

Comparison of SLAM algorithms on Unity-ROS(Robot Operating System) Simulator for Space Applications(URSSA)

Midhun S. Menon¹ and Michael E. Walker² and Daniel Koris² and Daniel Szafrir^{2,3} and Jack Burns¹

Abstract—Robots are critical enablers for space exploration as they help offset safety risks associated with a human astronaut, aid in precursor missions prior to manned missions, provide critical on-mission and post-hoc support. However, designing robotic systems to navigate, map and localize itself in hostile and unknown extraterrestrial worlds remains open challenge. The prime reason for this being the testing the semi-autonomous agents for robust performance in such environments. In this paper, we address this issue by using Unity-ROS(Robot Operating System) Simulator for Space Applications(URSSA) for compare performance of three popular Visual Inertial Odometry (VIO) Simultaneous Localization And Mapping (SLAM) algorithms on lunar surface. The test architecture, agent modelling, rationale for test cases and generation of ground truth data is discussed. Paper concludes with results from simulations comparing the algorithms and discusses on failure reasons and potential directions for improving algorithms for better applicability such environments. We believe such a comparative study can give vital pointers for future research directions and help accelerate development of specific algorithms for SLAM in extra terrestrial environments.

I. INTRODUCTION

The last decade has seen major space faring nations chart out focussed plans for deep space exploration starting with the moon. The Artemis I(Ref.), Chang'e-4 mission(Ref.) and Chandrayaan-2(Ref.) are some of the ongoing and completed precursor missions to put human explorers on lunar surface. These missions have robotic systems onboard to do remote investigation,survey and data collection of the lunar surface. The follow-up missions to all of these missions have a increasing component of robotic systems of which ground based mobility systems are central. Wheeled rovers are the most commonly used platform for planetary surface mobility. To date, they have been used for surface exploration, mapping and scientific investigation. However, future missions call for advanced mission operations including periodic monitoring of geo-spatially distributed scientific assets, remote assembly and In-Situ Resource Utilization(ISRU)(Ref.) tasks. The first of these call for navigation, mapping and localization capabilities over large ranges and the ISRU tasks require advanced manipulation and dexterity capabilities. More importantly, these algorithms need to be tested under a rich and varying environment to make sure they perform reliably. In this paper, the focus is on the testing of algorithms for navigation, mapping and

localization. Currently, space robotics community relies on mission analogs for evaluation and testing of such on-board algorithms. These are operations carried out on Earth in a mock-up of the true mission environment. However, mission analogs are incapable of simulating certain critical, unique and finer aspects of the extraplanetary environment (e.g., gravity, photometric anomalies etc.) which impacts testing aefficacy. As an example, visual artefacts present in target environment but not simulated in a mission analog may artificially inflate expectations and over-estimate algorithm performance during testing. On similar lines, locating a matching topography for a mission analog on earth can also prove increasingly challenging as the geological processes governing surface formation and evolution will differ entirely in most cases.

In this research, we wish to address the problem of designing a virtual analogs for extra-terrestrial environments within a multiphysics framework for advancing research in algorithms that govern low-level autonomy and support interactive trade-offs between various levels of supervisory control. Furthermore, such virtual environments may enable future explorations into the design of new interfaces that support ground control and/or astronaut operation of surface robots from orbital stations to significantly improve critical space exploration missions.

Some of the existing frameworks for simulating the Moon have been used by NASA[1]. However, they have not been designed for evaluating robotic vision-based navigation algorithms and are limited in scalability and their ability to provide perceptually real-time simulation. In addition, other frameworks are mostly based on classic simulation environments such as Gazebo[2], which are not necessarily as efficient, realistic, or scalable as modern engines such as Unity ([3], [4]).

This is the reason for simulators becoming increasingly based on frameworks like ROS[5] and Unity([6], [7]). Hence, in this paper, in-house developed simulation framework Unity-ROS Simulator for Space Applications(URSSA)(Ref.) is used, that leverages the Unity game engine as a virtual environment simulator to take advantage of its advanced rendering pipeline, support for reliable physics engines and scalable memory management capabilities. ROS is used as the framework to model the autonomous/semi-autonomous robotic systems interacting with the virtual world in Unity. We demonstrate the comparative testing of three popular Visual and/or Inertial Odometry (VIO)(Ref.) Simultaneous Localization And Mapping (SLAM) algorithms for surface exploration of moon. Only visual and inertial sensing modal-

*This work is directly supported by the NASA Solar System Exploration Research Virtual Institute cooperative agreement 80ARC017M0006.

¹Center for Astrophysics and Space Astronomy, University of Colorado Boulder

²Department of Computer Science, University of Colorado, Boulder

³ATLAS Institute, University of Colorado, Boulder

ities have been considered here as these are the ubiquitous sensors reliably flight tested and proven with minimal mass penalties on multiple missions to date by space agencies across the world. However, it may be noted that URSSA has no such restrictions.

II. SIMULATOR MODEL

The simulator architecture is shown in Figure (Add architecture Figure) and consists of three main components. Unity is used for simulating the environment, topography, photometry, rover dynamics and terrain interaction. The second main component is ROS and it is used to simulate the agent behaviour, which essentially takes in the cognitive information of the environment through the simulated sensors, and generates appropriate inferences(SLAM) and actions on how to interact with the environment(path planning). These two independent components are coupled through a communication framework called ROS#([7]), which also takes care of time synchronization and guarantees across the components. The details of the modelling are in (Ref.). The following sections give a brief overview each of these components.

A. Environment model(Unity)

The virtual environment is modeled as multiple interacting sub-systems inside Unity environment.

1) *Topographic Model*: The lunar topography is simulated using high-resolution synthetic terrain modeled on actual lunar surface data from Lunar Reconnaissance Orbiter(LRO)-Narrow Angle Camera(NAC) Digital Terrain Models(DTM) which are available to a resolution of 0.5m(at best). Sub-millimeter super-resolved terrain is generated from these DTMs using observed lunar surface roughness parameters([8], [9]), crater distributions([10], [11]) and boulder distributions [12] from the Apollo missions to generate physically realistic terrain by fractal expansion algorithms. Rocks of envelope diameters in range of 0.5m–2.0m have also been procedurally generated and uniformly distributed on the terrain as these are not captured by LRO-NAC DTM in its resolution limits.(Add teaser figure of terrain)

2) *Photometric Model*: Custom shaders are developed based on Hapke photometric model[13] to simulate the photometric response of moon called Opposition Surge(OS)[14](add figure of OS), a specific behaviour exhibited by moon at low phase angles which induces visual artefacts like glare, lens flares and sensor saturation in optical imaging systems. Such artefacts can drastically affect mapping algorithms and perception-aided planners [15]. The Hapke Bidirectional Reflectance Distribution Function(BRDF) is implemented in the Unity Physically Based Rendering(PBR) pipeline[16] and the parameters for the same are derived from lunar parameter maps[17]. The variables in the BRDF are the incident angle θ_i , the reflected angle θ_r and the phase angle ϕ . The plot of the Hapke BRDF is shown in Figure (Add BRDF plot), which shows the OS when the phase angle approaches zero.

3) *Rover Mechanical Model*: The rover is modeled as a multi-body system based off the design of Mars Exploration Rover(MER). The rover geometry has been modeled inside Unity((Add and reference rover image)) along with a stereo camera pair at the mast head and Inertial Measurement Unit(IMU) placed on the body. Every attempt is made to match the model rover design and parameters to the MER[18]. However, the rocker-bogie suspension stiffness/damping parameters and the wheel masses have been tuned to minimize disturbances due to terrain irregularities on the rover body. Thus the suspension in-effect acts as a mechanical low-pass filter to terrain disturbances and also helps guarantee minimum three point contact at all times. The six independently powered wheels are distributed as three pairs - front, middle and back. Of these, the front and back pairs act as steering wheels. Moreover, rover left and right rockers counter rotate actively to self balance the rover when moving over obstacles, as implemented in MER.

4) *Rover Sensor Models*: An exploratory rover operating in an unknown/partially known environment will likely have multiple sensors that collect data across various modalities [19], [20], ranging from hyper-spectral optical imagers (visible, infrared, ultraviolet, radio-spectrum etc.), haptic sensors, 3D ranging sensors (structured light sensors, stereo imagers, LIDARs etc.), odometry sensors, inertial sensors, etc. While the framework theoretically enables simulating any of these sensor modalities, two types of sensors are used in this paper : visible spectrum cameras (monocular and binocular) and Inertial Measurement Unit (IMU), the reasons being that these are the most reliable and least space/mass penalizing sensors on extra-terrestrial mobility platforms.

The IMU is modeled with based on sensor models suggested by [21] and [22] and elucidated in [23]. In short, the ground truth pose data of the frame is corrupted with noise from double integration of bias, scale factor uncertainties, and Additive White Gaussian Noise (AWGN) on angular rates (gyros) and accelerations (accelerometers).

The imaging sensor is modeled as a pinhole camera with radial distortion. The camera parameters are kept at the same values as the MER mast camera stereo pair(add table of parameters).

5) *Wheel-Terrain Interaction Model*: The wheels on the rover have been modeled as ellipsoids so as to simplify collider computations. Wheeled vehicles on lunar regolith experiences skid, slip and sinkage[24]. There are various traction models in literature to predict the contact forces at the wheel soil interface based on continuum methods([25], [26], [27]), Discrete Element Models([28], [29], [30], [31], [29]) and empirical models([32], [33]). However, as a first level simplification, in this paper, slip alone is considered with Coulomb dry friction model, wherein traction force is proportional to the normal reaction at the wheel-soil interface. Subsequent studies will incorporate some of the more advanced traction models mentioned above.

B. Agent Model(ROS)

The virtual robotic agents which are the processing and decision making portion of the robot is implemented as nodes in ROS. The SLAM algorithm runs as a ROS node, which estimates the robot's pose and trajectory. A ROSbridge node[34] acts as the modem between ROS and Unity. ROSbridge is used to stream sensor data from Unity to ROS nodes and send drive commands ROS to Unity.

C. Communication Framework(ROS#)

The ROS# framework is used to enable communication between Unity and ROS. The framework enables Unity to publish and subscribe to topics on a ROS nodes by encoding/decoding information using JSON format and communicating through websockets.

1) *Time Synchronization*: Right out-of-the-box, the system does not have time synchronization as the time-scales vary between ROS and Unity, due to which time flows at different rates. As the publisher-subscriber model is asynchronous, time needs to be explicitly synchronized as all estimates are time stamped. Our architecture leverages a feature of ROS to coordinate timing across its distributed architecture as a properly designed ROS node may use all internal ROS timing mechanisms rather than machine-level timing mechanisms. These ROS timing mechanisms can be driven by an external synchronizing entity. In our case, since Unity is providing the simulation environment and is seen as the potential timing bottle-neck in the system, we publish a topic from Unity to drive the ROS internal timings. Since all data is time-stamped on both ends, and Unity is the singular clock generating timestamps, our framework is robust in terms of distributed temporal synchronization; in other words, if there is a delay on the Unity side, that delay will be propagated in the ROS system (although we note in our testing to-date, the system runs in perceptually real time).

2) *Real Time compliance*: If the simulator is to plugin to an external agent with independent clock, the combined system will loose time-synchronization. A typical scenario is the case where the simulator is interfaced with a independent SLAM algorithm running its own clock internally, in which case estimation results will be wrongly time stamped causing accumulated error over time. In order to avoid this, the WebRTC architecture[?] is implemented for the Unity websocket communication. The WebRTC is a Real-Time Compliant websocket communication architecture with hard-realtime guarantees, thereby helping to mitigate the timing issue and making the simulator robust.

3) *Fixed Update*: The *FixedUpdate* method is used for the physics calculations in Unity by which fidelity of the physics calculations are guaranteed at a fixed rate independent of the rendering frame rate. This helps make sure that physics computations are not skipped when the rendering frame rate gets throttled, making the simulation more robust to numerical errors and instabilities from large integration steps.

III. TEST SYSTEM MODEL

A. Test-Case generation

IV. SIMULATION

V. RESULTS AND DISCUSSION

VI. CONCLUSION

Before you begin to format your paper, first write and save the content as a separate text file. Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar:

A. Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

B. Units

- Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as 3.5-inch disk drive.
- Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.
- Do not mix complete spellings and abbreviations of units: Wb/m² or webers per square meter, not webers/m². Spell out units when they appear in text: . . . a few henries, not . . . a few H.
- Use a zero before decimal points: 0.25, not .25. Use cm³, not cc. (bullet list)

C. Equations

The equations are an exception to the prescribed specifications of this template. You will need to determine whether or not your equation should be typed using either the Times New Roman or the Symbol font (please no other font). To create multileveled equations, it may be necessary to treat the equation as a graphic and insert it into the text after your paper is styled. Number equations consecutively. Equation numbers, within parentheses, are to position flush right, as in (1), using a right tab stop. To make your equations more compact, you may use the solidus (/), the exp function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash rather than a hyphen for a minus sign. Punctuate

equations with commas or periods when they are part of a sentence, as in

$$\alpha + \beta = \chi \quad (1)$$

Note that the equation is centered using a center tab stop. Be sure that the symbols in your equation have been defined before or immediately following the equation. Use (1), not Eq. (1) or equation (1), except at the beginning of a sentence: Equation (1) is . . .

D. Some Common Mistakes

- The word data is plural, not singular.
- The subscript for the permeability of vacuum μ_0 , and other common scientific constants, is zero with subscript formatting, not a lowercase letter o.
- In American English, commas, semi-/colons, periods, question and exclamation marks are located within quotation marks only when a complete thought or name is cited, such as a title or full quotation. When quotation marks are used, instead of a bold or italic typeface, to highlight a word or phrase, punctuation should appear outside of the quotation marks. A parenthetical phrase or statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.)
- A graph within a graph is an inset, not an insert. The word *alternatively* is preferred to the word *alternately* (unless you really mean something that alternates).
- Do not use the word *essentially* to mean approximately or effectively.
- In your paper title, if the words that uses can accurately replace the word *using*, capitalize the *u*; if not, keep using lower-cased.
- Be aware of the different meanings of the homophones *affect* and *effect*, *complement* and *compliment*, *discreet* and *discrete*, *principal* and *principle*.
- Do not confuse *imply* and *infer*.
- The prefix *non* is not a word; it should be joined to the word it modifies, usually without a hyphen.
- There is no period after the *et* in the Latin abbreviation *et al.*.
- The abbreviation *i.e.* means that is, and the abbreviation *e.g.* means for example.

VII. USING THE TEMPLATE

Use this sample document as your LaTeX source file to create your document. Save this file as **root.tex**. You have to make sure to use the **cls** file that came with this distribution. If you use a different style file, you cannot expect to get required margins. Note also that when you are creating your out PDF file, the source file is only part of the equation. *Your \TeX \rightarrow PDF filter determines the output file size. Even if you make all the specifications to output a letter file in the source - if your filter is set to produce A4, you will only get A4 output.*

It is impossible to account for all possible situation, one would encounter using \TeX . If you are using multiple \TeX

files you must make sure that the “MAIN“ source file is called **root.tex** - this is particularly important if your conference is using PaperPlaza’s built in \TeX to PDF conversion tool.

A. Headings, etc

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced. Styles named Heading 1, Heading 2, Heading 3, and Heading 4 are prescribed.

B. Figures and Tables

Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation Fig. 1, even at the beginning of a sentence.

TABLE I
AN EXAMPLE OF A TABLE

One	Two
Three	Four

We suggest that you use a text box to insert a graphic (which is ideally a 300 dpi TIFF or EPS file, with all fonts embedded) because, in an document, this method is somewhat more stable than directly inserting a picture.

Fig. 1. Inductance of oscillation winding on amorphous magnetic core versus DC bias magnetic field

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity *Magnetization*, or *Magnetization*, *M*, not just *M*. If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write *Magnetization (A/m)* or *Magnetization A[m(1)]*, not just *A/m*. Do not label axes with a ratio of quantities and units. For example, write *Temperature (K)*, not *Temperature/K*.

VIII. CONCLUSIONS

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

APPENDIX

Appendixes should appear before the acknowledgment.

ACKNOWLEDGMENT

The preferred spelling of the word acknowledgment in America is without an e after the g. Avoid the stilted expression, One of us (R. B. G.) thanks . . . Instead, try R. B. G. thanks. Put sponsor acknowledgments in the unnumbered footnote on the first page.

References are important to the reader; therefore, each citation must be complete and correct. If at all possible, references should be commonly available publications.

REFERENCES

- [1] M. Allan, U. Wong, P. M. Furlong, A. Rogg, S. McMichael, T. Welsh, I. Chen, S. Peters, B. Gerkey, M. Quigley, *et al.*, "Planetary rover simulation for lunar exploration missions," in *2019 IEEE Aerospace Conference*. IEEE, 2019, pp. 1–19.
- [2] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*(IEEE Cat. No. 04CH37566), vol. 3. IEEE, 2004, pp. 2149–2154.
- [3] U. G. Engine, "Unity game engine-official site," *Online*][Cited: October 9, 2008.] <http://unity3d.com>, pp. 1534–4320, 2008.
- [4] A. Konrad, "Simulation of mobile robots with unity and ros: A case-study and a comparison with gazebo," 2019.
- [5] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.
- [6] E. Babaian, M. Tamiz, Y. Sarti, A. Mogoei, and E. Mehrabi, "Ros2unity3d; high-performance plugin to interface ros with unity3d engine," in *2018 9th Conference on Artificial Intelligence and Robotics and 2nd Asia-Pacific International Symposium*. IEEE, 2018, pp. 59–64.
- [7] M. Bischoff, "ROS#," June 2019. [Online]. Available: <https://github.com/siemens/ros-sharp/releases/tag/v1.5>
- [8] M. S. C. (US), *Analysis of Apollo 10: Photography and Visual Observations*. Scientific and Technical Information Office, National Aeronautics and Space, 1971, vol. 232.
- [9] L. C. Rowan, J. F. McCauley, and E. A. Holm, "Lunar terrain mapping and relative-roughness analysis," 1971.
- [10] G. Neukum, B. König, and J. Arkani-Hamed, "A study of lunar impact crater size-distributions," *The moon*, vol. 12, no. 2, pp. 201–229, 1975.
- [11] G. H. Heiken, D. T. Vaniman, and B. M. French, "Lunar sourcebook-a user's guide to the moon," *Research supported by NASA., Cambridge, England, Cambridge University Press, 1991, 753 p. No individual items are abstracted in this volume.*, 1991.
- [12] R. Watkins, K. Mistick, B. Jolliff, and S. Lawrence, "Boulder distributions around young lunar impact craters: Case study of south ray crater," in *Lunar and Planetary Science Conference*, vol. 49, 2018, p. 1146.
- [13] B. Hapke, *Theory of reflectance and emittance spectroscopy*. Cambridge university press, 2012.
- [14] T. Gehrels, T. Coffeen, and D. Owings, "Wavelength dependence of polarization. iii. the lunar surface," *The Astronomical Journal*, vol. 69, 1964.
- [15] K. Otsu, A.-A. Agha-Mohammadi, and M. Paton, "Where to look? predictive perception with applications to planetary exploration," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 635–642, 2017.
- [16] A. Prancėvičius and R. Dude, "Physically based shading in unity," in *Game Developer's Conference*, 2014.
- [17] H. Sato, M. Robinson, B. Hapke, B. Denevi, and A. Boyd, "Resolved hapke parameter maps of the moon," *Journal of Geophysical Research: Planets*, vol. 119, no. 8, pp. 1775–1805, 2014.
- [18] J. A. Crisp, M. Adler, J. R. Matijevic, S. W. Squyres, R. E. Arvid-

- son, and D. M. Kass, "Mars exploration rover mission," *Journal of Geophysical Research: Planets*, vol. 108, no. E12, 2003.
- [19] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe, "Mobile robot positioning: Sensors and techniques," *Journal of robotic systems*, vol. 14, no. 4, pp. 231–249, 1997.
- [20] S. Ruocco, *Robot sensors and transducers*. Springer Science & Business Media, 2013.
- [21] P. G. Savage, "Strapdown inertial navigation integration algorithm design part 1: Attitude algorithms," *Journal of guidance, control, and dynamics*, vol. 21, no. 1, pp. 19–28, 1998.
- [22] —, "Strapdown inertial navigation integration algorithm design part 2: Velocity and position algorithms," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 208–221, 1998.
- [23] D. Titterton, J. L. Weston, and J. Weston, *Strapdown inertial navigation technology*. IET, 2004, vol. 17.
- [24] L. Ding, H. Gao, Z. Deng, K. Yoshida, and K. Nagatani, "Slip ratio for lugged wheel of planetary rover in deformable soil: definition and estimation," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 3343–3348.
- [25] M. G. Bekker, "Mechanics of locomotion and lunar surface vehicle concepts," *Sae Transactions*, pp. 549–569, 1964.
- [26] J.-Y. Wong, "Behaviour of soil beneath rigid wheels," *Journal of Agricultural Engineering Research*, vol. 12, no. 4, pp. 257–269, 1967.
- [27] L. Ding, Z. Deng, H. Gao, J. Tao, K. D. Iagnemma, and G. Liu, "Interaction mechanics model for rigid driving wheels of planetary rovers moving on sandy terrain with consideration of multiple physical effects," *Journal of Field Robotics*, vol. 32, no. 6, pp. 827–859, 2015.
- [28] K. Yoshida, T. Watanabe, N. Mizuno, and G. Ishigami, "Terramechanics-based analysis and traction control of a lunar/planetary rover," in *Field and Service Robotics*. Springer, 2003, pp. 225–234.
- [29] H.-J. Jeong, J. H. Lim, and J.-W. Kim, "Development of a coarse lunar soil model using discrete element method," *Journal of the Korean Society for Aeronautical & Space Sciences*, vol. 47, no. 1, pp. 26–34, 2019.
- [30] D. Rodríguez-Martínez, M. Van Winnendael, and K. Yoshida, "High-speed mobility on planetary surfaces: A technical review," *Journal of Field Robotics*, vol. 36, no. 8, pp. 1436–1455, 2019.
- [31] M. Jiang, Y. Dai, L. Cui, and B. Xi, "Experimental and dem analyses on wheel-soil interaction," *Journal of Terramechanics*, vol. 76, pp. 15–28, 2018.
- [32] H. Huang, S. Xu, Z. Meng, J. Li, and J. Zhang, "The sinkage characteristics and prediction of a planetary rover based on a similarity model experiment," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 233, no. 10, pp. 3762–3774, 2019.
- [33] J. Y. Wong, "Predicting the performances of rigid rover wheels on extraterrestrial surfaces based on test results obtained on earth," *Journal of Terramechanics*, vol. 49, no. 1, pp. 49–61, 2012.
- [34] C. Crick, G. Jay, S. Osentoski, B. Pitzer, and O. C. Jenkins, "Rosbridge: Ros for non-ros users," in *Robotics Research*. Springer, 2017, pp. 493–504.