# Preliminary Report - Group 26
Embodied AI

Frida Jul Pilegaard (frsch18) & Nicoline L. Thomsen (nthom18)

October 2021

## 1 Introduction

The objective is to find a tomato can in a model building, and bring it back out again. This presents three sub-objectives: getting to the can, grabbing the can and carry it out.

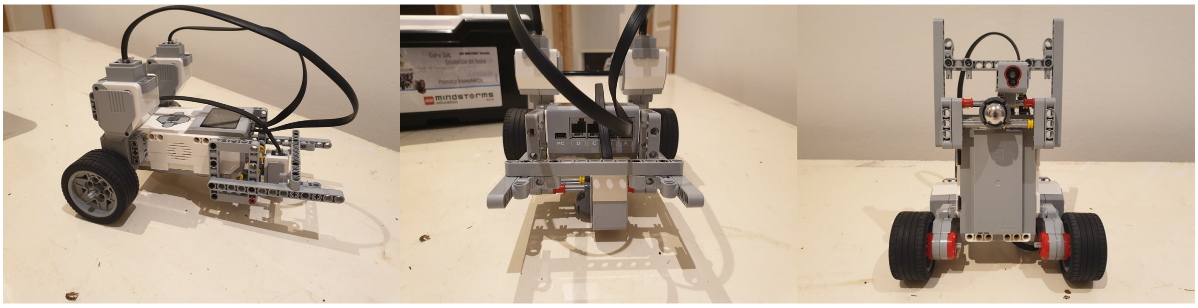
To navigate to the can there are three main challenges to solve:

- Drive up a ramp
- Follow solid line
- Follow dotted line
- Follow wall

Driving up the ramp is solved by designing the robot to have a low centre of gravity, and placing it close to the two wheels, to ensure the most grip. The centre of gravity is not directly under the wheels but it has been shown experimentally that the robot can drive both up and down the ramp and it is therefore satisfactory for the application.

The remaining three tasks are handled by software, that controls the robot motion using various senors. The system comprises of two light sensors, a sonic sensor and a gyro. The actuators are two motors which controls the wheels. Apart from the two motors for the wheels a third smaller motor is used to rotate the sonic sensor.

Images of the robot in its current form can be seen on figure 1.



(a) Side-view (b) Front-view (c) Bottom-view

Figure 1: Pictures of the robot at different angles.

## 2 Sensor Setup

The two colour sensors are placed facing down towards the floor. The sensors are placed in front of the wheels to allow for some prediction of what movement is necessary to follow the line. Currently only one sensor is attached to the robot and this i placed so there is approximately 14 cm between the centre of the wheels and the centre of the colour sensor. The second sensor will be placed similarly and adjusted experimentally to get good performance. The attached sensor is placed slightly offset to the left. This sensor is used to follow the left side of the line for line following. The second colour sensor will be centred in front of the robot and will be used to determine if the line is solid, dotted or non-existing. Meaning that the output from this sensor is used to change between the states.

The sonic sensor is not currently mounted on the robot, but will be placed on a rod which can be turned by a small motor. This will be used to turn the sonic sensor in different directions to be able to detect if a wall is present for wall following and later to follow said wall.

The last sensor, which also has yet to be mounted, is a gyro. This will be used to make the estimation of rotation more precise, which is to be used for the proposed strategies for locomotion, explained more in section 3.3. The gyro will be placed in the middle of the robot

## 3 Behaviours

### 3.1 Navigating To The Can

On figure 2 is a proposed behaviour-based model, to handle the different challenges. Depending on the signals from the sensors, a state machine decides what behaviour to deploy. All behaviours are at this stage mutually exclusive, meaning only one is active at a time, with no blending between them. They also have direct access to send signals to the actuators. Currently the design is made of four behaviours: `line_follow`, `ghost_follow`, `wall_follow` and `hug_can`.

`line_follow` is a simple PID line follow algorithm, that tracks a black line and applies corrections. It is connected to one of the light sensors (left-most), using it to determine the light intensity below the robot. This is the only implemented behaviour at the time of writing this report.

`ghost_line` is when no wall or line is detected. In this case the proposed solution is to simply drive in a straight line forward, until there is a line once more.

`wall_follow` is when there is no line but the sonic sensor detects a wall. A method to follow the wall is not yet developed. This behaviour is the only behaviour that can control the small motor, for the purpose of turning the sonic sensor.

`hug_can` will close the grabber around the can. It is a consideration to make this behaviour mechanical, and not triggered by software.

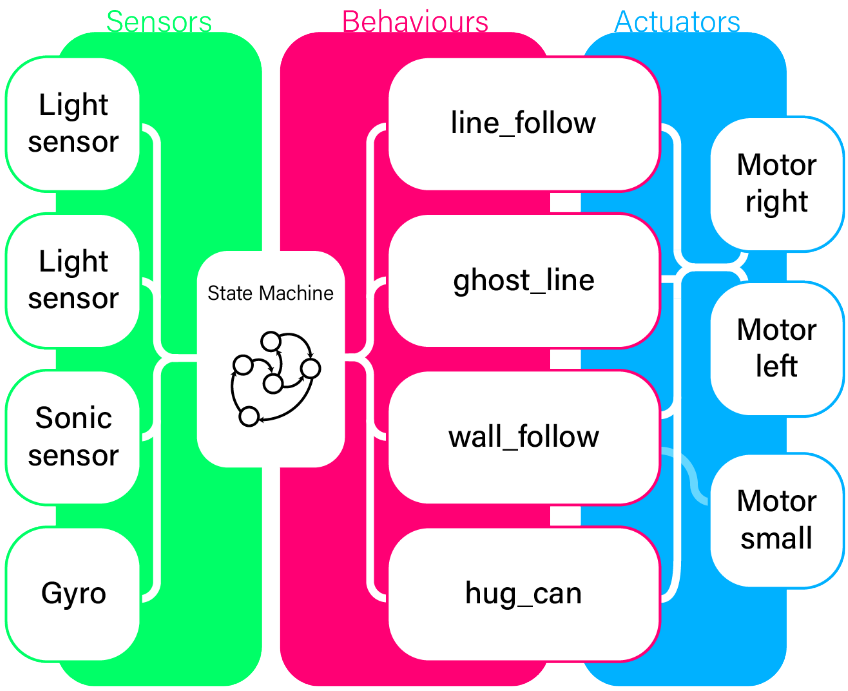



Figure 2: Behaviour based model for control.

Figure 3 shows action selection in form of a state machines, that coordinates the deployment of different behaviours. This state machine is solely dependent on the signals from the light sensors, depicting different states. If one of the sensors is seeing black, then there must be a solid line and `line_follow` is activated. Are both sensors seeing white, the line has stopped, and behaviours to deal with this situation will be activated instead. These behaviours are `wall_follow` and `ghost_line`, the chosen one is dependent on whether or not a wall is present. A way to handle a situation with no line nor wall, is driving straight until a sensor detects black.

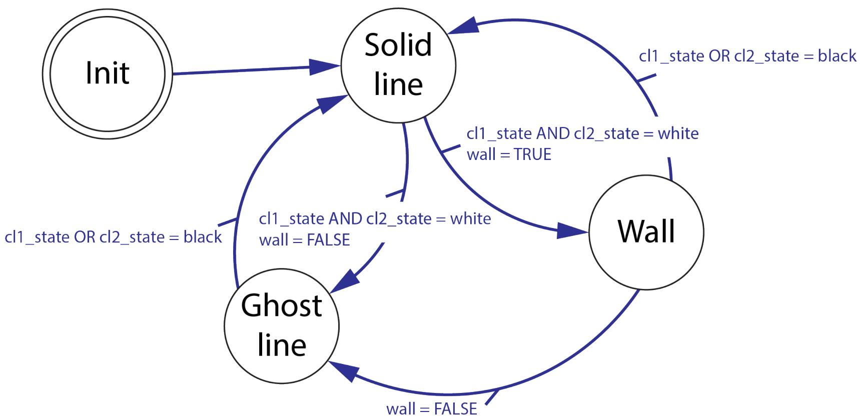



Figure 3: State machine that controls behaviours.

## 3.2 Grabbing The Can

To enable retrieval of the tomato can it is desired to construct a mechanical contraption which automatically closes around the tomato can when it makes contact with it, mounting it on the front of the robot. For the mechanism, inspiration is found by looking at mousetraps and their ability to release only based on touch. The design is not yet functional and therefore not described further.

## 3.3 Bringing The Can Back

When the can is retrieved the robot has to return to the start position. To do so two different approaches are being considered. Both use the gyroscope to assist in determining the movement of the robot. The first and simplest solution will use the gyroscope to turn the robot 180° degrees to be able to follow the line back to the start position, using different PID tuning parameters to account for the shift in mass. An alternative idea which will be explored is to map the environment during the route to the tomato can. Then using this to construct a path back to the start position. This method might improve the duration of the rescue mission, if the new path can cut some corners.

## 3.4 Testing of System

The speed with which the robot will be able to solve the task of retrieving the tomato can will depend upon the speed at which the robot can move forward while still keeping the line in sight. This speed will be determined during test later in the process. As previously mentioned, an attempt at reducing the time of the entire mission is to map the path, and using this mapping to eliminate parts of the path which would not be necessary, in order return back down to the start position after getting the can. To improve the individual behaviours the relevant PID controllers will be tuned to limit unnecessary side movements and will hopefully increase the speed with which the robot can move.

In terms of the individual behaviours the performance of these will be measured on how well they can perform the intended task. For example to test the ability of the robot to do line following, a number of tests with the robot following a line will be performed to ensure the final solutions consistency. To test the success of the behaviour, tests will be performed with different curvature and corners of a solid line to see in which scenarios the behaviour is successful and determine if this is satisfactory for the application. In these test the speed [m/s] and whether or not robot can successfully follow the line [success/failure] will be recorded. To test the other following behaviours similar tests will be conducted. Tests will also be constructed to test if the correct behaviours are chosen for the situation. To determine how well the robot manages to grab the can, a similar set of tests where the can position is changed slightly between tests to find the limit for when the robot can successfully grab the can. To test the robot's collective behaviour, different scenarios will be used to test how well the individual behaviours can be combined and used to solve the task. In this case what we measure will be the ability to perform the task [success/failure] and time of retrieval [s].