

# FORMULATION OF MICRO-ROVER AUTONOMY SOFTWARE FOR LUNAR EXPLORATION

Virtual Conference 19–23 October 2020

Varsha Kumar<sup>1,2</sup>, Shyam S. Sai<sup>1,2</sup>, Srinivas Vijayarangan<sup>1,2</sup>, David Wettergreen<sup>1,2</sup>, Heather Jones<sup>1,2</sup>, Patrick Callaghan<sup>2</sup>, Haidar Jamal<sup>1,2</sup> and William L. Whittaker<sup>1,2</sup>

<sup>1</sup>Carnegie Mellon University, Pittsburgh, PA, 15213, USA, E-mails: varshak@andrew.cmu.edu, sssai@andrew.cmu.edu, svijaya1@andrew.cmu.edu, dsw@cmu.edu, hlj@cs.cmu.edu, hjamal@andrew.cmu.edu, red@cmu.edu

<sup>2</sup>MoonRanger Project, Pittsburgh, PA, 15213, USA, E-mail: pjc8dj@virginia.edu

## ABSTRACT

Micro-rovers offer immense advantages of low mass, low cost and frequent flight opportunities. Due to the constraint of low mass micro-rovers of our time cannot be isotope-heated. Therefore, they cannot survive the extended planetary nights, so they must achieve their exploration goals in a single daylight period. Their small size, mass and power precludes a radio for direct communication with Earth. For this reason, they can only receive and relay data while in proximity to their lander, and hence, they cannot be constantly supervised or teleoperated from Earth like larger rovers with greater power and communication capability. In order to explore beyond lander communication range, micro-rovers must operate autonomously. Micro-rover autonomy software must achieve communication-denied, high-cadence, kilometer-scale exploration treks. This paper formulates a software architecture and component-wise design for achieving the required autonomous micro-rover exploration.

This technology will be integral to the MoonRanger micro-rover, which will fly to the lunar pole in December 2022 as a Lunar Surface Instrument and Technology Payload (LSITP) aboard the Masten XL-1 lander. MoonRanger will conduct long treks from and to the lander to explore for lunar polar ice. The software will incorporate perception, planning, navigation, and execution, log data. Upon return to its lander, it will transfer data, images and scientific information that are result from mission-relevant autonomy. It will do so at a leap of performance beyond that achieved in prior planetary roving, but with the power, sensing and size constraints of micro-roving.

MoonRanger hosts a two-computer system consisting of a space-hardened embedded processor and a higher-performance, less-hardened computer. Autonomy software and image processing run on a Linux-based OS on the higher-performance comput-

er, while motor control, sensor data collection, and low-level functionality run on a real-time OS aboard the embedded processor. A design prototype of the higher-performance computer's software is depicted in Figure 1. As shown in the figure, this software is organized into two categories, the navigation pipeline and the execution nodes. The navigation pipeline performs perception, rover pose estimation and planning, while the execution nodes handle executive control, data management and transfer, health monitoring and telemetry management.

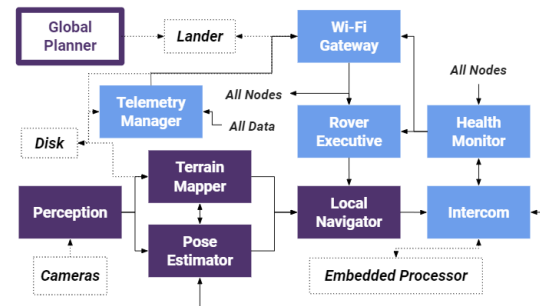


Figure 1: Software Architecture. The navigation pipeline is depicted in violet. Execution nodes are depicted in blue. The Global Planner will run on ground software. The lander, cameras, embedded processor, and disk are external to the design.

## 1 BACKGROUND

Despite their heavy reliance on teleoperation, rovers of the past have been precursors to micro-rover autonomy. The Sojourner micro-rover pioneered line striping for short-range autonomy [1], evolved by MoonRanger for perception in darkness for kilometer-scale polar lunar missions [2][3]. The Mars Exploration Rovers demonstrated the viability of visual odometry for safeguarding and navigation [4], which is vastly improved algorithmically and by processing power by MoonRanger. The Perseverance rover aligns with the ambitions of micro-rover autonomy software, although Perseverance autonomy will

operate at a more tempered pace of about 200 meters per day [5], versus MoonRanger's ambition for a kilometer per day. When measured by body length per day, the range comparison is even greater. Perseverance will be monitored and commanded, with a delay of 5 to 20 minutes [6] making it somewhat isolated from Earth. This isolation is extreme for micro-rovers, whose autonomy software must execute kilometer-scale treks for days in complete lack of any monitoring from Earth. As such, the detailing of design is critical in ensuring mission success.

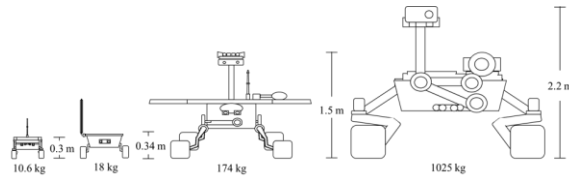


Figure 2: Rover size comparison. L to R: Sojourner [7][8], MoonRanger, Spirit/Opportunity (Mars Exploration Rover)[9][10][11], Perseverance [6]

## 2 SOFTWARE REQUIREMENTS PER LOCATION

Conducting a micro-rover mission requires software in three locations: the lander, the ground control station on Earth, and the rover. The lander hosts software responsible for relaying commands from the ground control station to the rover and transferring data from the rover to the ground control station. The ground control station software is for analyzing data, planning actions and commanding the rover. The rover software conducts autonomous treks as directed by ground control and, upon return to lander communication, sends data to the lander to forward to Earth. The unique reliance by micro-rovers on their landers for communication with the Earth compels careful design of interactions between the lander, ground control and rover for successful autonomy.

### 2.1 Lander

Micro-rovers of this class fly on commercial landers. These landers provide certain information and capabilities. Soon after landing on the lunar surface, the lander provides data from which the lander position on the Moon can be calculated. Knowledge of lander position is critical for conducting global planning and creating a Global Map for the rover. Another lander-provided capability is the transmission of rover data products to Earth. The lander stores and then later forwards rover data products to the ground control station at a certain rate per day.

The ability of a lander to store and later forward rover data products is critical for efficient usage of

the short mission duration, eight Earth days. Because of inherent limitations on bandwidth available between the Moon and the Earth, the current generation of micro-rovers generate data in amounts that vastly exceed the amount of data that can be transmitted to Earth over mission duration. The large Wi-Fi bandwidth between the rover and the lander allows all of the data generated over the course of a single trek to be transmitted from the rover to the lander within a few short hours, whereas transmission of that data to the Earth would take a few days. To allow the rover to continue exploring while data products are transmitted to Earth, landers provide a store-and-forward functionality. The store-and-forward functionality allows the rover to send data products to the lander for storage as soon as a communication link is established. The lander can then forward the data products to the ground control station independent of a communication link to the rover. As such, the rover can begin another autonomous trek while the lander downlinks rover data products to ground control.

Data products are sent to the lander and to Earth in a static priority order. Roughly speaking, the priorities are: current rover telemetry, scientific data, log of past rover telemetry, terrain models, compressed images, and raw images. Current telemetry is sent first to alert ground of rover status. Scientific data return is the mission purpose and, so it is sent second. The log of past telemetry is sent third to aid in debugging the rover. Terrain models are sent fourth, as they transmit quickly and are informative of lunar topology. Compressed images from an entire trek are sent last. Raw images are sent upon request, as they cannot all be downlinked to Earth within mission time. Ground control data requests pause store-and-forward transmission until explicit continuation.

### 2.2 Ground Control Station

Communication from the ground control station to the rover is sent to the Moon through a separate lander control station. This communication includes autonomy commands, which provide the rover with waypoints to guide the rover to a target, and teleoperation commands, which provide drive arcs for the rover to execute. Additionally, the ground control station can query the rover for a detailed status update, uplink Global Map data to the rover, and request specific data products from the data products stored on the lander. For communication with the rover, the ground control station must indicate to which of the two rover computers the message is intended, so the message is received accordingly.

### 2.3 Rover

Rover software runs on two computers: a space-hardened embedded processor and a higher-performance computer. Because the higher-performance computer has the power to process high-resolution images, generate and analyze point clouds and plan safe terrain navigation, its software hosts the majority of the autonomy software. Though processing power available on the higher-performance computer is leagues beyond rover processing power to date [6] and the computer's Graphics Processing Unit (GPU) is a rich resource for parallelizing image processing, it is important recognize that the quad-core, 8GB RAM computer [12] [13] is, in many aspects, still less powerful than a personal laptop. At the same time, the embedded processor is much more drastically limited computationally and hence is not responsible for much autonomy.

The embedded processor software runs on a real-time operating system (RTOS). Its primary functions are reading from peripherals and executing drive commands. Data from any onboard scientific instruments is read by and stored on the embedded processor. The embedded processor is responsible for the eventual transmission of this scientific data to the lander. Other high-rate sensors read by the embedded processor include an inertial measurement unit, encoders, a sun sensor, and temperature sensors. A simple, dead-reckoning pose estimator runs on the embedded processor, as does proprioceptive (vision-denied) safeguarding. In case of higher-performance computer failure partway through an autonomous trek, the dead-reckoning pose estimator and proprioceptive safeguarding are core to survival autonomy that would return the rover to lander communication range. Outside this failure mode, these components provide high-rate rover status monitoring during nominal roving, responding instantaneously to immediate danger, such as imminent tip-over. Nominal roving additionally relies on the embedded processor to command motors to execute drive arcs. These drive arcs may be sent by the higher-performance computer's navigation software or by the ground control station as teleoperation commands.

Both the embedded processor and the higher-performance computer are critical to micro-rover autonomy. However, the higher-performance computer is more heavily involved in autonomous navigation. As such, the rest of this paper focuses on the software components running on the higher-performance computer, detailing design component-wise. Prototypes of many software components exist in a Robot Operating System (ROS) environment. The inputs and outputs of each software component on the higher-performance computer is summarized

in the Appendix. Of these, the autonomy-centric components comprise the navigation pipeline, whereas the executive control and data management software components comprise the execution nodes.

### 3 NAVIGATION PIPELINE

The navigation pipeline includes perception, global planning, local planning, terrain mapping and pose estimation.

#### 3.1 Perception

Perception encompasses image capture through point cloud calculation, which may rely on stereo vision or line striping [2].

Given that this class of micro-rover has a maximum mechanical speed of 7 cm/sec [12], images are captured at a rate of 3-6 Hz, anticipating a 3 cm maximum displacement between frames. A micro-rover's stereo baseline is determined by the mechanical design of the rover and the desired placement of the location at which the stereo camera field of views (FOVs) begin to overlap. For MoonRanger, the field of view of each camera is approximately 90 degrees horizontally by 60 degrees vertically, the most reasonable from available camera options fitting avionic constraints. Thermal advantages drive the placement of cameras as recessed in the rover body at a height of approximately 0.25 meters.

The variable parameters of stereo baseline, vergence angle and camera tilt must be tailored for obstacle avoidance. The perception system should be able to generate point clouds covering at least rover width from terrain as close as half of the rover length, or 0.325 m, in front of the rover. An initial estimate of viable values for baseline, tilt angle, and vergence can occur through calculation. To simplify the initial search space, vergence is kept at 0 and the camera center is projected to the ground at the 0.325m distance. The baseline (**b**) and tilt angle (**t**) are determined by where camera field of views must overlap.

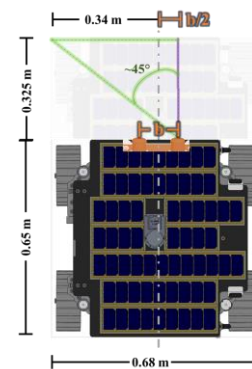


Figure 3: Top view of camera constraints, with fore cameras depicted in orange via a cutout view.

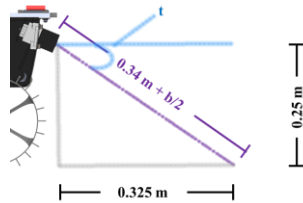


Figure 4: Side view of camera constraints. The dimension in violet is determined via simple trigonometry on the green and violet triangle in Figure #.

Given the constraints in Figures # and #, one can determine via trigonometry that a viable stereo baseline (**b**) is 0.14 m with a camera tilt angle (**t**) of approximately 35.5 degrees. Giving ten percent margin by bloating the width of the rover in the calculations from 0.68 m to 0.75 m yields another viable stereo baseline (**b**) of 0.07 m at the same tilt angle. Empowered by these calculations, field work to test navigation pipeline components can continue in parallel to experimental determination of optimal camera setup.

One such field-tested component was created to address the critical perceptive challenges of low-angle lighting and truly dark shadows at the lunar pole. In scenarios in which the rover cannot rely on stereo image processing to generate point cloud data due to lack of light, an alternative solution is necessary. One alternative solution is laser line striping, by which a set of laser planes is projected from the top of the rover solar panel onto the ground, creating bright lines on the ground that hug the terrain contour. Ribbons of points can be generated from these “laser stripes” and then aggregated via push-broom modeling to generate sufficient data for obstacle avoidance capability. An early pioneer of such obstacle avoidance, though teleoperated, was the Sojourner rover in 1996 [1]. Jamal, et. al. provides detail on generating ribbons of point clouds from laser line stripes for this class of micro-rovers [2].

### 3.2 Global Planner

The Global Planner creates an optimal trek between the current rover pose and a desired goal. The trek is represented as a series of waypoints from the current pose to the goal. The rover will navigate from waypoint to waypoint to complete the trek and return via reversing the order of the waypoints [14].

Global planning occurs on approximately 2 km x 2 km Global Maps of the region around the lander. Initial formulation utilizes graph-based planning. Global planning occurs at the ground control station, and the trek waypoints are sent up to the micro-rover. Goals are determined based on analysis of ice stability maps, and Global Maps of the lunar surface are

generated from digital elevation models, illumination maps, slope maps, and simulations run by the team [15]. A form of the Global Maps is sent to a rover at the beginning of a mission as a coarse terrain map on which to add terrain details observed by the rover.

A trek is optimized over rover physical limits, known obstacles, and sun angle. Physical limits include limits on slope angle to prevent rover tip-over. Accommodating for known obstacles involves planning a trek around large positive and negative obstacles, such as large boulders or craters. For lunar polar, missions sun angle must be incorporated into global planning. Because of the low sun angle, a rover’s ability to convert solar energy to electrical power depends heavily on the angle between sunlight and the vertical plane of the rover’s solar panel. The angle for maximum solar energy capture is 90 degrees.

The current solution for return to lander communication range is for the rover to reverse along the same path by which it travelled to a goal. At mid-latitudes, reversing causes little energetics concern relative to the sun angle because sunlight comes from overhead, not from the side. At the lunar pole this reversal also energetically feasible, despite the dependence of the of solar power capture on the angle between a rover’s solar panel and the sun. This is because, even on the longest treks travelled by these micro-rovers, one kilometer out, the sun angle will only shift by approximately 24 degrees [15]. As such, the lowest possible efficiency of solar power capture is 90 percent [16].

### 3.3 Local Navigator

The Local Navigator is responsible for taking the micro-rover from waypoint to waypoint along a trek created by global planning. After analyzing obstacle data generated by terrain mapping, the Local Navigator determines an optimal drive arc for the micro-rover to execute. Drive arcs are pre-determined arc pathways that the rover can execute at any given time. The determination of which drive arc to execute considers many factors and occurs via forward-simulation, a path evaluation method in which the footprint of the rover is projected through the local mesh in order to assess each drive arc’s traversability. At each time-step of this projection, the terrain is evaluated against various constraints, a cost is assigned to each arc, and the arc with the lowest cost is executed [18]. The most critical constraints are obstacle presence and rover stability. Consideration of energy consumption is accomplished via a bias towards arcs of larger radii to minimize the wheel torque necessary to move the rover. As arc radius increases, the path executed becomes straighter, re-

quiring less torque, and, therefore, less energy, to execute. Specifically on the power-rich MoonRanger rover [17], planning to optimize energy consumption is not strictly necessary. Local planning considering sun angle is included only when the computational power of the micro-rover is sufficient.

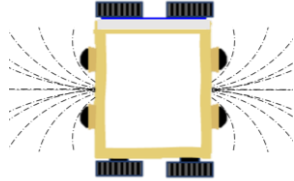


Figure 5: A Visualization of Drive Arcs

As local navigation functions on the map provided by the Terrain Mapper, the exact implementation of the planner depends on the implementation of the Terrain Mapper. In initial formulation, local planning occurred using grid-based planning for simplicity. However, mesh-based planning has historical usage on cutting-edge autonomous systems [19] and allows for better retention of terrain contour information. As such, a mesh-based implementation of local planning and terrain mapping is advantageous.

### 3.4 Terrain Mapper

Terrain mapping provides input to two major facets of a mission: local navigation on the rover and data analysis on the ground. Terrain mapping in terms of local navigation means consolidating the output from Perception into a map on which drive arcs can be forward-simulated. Terrain mapping for data analysis constitutes storing a model of the surface in a concise and correct format for downlinking to Earth within a few minutes. This terrain model is of a much higher resolution than any current models and can aid further data analysis on the ground. For example, at the lunar pole, terrain model analysis might correlate measurements of hydrogenous volatile presence with certain lunar geological formations.

The Terrain Mapper takes as input the point clouds generated by either nominal stereo matching or by line striping [2] point cloud generation. In original formulation, the terrain was represented for navigation by a 5-meter x 5-meter grid. However, a mesh-based map is a more efficient and representative map that allows for direct merging of the Global Map with modeling during an autonomous trek [19].

### 3.5 Pose Estimator

The Pose Estimator relies on input from an Inertial Measurement Unit, encoders, a sun sensor and cameras [12] to estimate robot position and orientation. The pose estimate generated using these four

inputs, known as the robust pose, is then sent by the Intercom to the embedded processor, which has its own, low-computational-cost, dead-reckoning pose estimator. On the lunar pole, there is a large variation in lighting conditions. In conditions where the sun is visible, the sun sensor, which measures sun angle, combined with knowledge of sun motion over time, provides an invaluable source absolute bearing. In ample sunlight, visual odometry methods can track features, further increasing estimate accuracy [4].

The challenging case occurs when the rover enters the darkness characteristic of the lunar poles. In this case, the sun sensor and nominal feature tracking is rendered useless. An initial approach to the lack of sunlight would be to estimate pose via only encoder and Inertial Measurement Unit data. However, this approach was determined to have error too large for mission survival. Using field data of only Inertial Measurement Unit and encoder data, collected by a surrogate rover at a lunar analog site, the position error over a short distance (Figure 6) was roughly 5 percent of the distance travelled [3]. This amount of error puts the rover and the mission at high risk of failure, especially for long-distance treks. For additional input to pose estimation when sun sensor data and visual odometry are unavailable, the use of laser line stripes [2] as input to a specialized feature tracking algorithm is being pursued.

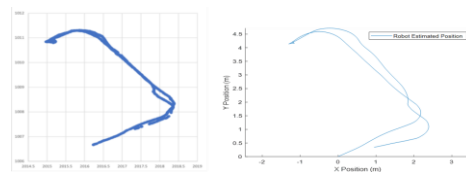


Figure 6: Ground truth (L) vs. estimate (R), using encoder and Inertial Measurement Unit data [3].

## 4 EXECUTION NODES

The execution nodes include the Rover Executive, the Health Monitor, the Telemetry Manager, the Wi-Fi Gateway, and the Intercom.

### 4.1 Rover Executive

The Rover Executive directs both the execution nodes and the navigation pipeline to ensure smooth execution on the higher-performance computer. This includes launching the correct software components and providing them with the relevant information for a given scenario, as well as determining and executing the correct response to any faults.

The Rover Executive communicates with the Wi-Fi Gateway to receive commands from the ground control station when the rover is in lander

communication range and the lander is in communication with the ground control station. Using these commands as well as access to information from the Health Monitor, the Rover Executive must run its state machine to determine the state in which the higher-performance computer should operate and the corresponding components to run. In non-autonomy states, the Rover Executive must still ensure that pose estimation, terrain mapping, and data storage functionalities are operational. For example, maneuvering the rover into an optimal position for stationary charging is executed by the embedded processor. However, the Pose Estimator must still track pose or risk the rover travelling in an incorrect direction post-charging. It is worth noting that although stationary charging may be rare, especially for power-rich micro-rovers like MoonRanger [17], the Rover Executive must still be able to handle these scenarios. For correct transition between states, the Rover Executive contains information on procedure to run when entering and exiting these states. Commands received from the ground control station preempt current execution.

In addition to launching all other software components on the higher-performance computer, the Rover Executive must ensure their faultless startup and continued operation. This requires interpreting the status messages created by the Health Monitor, which monitors information from all higher-performance computer components after each component is launched, and monitoring the Health Monitor itself. The Rover Executive executes the correct response to faults detected by the Health Monitor or detected by the Rover Executive about the Health Monitor. Therefore, the Rover Executive software must include mitigation strategies for any set of failure conditions during autonomous lunar exploration. One of the most critical mitigation strategies is prematurely returning to lander communication range. Because autonomous lunar exploration will collect data outside of the lander communication range, if a crucial fault occurs on the micro-rover during an autonomous trek, the rover must be able to return back into the communication range to send data back to ground. If a compromised rover is unable to return into communication range, all data from the trek and any future operations is at risk of being lost. The Rover Executive is tasked with making the critical decision of whether to return to lander communication range when faults occur and must guide other components accordingly.

#### 4.2 Health Monitor

The Health Monitor diagnoses major faults of four categories: software, state consistency, avionics tolerances, and mechanical limits. Software faults are those such as a software component crashing, an atypical component output, and an atypical output frequency. A state consistency fault is triggered when the state representing the mode of operation differs between the embedded processor and higher-performance computer. An example scenario may be that the embedded processor believes the rover is currently being teleoperated while the higher-performance computer believes the rover should be autonomously roving across terrain. Avionic tolerance faults are those concerning overheating, overcooling, and energy imbalance. Avionics faults are often handled by embedded processor software. State information and avionics data from the embedded processor are received by the Health Monitor via the Intercom. The final fault class, mechanical limits, is entirely handled by the embedded processor because of the embedded processor's ability to more instantaneously react to the rover exceeding tilt and other such mechanical limits. Health information and any of the preceding faults are signaled to the Rover Executive, and in a scenario in which Health Monitor output is absent, irregular, or inconsistent with embedded processor data relayed by the Intercom, the Rover Executive may decide to restart the Health Monitor.

#### 4.3 Telemetry Manager

The Telemetry Manager is responsible for handling information generated by the rover. It relays to the Wi-Fi Gateway a telemetry message including current pose and rover status, as analyzed by the Health Monitor, for continuous broadcasting while a rover is in lander communication range. Additionally, the Telemetry Manager is responsible for storing data products in a on organized fashion such that, when the rover transmits data to the lander, the Wi-Fi Gateway can find the necessary data. These data products include time-stamped rover poses, a log of telemetry messages over a trek, terrain maps, and images.

Cameras used on this class of micro-rover are of approximately half-megapixel resolution and capture images at a rate of at most 6 frames per second, creating many gigabytes of image data per autonomous trek [12]. Image compression is key to storing image data from the entirety of any autonomous trek on the rover and downlinking image data to ground within mission time. Critical in the compression process is an ability to remove the dark pixels from above the

horizon first. More data transmission strategies can be found in Schweitzer et. al. [12] [15].

#### 4.4 Wi-Fi Gateway

The Wi-Fi Gateway is responsible for handling communication between the lander and the higher-performance computer. This includes broadcasting live telemetry as received from the Telemetry Manager, which is received by a lander whenever a rover is in lander communication range. Additionally, the Wi-Fi Gateway transmits data products from the higher-performance computer to the lander in the pre-determined priority of current telemetry, log of past telemetry, terrain models, compressed images, and raw images. Messages from a ground control station via a lander are received by the Wi-Fi Gateway, which either translates the message into a format decipherable by the Rover Executive, stores a Global Map in the correct location, or returns an input error notification to the lander and the ground control station. There is a separate Wi-Fi Gateway on the embedded processor to receive commands that can only be executed on the embedded processor.

#### 4.5 Intercom

The Intercom handles communication between the higher-performance computer and the embedded processor. This includes receiving periodic messages sent from the embedded processor to the higher-performance computer, broadcasting the content of those messages as appropriate, listening to critical information calculated on the higher-performance computer, and sending that critical information to the embedded processor.

The messages received by the Intercom from the embedded processor consist of sensor data, the operating state of the embedded processor, and a basic rover health assessment. Sensor data received from the embedded processor includes readings from encoders, the inertial measurement unit, and the sun sensor, which are sent to and used by the Pose Estimator on the higher-performance computer. The operating state of the embedded processor is monitored by the Health Monitor for consistency with the Rover Executive state. The basic rover health assessment from is an indication of the thermal and power safety status of all avionic components and the rover body and is monitored by the Health Monitor, as well as an indicator of any mechanical faults being corrected by the embedded processor.

The higher-performance computer must send a periodic heartbeat, a record of current software component states, and the robust pose calculated with visual input to the embedded processor. The heart-

beat and report of higher-performance computer software state allow for the embedded processor to monitor that the higher-performance computer is running correctly and to reboot it upon observing irregular behavior. Sending the robust pose allows the embedded processor to store the robust pose in nonvolatile memory, in case the most recent pose ever needs to be recovered. Additionally, it provides a better starting point from which the dead-reckoning pose estimation on the embedded processor can occur. Finally, it allows for saving the robust pose as a data product on the space-hardened memory. This is critical when scientific data relies on a pose tag for full impact.

## 5 CONCLUSION

As micro-rover autonomy software components continue to be implemented, developed, and refined, this paper presents the core design of the software functionalities necessary to accomplish high-performance micro-rover autonomy. Beyond the bold, high-performance autonomy discussed here, the MoonRanger program pioneers the mechatronics of lunar micro-roving, exploring in darkness and ice-mapping from the lunar surface. The formulation of a micro-rover autonomy architecture will have contribution and impact reaching far beyond this early mission. The technology is enabling to diverse micro-rovers for campaigns of ice characterization, pit and cave explorations, site characterizations, power grid-ding, resource extraction, and ultimately the support of human extraterrestrial presence and enterprise.

### Appendix: Summary of Higher-Performance Computer Software Inputs and Outputs

	Component	Input	Output
Navigation Pipeline	Perception	Raw camera images	Point cloud
	Global Planner	Goal, Global Map, lander pose	Waypoints
	Local Navigator	Waypoints, terrain model, Global Map	Drive arcs
	Terrain Mapper	Point cloud	Terrain model
	Pose Estimator	Stereo Images, sensor (IMU, encoder, sun) data	Robust pose
Execution Nodes	Rover Executive	Command messages (from Wi-Fi Gateway), health status/fault info	Commands to various components (e.g. Waypoints), rover operation state
	Health Monitor	Embedded processor operating state, basic health assessment  Heartbeat, error information, and selected output from all higher-performance computer components	Health status/fault information

Telemetry Manager	Output from all higher-performance computer components, images	Live telemetry, log of telemetry, data products (stored in internal memory)
Wi-Fi Gateway	Messages from ground control station, direction from Rover Executive to downlink data to lander	Command messages (to Rover Executive), Global Map (stored in memory), data products (to lander)
Intercom	<u>Embedded processor:</u> sensor (IMU, encoder, sun) data, operating state, basic health assessment  Robust pose, drive arcs	Embedded processor sensor (IMU, encoder, sun) data, operating state, embedded processor basic health assessment  <u>To embedded processor:</u> Robust pose, drive arcs, heartbeat

### Acknowledgements

We thank the MoonRanger for contributions, including: Justin Morris for early architecture work, Lakshmi Maddirala and Guru Koushik for early planning work, William Lee-Moore and Neil Khera for development of perception components, Vaibhav Gupta and Jim Teza for embedded software development, Olivia Corrodi for simulation, the Message Dictionary Working Group, Chuck Whittaker and Lauren Jablonski for field work, and mechanical team for CAD.

The technology, rover development and flight are supported by NASA LSITP contract 80MSFC20C0008 MoonRanger.

### References

- [1] Matijevic, J., et al. "A Description of the Rover Sojourner," [mars.nasa.gov/MPF/rover/descrip.html](https://mars.nasa.gov/MPF/rover/descrip.html).
- [2] Jamal, H., et. al. "Terrain Mapping and Pose Estimation for Polar Shadowed Regions of the Moon," iSAIRAS, Virtual, October 2020.
- [3] Kumar, V. "Resource-Limited Exploration Autonomy for Planetary Rovers," SCS Honors Senior Thesis Posters, May 2020.
- [4] Maimone, M., et. al., "Two Years of Visual Odometry on the Mars Exploration Rovers," Journal of Field Robotics, March 2007.
- [5] Oberhaus, D. "How NASA Built a Self-Driving Car for Its Next Mars Mission", <https://www.wired.com/story/how-nasa-built-a-self-driving-car-for-its-next-mars-mission/>.
- [6] NASA Science. "Mars 2020 Mission Perseverance Rover [Rover][Communications][Brains]," <https://mars.nasa.gov/mars2020/spacecraft/rover/>.
- [7] Jet Propulsion Laboratory. "Mars Pathfinder," [https://mars.nasa.gov/internal\\_resources/815/](https://mars.nasa.gov/internal_resources/815/).
- [8] NASA Science. "Mars Pathfinder," <https://mars.nasa.gov/mars-exploration/missions/pathfinder/>.
- [9] NASA Mars Exploration Rovers. "Spirit and Opportunity: Twin Mars Exploration Rovers," <https://mars.nasa.gov/mer/mission/rover/>.
- [10] Jet Propulsion Laboratory. "Mars Exploration Rover," [https://www.jpl.nasa.gov/news/fact\\_sheets/mars03rovers.pdf](https://www.jpl.nasa.gov/news/fact_sheets/mars03rovers.pdf).
- [11] Razor Robotics. "Spirit and Opportunity Rovers," <https://www.razorrobotics.com/robots/spirit-and-opportunity-rovers/>.
- [12] Schweitzer, et. al. "Micro-Rover Technologies and Mission for Measuring Lunar Polar ICE," IEEE Aerospace Conference, March 2021.
- [13] Nvidia. "Extend AI Computing with the Jetson TX2i," <https://developer.nvidia.com/embedded/jetson-tx2i>.
- [14] Wettergreen, D., et. al., "Sun Synchronous Robotic Exploration: Technical Description and Field Experimentation," Intl. Journal of Robotics Research, January 2005.
- [15] Schweitzer, L. & Tang, G. "Mission Operations for Autonomous Science-Driven Lunar Micro-Roving," iSAIRAS, Virtual, October 2020.
- [16] Clean Energy Bright Futures. "Incident Angle of Sunlight" <https://www.cebrightfutures.org/learn/incident-angle-sunlight>
- [17] Bitanga, J. M., Fisch P. R. M., & Whittaker, W. L. "Thermal Modeling and Design of a Micro-Rover for Lunar Polar Exploration," iSAIRAS, Virtual, October 2020.
- [18] Kelly, A., et. al. "An approach to rough terrain autonomous mobility." International Conference on Mobile Planetary Robots, Santa Monica, 1997.
- [19] Wettergreen, D. & Wagner, M. "Developing a Framework for Reliable Autonomous Surface Mobility," iSAIRAS, Turin, September 2012.
- [20] Usenko, V., et. al. "Visual-Inertial Mapping with Non-Linear Factor Recovery," IEEE Robotics and Automation Letters (RA-L) & Int. Conference on Intelligent Robotics and Automation (ICRA), IEEE, volume 5, 2020.
- [21] Wong, U., et. al. "Polar Optical Lunar Analog Reconstruction (POLAR) Stereo Dataset," NASA Ames Research Center, May 2017.