The Elevator Problem: Concept of Operations

EG2310 Group 5

Team Members

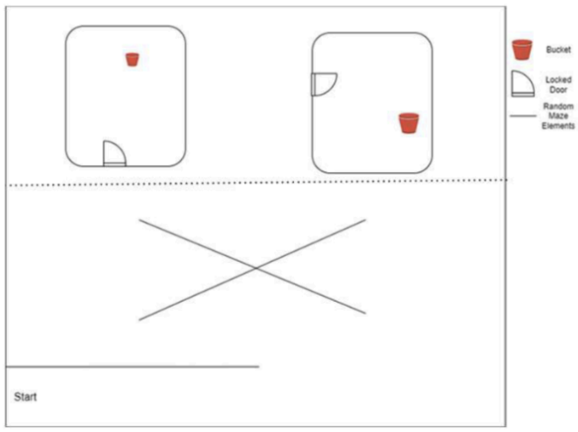
Datta Anitej	A0284708U
Namit Deb	A0273659M
Sriram Anupama	A0276023L
Vishwanath Annanya	A0277110N
Wong Weng Hong	A0272156E

1.0 Problem Description

1.1 Mission Overview

The mission is to design a robot which can solve the "elevator" problem. The robot must autonomously navigate from a start point, map the randomised maze, communicate with and open a secured door, locate a bucket inside the room, and fire 5 ping pong balls into said bucket.

1.1.1 The Map



The mission area (square of side length 3.5m) is divided into three zones: the start zone, the randomised maze, and the locked room with a door (firing zone). The robot must navigate the maze and develop a full map of the area using Simultaneous Location and Mapping (SLAM).

1.1.2 The "Elevator" Door

Beyond the maze, there will be 2 doors secured by electronic locks wired to a system which unlocks via a web server. The robot will need to make an HTTP request to the web server, and will in return receive a standard JSON response. If the robot's request is valid, the response will contain the ID of the correct door, which the robot must parse in order to identify and enter the correct room.[1]

1.1.3 The Bucket

The robot must locate a bucket which will be placed at an unknown location in the room. The robot is to locate the bucket and fire 5 ping pong balls into it.

1.2 Assumptions and Constraints

Assumptions

- Elements of the mission area remain constant during mission execution

Constraints

- Temporary markers cannot be change/shifted by the team during

[1] Note that as of 2nd February, details of the web server and HTTP request have not yet been provided

- Environmental factors (temperature, humidity, lighting, etc.) are constant
- No external interference with communications hardware
- Web server to respond to request in reasonably short time
- The maze corridors & door will be wide enough for the robot to maneuver
execution
- Team will have 25 minutes for mission including documentation review
- Robot cannot physically alter the maze or interact with bucket
- The maze will be completely randomised and thus not necessarily simply connected[2]
- The elevator door remains unlocked for a limited time

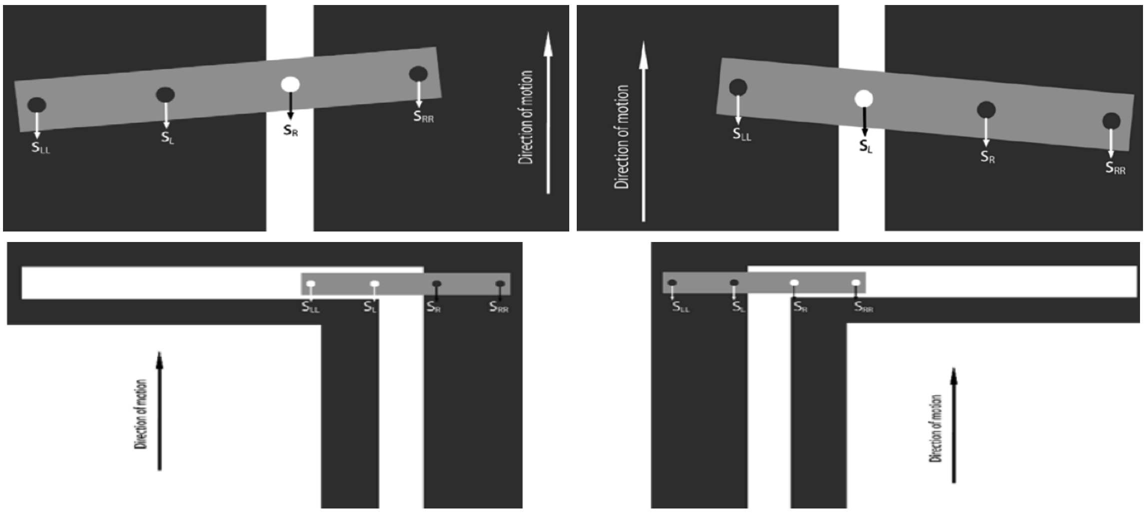
2.0 Literature Review

2.1 Navigation

To navigate the randomised maze the robot will need some maze solving algorithms. Options include:

<u>Line Following</u>

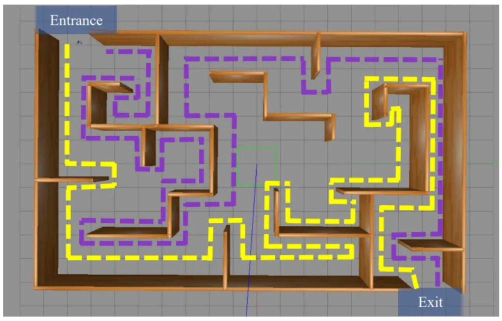
Team will use a version of the algorithm by Chowdhury, Khushi, and Rashid [1], and lay out a line for the robot to follow using coloured tape. With a system of 4 sensors, the robot will follow along the line correcting its path by adjusting the left/right motor speeds to steer. The outermost sensors are to help detect sharper 90º turns.



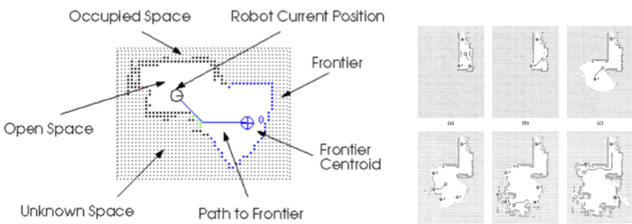
<u>Wall Following</u>

Robot chooses a direction (left/right) and follows the wall along that side. It will eventually completely explore a simply connected maze. This is a form of depth-first search in-order tree traversal [2].

[2] A simply connected maze is one where all walls are connected to the maze's outer boundary



Frontier Based Exploration
Here, the robot develops frontiers on the boundary between explored and unexplored space on the map. By moving to new frontiers, the robot can build the environment map until there are no more frontiers left [3].



2.2 Communication & Sensing

2.2.1 Python Requests Library

Python has a developed Requests library which the robot can use for interaction with the web server. This allows for sending of HTTP requests and receiving of JSON responses [4].

2.2.2 Laser Distance Sensor (LDS-02)

The robot uses a LDS-02 sensor to map its environment by identifying obstacles like walls. The LDS sensor has an angular range of 360° which is used for SLAM. This involves using a Python program to continually update the position of the robot on the current mapped area in its internal memory [5].

2.2.3 Door & Bucket Identification

NFC Tags
NFC tags can be placed around the maze for the robot to receive information on specific locations relative to the map, for example its location relative to the elevator doors and instructions on how to proceed. The tags can contain data allowing the robot to differentiate doors [6].

Open Computer Vision (OpenCV) Library
Applications of OpenCV such as Object Detection and Recognition can be used by using cameras for the algorithm to detect a marker or recognise the door and the bucket [7].

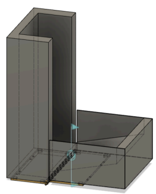
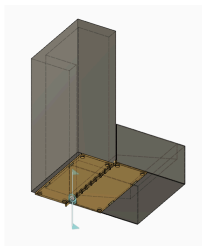
Lines (If Implementing Line Following)

Lines made of tape can be placed in the map for the robot to directly follow, without needing any decision making [1]. Can be placed for specific tasks such as moving from door to bucket.

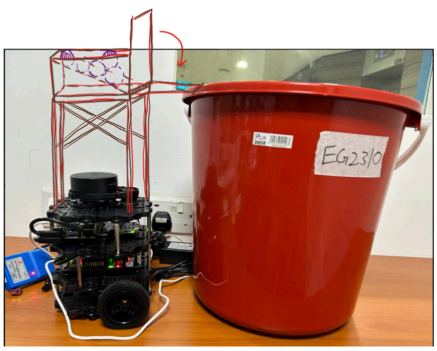
2.3 Ball Delivery System

<u>Projectile System</u>
TurtleBot navigates to a predefined distance from the bucket, and employs a high-speed rotating wheel in a chamber to launch ping pong balls.

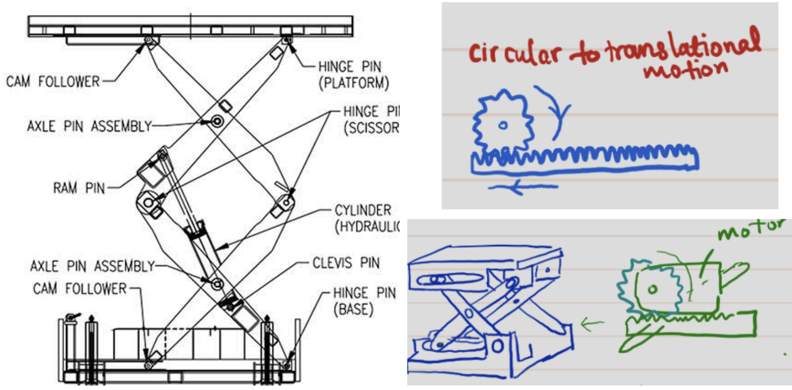
<u>Payload Mechanism</u>
The design features a box with a sloped floor, facilitating the rolling of the ping-pong balls into the bucket. To enhance functionality, one side of the box is designed as a convertible ramp. A butt hinge connects the ramp to the box, and a servo motor adds automation to the system, enabling the opening of the box up to a predefined angle. A thin gauze will be attached over the top of the box to prevent any balls from falling out of the box during the run.

<u>Tower System</u>
TurtleBot moves as close to the bucket as possible and rolls the ping pong balls into the bucket. The box containing the balls will be placed at a height, taller than the bucket, using stilts.



<u>Scissor lift + Ramp System</u>
An alternative to stilts in the tower system is employing a scissor lift mechanism, activated once the TurtleBot reaches the bucket [8]. This ensures a compact design for the TurtleBot throughout the run. Instead of a hydraulic actuator, as depicted in below, a heavy-duty servo motor will be utilised to initiate the scissor lift.



Positioned at the base's centre is a motor, paired with a spur gear and a rack, enabling the conversion of the motor's circular motion into translational motion. Extending from the sliding end of one scissor arm to the rack, a long cylinder is integrated. This configuration, activated by the motor, induces the movement of the sliding arms, raising the lift's height.

3.0 Concepts Design

The mission's three phases each have their own subsystem, of which the options are detailed below, along with their respective requirements.

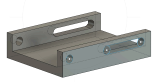
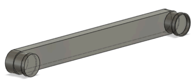
3.1 The Maze

Proposed Subsystem	**Hardware**	**Software**
Line Following	Floor facing reflective sensors OpenCR Raspberry Pi 3 DynaMixel Motors (Wheels) Additional: Tape as temporary marker (line)	ROS2 OpenCR interfacing node PID controller for motor speed Line following algorithm RViz
Wall Following	LDS-02 OpenCR Raspberry Pi 3 DynaMixel Motors (Wheels)	ROS2 OpenCR & LDS-02 interfacing node PID controller for motor speed Wall following algorithm Data structure for backtracking & path memory RViz
Frontier Based Exploration	LDS-02 OpenCR Raspberry Pi 3 DynaMixel Motors (Wheels)	ROS2 OpenCR & LDS-02 interfacing node Frontier based navigation algorithm RViz

3.2 The Elevator Door & Bucket Identification

Proposed Subsystem	**Hardware**	**Software**
NFC Tags	NFC Tag NFC Sensor	Python nfcpy library A* Algorithm for pathfinding
OpenCR	Raspberry Pi Camera	Algorithm for recognizing and interpreting door features ROS2 camera interfacing node OpenCV library RViz
Line Following	See Section 3.1 for details	

3.3 The Bucket

Proposed Subsystem	**Hardware**	**Software**
Scissor Lift Mechanism	3D printed Base 3D printed scissor arms x 8 EzRobot Heavy Duty servo which can lift 15 kg/cm [9] Spur gear and rack  	Program to initiate the servo motor and stop the motor once a the necessary height has been achieved
Tower mechanism	3D printed stilts x 4 Triangular truss components	
Payload	3D printed box & ramp Butt Hinge Servo motor Thin gauze	Program to initiate the servo motor to ‘open’ the box until a predefined angle.

3D printing

For 3D printing, all components will be crafted using Polylactic Acid (PLA) filaments, with an estimated tensile strength of 40MPa-50MPa [10]. To enhance the overall strength of the printed parts, we are considering the implementation of a honeycomb or triangle infill pattern.

4.0 Bunch of Guys Around Table (BOGAT)

The final solution is an integrated system which must balance mechanical, electrical, and software requirements. The options presented in Section 2.0 have been evaluated below, explaining the rationale behind our final decision.

4.1 Navigation

	Line Follower	**Wall Follower**	**Frontier Based**
Pros	Turtlebot does not need to make any decisions	Straightforward and guarantees full exploration	Have little setup as the algorithm needs little markers
Cons	Requires additional instructions to determine which door to move to Takes time to set up and remove the tape, might exceed time limit	Inefficient as every path is explored including those already mapped Relies on simply connected maze Needs separate algorithm to locate the elevator doors	Robot may not be able to handle unforeseen scenarios or edge cases Needs a separate algorithm to locate the elevator doors
Decision	The team has decided to follow a Frontier Based navigation algorithm as it can efficiently map the entire maze portion without repeating previously explored segments, while also minimising setup time during the 25 minute mission execution window. A separate set of instructions are required once the robot exits the maze, however. This is the phase where the temporary markers will be crucial for assistance.		

4.2 Temporary Markers

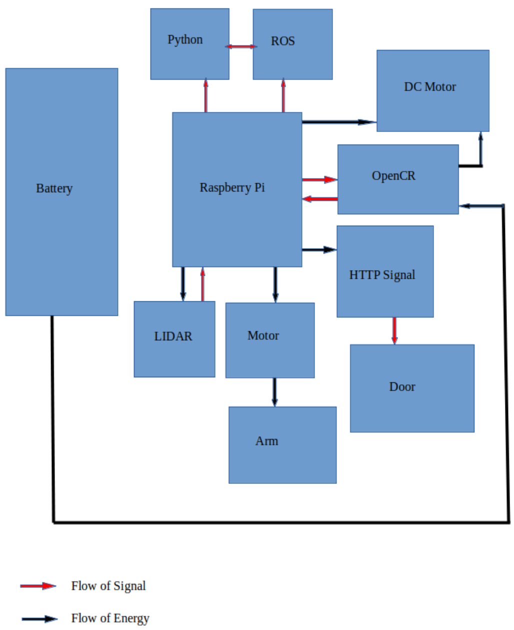
	NFC Tags	**OpenCV**	**Lines (Line following)**
Pros	Cheap Can be embedded with data to help differentiate doors	Able to detect and recognise from afar	Cheap
Cons	Short range, requires accuracy of movement	Potentially unreliable and expensive	Takes time to setup
Decision	The team has decided to use NFC tags placed on the floor to determine the robot's movement towards the door and location of the bucket. It is an inexpensive, reliable solution, and the NFC tags can contain data allowing the robot to move from each subsequent tag with ease.		

4.3 Ball Delivery System

	Scissor Lift	**Tower**
Pros	More compact Will not disrupt the Lidar sensor	Simpler execution and construction
Cons	Complex design, requiring a separate platform Potential need for counterweights Decision-making between a motor or hydraulic actuator for automation.	Disrupt the Lidar sensor Unstable due to height Affects the balance of the turtlebot
Decision	Since there is no height limit throughout the run, the team has decided to go forward with the tower design due to the ease of executing this design, and minimal disruption to the Lidar sensor.	

5.0 Preliminary Design

The proposed solution uses a TurtleBot3 Burger, integrated with ROS2 Foxy Fitzroy, OpenCR, and a Raspberry Pi 3. The robot is equipped with an LDS-02 for SLAM capabilities, visualised through RViz. For the maze, the team will use a Frontier Based maze solving algorithm. For the bucket section, the robot will be equipped with 4 stilts to elevate the balls and then roll them into the bucket. For the temporary markers, the team will use NFC tags to identify each door and the A* Algorithm to determine the shortest path to the bucket.



Works Cited

[1] N. H. Chowdhury, D. Khushi, and Md. M. Rashid, "Algorithm for Line Follower Robots to Follow Critical Paths with Minimum Number of Sensors," *International Journal of Computer*, vol. 23, no. 1, pp. 13–22, Jan. 2017. Accessed: Feb. 1, 2024. [Online]. Available: https://core.ac.uk/download/pdf/229655734.pdf

[2] A. Y. Z. Dao, "Maze escape with wall-following algorithm.," Medium, https://andrewyong7338.medium.com/maze-escape-with-wall-following-algorithm-170c35b88e00 (accessed Jan. 27, 2024).

[3] A. Topiwala, P. Inani, and A. Kathpal, *Frontier Based Exploration for Autonomous Robot*, Jun. 2018. doi:1806.03581

[4] K. Reitz, "Requests," PyPI, https://pypi.org/project/requests/ (accessed Jan. 27, 2024).

[5] "Laser sensors: All types and how they work," Sensor Partners, https://sensorpartners.com/en/knowledge-base/the-different-types-of-laser-sensors/#:~:text=These%20distance%20sensors%20work%20on,can%20internally%20determine%20the%20distance (accessed Feb. 2, 2024).

[6] Tim, "How to use a NFC reader with Raspberry Pi - Tutorial Australia," Core Electronics, https://core-electronics.com.au/guides/how-to-use-a-nfc-reader-with-raspberry-pi/ (accessed Feb. 2, 2024).

[7] TechieYan Technologies, "OpenCV for robotics: Building A vision system for your robot," Medium, https://medium.com/@jagdishtechieyan/opencv-for-robotics-building-a-vision-system-for-your-robot-d79cd64d0794 (accessed Feb. 2, 2024).

[8] "What is the mechanism of a scissor lift used for?," Handling, https://handling.com/guide/scissor-lift-mechanism/ (accessed Feb. 2, 2024).

[9] "Ezrobot Heavy Duty Servo," RobotShop USA, https://www.robotshop.com/products/ezrobot-heavy-duty-servo (accessed Feb. 2, 2024).

[10] G. Gao, F. Xu, J. Xu, and Z. Liu, "Study of Material Color Influences on Mechanical Characteristics of Fused Deposition Modeling Parts," *Materials*, vol. 15, no. 19, p. 7039, Oct. 2022, doi: https://doi.org/10.3390/ma15197039.

Appendix A: Acronyms and Abbreviations

BOGAT	Bunch of Guys Around Table
HTTP	HyperText Transfer Protocol
ID	Identification
JSON	JavaScript Object Notation
LDS-02	Laser Distance Sensor 2
NFC	Near Field Communication
OpenCR	Open-Source Control Module for Ros
OpenCV	Open Computer Vision
ROS2	Robot Operating System 2
SLAM	Simultaneous Location And Mapping

Appendix B: Glossary of Terms

Depth-First Search	Algorithm for searching tree data structures, starting at the root and going as far as possible on each branch before backtracking
Frontier	Area between what has already been mapped and what is unknown to the map
Gazebo	Simulation environment that provides a physics engine and a 3D graphical interface for robots operating on ROS2
Hydraulic Actuator	Hydraulic actuators are cylindrical tube and piston assemblies that utilize hydraulic power to generate linear, rotary, or oscillatory motion
HyperText Transfer Protocol	Set of methods indicating desired action to be performed for a given resource
Infrared	Electromagnetic radiation with longer wavelengths than visible light, typically used to create markers that emit infrared light for detection by infrared sensors
In-Order Tree Traversal	Method of traversing tree structures where the left child of the node is visited, followed by the node itself, followed by the right child
JavaScript Object Notation	Standard data interchange format that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and arrays
LDS-02	Laser scanner capable of sensing 360 degrees that collects a set of data around the robot to use for SLAM (Simultaneous Localization and Mapping) and Navigation and has a range of 160~8000mm
NFC Tag	Short-range wireless technology (range ~4cm) used to initiate connections
OpenCR	Control module developed for ROS2 embedded systems
OpenCV	Library of programs for real-time computer vision
Payload	Section of a systems' physical load
Python	Dynamically typed high-level programming language which supports multiple programming paradigms
Rack	Gear with infinite diameter
Raspberry Pi 4	Single-board computer, run on Linux, designed for computer education
ROS2 Foxy Fitzroy	Open-source software development kit for robotics applications
RViz	3D visualisation tool used in the Robot Operating System (ROS) to display sensor data, robot models, and other relevant information for debugging and analysis in a

	graphical user interface
Scissor Lift	A surface raised or lowered by the closing or opening of crossed supports pivoted like the two halves of a pair of scissors
Servo Motor	Rotary actuator which allows precise rotational control
Simply Connected Maze	Maze where all walls are connected to the maze's outer boundary
Simultaneous Location and Mapping	Process of constructing and/or updating a map while keeping track of an agent's location within it
Spur Gear	Simplest type of gear
Subsystem	Self-contained system within the larger system
TurtleBot3 Burger	Customisable robotics educational platform
Wall-Following Algorithm	Maze-solving algorithm designed around following either the left or right wall in a simply connected maze
Web Server	Software which accepts requests via HTTP

Appendix C: Component Specifications

Component	**Total Weight/g**	**Misc**
Dynamixel Motor (2x)	113	
Raspberry Pi 3	45	
OpenCR 1.0	62	
LDS-02	129	
Li-Po Battery	139	11.9V, 1800mAh Charge/Runtime:2hr 30min
PCB Support (12x)	19	
Waffle Plates (8x)	317	
Wheel (2x)	42	
M3x45 Support (15x)	99	
M3x35 Support (5x)	25	

Table C1: TurtleBot3 Components & Weights

Dimension	**Magnitude**
Bucket Height	32.5cm
Bucket Diameter	33.9cm
Ping-Pong Ball Diameter	4cm
Ping-Pong Ball Weight	2.7g

Table C2: Measurements of Bucket and Ball