Development of a Robotic System for the Introduction of Space Robotics Concepts in Basic Education

Lucas Pereira Lima
IFRN – Campus Parnamirim
Parnamirim, Brazil
pereira.lima1@escolar.ifrn.edu.br

Ryan Lucas Soares Laranjeira
IFRN – Campus Parnamirim
Parnamirim, Brazil
ryan.lucas@escolar.ifrn.edu.br

Cecilia Helena Gurgel de Andrade
IFRN – Campus Parnamirim
Parnamirim, Brazil
gurgel.andrade@escolar.ifrn.edu.br

Maria Gabriella Monteiro Alves
IFRN – Campus Parnamirim
Parnamirim, Brazil
gabriella.monteiro@escolar.ifrn.edu.br

Fabio Augusto Procopio de Paiva
IFRN – Campus Parnamirim
Parnamirim, Brazil
fabio.procopio@escolar.ifrn.edu.br

Credson Isaac Lopes dos Santos
IFRN – Campus Parnamirim
Parnamirim, Brazil
credson.isaac@ifrn.edu.br

Givanaldo Rocha de Souza
IFRN – Campus Parnamirim
Parnamirim, Brazil
givanaldo.rocha@ifrn.edu.br

Filipe Campos de Alcântara Lins
IFRN – Campus Parnamirim
Parnamirim, Brazil
filipe.lins@ifrn.edu.br

***Abstract*—This work presents the Zênite system, a robotic platform based on global vision developed to support educational activities within the Science, Technology, Engineering, Arts and Mathematics (STEAM) methodology, with emphasis on space-related applications carried out at the Centro Vocacional Tecnológico Espacial (CVT-E). The project is used as a practical tool in workshops aimed at elementary and secondary education, enabling the exploration of robotics and engineering concepts through missions inspired by space rovers. Zênite is structured on a modular architecture implemented in Robot Operating System 2 (ROS 2), composed of nodes responsible for image acquisition, environment calibration, robot localization, user interaction, motion control, and command transmission. Robot navigation is performed using a ceiling-mounted camera that provides a global view of the environment, allowing target position definition and trajectory execution without the need for onboard sensors. The robot has a nonholonomic configuration with differential drive and performs controlled displacements based on commands generated by the system. In the educational context, students actively participate in the robot assembly, system calibration, and definition of goal points for the robot to navigate to in an environment simulating the surface of Mars. These activities enable the practical application of concepts such as coordinate systems, robot kinematics, perception, and control, fostering active learning and the development of logical reasoning skills. Initial results indicate that the system is functional and suitable for educational purposes, demonstrating stable behavior during the proposed missions. The system shows potential as a didactic tool for teaching robotics in an interdisciplinary context oriented toward space exploration.**

***Index Terms*—Educational Robotics, STEAM Education, Space Robotics, Computer Vision, Global Vision, ROS 2**

979-8-3195-1715-9/26/$31.00 ©2026 IEEE

I. Introduction

The project, named Zênite, originated from an educational initiative aimed at training students from public and private schools through the STEAM [1] teaching methodology applied to the space domain, known as the CVT-E [2]. The program promotes practical and laboratory-based activities, such as rover workshops, which aim not only to stimulate interest in science and engineering, but also to provide experiences that faithfully and contextually reproduce concepts of astronomy and astronautics. In this context, the use of robots goes beyond mere "maker" practice, treating the incorporation of robotics tools as educational instruments that reflect real procedures, constraints, and challenges of space engineering.

In general, Zênite consists of a robot control system based on zenithal vision, designed for teaching elementary and secondary school students. The adopted didactic approach is structured in progressive stages, in which students participate in both the use of the software and the assembly of the robot, allowing traditionally complex astronautics concepts to be assimilated in a more accessible and intuitive manner.

From a technical perspective, the physical structure of the system consists of a camera installed on the ceiling of the environment, responsible for capturing a global view of the scene. From this single view, the system identifies the robot's position, defines destinations, and determines safe and efficient trajectories. The project architecture was developed using the ROS 2 framework, adopting a modular design that allows a clear separation of functionalities, such as detection and tracking, camera calibration, path planning, and command transmission. This organization not only favors system scalability and maintenance, but also contributes to students' understanding of its internal operation.

Within the practical activities, students are initially introduced to theoretical fundamentals related to mission planning with space rovers. Subsequently, they assemble the robot and are guided through experimental stages that involve system calibration in a control room and the positioning of the robot in a physical environment that simulates the surface of Mars, monitored by a zenithal camera. After the configuration, students use the software to define the robot's destinations, represented by craters in the simulated environment, and execute a displacement mission for a hypothetical data collection task, integrating robotics concepts from a space exploration perspective.

Based on this introduction, this article presents in detail the architecture and operation of the Zênite system, describing the interaction dynamics between the ROS 2 nodes, their functions, and the strategies adopted for the implementation of the interface, trajectory control, rover tracking, data transmission, among other aspects. Finally, the obtained results and perspectives for future work are discussed.

II. Related Works

The use of active learning approaches has become increasingly popular in recent years. Special emphasis has been placed on the use of robotics concepts in teaching subjects at the elementary and secondary school levels. More recently, the STEAM approach has gained prominence for enabling the integration of different areas of knowledge into teaching and learning activities. Several studies highlight the advantages of these methodologies in the educational process, emphasizing improvements in student engagement, motivation, and the development of interdisciplinary skills [3].

Over the past decades, governments and space agencies have increasingly invested in the training and orientation of students in topics related to astronautics and astronomy. The STEAM methodology has been widely applied in this context. An example is presented by [4], who describe a project funded by the U.S. government in which a classroom was transformed to simulate a Martian environment. In this space, students operate replicas of rovers sent to Mars and are challenged to carry out robot control missions through programming. As a result, the authors reported a significant increase in student engagement.

Complementarily, [5] developed a robotic system designed for use in museums and situations involving quick interactions between users and robots. In this case, the robotic platform is equipped with an onboard monocular camera that transmits images to a human-machine interface. Visitors can select, within the image, the point where they want the robot to move. The system then responds to the user's request, guiding the robot to the desired position while autonomously avoiding obstacles.

In Brazil, several studies have been developed focusing on the creation of robots for educational purposes. In [6], through a literature review, researchers presented different platforms designed for teaching. However, most of these initiatives were aimed at robotics education at the academic level.

This work, in contrast, proposes the development of a robotic system designed for elementary school students, allowing them to assemble the robot in a hands-on activity and learn fundamental concepts of electronics. After assembly, students can test the robot in an arena designed to represent the surface of Mars. Interaction takes place through a human-machine interface that displays an aerial view of the arena, simulating a control room in which students must plan and monitor the robot's exploration.

This approach enables the introduction of concepts of communication and autonomy in robotic systems in a practical school context, bringing students closer to the technologies used in space missions.

III. System Architecture

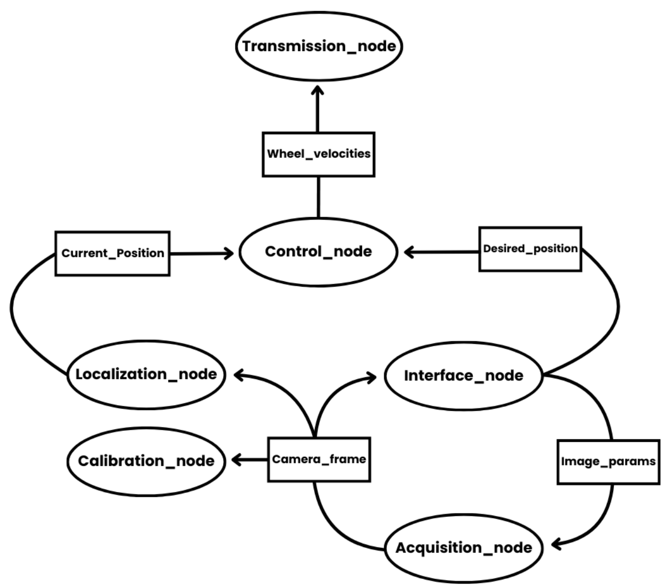

Fig. 1. System architecture overview.

The Zênite system was designed with a modular architecture based on the ROS 2 framework, aiming at flexibility, scalability, and ease of maintenance. The architecture adopts a distributed approach, in which different nodes are responsible for specific tasks, such as robot tracking, control, and transmission.

Fig. 1 illustrates the overall system architecture and the information flow among its main components. A camera installed in a zenithal position provides images to the mapping, tracking, and interface nodes through the acquisition node. Subsequently, the calibration node is responsible for recognizing the field and storing this information in a .yaml file, which is later used by the localization node, responsible for detecting and tracking the robot in the environment. The display of the camera feed, the adjustment of its parameters, and the definition of the rover's destination by the user are handled by the interface node. After these processes, the position information and the reference point are forwarded to the control node, which performs the robot's kinematic reasoning using integral and derivative calculations. Finally, the transmission node establishes Bluetooth communication between the system and the rover's HC-06 transmission module, converting the numerical velocity values received from the control node into PWM signals and then into bytes, allowing the Arduino to properly distribute the voltage levels to the left and right wheels.

Communication among the nodes is carried out through ROS 2 topics, enabling asynchronous data exchange and decoupling between modules. This organization facilitates both individual and integrated testing, in addition to allowing future system extensions, such as the inclusion of new sensors or pose estimation algorithms.

The choice of a modular architecture proved to be suitable for the educational and experimental context of the project, as it allows each component to be developed, tested, and adjusted independently, without compromising the overall system operation.

A. Acquisition Node

The acquisition_node is responsible for establishing direct communication with the project's camera. It opens the video device, continuously captures images, and publishes these frames within ROS 2, allowing other nodes to access the camera data in real time. In addition, it receives adjustments such as brightness, saturation, and hue from the interface_node and applies these settings directly to the camera, ensuring that the displayed image always matches the user's configuration. To achieve this, the node disables the automatic exposure mode and switches to full manual control. The node operates at a rate of approximately 30 frames per second, converting each capture into the appropriate ROS message format and publishing it on the corresponding topic. In this way, it acts as a bridge between the physical camera and the rest of the robotic system, providing updated and configurable images.

B. Calibration Node

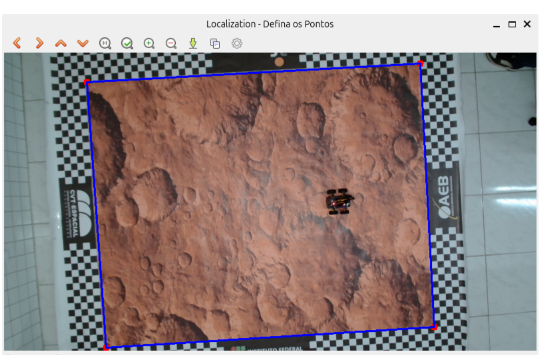

Fig. 2. Calibration node working.

The calibration_node is responsible for calibrating the robot's operating area, enabling the camera to convert image points into real-world ground coordinates through homography calculation. It continuously receives frames from the camera and displays them in a window, allowing the operator to select four points in the scene, typically the corners of the robot's workspace, as illustrated in Fig. 2. Based on these coordinates, the node records the image positions and uses them to perform matrix calculations that associate image points with real measurements of the environment, forming a rectangle with known dimensions. Using these point pairs, the node computes the transformation required to convert pixel coordinates into meters and stores this configuration in a .yaml file for use by other nodes. In this way, the calibration_node establishes the foundation for the robot to understand its position and navigate accurately.

C. Localization Node

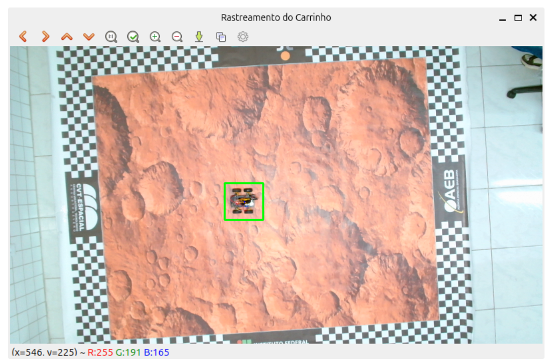

Fig. 3. Localization node working.

The localization_node continuously receives images from the camera and tracks the robot's position in real time using a C++ algorithm based on the Channel and Spatial Reliability Tracking (CSRT) method from OpenCV [7]. Initially, the user selects the robot on the screen through a pixel-based bounding box selection, allowing the tracking function to automatically follow the robot in subsequent frames. The node then draws a rectangle around the robot, records its trajectory, and computes the center of this region, as shown in Fig. 3. This point is converted from pixels to meters using the transformation (homography) obtained by the calibration_node, generating the robot's real-world position in the environment, which is then sent to the control_node. In this way, the localization_node combines visual tracking with physical coordinate conversion, enabling continuous and straightforward monitoring of the robot's motion.

D. Interface Node

The interface_node acts as the visual and interactive component of the system, as illustrated in Fig. 4. It receives the camera images in real time and displays them on the screen using OpenCV. From this visualization, the user can click on any point in the image, and the node automatically converts

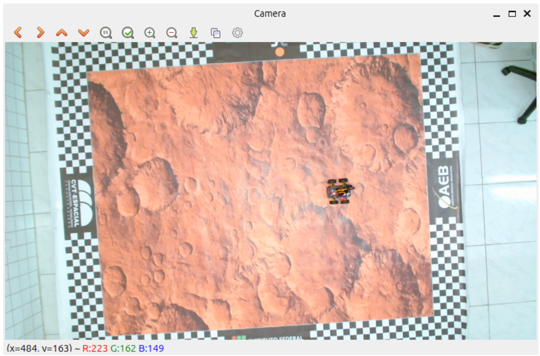

Fig. 4. Interface node working.

this pixel into real-world coordinates using a transformation previously calibrated by the calibration_node. This converted point is then sent to the control_node as the new desired target position for the robot. In addition, the node provides sliders to adjust image brightness, saturation, and hue. Whenever these sliders are changed, the new values are sent to the acquisition_node, which applies the adjustments directly to the camera. In this way, the interface_node serves as a bridge between the user and the robot, allowing destination selection through simple clicks and intuitive image adjustments.

E. Control Node

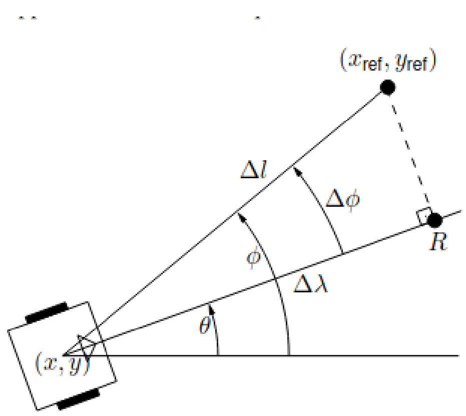

Fig. 5. Robot turning movement overview.

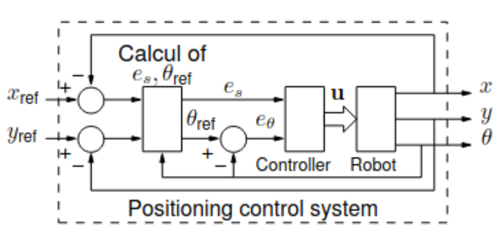

Fig. 6. Robot calculating overview.

The *control_node* is responsible for computing the rover's motion commands, based on the mathematical model described in [8], as illustrated in Fig. 6. It continuously receives the current position of the robot provided by the *localization_node* and the desired position defined by the *interface_node*. When a target point is provided, the node calculates the motion required for the robot to reach that position, determining both the orientation adjustment and the forward displacement.

The robot is modeled as a non-holonomic differential-drive mobile robot, whose posture in the Cartesian plane can be described by the state vector:

$$q = [x, y, \theta]^T$$

where x and y represent the robot position in the plane and θ represents its orientation with respect to the global reference frame.

The kinematic model of the robot can be described by the following equations:

$$\dot{x} = v\cos(\theta)$$

$$\dot{y} = v\sin(\theta)$$

$$\dot{\theta} = \omega$$

where v represents the linear velocity of the robot and ω represents its angular velocity.

Given a desired position (x_{ref}, y_{ref}), the reference orientation used to guide the robot toward the target is computed as:

$$\theta_{ref} = \arctan 2(y_{ref} - y,\ x_{ref} - x)$$

The angular error is then defined as:

$$e_\theta = \theta_{ref} - \theta$$

The distance between the robot and the target point is given by:

$$\Delta l = \sqrt{(x_{ref} - x)^2 + (y_{ref} - y)^2}$$

Following the positioning control strategy, the linear error can be defined as:

$$e_s = \Delta l \cos(e_\theta)$$

This value represents the projection of the distance to the target along the current direction of motion of the robot.

The *control_node* employs proportional–integral controllers to reduce both the linear error and the angular error, generating the linear velocity v and angular velocity ω commands. These commands are continuously updated during the control cycle, allowing the robot to correct its trajectory while approaching the target.

Finally, the linear and angular velocities are converted into the individual velocities of the right and left wheels according to the kinematic model of the differential-drive system:

$$v_r = \frac{2v + \omega L}{2R}$$

$$v_l = \frac{2v - \omega L}{2R}$$

where L represents the distance between the wheels and R is the wheel radius.

The computed velocities are then sent to the *transmission_node*, which converts them into control signals for the robot motors. This process occurs continuously, with the control cycle running several times per second until the robot reaches the goal within an acceptable error tolerance.

F. Transmission Node

The transmission_node is responsible for sending the commands required for the robot's motion to the microcontroller. It establishes a Bluetooth communication, configures the serial port, and waits for the wheel speeds calculated by the control_node. When these values are received, the node converts each wheel speed into a PWM value in the range from 0 to 255 and determines the direction of each motor, using bits to indicate forward or reverse motion. It then sends a three-byte packet containing the direction and power levels of both motors to the microcontroller. In this way, the node acts as a bridge between ROS 2 and the robot's physical hardware, ensuring that the computed velocities are translated into actual motion signals.

G. Robot

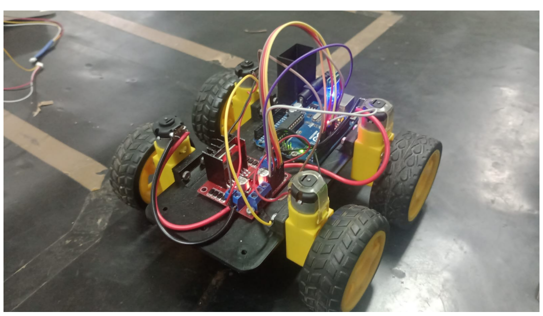

Fig. 7. Robot structure.

The robot has a non-holonomic configuration, being kinematically equivalent to a differential-drive platform, in which the wheels on each side are driven together, as shown in Fig. 7. It continuously receives packets via the HC-06 Bluetooth module containing the direction and speed level for each side of the drivetrain. The microcontroller interprets these three bytes, identifies the correct rotation direction, and applies PWM signals to the two channels of the L298N H-bridge, driving each side of the robot independently and enabling forward motion, reverse motion, and turning maneuvers. This entire process runs in real time, with constant packet reading and immediate application of the speeds, making the robot's motion stable and responsive to the commands sent by the ROS 2 system.

IV. Experimental Results

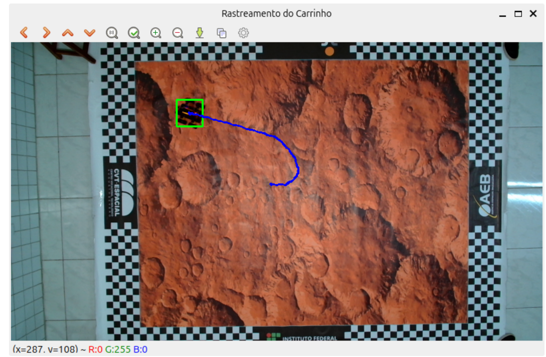

Fig. 8. Example of the robot moving toward the position specified by the user.

Throughout the development of the Zênite Project, as the architecture became more solid, the individual modules went through successive testing cycles, both independently and integrated with other ROS 2 nodes. This incremental approach made it possible to identify unexpected behaviors at early stages, enabling quick interventions and ensuring that, when integrating the complete system, the source of possible issues could be more easily predicted.

In the initial tests, which consisted of sending linear destination commands to the robot, successive adjustments to the control and perception nodes led to positive convergence of results: the robot began moving toward the reference point, reducing its speed as it approached and stopping once it reached the specified coordinate, with a distance error of approximately 5 cm and an average response time of 485 ms.

After further refinements in the control and perception nodes, particularly in the mathematical computation sections of the code, the system demonstrated the ability to perform more complex movements. These improvements resulted in smoother trajectory execution and more stable orientation control, as illustrated in Fig. 8.

V. Conclusion and Future Work

Overall, the Zênite Project showed that the chosen architecture works well for a vision-based robotic system. The organization of the nodes and the communication between perception, control, and actuation allowed the robot to perform the proposed tasks in a consistent manner, even with the current limitations of the system. The obtained results indicate that the project has a solid foundation and is suitable for future improvements. The difficulties observed during development do not invalidate the proposal, but rather highlight aspects that can be improved. Many of these limitations are related to the robot's physical configuration and to the accuracy of the tracking methods, opening room for enhancements in both hardware and software.

As future work, the physical structure of the rover is expected to be replaced by a two-wheel platform, along with the use of an ESP32 microcontroller, which should simplify communication and provide greater processing capability. In addition, improvements to the localization_node algorithms are planned, aiming at more accurate robot tracking and direct estimation of its orientation, without relying on previous positions. This is expected to make the control calculations more stable and reliable.

Additionally, a new node called path_planning will be implemented. This node will receive the current position of the robot, the desired destination, and the location of possible obstacles in the environment. Initially, the implementation is expected to employ the A* algorithm for path planning, due to its efficiency in grid-based environments and its ability to compute optimal trajectories. Based on this information, the node will plan the most appropriate trajectory, aiming to follow the path in an efficient and safe manner, with a focus on obstacle avoidance. This functionality will allow the system to demonstrate, in a way closer to real conditions, the autonomous behavior of a real rover.

Finally, a dedicated structure for camera support will be designed and built to improve the stability of the vision system and to enable easy transportation of the project for demonstrations and events. With these improvements, Zênite is expected to achieve more robust performance and become better suited for autonomous navigation scenarios and practical demonstrations.

ACKNOWLEDGMENTS

The authors would like to express their sincere gratitude to the Brazilian Space Agency (AEB) for its continuous support of initiatives related to space education in Brazil. The Centro Vocacional Tecnológico Espacial (CVT-E) is also acknowledged for providing the environment and resources necessary for the development of this project. Special thanks are extended to the Instituto Federal do Rio Grande do Norte – Campus Parnamirim (IFRN) for its institutional support and encouragement of research and technological development. Finally, the authors acknowledge all the professors, students, and collaborators who contributed directly or indirectly to the conception, development, and implementation of the Zênite project.

REFERENCES

[1] Rodrigues-Silva J, Alsina Á., Conceptualising and framing STEAM education: what is (and what is not) this educational approach?, 2023, Available from: https://doi.org/10.1590/1983-3652.2023.44946.

[2] IFRN – Campus Parnamirim, "CVT Espacial," 2026. [Online]. Available: https://cvt-espacial.ifrn.edu.br/. [Accessed: 02-Mar-2026].

[3] Kalaitzidou M, Pachidis TP. Recent Robots in STEAM Education. Education Sciences. 2023; 13(3):272. https://doi.org/10.3390/educsci13030272

[4] Rieksts, Isaac., Blank, Glenn. (2008). Mars Rovers in Middle School.

[5] Nourbakhsh Illah, Hamner Emily, Ayoob Ellen, Porter Eric, Dunlavey Brian, Bernstein Debra, Crowley Kevin, Lotter Mark, Shelly Skip, Hsiu Thomas, Clancy Daniel. The Personal Exploration Rover: Educational Assessment Of A Robotic Exhibit For Informal Learning Venues. International Journal of Engineering Education, 2006, 22. 777-791.

[6] Amorim, J. S. J. C. C., Silva, Y. N., Canella, A. L. C., Queiroz, G. F. C., Marcato, A. L. M., Neto, A. F. dos S., and Pinto, M. F. (2024). Design of a Mobile Robot for Educational and Research Purposes. In Congresso Brasileiro de Automática (CBA 2024). Sociedade Brasileira de Automática.

[7] Ianni, D. S. (2023). Integrando visão computacional e sistemas embarcados: rastreio de objetos com CSRT e controle automático de câmera pan-tilt usando ESP32 [Universidade Estadual Paulista (Unesp)]. http://hdl.handle.net/11449/239747

[8] Vieira, F. C.; Medeiros, A. A. D.; Alsina, P. J.; Araújo Jr., A. P. (2004). Position and Orientation Control of a Two-Wheeled Differentially Driven Nonholonomic Mobile Robot. Federal University of Rio Grande do Norte (UFRN), Department of Computer Engineering and Automation.