Autonomous Robot Navigation (team: RoboCafé)

1st Soňa Michalková
Dept. of Control and Instrumentation
Brno University of Technology
Brno, Czech Republic
247277@vutbr.cz

2nd Kryštof Wojnarowský
Dept. of Control and Instrumentation
Brno University of Technology
Brno, Czech Republic
247501@vutbr.cz

***Abstract*—This paper presents a navigation system for an autonomous robot to solve maze-type environments using LiDAR, IMU, and ArUco markers. The system integrates sensors and algorithms for autonomous obstacle detection, wall-following, and decision-making at intersections. The robot's movement is controlled by odometry and PID algorithms, adjusting speed and turning rate based on sensor data. Maze junctions are detected using LiDAR and modulo distance logic, while decisions are made based on the robot's orientation, available exits, and ArUco markers. This work demonstrates the potential of combining traditional sensors with computer vision for autonomous navigation in constrained environments.**

***Index Terms*—Robotics, project, BPC-PRP, sensor fusion, robot navigation**

I. Introduction

Autonomous robots are increasingly used in industrial automation, transportation, and robotics research, particularly for navigating complex environments like mazes. This paper presents a navigation system for solving mazes using LiDAR sensors, odometry, IMU for orientation, and ArUco markers for decision-making at intersections.

LiDAR serves as the primary sensor, enabling the robot to detect obstacles and follow maze walls. The IMU ensures the robot maintains accurate orientation for smooth turns. ArUco markers at intersections guide the robot's path. By integrating these sensors with odometry, the robot can navigate autonomously, reacting to obstacles and decision points.

The paper discusses the system's design, sensor integration, path-following algorithm, and decision-making logic for efficient maze-solving.

II. Methodology

A. Sensor Fusion for Navigation

The robot integrates multiple sensors for comprehensive environmental perception. These sensors are connected to the central control system via ROS 2, which provides seamless communication between various nodes and sensor data processing.

- The LiDAR provides essential distance measurements around the robot. It is used to detect obstacles, follow walls, and assist in localizing the robot by measuring the distance to nearby walls.
- IMU provides critical orientation data, helping to track the robot's rotation and adjust movements, especially during turns.

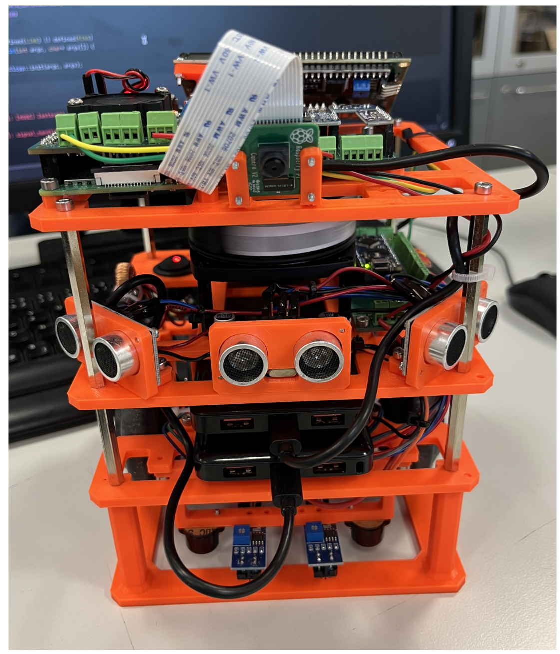

Fig. 1. Autonomous Robot with Sensors

- The camera captures images of ArUco markers placed at intersections within the maze. By recognizing these markers, the robot can make decisions about its next move, either continuing forward or turning at an intersection.

Sensor data from these components are processed concurrently in real time, providing the robot with accurate feedback on its environment, which is essential for both localization and obstacle avoidance.

B. Odometry: Position and Movement Calculation

Odometry is essential for tracking the robot's position and movement in the maze, enabling localization without GPS or external references. The system uses wheel encoders to measure the distance each wheel travels by counting pulses generated during wheel rotations. It also calculates the robot's orientation change (turn angle) based on the differential movement of the two wheels.

1) Calculating the Robot's Position: The formula:

$$\text{pos\_in\_cell} = \text{front}\%\text{cell width}$$

calculates the robot's position within the current cell. The modulo operation ensures the robot's position is within the bounds of the cell by returning the remainder when the distance traveled (front) is divided by the cell width. This approach effectively maps the robot's movement in grid-based environments like mazes, ensuring its position remains confined within each cell.

This variable and the number of exits from the current cell can reliably identify turns and intersections, where the required operation will then be performed based on the detected ArUco code.

C. PID Control for Motion and Stability

To ensure that the robot moves smoothly and accurately, a PID controller is used to adjust the robot's velocity and steering. PID control is essential for fine-tuning the robot's movements, particularly in wall-following and turning scenarios where precise control is needed.

The PID values set for our system are as follows. PID for Wall Following (Steering):

- Proportional (P): 12.0
- Integral (I): 5.3
- Derivative (D): 9.8

PID for Turning (Rotation):

- Right Turn: Proportional (P): 5.0, Integral (I): 7.0, Derivative (D): 1.3
- Left Turn: Proportional (P): 5.0, Integral (I): 7.0, Derivative (D): 1.5

These values were optimized through precise calibration, utilizing the debug mode in the main program. The robot's behavior was monitored and PID values were empirically adjusted to minimize errors and improve performance.

The proportional component responds to current errors, the integral component addresses accumulated errors, and the derivative component prevents overshooting by adjusting based on the error's rate of change. This approach ensures stable and accurate motion, allowing the robot to follow walls and turn precisely in the maze.

D. Finite State Machine (FSM)

- **Wall Following**: The robot follows the walls and, using PID, keeps itself centered in the corridor. In this state, the robot moves only straight and, upon hitting a wall, switches to either *Initiate Turn* or *Initiate Intersection*, depending on the number of exits detected from the current cell.
- **Initiate Turn**: Based on the available exit to the left or right, the robot sets the turning direction and switches to the *Turning* state.
- **Initiate Intersection**: Based on the detected ArUco code, the robot sets the direction of movement to the appropriate path and switches to the *Turning* state. If the robot is instructed to continue straight at a T-junction into a wall, the program will automatically terminate with a warning about an incorrect turn.
- **Turning**: In this state, the robot makes a turn in the predetermined direction based on one of the previous states, utilizing precise PID controllers. After completing the turn, it automatically switches back to the *Wall Following* state.

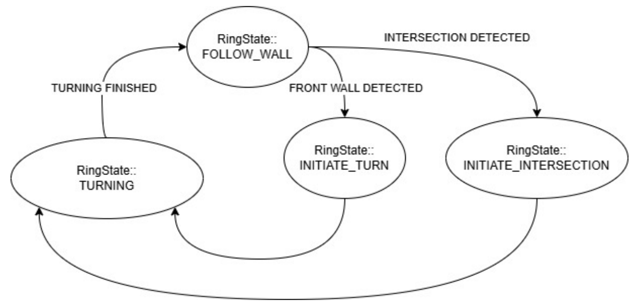

Fig. 2. Component Scheme

III. Software Integration and System Workflow

The robot's control system is built using ROS 2, where each task is managed by a separate software node.

- Sensor Nodes: LiDAR, IMU, and camera nodes gather data for real-time updates.
- Motor Control: Odometry computes movement.
- FSM: Coordinates behavior based on sensor data, deciding when to turn, follow walls, or stop.
- PID Controllers: Ensure smooth and accurate movement by adjusting velocity and steering.

IV. Results and Discussion

The robot successfully navigated out of the maze in just 50 seconds, efficiently avoiding obstacles and following the path laid out by the walls. At the last intersection in the maze, the robot detected and collected the treasure placed at this key location. This achievement highlights the effectiveness of the robot's navigation system, which integrates sensor fusion, real-time decision-making, and precise motion control. By utilizing LiDAR for obstacle detection, an IMU for orientation, and ArUco markers for decision-making at intersections, the robot was able to autonomously navigate, adapt to dynamic changes, and complete the task within the given time frame.

V. Conclusion

This paper described the design and implementation of an autonomous robotic navigation system capable of solving a maze using a combination of LiDAR, IMU, ArUco markers, and odometry. The system integrates these sensors to allow the robot to autonomously detect obstacles, follow walls, and navigate intersections. By combining PID control with real-time sensor data processing, the robot demonstrated reliable performance in following paths and making decisions at key junctions.

Acknowledgment

We would like to thank our supervisors, Ing. Šopák and PhD. Ligocki, for their guidance, expertise, and support throughout the development of this project.