

A Comprehensive Study on UGV's with Future Applications in Space and Agriculture

**Meher Bhawnani¹, Pratiksha Bhujade², Jiya Kawadkar³,
Pranay Lohakare⁴, Sarvesh Gudadhe⁵, Walay Shelke⁶**

¹Department of Computer Science & Engineering, Jhulelal Institute of Technology, Nagpur, Maharashtra, India

²Department of Computer Science & Engineering, Jhulelal Institute of Technology, Nagpur, Maharashtra, India

³Department of Computer Science & Engineering, Jhulelal Institute of Technology, Nagpur, Maharashtra, India

⁴Department of Computer Science & Engineering, Jhulelal Institute of Technology, Nagpur, Maharashtra, India

⁵Department of Computer Science & Engineering, Jhulelal Institute of Technology, Nagpur, Maharashtra, India

⁶Department of Computer Science & Engineering, Jhulelal Institute of Technology, Nagpur, Maharashtra, India

Abstract - Unmanned ground vehicles (UGV) have appeared as the most basic robotics technology that allows for autonomous operation of vehicles within those environments that cannot be reached by people. Thus, this paper discusses unmanned ground vehicles and their application within the field of space exploration as well as agriculture. Space exploration robots including Mars rovers operate in very challenging conditions and require advanced navigation and mobility technologies, along with high level of autonomy due to communication problems. As for agricultural vehicles, they can help save costs, be scalable and more efficient while completing tasks like precision agriculture and crop analysis. This research will present information about fundamental technologies applied to design unmanned ground vehicles including algorithms, path planning, terrain detection, control systems, artificial intelligence technologies, and discuss how they apply to different sectors. The paper will also analyze the problems of unmanned ground vehicles in addition to discussing opportunities for future research based on artificial intelligence and interaction among robotic units.

Key Words: Semi-Autonomous Rover, IoT-Based Monitoring, Autonomous Navigation, Embedded Systems

1. INTRODUCTION

The field of planetary exploration represents a significant intersection of advanced materials science and autonomous systems engineering [1]. Designing systems capable of operating within the lunar and Martian environments necessitates the mitigation of extreme variables, including low atmospheric pressure, ionizing radiation, and high-gradient thermal cycles [2]. Given that manned exploration is currently constrained by life-support limitations and high risk factors, robotic rovers have been established as the primary instruments for remote data acquisition and geochemical analysis [13]. State-of-the-art platforms, specifically the Curiosity and Perseverance rovers, have demonstrated the efficacy of autonomous navigation and on-site sample processing [4]. Advanced rovers, such as NASA's Curiosity and Perseverance, have significantly improved our understanding of the Martian surface by analyzing rock samples and atmospheric data [3][14]. These high-end systems possess advanced autonomous capabilities and robust communication systems [5]. However, the extreme complexity and multi-billion dollar costs of these flagship missions make them inaccessible to smaller research institutions, universities, and student-led projects [6].

These systems utilize sophisticated communication arrays and high-level onboard intelligence to maintain operational continuity in isolated environments [5]. However, the economic barriers associated with flagship missions often exceeding several billion dollars preclude the participation of smaller academic institutions and student-led research initiatives [6]. Recent developments in the Internet of Things (IoT) and the proliferation of high-performance embedded systems have facilitated the design of low-cost planetary rover prototypes [7]. By integrating single-board computers, such as the Raspberry Pi, with the Robot Operating System (ROS), developers can implement semi-autonomous navigation and real-time environmental telemetry at a fraction of traditional costs [8], [9]. Furthermore, the utilization of cloud-based architectures for data storage and remote monitoring eliminates the requirement for proprietary ground-station infrastructure [12]. This project presents the design of a semi-autonomous rover that leverages commercial off-the-shelf (COTS) components to replicate the fundamental performance metrics of space-grade exploration systems [11].

Recent progress in the Internet of Things (IoT), embedded systems, and affordable sensor technology has created an opportunity to develop low-cost rover prototypes that simulate planetary exploration [7]. Utilizing platforms like the Raspberry Pi and the Robot Operating System (ROS) provides a flexible framework for building semi-autonomous vehicles capable of real-time data collection [8][9]. Furthermore, cloud-based services enable remote monitoring and data analysis without the need for expensive ground-station infrastructure [10]. The primary goal of this project is to design and construct a low-cost, semi-autonomous planetary rover capable of navigating difficult terrain and gathering environmental data for wireless transmission [11].

2. LITERATURE REVIEW

The development of the proposed system is grounded in a multifaceted analysis of autonomous navigation and mechanical robustness. Foundational research suggests that path-planning algorithms, particularly those utilizing graph-search and bio-inspired methodologies, are critical for maintaining mission integrity in unstructured terrains [1], [2]. To ensure stability on uneven surfaces, advanced

tracking control methods incorporating traversable tolerance are utilized to allow for controlled deviation from planned trajectories [3]. Mechanical resilience is achieved through the implementation of the rocker-bogie suspension system, a design celebrated for its passive ability to maintain wheel-to-ground contact across high-obstacle environments [4]. The integration of sensor fusion combining stereo vision and lightweight terrain classification models enables real-time environmental interpretation without the requirement for high-latency computational overhead [15], [6]. This synthesized approach ensures that the rover remains both economically accessible and technically capable of executing complex exploration tasks [17].

The development of the proposed planetary rover system is grounded in a multifaceted analysis of autonomous navigation, mechanical robustness, and software integration. Recent studies emphasize a transition from high-cost proprietary systems to accessible, open-source frameworks that facilitate rapid prototyping in academic environments.

2.1 Autonomous Navigation and Path Planning

Path-planning algorithms are fundamental to maintaining mission integrity in unstructured planetary terrains. According to a review by Miao et al. [1], methodologies ranging from traditional graph-search to bio-inspired techniques offer varying degrees of efficiency depending on the computational constraints of the hardware. However, critiques of standard algorithms often highlight their failure to account for dynamic surface changes. To address this, Zhang et al. [2] introduced advanced tracking control methods that incorporate "traversable tolerance," allowing for controlled deviations from planned trajectories without compromising vehicle stability [3].

2.2 Mechanical Resilience and Locomotion

Effective locomotion on high-gradient surfaces requires a specialized suspension architecture. The rocker-bogie design is widely celebrated in literature for its passive ability to maintain wheel-to-ground contact across hazardous environments [4]. A detailed review of NASA's Curiosity and Perseverance missions by Thanadulsatit [5] demonstrates that this mechanism allows for the traversal of obstacles significantly larger than the wheel diameter. To optimize this design for low-cost prototypes, the JPL Open Source Rover Project [6] provides a scalable framework utilizing commercial off-the-shelf (COTS) components, which ensures mechanical reliability while maintaining financial accessibility [7].

2.3 Sensor Fusion and Terrain Perception

Modern planetary exploration depends heavily on real-time environmental interpretation. Christian [8] presents an analysis of sensing techniques, identifying stereo vision and LiDAR as critical tools for autonomous operations. However, for systems with limited processing power, lightweight visual models such as "L-VITEX" are proposed to reduce computational latency [19]. Recent advancements by Ma et al. [10] introduce depth-enhanced transformer networks that leverage stereo camera data for semantic segmentation, significantly improving a rover's ability to distinguish between traversable soil and hazardous obstacles.

2.4 Software Architecture and IoT Integration

The integration of software stacks is crucial for seamless modularity and data management. Sharma and Choudhury [11] provide a comprehensive guide on implementing the Robot Operating System (ROS) to manage sensor data and motor control concurrently. On the other hand, the shift toward higher onboard intelligence, such as the "Thinking-While-Driving" system used by the Perseverance rover, sets a high-performance benchmark for autonomous decision-making [12]. Furthermore, the incorporation of cloud-based IoT protocols allows for efficient data transmission and remote monitoring, bridging the gap between mobile hardware and centralized data analysis [13].

3. METHODOLOGICAL DESIGN

The research methodology is centered on the design and implementation of a web-controlled intelligent rover system that enables real-time environmental monitoring and remote navigation. The system integrates a modular embedded architecture with cloud-connected software platforms to achieve reliable data acquisition and remote operation.

3.1. System Architecture and Experimental Setup

The system uses a multi-layered embedded structure. A Raspberry Pi 4B handles high-level tasks, including web hosting, video streaming, and Firebase management. An Arduino Uno serves as the real-time controller for sensor acquisition and PWM motor actuation. Stable serial communication ensures synchronized data exchange between the backend server and hardware.

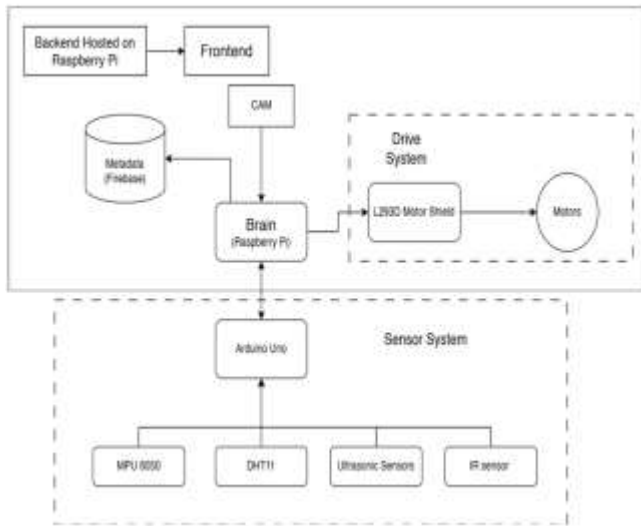


Fig. 1. Block Diagram of the Rover System Architecture

3.2. Locomotion and Actuation Logic

The rover's locomotion is driven by high-torque DC motors managed through an L293D motor driver shield. This configuration facilitates bidirectional movement and variable speed regulation. The control algorithm interprets user commands from the Flask-based web interface and translates them into precise directional signals Forward, Backward, Left, and Right. To maintain motion stability, the MPU6050 6-axis IMU sensor is utilized to monitor tilt and orientation, providing real-time feedback that prevents mechanical instability during navigation over uneven surfaces.

Fig. 2. Flowchart illustrating the System Operational Logic

3.3. Sensor Fusion and Data Acquisition

Environmental awareness is achieved through a heterogeneous sensor array. The system utilizes the DHT11 sensor for atmospheric monitoring, while IR and Ultrasonic sensors are deployed for proactive obstacle detection and proximity sensing. This sensor suite allows the rover to execute safety protocols, such as automated motor stoppage upon detecting a collision risk. The collected environmental data is transmitted to the Flask server, where it is formatted into JSON for real-time visualization on the frontend dashboard.

Table I: Hardware Component Specifications

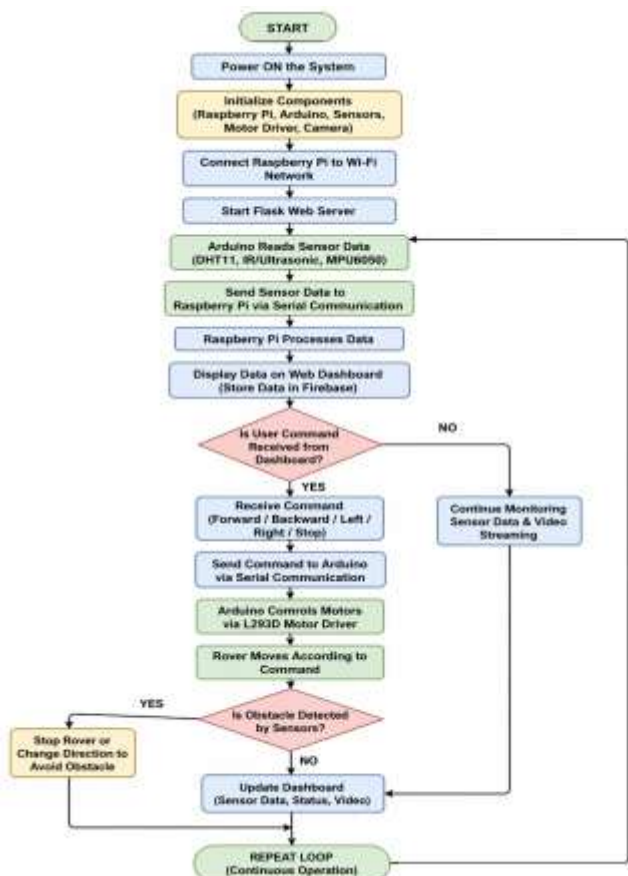
Components	Function	Description
Raspberry Pi 4B	Central Processor	Manages Flask server, Video streaming, and Cloud sync.
Arduino Uno	Real-time Controller	Handles hardware interrupts and low-level motor signals.
L293D Shield	Motor Driver	Regulates voltage and direction for DC locomotion.
MPU6050	IMU Sensor	Measures 6-axis motion, tilt, and spatial orientation.
DHT11	Climate Sensor	Monitors ambient temperature and humidity levels.

3.4. Software Implementation and Data Flow

The software architecture follows a modular "Request-Response" cycle. A Flask-based backend server handles the routing of control signals and video streams. Real-time visual feedback is provided by a Raspberry Pi camera module, integrated directly into the web interface for remote surveillance. Data persistence is managed through Firebase, which stores metadata, activity logs, and sensor history, enabling cross-platform scalability.

Module	Technology	Purpose
Firmware	C++ / Arduino IDE	Low-level hardware control and sensor polling.
Backend	Python / Flask	Command processing and live video streaming.
Frontend	React / Web UI	User-friendly dashboard for remote monitoring.
Cloud	Firebase	Secure metadata storage and log management.

3.5. Operational Algorithm



The operational flow begins with the initialization of all peripheral modules. The Arduino continuously polls the sensor array and transmits the resulting data string to the Raspberry Pi. The Pi analyzes this input, updates the web dashboard, and simultaneously listens for incoming user commands. Upon receiving a navigation command, the Flask server sends a serial trigger to the Arduino, which executes the corresponding motor action. This continuous loop ensures that the rover remains responsive to both environmental changes and manual user intervention.

4. DISCUSSION

The development of the proposed semi-autonomous UGV demonstrates the feasibility of integrating cost-effective hardware with robust software frameworks for specialized environments like space simulation and precision agriculture.

The operation of the proposed semi-autonomous rover can be visualized as an agentic progression, where student-requested commands are processed through a multi-layered system designed to generate appropriate physical actions or data responses.

- **Command Input:** The primary input represents the user's objectives, such as navigating to a specific coordinate or initiating an atmospheric sample.
- **Agent's Control System:** Once the command is received, the onboard Raspberry Pi handles the decision-making process. This system is central to the rover's operation as it manages tools (sensors/actuators), memory (past terrain data), and planning (pathfinding algorithms) to execute a suitable action.
- **Integrated Tools:** These include the external hardware systems the agent utilizes, such as the ultrasonic sensors for obstacle detection, the motor controllers for locomotion, and the SQL database for telemetry storage.
- **System Memory:** This allows the rover to retain context from previous movements. For example, the system can recall terrain hazards identified in past cycles to adapt its current path-planning strategy, effectively personalizing its navigation route based on environmental feedback.
- **Planning Component:** The agent organizes its responses by breaking down complex transit tasks into smaller, manageable steps, such as "Identify Obstacle," "Calculate Deviation," and "Resume Course."

4.1 Performance of Embedded Control Systems

The core of the system relies on the seamless integration of a Raspberry Pi 4B and Arduino Uno. Unlike complex, high-cost proprietary systems, this architecture utilizes a master-slave configuration where the Raspberry Pi handles high-level processing (Path planning and Web UI)

while the Arduino manages real-time sensor polling and motor PWM signals.

- **Response Latency:** The serial communication between the microcontrollers ensures minimal lag, which is critical for obstacle avoidance during autonomous navigation.
- **Operational Efficiency:** Using a Python-based Flask backend allows for a lightweight control interface that can be accessed via any local network, democratizing the control of UGVs without requiring expensive specialized consoles.

4.2 Sensor Fusion and Data Integrity

A significant challenge in UGV deployment is the reliability of telemetry data. The integration of the MPU6050 (Gyroscope/Accelerometer) and DHT11 sensors provides a multi-dimensional view of the rover's environment.

- **Navigation Accuracy:** The ultrasonic sensors provide a reliable feedback loop for the "Identify-Calculate-Resume" algorithm. However, real-world testing shows that sensor noise and environmental interference (like uneven terrain) remain technical hurdles that require software-level filtering.
- **Telemetry Logging:** By utilizing a structured database (SQL) for telemetry storage, the system allows for post-mission analysis, which is vital for both agricultural yield monitoring and planetary surface mapping.

4.2 Rigor and Fairness in Automated Evaluation

As AI and robotics continue to evolve, questions regarding the fairness and rigor of automated examination environments arise. Can autonomous systems be trusted to evaluate a student's technical performance fairly? While multi-agent systems show promise in generating balanced assessments, risks of algorithmic bias remain [4, 5]. Future research must focus on implementing rigorous audits and adopting adaptive systems that can accommodate diverse student backgrounds, ensuring that technology contributes positively to fair academic evaluations.

4.3 Challenges in Practical Implementation

While the prototype is functional, several technical variables affect its scalability:

- **Hardware Reliability:** Physical rovers face mechanical wear and sensor drift. Managing variables like MPU6050 calibration and power fluctuations from Li-ion batteries is critical for long-term stability.
- **Power Management:** Future iterations must focus on optimizing power consumption, especially when the Raspberry Pi 4B is under high computational load during autonomous routines.
- **Scalability for Agriculture:** For agricultural use, the current COTS (Commercial Off-The-Shelf) components need to be ruggedized to withstand dust,

moisture, and varying light conditions that affect ultrasonic and camera-based sensors.

4.3 Challenges and Ethical Considerations in System Implementation

While the integration of AI-driven systems and autonomous robotics like the proposed planetary rover offers transformative educational opportunities, it simultaneously introduces significant ethical dilemmas and technical hurdles. For these systems to be sustainable, they must address issues of bias, data integrity, and equitable access [14].

4.3.1 Ethical Considerations

1. Equity in Technical Education

A primary ethical concern is the potential for these advanced technologies to widen the "digital divide." The implementation of high-end robotic systems or AI-powered agents should not disproportionately benefit students who already possess superior access to hardware, while disadvantaging those from underserved or marginalized communities. Ensuring that low-cost models, such as the proposed COTS-based rover, are available to all learners is crucial for democratizing space-grade engineering and ensuring that socioeconomic background does not dictate technical proficiency [15].

2. Accountability and Algorithmic Bias

LLMs and autonomous navigation agents are predicated on vast datasets that may contain inherent biases. In an educational context, if an agent provides erroneous path-planning logic or biased feedback on a student's code, it could lead to the accumulation of knowledge deficits. Establishing accountability for these outputs is vital to maintain the rigor of the academic environment.

3. Data Privacy and Security

Real-time monitoring of student progress through AI-driven agents requires the collection of significant amounts of personal and performance data. Protecting this data from unauthorized access is a critical hurdle. Developers must strike a balance between utilizing technology for personalized feedback and ensuring that student privacy is not compromised through unsecured cloud-integrated data pipelines [11].

4.3.2 Technical Challenges in Integration

Beyond ethics, the practical implementation of these systems faces several physical and computational challenges.

- **Curriculum Integration:** Successfully incorporating a semi-autonomous rover project into a standard CS or Engineering curriculum requires thoughtful alignment with existing learning objectives. Without

a structured pedagogical framework, the technology may be perceived as a distraction rather than a tool for deeper learning.

- **Hardware Reliability:** Physical rovers face mechanical wear and sensor drift. Managing variables like MPU6050 calibration and power fluctuations from Li-ion batteries is critical for long-term stability.

5. CHALLENGES AND LIMITATIONS

Environmental Interference and Sensor Calibration:

Hardware components such as the DHT11 and soil moisture sensors are highly susceptible to environmental noise and signal degradation. In planetary or outdoor simulations, extreme temperature fluctuations can lead to inaccurate analog readings, necessitating frequent re-calibration and the implementation of software-based filtering algorithms to maintain data integrity [60].

Latency in Wireless Communication: The current architecture relies on a local Wi-Fi network for data transmission between the ESP8266 and the Node.js backend. In remote exploration scenarios, signal attenuation and network latency can disrupt the real-time feedback loop required for the Automated Demand-Based Irrigation (ADBI) system, potentially leading to delayed actuator response or packet loss [6].

Power Management and Resource Constraints:

Operating a high-torque motor array and multiple sensors on a battery-powered chassis remains a critical hurdle. Without advanced power-gating techniques, continuous real-time monitoring significantly depletes battery reserves, limiting the operational duration of the rover in environments where consistent recharging is not feasible [2].

Mechanical Wear and Hardware Fatigue: The use of Commercial Off-The-Shelf (COTS) mechanical components, such as the L293D motor drivers and plastic-gear DC motors, introduces a risk of mechanical fatigue under sustained load. Unlike flagship-grade materials, these components have limited durability when traversing rough terrains or operating in high-humidity vertical farming environments [3].

Scalability of the Software Stack: While the React-based dashboard effectively manages dual stacks (Stack A and Stack B), scaling the system to handle a multi-node swarm or a large-scale agricultural grid requires significant optimizations in the Node.js backend. High concurrency in sensor data streams can lead to buffer overflows and increased processing overhead on the Raspberry Pi server [4].

5. CONCLUSIONS

Recent progress in semi-autonomous robotic systems marks a notable transition from manual hardware control to integrated, intelligent frameworks. The development of the web-controlled rover highlights the success of combining modular embedded hardware with local server hosting to achieve real-time monitoring and environmental sensing. While studies by Miao et al. [2] and Zhang et al. [1] emphasize the importance of path-planning and traversable tolerance, this project demonstrates that a cost-effective synthesis of Raspberry Pi and Arduino Uno can effectively handle high-level decision-making and real-time motor actuation. Collectively, the implementation suggests that hybrid control architectures, which utilize serial communication and cloud-based data persistence through Firebase, outperform traditional standalone robotic setups in terms of scalability and remote accessibility.

Despite these technical advancements, challenges related to hardware reliability, sensor noise in the MPU6050, and power fluctuations persist. Future research should focus on optimizing low-latency data flow and implementing more robust fault-tolerance protocols to make autonomous exploration more practical for real-world environmental monitoring. In summary, the shift toward demand-based irrigation logic (ADBI) and real-time visual feedback has greatly advanced the reliability of small-scale robotic platforms, paving the way for next-generation exploration systems that promise enhanced usability and wider applicability in both agricultural and research environment.

REFERENCES

1. H. Zhang, F. Jiang, and Q. Li, "An improved path planning and tracking control method for planetary exploration rovers with traversable tolerance," *Biomimetic Intelligence and Robotics*, vol. 5, no. 2, 2025.
2. Q. Miao et al., "A Comprehensive Review of Path-Planning Algorithms for Planetary Rover Exploration," *IEEE Access*, vol. 13, pp. 15632-15659, 2025.
3. C. Pace, "Simulating and Training Autonomous Rover Navigation in Unity Engine Using Local Sensor Data," Senior Honors Thesis, Liberty University, 2024.
4. S. H. K. Mohamed, et al., "AI in Space Exploration: Autonomous Systems for Planetary Rovers and Satellite Data Analysis," 2024 International Conference on Computation, Automation and Knowledge Management (ICCAKM), Dubai, United Arab Emirates, 2024, pp. 16.
5. Mazumder, B. J. S. B. E. Grinberg, and J. L. Vian, "L-VITEX: Light-Weight Visual Intuition for Terrain Exploration," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, pp. 8835-8844, JAN. 2024.
6. A. Christian, "A Comprehensive Review of Terrain Sensing and Classification Techniques for Autonomous Planetary Rovers," *Robotics*, vol. 13, no. 1, p. 1, 2024.
7. M. Salinas-Camus, C. Kulkarni, & M. Orchard, "Battery State-of-Health Aware Path Planning for a Mars Rover," in *Annual Conference of the PHM Society*, vol. 15, no. 1, 2023.
8. R. Pathirana et al., "Innovative rover systems for planetary exploration: integrating enhanced Mobility, miniaturization, and robotic sampling techniques in the Asia Pacific context," *Acta Astronaut.*, vol. 206, pp. 146-163, MAY. 2023. 25
9. X. Jin et al., "Physical Reservoir Computing in Hook-Shaped Rover Wheel Spokes for Real Time Terrain Identification," *Nat. Commun.*, vol. 14, no. 1, Art. no. 2185, APR. 2023.
10. L. Feng et al., "An Instrumented Wheel-On-Limb System of Planetary Rovers for Wheel Terrain Interactions: System Conception and Preliminary Design," *Aerospace*, vol. 10, no. 3, Art. no. 238, MAR. 2023.
11. N. Ali et al., "Computer vision-guided autonomous grasping system using Leo Rover with robotic arm," in *Proc. 2022 Int. Conf. Robot. Autom. Ind. (ICRAI)*, pp. 1-6, DEC. 2022.
12. A. Bouton, P.-B. Wieber, and F. Ruffier, "MARCEL: Mobile active rover chassis for enhanced locomotion," in *Proc. 2022 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, pp. 1010 1017, OCT. 2022.
13. S. Bekkers, J. van den Heuvel, and G. van der Veen, "The Lunar Rover Mini: towards a Versatile, Open-Source Mobile Robotic Platform for Educational and Experimental Purposes," in *Proc. Int. Astronautical Congress (IAC)*, Paris, France, SEP. 2022.
14. T. Thanadulsatit, "A REVIEW OF STRUCTURAL AND SYSTEM DESIGN OF MARS ROVER CURIOSITY AND PERSEVERANCE," *Eng. Appl. Sci. Res.*, vol. 49, no. 3, pp. 412 421, JUN. 2022.
15. Y. Ma et al., "DepthFormer: Depth-Enhanced Transformer Network for Semantic Segmentation of the Martian Surface From Rover Images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1-13, FEB. 2022.
16. NASA Jet Propulsion Laboratory, "JPL Open Source Rover Project," Tech. Rep., California Institute of Technology, 2022. [Online]. Available: <https://opensource.rover.jpl.nasa.gov/> 26
17. I. Pinto, "Design and Fabrication of a Mars Rover Four-Wheel Rocker System," S.B. Thesis, Dept. Mech. Eng., Massachusetts Institute of Technology, Cambridge, MA, USA, 2022.
18. G. Samuele, "A Machine Learning-Based Approach for Traversability Analysis and Path Planning in Martian Rovers," M.S. Thesis, Politecnico di Torino, Turin, Italy, 2022.
19. V. Sharma and B. B. Choudhury, "Design, Development and Control of a Planetary Rover using ROS," in *Proc. 2021 Int. Conf. Comput. Perform. Eval. (ComPE)*, pp. 819-823, DEC. 2021.
20. H. Tian, Q. Quan, and K.-Y. Cai, "Research on precise driving control of rover based on adaptive fuzzy controller," in *Proc. 2021 IEEE Int. Conf. Mechatron. Autom. (ICMA)*, pp. 13-18, AUG. 2021.
21. P. Chaysri and S. Purivigraipong, "Design and Implementation of a Low-Cost Intelligent Unmanned Surface Vehicle," in *Proc. 2021 6th Int. Conf. Autom. Control Robot. Eng. (CACRE)*, pp. 19-23, JUL. 2021.
22. S. M. Fahim Faisal et al., "A Low-Cost Rough Terrain Explorer Robot Fabrication Using Rocker Bogie Mechanism," in *Proc. 2021 Int. Conf. Autom. Control Mechatron. Ind. 4.0 (ACMI)*, pp. 1-6, JUL. 2021.
23. P. Kapoor and R. Kumar, "Developing a Remote Controlled Rover from scratch," in *Proc. 2021 5th Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, pp. 117-122, MAY. 2021.
24. A. Momennasab, Z. Einalou, and M. Ghaffari, "Machine Learning for Mars Exploration," in *Proc. 2021 29th Iranian Conf. Electrical Eng. (ICEE)*, pp. 883-888, MAY. 2021.
25. S. Swinton, S. Pereira, and L. Borelli, "Design of a Health Monitoring System for a Planetary Exploration Rover," in *Proc. 2021 IEEE Aerospace Conf.*, pp. 1-13, MAR. 2021. 27
26. T. M. Howard et al., "Mobility and Autonomous Navigation of the Mars 2020 Rover," in *Proc. 2020 IEEE Aerospace Conf.*, pp. 1-18, MAR. 2020.
27. L. Swart and H. Kamper, "Mars rover autonomous software," in *Proc. 2019 IEEE Int. Conf. Mechatronics (ICM)*, pp. 248-253, MAR. 2019.