Simpósio de Integração Acadêmica

Simpósio de Integração Acadêmica

"Das Montanhas de Minas ao Oceano: Os Caminhos da Ciência para um Futuro Sustentável"

SIA UFV 2025

UFV Universidade Federal de Viçosa

Blockly and ROS: A Visual Programming Interface for Robot Control

Guilherme S. F. Pinel, Alexandre S. Brandão e Rejane W. S. de C. Faria

ODS4

Pesquisa

Introduction

This work presents the development of a block-based programming platform integrated with the Robot Operating System (ROS) for mobile robot control. The proposed solution combines a visual programming interface based on Blockly and a Python backend, enabling direct communication with the Pioneer 3DX robot. The platform allows defining movement and rotation commands, as well as executing customized programs. Experimental validation was conducted by programming the robot to draw geometric figures (square, equilateral triangle, and lemniscate), demonstrating the platform's effectiveness in executing movement commands with precision and stability. This solution is particularly suitable for educational contexts and robotics experiments, offering an accessible, modular, and adaptable approach to robot programming.

Objectives

Develop and validate a block programming platform integrated with ROS1 for controlling mobile robots (Pioneer 3-DX), with modular architecture.

- Integrate Blockly to ROS for Pioneer 3-DX navigation control.
- Validate the platform by running geometric trajectories as a case study (educational proposal).
- Evaluate the platform's accuracy and usability, highlighting its potential for application in an educational environment.

Methodology

Architecture (Fig. 1):

- Modular flow in 5 steps: (1) user assembles program in blocks; (2) code in JSON goes to backend; (3) backend publishes in ROS topics; (4) robot executes; (5) feedback returns to frontend.

Backend (Flask + ROS1):

- Converts Blockly commands (JSON) into ROS publications; functions: move_forward(distance), turn(angle), stop();

Middleware ROS (Pioneer 3-DX):

- Topics: /RosAria/cmd_vel (linear speed/ang.) and /RosAria/pose (odometry);

Experimental set-up (Fig. 2):

- Windows 10 notebook (Blockly on Ubuntu 20.04.6/WSL) connected to ROS Master on Ubuntu 20.04.6 PC (ROS1 Melodic); Pioneer 3DX on Raspberry Pi 3 (Ubuntu 18.04, ROS1 Melodic).

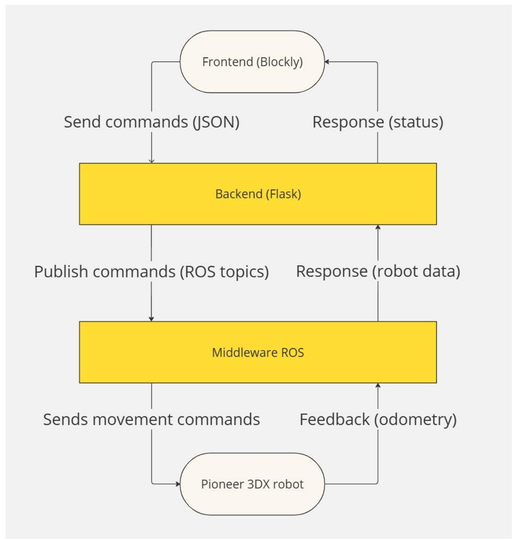


Fig. 1: Interaction diagram of the platform with the robot.

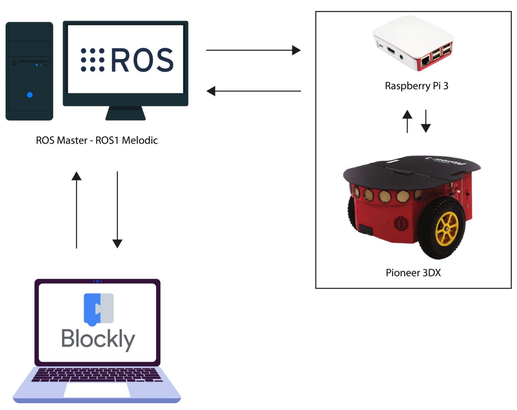
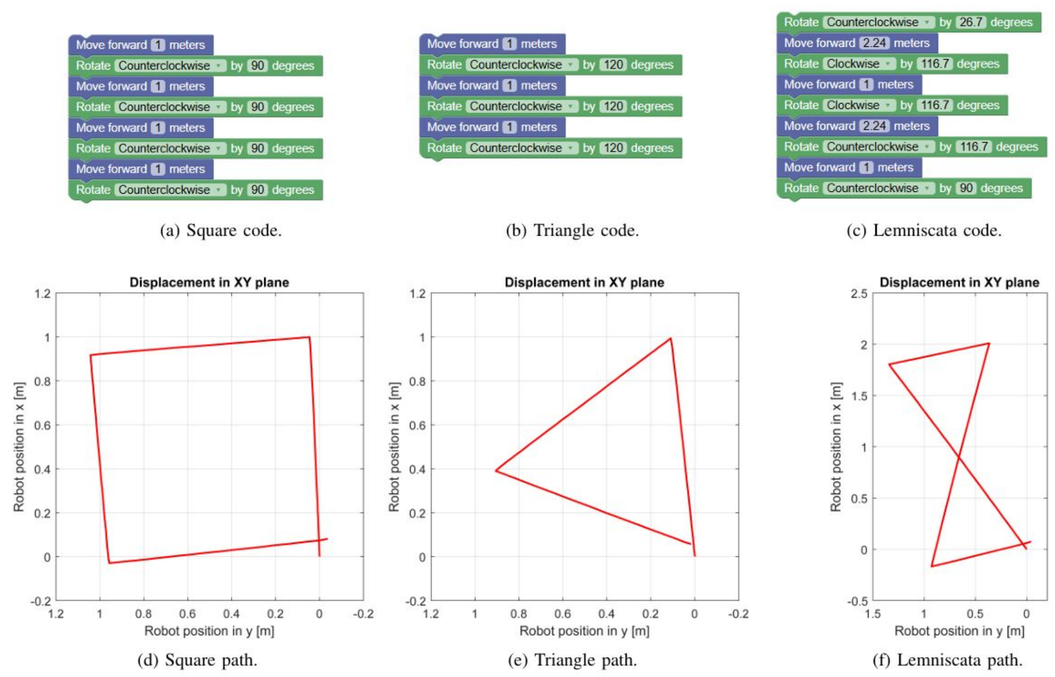


Fig. 2: Experimental setup.

Results

We implemented and validated the platform in three experiments with the **Pioneer 3-DX** robot: **square**, **equilateral triangle** and a representação discretized representation of **Bernoulli's lemniscate**. This evidenced **coherence** between the block program and the recorded **trajectories**. For the square, the path **returned near the starting point** (Fig. 3a/3d). In the equilateral triangle, the **three sides** and **turns (120°)** maintained the predicted proportions (Fig. 3b/3e). For the lemniscate, an "8" shaped curve with "**abrupt**" transitions was obtained (Fig. 3c/3f), with trajectory accuracy analyzed by **odometry graphs**. The core of the solution includes **move and turn blocks connected in sequence**, translating **blocks → JSON → backend (Python/Flask) → ROS** for publishing to the **/cmd_vel** topic (velocity commands) and state monitoring with the **/pose** topic, forming an **end-to-end flow**.



(a) Square code. (b) Triangle code. (c) Lemniscata code.

(d) Square path. (e) Triangle path. (f) Lemniscata path.

Fig. 3: Blockly code implementation (top) and resulting paths (bottom) for three geometric paths: square (left), equilateral triangle (middle), and lemniscate (right), showing the robot's path-following.

Conclusions

The experiments confirmed the viability of the Blockly programming platform integrated with ROS to perform navigation and figure drawing tasks with the Pioneer 3-DX; the odometry graphs showed good accuracy in a controlled environment and pedagogical potential for interactive geometry teaching. The platform achieved its objectives, offering an accessible and modular environment for programming mobile robots, with effective Blockly-ROS integration.

Bibliography

PINEL, G. S. F.; BRANDÃO, A. S.; FARIA, R. W. S. C. Blockly and ROS: A Visual Programming Interface for Robot Control. In: BRAZILIAN CONFERENCE ON ROBOTICS (CROS), 2025, Belo Horizonte. Anais [...]. Belo Horizonte, Brasil: IEEE, v. 1, p. 1-5, 2025. DOI: 10.1109/CROS66186.2025.11066105.

Financial Assistance

CAPES FAPEMIG CNPq