OPTIMAL GLOBAL PATH PLANNING FOR MULTIMODAL LOCOMOTION ON LUNAR TERRAIN

Sandra C. Wells[1], Peter Zhang[1], Hendrik Kolvenbach[1], Lorenz Wellhausen[1], Nikita Rudin[1], and Marco Hutter[1]

[1]*Robotic Systems Lab, ETH Zurich, Leonhardstr. 21, 8092 Zurich, Switzerland*

ABSTRACT

Identifying robotic traverses on the surface of other celestial bodies is essential to assess the capabilities of the required system at the mission planning stage. With increasingly diverse robotic systems designs for space, including wheeled, walking, and multimodal systems, a wider range of behaviors concerning electrical energy consumption and failure risk are becoming available. Thus, it becomes necessary to define path optimality for the two parameters individually, beyond a traditional minimization of path length. This paper proposes a path planning algorithm that finds optimal global paths on the lunar surface for robotic energy consumption and risk, where the user can define the energy and risk minimization functions and their relative importance. Based on a custom A* implementation, the proposed algorithm successfully minimizes the energy consumption and path risk in various scenarios. Exemplary, a cost function for the walking robot ANYmal was generated in simulation and applied to our planner. The results show that different optimized global paths were generated depending on the user's energy/risk trade-off.

1. INTRODUCTION

The past few decades have held a renewed interest in developing new robotic missions to the Moon. Robotic exploration of the surface of the Moon holds a large scientific and non-scientific value. The design of these new lunar surface exploration missions includes the selection of interest points from which to collect new data. These interest points may fall in a wide range of terrains that can be best fit for exploration by robots with different locomotion principles. In this context, especially the use of legged systems is of high relevance, as those systems have advanced rapidly over the last decade and show promising performance on a large variety of unstructured, terrestrial terrains [1]. Thus, it might only be a matter of time until we see such systems in space [2].

An example for a highly geologically diverse region of the Moon is the Aristarchus plateau, making it a potential destination for upcoming lunar surface robotic missions [3, 4, 5]. The exploration of the Aristarchus crater, found within the plateau, would encounter a large terrain diversity that might be most effectively and safely tackled by a legged robot (e.g. ETH Zurich's SpaceBok [6, 7] or ANYmal [8]), steep walls on sinuous rilles might be best explored by a climbing robot (e.g. NASA's LEMUR 3 [9]), and flatter plateau regions might benefit most from a wheel-legged robot (e.g. NASA's ATHLETE [10] or ETH Zurich's ANYmal-on-wheels [11]).

There is currently a lack of global planners that find optimal paths between interest points for different types of robots exploring lunar terrain. Factors such as energy consumption and path safety highly depend on traversed terrain characteristics [12]. Thus, a commonly available algorithm to minimize the total distance traveled by a robot may fail to account for the limitations of each robotic operation. Our contribution is an optimal path planning algorithm that optimizes over energy consumption and risk functions particular to different robot designs and locomotion modes. The algorithm allows the user to define these functions, as well as their relative importance in the path optimization problem. Moreover, the algorithm is implemented in an online platform that can easily be accessed during the planning stage of future lunar missions.

We start this paper with with introducing the path planning methodology, including the A* implementation as well as cost and heuristic function definition in Sec. 2, followed by our software implementation in Sec. 3, results and discussion in Sec. 5, and conclusion in Sec. 5.

2. PATH PLANNING METHODOLOGY

The implemented algorithm is based on the A* planner [13]. A* is a heuristic best-first search algorithm that finds the lowest-cost path on a weighted graph. At each step of the algorithm, A* visits the new node x that minimizes

$$f(x) = g(x) + h(x) \tag{1}$$

where $g(x)$ is the cost of reaching the new node x from the start node, and $h(x)$ is a heuristic estimate of the cost of the cheapest path from the new node x to the target node.

The algorithm stops when the new visited node matches the target node.

The heuristic function h is required to be *admissible*, meaning that it never overestimates the real cost of reaching the target node. Furthermore, if the heuristic function h is *consistent*, meaning that, for all nodes x and for each neighbor node x' of x it satisfies

$$h(x') \leq c(x, x') + h(x) \tag{2}$$

where $c(x, x')$ is the cost of reaching node x' from node x, then A* is guaranteed to return the least-cost path on the graph.

2.1. Cost Function

In the proposed A* algorithm implementation, the cost function g, evaluated on a path $P = \{x_0, x_1, \ldots, x_n = x\}$ ending at node x, is defined as

$$g(x) = \alpha \cdot E_P(x) + (1 - \alpha) \cdot R_P(x) \tag{3}$$

where $E_P(x)$ and $R_P(x)$ are the energy efficiency and failure risk faced by reaching node x on the path, and $\alpha \in [0, 1]$ is a tunable, user-defined weight. This cost function is calculated by summing the cost of each edge along the path, as

$$g(x) = \sum_{i=1}^{n} c(x_{i-1}, x_i). \tag{4}$$

Let $s(x_i, x_j)$ be the directional slope from x_i to x_j. Let $r(x_j)$ be the rock abundance at x_j, and let $d(x_i, x_j)$ be the euclidean distance from x_i to x_j. Then, the cost of the edge from x_i to x_j is

$$\begin{aligned} c(x_i, x_j) = \alpha \cdot E\left(s(x_i, x_j),\ r(x_j)\right) \cdot d(x_i, x_j) \\ + (1 - \alpha) \cdot R\left(s(x_i, x_j),\ r(x_j)\right) \cdot d(x_i, x_j) \end{aligned} \tag{5}$$

where $E(s, r)$ and $R(s, r)$ are the energy efficiency and risk cost of a robot per unit distance on a slope of value s and a rock abundance of value r. It follows easily that the quantities in equation (3) correspond to

$$\begin{aligned} E_P(x) &= \textstyle\sum_{i=1}^{n} E\left(s(x_{i-1}, x_i),\ r(x_i)\right) \cdot d(x_{i-1}, x_i), \\ R_P(x) &= \textstyle\sum_{i=1}^{n} R\left(s(x_{i-1}, x_i),\ r(x_i)\right) \cdot d(x_{i-1}, x_i). \end{aligned} \tag{6}$$

The quantities $E(s, r)$ and $R(s, r)$ have a high relevance in this proposed algorithm implementation. They define the relationship between lunar terrain and robotic performance and thus, are responsible for optimal paths changing for varying robotic designs or locomotion modes. We use an example set of these functions, corresponding to the quadrupedal robot ANYmal [8] (specifically, ANYmal C). These functions are defined following evaluation data of the robot on simulated lunar terrain, using a reinforcement-learning-based locomotion controller.

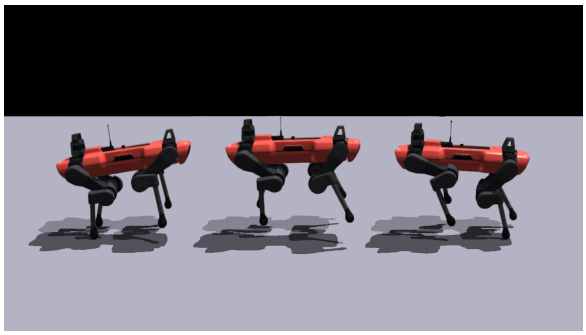

Figure 1: Bounding gait of ANYmal C, on which the example implemented cost functions are based on.

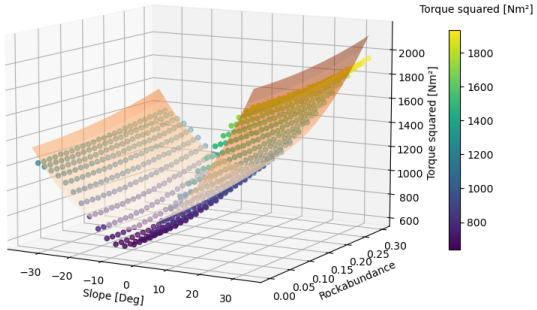

Figure 2: Second order polynomial surface fitted on the average torque squared data of Anymal C, traversing 8 m of a terrain with given slope and rock abundance.

To obtain these example functions, we trained a policy within the physics simulator Isaac Gym [14], using the approach presented in [15]. We simulated thousands of robots in parallel in lunar gravity, on a lunar terrain curriculum, consisting of slopes and boulder fields of varying difficulty. Besides velocity reference commands and proprioceptive measurements, we assume that the robot has a detailed elevation map of its surroundings available, thus also providing elevation points around the robot as observations. Following an end-to-end learning approach, we define the actions as joint position references, giving the agent the freedom to adopt any type of gait. The resulting gait, for which we characterize the energy consumption and risk functions, is a dynamic bounding gait with a long flight phase depicted in Figure 1. This gait is able to overcome slopes of up to about 40° and obstacles as high as 0.5 m. The energy efficiency and reliability are then evaluated at a fixed velocity of 0.8 m/s, traversing 8 m on different slopes and boulder fields. The robot's energy efficiency cost is calculated based on the average torque squared, approximating the actual power consumption. The evaluation data is fitted with a second order polynomial as in equation (7), using least squares. The resulting fit of the surface is shown in Figure 2, while the fitted coefficients are presented in Table 1.

$$E(s, r) = p_1 + p_2 s + p_3 r + p_4 s^2 + p_5 sr + p_6 r^2. \tag{7}$$

To prevent the robot from venturing into any terrain we deem too dangerous and unreliable, limits on both slope

and rock abundance are introduced. Rock abundance is measured via the Cumulative Fractional Area (CFA) as defined in [16]. The limits for negative and positive slope s_{min} and s_{max} are set to $\pm$ 30 degrees, while the maximum rock abundance is considered to be at 0.3. This leads to our final efficiency cost of

$$E(s,r) = \begin{cases} p_1 + p_2 s + p_3 r + p_4 s^2 + p_5 sr + p_6 r^2 & , \\ \qquad \text{if } s_{min} \leq s \leq s_{max} & \\ \infty & , \\ \qquad \text{else.} & \end{cases} \tag{8}$$

In turn, the design of the risk cost is based on the crash rate, where a crash is defined as a contact of the robot's base with the environment or itself. It is set up in a similar fashion, fitting the second order polynomial surface from equation (7) to performance data of ANYmal C. In addition, we bound the minimal value at zero, as no negative crash rate is possible. While the standard least squares method was used for the efficiency cost, a weighted approach was taken in this case. To maintain the right trend that the minimal risk at a fixed slope is at zero rock abundance, we assign the following weighting to each data point:

$$\sigma(s,r) = \left(\frac{1}{10}\left(|s| + 100r\right)\right)^2 + 0.1. \tag{9}$$

The weighted least squares approach minimizes

$$\sum_{i}^{n_{data}} \left(\frac{r_i}{\sigma(s_i, r_i)}\right)^2, \tag{10}$$

where we sum over all data points n_{data}, weighting the residual fitting error r_i of each point with the assigned σ. The resulting surface is shown in Figure 3. The fit sets a higher importance on commonly found, low rock abundance values and slopes around zero. As a consequence, the fitting errors close to s_{min}, s_{max} and r_{max} are high. Taking a more complicated function would prevent such errors, but since these combinations of terrain properties are rarely encountered, our choice only leads to a minimal error. The coefficients of the surfaces are presented in Table 1. The risk cost is then calculated based on the fitted function for the crash rate $crash(s,r)$. We first calculate the crash rate on the relevant path section of length d based on $crash(s,r)$, as the latter is the value for traversing a distance of 8 m. We get the final cost by scaling it with the maximum possible efficiency cost and the crash rate at $(\frac{1}{2}s_{max}, \frac{1}{2}r_{max})$. As such, the risk cost is given by:

$$\begin{aligned} R(s,r) =& E(s_{max}, r_{max}) \\ & \cdot \frac{1 - (1 - crash(s,r))^{\frac{d}{8}}}{1 - (1 - crash(\frac{1}{2}s_{max}, \frac{1}{2}r_{max}))^{\frac{d}{8}}}. \end{aligned} \tag{11}$$

The coefficients are specific to a robotic design and locomotion mode and by changing the values of the cost function on each node on the terrain, they change the optimality of different possible paths from start to goal.

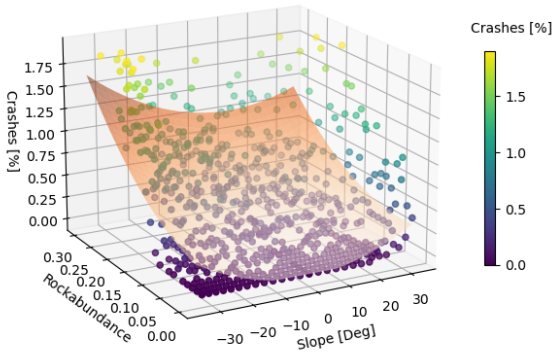

Figure 3: Second order polynomial surface fitting the crash rate with a weighted least squares approach, weighting slopes and boulder fields around zero higher.

Table 1: Resulting coefficients from fitting the torque squared and crash rate with a second order polynomial surface.

	p_1	p_2	p_3	p_4	p_5	p_6
Energy	803	10.5	70.3	73.9	-1.42	1770
Risk	-0.0288	0.00531	0.319	0.00314	-0.0230	10.8

2.2. Heuristic Function

Let x_t be the target node of a given path planning problem, and let $d(x, x_t)$ be the euclidean distance between node x and node x_t. The heuristic function implemented in the A* algorithm is

$$h(x) = \alpha \cdot E(s_h, r_h) \cdot d(x, x_t) \tag{12}$$

where α and E are the importance weight and efficiency cost, as defined in Section 2.1. We take the efficiency cost at its minimum at $(s_h.r_h)$, found with Newton's method to be $s_h = -7.158$ degrees and $r_h = 0$ in our case. Intuitively, it can be seen that this choice of heuristic function never overestimates the cost $g(x)$ defined in equation (3). In particular, $E(s,r) \geq E(s_h, r_h)$ and $R(s,r) \geq 0$, meaning that the cost of a path starting at x and ending at x_t, $\{x_0 = x, x_1, \ldots, x_n = x_t\}$, can be lower bounded as

$$\begin{aligned} c(x, x_t) =& \ \alpha \cdot \sum_{i=1}^{n} E\left(s(x_{i-1}, x_i),\ r(x_i)\right) \cdot d(x_{i-1}, x_i) \\ & + (1 - \alpha) \cdot \sum_{i=1}^{n} R\left(s(x_{i-1}, x_i),\ r(x_i)\right) \cdot d(x_{i-1}, x_i) \\ & \geq \alpha \cdot E(s_h, r_h) \cdot \sum_{i=1}^{n} d(x_{i-1}, x_i) + 0 \\ & \geq \alpha \cdot E(s_h, r_h) \cdot d(x, x_t) \end{aligned} \tag{13}$$

where the last inequality uses the fact that since all distances are calculated on a two-dimensional plane, the distance traveled by any path from x to x_t can only be larger

or equal to the straight line connecting them. The implemented heuristic is thus admissible.

This choice of heuristic can, furthermore, be shown to be consistent. Some initial results are

$$\begin{aligned} h(x') &= \alpha \cdot E(s_h, r_h) \cdot d(x', x_t) \\ c(x', x) &\geq \alpha \cdot E(s_h, r_h) \cdot d(x', x) \\ h(x) &= \alpha \cdot E(s_h, r_h) \cdot d(x, x_t) \end{aligned} \tag{14}$$

and, from the triangle inequality, we have

$$d(x', x_t) \leq d(x', x) + d(x, x_t). \tag{15}$$

Equation (15) can be multiplied on both sides by the constant non-negative value $\alpha \cdot E(s_h, r_h)$, giving

$$\begin{aligned} &\alpha \cdot E(s_h, r_h) \cdot d(x', x_t) \\ &\quad \leq \alpha \cdot E(s_h, r_h) \cdot d(x', x) + \alpha \cdot E(s_h, r_h) \cdot d(x, x_t) \end{aligned} \tag{16}$$

which, using the initial results from equation (14), gives

$$h(x') \leq \alpha \cdot E(s_h, r_h) \cdot d(x', x) + h(x) \leq c(x', x) + h(x). \tag{17}$$

Since the heuristic is consistent, the A* implementation is guaranteed to return the optimal path on the searched graph, for the specified cost function.

3. SOFTWARE IMPLEMENTATION

The proposed algorithm is implemented in web application, with the intention of making it an accessible tool where users can obtain and assess optimal paths on lunar terrain, customized for their specific robot. The platform that runs the planner contains different stages for terrain preprocessing, online user interaction, and path planning. Through the online user interface, the user can set parameter values for their robot's cost function, and choose the terrain patch to plan a path on. Moreover, paths can be planned with multiple stops along the way.

Patches of lunar terrain data are obtained from Lunar Reconaissance Orbiter Camera (LROC) [17] data and downloaded from QuickMap [1]. The downloaded maps have a resolution of 256 by 256 px and they are processed in Python by the GDAL library [18]. From each terrain patch, we obtain the resolution, height, slope and rock abundance information.

The resolution, height, slope and rock abundance data, as well as the energy and risk function parameters, their relative weighting and the start and target point of a path segment, are given as inputs to the planner backend. The planner then obtains both the direct and optimal paths between the input points, together with their height, traveled distance, directional slope and rock abundance data.

[1] https://quickmap.lroc.asu.edu

Through the user interface, the user can input their cost function and path data. The platform returns the optimal path between the interest points, downloadable as a file, and shown graphically on height, slope and rock abundance maps of the terrain. It also provides the user with plots and histograms of the height, directional slope and rock abundances encountered along the path, for easy assessment of the obtained path by the user.

4. RESULTS AND DISCUSSION

We tested our algorithm for a set of start and target points around the Aristarchus crater. It successfully minimized the energy cost and path risk in a variety of scenarios. We present an example of these results, near the Aristarchus central peak, a geologically interesting terrain which has been identified as one of the scientific targets for future exploration mission by the Constellation program office [19]. Figure 4 depicts two optimal paths that connect potential points of interest on either side of the central peak, showing two different types of approaches. Depending on the weighting α, either a safer or a more energy-efficient path is optimal; taking a detour around or climbing over the peak.

Table 2: Values given by the A* planner of two optimal paths, for varying α, around the Aristarchus central peak, shown in Figure 4.

	Distance (m)	Crash rate	Energy cost (-)
$\alpha = 0.3$	3197	39%	2750
$\alpha = 0.5$	1659	52%	1718

The path planner successfully minimizes the energy cost and path risk in a variety of scenarios. Moreover, changing the importance weighting, one can plan and select different paths according to the robot's robustness. In our case, we can see from the resulting paths in Table 2 and Figure 4, and from the cost functions in section 2.1, that additional effort in the robot's reliability on boulder fields could greatly cut the mission's energy consumption and traveled distance.

5. CONCLUSION

We propose an algorithm that performs path planning that successfully optimizes energy consumption and path risk. The robot's energy consumption and risk as functions of the traversed terrain characteristics, as well as their relative importance, are user-customizable. This implies that the optimal path planner can incorporate different robot designs and locomotion modes. Planned paths can include multiple intermediate goals and all segments in a resulting multiway point path are optimal paths. Moreover, paths can be planned on any lunar terrain patch after its incorporation into the pre-processing stage and into

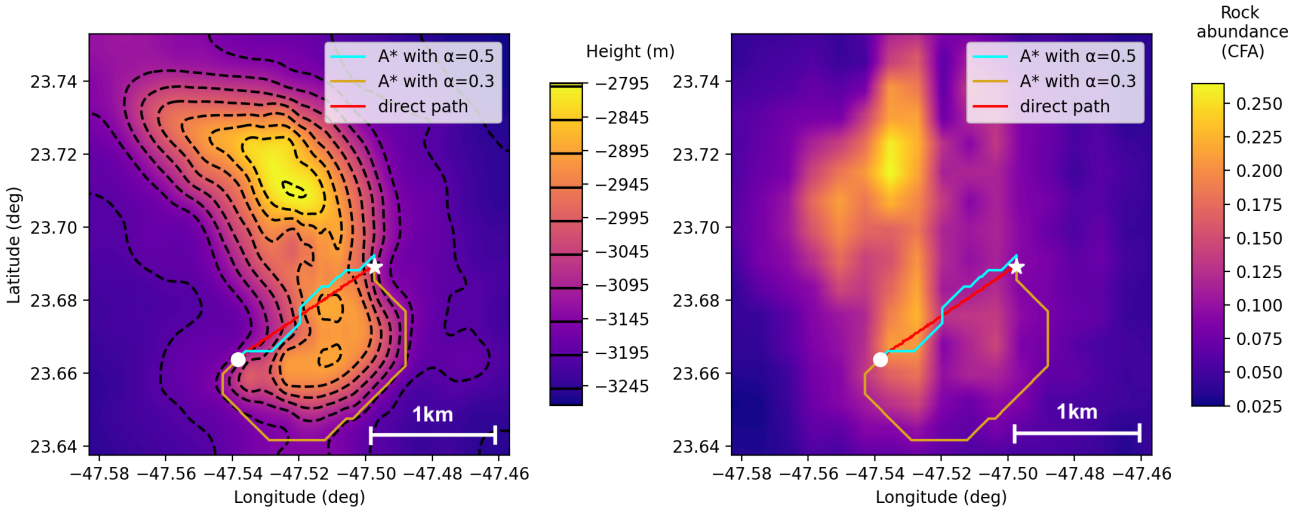

Figure 4: Two optimal paths around the Aristarchus central peak, planned with different importance weighting α, starting at the circle and targeting the star, together with the direct path between the two points. The paths are depicted on filtered elevation and rock abundance maps, covering a 3.5 by 3.5 km area.

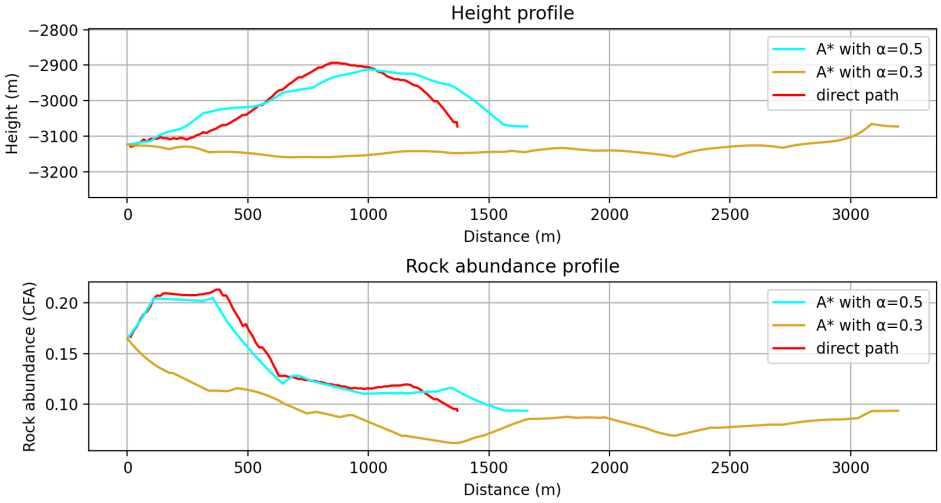

Figure 5: Height and rock abundance profiles encountered along the two optimal paths and the direct path shown in Figure 4.

the website framework. Last, the developed web platform is easy to use and provides the user with path data that are relevant for a robotic lunar mission's planning stage. We believe that tools like the one presented in this work will become a useful addition in mission planning phase to quickly iterate on potential robot trajectories.

Some natural continuations of this work might include advancing the website development to improve the user experience and open-sourcing the software. They might also include proposing algorithm modifications for optimizing over gait switches along a path, for robots that allow for multimodal locomotion. Gait-switching implies an additional dimension for the planning problem and the selection of the most adequate planning method could be revised accordingly.

ACKNOWLEDGEMENTS

The authors thank Heinrich Heinzer for his contribution to setting up the online platform for this project.

We acknowledge the use of imagery from Lunar QuickMap (https://quickmap.lroc.asu.edu), a collaboration between NASA, Arizona State University Applied Coherent Technology Corp.

The project that gave rise to these results received the support of a fellowship from "la Caixa" Foundation (ID 100010434). The fellowship code is LCF/BQ/EU20/11810074.

REFERENCES

[1] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62):eabk2822, 2022.

[2] Hendrik Kolvenbach. *Quadrupedal Robots for Planetary Exploration*. PhD thesis, ETH Zurich, Institute of Robotics and Intelligent Systems (IRIS), 2021.

[3] H.J. Moore S.H. Zisk, C.A. Hodges. The Aristarchus-Harbinger region of the moon: Surface geology and history from recent remote-sensing observations. *The Moon*, 17:59–99, 1977.

[4] John F. Mustard, Carle M. Pieters, Peter J. Isaacson, James W. Head, Sebastien Besse, Roger N. Clark, Rachel L. Klima, Noah E. Petro, Matthew I. Staid, Jessica M. Sunshine, Cassandra J. Runyon, and Stefanie Tompkins. Compositional diversity and geologic insights of the Aristarchus crater from Moon Mineralogy Mapper data. *Journal of Geophysical Research: Planets*, 116(E6), 2011.

[5] W Brent Garry and Jacob E Bleacher. Emplacement scenarios for Vallis Schröteri, Aristarchus Plateau, the Moon. *Geological Society of America Special Papers*, 477:77–93, 2011.

[6] Philip Arm, Radek Zenkl, Patrick Barton, Lars Beglinger, Alex Dietsche, Luca Ferrazzini, Elias Hampp, Jan Hinder, Camille Huber, David Schaufelberger, Felix Schmitt, Benjamin Sun, Boris Stolz, Hendrik Kolvenbach, and Marco Hutter. Spacebok: A dynamic legged robot for space exploration. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6288–6294, 2019.

[7] Hendrik Kolvenbach, Elias Hampp, Patrick Barton, Radek Zenkl, and Marco Hutter. Towards jumping locomotion for quadruped robots on the moon. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5459–5466, 2019.

[8] Marco Hutter, Christian Gehring, Andreas Lauber, Fabian Gunther, C. Dario Bellicoso, Vassilios Tsounis, Peter Fankhauser, Remo Diethelm, Samuel Bachmann, Michael Bloesch, Hendrik Kolvenbach, Marko Bjelonic, Linus Isler, and Konrad Meyer. ANYmal - toward legged robots for harsh environments. *Advanced Robotics*, 31(17):918–931, 2017. doi: 10.1080/01691864.2017.1378591.

[9] Aaron Parness, Neil Abcouwer, Christine Fuller, Nicholas Wiltsie, Jeremy Nash, and Brett Kennedy. Lemur 3: A limbed climbing robot for extreme terrain mobility in space. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5467–5473, 2017.

[10] Brian H. Wilcox, Todd Litwin, Jeff Biesiadecki, Jaret Matthews, Matt Heverly, Jack Morrison, Julie Townsend, Norman Ahmad, Allen Sirota, and Brian Cooper. Athlete: A cargo handling and manipulation robot for the moon. *Journal of Field Robotics*, 24 (5):421–434, 2007.

[11] Marko Bjelonic, C. Dario Bellicoso, Yvain de Viragh, Dhionis Sako, F. Dante Tresoldi, Fabian Jenelten, and Marco Hutter. Keep rollin'—whole-body motion control and planning for wheeled quadrupedal robots. *IEEE Robotics and Automation Letters*, 4 (2):2116–2123, 2019. doi: 10.1109/LRA.2019.2899750.

[12] Hendrik Kolvenbach, Philip Arm, Elias Hampp, Alexander Dietsche, Valentin Bickel, Benjamin Sun, Christoph Meyer, and Marco Hutter. Traversing Steep and Granular Martian Analog Slopes With a Dynamic Quadrupedal Robot. *arXiv e-prints*, art. arXiv:2106.01974, June 2021.

[13] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, USA, 3rd edition, 2009. ISBN 0136042597.

[14] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance GPU-based physics simulation for robot learning. *arXiv*, 2021. URL `https://arxiv.org/abs/2108.10470`.

[15] Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 91–100. PMLR, 08–11 Nov 2022.

[16] Yuan Li and Bo Wu. Analysis of rock abundance on lunar surface from orbital and descent images using automatic rock detection. *Journal of Geophysical Research: Planets*, 123(5):1061–1088, 2018.

[17] Brylow S.M. Tschimmel M. et al. Robinson, M.S. Lunar Reconnaissance Orbiter Camera (LROC) instrument overview. *Space Sci Rev*, 150:81–124, 2010.

[18] GDAL/OGR contributors. *GDAL/OGR Geospatial Data Abstraction software Library*. Open Source Geospatial Foundation, 2020. URL `https://gdal.org`.

[19] PG Lucey, JT Gillis-Davis, BR Hawke, LA Taylor, MB Duke, T Brady, and T Mosher. Leag review of constellation program regions of interest for human exploration of the moon. In *Lunar Reconnaissance Orbiter Science Targeting Meeting*, volume 1483, pages 73–74, 2009.