BALL-IN-MAZE SOLVING ROBOT

Author:
Avvienash Jaganathan
Supervised by:
Michael Zenere

INTRODUCTION AND OBJECTIVES

The aim of this project is to develop a vision-based maze-solving robot capable of autonomously controlling a ball within a maze, using image processing, control techniques, and maze solving algorithms to achieve accurate and efficient navigation. This robot will showcase the capabilities of robotics systems for educational and recreational purposes.

COMPUTER VISION AND DETECTION

The Detection Mechanism utilizes Computer Vision Techniques on the image captured by the Raspberry Pi Camera to detect the ball position and maze layout, as shown in Figure 1.

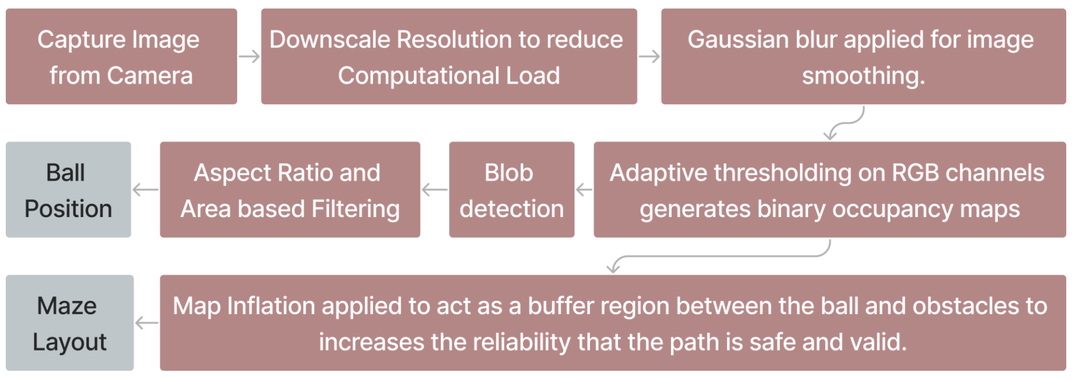

Figure 1: Detection Mechanism Flowchart

PATH PLANNING (RRT*)

The Rapidly Expanding Random Tree Star algorithm was used to generate the path from the Ball to the Goal as shown in Figure 2. The Goal Position was set as the head of the Tree instead of the ball, to ensure reusability.

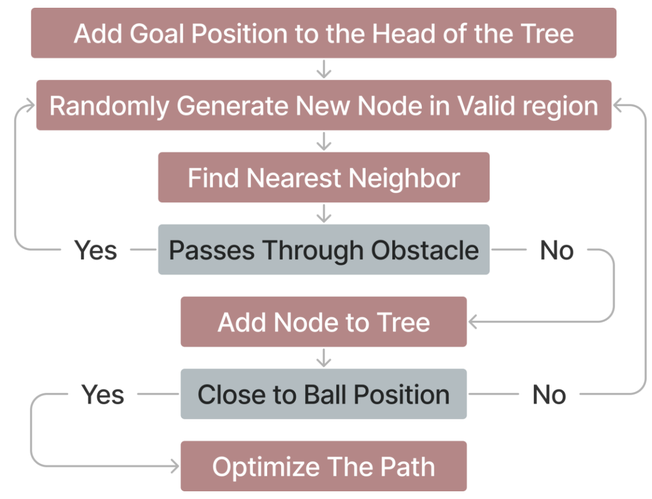

Figure 2: Path Planning Flowchart

PROJECT REPOSITORY

PATH CORRECTING

If the ball goes of course due to external disturbance, it will expand the existing tree and obtain a new path very quickly (~5ms).

MECHANICAL DESIGN

A Ball on Plate Model was designed to control the position and trajectory of a ball on a maze as shown in Figure 3. This Model was controlled via two rotary actuators, each independently responsible for the two degree of freedoms.

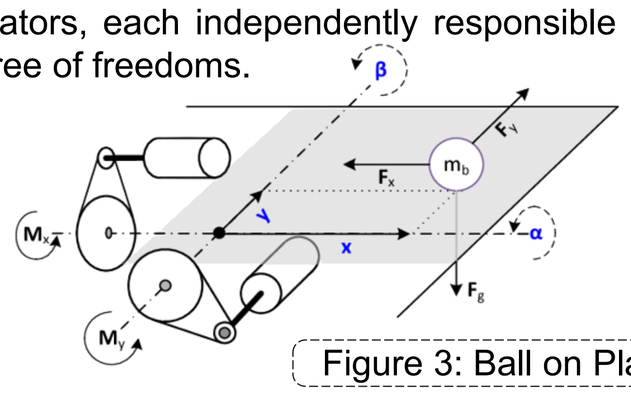

Figure 3: Ball on Plate Model

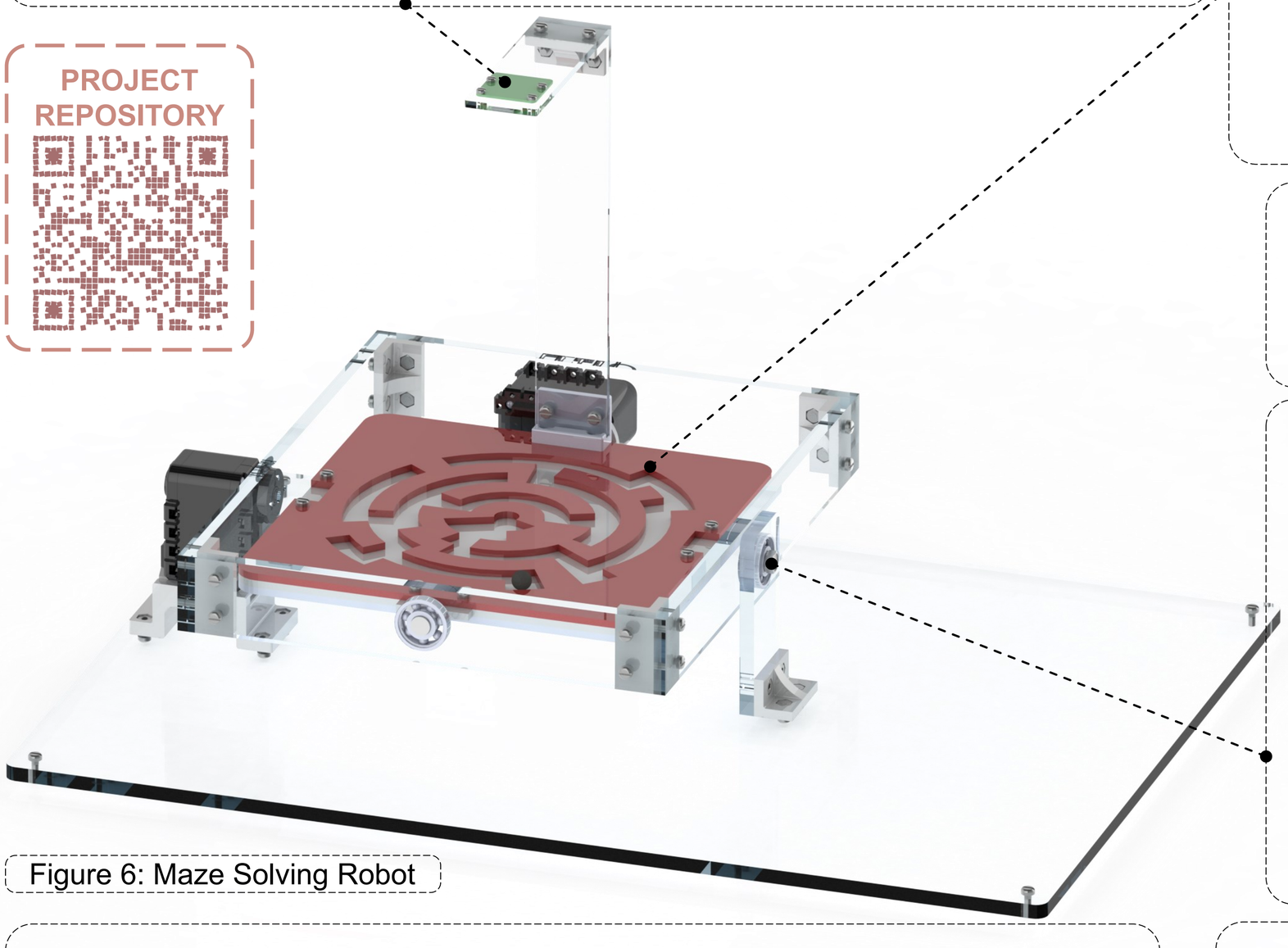

Figure 6: Maze Solving Robot

USER INTERFACE (UI)

Flask-based, intuitive design for easy access as shown in Figure 5. The web-based interface enables real-time control of robot and visual tracking of path and ball position. Users can select goal positions by clicking on the exact position in the interface. The interface also includes basic functionalities such as plotting paths, solving mazes, and canceling a run at any time

Figure 5: Web Based UI

CONTROL SYSTEM

A PID Controller is used to control the ball in the Maze as shown in Figure 4. It obtains the target waypoint (x, y) from the RRT* path. Then controls the servos based on the relative error using PID.

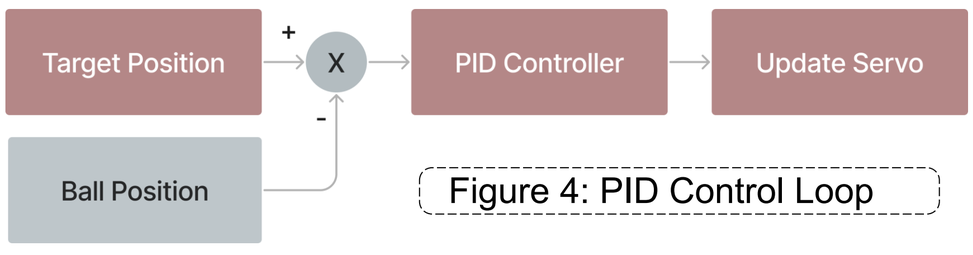

Figure 4: PID Control Loop

REFERENCE

[1] S. Paul and J. van Baar, "Trajectory-based Learning for Ball-in-Maze Games," ArXiv, vol. abs/1811.11441, 2018.
[2] M. O. A. Aqel, A. Issa, M. Khdair, M. ElHabbash, M. AbuBaker and M. Massoud, "Intelligent Maze Solving Robot Based on Image Processing and Graph Theory Algorithms," 2017 International Conference on Promising Electronic Technologies (ICPET), Deir El-Balah, Palestine, 2017, pp. 48-53

RESULTS AND CONCLUSION

The vision-based maze-solving robot successfully demonstrated its capability to autonomously navigate a maze while controlling a ball. The robot efficiently detected the ball's position and maze layout, generated optimal paths using the RRT* algorithm, and executed precise movements using PID control.