Faculty of Engineering and IT
School of Mechanical and Mechatronic Engineering
41069 Robotics Studio 2

MazeRunner

Client: Tan Huynh

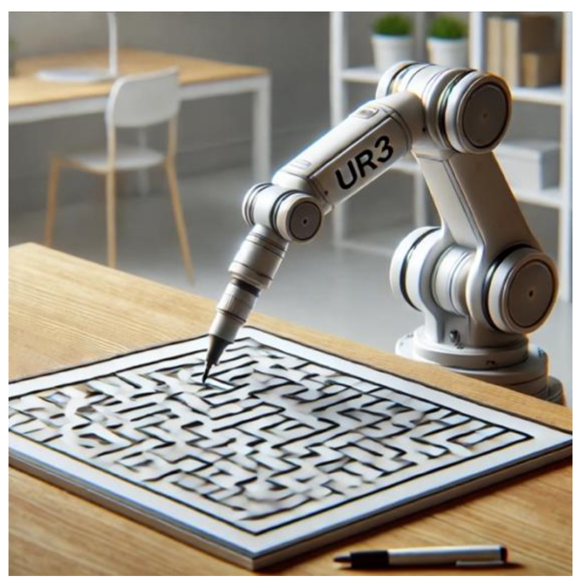
Coach: Ravi Ranasinghe

Project Team:

Full Name	Student ID	Expertise
Cristian Corso	13549237	Mechatronic Engineering
Hugh Radvan	13549340	Mechatronic Engineering
Nicholas Uremovic	14174268	Mechatronic Engineering
Ryan Thomas	13938802	Mechatronic Engineering

Overview

Maze Runner is a robotics project that aims to automate maze solving using a UR3 robotic arm. A camera captures an image of a maze, which is then processed to detect its structure. A path-planning algorithm determines the optimal route, and the UR3 traces the solution using a marker-holding end-effector. This project integrates image calibration, computer vision, motion planning, and robotic control, making it a challenging and valuable exercise in autonomous system development.



Key Challenges

The key challenges associated with our project include the creation of a maze detection system using a RealSense RGB-D Camera and the creation of a maze solving algorithm that can pass useful information to the UR3 that will allow it to draw the solution to the maze. The maze detection system will not only have to be capable of distinguishing the maze boundaries, start, and end points from background information, but also be capable of determining the mazes position within the world frame of the UR3. The maze solving algorithm on the other hand must be robust enough to identify the features of this maze to then then solve said maze using the most efficient solution, passing on appropriate waypoints to the UR3 allowing it to draw the solution. Additionally, a robust custom end effector must be designed to hold a marker.

Expected Learning Outcomes

1. Apply design/systems thinking to the analysis of a robotics problem. (C.1)
2. Apply technical skills to develop, model and/or evaluate a robotics path planning or control solution. (D.1)
3. Demonstrate effective collaboration and communication skills as an effective member or team leader of team/s. (E.1)
4. Conduct critical self, peer and team reflection for performance evaluation. (F.1)

Aims

Goal

The goal of the project is to develop an autonomous system that enables a UR3 robotic arm to detect, solve, and physically trace the solution of a maze using a marker-holding end-effector. This project integrates computer vision, motion planning, and robotic control to create a fully automated maze-solving system.

To achieve this, the following key objectives and milestones have been set:

Objectives & Milestones

1. Develop a vision-based system that can **detect and localise a maze within the UR3 workspace**, accounting for variations in position, orientation, and scale.
2. **Process the maze image** to extract walls, paths, and key points before converting it into a structured graph representation for solving.
3. Implement a **path-planning algorithm** to determine the optimal solution for navigating through the maze.
4. **Control the UR3** robotic arm to follow the solution path smoothly and accurately using trajectory planning and inverse kinematics.
5. Design and integrate a **marker-holding end-effector** that ensures consistent contact with the maze surface for precise tracing.

Deliverables

1. An integrated robotic system based around the UR3 that can accurately draw the shortest solution to a maze place within the robot's workspace.
2. An image processing system that can detect key features of a presented maze, including maze boundaries and accurate world frame coordinates.
3. A maze solving algorithm that can determine the shortest solution to a presented maze, producing a series of sequenced waypoints that define the maze solution.
4. A custom UR3 end effector that can hold a marker to draw the maze.

Resources

We expect to require the following resources during the completion of this project:

- <u>**Hardware**</u>:
    - **UR3 robotic arm** – For drawing the maze solution
    - **Intel RealSense depth camera** – To detect the provided maze
    - **Whiteboard & markers** – For drawing the maze and its solution
    - **Custom end-effecter** – To hold the marker used to draw the maze solution
- <u>**Software**</u>:
    - **ROS1 (Noetic)** – For integrating our code with the UR3
    - **C++** – For coding our systems software
    - **OpenCV** – To detect the provided maze
    - **Gazebo** – For system simulations
    - **Rvis** – For system simulations
    - **MoveIt**! – To help control the UR3
    - **Fusion 360** – To 3D model custom parts for the robot end effector
- <u>**Tools**</u>:
    - **3D printer** – To create the custom robot end effector
    - **Hand tools & fasteners** – To create the custom robot end effector
    - **Mechatronics lab** workspace – To test the UR3 in a safe environment

Subsystems and Responsibilities

Subsystem	Description	Lead by	Supported by
Maze Localisation and Capture	Identifies a maze, and employs visual servoing to move to, and capture an image of the maze. Calculates the maze's position, orientation, and scale within a global frame of reference.	Nicholas Uremovic	Hugh Radvan Cristian Corso Ryan Thomas
Image Processing	Process and correct the captured maze image, removing distortions and extracting key features. Convert the maze into a readable format, identifying walls, pathways, and start/finish points for use in the path-planning algorithm.	Ryan Thomas	Hugh Radvan Cristian Corso Nicholas Uremovic
Maze Solving & Path Planning Algorithm	Algorithm that determines the shortest possible route through the maze, outputting a list of sequential coordinates/instructions to navigate the maze.	Hugh Radvan	Cristian Corso Ryan Thomas Nicholas Uremovic
Robot Motion & Control	Generate smooth and accurate trajectories and integrate control systems with the UR3 to navigate through the maze using the pre-generated algorithm waypoints.	Cristian Corso	Hugh Radvan Ryan Thomas Nicholas Uremovic
Marker-holding End-effector	A robust, custom end effector that securely holds the drawing instrument.	Hugh Radvan	Cristian Corso Ryan Thomas Nicholas Uremovic

Evaluation

Maze Localisation and Capture	
P	Maze is placed in a known fixed location. Camera moves to maze to take top-down image.
C	Maze is placed in a known fixed location with rotation. Camera moves to maze to take top-down image.
D	Maze is placed in an arbitrary, unknown location. Camera moves to maze to take top-down image. Camera can determine the maze world coordinates.
HD	Maze is placed in an arbitrary, unknown location with rotation. Camera moves to maze to take top-down image. Camera can determine the maze world coordinates.
Extension	Maze is placed in a location with varied translation & rotation. Camera takes an image of the maze from an angle, accounting for the perspective shift. Camera can determine the maze world coordinates.
Image Processing	
P	Successfully process a perfectly framed maze supplied from an image file (no photo).
C	Successfully process a clear top-down image of a printed maze.
D	Successfully process an image of a perfectly hand drawn maze using a grid guide.
HD	Successfully process an image of a roughly hand drawn maze using a grid guide, accounting for some drawing errors such as poorly drawn lines.
Extension	Successfully process an image of a hand drawn maze using a grid guide that includes diagonal paths within the maze.
Maze Solving & Path Planning Algorithm	
P	Establish waypoints at the start and finish point of the maze.
C	Establish a maze solution allowing for periodic mistakes (moving back on itself, boundary crossing, etc).
D	Establish a maze solution with no mistakes (moving back on itself, boundary crossing, etc).
HD	Establish the most optimal solution (shortest route) to the maze with no mistakes.
Extension	Calculate waypoints that would allow the robot to re-draw a supplied maze from scratch before solving it.
Robot Motion & Control	
P	Draws a line between two waypoints.
C	Draws a continuous line between a series of waypoints.
D	Draws a continuous line between a series of waypoints utilising minimum jerk trajectory.
HD	Draws a continuous line between a series of waypoints utilising minimum jerk trajectory avoiding maze boundary crossings.
Extension	Incorporation of Singularity avoidance systems along planned path.
Marker-holding End-effector	
P	Rigidly attached marker using custom built end-effector.
C	Custom built end-effector with low tolerance (no wiggle or looseness at marker).
D	Custom built end-effector that mitigates excessive force imparted on the marker using a spring-loaded system.
HD	Custom built end-effector that mitigates excessive force imparted on the marker using UR3e force sensor.
Extension	Custom built Modular end-effector (capable of holding different marker brands/types)