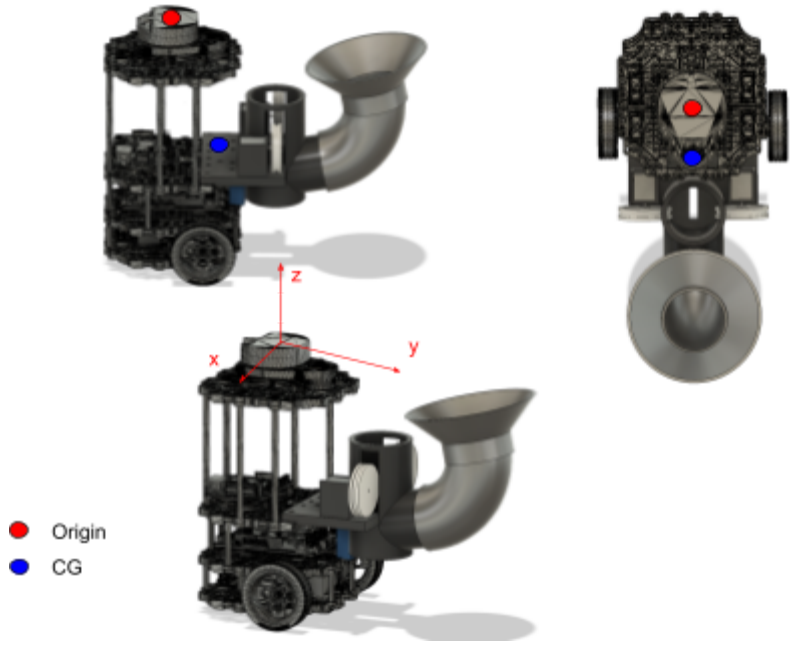


1. General System Description & Critical Data

Technical Data

Max linear Speed (m/s) 0.22	**Max Rotational speed (deg/s)** 62	**Battery Capacity (mAh)** 1800
Total Mass 1.445 kg (without counterweights)	**Dimension W, L, H (cm)** 18, 35, 26	**Centre of Gravity X,Y,Z (cm)** 0.05, 3.84, -13.2
Navigation Algorithm Frontier-based exploration	**Temperature detection system** AMG8833	**Firing System** Dual Flywheel
Software Ros2 Humble	**Chassis** Robotis Co Turtlebot 3	

System Description

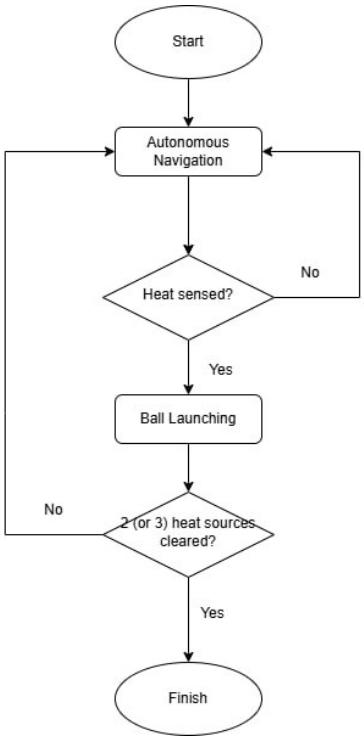
The ZelephantBot is a search and rescue robot built on the TurtleBot platform by Robotis. It is designed to autonomously navigate enclosed maze-like environments, simulating disaster scenarios. Equipped with a temperature sensor, the ZelephantBot can detect heat sources that may indicate the presence of survivors. To signal for immediate assistance, it launches flares upward—represented by ping pong balls. This launching mechanism is powered by a dual flywheel system, and controlled precisely through a single servo motor.

2. Technical Guide

Hardware Diagram

S/N	Description	Qty
01.	Lidar	01
02.	Raspberry Pi	01
03.	Launcher Tube	01
04.	Ball Feeder	01
05.	PVC Elbow Connector	01
06.	Flywheel	02
07.	Flywheel Motor	02
08.	SG90 Servo Motor	01
09.	OpenCR	01
10.	Dynamixel Motor	02
11.	LiPo Battery	01

Software Flowchart



3. Acceptable Deferred Defects Log

Acceptable Deferred Defects Log				
S/N	**Defect Description**	Risk classification Level		
		Low	Med	High
01.	Feeder Servo arm deforming due to prolonged use	X		
02.	Flywheel tires being worn out. Flywheel able to launch ball above the wall	X		
03.	Slight shaking of the top waffle plate due to the increased standoff heights making it unstable	X		
04.	Grooves in the PVC pipe connector may cause the ball to get stuck. Cushion tape had been added to ensure balls will roll over		X	
05.	Motor does spin when screwed into housing, used tight fit to hold it in place	X		

Risk Classification Level

Low: The ZelephantBot will be able to complete the mission as expected.
Med: Although its performance is compromised, the ZelephantBot can still complete the mission.
High : Supervision of the ZelephantBot is required for this aspect. Should issues occur, immediately reset the ZelephantBot for maintenance before initiating another mission.

4. Factory Acceptance Test

Factory Acceptance Test				
S/N	Test Description	Steps to Conduct Test	Expected Result	Proposed rectification
General System				
1.1	Shake test	Remove the battery then shake the ZelephantBot	No loose fasteners and components	Replace loose fasteners with Nyloc nuts
1.2	Is the ZelephantBot able to complete bring up	Run rosbu with the robot placed on the floor.	No error messages and ‘Run!’ message is expected.	-

1.3	Able to drive around?	Run rteleop and drive the ZelephantBot around	Able to drive around without stopping	Observe rosbu to see if motors are disconnected by checking whether the output “Process died: stack smashing detected” is present. Fix connection.
Coordinator System				
2.1	Is the coordinator node initialized and ready to start?	Run the command “python3 ~/Documents/eg2310/coordinator_node.py” on the laptop.	Check that the terminal output reads “Enter ‘r’ to start navigation:”	Try rerunning the command.
Heat Detection System				
3.1	Can the ZelephantBot detect the ‘human’?	Run the command “python3 heat_detection_node.py” on the RPi	The initial mean temperature will be outputted on the terminal and it should not be 0.	Check i2c detect to see whether the address is still available. If so, rerun the command.
Navigation System				
4.1	Is the LIDAR and SLAM algorithm working?	Run rteleop and ros2 launch auto_mapper auto_mapper.launch.py map_path:=~/map is_sim:=true on the laptop from two different terminals and manually navigate allowing ZelephantBot to map the area	Live map constantly recorded and updated.	Keep restarting until the map topic receives updates.
Firing system				

5.1	Can the servo feed the balls on command?	1. Run “python3 motor_driver.py” on the RPi 2. Enter the following input on a new terminal: “ros2 topic pub /shooting_command_in std_msgs/msg/String "{data: 'START'}"”	Check that the flywheel motors spin and the servo moves according to the firing sequence of 2 seconds and 4 seconds	Check wiring connections.
5.2	Can the flywheel spin up at the same time on command?			
5.3	Can the ball be shot upwards high enough	Insert payload into the firing system and run the test codes above.	Observe if the ball gets fired to the acceptable height of 1.5m	Check servo placement.

5. Maintenance and Part Replacement Log

Maintenance and Part Replacement Log			
S/N	Date Reported	Description of Parts Replaced	Remarks
01.	09/04/2025	OpenCR Fuse Replacement	Fuse popped from overcurrent