

# Design Document

Daiwei Fan (1594516), Markus Fuernkranz (1585495),  
Yaodong Li (1536532), Rishi Ramkannan (1537415), Soshala Weerathunge (1563890),  
Shaun Boyteen Joseph (1554972)

Group 02

This design document was created with the assumption that the robot is to take part in the Escape Room Challenge.

## 1 Requirements

The requirements are the demands that we place on the robot (Everything that the robot should be able to do). They are classified as Environment requirements, Border requirements and System requirements.

### 1.1 Environment Requirements

- Wall collision prevention: The Robot should not collide with walls.

### 1.2 Border Requirements

- Task completion time: The escape room challenge must be completed in 300 seconds.
- Idle time: The robot should take sensible actions at least once every 30 seconds.
- Follow wall: The robot behaviour must be designed to follow walls and not breach a “safe” distance from the walls during the entire duration of the escape room challenge.
- Exit detection: In order to complete the task successfully, the robot must be equipped with the ability to recognise exit openings.

### 1.3 System Requirements

- Speed limit: The robot needs to have a maximum speed limit which is less than 0.5.

## 2 Functions

Functions are organized blocks of reusable code which help accomplish the requirements placed on our robot.

### Move

1. Take in to rotate or translate. If translate goto 2), if rotate goto 3). If rotate and translate 4)
2. Translate on
3. Rotate on
4. Rotate then translate

### Exit detection

1. Obtain the point cloud from the Lidar
2. Compare each consequent finite data point with the neighbouring data.
3. If the difference of the two data points exceed a preset threshold value then exit is found. If not go to 4)
4. Obtain the point cloud from the Lidar
5. Compare each consequent finite data point with the neighbouring data.
6. If the difference of the two data points exceed a preset threshold value then exit is found. If not exit is not in the range of the Lidar.

## Wall collision prevention

1. Measure instantaneous distance from the wall.
2. Measure velocity component in the direction of the wall.
3. Based on the data in 1) and 2), compute the threshold distance from the wall based on the maximum deceleration of the robot taking into account the minimum distance that should be maintained by the robot and the wall (specification).
4. If the instantaneous distance is less than the calculated threshold value, then apply braking and aligning the wall, or else move back to 1).

## Aligning the wall

1. Obtain the point cloud from the Lidar.
2. Compare the obtained radii and find the shortest distance between the robot and the wall.
3. Based on 2) obtain the angle that the robot should rotate to align with the wall.
4. Rotate the robot by the computed angle.

## Exit detection robust

1. Measure left wall distance and store value in variable *left initial*.
2. Measure right wall distance and store value in variable *right initial*.
3. Measure rotational velocity.
4. Delay Exit detection for a preset time,  $T$ .
5. Measure left wall distance and store value in variable *left final*.
6. Measure right wall distance and store value in variable *right final*.
7. If  $[(\text{rotational velocity}=0) \ \&\& \ (\text{mag}(\text{leftinitial} - \text{leftfinal})/T) > (\text{Threshold preset value})]$  and  $[(\text{mag}(\text{right initial} - \text{right final})/T) > (\text{Threshold preset value})]$  , then exit is found and go to 8). Else go to 1).
8. End task

## Following the wall

1. Identify a minimum distance from the wall that the robot has to maintain while following it.
2. Measure the distance of the robot from the right wall and the distance of the robot from the left wall and store.
3. If the distance to the right wall and the distance to the left wall are both greater than minimum distance, then go to 4). Else go to 6).
4. If distance to right wall and distance to left wall are both greater than the range of the lrf (30 meters), then the robot is not following a wall. So, find a wall and go to 2). Else go to 5).
5. If the distance to right wall is greater than then the left, then add a component of translational velocity towards the left wall. If the distance to the left is less than the right then translate towards the left wall and go to 7).
6. If left wall distance is less than “safe distance”, then add a small component of translational velocity away from the left wall. If right wall distance is less than “safe distance”, then add a small component of translational velocity away from the right wall.
7. Go to 2)

## 3 Components

- Sensors : Continuously scanning the surrounding environment and find target location.
- Planner : Convert the collected processed data as input and make the decision through the algorithm.
- Navigation : Convert readable decision into the actuator.
- Actuators : Execute the specified direction and speed to let the robot move.

## 4 Specifications

Specifications are the inherent abilities of the robot (Everything that the robot can do).

- The robot can execute translational and rotational motion. The maximum translational velocity of the robot is 0.5 m/s and the maximum rotational velocity of the robot is 1.2 rad/s.
- The range of the laser range finder (LRF) is limited to 30 meters.
- The robot has omni wheels for locomotion.

## 5 Interfaces

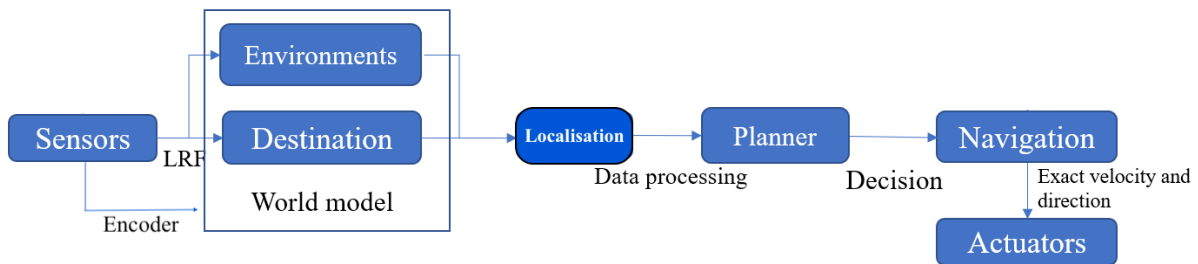


Figure 1: Interface of escape room challenge

The strategy used to solve the task of the escape room challenge is depicted below using finite state machines.

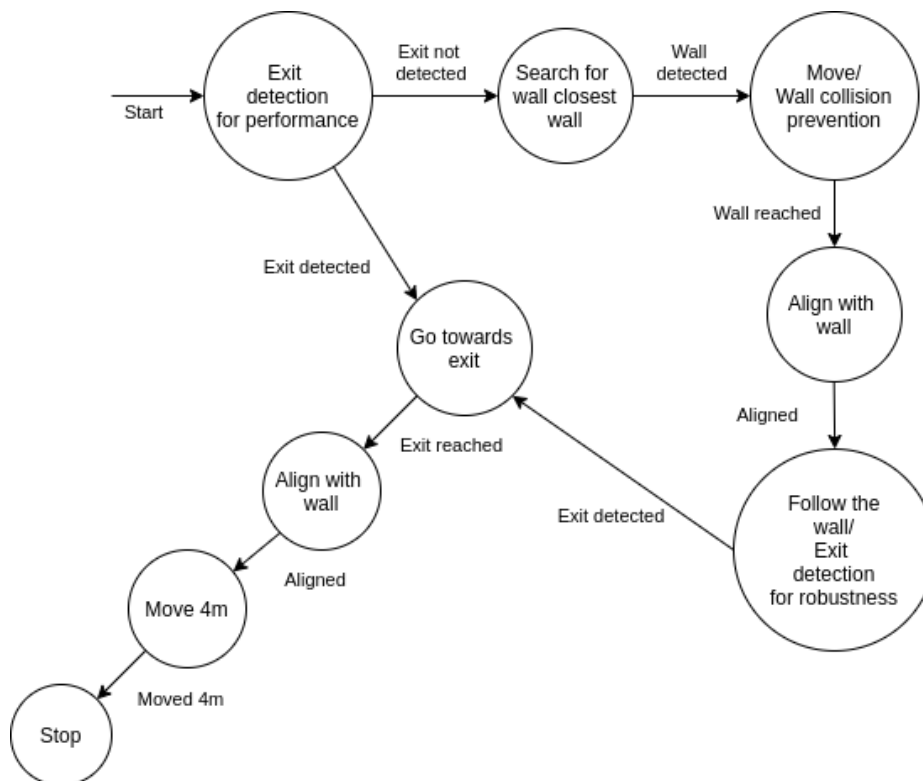


Figure 2: Finite state machine