AIAA SPACE 2010 Conference & Exposition
30 August - 2 September 2010, Anaheim, California

AIAA 2010-8831

Surface Operations Analyses for Lunar Missions

Hari D. Nayar[*], Bob Balaram[*], Jonathan M. Cameron[*], Matthew DiCicco[*],
Thomas M. Howard[*], Abhinandan Jain[*], Yoshiaki Kuwata[*], Christopher Lim[*],
Rudranarayan Mukherjee[*], Steven Myint[*], Alex Palkovic[†],
Marc I. Pomerantz[*], Stephen Wall[*]

Jet Propulsion Laboratory, California Institute of Technology,
4800 Oak Grove Drive, Pasadena, CA 91109.

The Jet Propulsion Laboratory is developing the Lunar Surface Operations Simulator software package to support analyses for future NASA lunar missions. The package is built on and extended from previous simulation packages developed at JPL. It simulates mechanical motion, soil interaction, environmental, and physical processes. Physical process dynamics include environmental control and life support, thermal, radiation and power transients. An integrated architecture allows use of common models and enables interactions between components operating in different domains to be easily modeled. We describe recent developments and analyses performed to support lunar surface missions and analog field trials.

I. Introduction

The Lunar Surface Operations Simulator (LSOS)[1,2] is a dynamics simulation software package that models lunar surface systems and the lunar environment. Simulations of lunar mission scenarios are performed with LSOS to determine performance of surface assets and to analyze feasibility and optimize mission plans. Physics models of vehicles, habitats, and the lunar terrain are implemented in LSOS. LSOS also accurately models the locations of the sun and other planetary bodies with respect to the surface assets on the terrain. In addition, process models and the corresponding dynamic behavior of power, environmental control and life support, thermal and other systems are incorporated during simulations.

The LSOS software package has been extended from a family of dynamics simulation packages[3,4,5,6,7,8] developed at JPL for modeling, simulating and visualizing systems for planetary exploration missions. These packages have been used for simulating entry, descent and landing on Mars, rovers exploring the surface of Mars, the cruise, orbit-insertion, orbit phases for missions to Jupiter and Saturn and many other space and research projects.

Dynamics simulations are an essential component of the extensive engineering analyses performed in the formulation and planning stages of space missions. Software simulations provide the following capabilities that are expensive to duplicate with alternative approaches:

a. Replicating environmental conditions found in space or other extra-terrestrial bodies (for example, different or complex gravitational fields),

b. Investigating the feasibility of new concepts and evaluating alternatives,

c. Performing studies that cover a large range of time and space scales,

d. Performing Monte-Carlo and parametric analysis that modify variables in multiple simulation runs,

[*] Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109.
[†] Department of Computing Sciences, Villanova University, PA 19085.

e. Sharing and re-playing simulation results with wide and physically distributed analyses and planning teams,

f. Investigating alternative operational scenarios,

g. Validating and verifying against alternative analytical studies and physical analog trials, and

h. Visualizing planned operations.

LSOS has been developed for many of the reasons listed above. In the following section, we describe elements of LSOS. In Sub-section II.A we describe habitat vehicle models developed to support LSOS simulations. Another important component of LSOS, terrain models, are described in Sub-section II.B. The 3D visualization system is described in Sub-section II.C, path planning in Sub-section II.D and scenario scripting in Sub-section II.E. LSOS has been used to perform analyses in support of mission planning and system design. Two examples are reported in Section III. They are a study of power usage in a traverse from Shackleton Crater to Malapert Mountain and back (Sub-section III.A) and solar panel performance at Shackleton Crater and Malapert Mountain (Sub-section III.B). We conclude with a summary of the development of LSOS and future plans in Section IV.

II. Lunar Simulator Components

A. Vehicle and Habitat Models

LSOS simulations incorporate a number of vehicles and habitats. The Lunar Electric Rover (LER), shown on Figure 1a, is a concept vehicle proposed for rapid scouting missions on Lunar missions. Modeling the high-fidelity dynamics of the LER in LSOS was essential to ensure good physical models for exploring the locomotion and energy design space. An articulating multi-rigid-body model of the LER was developed and implemented in LSOS. An extension to the dynamics formulation was developed to handle the four-bar spring-loaded suspension mechanism.

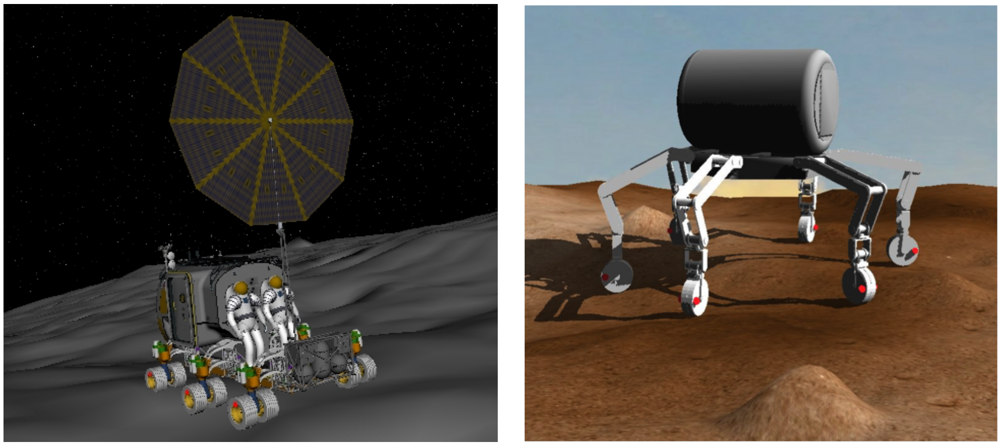
The ATHLETE, shown on Figure 1b, is a six-legged vehicle with wheels on each leg. The ATHLETE's suspension system is its legs. During simulations that incorporate the ATHLETE, its legs are actively controlled to comply to the ground to maintain the vehicle's chassis level when driving over undulating terrain. An algorithm was developed to

- keep all wheels in contact with the ground,
- keep the chassis level with adequate clearance from ground,
- keep the chassis heading in the drive direction
- keep wheel mounts vertical, and
- keep wheels steered correctly.

Other lunar surface systems modeled in LSOS are lunar habitat modules (see Figure 2a), the Altair lunar lander (see Figure 2b), and the Portable Utility Palette (PUP) shown on Figure 2c.The PUP has a solar panel and battery for energy generation and storage. LSOS has also modeled placing the PUP on the back of an LER rover as shown on Figure 1a. This option provides additional power to the LER and the PUP can be dropped off and left on its own to charge batteries for later use. LSOS can simultaneously incorporate models of any number of these vehicle and habitat elements in dynamic simulations.

The LSOS vehicle and habitat systems provide production and consumption modeling of numerous internal properties like O2, and water usage, CO2 production, and energy production (through solar panels) and consumption (through driving and on-board devices). For environmental control and life support (ECLS) parameters like O2, water and CO2, simple constant linear charge and discharge rates are used and provide enough fidelity to aid in the planning of long duration lunar excursions.

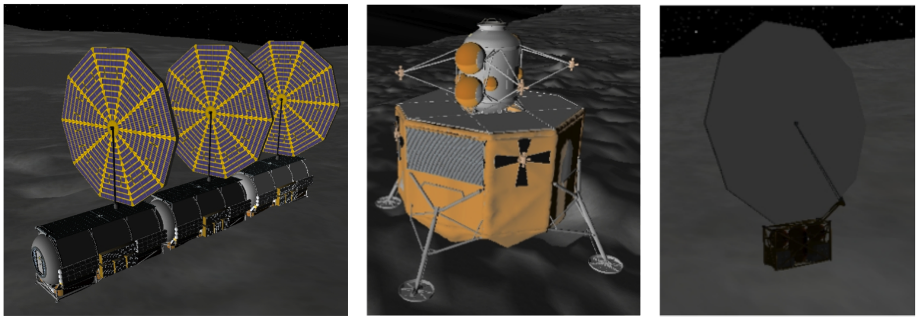
Energy modeling is more sophisticated and focuses on two core components: a consumption model based on vehicle velocities and on-board system energy usage rates and a production model based on solar panels. Battery models are used represent the total stored charge level on the vehicles and habitats. These batteries do not model any dynamics on their own and only represent storage of charge produced by the solar panels and consumed by the vehicle driving.



(a) LER with PUP model

(b) ATHLETE model

Figure 1: The LER vehicle carrying a PUP and the ATHLETE vehicle models.



(a) Habitat model

(b) Altair Lander model

(c) PUP model

Figure 2: The habitat, lander and PUP models.

The energy consumption model uses the vehicles center of mass and linear and angular velocity. The vehicles velocity is combined with a rolling resistance to calculate the basic driving load. The vertical component of linear velocity is used to calculate additional energy loads from going up and down hills. This energy consumption estimate has been compared to experimental results on a prototype LER rover. A major component of the vehicle energy consumption while driving is the vehicle mass, which is used to calculate the rolling resistance. To more accurately model long-term rover excursions, the LSOS simulator supports a concept of discrete changes in rover mass. This helps to model the addition of large sample masses, which would be acquired by scientists or the removal of the PUP mass when it is left on the surface to charge independent of the rover.

Energy production for surface operations comes from deployable solar panels. Power levels are estimated through the use of the LSOS rendering and visualization system. Each time the solar panel model needs to calculate the power draw, a special target painted with a emissive green color and of the same size and shape of the solar panel is made visible on the solar panel. A rendering of this green panel is made as viewed from the sun location and the number of green pixels is counted and compared to the maximum number of pixels calculated from another imaged rendered straight at the panel with no obstructions. Example of these rendered images is shown on Figure 3. Terrain features, other vehicles or other panels that occlude and shadow the panel will reduce the panels visible green area. If the solar panels are not angled directly at the sun, their image projection in the direction of the sun will decrease and the power produced will decrease accordingly. All of the solar panel models (on the PUP and habitats) are designed to track the sun throughout the lunar day, but the solar panels are only actuated in one degree of freedom, resulting in imperfect tracking, especially when the sun is at high angles above the horizon. For the LER vehicle with attached PUP and solar panel, the solar panel sun tracking motor is locked out during rover driving, resulting is a large degree of variation in sun angle for some drives. The computation technique for computing energy generated by the solar panels provides a fast and accurate measurement of predicted energy generation.

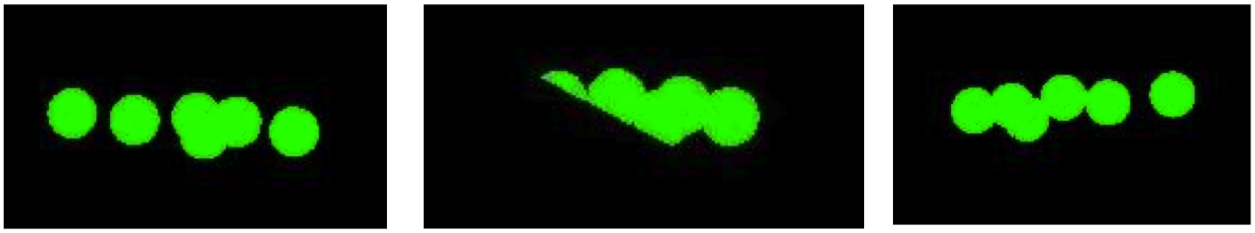


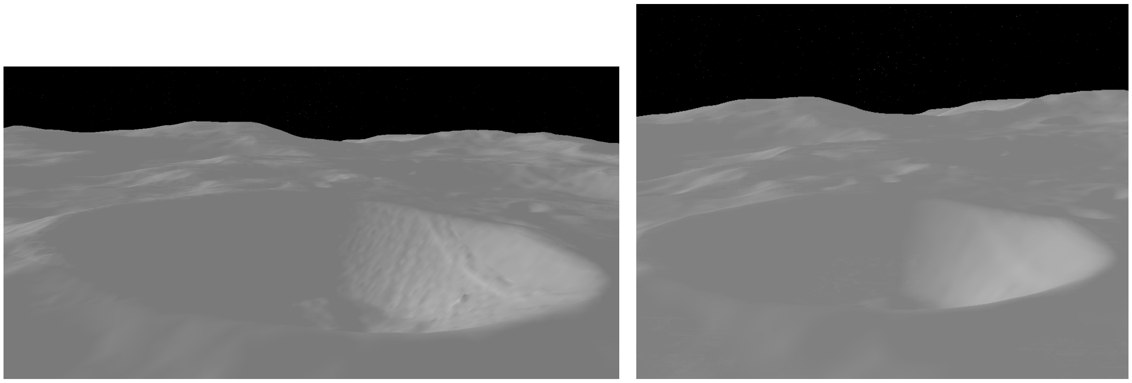
Figure 3: Three views of solar panels as some are partially occluded by terrain and each other.

B. Terrain Models

Two new terrain models were developed recently for LSOS, a Lunar Orbiter Laster Altimeter (LOLA) based terrain of the south pole of the moon and a high-resolution synthetic terrain.

1. LOLA 75 Degree South Pole Lunar Terrain Model

The Lunar Reconnaissance Orbiter (LRO) mission is now orbiting the moon and collecting elevation data of the surface of the moon using the LOLA instrument. Recently, a set of data from LOLA that covered the south pole of the moon was processed by a United States Geological Survey (USGS) team. This data, received from the USGS, was processed to create a terrain store for the south pole of the moon up to 75 degrees south latitude. The area covered is approximately 940 km square centered at the south pole of the moon. Unfortunately, the process to combine the strips of LOLA data currently has some short-comings and the resulting surface shows significant artifacts near the edge of each of the LOLA strips. This creates artificial ridges and peaks that make the LOLA elevation data unusable for our vehicle simulation purposes (see Figure 4a). We therefore did a significant Gaussian smoothing of the raw LOLA elevation data to produce a usable terrain as shown on Figure 4b. The original resolution of the data was 240 meters. After smoothing, the pixel spacing is still 240 meters, but the accuracy is significantly less. The resultant data store requires approximately 240 megabytes of disk storage.



(a) Raw LOLA terrain model with processing artifacts.

(b) Filtered LOLA terrain model.

Figure 4: The LOLA terrain model with a view from Shackleton Crater looking towards Malapert Mountain.

2. Synthetically Augmented South Pole Lunar Terrain

Vehicle simulations in LSOS model the vehicles wheel interaction with the terrain. With a 240 meter resolution terrain model, the simulations have limited validity because the terrain is smoother than it will be in reality. So it is important to use terrains with resolutions that are comparable to the size of the vehicles wheel. We set out to create a high-resolution terrain with 1 meter resolution based on the LOLA 75 Degree south terrain. Using algorithms and software provided by Bob Gaskell,[9] we added craters, rocks, and noise to the 240m LOLA terrain model to construct a realistic 1 meter resolution synthetic terrain model. We used the JPL Supercomputer cluster to execute many hundreds of programs in parallel to create the terrain. The process takes over 100 hours of computing time on a multi-node super computer cluster and produces a dataset of approximately 102 gigabytes broken into 145 tiles. The resulting terrain is 191 km by 136 km and is centered approximately half way between Shackleton Crater and Malapert Mountain, two prominent features close to the South Pole of the moon. An image of the terrain is shown on Figure 5. The terrain covers Shackleton crater, Malapert Mountain, Shoemaker crater, Haworth crater, and de Gerlache crater.

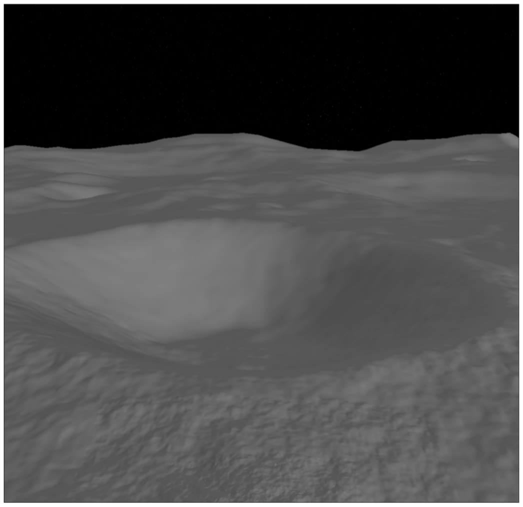
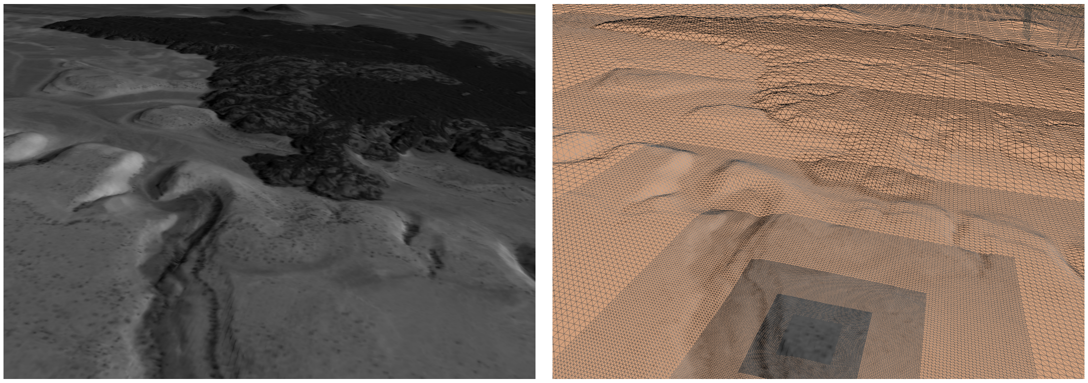


Figure 5: The synthetically augmented high resolution lunar terrain model.

C. 3D Visualization System

The LSOS 3D visualization system, called DspaceOgre, provides accurate real-time 3D visualization of simulations. Built on top of the open-source OGRE 3D graphics engine, DspaceOgre allows LSOS to create graphical objects to represent rovers, terrains, planetary bodies, and skies, among other things and arrange them in a scene graph hierarchy.

The terrain data sets that LSOS deals with are too large to be displayed all at once in full detail with modern graphics cards. To render these large data sets, DspaceOgre uses a level-of-detail technique called clipmapping. This technique allows the highest-resolution mesh to be displayed where it is needed. This high-resolution area is surrounded by rings of gradually lower resolution meshes (see Figure 6). The high-resolution mesh's location on the terrain changes dynamically. For example, the high-resolution mesh can dynamically follow the rover. As the displayed scene changes, each of the vertices are dynamically adjusted to conform to the terrain's digital elevation map. DspaceOgre utilizes the OpenGL Shading Language (GLSL) to implement the clipmapping technique on the Graphics Processing Unit (GPU). This allows for real-time rendering of terrains at greater than 30 frames per second.



(a) Terrain

(b) Wireframe View of Terrain

Figure 6: A rendering of the terrain along with the same terrain rendered in wireframe mode to show the underlying clipmapping levels of detail.

DspaceOgre also uses advanced GPU techniques to support rendering shadows, bump mapping and wheel tracks. Shadows are rendered using shadow mapping, which involves rendering scene from the vantage point of the sun and using the depth values to determine which pixels in the scene are in shadow. Surface normals are perturbed in the fragment shader to do bump mapping, which gives the terrain a synthetic roughness. Surface normals are also modified to give the impression of wheel tracks on the terrain. See Figure 7 for an example of these features.

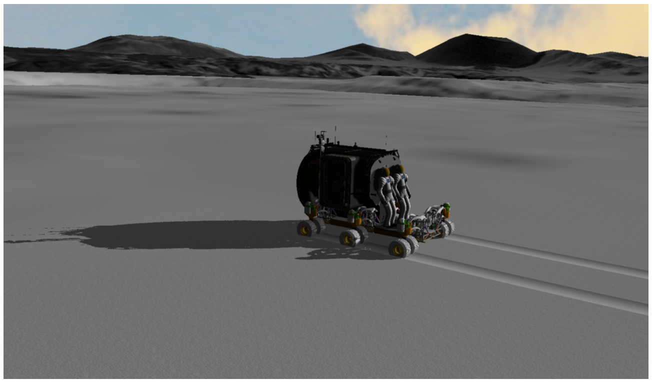
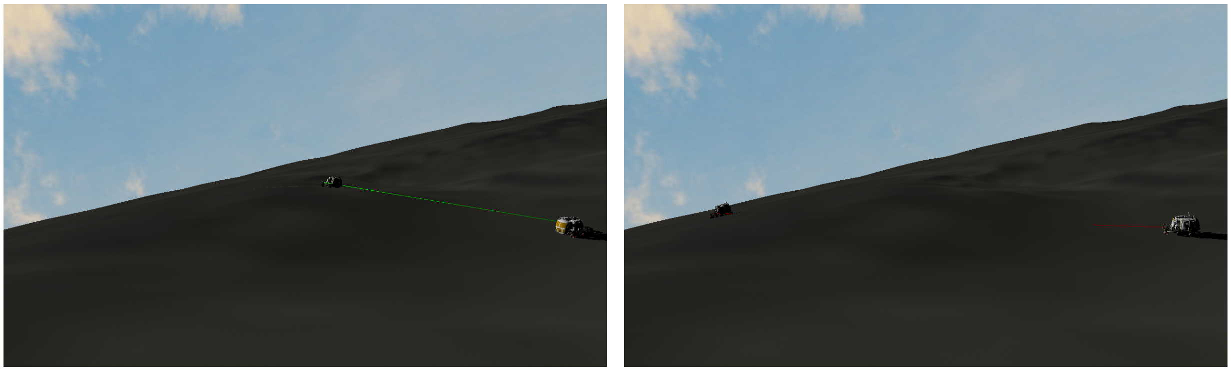


Figure 7: A rendering of a rover that shows DspaceOgre's ability to render wheel tracks, shadows, and terrain bump-mapping.

In addition to being used for visualization, DspaceOgre is also used for GPU-accelerated scene analysis. Results from these analyses can be fed back into the simulation. One analysis capability that DspaceOgre provides LSOS is line-of-sight detection between pairs of objects. This is used by LSOS to determine whether there is line of sight between two vehicles or between a vehicle and a planetary body. Computation of line of sight between objects is done by rendering the scene from the vantage point of one object and scanning the image for existence of pixels that represent the other object. Special graphic objects are used to represent the objects to distinguish them from the rest of the scene. See Figure 8 for an example, where DspaceOgre was used to determine of sight. Green and red color lines were used to give a visual cue of the line-of-sight results. Similar techniques are used for determining solar panel power level (see Sub-section II.A). DspaceOgre counts the percentage of pixels that represent the panel that are visible when rendering from the vantage point of the sun.



(a) Line Of Sight Between Vehicles

(b) No Line Of Sight Between Vehicles

Figure 8: An example where DspaceOgre is used to determine if there is line of sight between vehicles. Green or red lines are drawn to give a visible cue to the user of the line-of-sight results.

LSOS supports rendering on multiple workstations from a single simulation. This is done by creating multiple instances of DspaceOgre and sending messages to the instances via either the Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) communication. This allows for rendering the identical instances of the scene on each workstation. This capability has been used to run LSOS at the NASA Johnson Space Center's immersive dome facility, where there are multiple projectors, each projecting to portions of the dome. Each projector is controlled by a workstation. When running LSOS, an instance of DspaceOgre is run on each workstation with slightly different camera position and orientation settings. The projected images are blended to give the illusion of a large contiguous screen without any visible seams.

D. Path Planning

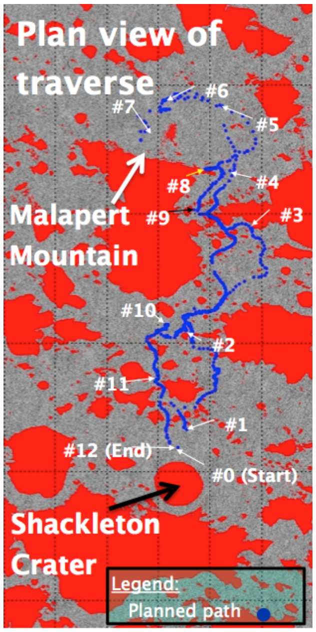
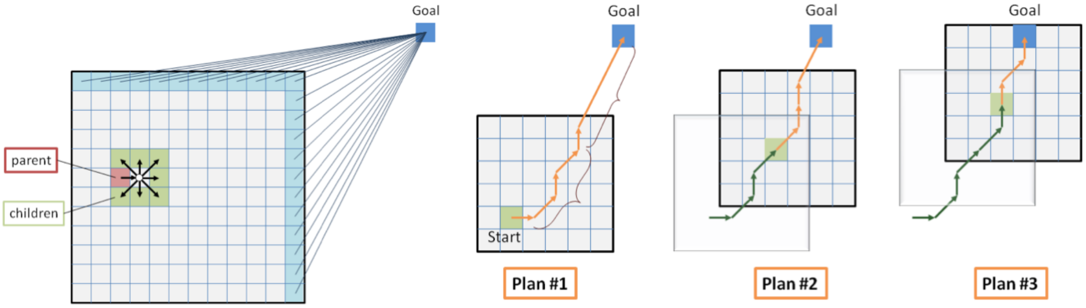


Figure 9: Traverse path from Shackleton Crater to Malapert Mountain and back.

The traversal distance of planned lunar activities can be very long. For example, one proposed mission is a round-trip traverse, from Shackleton Crater to Malapert Mountain while visiting 13 science waypoints, could be as long as 600 km over a lunar day (15 Earth days), The path is shown on Figure 9. Even though lunar vehicles such as LERs and ATHLETEs could be driven by the astronauts or remotely operated, it is tedious to manually find safe paths, especially for long mission on a hazardous terrain full of craters and mountains of various sizes. Given terrain information obtained through remote sensing, the Path Planning component computes safe paths for vehicles involved in such missions. The current planner is based on an A* algorithm applied to a simple 8-connected graph, although it could be extended to a graph that accounts for heading/non-holonomic constraints.[10] The traversal cost of each edge is a Euclidean distance with some penalty on the roughness and slope value evaluated at each terrain pixel (e.g., 1x1m square) obtained by querying the terrain model. Because it is computationally infeasible to construct a graph of 1m resolution over a long 600km course of travel, the path planner iteratively solves subproblems with multiple resolutions: high-resolution grid and straight line approximation. In each subproblem, the graph consists of a high-resolution grid around the start node (e.g., 100x100m region), and each node on the high-resolution grid boundary is connected directly to the goal, as shown in Figure 10a. The cost of traversal from a boundary node to the goal is approximated by an Euclidean distance between them with a hazard penalty (e.g., slope and roughness) evaluated at the boundary node.



(a) Multi-resolution grid.

(b) Sequence of plans towards the goal.

Figure 10: Cells in the square region 10a on a high-resolution grid (8-connected graph in this figure) and its boundary nodes (shaded in cyan) are connected to the goal by a straight line. Three steps in finding a path are shown on figure 10b. The first shows the path found in the first iteration (colored in orange). The next plan in the middle figure starts from a node closer to the goal on the found path. The green line represents the path taken from the previous iteration.

Once a path is found from the start node to a boundary node that is connected to the goal by a straight-line as shown on Figure 10a, the path planner moves the start node towards the goal along the path just found, for the next iteration. As shown in the middle figure of 10b, a portion of the path in the high-resolution grid is taken from the previous plan (represented in green), and the new start node is set at its end. The boundary nodes are also moved with the start node, and the A* search computes a new path. The process is repeated until the goal is reached within the high-resolution grid. Note the start node is not at the boundary node of the previous plan, so Plan 2 does not necessarily follow the path found from Plan 1. Figure 11 shows a snapshot of the vehicles, terrain evaluation (green is feasible, orange is infeasible), and the planned path (shown in yellow) to a goal 5km away on a GSSR lunar terrain data.

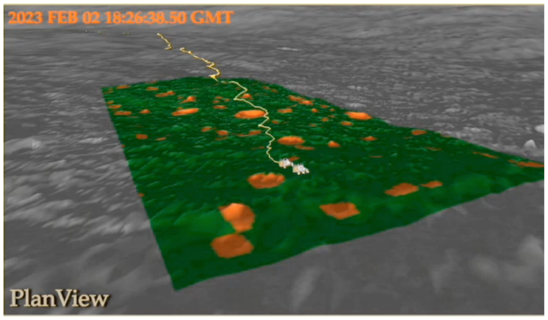


Figure 11: A path from the mission start towards a distant goal. The lunar terrain model is represented in gray; the safe region is in green, and the hazardous region in orange.

E. Scenario Scripting

The LSOS scenario scripting component provides an efficient interface for defining, simulating, and analyzing lunar missions. The simulator is driven by a finite state machine where individual states represent particular behaviors required to simulate such complex scenarios. In this implementation, each mission-level finite state machine creates platform finite state machines that control the behavior of particular platforms (vehicles, assets, and habitats). Each platform finite state machine is defined by a sequence of events that represent traverses, extravehicular activities, waits, pauses, or any other required mode of operation. An illustration of an example mission finite state machine composed of two vehicle models and one habitat model is seen below in Figure 12.

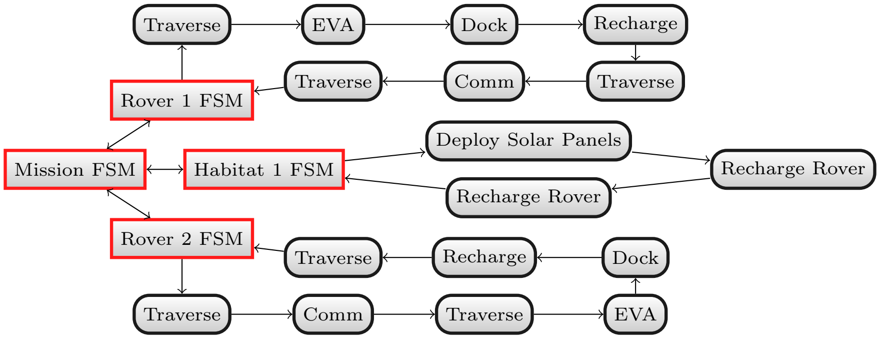
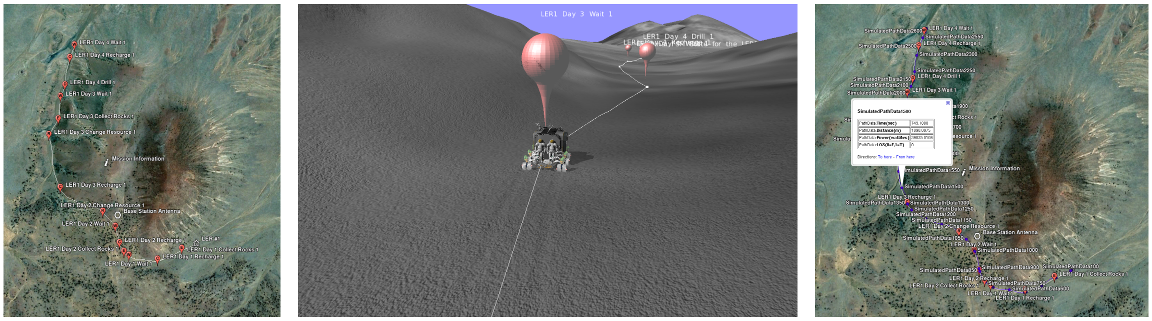


Figure 12: An example of a finite state machine that guides a multi-vehicle mission simulation. In this example, three finite state machines control the behavior of three independent models (two vehicles and one habitat) to simulate a mission.

Custom finite state machines becomes increasingly difficult to implement as mission complexity increases. An alternative approach is to initialize the finite state machine using a markup language. Keyhole Markup Language (KML) in particular is a useful tool for describing such scenarios because geometry information is available, schema definitions can be augmented, and tools exist to visualize and manipulate such data. In order to represent the finite state machine in Figure 12, individual KML placemarks are defined with LSOS-specific schemadata that determines the platform, sequence order, event type, duration, energy consumption, or any other measure of interest. LSOS provides an interface that constructs, simulates, and records the finite state machine described in a KML file. The recorded data is then processed through a script to construct a KML file that overlays the logged data with the original mission plan. An overview of this procedure is illustrated in Figure 13



SimulatedPathData1500

PathData:**Time(sec)**	749.1000
PathData:**Distance(m)**	1090.8975
PathData:**Power(watt/hrs)**	39835.0106
PathData:**LOS(0=F,1=T)**	0

Directions: To here - From here

(a) KML-Based Mission Plan (b) LSOS Rover Simulation (c) KML-Based Analysis

Figure 13: An overview of the mission planning, simulation, and analysis approach for large, complex missions with a KML interface.

F. User Interface

Setting up and using LSOS is typically performed through a graphical user interface (GUI) or through customized scripts written in the Python programming language. For novice users, a simplified GUI was designed with predefined capabilities for the anticipated use of LSOS at other NASA centers. The interface separates the use of LSOS into 2 steps. The first is the configuration of the simulation and the second is the execution of the simulation. The set-up GUI used in the first step provides:

- the selection of a terrain model from a list of models
- setting the start time and simulation time step size to use
- the selection of models from a list of vehicles and habitats to use in the simulation
- the choice of setting other parameters like choice of display of shadows and wheel tracks
- an interactive interface to specify drive paths for vehicles, or to specify a drive path definition file

During the execution of the simulation, the run-time GUI provides:

- simple mouse motions to configure the view
- buttons to start and stop the simulation and to display stripchart of simulation data and documentation on use of the GUI
- a joystick to interactively drive the vehicle
- intuitive display of data, for example, a speedometer for vehicle speed, a battery icon with red and green areas to indicate energy battery charge, etc.

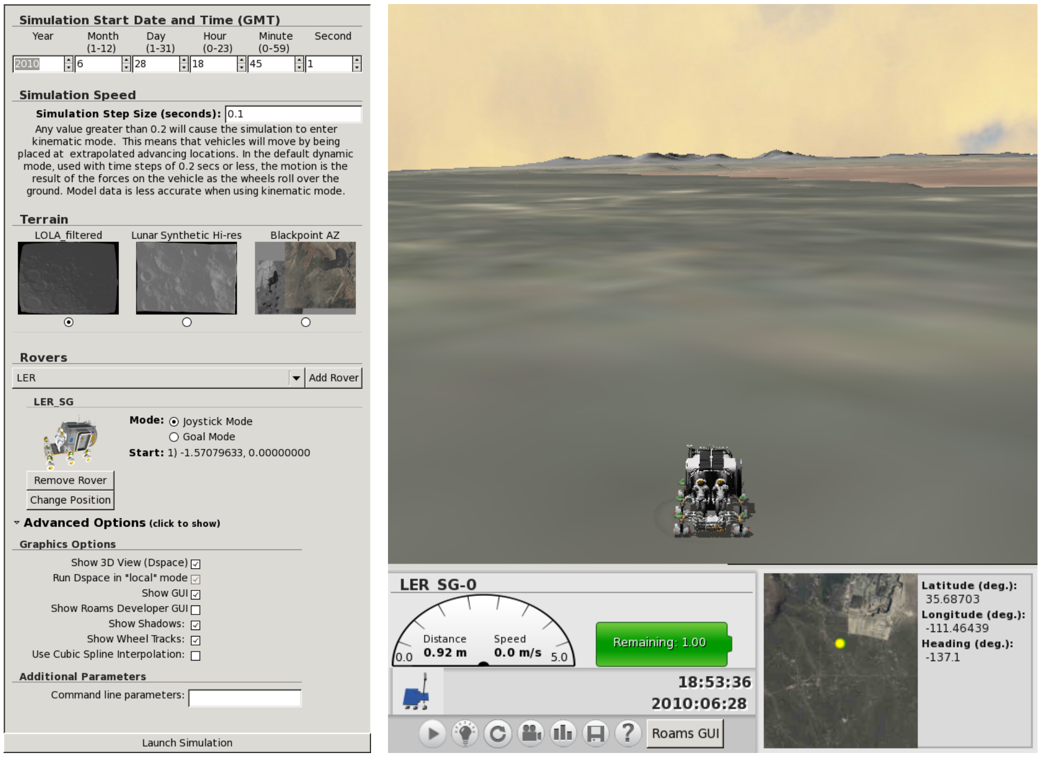
Figures 14a and 14b show the respective GUIs.

III. Analyses and Simulation Studies

A. Shackleton-Malapert-Shackleton Traverse Simulation

A simulation of a Lunar Electric Rover (LER)[11] driving on the moon from Shackleton Crater to Malapert Mountain and back to Shackleton Crater was recently performed with LSOS. The 40-meter resolution Goldstone Solar System Radar (GSSR)[12] lunar terrain model was used for the simulation. The traverse visited 11 locations selected by Weisbin, et. al.[13] for their science value. The path followed by the rover is shown on Figure 9. Areas in red indicate no data value areas on the GSSR terrain model.

The schedule for the 14-day mission was based on a plan developed at the National Administration for Space Astronautics (NASA) Langley Research Center (LaRC). A day in the simulation is a 24-hour period. The mission was designed to occur during the daytime of a day/night cycle on the moon. Unless specified as a Lunar day, the use of the word *day* in the remainder of this paper represents a 24-hour period. With the exception of two days, the mission plan specified a daily schedule with the following sequence of activities:



Simulation Start Date and Time (GMT)

Year	Month (1-12)	Day (1-31)	Hour (0-23)	Minute (0-59)	Second
2010	6	28	18	45	1

Simulation Speed

Simulation Step Size (seconds): 0.1

Any value greater than 0.2 will cause the simulation to enter kinematic mode. This means that vehicles will move by being placed at extrapolated advancing locations. In the default dynamic mode, used with time steps of 0.2 secs or less, the motion is the result of the forces on the vehicle as the wheels roll over the ground. Model data is less accurate when using kinematic mode.

Terrain

● LOLA_filtered ○ Lunar Synthetic Hi-res ○ Blackpoint AZ

Rovers

LER

LER_SG

Mode: ● Joystick Mode ○ Goal Mode

Start: 1) -1.57079633, 0.00000000

Advanced Options (click to show)

Graphics Options

Show 3D View (Dspace) ☑
Run Dspace in "local" mode ☑
Show GUI ☑
Show Roams Developer GUI □
Show Shadows: ☑
Show Wheel Tracks: ☑
Use Cubic Spline Interpolation: □

Additional Parameters

Command line parameters:

LER SG-0

Distance 0.92 m

Speed 0.0 m/s

Remaining: 1.00

18:53:36
2010:06:28

Latitude (deg.): 35.68703
Longitude (deg.): -111.46439
Heading (deg.): -137.1

(a) Set-up GUI

(b) Run-time GUI

Figure 14: The set-up and run-time GUIs used with LSOS.

- traverse to the next science site with the solar panels on the LER locked in an optimal configuration. The vehicle was commanded to drive at 8 kilometers/hour during this traverse phase.
- upon arrival at the science site, place the solar panel power system on the ground and set it to servo continuously to point to the sun.
- explore the science site for a specified time (modeled as driving at 2 kilometers/hour in a circular path).
- place the solar panel power system back on the LER.
- rest (no driving but the solar panel continues to servo to point to the sun) until the start of the next day.

These activities were repeated for 12 days of the 14-day simulation. On days 7 and 10 of the mission, the LER did not perform the first item on its activity list – drive to the next science location. Instead, it explored the current science site. A simulation scenario was constructed to follow the specified schedule. The the simulation was set to begin at 10:00 am on Nov 20, 2023 and ran for fourteen 24-hour days. During the simulation, the sun position was calculated using the SPICE toolkit.[14] Solar panel Illumination was accurately modeled through a solar panel model which takes into account the panel's line of sight to the sun (occlusion by the terrain features is accurately captured by the model).

The following data were collected during the simulation:

- Energy generated from solar panels
- Energy used for driving and on-board vehicle operations
- Location of the vehicle on the lunar surface

The power data shown below in Tables 1 and 2 from this simulation was of most interest to the NASA Langley Research Center(LaRC) mission planners. Units of time are in hours, distance in kilometers and energy in kilo-Watt-hours.

Table 1: Energy data for Days 1-7 of Shackleton to Malapert traverse.

Day	1		2		3		4		5		6		7	
Time														
Traverse	0.74		7.47		6.39		4.26		2.69		2.05		0.00	
Explore	3.50		3.50		3.50		4.25		5.25		5.00		5.75	
Parked	19.26		12.53		13.61		14.99		15.56		15.95		17.25	
Distance														
Traverse	7.72		77.53		66.57		44.38		28.18		21.33		0.00	
Explore	6.59		6.65		6.60		8.14		9.93		9.73		11.19	
	LER	PUP	LER	PUP	LER	PUP	LER	PUP	LER	PUP	LER	PUP	LER	PUP
Energy														
Start Day	90.00	100.00	90.00	100.00	90.00	100.00	90.00	44.32	90.00	68.87	90.00	99.60	90.00	100.00
Traverse														
Consumed	-2.62	-0.15	-34.68	-1.49	-25.43	-1.28	-19.68	-0.85	-9.87	-0.54	-9.47	-0.41	0.00	0.00
from PUP	2.62	-2.67	34.68	-35.39	25.43	-25.95	19.68	-20.08	9.87	-10.07	9.47	-9.66	0.00	0.00
Recharge	0.00	2.82	0.00	12.17	0.00	8.02	0.00	6.33	0.00	6.08	0.00	4.17	0.00	0.00
End Traverse	90.00	100.00	90.00	75.29	90.00	80.80	90.00	29.72	90.00	64.35	90.00	93.70	90.00	100.00
Exploration														
Consumption	-7.95	-0.70	-8.21	-0.70	-7.90	-0.70	-10.29	-0.85	-12.25	-1.05	-12.68	-1.00	-14.58	-1.15
from PUP	0.00	0.70	0.00	15.37	0.00	0.00	0.00	18.04	0.00	13.59	0.00	7.30	0.00	1.15
End explore	82.05	100.00	81.79	89.96	82.10	80.10	79.71	46.90	77.75	76.89	77.32	100.00	75.42	100.00
Parked														
Consumption	-34.67	-3.85	-22.55	-2.51	-24.50	-2.72	-26.98	-3.00	-28.01	-3.11	-28.71	-3.19	-31.05	-3.45
from PUP	42.61	-43.48	30.76	-31.39	32.40	-33.06	37.27	-38.03	40.26	-41.08	41.39	-42.23	45.63	-46.56
Recharge	0.00	47.34	0.00	43.94	0.00	0.00	0.00	63.00	0.00	66.91	0.00	45.42	0.00	50.01
End Day	90.00	100.00	90.00	100.00	90.00	44.32	90.00	68.87	90.00	99.60	90.00	100.00	90.00	100.00

Table 2: Energy data for Days 8-14 of Shackleton to Malapert traverse.

Day	8		9		10		11		12		13		14	
Time														
Traverse	2.31		10.07		0.00		2.89		9.51		3.24		3.02	
Exploration	5.25		6.00		5.25		3.50		3.50		3.50		0.00	
Parked	15.94		7.68		18.50		17.11		10.49		16.76		20.98	
Distance														
Traverse	23.85		104.94		0.00		29.96		99.09		33.66		31.52	
Exploration	10.06		11.57		10.13		6.57		6.73		6.78		0.00	
	LER	PUP	LER	PUP	LER	PUP	LER	PUP	LER	PUP	LER	PUP	LER	PUP
Energy														
Start Day	90.00	100.00	90.00	100.00	90.00	82.11	90.00	29.90	53.93	10.00	40.00	10.78	40.00	37.33
Traverse														
Consumption	-17.52	-0.46	-49.34	-2.01	0.00	0.00	-13.75	-0.58	-42.91	-1.90	-17.89	-0.65	-14.44	-0.60
from PUP	17.52	-17.88	49.34	-50.35	0.00	0.00	13.75	-14.03	10.00	-10.20	0.00	0.00	0.00	0.00
Recharge	0.00	1.57	0.00	11.20	0.00	0.00	0.00	1.45	0.00	6.69	0.00	6.53	0.00	3.56
End traverse	90.00	83.24	90.00	58.83	90.00	82.11	90.00	16.74	21.02	4.58	22.11	16.67	25.56	40.29
Exploration														
Consumption	-13.77	-1.05	-15.10	-1.20	-13.21	-1.05	-7.83	-0.70	-9.19	-0.70	-8.84	-0.70	0.00	0.00
Recharge	0.00	17.81	0.00	26.18	0.00	0.00	0.00	0.00	0.00	14.21	0.00	14.78	0.00	0.00
End explore	76.23	100.00	74.90	83.82	76.79	81.06	82.17	16.04	11.83	18.09	13.26	30.75	25.56	40.29
Parked														
Consumption	-28.69	-3.19	-13.82	-1.54	-33.30	-3.70	-30.81	-3.42	-18.89	-2.10	-30.17	-3.35	-37.76	-4.20
From PUP	42.46	-43.33	28.92	-29.51	46.51	-47.46	2.57	-2.62	47.06	-48.02	56.91	-58.07	52.20	-53.27
Recharge	0.00	46.52	0.00	29.33	0.00	0.00	0.00	0.00	0.00	42.81	0.00	68.00	0.00	27.73
End Day	90.00	100.00	90.00	82.11	90.00	29.90	53.93	10.00	40.00	10.78	40.00	37.33	40.00	10.56

The results from the simulation showed that the energy available during the mission dipped to dangerously low levels during days 11, 12, 13 and 14. The start of the mission was chosen to coincide with the dawn of the Lunar day. The outgoing path to Malapert from Shackleton followed an easterly route that kept the vehicle in mostly illuminated regions during the early parts of the Lunar day. As the Lunar day progressed and the Sun moved to the West. Returning to Shackleton Crater via the Eastern route left the vehicle in shadowed regions, severely reducing the amount of energy generated by the solar panels.

B. Solar Panel Power Generation at Shackleton and at Malapert

A comparison of different solar panel control strategies was performed with LSOS to support the design of a new version of the LER[11] cabin. The objective of the analysis was to help determine a desirable battery capacity for the new LER cabin design. The results from the simulation analysis were used as inputs to the analysis performed by the LER cabin design team at the Johnson Space Center. The simulations in LSOS studied the power generation capability of the solar panels on the LER and the Portable Utility Palette (PUP) carried by the LER. The PUP is a deployable asset that can be placed on the ground to collect solar energy while the LER explores a site. While deployed, the PUP can servo its solar panels to maximize solar energy collected by its solar panel and saved to its battery. The study evaluated performance at different locations on the moon under two different solar panel control strategies. The first solar panel control strategy was to optimally point the PUP solar panels at the sun at the start of a traverse and lock them before commencing the traverse. The second was to continuously re-orient the PUP solar panel mimicking its configuration when deployed on the ground. For all simulations, the LER solar panel was modeled as rigidly attached to the vehicle. The simulations were performed at two sites: Shackleton Crater close to the South Pole of the moon and at Malapert Mountain at a latitude of about 85 degrees South.

The following data were logged at 10 minute intervals:

- Time
- Sun position (azimuth and elevation)
- LER solar panel illumination
- PUP solar panel illumination

Energy generation data collected from simulations run for each site were for the following conditions:

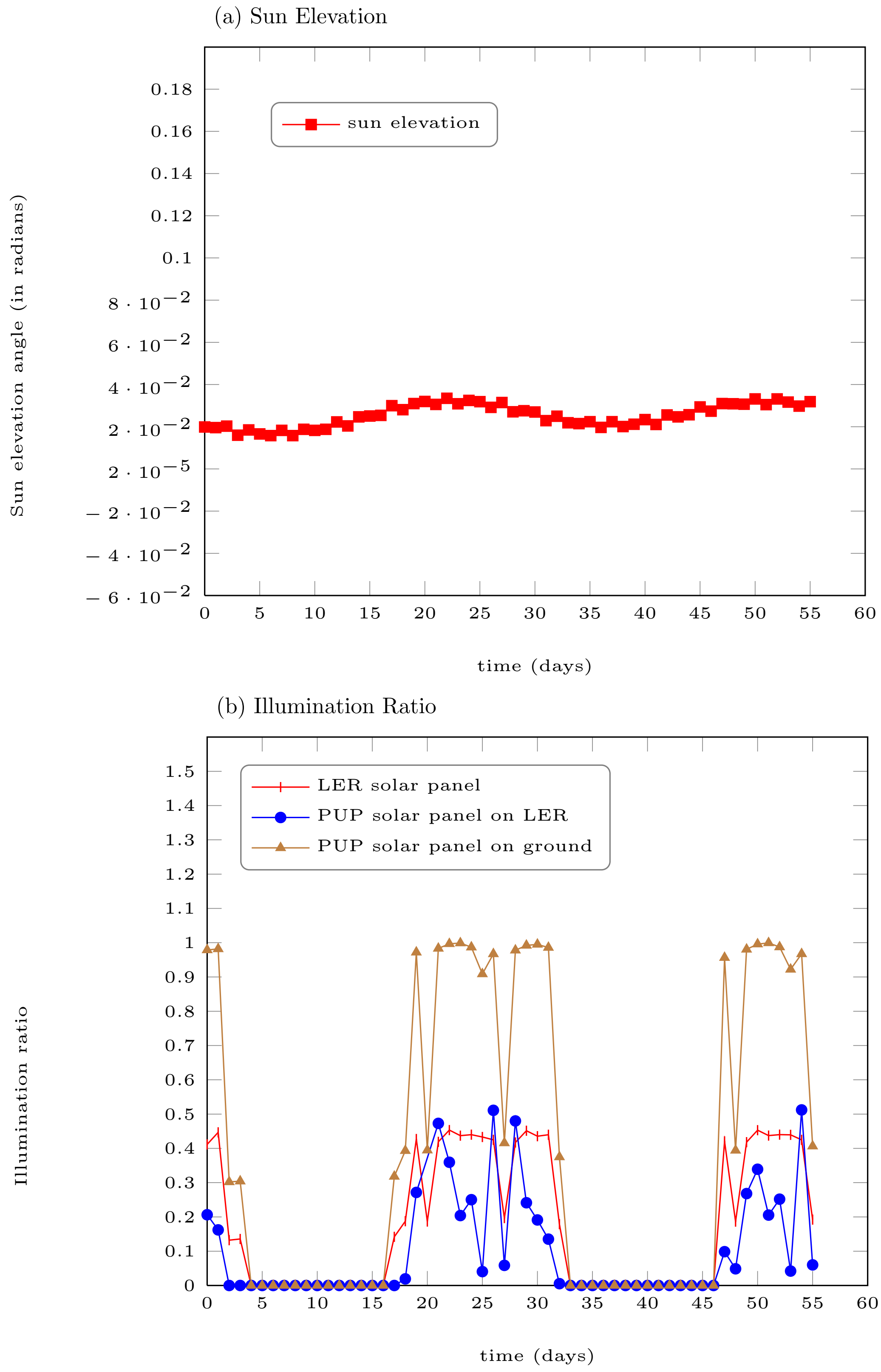
- PUP solar panel always pointing to the sun simulating a parked PUP with actively controlled solar panel.
- PUP solar panel pointed to the sun once and only at the start of each day and a rigidly mounted solar panel on the LER.
- LER solar panel rigidly attached to the vehicle.

Each simulation ran for 56 days covering about two Lunar day/night cycles. During the simulations, the sun position was positioned from data obtained from the SPICE toolkit.[14] The 40-meter resolution GSSR lunar terrain model was used for these simulations. PUP solar panel Illumination was accurately modeled through a solar panel model which takes into account the panel's line of sight to the sun (occlusion by the terrain features are accurately captured by the model). We did not have a solar panel model for the LER so the LER Illumination was estimated by taking the cosine of the angle between the vector from the rover to the current sun position and the vector from the rover to the sun position at the start of the day.

The data were stored in the HDF5 Table format (http://hdfgroup.org) which allows the data to be easily retrieved and processed. Scripts were written to display the sun elevation, PUP illumination and LER illumination as a strip chart as well as saving the data to a comma delimited file for importing to spreadsheets.

The Shackleton Crater site was about 10 km North of the South Pole of the moon. The traverse was to drive in a 20km square loop with 5km sides. Shackleton Crater is an elevated terrain feature to the south of the site. The Malapert Mountain site was on the northern plain of Malapert about 10 km north of the Malapert peak. The same traverse, a drive in a 20km square loop with 5km sides, was performed at this site. The simulations were all begun at 10:00 am on Dec 4, 2023. Data from the simulations are shown on Figure 15 for the simulations at the Shackleton Crater site and Figure 16 for the Malapert Mountain site.

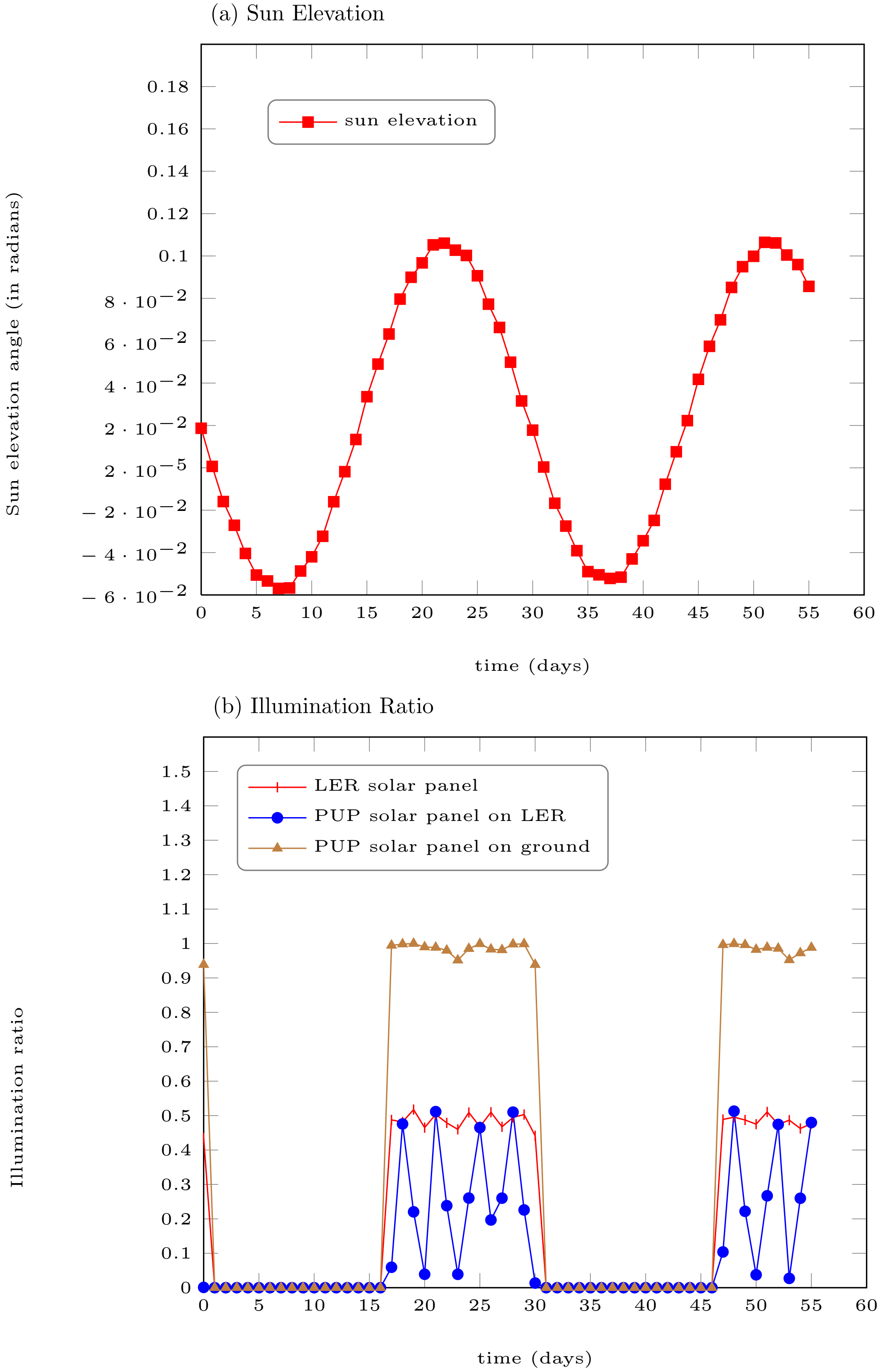
At sites close to the poles, the sun is always low on the horizon. At Shackleton Crater at the South Pole of the moon, the highest elevation of the sun is about 0.034 radians (about 2 degrees). At Malapert Mountain at a latitude of 85 degrees South, the highest elevation of the sun is about 0.106 radians (about 6 degrees). It is not surprising then that the results showed that the terrain greatly influences the solar panel illumination - more so than the lunar day cycle. There were significant terrain features near these sites (Shackleton and other crater rims at the polar site and Malapert peak near the Malapert site) that obscure illumination by the sun.



(a) Sun Elevation

(b) Illumination Ratio

Figure 15: Sun elevation and solar panel illumination at Shackleton Crater.


(a) Sun Elevation

(b) Illumination Ratio

Figure 16: Sun elevation and solar panel illumination at Malapert Mountain.

IV. Conclusion

Recent developments in LSOS have been described in this paper. New surface system elements and terrain models, and procedures for use in simulating mission scenarios have been developed. The results from the analyses and simulations have been fed back to teams at other NASA centers to assist in the design and planning of missions. LSOS has demonstrated that high-fidelity dynamics simulation is a powerful tool to assist during the concept development and planning stages of space missions.

Work on the development of LSOS continues. The general approach pursued in the development of the family of dynamics simulators at JPL can be easily adapted to simulate a variety of space applications including missions to near-Earth objects, and landing and surface operations on planetary and other space bodies. As NASA considers new mission objectives and develops plans using new designs of transport systems to new space environments, our simulation capability is well-positioned to adapt to and address NASA's new requirements. In preparation for this change, LSOS is transitioning to a more general project and software package named Surface Exploration Operations Simulator (SEOS).

V. Acknowledgments

This work was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. We thank our sponsor, Doug Craig from NASA Headquarters, Bob Gaskell at the Planetary Science Institute and our collaborators at the United States Geological Survey (USGS), NASA Glen Research Center, NASA Langley Research Center, NASA Johnson Space Center, NASA Ames Research Center, NASA Marshall Space Flight Center, and Jet Propulsion Laboratory for support in the development of LSOS.

References

[1] Nayar, H., Balaram, J., Cameron, J., Jain, A., Lim, C., Mukherjee, R., Peters, S., Pomerantz, M., Reder, L., Shakkottai, P., and P., S. W., "A Lunar Surface Operations Simulator," *Proceedings of the Simulation, Modeling, and Programming for Autonomous Robots 2008 (SIMPAR 2008)*, Venice, Italy, November 2008, pp. 65–74.

[2] Nayar, H., Balaram, A. J. J., Cameron, J., Lim, C., Mukherjee, R., Pomerantz, M., Reder, L., Myint, S., Serrano, N., and Wall, S., "Recent Developments on a Simulator for Lunar Surface Operations," *Proceedings of the AIAA SpaceOps Conference 2009*, AIAA, Pasadena, CA, September 2009.

[3] Jain, A. and Man, G., "Real-time simulation of the Cassini spacecraft using DARTS: functional capabilities and the spatial algebra algorithm," *5th Annual Conference on Aerospace Computational Control*, Pasadena, CA, August 1992.

[4] Biesiadecki, J., Henriquez, D., and Jain, A., "A Reusable, Real-Time Spacecraft Dynamics Simulator," *6th Digital Avionics Systems Conference*, Irvine, CA, October 1997.

[5] Jain, A., Guineau, J., Lim, C., Lincoln, W., Pomerantz, M., Sohl, G., and Steele, R., "ROAMS: Planetary Surface Rover Simulation Environment," *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, Nara, Japan, May 2003.

[6] Jain, A., Balaram, J., Cameron, J., Guineau, J., Lim, C., Pomerantz, M., and Sohl, G., "Recent Developments in the ROAMS Planetary Rover Simulation Environment," *IEEE Aerospace Conference*, Big Sky, Montana, March 2004.

[7] Balaram, J., Austin, R., Banerjee, P., Bentley, T., Henriquez, D., Martin, B., McMahon, E., and Sohl, G., "DSENDS - A High-Fidelity Dynamics and Spacecraft Simulator for Entry, Descent and Surface Landing," *IEEE Aerospace Conference*, Big Sky, Montana, March 2002.

[8] Jain, A., Cameron, J., Lim, C., and Guineau, J., "SimScape Terrain Modeling Toolkit," *Second International Conference on Space Mission Challenges for Information Technology (SMC-IT)*, Pasadena, CA, July 2006.

[9] Gaskell, R., Husman, L. E., Collier, J. B., and Chen, R. L., "Synthetic Environments for Simulated Missions," *IEEE Aerospace Conference*, Big Sky, Montana, 2001.

[10] Pivtoraiko, M., Knepper, R. A., and Kelly, A., "Differentially constrained mobile robot motion planning in state lattices," *Journal of Field Robotics*, Vol. 26, No. 3, 2008.

[11] Ambrose, R., "Human-Robotics Interactions: Field Test Experiences from a collaborative ARC, JPL and JSC Team," *AIAA NASA 3rd Space Exploration Conference*, Denver, CO, February 2008.

[12] Cooke, D., Yoder, G., Coleman, S., and Hensley, S., "Lunar Architecture Update," *AIAA NASA 3rd Space Exploration Conference*, Denver, Colorado, February 2008.

[13] Weisbin, C. R., Mrozinski, J., Lincoln, W., Elfes, A., Shelton, K., Hua, H., Smith, J. H., Adumitroaie, V., and Silberg, R., "Lunar Architecture and Technology Analysis Driven by Lunar Science Scenarios," *Systems Engineering Journal*, Vol. 13, No. 3, 2010.

[14] NAIF, "SPICE Toolkit," http://naif.jpl.nasa.gov, 2010.