical losses such as friction torque in the wheels. Still it manages to model all cases to within 2 W, except for the Marsokhod 75. For rovers larger than Sojourner in mass, it overestimates power consumption and should be refined in order to be more consistent.

3.4.10.3 WEB and mobility mass

Unfortunately, no accurate data were found about existing rover WEB and mobility masses; however, according to a general design guideline the mass of the WEB plus mobility system is 40% of the “core equipment/payload mass” [CWh]. The phrase ‘core equipment/payload’ is assumed to mean total mass. Still the model gives in the case of the MER scenario a WEB mass of 6.5 kg and a mobility mass of 42 kg for a total rover mass of 120 kg. The total rover mass estimate is lower than the actual mass of 185 kg; however, the modeled mass of the mobility plus suspension is effectively 40% of the total mass.

3.4.10.4 Thermal

Figure 3.4.10 shows an example plot that serves to validate one aspect of the thermal model. The example payload (e.g. the instruments) levy extremely tight requirements on the thermal system; namely, that the payload temperature must remain fixed at 250 K regardless of the external temperature. The plot shows the surface temperature on the inside and outside surfaces of the WEB structure (they are almost identical due to the efficiency of conduction, and only one is visible in the plot) and the thermal power required to maintain the payload at a constant temperature. As is expected, as the ambient temperature increases, the amount of power required for heating decreases. At approximately 170 K, the heat output of the payload is sufficient to maintain the temperature of the payload at the desired level, and for ambient temperatures hotter than 170 K, cooling is required. Qualitatively, this is the expected behavior for the algorithm. There are not enough data on rover thermal systems to make a quantitative comparison between the final outputs of the thermal model and proven rover designs.

¹ Data for rovers: Marsokhod in [CNE], MER in [MER], ExoMars in [EXO], Sojourner in [MIS]

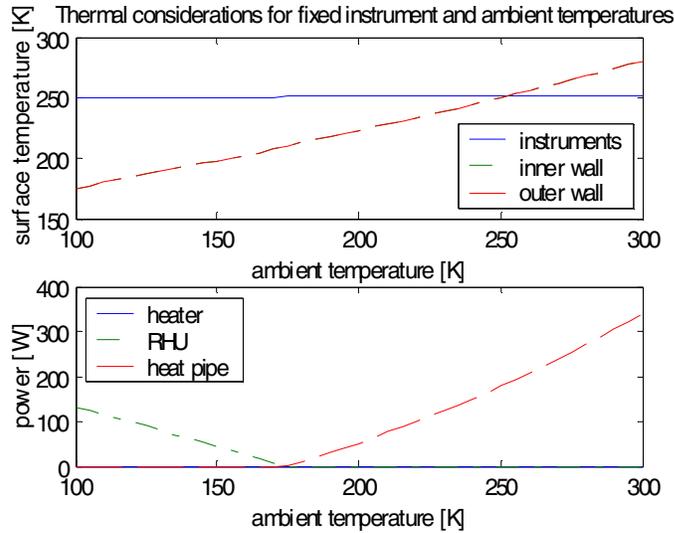


Figure 3.4.10: Heater and heat pipe power as a function of steady-state ambient temperature.

3.4.11 Expandability

Several aspects of the mobility, structure, and thermal models can be improved.

3.4.11.1 Mobility

Improvements identified for the mobility system are as follows.

- Improve the drive power estimate. Include soil interactions for the power equation by using Bekker equations for trafficability [HVF].
- Improve the rover maximum speed relation. Develop an analytical model and use the available data on Sojourner, MER and MSL for validation. For a vehicle driving on uneven terrain, the maximum speed is related to vehicle vibrations (see Figure 3.4.11). Hence requirements on maximum vehicle vibrations would limit its maximum allowable speed.

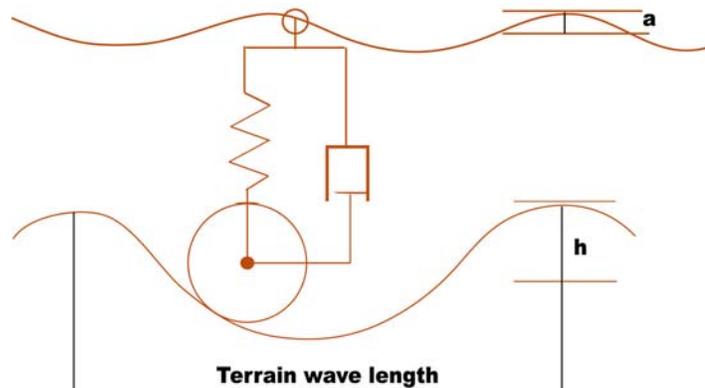


Figure 3.4.11: Response factor of forced vibrations

Assuming the terrain has a waveform of wavelength l_w , the general equation of the road wave is given by [MGB]:

$$\begin{cases} x = \frac{h}{2}(1 - \cos(\omega_g t)) \\ \omega_g = 2\pi \frac{V}{l_w} \end{cases} \quad (3.4.33)$$

Where h is the wave height and V the vehicle speed. The response factor of forced vibration is then

$$\begin{cases} \frac{a}{h} = \frac{\sqrt{1 + 4 \frac{D^2}{\lambda^2}}}{\sqrt{(1 - \lambda^2)^2 + (4D^2 \lambda^2)}} \\ \lambda = \frac{\omega_g}{\omega_n} \\ D = \frac{\rho}{2m\omega_n} \\ \omega_n = \sqrt{\frac{c_s}{m}} \end{cases} \quad (3.4.34)$$

Where ρ and c_s are the suspension damping factor and spring constant, respectively. The vehicle velocity is related to ω_g in Eq.3.4.33 and therefore to λ in Eq. 3.4.34. Hence from requirement on the maximum response factor, a/h follows a limitation on the maximum allowable velocity.

- Refine the structural design of the WEB. The wall design needs to capture the moments due to the externally attached hardware. The drill option remains to be modeled properly, so far it is just considered as a load on the WEB bottom plate. The model also needs to take into account manufacturability and packaging issues.

3.4.11.2 Thermal

There are multiple ways in which the thermal algorithm accuracy could be improved.

1. If passive power is determined to be less than zero in the worst-case hot temperatures, then a heat pipe is required. The algorithm determines the heat transfer rate required to maintain the instruments at their maximum temperatures. The algorithm then uses this same heat transfer rate when calculating the temperature characteristics for average and minimum temperature cases. This approach is incorrect, as the heat pipes should be sized based on the maximum required heat transfer rate, but the heat transfer rate varies based on the ambient and internal temperatures at any particular time. This error should lead to overestimates of the mass and power required for heating.
2. The thermal module does not consider the thermal influence of the power system. If unused energy from the solar array or RTG passes through the WEB as excess heat, the outputs of the

module may change dramatically. This is not an issue if unused energy is not converted to heat inside the WEB.

3. In step 3 of the algorithm, the external and internal temperatures are assumed to be constant on all sides of the structure. In reality, different walls may have different temperatures.
4. The convection coefficient currently in use in the module ($k=0.5 \text{ W}/(\text{m}^2 \text{ K})$) is a guess. Depending on the accuracy of this guess, replacing this number with a measured or modeled number could dramatically affect the results.

The algorithm that calculates the temperature due to convective and radiative heat transfer (in `temp_convection_radiation.m`) is extremely slow and currently dominates the processing requirements of the complete design tool, requiring 93% of the calculation time. Implementing a more efficient algorithm for convective and radiative transfer would result in nearly a one to one gain in the overall speed of the tool.

3.4.12 References

- [AEI] Ellery, A., An introduction to Space Robotics, Springer-Praxis, April 2000.
- [CWh] Charles Whetsel document.
- [CNE] "Missions, technologies and design of planetary mobile vehicles," CNES, Toulouse, September 1992.
- [EXO] Gardini, B., "CDF study Report ExoMars 09."
- [HEX] "Design Handbook for honeycomb sandwich structures" printed by Hexcel, 1970.
- [HVF] Heiken, G.H., Vaniman, D.T., and French, B.M., Lunar sourcebook: a user's guide to the Moon, Cambridge University Press, 1991.
- [HWS] Stone, H.W., "Mars Pathfinder microrover a small, low-cost, low-power spacecraft," Jet Propulsion Laboratory.
- [IDF] Incropera, F.P. and DeWitt, D.P., Fundamentals of heat and mass transfer, fourth edition, John Wiley & Sons, 1996.
- [JMa] Matijevic, J., "Mars Pathfinder Microrover – Implementing a low cost planetary mission experiment," Jet Propulsion Laboratory.
- [JRW] Wertz, J.R. and Larson, W.J., 'Space Mission Analysis and Design', third edition, Space Technology Library.
- [MIS] Mishkin, "Experiences with Operations and Autonomy of the Mars Pathfinder Microrover."
- [MER] Roncoli, R.B., Mars Exploration Rover (MER) project mission plan, Jet Propulsion Laboratory, April 24 2002.
- [MGB] Bekker, M.G., Theory of land locomotion; the mechanics of vehicle mobility, University of Michigan Press, 1956.

[ST-40] Timoshenko, S., Theory of plates and shells, first edition, McGraw-Hill, 1940.

[ST-59] Timoshenko, S. and Woinowsky-Krieger, S., Theory of plates and shells, second edition, McGraw-Hill, 1959.

3.5 Power

3.5.1 Responsibilities

The role of the Power module is to analyze and model various power systems that can satisfy the power requirements needed for a Mars rover mission. Power is needed to keep computers and scientific instruments operating, to communicate with Earth and to drive the rover. In addition, the power system must be able to survive Mars' harsh environment and function long enough for the mission to accomplish its science goals.

As inputs to the Power module, all subsystems' average and peak power requirements for daytime and nighttime operation, along with the duration for which the power needed to be supplied were collected. The Power module outputs the type of hardware (solar panel, battery, RTG, power interfaces); the mass, size, and location for each hardware piece (web, deck, external); the time required for battery recharge; and the total cost of the power system.

3.5.2 Assumptions

There are several main assumptions made in the Power Module:

Silicon solar cells: Based on research done by the Photovoltaic & Space Environments Branch at the NASA Glenn Research Center, silicon solar cell technology has been chosen. The module can also test GaAs or multijunction solar cells.

Nickel-Hydrogen batteries: This choice is made to decrease the total possible number of architectures. Switching to Nickel cadmium, lithium ion or sodium-sulfure is also allowed by the code. Some features of the Ni-H battery that were found to be advantageous for energy storage systems in aerospace applications are as follows: long life cycle, exceeds all other maintenance free batteries, high specific energy (gravimetric energy density), high power density, tolerant to overcharge and reversal, and no memory effects.

Area ratio constraint: A restriction applied to the solar cell module is that the solar cell area cannot be bigger than the web footprint size multiplied by a constant.

Depth-of-discharge: The battery depth-of-discharge for the solar option is assumed to be 75%, while that for the RTG option was 50%. The decision was based on the fact that the selection of an RTG architecture will most probably be accompanied by longer mission life, which means more cycles are expected.

Cost models: For solar array cost calculation, the AIAA 994066 paper (Solar Cell Array System Trades - Present and Future, E. Ralph and T. Woike) and AIAA paper 99-1066 were used. Joe Parrish provided RTG cost estimates. The battery cost model used is from SMAD.

MMRTG is the RTG of choice: It is assumed that the most probable choice should be the Multi-Mission Radioisotope Thermoelectric Generator (MMRTG) and not a Stirling Radioisotope Generator (SRG) or the Cassini-type RTG for the following reasons: 1) The Cassini-type RTGs use too much Pu-238 and will probably be out of use after 2005; 2) SRGs are much more efficient than MMRTGs but their current Technology Readiness Level (TRL) is only 4, while MMRTGs have a TRL of 7. If desired, Cassini or SRG specifications can be tried by the code as well.

MMRTG will be technologically validated by 2009 and could be used for later missions.

Nuclear reactors will not be considered for recent Mars rover missions: This is not only because of the enormous mass and size of the power system, but mostly because of the technology readiness level of such devices and policy issues.

3.5.3 Background

3.5.3.1 Solar cells

Mars receives approximately 44% as much solar radiation as Earth, and therefore solar power is feasible as a power source. The amount of solar energy available on Mars is shown in Figure 3.5.1. As can be seen, Mars is almost the farthest planet from the sun where a solar option is still a feasible option.

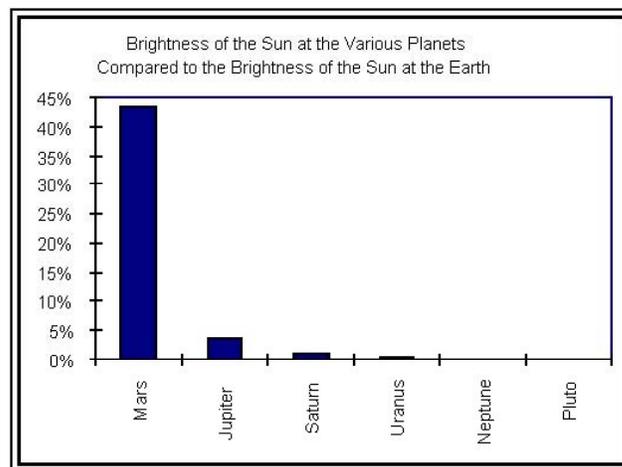


Figure 3.5.1: The amount of solar energy available on Mars

The conventional approach to the design of the solar array is using deployable high-efficiency flat plate arrays. These are what are modeled in the tool. Three main types of solar cells have been considered in the Power module. These are:

- Crystalline silicon (Si) with 14.5% efficiency
- Gallium arsenide (GaAs) with 18.5% efficiency
- Multi junction with 15% efficiency

In the design of the solar panel for Mars surface missions, many factors should be taken into consideration. These are as follows:

- *Suspended atmospheric dust*
The atmospheric dust does not remain suspended in the atmosphere indefinitely, but deposits out of the atmosphere onto any horizontal surfaces. This dust deposits out of the atmosphere and onto any flat surface; the time scale for this settling has been measured to be on the order of one hundred sols.
- *Low operating temperatures*

Orbital solar arrays typically operate at temperatures between 50° and 100° Celsius. On Mars, however, the temperature is lower than the standard temperature, rather than higher. This means that a high temperature coefficient is in fact desirable, since this leads to higher efficiency at Mars temperatures than at test temperatures. This shifts the technology choice toward lower band gap materials and away from higher band gap materials.

- *Deposition of dust on the arrays*
The suspended dust limits the lifetime of the solar cells. Unless a cleaning mechanism is designed to remove the deposited dust, the solar panel will cease function after a specific amount of time. The worst-case scenario would be landing in a dust storm.
- *Wind loading*
Wind is an array design issue. The low atmospheric density on Mars means that dynamic pressures are low, which makes wind far less of a problem on Mars than on Earth. In most cases, wind loading will not be a major problem.
- *Peroxide components of the soil*
The soil and dust of Mars contain a highly energetic oxidant, presumably produced by the action of ultraviolet light on the soil. This fact might lower the performance of the solar array with time.
- *Radiation*
The radiation environment includes ultraviolet (UV) and particulate radiation (primarily high-energy protons and electrons). The radiation environment of Mars is actually quite benign compared to the usual orbital environment, since Mars has no trapped radiation belts, and the Martian atmosphere serves as a mass shield against coronal mass-ejection ("solar flare") events. The shielding provided by the Martian atmosphere is sufficient that it could be possible to use a solar array without the conventional glass cover for protection against radiation. However, it must be cautioned that in this case, radiation degradation during the cruise to Mars may be a factor.
- *Low atmospheric pressure*
The atmosphere of Mars consists of primarily carbon dioxide, at a pressure slightly lower than 1% of the pressure at the Earth's surface, varying somewhat with landing site elevation and season. This atmospheric pressure is close to the Paschen minimum for plasma breakdown, and thus sets a significant limit to the maximum voltage that can be applied to any exposed conductors. This fact should be considered in design of the solar panels.

3.5.3.2 Batteries

Batteries are an essential part of solar panel system design. The parameters important in battery design are specific energy (W-h/kg), energy density (W-h/liter), ampere-hour capacity, rechargeability, depth of discharge (DOD), lifetime, temperature environments, ruggedness, and mass. The secondary batteries are designed in combination with the solar panel. There is a choice between secondary batteries for energy storage as shown in Table 3.5.1.

Table 3.5.1: Secondary battery types available

Secondary Battery	Specific energy density (W-hr/kg)
Nickel-Cadmium	25-35
Nickel hydrogen	30
Lithium-Ion	70
Sodium-Sulfur	140

The following parameters also should be determined:

- Mission length
- Night frequency
- Night length
- P_e = Power required at the time of Martian night
- Depth of discharge
- Duty cycles

3.5.3.3 *Solar cell and battery calculation*

Power profiles are analyzed to size the area of the solar panel and the batteries. In Figure 3.5.2, a simple power profiling for the rover design is shown. The power consumption pattern consists of driving cycles, thinking cycles, communication recharging cycles, communication cycles and nighttime power. These cycles are defined as:

Driving cycles: the amount of power necessary for the rover when it is driving

Thinking cycles: the amount of power necessary for processing information when the rover is not moving and is planning for the next move

Communication cycles: the amount of power necessary for the rover when it is not moving, but is communicating

Communication recharging cycle: the amount of time that the rover is not moving or communicating, but is recharging its communication batteries.

Nighttime power: the amount of power necessary for housekeeping during the night. No nighttime navigation has been considered.

A schematic of the power profiling is shown in Figure 3.5.2.

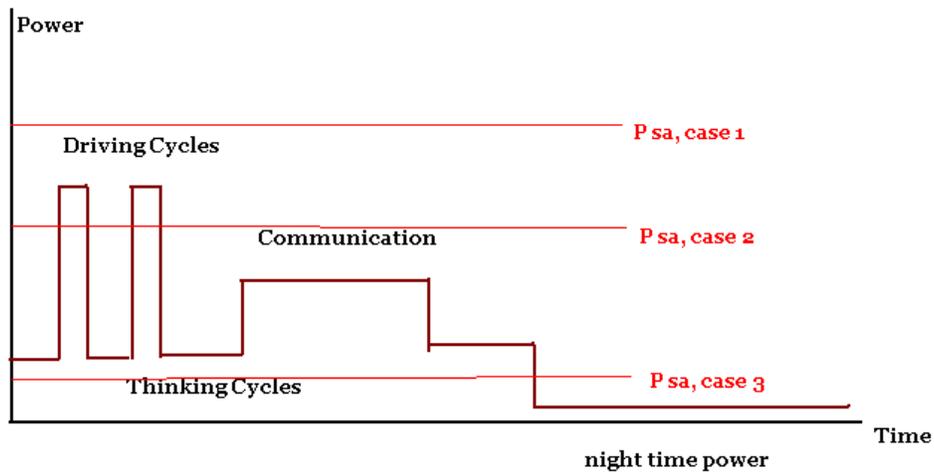


Figure 3.5.2: Power profiling for Mars surface rovers

Three different cases are considered for calculation of the solar panel and battery mass based on the maximum power level. In case 1, the power necessary for operation of the rover is less than the maximum power that can be provided by the maximum allowed solar panel area. In this case, design of the solar panel and batteries are feasible. In case 2, as is true for most of the cases, adjusting the energy required for the rover with consideration of power profiling, the design of a solar panel is feasible with extra batteries on board. In this case, extra batteries are designed for use in each driving and communication cycle. In case 3, the maximum power that can be provided is less than the minimum power during the day. In this case, the design of additional batteries for operation of the rover is not helpful, because there is not enough energy absorption to recharge the batteries during the day. In case 3, no solar option is feasible.

The calculations of the solar panel and batteries are as follows:

For peak power tracking:

$$X_e = 0.6$$

$$X_d = 0.8$$

Performance degradation = 3.75% for silicon per year
 2.75% for GaAs per year
 0.5% for Multi junction per year

P_{sa} = The amount of power that the solar cell should provide for rover

$S_{intensity}$ = Solar intensity at a specific region on mars

P_o = Power estimate out put

$K_{silicon}$ = 0.148

K_{GaAs} = 0.185

$K_{Multijunction}$ = 0.22

I_d = inherent degradation

θ = Worst case angle of the sun

n_1 = Number of driving cycles

n_2 = Number of communication cycles

T_e = Night duration

T_{drive} = Driving duration in each cycle

T_{Comm} = Communication duration

T_{Think} = Autonomy think time duration

P_{night} = The amount of power that the solar cell should provide for the rover during the night

P_{High} = The highest amount of power in a power profile when the rover is roving in daytime

P_{Low} = The lowest amount of power when the rover is thinking and not moving

P_{Comm} = The amount of power needed for the time that rover is communicating

N = Number of batteries

n = Transmission efficiency between battery and load

DOD = Depth of discharged being determined from following graphs.

T_e = Time spent at eclipse

P_e = Power necessary at eclipse

$$P_{sa-high} = \frac{\left(\frac{P_e T_e}{X_e} + \frac{P_{high} n_1 T_{drive}}{X_d}\right)}{T_d} \quad (3.5.1)$$

$$P_{sa-low} = \frac{\left(\frac{P_e T_e}{X_e} + \frac{P_{low} T_{Think}}{X_d}\right)}{T_d} \quad (3.5.2)$$

$$P_{sa-comm} = \frac{\left(\frac{P_e T_e}{X_e} + \frac{P_{Comm} n_2 T_{Comm}}{X_d}\right)}{T_d} \quad (3.5.3)$$

$$P_o = K \times S_{intensity} \quad (3.5.4)$$

$$P_{BOL} = P_o I_d \cos \theta \quad (3.5.5)$$

Life degradation calculation:

$$L_d = \left(1 - \frac{\text{degradation}}{\text{year}}\right)^{\text{Rover's lifetime}} \quad (3.5.6)$$

$$P_{EOL} = P_{BOL} L_d \quad (3.5.7)$$

Calculation of the area needed for solar cell:

$$A_{panel} = \frac{P_{rover}}{P_{EOL}} \quad (3.5.8)$$

Estimation of the mass of solar array:

$$M_{Panel} = 0.04 \times P_{rover} \quad (3.5.9)$$

The Battery Capacity is calculated using:

$$C_r = \frac{P_e T_e}{(DOD) N n} \quad \text{W-hr} \quad (3.5.10)$$

3.5.3.4 Nuclear Power

Figure 3.5.3 shows the power levels and mission lifetime achieved by different types of power systems. As shown, nuclear reactors can provide many orders of magnitude more power. Additionally, they can support significantly longer mission lifetimes. Radioisotope thermoelectric generators (RTGs) are the only other viable option when considering missions of long duration.

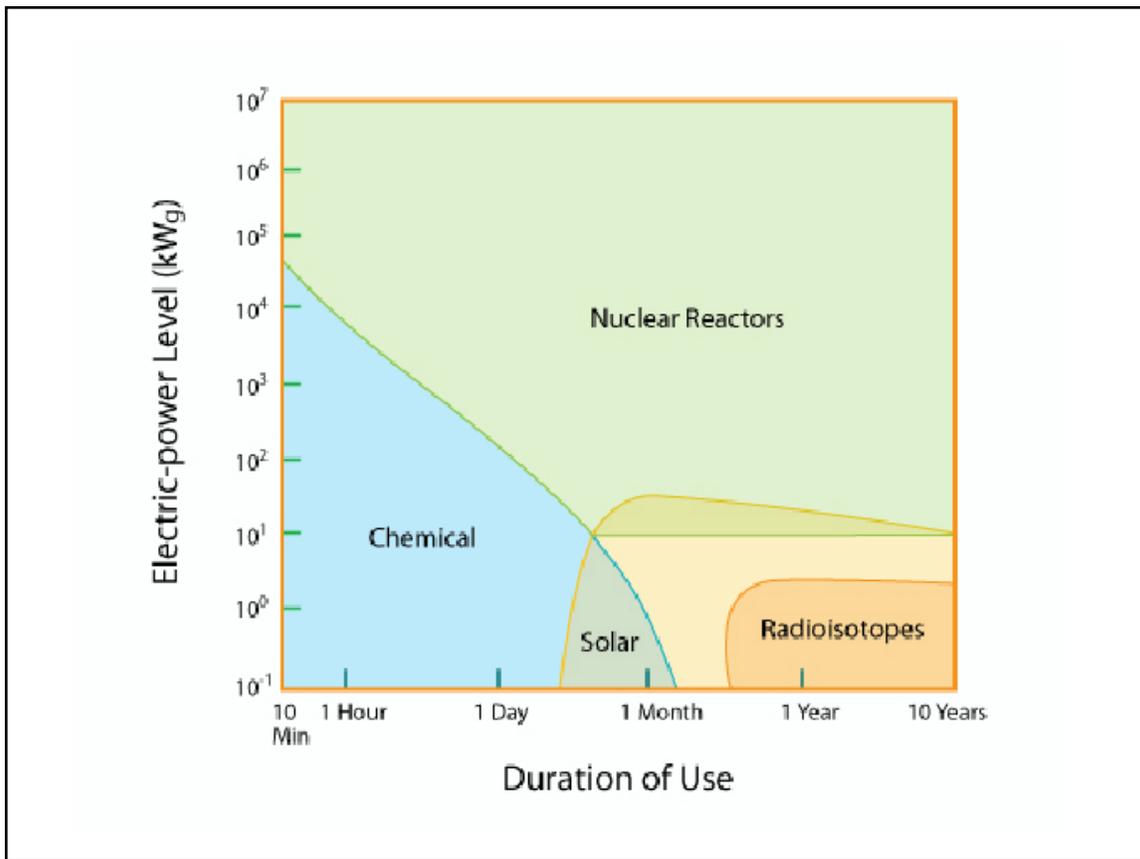


Image by MIT OpenCourseWare.

Figure 3.5.3: Power system comparison

The advantages and disadvantages of RTGS are compared to better see the trade-offs in the power system selection process:

Advantages:

1. Radioisotope power systems do not require any sunlight to operate, permitting the rover to work at a greater variety of scientifically important locations regardless of season, time of day, or latitude. Shadowing in rugged terrain such as within canyons or adjacent to ridges limits the effectiveness of solar-powered landed spacecraft. Radioisotope powered vehicles are not limited by such conditions.
2. They are long lasting and relatively insensitive to the chilling cold of space and virtually invulnerable to high radiation fields.
3. RTGs provide longer mission lifetimes than solar power systems. Supplied with RTGs, the Viking landers operated on Mars for four and six years, respectively. By comparison, the 1997 Mars Pathfinder spacecraft, which used only solar and battery power, operated only three months.
4. They are lightweight and compact. In the kilowatt range, RTGs provide more power for less mass (when compared to solar arrays and batteries).
5. Because there are no moving parts and no moving fluids, conventional RTGs are highly reliable.
6. RTGs are safe and flight-proven. They are designed to withstand any launch and re-entry accidents.
7. RTGs are maintenance free.

Disadvantages:

1. The nuclear decay process cannot be turned on and off. An RTG is active from the moment when the radioisotopes are inserted into the assembly, and the power output decreases exponentially with time.
2. An RTG must be cooled and shielded constantly.
3. The conversion efficiency is normally only 5 %.
4. Radioisotopes, and hence the RTGs themselves, are expensive.

Nuclear reactors and RTGs are the two main types of nuclear power systems.

<http://www.seds.org/spaceviews/cassini/RTG1.gif> RTGs are devices that provide power through the natural decay of radioisotopes. The decay generates heat, which is used by equipment in the RTG to generate electricity. RTGs can supply power of the order of a few watts to a few kilowatts. For higher power demand, nuclear reactors are recommended. They operate based on the concept of nuclear fission. If the nucleus is heavy and unstable, like that of uranium, adding energy destabilizes it even more to a point when it may break up immediately. As mentioned above, however, it is assumed that only RTGs are an option for Mars rovers in the near future, so nuclear reactors are not further discussed in this document.

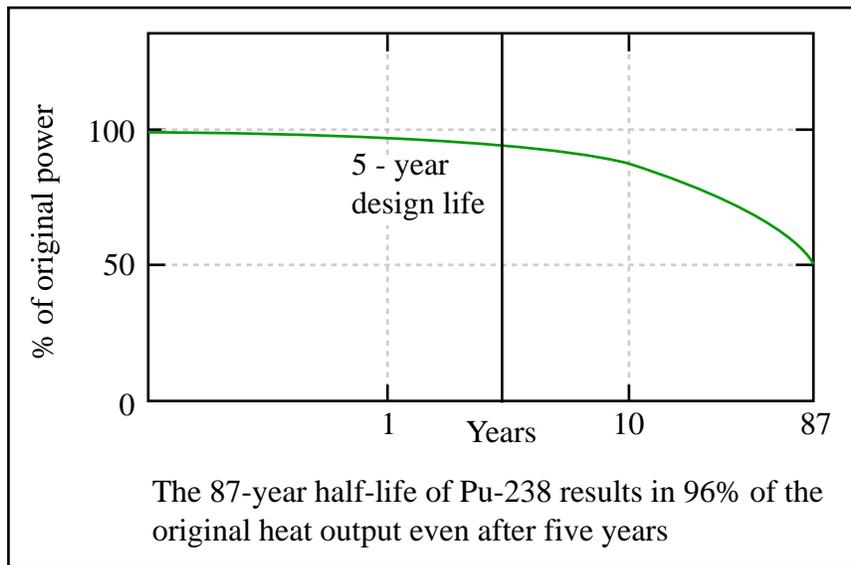


Image by MIT OpenCourseWare.

Figure 3.5.4: Pu-238 power supply

There are different kinds of radioisotopes that can be used in an RTG. Although plutonium is by far the most expensive, it is currently the sole element the US Department of Energy (DoE) uses for RTGs because of its long half-life (Figure 3.5.4), relatively high heat to mass ratio, and low gamma ray emissions.

During the 1950s and 1960s NASA spent billions of dollars to build a nuclear rocket program that was cancelled because of the fear that a launch accident would contaminate major portions of Florida and beyond. Many people also feared that the so-called "peaceful" uses of nuclear power in space, such as nuclear Martian rockets, are merely a cover to develop power systems that can be used for space-based weapons. After a thirty-year shutdown of plans for nuclear rockets, the Bush administration has resuscitated the interest in this technology, budgeting NASA to further its work on space nuclear power and propulsion. The fact of the matter is that nuclear power is safe and can get a spacecraft to the Moon, Mars and elsewhere in the solar system faster than any alternative means. There is little doubt that if public relations efforts gain acceptance for the possibility, future interplanetary missions will definitely include nuclear-power options.

Many years have been invested in the engineering, safety analysis and testing of RTGs. As a first precaution, the plutonium is not placed in pure form in the RTGs but is installed as bricks of plutonium dioxide (PuO₂), a ceramic that, if shattered, breaks into large pieces rather than smaller, more dangerous dust. The plutonium dioxide is encased in layers of materials, including graphic blocks and layers of

iridium. Both materials are strong and highly heat resistant and would serve as a reliable protection for the plutonium bricks in the event of a launch explosion. So far, RTGs have had a remarkable performance record. RTGs have been used nearly two-dozen times in the last 30 years. Some of the missions that included RTGs have been a number of the manned Apollo missions (for use as a power supply for experiments left behind on the lunar surface), Pioneers 10 and 11, Voyagers 1 and 2, the Galileo mission to Jupiter, and Cassini. The Viking landers were also electrically powered by radioisotope power systems. A table listing all US-made RTGs and their specifications can be found in the appendix. The SNAP series RTGs are long out of production, since lead-telluride thermocouples are no longer fabricated. Removing all obsolete technologies and RTG designs that are not expected to be validated soon (e.g. with TRL < 4), leaves only three valid concepts to be considered in the model. Although the first one is referred to as "Cassini type RTGs" (Cassini had three RTGs of this type), it was also used in Galileo (2 RTGs) and Ulysses (1 RTG). The other two options are the Multi-Mission RTG (MMRTG) and the Stirling Radioisotope Generator (SRG). Table 3.5.2 displays the information about these three options.

Table 3.5.2: RTG options for a Mars surface rover mission

Power Source	PBOL [We]	PEOL[We]	Mass [kg]	Dimensions [m]	Life[yr]	Pu[kg]	Cost [M\$]	TRL	Notes
Cassini RTG	285	210	55.5	D = 0.41,L=1.12	10.75	8	35.00	9	18 GPHS
New MMRTG	140	123	32	D = 0.41,L = 0.6	10	4	25.00	7	9 GPHS
SRG 1.0	114	94	27	D = 0.27,L = 0.89	14	0.9	20.00	4	2 GPHS

The General Purpose Heat Source (GPHS) modules are the main building blocks of any of the three RTGs selected for consideration. The Cassini type RTG is composed of 18 GPHS units, the MMRTG - 9 units, and the SRG – 2 units. This modularity might lead one to the conclusion that a wide variety of discrete power levels is possible by selecting a different numbers of GPHS units. Programmatically, however, it is unlikely that the Department of Energy would develop a new RTG just to fit a specific NASA mission. There has been much effort to try to define a reasonable canonical size for RTGs to cover the perceived set of upcoming NASA missions, and a decision was made to go with increments of 100W. For example, if there is a 200W requirement for a mission, a possibility would be to choose to fly two MMRTG's and not to try to design an RTG with 13 GPHS units. The cost of requalifying a new design, even if it is a matter of resizing only, is significant. JPL/NASA would be better off flying two, three, or more validated RTG's than attempting to size a new unit up or down. In addition, there is a large lead time for redesign and requalification, which may not be available.

Another issue that was considered was the possibility of combining different types of RTGs. Technically there is no reason why two different types of RTGs are incompatible, but no real mission will adopt such an option. The costs of maintaining two separate contractors, two testing programs, etc. would overwhelm any benefit. Therefore, this option was discarded and one type of RTG (namely MMRTG) was selected for the design of the nuclear power system.

3.5.4 Program Flow

The Power module consists of two parts: Solar and RTG. Figure 3.5.5 is a schematic representation of the general process flow of the module.

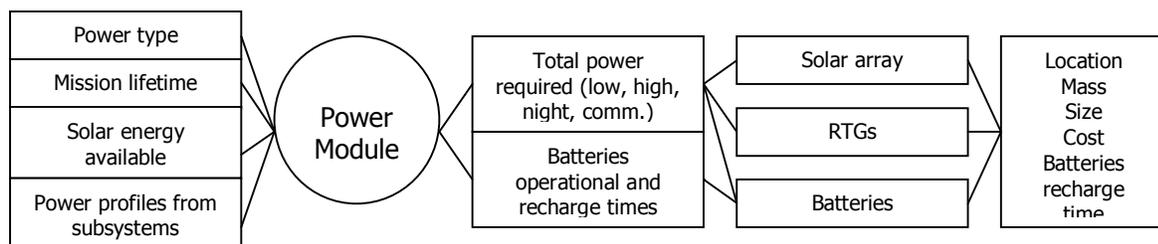


Figure 3.5.5: Process flow of the Power module

The inputs to the Power module are the mission duration and type of power system from the Design Vector, the solar energy available from the Environment module, and all subsystems' power profiles for daytime and nighttime operations. The Power module outputs the type of each individual piece of hardware (solar panel, battery, MMRTG, power interfaces), the location for each type (web, deck, external), the mass and size for each hardware piece, the cost of the entire power system, and the battery recharge time. The intermediate calculations valid for both the solar and the RTG module estimated four types of total power required: low, high, night, and communications. The low power requirement is a sum of all subsystems' average requirements during a sol; the high power estimate is a sum of the peak power demands during the day. The night power accounts for all nighttime power requirements. The communications power calculations need the same inputs as for low power plus the average communications power per sol and the communication power needed by the Autonomy module. The rest of the calculations are specific to the Solar and RTG sub-modules.

3.5.4.1 Solar Module

The solar module logical flow can be seen in Figure 3.5.6.

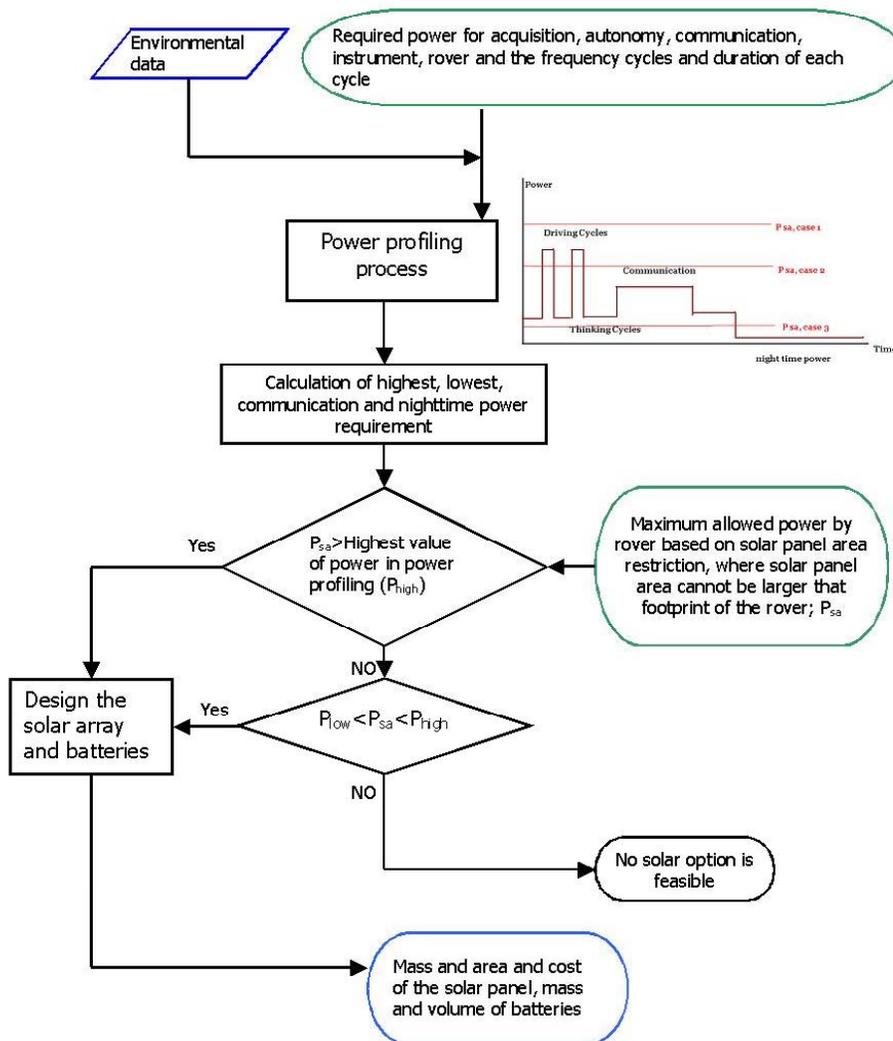


Figure 3.5.6: Process flow of the solar module

3.5.4.2 RTG Module

As already stated, the MMRTG is the assumed RTG of choice. The RTG modeling code can be executed for Cassini RTGs or SRGs as well, however the current code uses the mass, size, design lifetime, cost and end-of-life power of an MMRTG. Thus, the minimum power that the nuclear power system will provide is just under 140 W. If this value is between the “low power” and the “high power” required, the code offers the option of either adding a battery or more RTGs or both to the rover design. The RTG limit set is 1kW, which can be achieved by using 8 RTGs. For power requirements greater than 1kW, nuclear reactors are the recommended power source.

When modeling, the estimated average power is what determines the size of the power source. The peak power drives the size of the energy storage system, in this tool those are the secondary batteries. Using a battery in combination with RTGs is not a common situation. For the Mars surface mission this tool is designed for, however, it is desirable to design for this option because very brief periods of peak power demand are expected. Without considering batteries and choosing a number of RTGs that would

accommodate for the peak power requirements, too much excess power would result. This is undesirable, not only from a cost point of view, but also because of heat dissipation considerations. Designing a battery then becomes one of the most important parts of the RTG module. The same steps are followed as in the solar module, except that night power requirements were excluded from the calculations of battery power and capacity. The same holds true for the calculation of recharge time. After the number of RTGs and the battery characteristics are estimated, the code calculates the size and the mass of the power interfaces (e.g. cables, wires, etc.). An assumption commonly accepted in practice is that the mass of the power interfaces is roughly six percent of the total mass of the power system. A loop is used to calculate this value for each design and then hardware types, sizes, masses, location, and total cost are output to the utility module. The time needed for battery recharge is output to the Autonomy module.

3.5.5 Validation

3.5.5.1 Solar Module

The following graphs have been used to validate the solar cell design module. As can be seen in Figure 3.5.7, the mass of the solar array increases with increasing mission lifetime. The major drive for this behavior is the increase in solar panel degradation and decrease in end of life power.

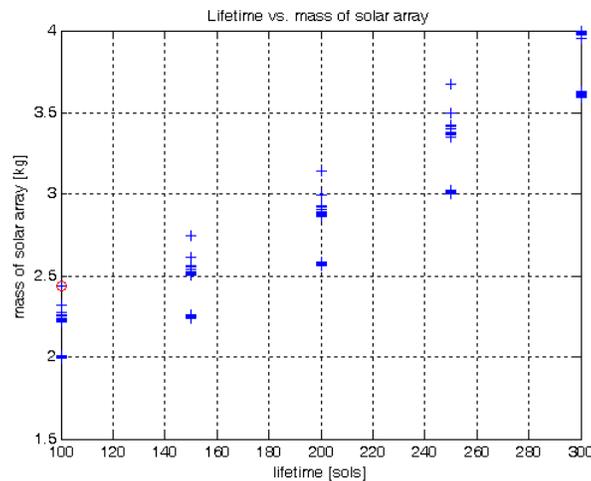


Figure 3.5.7: Mission lifetime versus solar array mass

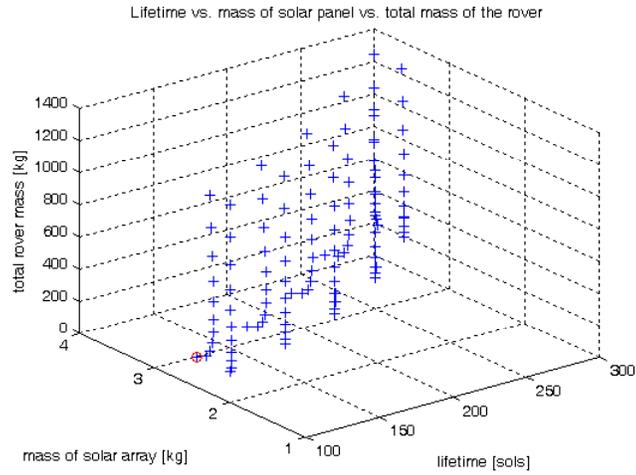


Figure 3.5.8: Mission lifetime versus power system cost versus solar array mass

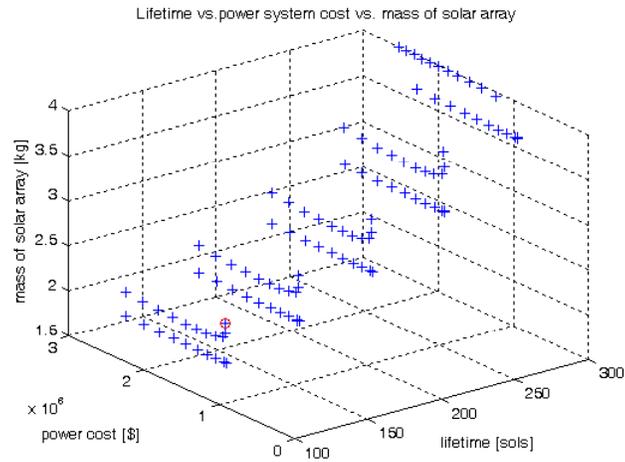


Figure 3.5.9: Mission lifetime versus total rover mass versus solar array mass

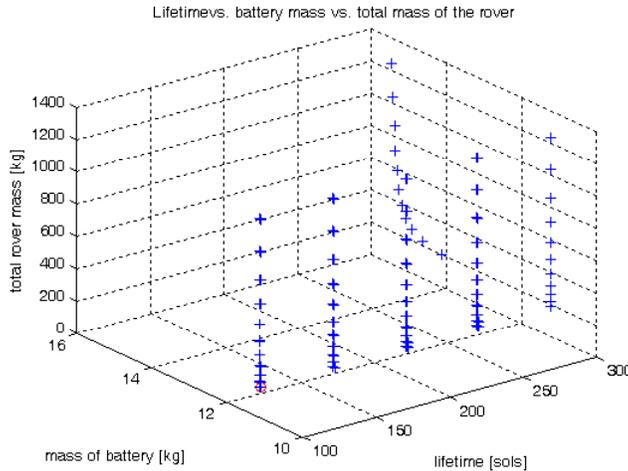


Figure 3.5.10: Mission lifetime versus total rover mass versus secondary battery mass

3.5.5.2 RTG Module

The RTG related data and equations used were confirmed by a number of reliable sources (books, experts, www). Most battery related assumptions were based on AAIA papers written about Mars missions. The only concern is the validation of the cost calculation for batteries. The SMAD battery cost model was used, but it must be noted that the model is designed for Earth orbiting satellites and is not always applicable for Mars rover missions. It is believed that the SMAD cost relation is conservative and prices the batteries higher than actual costs. Nevertheless, the model can be used as a rough estimate in comparing different options and selecting the most cost-effective one.

To demonstrate whether the Power module executes properly, all possible solar and RTG architectures were tested for the case when the rover contained all possible instruments listed in the Science Vector. The results of both plots appear logical and can be explained. Figure 3.5.11 shows that the solar power option is limited to 300 sols and 11 samples. This option represents the best solar case. All designs that represent longer mission lifetimes and more samples beyond this point correspond to rovers using RTGs, though RTGs are of course possible for <300 sols as well. Figure 3.5.12 shows the same result for the number samples collected, but from the point of view of the mass of the selected power system. The mass of the solar option is much less than its nuclear counterpart, however the nuclear option allows for more samples to be collected. The figures also show that the solar power option mass can vary greatly, while the RTG options have a more discrete mass profile.

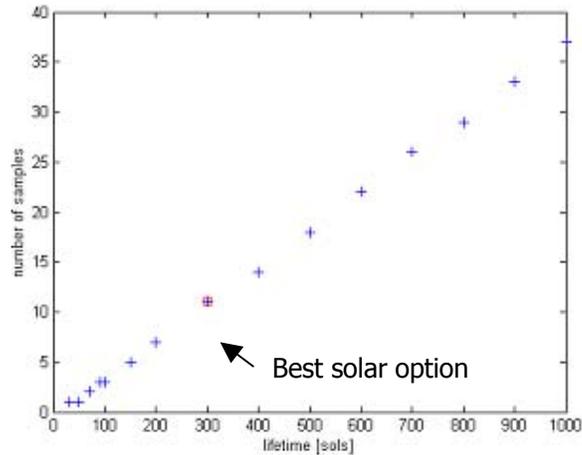


Figure 3.5.11: Number of samples versus mission lifetime

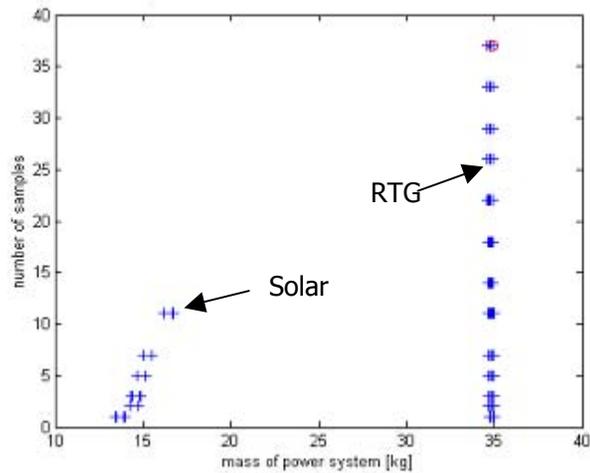


Figure 3.5.12: Number of samples versus power system mass

3.5.6 Expandability

3.5.6.1 Solar Module

The solar module can be expanded to incorporate new and different solar cell technologies (e.g. quantum dot solar cells). Dust removal mechanisms could be added to this module, resulting in lowering the total mass of the solar panel and increasing the solar panel life expectancy dramatically. For a short-term mission, the possibility of using fuel cells could also be considered. A fuel cell module could be added to the battery design code. Another improvement could be adding different topologies for the solar panels. In the current module, only flat solar panels have been considered.

3.5.6.2 RTG Module

One way to improve the RTG module is to input more detailed power profiles from the other subsystems, indicating exact times when power is needed. This would lead to a better estimation of the capacity of the batteries. More importantly, a different cost model should be used for calculating the battery cost. The current cost model should be used only for comparative purposes. The sizing of the power interfaces could also be made more exact. Modeling nuclear reactor power options is another recommendation for future expansion of the tool.

3.5.7 References

1. AIAA-2002-0718, Photovoltaic Power for Future, NASA Missions, Geoffrey A. Landis and Sheila G. Bailey, NASA John Glenn Research Center
2. AIAA Meeting Paper, A9715175
3. AIAA Paper 97-0086
A survey of next generation solar arrays (for spacecraft electric power), Douglas M. Allen W. J. Schafer Associates, Inc., North Olmsted, OH, AIAA, Aerospace Sciences Meeting & Exhibit, 35th, Reno, NV, Jan. 6-9, 1997
4. Dust Accumulation and Removal Technology (DART) Experiment on the Mars 2001 Surveyor Lander, Geoffrey A. Landis and Phillip P. Jenkins, Ohio Aerospace Institute, NASA John Glenn Research Center, Cosmo Baraona, David Wilt, Michael Krasowski, and Lawrence Greer, NASA John Glenn Research Center
5. Professor G. L. Kulcinski, University of Wisconsin-Madison
6. Joe Parrish, personal communication.
7. Quantum Dot Solar Cells, V. Aroutiounian, S. Petrosyan, A. Khachatryan, K. Touryan* Yerevan State University, 1Manoukian St., Yerevan 375049, Armenia
* National Renewable Energy Laboratory, Golden, Colorado 80401, USA
8. Shaltens, R. K. et al., "Stirling Radioisotope Power System As An Alternative For NASA's Deep Space Missions," published in *Innovative Approaches to Outer Planetary Exploration 2001-2020*, NASA Glenn Research Center
9. SpaceWorks Engineering, Inc. (SEI), "Advanced Concepts Database," available at <http://sei2.sei.aero/ACDB/RWDdetails.asp?ID=1>
10. Wahlquist, E., "Radioisotope Power Systems for Future NASA Space Exploration Missions," Office of Space and Defense Power Systems, U.S. Department of Energy, Washington, D.C.; http://centauri.larc.nasa.gov/newfrontiers/July23_02_Future_Space_Missions.pdf
12. Website reference:
http://calspace.ucsd.edu/marsnow/library/future/human_exploration/technology_development/power_generation_and_storage1.html
13. Website reference: <http://nuclear.gov/space/space-history.html>
14. Website reference: <http://powerweb.grc.nasa.gov/pvsee/sitemap.html>

-
-
15. Website reference: <http://pwg.gsfc.nasa.gov/stargaze/>
 16. Website reference: <http://spacepwr.jpl.nasa.gov/>
 17. Website reference: <http://spacepwr.jpl.nasa.gov/RadioPwr.htm>
 18. Website reference: http://spacesolarpower.nasa.gov/programactivities_b.1.html
 19. Website reference: <http://www.d230.org/vja/research/science/nuke/nuke.htm>
 20. Website reference: http://www.esqsec.unibe.ch/pub_89.htm
 21. Website reference: http://www.space.com/scienceastronomy/solarsystem/nuclearmars_000521.html
 22. Website reference: <http://www.islandone.org/APC/Electric/19.html#Nuclear>
 23. Website reference: <http://www.seds.org/spaceviews/cassini/rtg.html>
 24. Website reference: <http://www.iei.ie/papers/nuclear/nuclear4.html>
 25. Website reference: http://www.jpl.nasa.gov/news/fact_sheets/future.html
 26. Website reference: <http://www.gosolarpower.com/activesolarpower/>
 27. Website reference: <http://www.spacedaily.com/news/ssp-01b.html>
 28. Wertz, J. R. and Larson, W. L, "Space Mission Analysis And Design," Boston: Kluwer Academic Publishers, 1999

3.5.8 Appendix

US Radioisotope Thermoelectric Generators

Power Source	Power [We]	Mass [kg]	Dimensions [m]	Des.Life[days]	Actual Life	Cost [M\$]	Fuel	TRL	Comments
SNAP-1	500	159		60			Ce-144	6	Hg-Rankine
SNAP-1A	125	91	D=0.61,L=0.865	365			Ce-144	6	Hg-Rankine
SNAP-3B	2.4	2.3		90			Po-210	9	thermoelectric
SNAP-3B7	2.7	2.1		1826	15 years	19.37	Pu-238	9	thermoelectric; Transit 4A
SNAP-3B8	2.7	2.1		1826	9 years	19.37	Pu-238	9	thermoelectric; Transit 4B
SNAP-9A	25	12.3		1826		20.84	Pu-238	9	thermoelectric; Transit 5-BN-1
SNAP-11	25	13.6		90			Cm-242		thermoelectric; Surveyor lunar landing
Soft-landing generator	19	7.55		180			Cm-242	2	thermoelectric; lunar landing
Hard-landing generator	13	2.8	D=0.184,L=0.228	60			Cm-242	2	thermoelectric; lunar landing
SNAP-13	12.5	1.8		90			Cm-242	9	thermionic
SNAP-17A	30	20		1826			Sr-90		thermoelectric; comm. sat.
SNAP-17B	30	25		1826			Sr-90		thermoelectric; comm. sat.
SNAP-19	40.3	13.6	D=0.508,L=0.23	1095	15 years	21.85	Pu-238	9	thermoelectric; Pioneer 10
SNAP-19	40.3	13.6	D=0.508,L=0.23	1095	14 years	21.85	Pu-238	9	thermoelectric; Pioneer 11
SNAP-19	43	15.4	D=0.58,L=0.4	>365	> 6 years	22.02	Pu-238	9	thermoelectric; Viking 1 and 2; 2RTGs
SNAP-19A	20	8		>365	6 years	20.51	Pu-238	9	thermoelectric; NASA sat
SNAP-19A	250	-		>365			Sr-90	9	thermoelectric; various satellites
SNAP-19A	40	10		365		21.83	Pu-238	9	Surveyor lunar rover
SNAP-19A	1500	-		30-90			Po-210	9	thermoelectric; Extended Apollo missions
SNAP-19B3	28	13.6		>365	>2.5 years			9	thermoelectric; Nimbus III
SNAP-25	75	16		>365		24.13	Pu-238		program cancelled
SNAP-27	73.4	42		365	8 years	24.02	Pu-238	9	Apollo 12 to 17
SNAP-29	400	180		90		45.53	Pu-239	4	various satellites
SNAP-29	250	-		1826		35.65	Pu-238	3	US DoD and NASA
Transit-RTG	35.6	13.5		1826	> 15 years	21.54	Pu-238	9	Transit
High-power Thermionic	500	102		365-1826		52.11	Pu-238		thermionic
MHW-RTG	158	38.5		1826-3652	> 15 years	29.59	Pu-238	9	thermoelectric; Voyager 1 and 2; 3 MHW
DIPS	500	215		2557			Pu-238	4	NASA and DoD missions
GPHS-RTG	290	56	0.097x0.093x0.053 each	1826-3650	>15 years	38.29	Pu-238	9	thermoelectric; Galileo (2); Ulysses(1)

Power Source	Power [We]	Mass [kg]	Dimensions [m]	Des.Life[days]	Actual Life	Cost [M\$]	Fuel	TRL	Comments
GPHS-RTG	285	56	D = 0.41,L=1.12	3925.75		35.00	Pu-238	9	thermoelectric; Cassini; 3 RTGs
New MMRTG	140	32	D=0.41,L=0.6	3652	>14 years	25.00	Pu-238	7	uses 8 GPHS units;PbTe/Si/Ge
New SRG	110	27	D=0.27,L=0.89	3652	>14 years	20.00	Pu-238	4	2 GPHS units

DIPS = Dynamic Isotope Power System
GPHS = General Purpose Heat Source RTG
MMRTG = Multi-Mission RTG
SRG = Stirling Radioisotope Generator

References:

1. "Radioisotopic Power Generation" by William R. Corliss and Douglas G. Harvey; Englewood Cliffs, NJ: Prentice-Hall, Inc., 1964
2. Joe Parrish (Payload Systems, Inc)
3. NRC. 1996a. Assessment of the TOPAZ International Program, Committee on the TOPAZ
4. International Program, Aeronautics and Space Engineering Board. Washington, D.C.: National Academy Press
5. "Atomic Power In Space" by U.S. Department of Energy; Washington, DC, 1987.
6. Angelo and Buden, 1985; US DOE, 1987

3.6 Communications

3.6.1 Responsibilities

The responsibilities of the Communications module are to size the rover telecommunications system, including antennas and transponders; schedule communications activities based on communication window opportunities; and estimate power usage and subsystem cost, including DSN usage cost and equipment cost.

Based on this functionality, the two major components of the Communications module are:

- Communication schedule calculations including duration of communication, the total delay associated with communicating given data volumes, and average communication durations per day and night.
- Link budget calculations involving the sizing of the telecommunication subsystem for a Mars rover, power usage calculations and cost estimates.

The outputs of the Communications module are the mass and volume of antennas and transponders, power requirements, average communication duration per day and night, communication delays and cost estimates.

3.6.2 Assumptions

The following assumptions underlie the Communications module:

- The communications architecture is either direct to Earth (DTE) (34m HEF Deep Space Network (DSN) antenna use is assumed), via low altitude orbiter, via high altitude satellite operating in UHF, or via high altitude dedicated tele-satellite operating in both UHF and X bands, as well as hybrid communication architectures: DTE and low orbit, DTE and high orbit (UHF) or DTE and high orbit (X band UHF). For the hybrid architectures, DTE is assumed to be the primary communication method, while the other methods are used as backup. No hybrid scheduling is performed.
- Half of the communication windows for low orbit and high orbit are at night. If there is nighttime operation, the rover can both operate and communicate at night.
- For the purpose of sizing the antenna, the maximum, worst-case Earth-Mars distance is assumed. This means a maximum propagation delay as well. This allows a worst-case scenario design, since the exact launch date is not a parameter in the tool.
- It is assumed that the delay associated with human-in-the-loop is two hours.
- For communication via a relay, the Odyssey antenna parameters are used.
- Antenna tracking and acquisition times are assumed to be negligible and are ignored in the delay calculation.
- Availability of Deep Space Network (DSN) resources and relay satellite resources is assumed.
- Blackout periods due to celestial mechanics are neglected since the exact launch date is not a parameter in the tool.
- Blackout periods due to dust storms are neglected.
- An average landing latitude is used for the purpose of calculating latitude effect on communication windows.
- In designing the link budget, typical performance metrics and parameter values are assumed (such as Bit Error Rate).
- The current cost model neglects human operations cost.

- Uplink and downlink rates are equal.
- For communications via relay satellite, all data rates are assumed equal (i.e. surface to orbit and orbit to Earth), and delays introduced within satellites are ignored.
- Redundancy considerations are ignored.

3.6.3 Background

The communications architecture for any given rover design is specified in the Design Vector. The range of options for the communications architecture includes: DTE, via low altitude orbiter, via high altitude satellite operating in UHF, via high altitude dedicated satellite operating in X-band and UHF, and hybrid combinations of DTE with low orbit and high orbit relays.

The antenna diameter and transmitter power are determined using the link budget equations. Power usage is determined on the basis of average communication duration per day and night, provided typical command cycles according to the level of autonomy.

The algorithm for delay calculation is based on the type of communication architecture selected. The architecture specifies communication window distribution and data rate and nighttime operation capability. In addition to the uplink duration from Mars, if a response is expected from Earth the delay incorporates round-time propagation delay, downlink duration, human response time on Earth and the time taken by the rover to execute the received command. This assumes that the next command cycle will begin after the execution of the current command cycle. Note that the communication window duration varies with latitude of the rover.

The communications cost model includes equipment costing based on costing relationships in SMAD, and DSN usage cost estimates as outlined in 'NASA's Mission Operations and Communications Services' document.

3.6.4 Program Flow

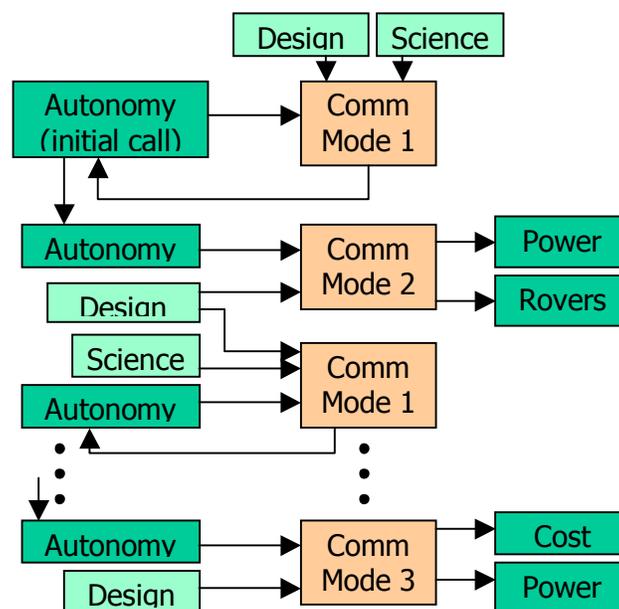


Figure 3.6.1: Communications subsystem program flow

Figure 3.6.1 shows the program flow and execution sequence of the Communications module. Inputs to the Communications module come from the Autonomy module, the Design Vector and the Science Vector. There are three different modes of accessing the Communications module:

- Mode 1: Autonomy accesses the Communications module. This mode involves communication duration and delay calculations.
- Mode 2: The master program accesses the module. This involves link budget calculations to estimate power usage, size and mass of the communication subsystem.
- Mode 3: The master program accesses the module. This involves link budget calculations to estimate the cost of the communications subsystem and power usage. The cost calculation cannot be performed in Mode 2 because of additional information regarding the total number of command cycles required from the Autonomy module.

The execution sequence of the Communications module is as follows:

Initially, the Autonomy module accesses the Communications module (Mode 1), providing a typical data volume to be communicated, and specifies whether there is nighttime operation. The Design Vector provides the communications architecture, while the Science Vector specifies the latitude range of the landing site that is used to estimate the window duration for the purpose of scheduling communication periods. The Communications module calculates the delay and communication duration for the given data volume and outputs this information to the Autonomy module. The purpose of this initial execution is to initialize the Communications and Autonomy modules in order to provide an initial estimate of communication power usage for the Power module.

Following the initialization, the master program executes the Communications module (Mode 2). In this execution, the Communications module uses the communication duration and delay information previously determined to calculate the communication duration per day and per night. It also performs link budget calculations to estimate the power requirement and determines the size and mass of the communication subsystem.

In the third phase, the Autonomy module accesses the Communications module multiple times (Mode 1), outputting a different set of data volumes to be communicated for each of the operations-intensive phases, such as traverse or sample acquisition. The Communications module then calculates the total delay associated with communicating the data volumes, as well as associated values including the total duration of data volume communication. Following this sequence of execution, Autonomy determines the total number of communication cycles required during the mission lifetime.

In the last phase, the master program executes the Communications module (Mode 3). Information regarding the total number of communication cycles is now available and can be used for estimating the cost and power usage based on the maximum communication duration and total delay. In this mode, link budget calculations are performed to estimate the equipment cost, DSN usage cost, power usage and the average duration of communication per day and night.

3.6.5 Validation

3.6.5.1 Link Budget

The link budget equations are used for calculating the antenna diameter and transmitter power. These equations have been validated using the FireSat benchmark. The Communications module produced the correct FireSat outputs based on FireSat input parameters for both command and telemetry, as shown in Table 3.6.1.

Table 3.6.1: Validation using FireSat benchmark

	Item	Symbol	Units	Command	Telemetry
FireSat Inputs	Data Rate	R	bps	100	8.50E+07
	Frequency	f	GHz	2	2.2
	Transmit Antenna Beamwidth	θ_t	deg	2	32
	Transmit Antenna Pointing Offset	e_t	deg	0.2	27
	Propogation Path Length	S	km	2831	2831
	Receive Antenna Diameter	D_r	m	0.07	5.3
	Receive Antenna Pointing Error	e_r	deg	70	0.2
	System Noise Temperature	T_s	K	614	135
	Signal to Noise	Eb/No	dB	45.5	15.9
	Transmitter Line Loss	L_l	dB	-1	-1
	Propagation & Polarization Loss	L_a	dB	-0.3	-0.3
FireSat Outputs	Transmit Antenna Diameter	D_t	m	5.3	0.3
	Equiv. Isotropic Radiated Power	EIRP	dBW	37.1	17.7
	Transmitter Power	P	W	1	20

Table 3.6.2 lists the link budget inputs and corresponding outputs of the Communications module for each of the basic communication architectures. Hybrid options are simply a combination of the columns shown.

Table 3.6.2: Link budget results

	Item	Symbol	Units	DTE	LMO	HMO (UHF)	HMO (X)
Inputs	Data Rate	R	bps	8000	2.56E+05	6.40E+04	6.40E+04
	Frequency	f	GHz	7.145	0.4597	0.4597	7.145
	Transmit Antenna Beamwidth	θ_t	deg	8.5	180	180	60
	Transmit Antenna Pointing Offset	e_t	deg	0.005	5	5	5
	Propogation Path Length	S	km	4.01E+08	1600	20000	20000
	Receive Antenna Diameter	D_r	m	34	1.3	1.3	1.3
	Receive Antenna Pointing Error	e_r	deg	0.005	0.95	0.95	0.95
	System Noise Temperature	T_s	K	30	200	200	200
	Signal to Noise	Eb/No	dB	2.7	2.7	2.7	2.7
	Transmitter Line Loss	L_l	dB	-0.3	-0.3	-0.3	-0.3
	Propagation & Polarization Loss	L_a	dB	-0.3	-0.3	-0.3	-0.3
Outputs	Transmit Antenna Diameter	D_t	m	0.346	0.254	0.254	0.049
	Transmitter Power	P	W	78.3	0.081	3.14	0.94

In Table 2, the input parameters are based on several sources:

- The data rates are as specified in Charles Whetsel's 'Surface System Design Process and Sizing Relationships' document.
- The frequency for DTE is in the X-band. The frequency for low Mars orbit (LMO) is in the UHF band. The frequency for high Mars orbit (HMO) is either UHF or a combination of UHF and X-band.
- For DTE, the use of the 34m HEF DSN antenna is assumed. The 70m antenna may be used also, but this fixes the receiver parameters. For LMO, Odyssey parameters are used (1.3m diameter antenna). HMO telecommunication satellites have not been launched yet, thus the Odyssey parameters are also assumed for HMO.

-
- The propagation path length for DTE is assumed to be maximum Earth-Mars distance (400×10^6 km), to allow worst-case scenario design. LMO orbiters typically fly at 400km altitude. However, for low orbit, the slant range can be double or more the overhead range, therefore a 1600km range is used. It is planned that HMO telecommunication satellites will be launched at typical altitudes of 10,000km. A slant range of 20,000km is used for this case.
 - E_b/N_0 is the ratio of received energy per bit to noise density. Bit Error Rate (BER) and modulation and coding schemes determine its value. Here, it is assumed that $BER=10^{-5}$, and the modulation scheme is BPSK and Plus RS Viterbi Decoding.
 - Typical values for line loss and polarization loss are assumed.

The system noise temperature is found to be a critical parameter in the transmitter power calculation. Whetsel's document suggests a power level of 50W. This level of power cannot be achieved unless the system noise temperature is as low as 20K. Receivers typically operate at around 30K. This indicates that the power must be around 78W, which is within 2dB of Whetsel's 50W power requirement. Another contribution to the larger power level at DTE (78W as opposed to Whetsel's 50W) is the assumption of maximum Earth-Mars range. The entire mission does not occur at this range, and the minimum range is a factor of five less, or approximately 80×10^6 km. The power required at minimum range is calculated to be ~ 2 W. Therefore, Whetsel's 50W seems to be an average power requirement over the mission lifetime.

For LMO, HMO (UHF) and HMO (X-band and UHF), the power requirements are calculated to be ~ 0.1 W, ~ 3 W, and ~ 1 W respectively. The significantly lower power levels are due to much shorter propagation path lengths (1600km for LMO, 20000km for HMO as opposed to 400×10^6 km for DTE). This is a discrepancy with Whetsel's document that suggests power usage of 50W for UHF communications (LMO, HMO). Since DTE communication capability exists on most Mars rover missions, one possible explanation may be that the 50W power requirement incorporates some DTE communications in addition to LMO/HMO.

The antenna diameter for DTE is calculated to be approximately 35 cm, for LMO and HMO(UHF) it is approximately 26 cm, and for HMO(X-band) it is approximately 5 cm. These are similar to the antennas used on MER.

3.6.5.2 Communication Delay

The implementation of the communication delay algorithm has been tested and validated by comparing against manual calculations of the delay.

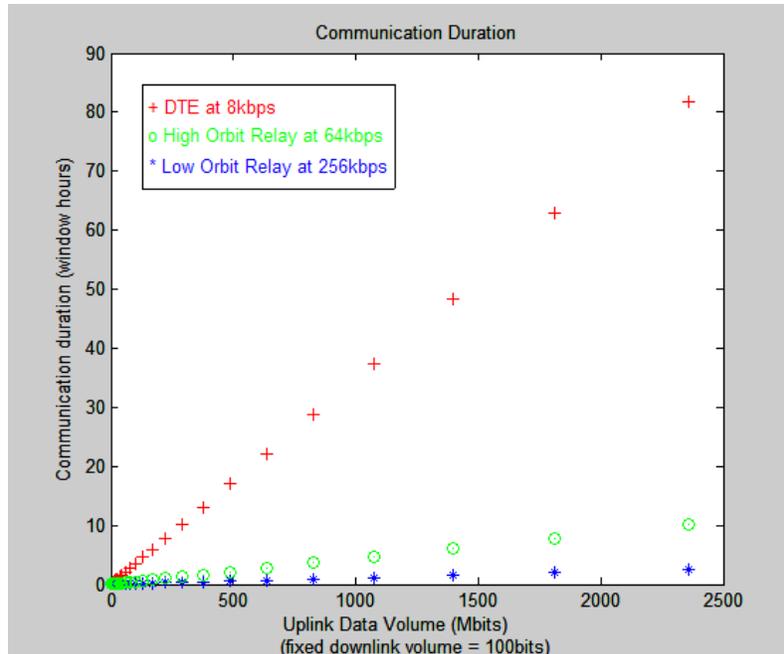


Figure 3.6.2: Communication duration

Figure 3.6.2 shows the total window duration required for uplink from the rover and downlink to the rover. Shown are plots of the communication duration for one command cycle consisting of a varying uplink volume and fixed downlink volume. This shows the relative link capability of the three types of communication architectures. As expected, DTE communication duration is the longest at the lowest data rate of 8 kbps, followed by communication via a high orbit relay satellite at 64kbps and the fastest link is the low orbit relay at 256kbps. Note that uplink and downlink data rates are equal for each type of communication architecture.

The communication delay is the total delay including communication window availability. It includes uplink duration from Mars, and if a response is expected from Earth, it also includes round-trip propagation delays, downlink duration, human response time on Earth and command execution time by the rover before the next command cycle starts. Figure 3 shows the total communication delay for one command cycle with varying uplink data volume from Mars, and a fixed downlink volume of 100 bits expected from Earth. DTE communication is available anytime during the day. However, it is assumed that the DSN is only available for four hours per day for this mission. Thus the window duration is limited to four hours. Low orbit relay satellites typically have overflights every twelve hours, with a duration of seven to twelve minutes per overflight, depending on the rover's latitude. High orbit relay satellites will be available more frequently than low orbit satellites, with overflights centered approximately six hours apart and a window duration of seventy-two minutes per overflight. The longest Earth-Mars round-time propagation delay of approximately forty minutes is used. It is assumed that operators on Earth will only be making "tactical" choices with a response time of two hours. These "tactical" choices refer to strategic re-planning of a whole day's activities that may require the overnight command cycle to be neglected in delay calculations. Another simplification that is assumed is that in case of communication via a relay satellite, there are no inherent delays associated with the relay satellite.

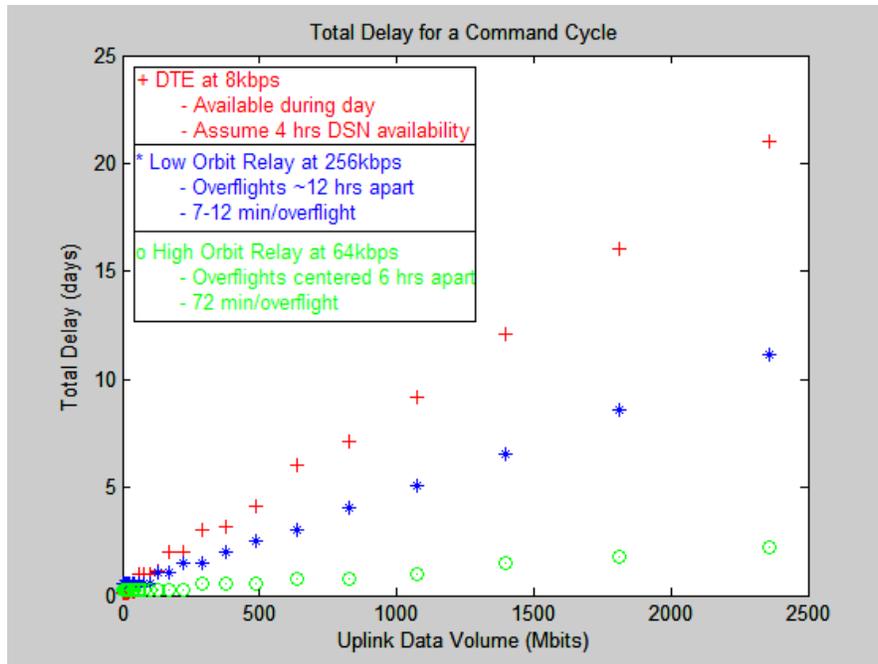


Figure 3.6.3: Total communication delay for a command cycle

Figure 3.6.3 indicates that the longest delays are associated with DTE. Although DTE has the longest communication window opportunity, the relatively low data rate results in longer delays for large data volumes. Low orbit relay, which has the highest data rate, is only available for short window durations, and although it represents an improvement over DTE, it does not provide the shortest delays. Better performance is achieved by communicating via a high altitude relay satellite. This is available more frequently than low orbit satellites and provides data rates much faster than DTE, thus achieving a relatively better performance than the other two architectures. This justifies future plans for high altitude telecommunication satellites orbiting Mars.

Note that Figure 3.6.2 and Figure 3.6.3 are generated for one command cycle. Typically, there are multiple command cycles associated with each of the operation-intensive activities. This means that the total delay is the cumulative delay of communicating and executing all command cycles. The level of autonomy has a direct impact on the number of command cycles and data volumes, which in turn affect the communication delays.

Figure 3.6.2 and Figure 3.6.3 also assume that nighttime operation capability exists. If the rover does not have nighttime operation capability, this will increase the communication delay associated with high altitude relay, although the high altitude relay will still provide relatively better performance than DTE and low orbit relay architectures.

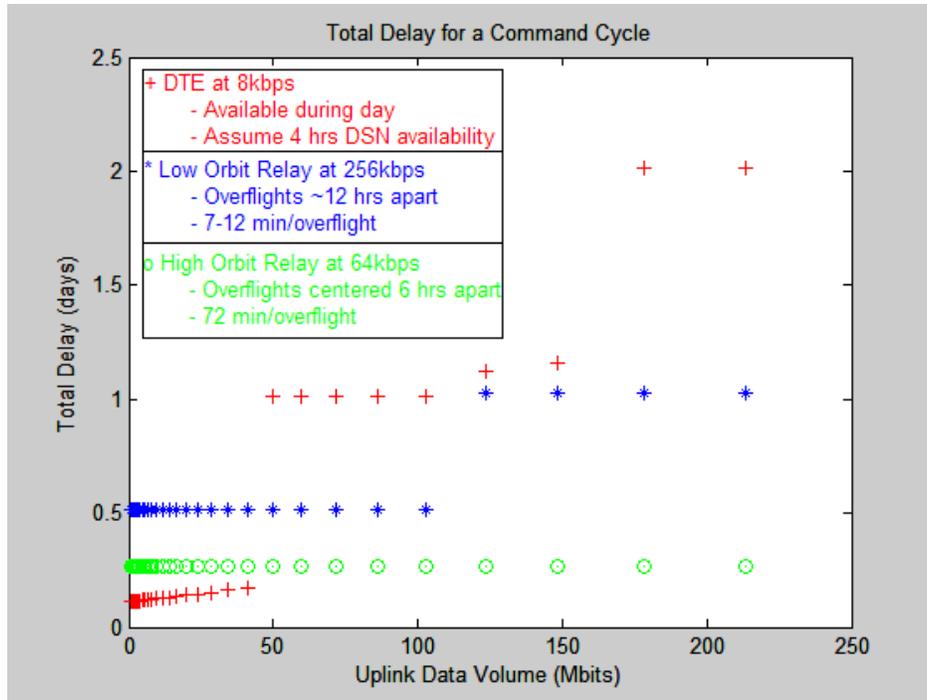


Figure 3.6.4: Total communication delay for a command cycle

Figure 3.6.4 is a finer scale of Figure 3.6.3 and shows variations of delay with smaller increases of data volume. There are discontinuous jumps in the delays, due to finite communication windows. For instance, for data volumes around 20 Mb, DTE has the shortest delay because the command cycle delay is less than one window duration. However for low orbit relay, the command cycle cannot be accomplished within the duration of one communication window, which is approximately seven minutes long. This means waiting for the next available communication window, which is half a day apart, thus the discontinuous jumps in communication delay by half a day. Similarly, high orbit relay has discontinuous jumps of approximately six hours, which explains why this communication architecture has longer delays than DTE for smaller data volumes. Notice that as the data volume increases, crossover occurs and DTE becomes less efficient, since the relatively low DTE data rate essentially starts giving rise to longer delays.

It is difficult to compare these results to MER or other Mars missions because of the assumptions underlying the modeling. Furthermore, data on command cycles and associated delays is generally unavailable for these missions.

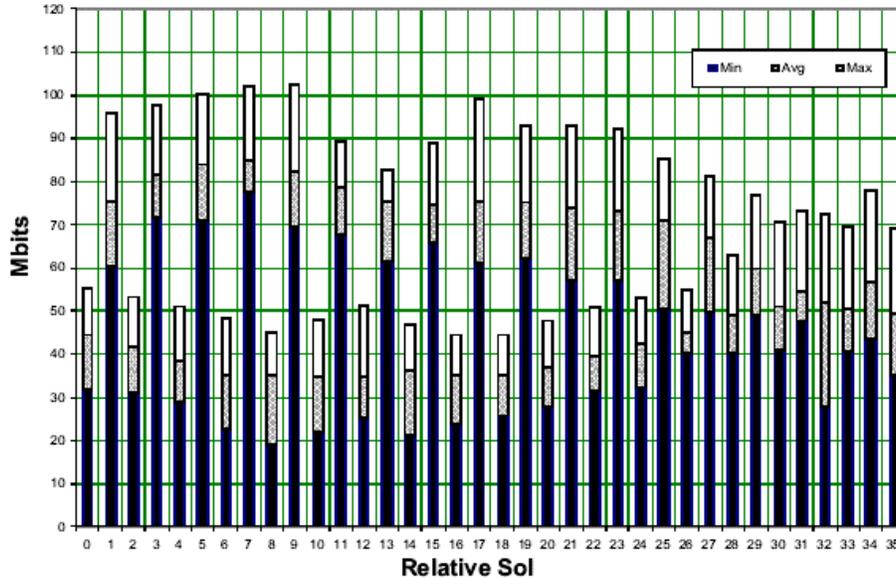


Figure 3.6.5: Odyssey data volume from MER per sol. The data rate is 128 kbps. [4]

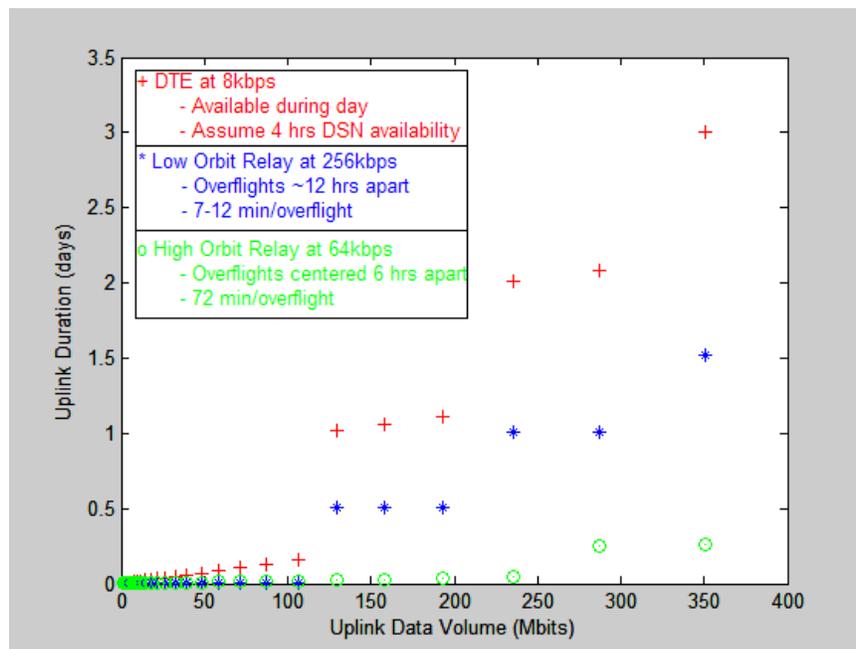


Figure 3.6.6: Uplink duration

The only comparison for validation was made with the Odyssey data volume to be received from MER per sol at a rate of 128kbps. Note that Odyssey is a low altitude science orbiter. Figure 3.6.6 shows uplink data duration as a function of data volume. For low orbit, the data rate assumed is double the data rate used to uplink MER data to Odyssey, based on Whetsel’s estimations. Figure 3.6.5 shows that a maximum of approximately 100Mb can be uplinked per sol at 128kbps. Figure 3.6.6 indicates approximately 300Mb per day at 256kbps, which is the equivalent of approximately 150Mbits at 128kbps. The discrepancy is attributed to the assumption that all relay satellite resources are available for the Mars rover mission. In reality, Odyssey has a limited memory availability (100Mb per sol for MER) that constrains the amount of data that can be uploaded per sol. Additionally, the uploaded data is buffered, and emptying the buffer relies on the availability of the DSN, which may further constrain the amount of

data that may be uploaded to a relay satellite. This demonstrates a limitation of the Communications module, which is the resource availability assumption.

Note that the Communications model incorporates the affect of rover latitude to determine the approximate window duration for each of the different communication architectures. It also provides a nighttime operation option. Restriction to daylight operation results in longer total delays.

3.6.6 Expandability

The major limitations of the Communications module are:

- Parameter value assumptions: A number of parameters are unknown, and reasonable value assumptions have been made for these parameters. Examples include the altitude of high orbit relay satellites and the diameter of the receiving antenna on high orbit relay satellites. For communication via a relay, the Odyssey antenna parameters are used. Other parameter assumptions include link budget parameters such as Bit Error Rate (BER).
- Uncertainties associated with communication window durations and overflights. These depend on the orbits of the communication satellites, as well as on the exact landing latitude of the rover, which are not accurately known.
- Availability of Deep Space Network (DSN) resources and the relay satellite resources is assumed.
- No hybrid scheduling is performed. For the hybrid architectures, it is assumed that DTE is the primary means of communication, while the relays are used as backup.
- Ignored redundancy considerations

Improvements involve finding better parameter values, incorporating more accurate models of the effects of celestial mechanics on the communication windows, modeling resource availability, modeling reliability and redundancy and devising schemes of dispatching data for hybrid communications.

The code can easily be expanded to incorporate these improvements, as well as to include future communication architectures such as optical communications.

3.6.7 References

1. DSMS Telecommunications Link Design Handbook, 810-005, Rev. E, 34-m HEF Subnet Telecommunications Interfaces
2. Frequency Assignment Guidelines for Communications in the Mars Region, Recommendation 22-1, Space Frequency Coordination Group
3. Mars Odyssey Fact Sheet
4. MER Mission Plan document
5. NASA's Mission Operations and Communications Services, AO 02-OSS-xx
6. Wertz, J.R. and Larson, W.J., 'Space Mission Analysis and Design', third edition, Space Technology Library
7. Whetsel, Charles. Notes on Surface System Design Process and Sizing Relationships for 16.89 class, Spring 2003.

3.7 Autonomy

3.7.1 Responsibilities

The Autonomy subsystem is responsible for modeling the effect of implementing different levels of autonomy on mission science return. This is ultimately achieved by determining how long it takes the rover to complete various tasks. The autonomy levels for long-distance traverse (site-to-site), short-distance traverse (sample-to-sample within a site), and sample acquisition are modeled. In addition, the Autonomy subsystem models the effects of whether or not the rover can process data during the night and the effects of allowing the computing power to vary by multiples of RAD 6000s. The Autonomy subsystem uses the combination of these capabilities to determine the performance of the rover in terms of the rate of samples analyzed, total number of samples obtained in the mission lifetime, and the time required to perform each of the rover's major tasks.

3.7.2 Assumptions

Since the Mars Exploration Rovers (MER) possess the most recent Mars rover technology, several parts of the code are modeled using MER characteristics as a baseline. For instance, the navigational sensors modeled in the program are identical to those on MER. If the user attempts to analyze the tradespace for a future Mars surface mission using the current code, it must be done with the realization that MER navigational sensors are the ones used in each design. However, the code is easy to update in the future when better navigational sensors are available by simply adding another sensor into the navigation.m file. The same is true for science and acquisition instruments. Similarly, the flight computer modeled in the code is a RAD 6000, and this can also be updated in the future.

Several modeling assumptions are used in the autonomy algorithm. These assumptions fall into two main categories, navigation and sample acquisition and processing. The navigation assumptions are:

- Only one type of environment will be encountered during the mission.
- When the rover drives, it drives at a maximum speed provided by the Rover module.
- Driving over a rock does not slow the rover down.
- The time required to turn the rover is negligible.
- For low levels of autonomy, the rover will not drive itself farther than the terrain seen by the Navcam images processed on the ground.
- All samples in a site are assumed to be the same average distance apart.
- The rover will only perform one reconnaissance per site.

The sample acquisition and processing assumptions are:

- Only one acquisition tool is used per location at a site.
- There is no parallel processing. Instruments process samples sequentially.
- Multiple instruments can process one sample.
- The user determines the samples to be acquired once per site during reconnaissance.
- The Instruments and Acquisition subsystems provide their own power requirements to the Power module.
- There is no remote analysis. All samples to be analyzed need to be acquired.

An additional assumption made was to deduct a 33% mission lifetime margin at the front end of the calculations. This will cause the amount of science that the program says can be accomplished in a ninety sol mission to be limited to the calculated science return for a sixty sol mission. This margin is incorporated for two reasons. First, it makes the science return estimates more conservative. Secondly, a 33% margin is used in some of the calculations performed by NASA for MER and MSL for such

performance estimates. The user can easily modify this margin by changing the appropriate constants number in the Autonomy module.

3.7.3 Background

A decision was made to limit the number of autonomy levels to the two outlined in various NASA SDT Reports, namely A1 and A3. A1 is defined as the autonomous capabilities of the MER rovers, and A3 is defined as having fully autonomous navigational capabilities.

The idea of adding an additional level, "A2", into the modeling as somewhere between A1 and A3, was not implemented for several reasons:

- A1 and A3 are already defined in NASA literature, whereas a definition of A2 is ambiguous.
- The trades between A1 and A3 would be sufficient since A2 would have performance levels somewhere between those of A1 and A3. Thus, the information that could be gleaned from calculating the effects of a new level of autonomy seems to be extraneous.
- A3 appears to be achievable in the near future, and thus if autonomy is to be increased to this level, an intermediate level for comparison is irrelevant

It was also decided that a model for estimating the cost of implementing a future level of autonomy would not be created. There is simply not enough information to accurately determine how much it will cost to develop, test, validate, and employ a future autonomous capability. Therefore, the goal of this project is to demonstrate the value of autonomy in terms of performance. A future program manager could weigh the potential cost against the added performance that this tool calculates.

3.7.4 Program Flow

The following is a summary of the tasks performed by the Autonomy module based upon the previously mentioned assumptions.

The Autonomy module is made up of two separate pieces of code. The first set of code (`autonomy_init.m`) determines the rover's theoretical average speed based on its actual driving speed and path planning time. The path planning time is determined by the processing power of the rover (the number of computers it has onboard) and the size of the rover. This section also works with the Communications module to determine the communications time for a driving sol. From those calculations, the power requirements for a traverse sol can be determined and subsequently passed to the Power module. The sizing of the power system is based upon a traverse sol, as opposed to a sample acquisition or processing sol, because a traverse sol will require the most power.

After the Communications, Rover, and Power modules are executed, the second set of autonomy code (`autonomy.m`) takes into account the power limitations on roving capabilities. Based on the levels of autonomy and power available, the site-to-site traversal and sample approach times are calculated. Next, the time it takes the rover to reconnoiter a site, process instrument data, and communicate data to Earth are calculated.

After the time required for all of the rover's tasks are calculated, a loop in the code is initiated that begins with the mission lifetime (less the aforementioned 33% margin and a rover deployment and egress time), and, based on the order that the rover must perform tasks, the time for each task is sequentially subtracted from the lifetime until the lifetime is depleted.

The on-site operations are modeled as follows:

1. Reconnaissance of a site is performed, information is sent back to the user on Earth, and then a reply is received specifying which samples to analyze. The total time required for these events is stored as the reconnaissance time. The rover performs only one reconnaissance per site.

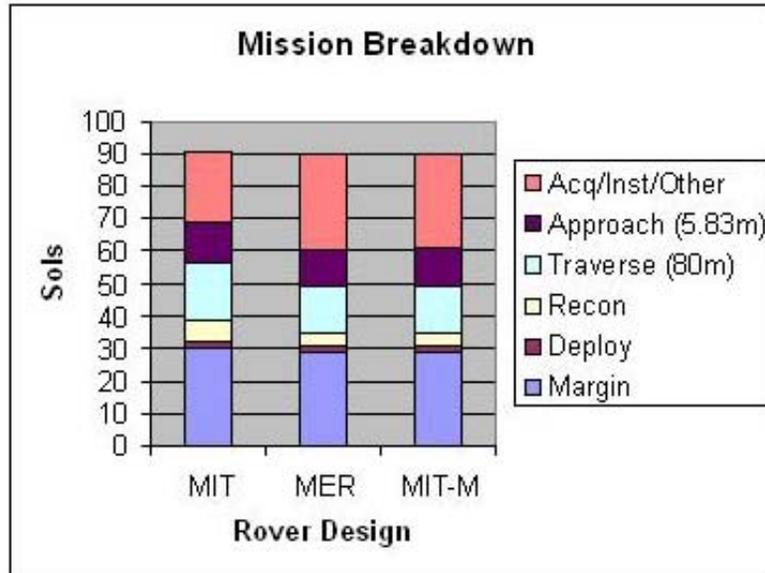


Figure 3.7.2: Autonomy module validation

Table 3.7.1: Module validation for A1

	MIT	MER	MIT-M
Total Samples	6	9	9
Traverse Distance (m)*	420	560	560
Approach Distance (m)*	23.3	35	34.98

* Does not include added obstacles avoidance distance

Each column in Figure 3.7.2 represents the breakdown in sols of the various tasks that the rover completes during its lifetime. All calculations take into account a 33% margin on mission lifetime up front. The first column is the initial MIT calculations, and the second column is the time taken from the MER mission plan. As can be seen from the graph, when MER is simulated using this module, the results are generally close values but not exact matches. In addition, Table 3.7.1 shows that the total number of samples obtained and distance traveled are also short of the MER scenario. These discrepancies can be explained.

The MER mission scenarios do not exactly match the general structure of the rover analysis tool, which models traversing to a site, obtaining a constant number of samples per site, and then moving to the next site. The plan for MER is simply too short with the low levels of autonomy onboard to be able to obtain several samples before moving on to the next site. Furthermore, the sites that MER goes to are not so far apart that a reconnaissance step must be completed upon arrival at each one, whereas the MIT code requires a reconnaissance step first at each site visited. Nevertheless, the values in the MER mission plan can be mimicked if a few mission structure parameters in the code are changed to match the MER mission structure. The following modifications were made to reduce the column marked MIT-M (MIT-Modified):

1. Add a point IR-Spectrometer to simulate the MINI-TES (which was not modeled). This increased the reconnaissance time from one to two sols.
2. Perform reconnaissance only twice over the mission lifetime. The original code performs a reconnaissance step first at each site. In order to provide a model with a reasonable fit to MER, the

Science Vector was set so that the rover would only obtain one sample per site. Since the program requires the rover to perform one reconnaissance per site, the code would have the rover perform that activity for every sample.

3. Combine two sites to provide the model of the 560m long-distance traverse.
4. Set samples 5.83m apart to provide the model of the 35m total sample approach distance.
5. Combine three pairs of samples, as in the selected MER mission plan (Figure 3.7.1).
6. Round down margin to better model the MER model.
7. Modify the Navcam range. The low-level autonomy only allows the rover to traverse a maximum of the Navcam range in a given sol (which is currently set at 30m). The actual limit depends on the available power, the terrain, and the time-out value. Instead of having the user define the time-out value and then modeling the probability of failure, it is simpler to replace that limitation with the range of the Navcam. The MER mission plan scenario allows for a daily traversal of 40m. Increasing the Navcam range to match the MER plan will better model this particular mission scenario.

Once these changes were made, the MIT-M results mimic MER almost exactly – including the number of samples obtained and total traverse distances, as seen in Table 3.7.1. The only remaining discrepancy is a difference of a single sol modeled into the sample approach time. This is explained by noting that the mission plan scenario did not bother with the accurate rover placement when performing soil analysis, so the one sol that would place it at a precise location was omitted.

These results show that the reason why the Autonomy module does not model MER exactly is primarily because the mission plan structure implemented does not match the mission plan structure of MER. Though the values are rather close, “exact” values are obtained when the structures are made to be similar. Therefore, the time calculations performed by the Autonomy module are able to roughly model MER’s mission scenario and will be able to model a MER-like rover using the implemented mission plan structure fairly accurately.

Table 3.7.2: Module validation for A3

# sols for:	MSL*	MIT
3000 m traversal	7.86	10
Sample approach (10-20)	< 0.5	0.020 – 0.03
Approach, acquisition, processing (rock analysis)	3	3

*2001 MSL SDT Report (using wheel diameter = 0.5m, Ls = 185, Latitude = -30, Rock Coverage = 20%)

The validation of a MER-like rover is only a validation for the modeled A1 level of autonomy. The validation of the A3 autonomy level code was done using the 2001 MSL SDT Report. Using a Design and Science Vector properties similar to that used to model MSL, the results obtained are close enough to those calculated by the SDT report to be considered valid (Table 3.2). The only significant discrepancy is in the number of sols required for a 3000m traverse. This discrepancy can be explained by the fact that the purpose of the SDT report was to be a quick big-picture overview rather than a high-fidelity analysis. Thus, the fact that the results from the module are within a reasonable range of those in the SDT report is satisfactory.

3.7.6 Expandability

The Autonomy module, like the rest of this software tool, is very expandable and upgradeable. The code itself is very well documented so that future users can update and even change specific parts of the algorithm easily when higher-fidelity models are developed. In addition, new code can be written to

accurately model effects that are currently assumptions. As previously mentioned, new navigation instruments must be added into the navigation.m portion of the code as they are developed. Also, if a new level of autonomy is to be added in the future, it can be done easily by simply following the template in the code for the other two levels of autonomy and modifying the Design Vector code (create_design_structure.m) only slightly in order to iterate over and analyze the new level of autonomy in the tradespace.

Some of the areas within the Autonomy module that should be update in the future are:

Avionics: The current size of the avionics package is similar to that for MER, since MER was used as the baseline. One of the direct effects of creating the code with a MER baseline is that many rover designs smaller than MER will be considered invalid since it cannot support the payload designed for MER's size and capability. One of the major examples of a limiting component is the size of the avionics package. Should the user decide to incorporate the capability to model smaller avionics packages, this can be accomplished by changing the avionics properties in the constants section of the autonomy_init.m file. However, a smaller avionics package might not be able to handle all the available sensors and instruments (e.g. a Sojourner avionics package cannot support the instruments on a MER). Therefore, to be able to correctly scale the avionics package with the rover, while maintaining a reasonable representation of its capability, the code will have to include checks on the number and types of instruments onboard and determine if the avionics package can either accommodate the demand or be scaled to meet the demand. In addition, since other aspects of the current module use MER as a baseline, such parts of the code would also need to be changed if a rover smaller than MER is to have a valid design.

Consumables: Currently, the Autonomy module discards any rover design that completely uses up either its science instruments or acquisition tools due to consumables. If the rover design has at least one non-consumable science instrument and acquisition tool in its payload, the design will never be invalid because it can always resort to the non-consumable item once its consumables are exhausted. Since most science packages come with instruments that do not require the use of consumables, it is generally the acquisition tools that can be completely used, rendering a design invalid. However, not all instruments require an acquisition tool to obtain a sample. Such an independent system is not currently modeled; the code presently assumes that all samples need to be acquired with an acquisition tool prior to being analyzed by a science instrument. To improve the Autonomy module in this respect, the code can be changed to only use instruments that do not require an acquisition tool in the case that all acquisition tools become exhausted. For such a scenario, the rover will approach the sample, skip the acquisition phase and continue with instrument processing. All that would need to be modified in the code is to only allow the acquisition calculations to be performed if the instrument requires acquisition. This will require an additional flag for each instrument stating whether or not the sample needs to be actively acquired before it can be processed. The program will then be less likely to discard a design as invalid due to the exhaustion of its consumables.

3.7.7 References

1. "Basic Facts About the Planet Mars." [NASA](http://humbabe.arc.nasa.gov/mgcm/faq/marsfacts.html). 15-May-2003. <http://humbabe.arc.nasa.gov/mgcm/faq/marsfacts.html>
2. Epperly, Michael. "Common Instrument Data Processor Preliminary Design Review." [SRI](http://image.gsfc.nasa.gov/presentation/cidp_pdr/sld040.html). 15-Jan-1996. 15-May-2003. http://image.gsfc.nasa.gov/presentation/cidp_pdr/sld040.html
3. "Mars Exploration Rover Mission." [NASA](http://mars.jpl.nasa.gov/mer/mission/spacecraft_rover_wheels.html). 22-Oct-2002. 15-May-2003. http://mars.jpl.nasa.gov/mer/mission/spacecraft_rover_wheels.html

-
4. Mars Exploration Rover (MER) Project: Mission Plan. JPL D-19659. MER420-1-300. Jet Propulsion Laboratory. 24-Apr-2002.
 5. Mars 2007 Smart Lander: Reference Science Scenario for SDT Report. 17-Sep-2001. PowerPoint presentation.
 6. NASA Mars Exploration Program: Mars 2007 Smart Lander Mission. Science Definition Team Report. 11-Oct-2001.
 7. Charles Whetsel, personal communication

4.0 System Integration and Validation

4.1 System Validation

Prior to integrating the entire program, each module was validated at the subsystem level. Once the modules were successfully integrated into the program, the next step was to validate the code at the system level. Validation at the system level was accomplished in two parts: comparison to known missions (e.g. MER) and investigation of single-axis trades. Comparing the results of the code to published information about current and past missions is a good means of determining whether or not the program is outputting correct answers. Secondly, a look at various single-axis trades can help determine whether or not each Design Vector entry is modeled correctly.

4.1.1 Past Mission Representation

Since the purpose of this program is to model Mars rovers, the ability to closely reproduce known current and past rover designs is a good metric by which to determine the program's validity. The tool was validated by comparing its results of a MER mission design to the actual MER rover designs, since many aspects of the code have been modeled using the current state of Mars rover technology (namely, the characteristics of the technology on the MER rovers). Many of the input values were found in the MER Mission Plan, a document supplied to this project by JPL. Table 4.1.1, Table 4.1.2, and Table 4.1.3 show the values used for benchmarking against MER. Table 4.1.1 shows the Design Vector values, which were input to match MER parameters. Note: information on night instrument processing was not found for MER so it was assumed that night instrument processing was not used.

Table 4.1.1: Design Vector values based on MER information

Design Vector Entry (variable name)	MER	MIT
lifetime [sols]	90	90
power_system [-]	'sol'	'sol'
telecom [-]	'dte'	'dte'
wheeldiameter [m]	0.25	0.25
autonomy_l_d [-]	'A1'	'A1'
autonomy_s_d [-]	'A1'	'A1'
autonomy_night_navigation [-]	'n'	'n'
instrument_night_proc [-]	'n'	'n'
autonomy_acq [-]	'n'	'n'
lander [-]	'n'	'n'
number_computers [-]	1	1

Table 4.1.2 shows the Science Vector input values used to model MER. These inputs consist of payload characteristics and site information. The payload entries in the MIT column correspond to the variable labels found in the code itself. The Mini-TES was not modeled in the program due to lack of information, however, it was replaced for validation purposes with inst(11), which is a point IR spectrometer.

Table 4.1.2: Science Vector inputs based on MER information

Science Vector Entry (variable name)	MER	MIT
Science Instruments		
Pancam	1	inst
Mini-TES	1	inst
Microscopic Imager	1	inst
Alpha-Particle-X-ray Spectrometer (APXS)	1	inst
Mössbauer Spectrometer	1	inst
Acquisition Tools		
Rock Abrasion Tool (RAT)	1	acq
Magnetic targets	3	acq
Navigation Sensors		
Navcams	1	nav
Hazcams - front	1	nav
Hazcams - rear	1	nav
Site Information		
samp_site [-]	1	1
sep_site [m]	~70.7	70.7
site_diam [m]	~4.3	4.3
terrain [% rock coverage]	< 0.06	0.05
ls_landing [-]	330	330
obstacle_factor [-]	1	1

A problem was encountered when running the program with the MER payload. This was due to the acquisition tools listed being limited by consumables. As stated previously in Chapter 3.7, if all acquisition tools exhaust their consumables, the program, in its present state, will discard that design as invalid. Thus, in order to mimic the performance of MER, it was necessary to include an acquisition tool that could work indefinitely, thus allowing the design to be valid throughout the simulation. For this purpose, the regular scoop was added. It did not affect the number of samples obtained and processed, but simply served as a means to continue the MER benchmarking simulation. Potential future improvements for this aspect of the program are described in Chapter 3.7, in the future work of the Autonomy subsystem.

Table 4.1.3 shows the output values from the program and the corresponding values published for MER. The data shows that the traversal distance per sol, top speed, and actual speed match MER values. Roving time and "thinking time" (time to calculate next drive step) are also very close to those of MER, though these values are difficult to calculate because they vary significantly and are represented here as averages. The important factor in this case then becomes the ratio between roving time and thinking time. Rover mass and rover power are correlated, and are both lower than MER values. This can be explained by the fact that the code designs a very lean rover, and some mass margin should be included at the top level in order to account for items such as cabling, connectors, etc. When proper margin values are added, the estimation of total rover mass will have higher fidelity. The rover size is similar to MER,

but not identical. This can be attributed to the fact that the tool outputs the same general rover shape for every size rover, and a specific NASA-designed rover will have dimensional relationships that are optimized more than those outputted by this program. The rover footprint, denoted by the length and width, (first two numbers in the rover size variable) is close to that of MER, but slightly larger. Finally, the height of the MIT-designed rover is slightly less than that of MER, which can be attributed to the way the program models the rover's mast.

Table 4.1.3: Code output parameters used for validation against MER

Program Outputs by Subsystem	MER	MIT
Autonomy		
Traverse Distance [m/sol]	25-40	30
Drive time per drive cycle [s]	~10	12.3
Time to calculate next drive cycle [s]	~20	28.9
Average Speed [cm/sec]	~1.5	1.4
Power		
Peak Power [W]	140	123
Rover		
Mass [kg]	185	124
Size - [L,W,H] [m]	[1.04, 0.84, 1.6]	[1.18, 1.08, 1.13]
Max speed [cm/sec]	~5	4.8
Total samples Obtained	5-9	6

Table 4.1.3 shows the significant numbers for benchmarking by comparison to a known model. The ability to reproduce known rover information is perhaps the most significant method of program validation. However, a second form of validation is given by examining some single axis trends in the design trade space.

4.1.2 Single Axis Trades

Single axis trades are examined by holding all Design Vector inputs constant except for one, which is varied to see its effect on the trade space. Since the program has already produced results similar to MER, the MER Design Vector from Table 4.1.1 is used as the reference design. The process of keeping the MER Design Vector constant except for one variable serves as a check to make sure that the effects of varying each Design Vector element are expected. It also shows, for example, what MER might look like if just one parameter was allowed to change (e.g. MER with A3 versus A1 level autonomy).

A number of single axis trades are shown here to further demonstrate the validity of the design tool. These trades are examined strictly in an intuitive sense. In other words, the following plots are only examined to ensure that the trends they show are the ones that are expected. This is not a stand-alone method to declare that the program is valid, however, since a reproduction of MER results has already been demonstrated, these plots are offered to add to the level of validation already demonstrated.

Figure 4.1.1, Figure 4.1.2, and Figure 4.1.3 are plots of lifetime against number of samples. As expected each plot shows that as the lifetime increases, the number of samples also increases linearly.

Varying the level of navigation autonomy in the Design Vector created Figure 4.1.1. The four curves in this plot represent the four possible combinations of long and short distance autonomy levels. The first designation is the level of autonomy for long distance traversal and the second is the level for short distance traversal. As described in Chapter 3.7, an A1 level of autonomy is equivalent to the MER capability, while an A3 level of autonomy is full navigational autonomy. As expected, the higher the level of autonomy, the greater the number of samples that can be obtained. It is interesting to note that the 'A1, A3' combination yields very similar performance to the 'A3, A1' combination in Figure 4.1.1. This is because the benefit of navigational autonomy increases with the amount of distance that must be covered. Since MER's mission scenario (as defined in the Science Vector) does not have very long distances for long and short term traversals, the results of these two combinations will be close.

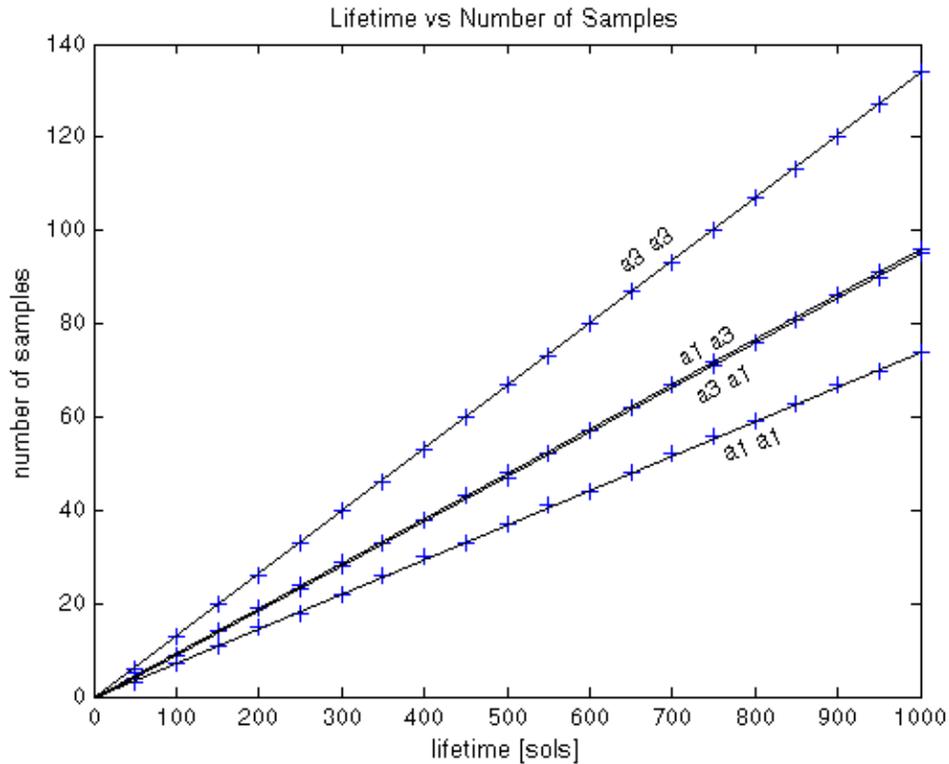


Figure 4.1.1 Lifetime vs. number of samples using navigational autonomy levels as the varied parameter.

Figure 4.1.2 is a plot that is also based upon varying autonomy capabilities. In this case, navigational autonomy levels have been held fixed at MER levels (A1, A1) while the rover is allowed to either have or not have the capability to acquire samples autonomously. Once again, the trend is as expected: more autonomy yields more samples.

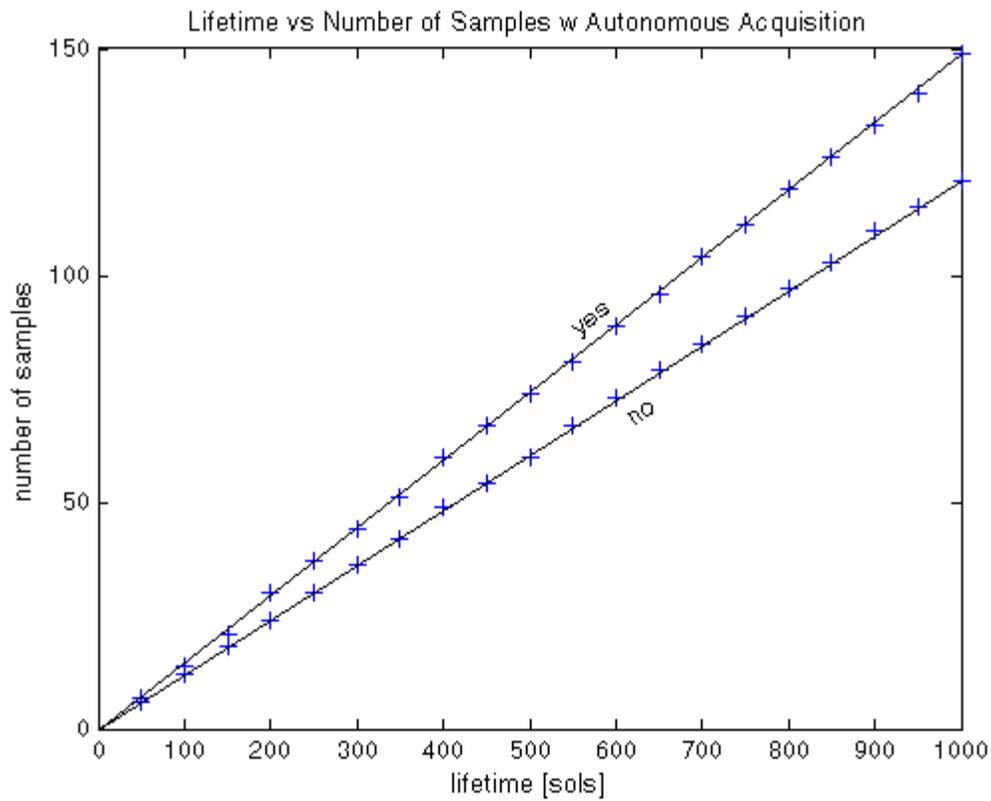


Figure 4.1.2 Lifetime vs. number of samples using acquisition autonomy as the varied parameter.

Figure 4.1.3 shows the effect that the communication system has on the number of samples that can be acquired. This plot shows the trends for direct-to-earth, low orbit relay, and high orbit relay communication options. In this case, using low orbit relay will cause fewer samples to be obtained than using direct-to-earth or high orbit relay. However, each of the latter two communication systems will allow the same number of samples to be analyzed. This is caused by a few factors. First, for this trade space, the amount of data to be transmitted to earth at a given time is too much for low orbit relay to handle efficiently, while high orbit relay and direct-to-earth can communicate this amount of data in less time. Second, it should be noted that while the low orbit relay has the largest data transfer rate, it also has the shortest communication window duration, which accounts for the lower efficiency in this case. While the rover has to wait for another overpass by a low orbit relay satellite, the high orbit relay and direct-to-earth options provide enough window duration to efficiently accomplish the data communication needs of the rover.

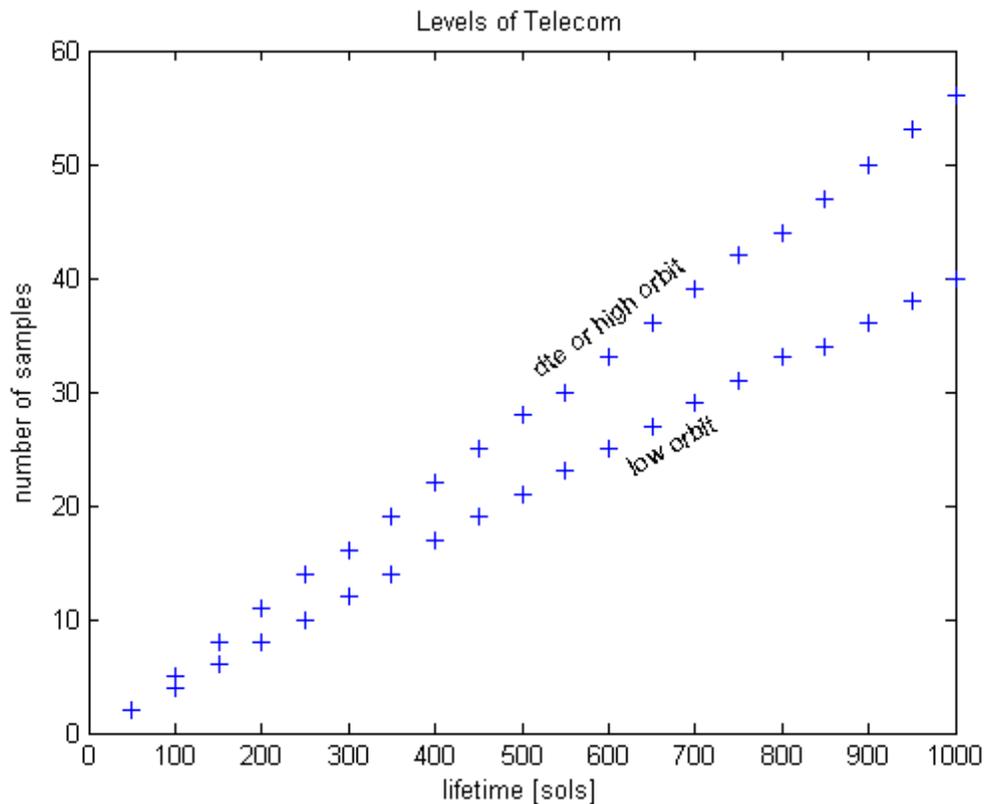


Figure 4.1.3 Lifetime vs. number of samples with communication system as the varied parameter.

Figure 4.1.4 shows the effect of increasing rover lifetime on the solar power system. As lifetime increases for a constant rover size, the solar array size must increase as well in order to combat solar degradation due to dust buildup. Once the lifetime gets close to 300 sols, the solar option becomes infeasible, and is discarded as invalid. Thus, there are no data points on the graph beyond a lifetime of about 300 sols. This is an expected result. In order for the solar option to be feasible for longer lifetimes, some kind of cleaning mechanism would have to be present to remove dust.

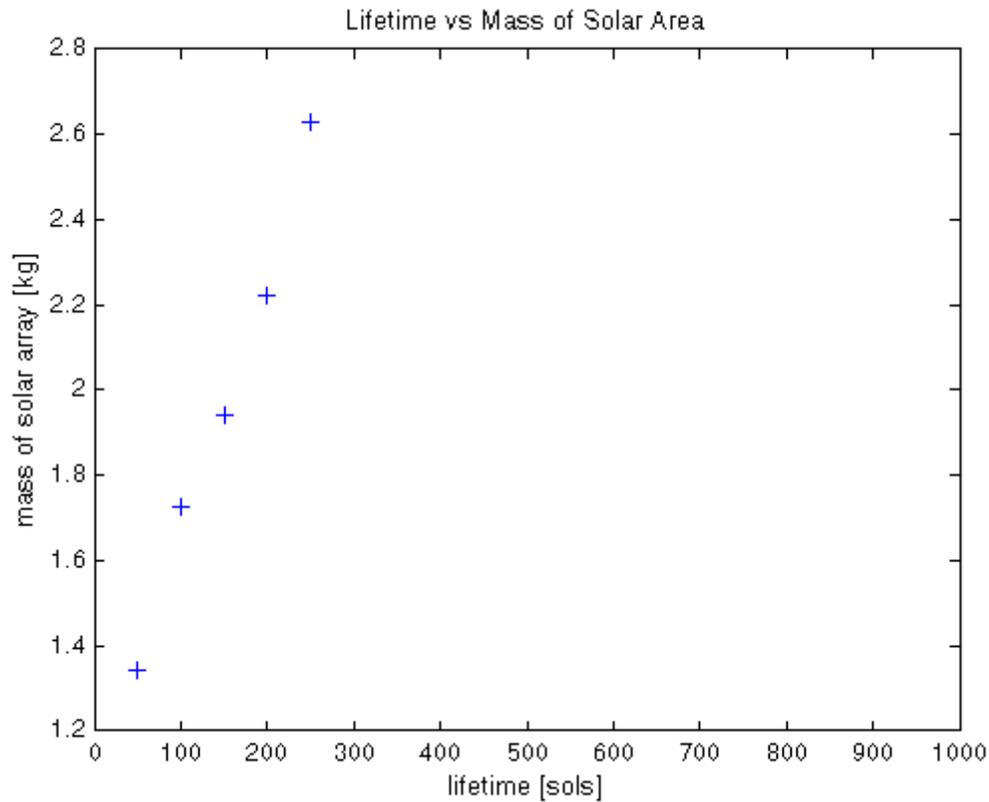


Figure 4.1.4 Lifetime vs. mass of solar array

The next several figures show the effect of allowing the size of the rover (scaled with the wheel diameter) to vary. Figure 4.1.5 shows how wheel diameter impacts the total mass of the rover. It is intuitive that larger wheels mean larger and heavier rovers, and this plot shows this result. The plot is not a linear relationship, which also makes sense. As a rover gets larger, it not only requires more structure for support and mobility, but it also requires a larger power system and is capable of carrying a larger payload. This non-linear trend is captured in Figure 4.1.5.

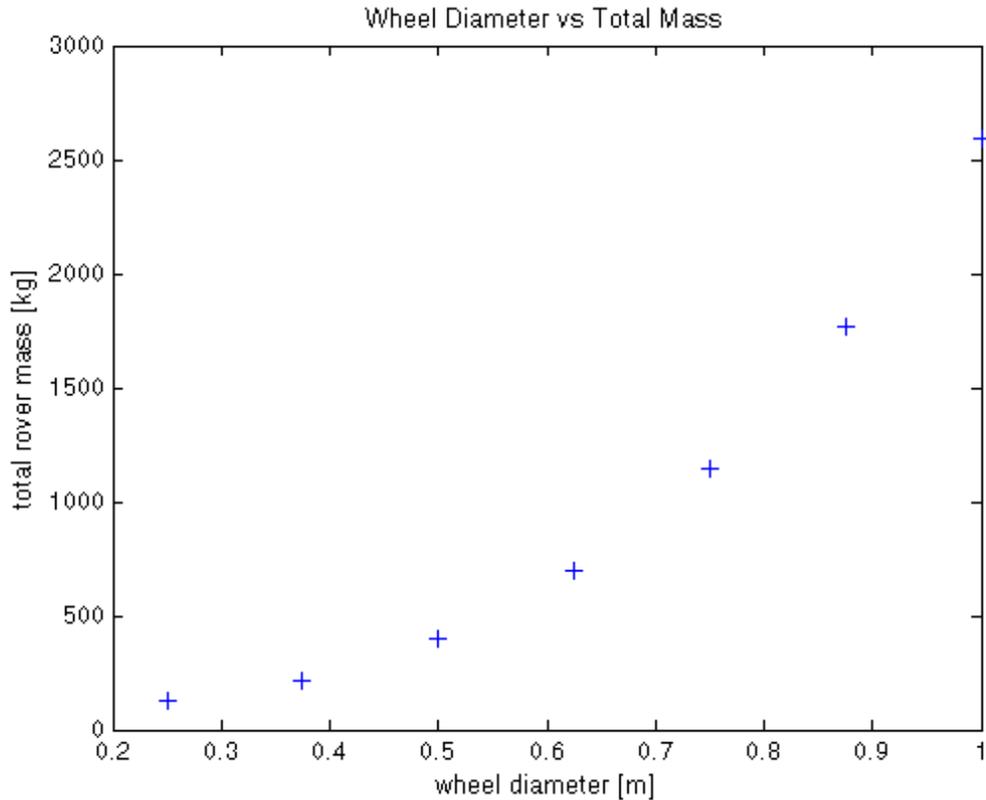


Figure 4.1.5 Wheel diameter vs. total rover mass

Figure 4.1.6 demonstrates how wheel diameter affects the number of obstacles encountered during a rover's traverse. An obstacle is defined as a rock that must be driven around instead of driven over. As expected, the figure clearly demonstrates that as the wheel diameter increases, the number of obstacles decreases. Note also that this relationship is not linear. The reason for this is that the rate of occurrence of rocks on Mars falls off exponentially as rock size increases.

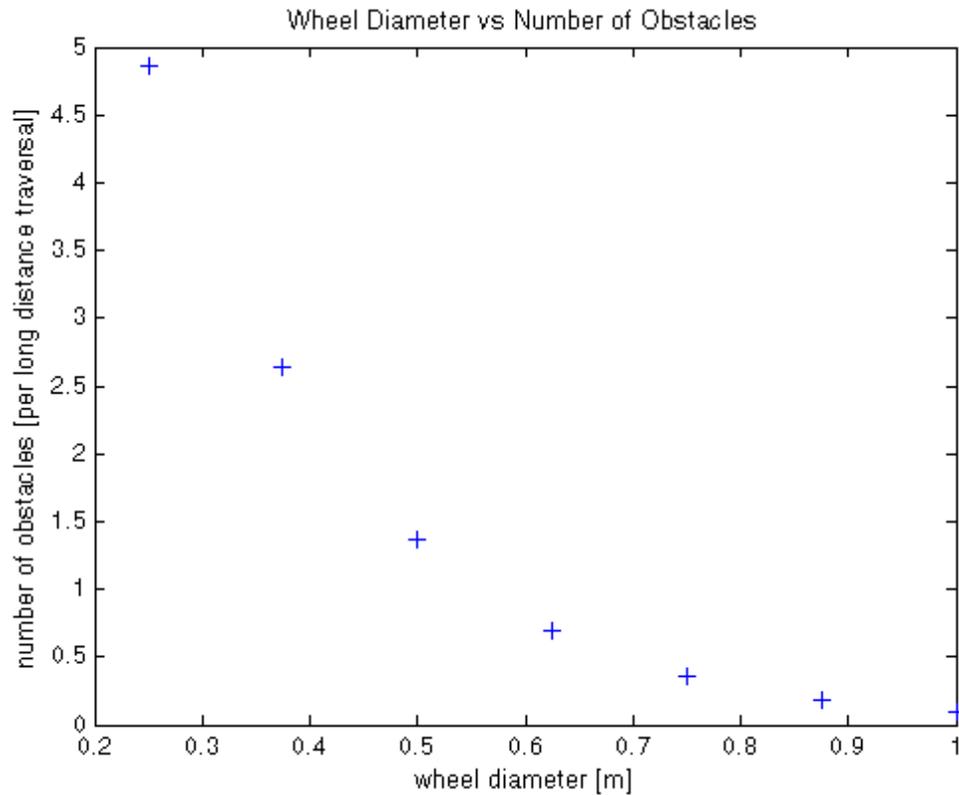


Figure 4.1.6 Wheel diameter vs. number of obstacles

Figure 4.1.7 shows an interesting trend between wheel diameter and the time the rover takes to plan its next drive step. The rover is assumed to travel half of its length per drive step. Since wheel diameter directly affects the length and width of the rover, a larger wheel diameter translates into a larger area to survey and process before moving. This relationship is not linear due to the second order effect that wheel diameter has on the area (length x width) surveyed. This plot is certainly an expected result.

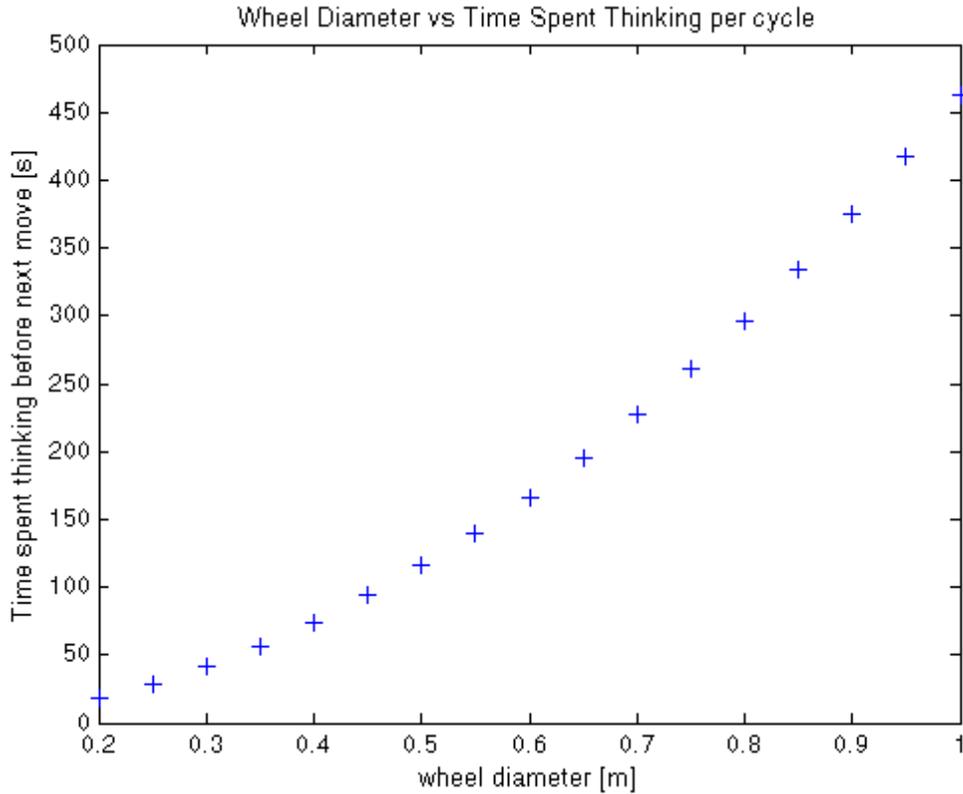


Figure 4.1.7 Wheel diameter vs. path planning time

Finally, there is an interesting single axis trade that is not intuitive at first glance, but with deeper inspection it can be understood. This is the relationship between wheel diameter and average rover traversal speed, shown in Figure 4.1.8. One would think that a larger rover would have a larger average speed of traverse. It is true that the actual driving speed of larger rovers is faster than that of smaller rovers. Larger rovers also have fewer obstacles to avoid, which boosts their average speed. However, as Figure 4.1.7 has shown, larger rovers must also think about their next drive step for longer than smaller rovers. It turns out that this factor actually dominates for larger rovers, while the number of obstacles to avoid is the dominating factor in smaller rovers. Thus, there is a maximum achievable average traversal speed, occurring around a wheel diameter of 0.25 meters.

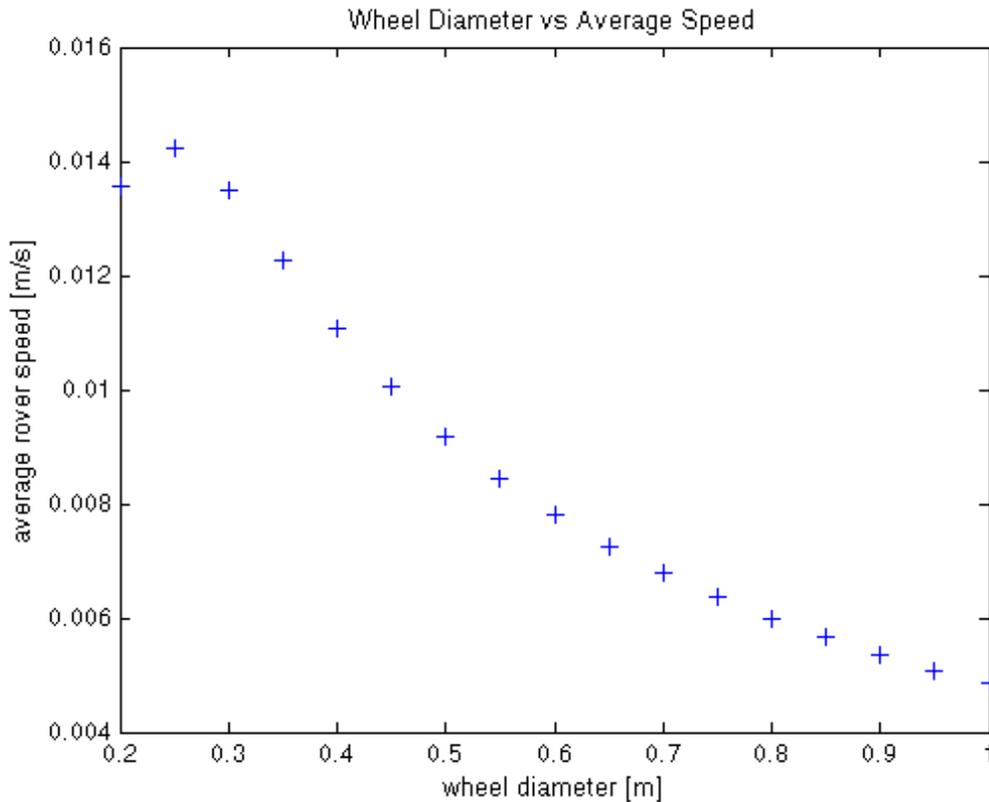


Figure 4.1.8 Wheel diameter vs. average rover traversal speed

4.2 Trade Space Results

4.2.1 MER Scenario Trade Space

The characteristics chosen for the trade space analysis are based on Mars Exploration Rovers (MER) characteristics in order to have some measure of how accurate the results of the tool are. It also allows the code to calculate what changes to the MER rover design might allow for improved performance, how much improvement could be expected, and what the costs of those changes are in terms of mass, development time, etc. In order to investigate the entire trade space of rover designs, three attributes were selected to assess the quality of a design: total number of samples over the lifetime, rover mass, and total cost (rover cost and operations cost). The first attribute is selected because it is the most analogous to science return and productivity. The total cost, although common to use as a measure, is not very accurate due to the limitations inherent in the cost modeling. Therefore, a third attribute of rover total mass is introduced because for interplanetary missions mass is very often a good surrogate to cost.

Designs are analyzed by looking either at the number of samples versus mass or at the number of samples versus total cost. The best designs would optimize both objectives: maximize samples while minimizing mass (or cost). Therefore a metric defined as number of samples over mass (or cost) is adopted in order to compare designs easily. The optimization is now a single objective where the best design is the one with the highest number of samples over mass ratio.

4.2.1.1 Number of Samples Over Mass Metric

Table 4.2.1: MER-like scenario Science and Design Vectors

<i>Science Vector</i>		<i>Design Vector</i>	
Landing Ls [deg]	330°	Lifetime [sols]	90 to 400
Latitude [deg]	0°	Communication	DTE, DTE-LMO
Rock Coverage	5%	Power Source	Solar, RTG
Number of Samples/Site	1	Wheel Diameter [m]	.25:.05:.4
Distance between Sites [m]	70.7	<i>Autonomy:</i> Long Distance Navigation	A1, A3
Site Size [m]	4.3	Short Distance Nav	A1, A3
Obstacle factor	1	Acquisition	Yes, No
MER Instruments	Micro. Imager, APXS, Mössbauer Spectrometer	Night Time Processing	Yes, No
MER Acquisition Payload	RAT, magnet, scoop	Computers	1, 2

In order to generate the trade space, a MER-like scenario was used as shown in Table 4.2.1. The Science Vector inputs are held constant while the inputs in the Design Vector are iterated. Examples of options included in the Design Vector that are not modeled on the real MER are an RTG power supply and autonomous nighttime processing. The plot in Figure 4.2.1 is the resulting trade space from these choices. Symbols refer to different levels of autonomy while colors refer to the power system. The blue symbols are solar powered designs and the red symbols are RTG powered. As seen from the distinct

pattern of coloration of the plot, solar powered designs are restricted to a maximum lifetime. Thus the number of samples that can be collected and analyzed is limited to a little over thirty. A more refined lifetime range (to define the trade space) results in a higher number of samples, but only about ten more.

There are three trends worth noting. First, as wheel diameter increases, the rover mass increases without more samples. Second, for solar missions, as the lifetime increases, the number of samples increases without much increase in mass. For RTG missions there is no dependency of mass to lifetime as expected. Third, for the two different types of communication selected, the result is an increase in mass for the DTE-LMO without any science benefits.

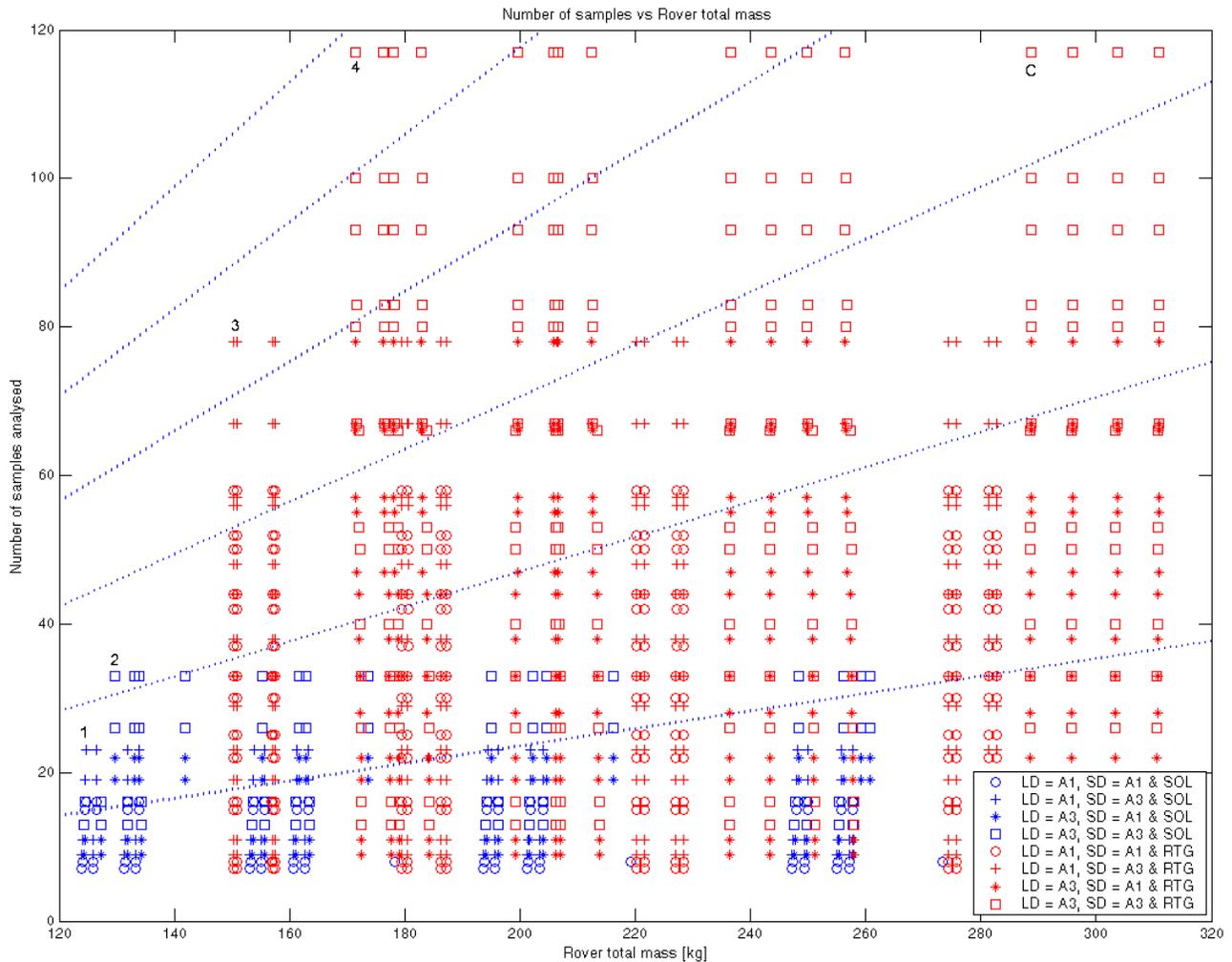


Figure 4.2.1: MER-like rover trade space; Number of samples versus rover mass

Highlighted in Figure 4.2.1 by numbers 1 – 4 in black, are the four non-dominated designs that constitute the Pareto front. The designs to the immediate left and below the design labeled 1 were not considered because the mass difference was negligible and the difference in the designs was not considered significant. In multi-objective optimization a design is dominated if it is possible to find another design that would increase an objective without decreasing at least one of the others. Before showing the specific Design Vectors for these designs, two apparent traits are seen from the plot itself. There is a pair of solar powered designs and one RTG powered design. Each pair has one design at A1 and the

other at A3 for long distance autonomy while short distance autonomy is at A3 for both. It was expected that the best designs would have the best possible autonomy (A3/A3) however, A1/A3 autonomy fairs better than A1/A3 autonomy with respect to mass.

Careful analysis of the trade space reveals that the extra mass for A3/A1 autonomy comes from a larger battery supply. A possible explanation is that a rover equipped with A1 long distance traversal autonomy stops more often and thus can charge its batteries more often, while with A3 autonomy it does not have enough time to charge and thus needs more batteries.

Table 4.2.2 shows the Design Vectors for the four Pareto front rover designs. All four designs have a DTE communication system and all are equipped with A3 short distance navigation autonomy as well as autonomous acquisition. Aside from designs 1 and 3 having A1/A3 long/short distance navigation autonomy, there are some other interesting results to note about the rover designs. Nighttime processing penalizes solar powered rovers because it entails a larger battery requirement. However, the best RTG designs have nighttime processing resulting in higher science return. Also, with the increase from A1/A3 to A3/A3 navigation autonomy in both the solar and RTG designs there is an increase in the number of on-board computers (from one to two).

	Design #1	Design #2	Design #3	Design #4
Mass [kg]	~124.5	~129.5	~150.5	~171.5
Samples	23	33	78	117
Lifetime [sols]	200	200	700	700
Wheel Diam. [m]	0.25	0.25	0.25	0.25
Power	Solar	Solar	RTG	RTG
Communication	DTE	DTE	DTE	DTE
Long Dist. Aut.	A1	A3	A1	A3
Short Dist. Aut.	A3	A3	A3	A3
Acquisition Aut.	Yes	Yes	Yes	Yes
Night Processing	No	No	Yes	Yes
Computers	1	2	1	2

Table 4.2.2: Design Vectors for rover designs on the Pareto front

In Figure 4.2.2 the highlighted designs lie along lines of isoperformance; rovers along these lines have the same performance defined as the ratio of number of samples over mass. The blue dotted lines overlaid on the plot represent the ratio of one sample per every 7kg of rover mass. As ratios increase (2:7, 3:7, etc), they are shown with a greater slope. Along the line 5:7, two RTG designs are highlighted whose lifetimes differ by 100 sols. The Design Vectors for these rovers are shown in Table 4.2.3. The main difference between these two designs is that the design with a shorter lifetime has two computers instead of just one, implying that the extra computer saves 100 sols.

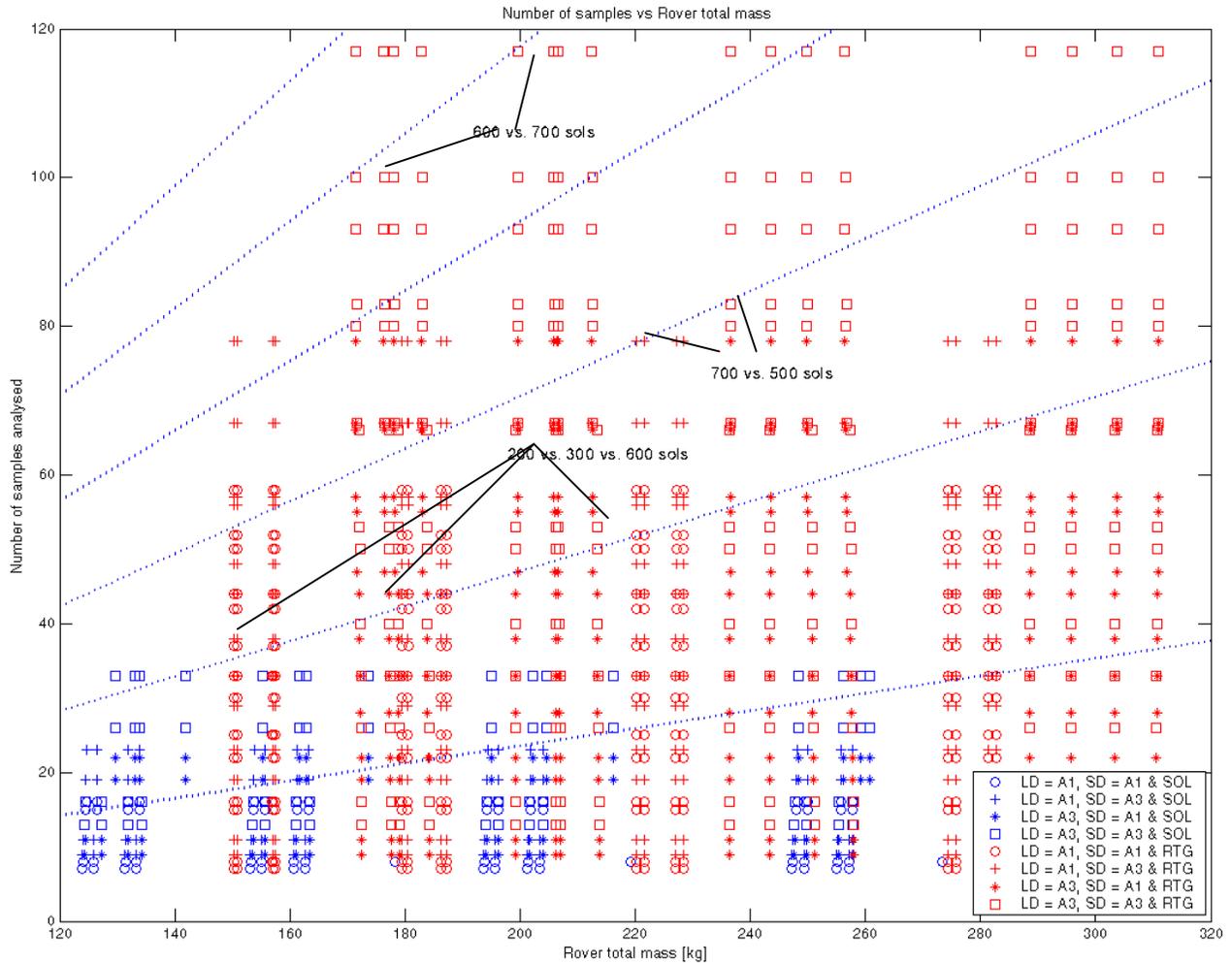


Figure 4.2.2: MER-like trade space; Isometric lines were used to compare designs with similar productivity ratios.

Table 4.2.3: Isoperformance line 5:7 Design Vectors

	Left Design	Right Design
Mass [kg]	175	200
Samples	~95	117
Lifetime [sols]	600	700
Wheel Dia. [m]	0.25	0.30
Power	RTG	RTG
Communication	DTE	DTE
Long Dist. Aut.	A3	A3
Short Dist. Aut.	A3	A3
Acquisition Aut.	Yes	Yes
Night Processing	Yes	Yes
Computers	2	1

Along the isometric line 3:7, two RTG designs have been highlighted; their Design Vectors are listed in Table 4.2.4. They are similar except that one is a 500 sols mission with long distance autonomy at A3

while the other is a 700 sols mission with long distance autonomy at A1. Both rovers have the same performance (samples over mass) but the smarter one needs 200 less sols, which does not impact mass but does impact operations costs. Reducing the mission lifetime by 200 sols results in roughly \$60 million saved on operations costs. This estimate can then be used as a cap on development costs required for A3.

Table 4.2.4: Isoperformance line 3:7 Design Vectors

	Left Design	Right Design
Mass [kg]	220	235
Samples	77	~82
Lifetime [sols]	700	500
Wheel Dia. [m]	0.35	0.35
Power	RTG	RTG
Communication	DTE	DTE
Long Dist. Aut.	A1	A3
Short Dist. Aut.	A3	A3
Acquisition Aut.	Yes	Yes
Night Processing	Yes	Yes
Computers	1	1

Finally, along the isometric line 2:7, designs with different power systems were compared and are listed in Table 4.2.5. Unfortunately, there are too many variables in the Design Vectors that are different, so the comparisons are not as compelling as some of the other observed differences. However, it is still interesting to note that across the same line of performance, using RTGs as a power source allows for longer lifetimes.

Table 4.2.5: Isoperformance line 2:7 Design Vectors

	Left Design	Middle Design	Right Design
Mass [kg]	145	175	245
Samples	33	~37	~57
Lifetime [sols]	200	300	600
Wheel Dia. [m]	0.25	0.25	0.35
Power	Solar	RTG	RTG
Communication	DTE	DTE	DTE
Long Dist. Aut.	A3	A3	A3
Short Dist. Aut.	A3	A3	A1
Acquisition Aut.	Yes	No	No
Night Processing	No	Yes	Yes
Computers	2	2	2

4.2.1.2 *Number of Samples Over Cost Metric*

Another way to plot the rover trade space is to create a 'number of samples' versus 'total cost' graph (Figure 4.2.3). Isometric lines are also shown in this plot. The following ratios are included – 40 samples: \$550 M, 80:550, and 120:550. Highlighted in this plot are the four designs that are listed in Table 4.2.2, which are those in Figure 4.2.1's Pareto front.

Notice that designs 2 and 4 are on this plot's Pareto front. Highlighted in Figure 4.2.3 letter C (also noted in Figure 4.2.1) is the optimal design; it is the rover with the highest number of samples over cost ratio as shown by the lines of isoperformance. Its Design Vector is listed in Table 4.2.6. This rover has the

longest lifetime, an RTG power source, and is comparatively large with 0.4 m wheel diameter; it also has the maximum autonomous ability but only one computer. The cheapest rover is also highlighted in this plot, labeled M. It has the MER Design Vector, except that it has autonomous acquisition ability.

Since there are limitations in the cost models used in this project, further cost comparisons are not reliable.

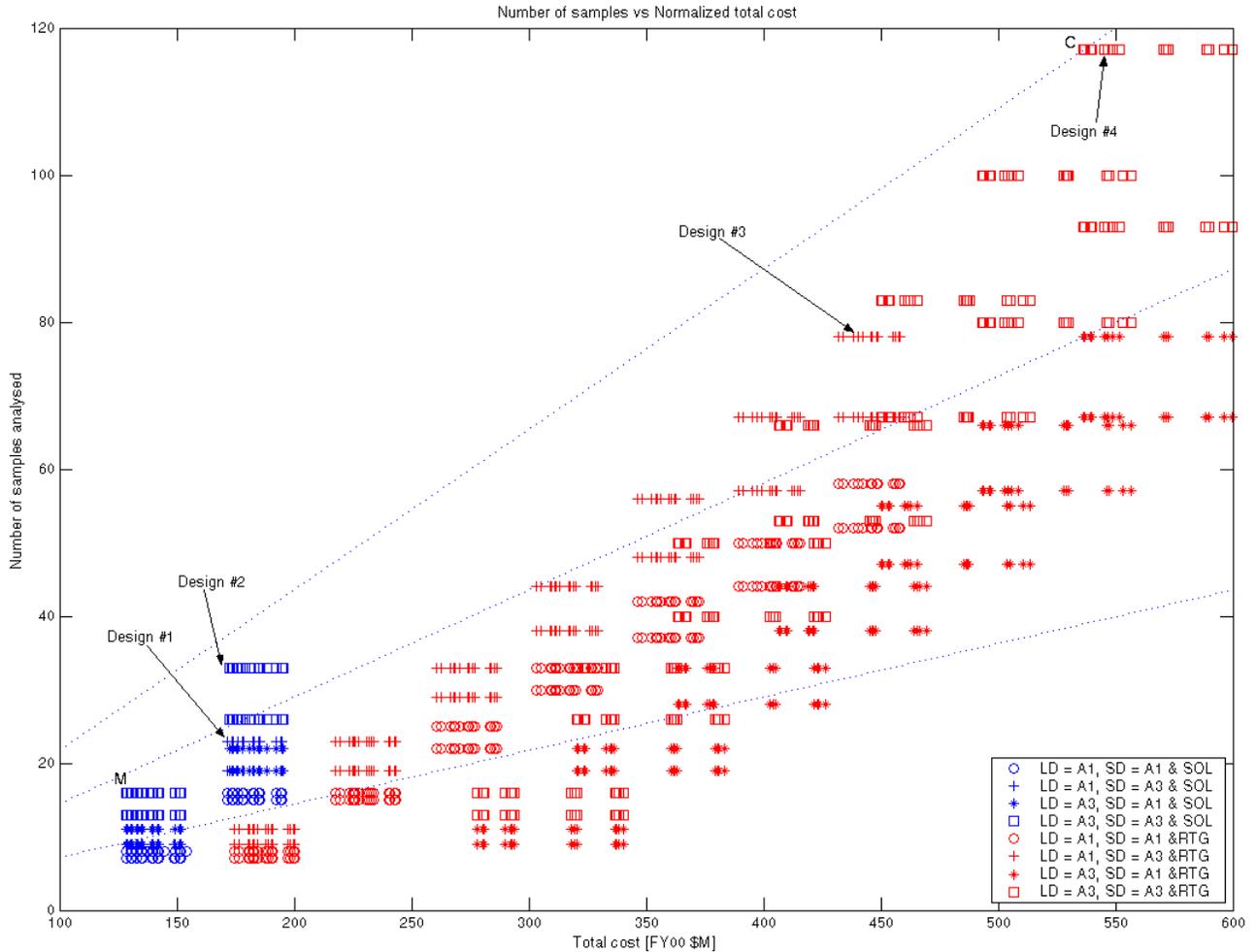


Figure 4.2.3: MER-like rover trade space; Number of samples vs. total cost

Table 4.2.6: Cost optimal Design Vector for rover labeled 'M'

	Cost-Optimal Design
Mass [kg]	~285
Samples	117
Lifetime [sols]	700
Wheel Dia. [m]	0.4
Power	RTG
Communication	DTE
Long Dist. Aut.	A3
Short Dist. Aut.	A3
Acquisition Aut.	Yes
Night Processing	Yes
Computers	1

4.2.1.3 *Sensitivity Analysis*

A trade space with the MER characteristics was calculated holding all parameters in the Design and Science Vectors constant with the exception of the landing date. In Figure 4.2.4, there are three different colors represented in the plot of number of samples versus rover total mass. The top four green shapes represent the LS_330 option, which corresponds to the beginning of fall. The bottom blue shapes represent the LS_150 option, corresponding to the beginning of spring. Both the green and blue indicators represent architectures with a solar power supply. The center red shapes are both the LS_150 and LS_330 but represent architectures with the RTG power supply.

This plot demonstrates the effects of moving the landing date from LS_150 to LS_330. Design 1 is the LS_330 landing date while design 2 has the LS_150 landing date. The designs with RTG power sources are not affected by changing the landing date, but designs with the solar arrays show increased mass for a given number of samples returned. There are more hours of sunlight each day during the summer than during the winter, so for a given daily energy requirement, larger solar arrays are required during the winter than during the summer. This corresponds with design 2 being heavier than design 1. Also, winter landings require larger batteries since colder temperatures lead to requirements for more heater energy, especially during the night hours when the rover is reliant on battery power. When the subsystem properties were examined using the main GUI, it was found that the battery mass of the two rovers is distinct. Design 1's battery mass is 8.9 kg while design 2's is 11.36 kg.

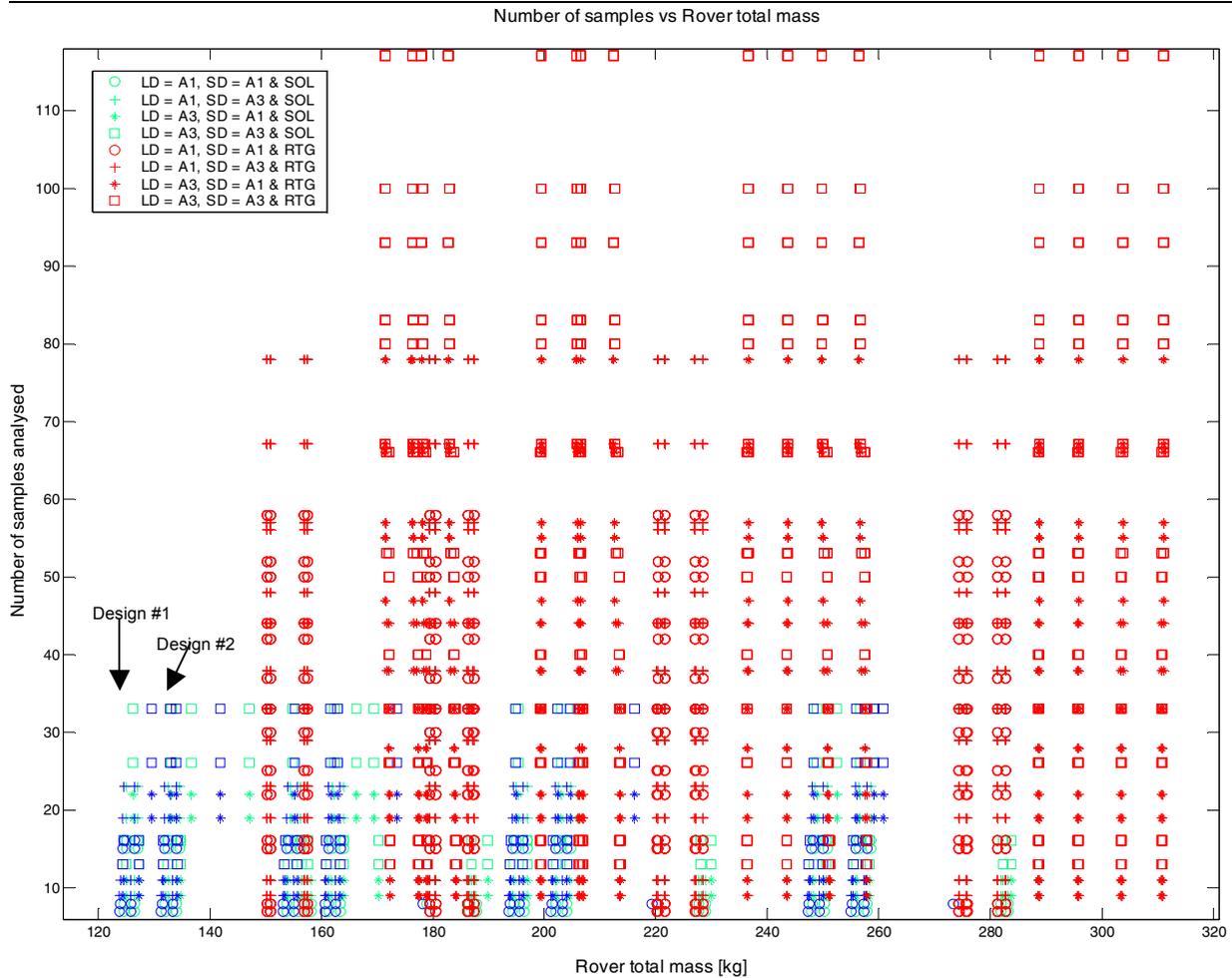


Figure 4.2.4: A MER scenario trade space with varying landing dates.

Colors represent landing date and power source. A second sensitivity analysis to the Design Vector was performed regarding the wheel diameter and percent rock coverage. The Science and Design Vectors were held constant with the exception of power sources, the iterations over the wheel diameters, and the rock coverage parameter. As shown in

Figure 4.2.5, the wheel diameters were iterated over from .2 to .45 m in steps of .05 m and three different trade space calculations are shown. The red represents 20% rock coverage (unlikely for Mars surface according to current data), blue is 10% rock coverage, and green is 5%. The Design Vector for this plot is shown in Table 4.2.7.

Table 4.2.7: Values in the Design Vector for

Figure 4.2.5

Design Vector	
Lifetime [sols]	100
Wheel Diameter. [m]	.2:.05:.45
Power	Solar, RTG
Communication	DTE
Long Distance Autonomy	A1, A3
Short Distance Autonomy	A1, A3
Acquisition Autonomy	Y
Night Navigation	N
Night Processing	Y
Computers	2

The plot compares the odometer distance between sites vs. the wheel diameter of the rover architectures. The odometer distance refers to the actual distance traveled while maneuvering around obstacles. It is first noted that as expected there is less distance that a rover in 5% rock coverage would have to travel versus a rover in 10% and 20% rock coverage.

Figure 4.2.5 also shows that there is some least optimal wheel diameter for a given rock coverage. The 20% rock coverage trade space appears to be more easily maneuvered with a smaller wheel diameter while the result of the 5% rock coverage trade space shows the opposite effect. Note that the actual change in distance from a maximum to a minimum point is on the order of a single meter, which will likely not affect rover productivity unless it is expected to travel unusually long distances. A factor that would more likely drive the design is to know that increasing from a .2 m wheel diameter to .34 m adds 80 kg to the total mass of the rover (not shown in figure).

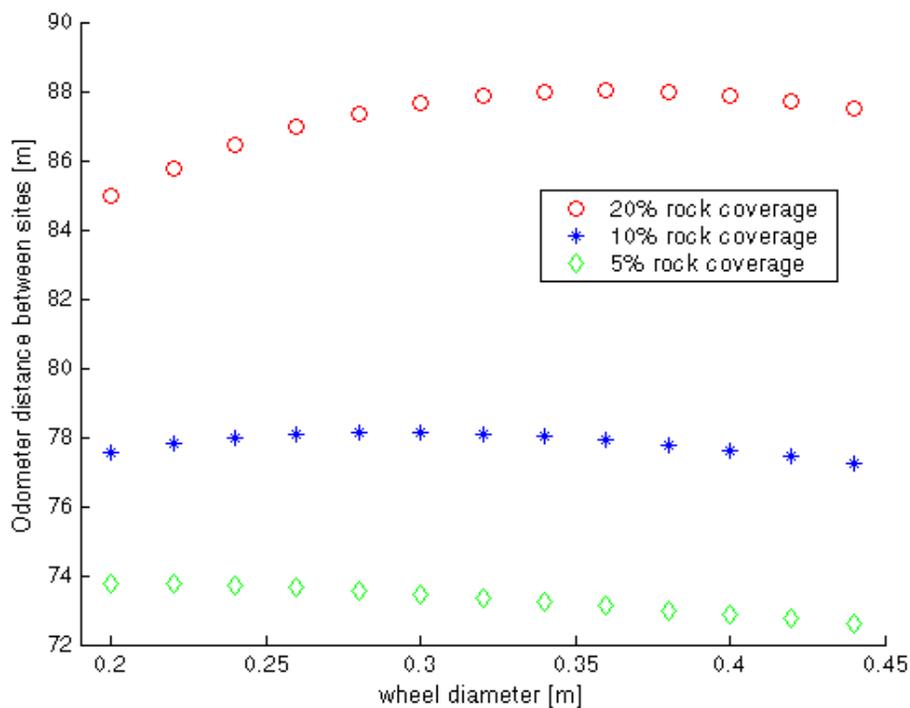


Figure 4.2.5: The MER-like trade space showing the traveled distance between sites versus the rover wheel diameter

At this point the trade is still unclear between having a large rover that can drive over more obstacles versus having a smaller rover that can maneuver between the obstacles it encounters.

4.2.2 MSL Scenario Trade Space

In order to see how the tool would handle the projected architecture traits of the Mars rover 2009 mission, a trade space was calculated. The values used are shown in Table 4.2.8.

Table 4.2.8: Values for MSL trade space calculation; Values not shown were set to MER default values

<i>Science Vector</i>		<i>Design Vector</i>	
Landing Ls [deg]	330°	Lifetime [sols]	200:100:800
Acquisition Payload	RAT, MUM, magnet, scoop	Communication	DTE, DTE-LMO, DTE-HMO, DTE-HMO_X
Rock Coverage	5%	Power Source	RTG
Samples per site	1	Wheel Diameter [m]	0.4:0.1:0.8
Instruments	Micro. Imager, APXS, Mössbauer Spectrometer, Mass spect. Gcms, LIBS, point IR Spect. Stereo pancam, Oxidation effects instrument, XRF	<i>Autonomy:</i> Long Distance Navigation Short Distance Nav Acquisition Night Time Processing Computers	A1, A3
Distance between Sites [m]	500		
Site Size [m]	15		

Figure 4.2.6 and Figure 4.2.7 were both calculated and used to compare where Pareto front designs fell in the different plots. The red and blue colors distinguish 1-computer from 2-computer designs respectively. The four shapes represent the four autonomy levels possible for short distance, long distance traversals: A1/A1, A1/A3, A3/A1, A3/A3. Looking at Figure 4.2.6, it seems that the A3/A3 designs dominate the Pareto curve and do so with a single computer. Figure 4.2.7 is a little less informative by itself except that the A3/A3 designs dominate the productivity levels and A1/A1 can generally be found toward the low end of productivity. The change in wheel diameter causes the separations from one total mass to another.

Four rover architectures were identified and compared across the two plots. They are labeled numbers 1-4. When comparing Figure 4.2.6 and Figure 4.2.7, it is interesting to compare where designs 1-4 fall in both plots.

Table 4.2.9: Design Vector parameters for comparing designs 1-4

Parameter	Design 1	Design 2	Design 3	Design 4
Mission duration	200	200	800	800
Wheel diameter [m]	.4	.4	.4	.4
Power (solar not an option)	RTG	RTG	RTG	RTG
Communication	DTE	DTE	DTE	DTE
Short, long distance autonomy	a1a1	a1a3	a3a3	a3a3

The difference between designs 1 and 2 is their autonomy levels. Design 1 has multiple autonomy/night options due to multiple indices at that point while design 2 has no nighttime sample processing capability. The difference between designs 3 and 4 is that design 3 only has one computer versus two, and is not capable of nighttime sample processing.

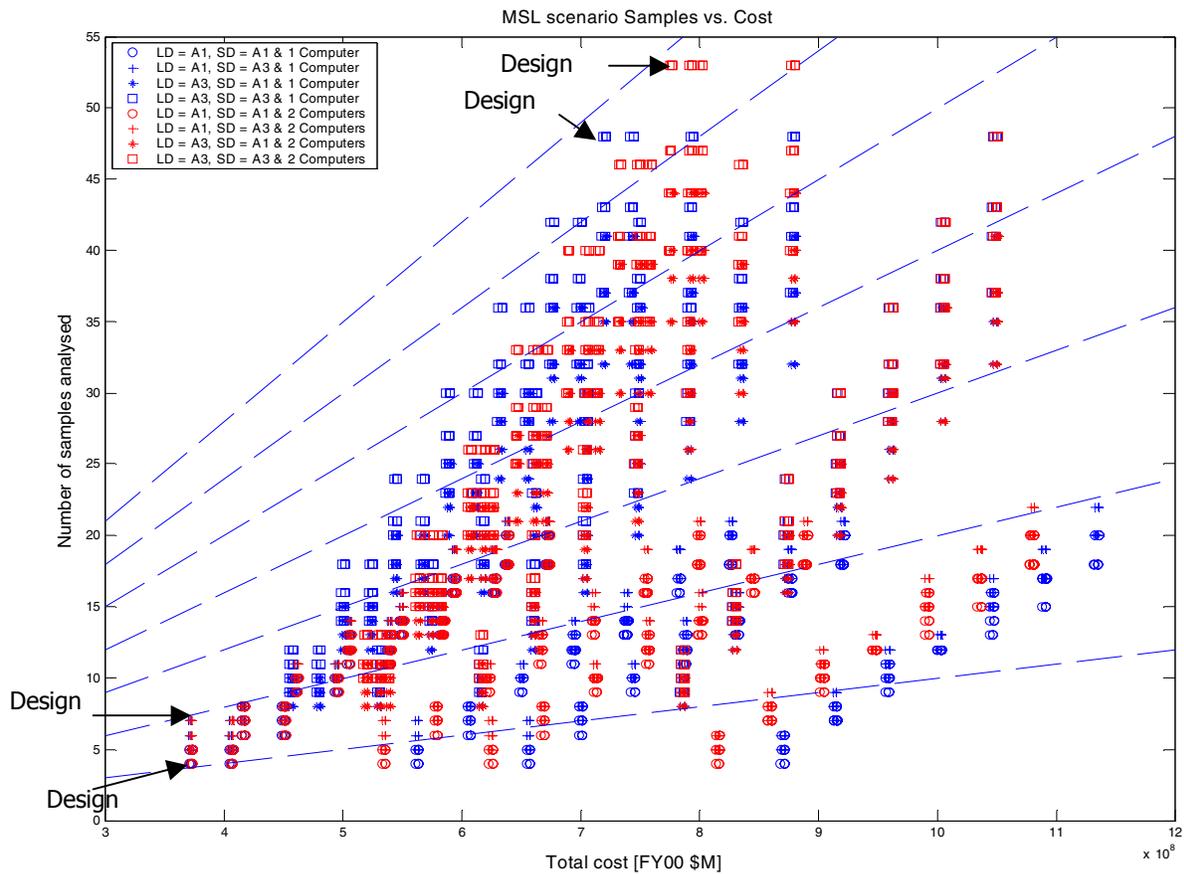


Figure 4.2.6: Total Number of samples vs. Cost for an estimated MSL trade space.

Comparing the two figures, it is clear that the four designs perform similarly within the two chosen plots. More interestingly, designs 3 and 4 are actually the best designs for both metrics (number of samples over mass and cost). Isoperformance lines in Figure 4.2.6 show that designs 3 and 4 have the best performance. They are both indeed closest to the isoperformance line with the largest slope. The same result is observed in Figure 4.2.7. These two designs are the best investments in terms of mass and cost.

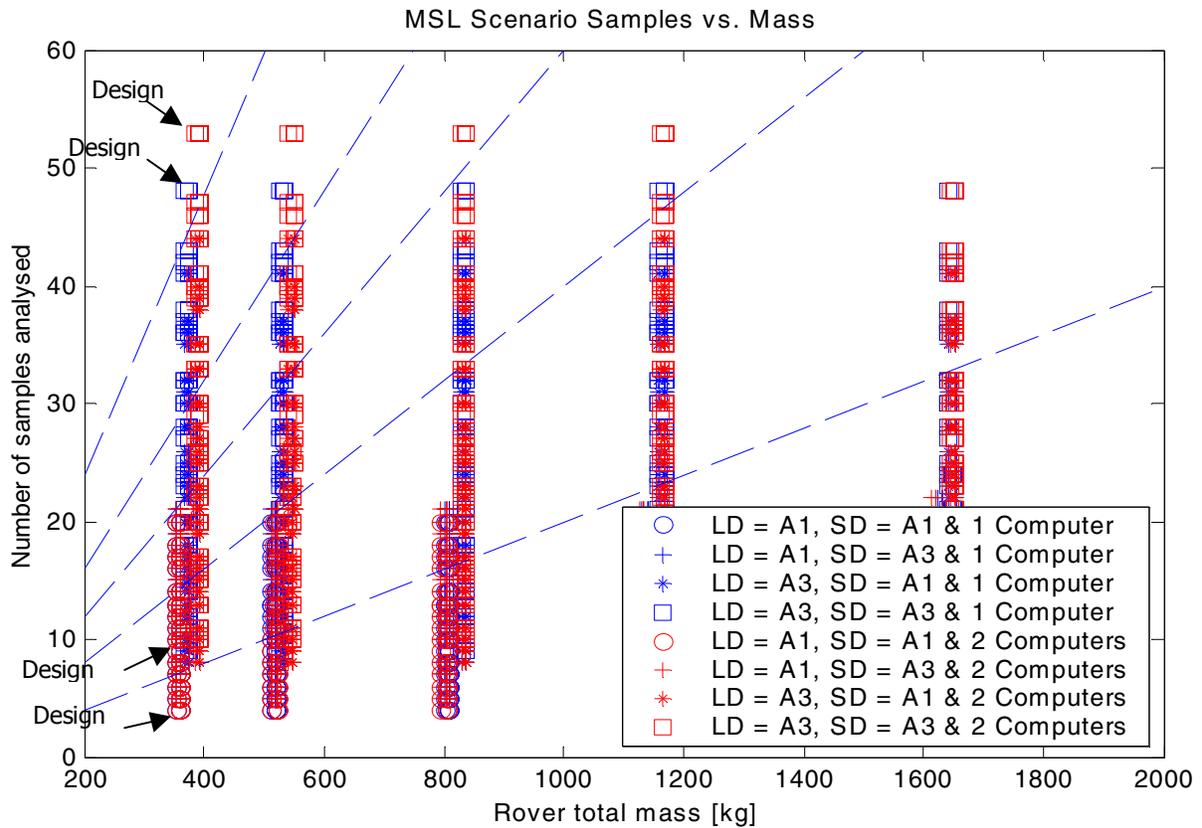


Figure 4.2.7: Rover total mass vs. Number of samples for an estimated MSL trade space.

4.2.3 Complexity

Loading a design space and plotting the number of samples analyzed vs. the complexity rating demonstrates the effect of increasing rover design complexity as shown in Figure 4.2.8.

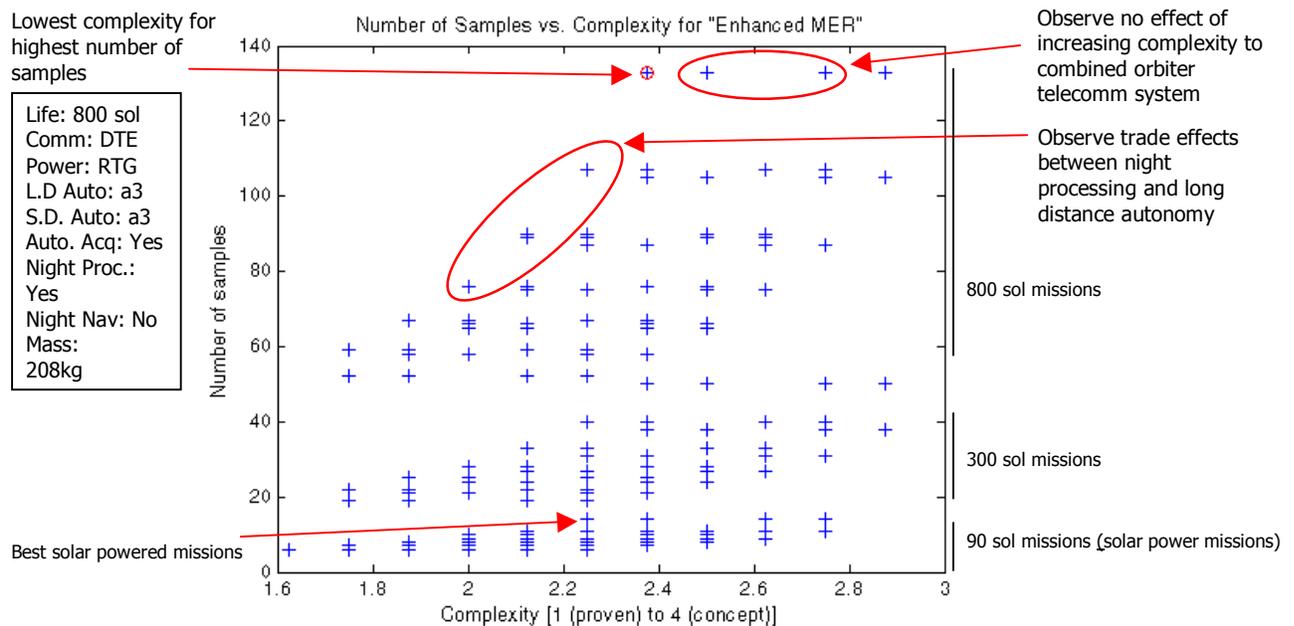


Figure 4.2.8 Number of Samples vs. Total Complexity for a MER mission

The optimum rover designs are ones that yield a high number of samples returned for the lowest amount of complexity, which corresponds to the upper left region in figure 1. For a MER mission scenario, a Pareto front is visible with several designs circled. The top most design on the front gives the best ratio of samples to complexity, and is the best design choice if the user is willing to accept nuclear power and high levels of autonomy, as shown in the design details box at left. If the user wants to trade off different levels of complexity, then designs leftward on the front should be examined. The three circled designs show the effect of trading night processing and the levels of short and long distance autonomy. Additionally, in this mission scenario, there is a fractal effect, where the different mission lifetimes each have smaller Pareto fronts, as indicated on the right vertical axis. The bottom most front is for 90 sol solar powered missions and the best design is pointed out in the lower left corner. Note that for nearly all points, there is a multiplicity of designs that usually reflects differences in wheel diameter and mass.

4.3 Conclusions

4.3.1 Technology Investments

In general the trade spaces analyzed have shown that autonomy, power sources and avionics packages have the most effect on increasing the science return for a given mission scenario. These areas will have the most impact on mission duration and productivity. Research and development should be focused on these areas in order for future Mars surface exploration missions to have a greater science return.

4.3.2 Future Work

The work previously discussed has provided a strong structure and basis for a Rapid Modeling of Mars Robotic Explorers tool. In addition, the first draft of the models and data has filled in this structure to provide an operational and useful tool. While this took an exceptional amount of effort and work to

accomplish, there is still much more work that could be done to improve the tool. Brainstorming among the team members, and comments from reviewers, has led to the following list of actions that should be taken to improve the tool in the future:

- **Rover subsystem model** – The Rover module is one of the most complex and difficult modules of the tool. This module is in charge of wrapping the payload of the rover with structural support, a mobility system, and a thermal system. While this is a very difficult task to accomplish, the validation tests that have been run to date show that there may be something that has not been modeled or has been modeled incorrectly. The most obvious pointer to this inaccuracy is the approximate 60 kg difference in the mass estimate for MER from the modeling tool and the actual MER mass.
- **Cost module** – Deriving accurate cost estimation relationships is a very difficult and long process. While the cost module currently in the tool provides a very rough estimate of cost, this cost is based solely on subsystem estimates, mass of the rover, and a very simple operations model. Several of the cost models used are known to be inaccurate and/or developed for a different type of system. In particular the rover and operations cost models should be re-examined. The possibility of using packing density, as opposed to or in addition to the rover mass, to predict the cost of the rover should also be followed up on. Additionally, the addition of a model to predict the cost of developing and testing autonomy would be very useful.
- **Communications module** – During the Critical Design Review it was noted that communications models developed at JPL have gotten different trend results than those output by the MIT Communications module. Specifically, the MIT model shows that high orbit telecommunications assets produce an architecture that is equally productive as architectures with only DTE and more productive than those with low orbiting assets. The JPL models show that architectures with low orbiting assets are more productive than those with high orbiting assets which are in turn more productive than pure DTE options, which makes intuitive sense. This trend result should be examined in the future.
- **Reliability** – The current model assumes that all architectures have the same reliability and risk, since no failure scenarios are considered. While this is a good assumption for the first draft of a model, it is clearly not the case. Therefore, reliability and risk models should be added to any future versions of the model.
- **Odometry Multiplier** – During the Critical Design Review it was noted that models developed at JPL have returned different trend results than those output by the MIT model in terms of the odometry multiplier. The odometry multiplier is the multiplier used to convert straight-line distance into actual distance traveled, and is a function of wheel size and rock distribution. It was noted that the odometry multipliers output by the JPL models are consistently higher than those output from the MIT models, implying that in the JPL models the rovers need to travel further real distances to report the same straight-line distance.
- **Landing Dates** – The current model assumes either an average or worst-case scenario for any function that depends on landing dates. Examples of these functions include the amount of sunlight received on the surface of the planet and the distance between Mars and the Earth. In actuality these types of functions can vary depending on the time of year and even the absolute date. While an average or worst-case scenario is a good assumption for the first draft of a model, future versions should take this variation into consideration.
- **Simulation Performance** – The current set of code has been written to provide functionality and to be as accurate as possible in the physical models. However, very little effort was given to optimizing the code for memory usage and/or speed. This is currently one of the major limitations of the tool and should be looked at for any future versions.
- **New Technologies** – The current version of the tool assumes only current technologies throughout. Future versions should include the possibility of upgrading to a more advanced technology and should model the increase in both the performance and the risk of using these new technologies.

-
- **New configurations and architectures** – The current code only allows rover designs based on the 6-wheel rocker-bogie rover configuration. Including other configurations (such as 4-wheel designs) could add significant value to the tool. Additionally, the architectures currently modeled are limited to single rovers without an active lander. Adding the necessary models to analyze architectures with multiple rovers and/or active landers would also be very valuable.
 - **Complexity models** – While the tool currently has the capability to roughly estimate the complexity of each design, several suggestions were made at the Critical Design Review on how this model could be improved. These suggestions include changing the name from complexity to maturity to better reflect the nature of what is being modeled, using the known and accepted Technology Readiness Levels, and including weightings to account for the difficulty and/or importance of each different technology.
 - **Utility model** – The current tool designs rovers around a given Science Vector, which includes the specific scientific payload to be included on the rover. An attempt could be made in the future to find a utility function relating scientific instruments to scientific return to allow the code to optimize not only the engineering design around a specific payload, but to optimize the payload as well.
 - **Polar Landing Site** – The current code does not allow for the possibility of landing at the polar regions of Mars. Since this is considered one of the more interesting regions to explore and is in consideration for future Mars missions, this ability should be added to the code.
 - **Launch, Cruise, and EDL Models** – The current design tool designs a rover based on the assumption that it has already landed on the Martian surface. While this is certainly a useful design tool, no Mars mission is complete with only a rover. Launch, cruise, and entry, descent, and landing (EDL) stages are all very costly and complex and can affect the mission design. These stages are all very difficult to model and could take a separate modeling effort as large, if not larger, than the one that was put forth for the rover design. While the modeling of these stages may be accomplished through separate tools, the utility of the overall systems tool would increase if the ability to analyze these portions of the missions were included.
 - **Avionics** – The current model bases avionics designs off the MER avionics. While computers can be added to this set of avionics, increasing the size of the avionics, there is currently no way to decrease the functionality, and therefore size, of the avionics systems. This capability should be added in the future.
 - **Consumables** – In the current code, the Autonomy module discards any rover design that completely uses up either its science instruments or acquisition tools due to consumables. However, not all instruments require an acquisition tool to function. Therefore, in the future, those designs in which the acquisition tools consumables are used up, but which still contain scientific instruments that do not require those acquisition tools, should be considered in the trade space

Appendix

A Project Code

See attached CD-Rom for the complete MATLAB code for the Rapid Modeling of Mars Robotic Explorers program.

B User Manual

This manual has been developed for use with the Revision 1 Code. It outlines the steps required for setup, creation, and post processing of a Mars rover design trade space. The rover design tool takes into account only the environmental constraints that relate to Mars surface operations; constraints relating to the launch, entry, descent, and landing environments are not considered. Each rover design is created based on a particular value for each of several design parameters such as wheel diameter, type of power system, and capability for autonomous operation. The user controls the scope of the trade space by defining the sets of allowable values for these design parameters. After a trade space has been created, there are several ways to compare the rover designs and visualize trends.

The rover design algorithms were written using Matlab 6.5. A graphical user interface (GUI) was created to simplify use of the code and evaluation of the results. Interactive help text aids the user in the design and analysis processes, and step-by-step directions for use of the tool can be displayed at any time. The design tool can also be executed from the Matlab command line.

Each rover design occupies 60 KB of space in the Matlab workspace. For large trade spaces, the memory requirements for storing the rover designs alone can quickly reach the level of hundreds of megabytes.

This software is intended to help rover designers make overarching architecture decisions, given a desired landing location and payload for the rover. It does not aid in the detailed engineering design that is required for manufacturing or production.

B.1 Overview

Setting up a rover design trade space consists of four processes, the first two of which may be performed in either order. The first step is to define the Science Vector, which defines the science and navigation payload and site-specific information. More information about the parameters in the Science Vector can be found in the Instruments, Acquisition, Autonomy, and Environment chapters in the design document. The Science Vector has a default setting similar to Mars Exploration Rovers (MER) values. The second step is to define the Design Vector. In the Design Vector, the user determines the size of the rover design trade space by limiting specific design options such as the autonomy level or mission lifetime. This enables the user to choose a broad overview of the trade space, or to explore an interesting area of the trade space in greater detail. The third step is to create the rover designs. This process can take some time, so a progress bar displays the progress of the code while status text on the GUI displays a real-time tally of the number of valid and invalid designs produced. Finally, with a trade space created, post-processing can be done on all feasible solutions. This includes viewing each of the subsystem values for a given rover design, creating two or three-dimensional plots of the trade space, and viewing a three-dimensional rendering of the rover.

B.2 Creating a Trade Space

The trade space creation and analysis tool has a graphical user interface that simplifies the creation, analysis, and visualization of rover design trade spaces. To begin using the tool, type `rover4mars` at the MATLAB prompt to open the main GUI window, shown in Figure B.1.

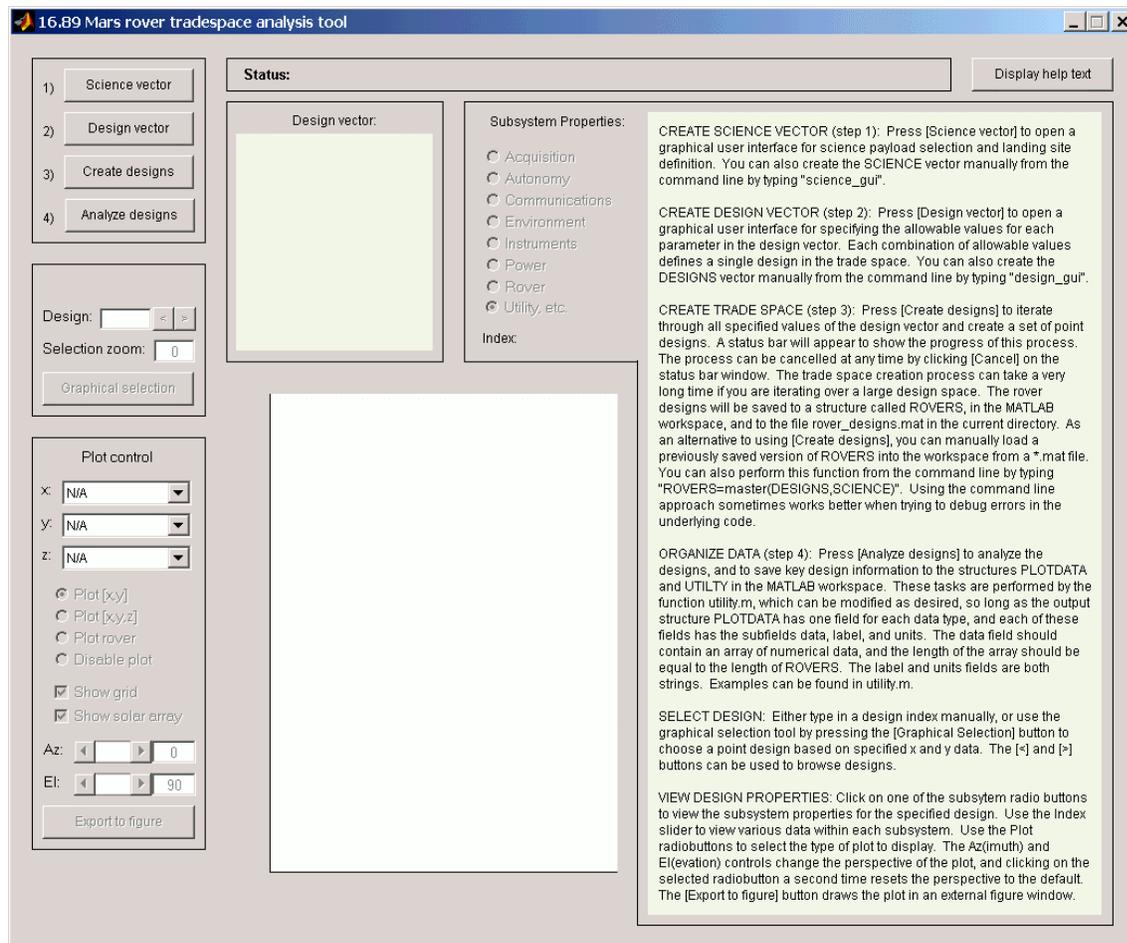


Figure B.1 The main graphical user interface (GUI) at startup. Called by typing rover4mars at the MATLAB command line.

At startup, the only interface controls available to the user are the buttons [Science vector], [Design vector], [Create designs], [Analyze designs], and [Display help text]. All other interface controls are disabled. The *Subsystem Properties* window displays the help text, which contains a step-by-step overview of the trade space creation and analysis process. The user may press the [Display help text] button at any time to redisplay this text. Note that the first two steps, selecting the Science and Design Vectors, may be performed in either order without affecting the results.

B.2.1 Selecting the Science Vector

The first step in the trade space creation process is to specify the rover payload, consisting of science and navigation instruments and acquisition tools, and to define the landing site characteristics. This is accomplished by pressing the [Science vector] button, which opens the Science Vector selection GUI, shown in Figure B.2. The science GUI is dynamically generated based on the instruments and tools available in the lookup tables contained in `create_acquisition_look_up.m`, `create_instruments_look_up.m`, and `create_navigation_look_up.m`. Adding an instrument to any of these lookup files automatically results in a new option being displayed on the science selector GUI the next time it is called. The science GUI can be opened directly from the Matlab command line by typing `science_gui` at the prompt.

The user selects the options corresponding to the desired payload and landing site, and submits the Science Vector by pressing the [Submit science vector] button. This saves a structure called `SCIENCE` containing the selected values to the Matlab workspace. Although selections are saved to the Matlab workspace, they are not reflected as saved changes if the Science Vector GUI is re-opened; each time the Science Vector selection GUI is opened, the selected items default to values based on MER. The Science Vector can be re-submitted at any time before the [Create designs] button is pushed.

Descriptions of the landing site characteristics can be found in the Environment chapter of the design document. Additional information about each of the instruments and acquisition tools can be found in the corresponding lookup table file.

There are some options in the Science Vector that are automatically selected or deselected, and cannot be explicitly selected by the user. In the *Acquisition tools* window, the processing options are not available because they are sized according to the science instrument payload. The two options listed as 'not available' in Figure B.2 are placeholders for components that have been commented out of the lookup table due to insufficient information. In the *Science instruments* window, the pancam is automatically selected but not accessible because the autonomy algorithm requires a pancam. In the *Navigation instruments* window, the navcam is selected but not accessible for the same reason.

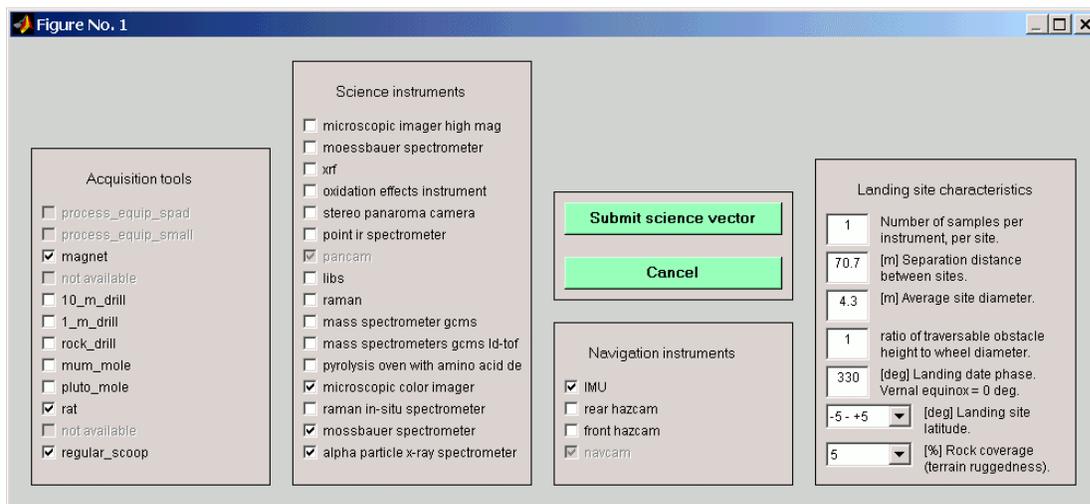


Figure B.2 The Science Vector selection interface is dynamically generated based on the contents of the lookup tables for the given subsystems.

B.2.2 *Selecting the Design Vector*

The second step in the trade space creation process is to specify a set of allowable values for each parameter in the Design Vector. This is accomplished by pressing the [Design vector] button, which launches the Design Vector selection GUI, shown in Figure B.3. Once selections have been made and the [Submit design vector] button is pressed, the values are saved to the `DESIGNS` structure in the Matlab workspace. At least one state must be selected for each line item. If there is an entry missing, the GUI will display an error message when the [Submit design vector] button is pressed. If the `DESIGNS` structure is successfully saved, the Design Vector selection GUI will close. If a `DESIGNS` structure already exists in the MATLAB workspace when the Design Vector selection GUI is started, the selected items change to reflect the Design Vector contents of that existing `DESIGNS` structure. The Design Vector selection GUI can be run directly from the Matlab command line by typing `design_gui` at the command prompt.

Following is a description of each parameter in the Design Vector. Each selected value or option increases the amount of time it takes to compute the entire rover trade space. It is suggested that if the user is unfamiliar with the tool, to initially run a limited trade space by deselecting most options.

- **Mission duration:** contains the values for the mission lifetime, in number of sols. The range of values is unlimited. The user can input a series of values, or use standard Matlab vector notation to specify, for example, an extended range of values, where the first and last numbers are the min and max values and the center number, separated by colons, is the step size. In Figure B.3, the lifetime range only includes a 100 and 200 sol mission.
- **Wheel diameter:** box contains the values for the diameter of the rover wheel, in meters. This quantity is set the same way as the Mission duration values. Due to a baseline avionics system modeled after MER, most designs with wheel diameter less than 0.2 m are rejected as being invalid.
- **Number of RAD6000 equivalent CPUs:** the user can determine the computational processing capability of the rover by specifying the processing capability in terms of RAD-6000 equivalent CPUs. In Figure B.3, half of the rover designs will have one RAD-6000 equivalent computer and the other half will have two.

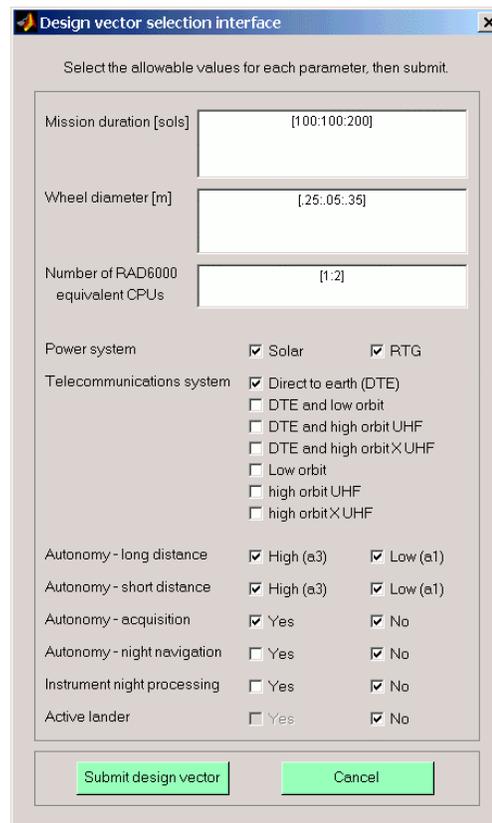


Figure B.3 The Design Vector selection interface. Default values span MER and MSL mission parameters.

Checking options makes the following choices. The user should deselect undesired options.

- **Power system:** the two options for the power system are solar power and radioisotope thermoelectric generators (RTGs). Batteries may appear in designs with either power system, as

determined by the design algorithm. More information about these options can be found in the Power subsystem chapter of the design document.

- **Telecommunication system:** there are seven available telecommunications architectures, assuming that there are low and high orbiting satellites capable of providing a communication link back to Earth. More information about these options can be found in the Communication subsystem chapter of the design document.
- **Autonomy-long distance** and **Autonomy-short distance:** there are two levels of capability for the autonomy algorithms used in traverse planning and execution. Level 'a1' is the level of autonomy used by MER, and level 'a3' describes the higher autonomy being considered for the 2009 Mars rover mission. More information about the autonomy levels can be found in the Autonomy subsystem chapter of the design document.
- **Autonomy-acquisition:** affects how quickly the rover is able to acquire samples. With the autonomous acquisition option on, the algorithm calculates the acquisition process time assuming there is no delay for Earth communications.
- **Autonomy-night navigation:** allows the rover to perform traverse operations at night.
- **Instrument night processing:** allows the rover to process samples at night. Sample acquisition still occurs only during daylight hours.
- **Active lander:** this option is currently not modeled.

Each of these options can have both 'yes' and 'no' selected. Selecting both 'yes' and 'no' for a particular option doubles the total number of designs in the trade space.

B.2.3 Creating Designs

The third step in the trade space creation process is to create a set of rover designs based on the Science and Design Vectors. This is accomplished by pressing the [Create designs] button on the main GUI. While the algorithm is running, a status bar and status text are displayed to provide feedback to the user. These feedback components are shown in Figure B.4. The status text updates in real-time to display the number of valid and invalid designs encountered so far. This step of the process can be time-consuming when large ranges of values in the Design Vector are specified.

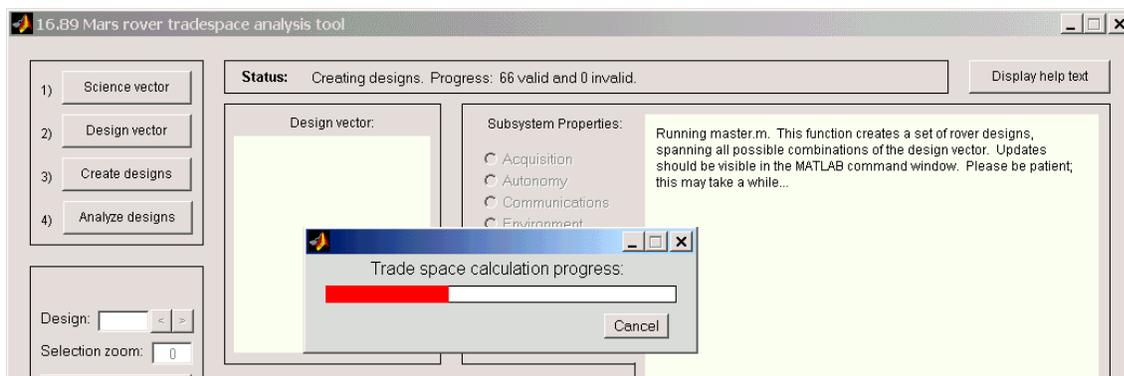


Figure B.4 The status bar and status text reflect the current progress of the trade space calculations.

The design creation process can be cancelled at any time by pressing the [Cancel] button on the status bar. Upon either cancellation or successful completion, the resulting rover designs are saved to the MATLAB workspace in the `ROVERS` structure. In addition, the `DESIGNS`, `SCIENCE`, and `ROVERS` structures are written to the file `rover_designs.mat`, overwriting any existing information.

Warning: if the user wishes to save an existing trade space, the file `rover_designs.mat` should be renamed. When another trade space calculation is started, the file `rover_designs.mat` will be overwritten with the contents of the new trade space, and any existing data will be lost.

The GUI provides feedback about how many valid and invalid designs were found, and which subsystems found invalid designs. Figure B.5 shows an example of this information.

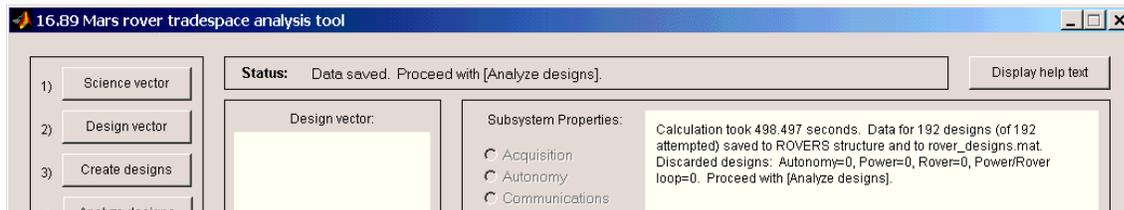


Figure B.5 An example of feedback after a trade space calculation has completed.

B.2.4 Preparing Designs for Analysis

The fourth step is to prepare the data for visualization and comparison. This is accomplished by pressing the [Analyze designs] button. This button calls the function `utility`, which creates the `PLOTDATA` and `UTILITY` structures. The contents of the `PLOTDATA` structure are read into the GUI, and are used to populate the x, y, and z popup menus. In addition, this button enables the remaining user interface controls on the main GUI, allowing the user to proceed with comparing different designs. The file `utility.m` can be easily edited to add or remove data choices from the plot lists. A fully enabled GUI is shown in Figure B.6.

It is also possible to load a previously saved version of `ROVERS` into the workspace, skipping steps one through three of the trade space creation process. This allows the user to modify the `utility` function to change the way that the existing data are compared, without re-calculating the trade space.

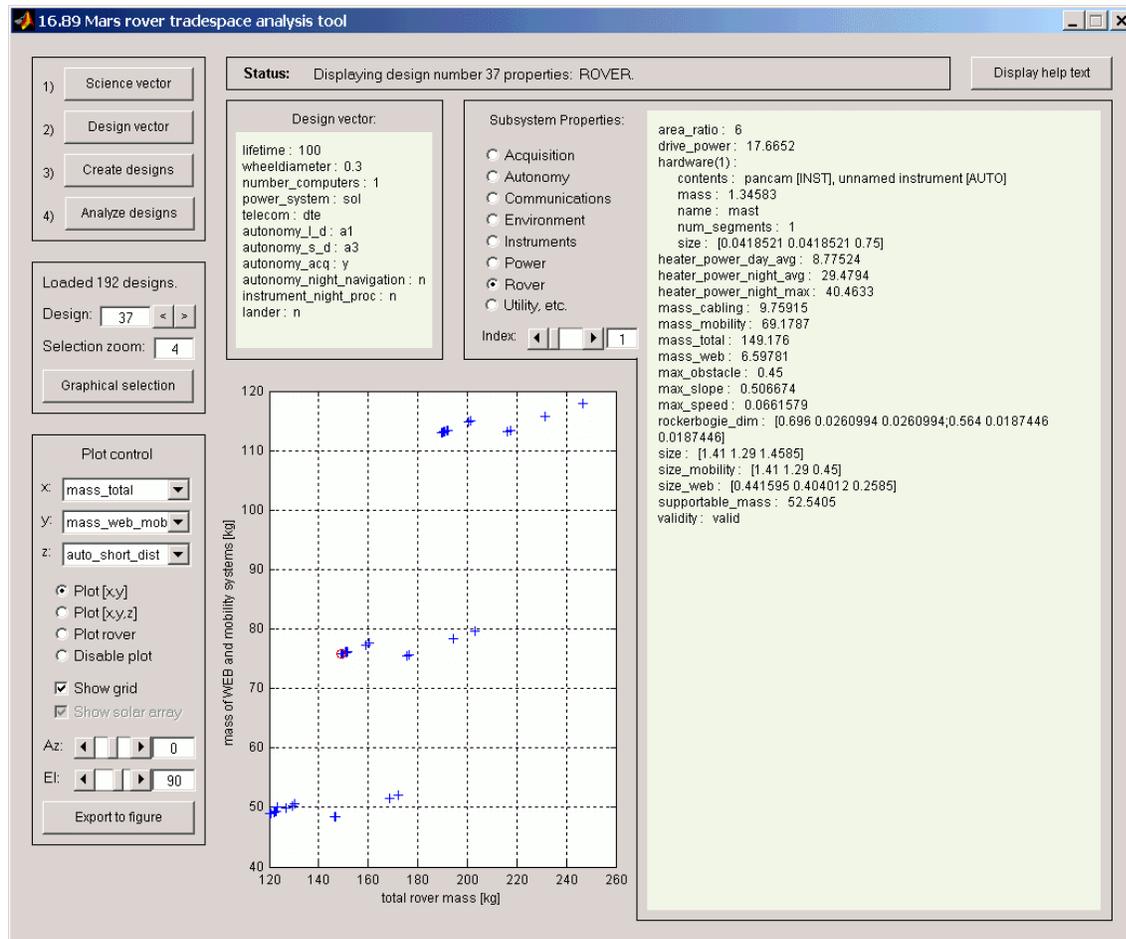


Figure B.6 Once the user presses [Analyze designs], the user interface is fully enabled and plots are ready to be generated

B.3 Analyzing Designs

The graphical user interface has several functions that facilitate comparison of rover designs and analysis of the trade space.

B.3.1 Viewing Subsystem Properties

On the right side of the main GUI there is a window that displays the properties of a given subsystem. The user selects the desired subsystem by choosing the corresponding radio button, and then uses the Index slider to scroll through associated hardware for that subsystem. Figure B.6 shows the properties of the Rover subsystem for design number 37 in this trade space. For the Autonomy, Communications, Power, and Rover subsystems, the Index slider determines which element of the hardware array is displayed. For the Acquisition and Instruments subsystems, the Index slider is used to view the properties of a specific tool or instrument. An example of this is shown in Figure B.7.

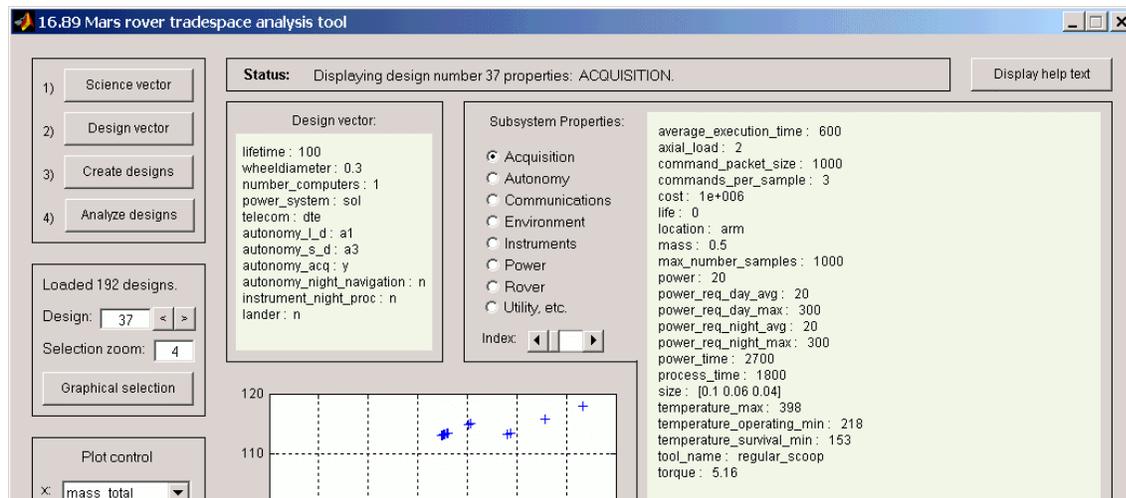


Figure B.7 For the Acquisition and Instruments subsystems, the Index slider determines which instrument or tool is displayed.

B.3.2 *Selecting a Design*

As seen in both Figure B.7 and Figure B.8.a, there is a box on the left side of the GUI that displays the total number of designs loaded in the trade space. Here, the user can select a specific design by placing the cursor in the box and entering a number, or by using the [**<**] and [**>**] arrow buttons to scroll through the designs. The properties of the selected design are displayed in the Design vector and Subsystem Properties windows, and the plot window updates to highlight the currently selected design with a red circle, or if Plot rover is selected, with a rendering of the design.

Pressing the [Graphical selection] button opens a figure window with cross-hairs on a plot of the trade space, as seen in Figure B.8.b. The user can click on a particular point design on the plot, and the GUI display updates with the design number and properties of that point design. The plot axes for graphical selection are determined by the x and y popup menus in the Plot control box. Above the [Graphical selection] button is the Selection zoom box, where a zoom factor can be specified. If the zoom factor is greater than zero, then clicking on the graphical selection plot produces a second figure, which is zoomed in by the specified factor. Clicking on a point in this second window selects a point design. The ability to zoom into an area of the plot facilitates the selection of a particular point in a dense region of designs.

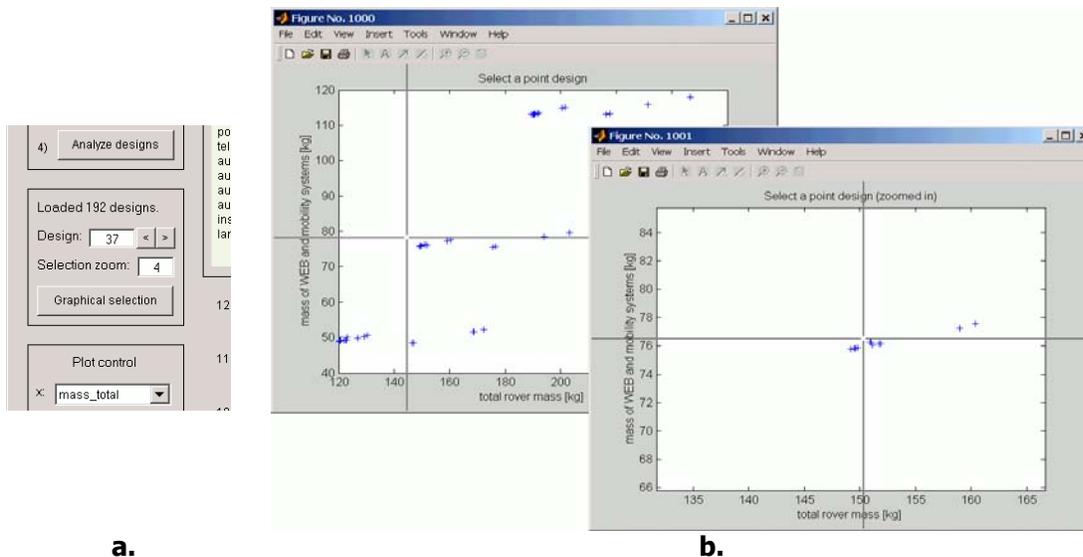


Figure B.8 a) The design number, selection zoom box, and [Graphical selection] button. b) The graphical selection utility allows the user to click on a point design to select it. If the selection zoom is greater than zero, then clicking on a point opens a second selection window that is zoomed in by the specified factor.

B.3.3 Visualizing Results

The main GUI has several functions built in to aid visualization of the rover trade space. In the Plot control box, there are popup menus for the x, y, and z-axes. These include parameters that may be useful in finding optimal rover designs and visualizing trends. Plots can be created in either two or three dimensions. The user selects the desired parameters from the popup menus, as shown in Figure B.9.a, and then selects either the Plot [x,y] or Plot [x,y,z] option. The plot will be automatically generated, and a red circle will highlight the currently selected design. If a two-dimensional plot is displayed and the user makes a choice in the z-axis pull-down menu, a three-dimensional plot, such as that shown in Figure B.9.b, will be generated.

Three-dimensional plots are useful due to the large number of input and output design parameters. The third dimension allows the user to visualize (such as in Figure B.9.b) how a design variable (such as level of short distance autonomy) affects other quantities (such as number of samples).

Opening the file utility.m and adding or deleting options can easily modify the choices given in the default plot axis menus. If utility.m is modified, press [Analyze designs] to update the GUI with the changes.

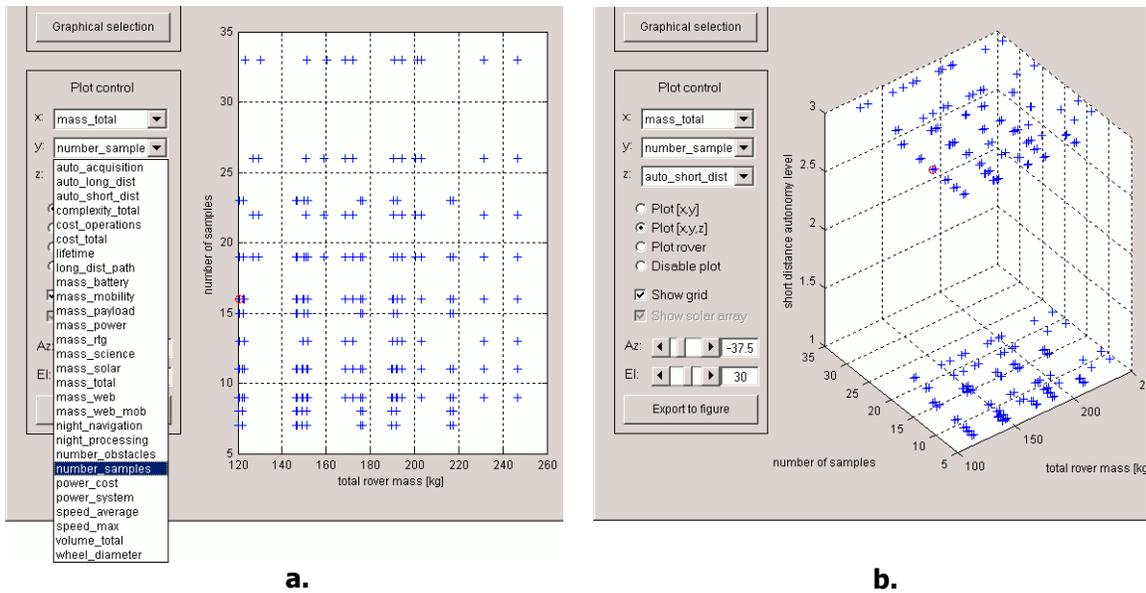


Figure B.9 a) A 2-D plot and a default plot pull-down menu. b) A 3-D plot.

Selecting Plot rover displays a three-dimensional rendering of the selected rover design. Figure B.10.a shows an example rover design as it appears in the main GUI. This graphical representation of the rover design depicts the wheels, the main rover body, the communications antenna, and the solar panel (if one is present in the design). If an arm and mast are present in the design, they are displayed along the horizontal and vertical directions, respectively.

All three plot types can be exported to an external Matlab figure by pressing the [Export to figure] button. When a rover rendering is exported to a figure, as shown in Figure B.10.b, the solar panel (if present) is made translucent to show underlying detail.

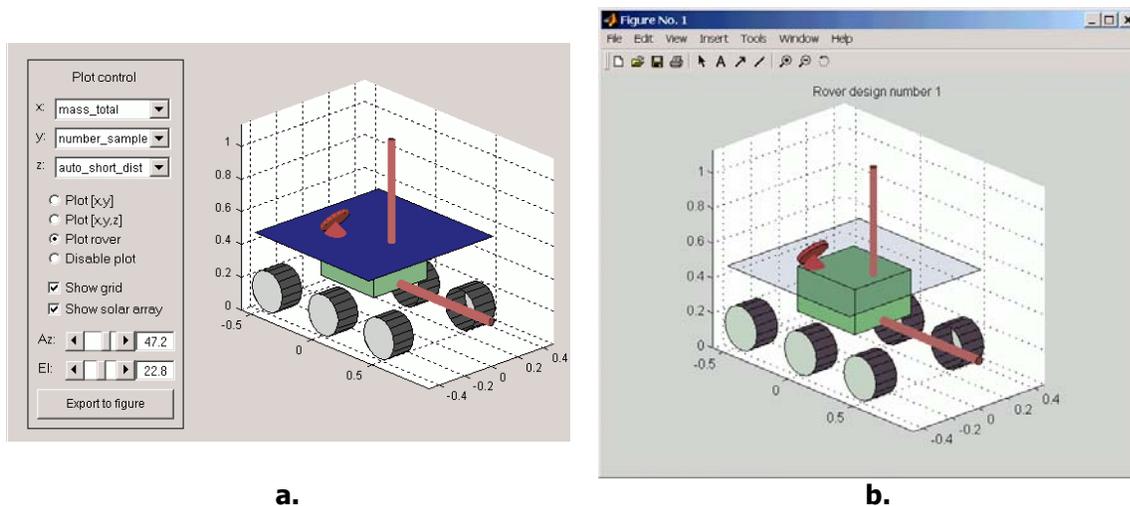


Figure B.10. a) A rendering of the currently selected rover design. b) When the rendering is exported to a figure, the solar panel is made translucent.

At the bottom of the Plot control window there are Az (azimuth) and El (elevation) sliders. They can be used to change the orientation of the on-GUI plot. The Disable plot radiobutton can be selected to disable plot generation.

B.4 Command Line Interface

The following commands may be used to run the design tool from the command line.

```
science_gui           % creates SCIENCE structure
design_gui            % creates DESIGNS structure
ROVERS = master(DESIGNS, SCIENCE); % creates rover designs
[PLOTDATA, UTILITY] = utility(ROVERS); % analyzes rover designs
```

Note that `science_gui` and `design_gui` are both graphical user interfaces, so additional user input is required at these two steps. The resulting rover designs are completely contained in the output `ROVERS` structure; the `PLOTDATA` and `UTILITY` structures contain additional information useful for visualization and comparison of results. The `DESIGNS`, `SCIENCE`, and `ROVERS` structures are automatically saved to the file `rover_designs.mat` by the `master` function, overwriting any previously existing data.

B.5 Validation

Validation of the user interface involves showing that the GUI can respond appropriately to both expected and unexpected user input, and that it provides useful feedback to the user about operational and algorithmic status, including errors. These issues are addressed in the implementation of the interface. The following three cases show examples of how off-nominal conditions are handled by the GUI.

Under nominal conditions, pressing the [Create designs] button calls the function `ROVERS=master(DESIGNS, SCIENCE)`. If either `DESIGNS` or `SCIENCE` does not exist in the workspace, the GUI prints an error message to the Subsystem Properties window, as shown in Figure B.11, and the GUI does not call the function `master`.

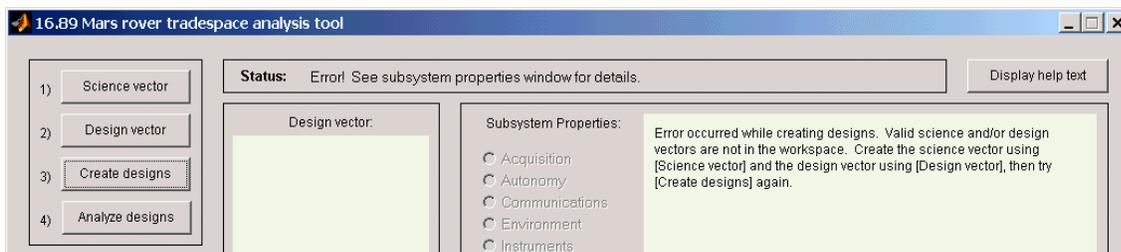


Figure B.11 The user is informed when [Create designs] is attempted without valid **SCIENCE** and **DESIGNS** structures in the workspace.

Similarly, pressing the [Analyze designs] button nominally calls the function `[PLOTDATA, UTILITY]=utility(ROVERS)`. If the `ROVERS` structure does not exist in the workspace, the GUI prints an error message to the Subsystem Properties window, as shown in Figure B.12, and the GUI does not call the function `utility`.

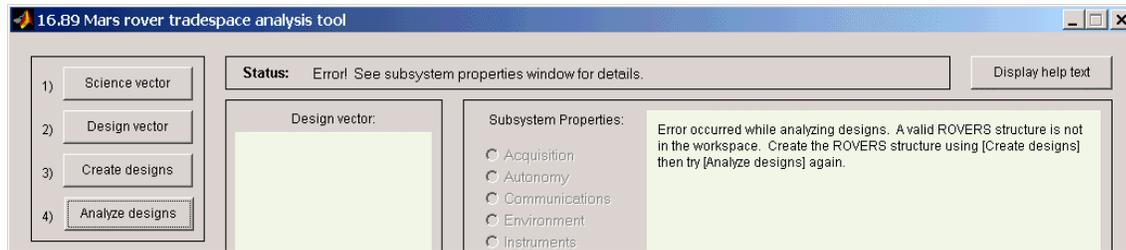


Figure B.12 The user is informed when [Analyze designs] is attempted without a valid ROVERS structure in the workspace.

If a run-time error occurs during execution of `master` or `utility`, execution of that function is halted and a description of the error is displayed on the Subsystem Properties window. To demonstrate this behavior, a reference was made in `master` to a non-existent entity called `I_am_not_a_function_or_variable`. On pressing [Create designs], execution of the function `master` halted and an error was displayed, as shown in Figure B.13.

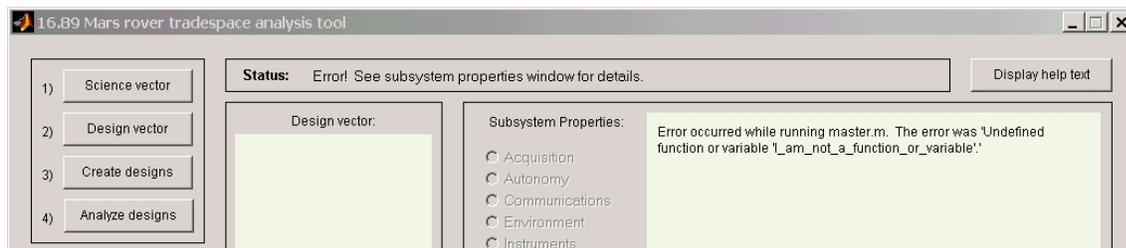


Figure B.13 The user is informed of an error. In this example, the function `master` attempts to access a non-existent entity called `I_am_not_a_function_or_variable`.

Real-time feedback to the user of nominal operational status and step-by-step instructions is described in the Graphical User Interface section of this document.

B.6 Expandability

The `guide` command in MATLAB provides a way to easily modify graphical user interfaces. The `rover4mars` and `design_gui` interfaces were built using `guide`, and may easily be modified using `guide`, if new functionality is desired.

The Science Vector selection interface was not built using `guide`; rather, it is dynamically generated by the function `science_gui`, based on the lookup tables in the files `create_instruments_look_up.m`, `create_acquisition_look_up.m`, and `create_navigation_look_up.m`. Changes to the Science Vector selector interface should be made by editing `science_gui.m` rather than by using `guide`. Additions to or deletions from the list of available science instruments, acquisition tools, and navigation instruments can be made by editing the lookup table files, and changes made to these files will automatically be reflected in the user interface when the GUI is dynamically generated at run-time.

D Project References

1. Goldstein, B. *Mars Geological Rover: Flight System Overview* [online]. Available at: http://karmann.jpl.nasa.gov/projects/Mars344/flight-system/flight_system_description.pdf. April, 2003.
2. Golombek, M., *Roving for Rocks on the Red Planet: the Mars Pathfinder Mission*. *EOS* Vol. 77, No. 49, Dec. 3, 1996, pp. 489, 492 – 493. Presented on website: http://www.agu.org/sci_soc/eosgolombek.html, April 2003.
3. Rosenberg, L. and Roust, K. *Mars Geological Rover Cost Analysis : DNP Cost Model Analysis Using PMCM2* [online]. Available at: [http://pso-lib.jpl.nasa.gov/dnp-lib/dscgi/ds.py/Get/File-1161/Mars_Geol_Rover\\$Anal-Rev_1.doc](http://pso-lib.jpl.nasa.gov/dnp-lib/dscgi/ds.py/Get/File-1161/Mars_Geol_Rover$Anal-Rev_1.doc). April, 2003.
4. Wertz, J. and Larson, W [ed.]. *Space Mission Analysis and Design, 3rd Ed.* Microcosm Press and Kluwer Academic Publishers. El Segundo, CA. 1999.