Technische Universiteit Eindhoven
University of Technology

Where innovation starts

Department of Mechanical Engineering

Design document - Embedded Motion Control

Group 8

M.J. van Haren	0953564
C.M. den Hartog	0953184
G. Bijlenga	0950642
R. Dorussen	0968849
E.D.T. Verhees	0950109

Case coordinators:
Dr. Ir. René van de Molengraft

May 6, 2019

Contents

1 Requirements and Specifications

In Table 1.1 an overview of the requirements and subsequent specifications is given.

Table 1.1: *Overview of the requirements and specifications.*

Requirements:	Specifications:	
A) The robot should autonomously complete the 'Escape room'.	1	Rear wheel must cross the finish line
	2	The robot should finish within 5 minutes.
	3	The robot should finish within 2 trials.
B) The robot may not bump into any walls.	4	Distance between the wall and the sensor of the robot must be at least the length of the bumper plus 0,05 meter.
C) The robot may not tip over.	5	The maximum translational speed is 0,5 meter per second.
	6	The maximum rotational speed is 1,2 radians per second.
	7	The robot must stop before changing directions.
	8	The maximum inclination angle should be kept around five degrees.
D) The robot should make progress towards its goal each 30 seconds.	9	The status of the robot must be reported at least each 30 seconds.
	10	The robot should correctly switch in states; only maintain state when still advancing in strategy plan and otherwise switch to an appropriate new state.
E) Software must be easy to set up.	11	The software can be updated by one command.
	12	The software is compatible with cmake/make.
	13	In order to start the software only one executable has to be called.
F) The robot must function robustly.	14	The functions must be scalable.
	15	The definition of data ownership must be explicit.

2 Components

The components of this challenge are apparent on two levels. At a high-level, the components can be seen as the robot, the room and the software. Within these three components, a further division of the components per element can be given. For the room, a further division is the walls, the corridor, the finish line and obstacles. For the robot, there are actuators, sensors, body and hardware. Within the software, a division is made between the world model, the interfaces, the plan, actuator control, monitoring, preception and the Life Cycle State Machine (LCSM). All these aspects are taken into account in the functions, specifications and requirements.

3 Functions

In order to give a comprehensive overview of the required functions, the specifications and requirements, the explanations are done by use of the overview of the design of the software of the robot, as given in Figure 3.1. The core idea is that there exists a World model (WM) that gets updated with the relevant (sensor) data each time the main execution loop is ran. The `main()` of the software comprises a LCSM in which various states can be distinguished, like scanning the environment, driving forward, driving backward, rotating and wall following. In each of the states, different functions which can extract data from the WM, will be called. These functions and the corresponding requirements (R) and specifications (S) that are met through these functions are grouped below.

3.1 Motion

A key asset of the robot is obviously that it is mobile. While driving around, it may not tip over (R.C). Therefore it must stop before significantly changing its direction (S.7). That is, for slight correctional rotational movements the robot does not have to stop entirely, but only adjust speed accordingly, while for larger rotational movements it must first have zero translational speed. Furthermore, the maximum speeds are (S.5) and (S.6).

Rotate: The robot has as a function to rotate. Rotation is only done when the speed is adequately low, thus increasing the odds of not tipping over.

Move forward / move backward: The robot can move with a certain speed in the required direction. This speed will be limited by the specifications of the robot, and the distance is given as an input.

Stop driving: This function stops actuation.

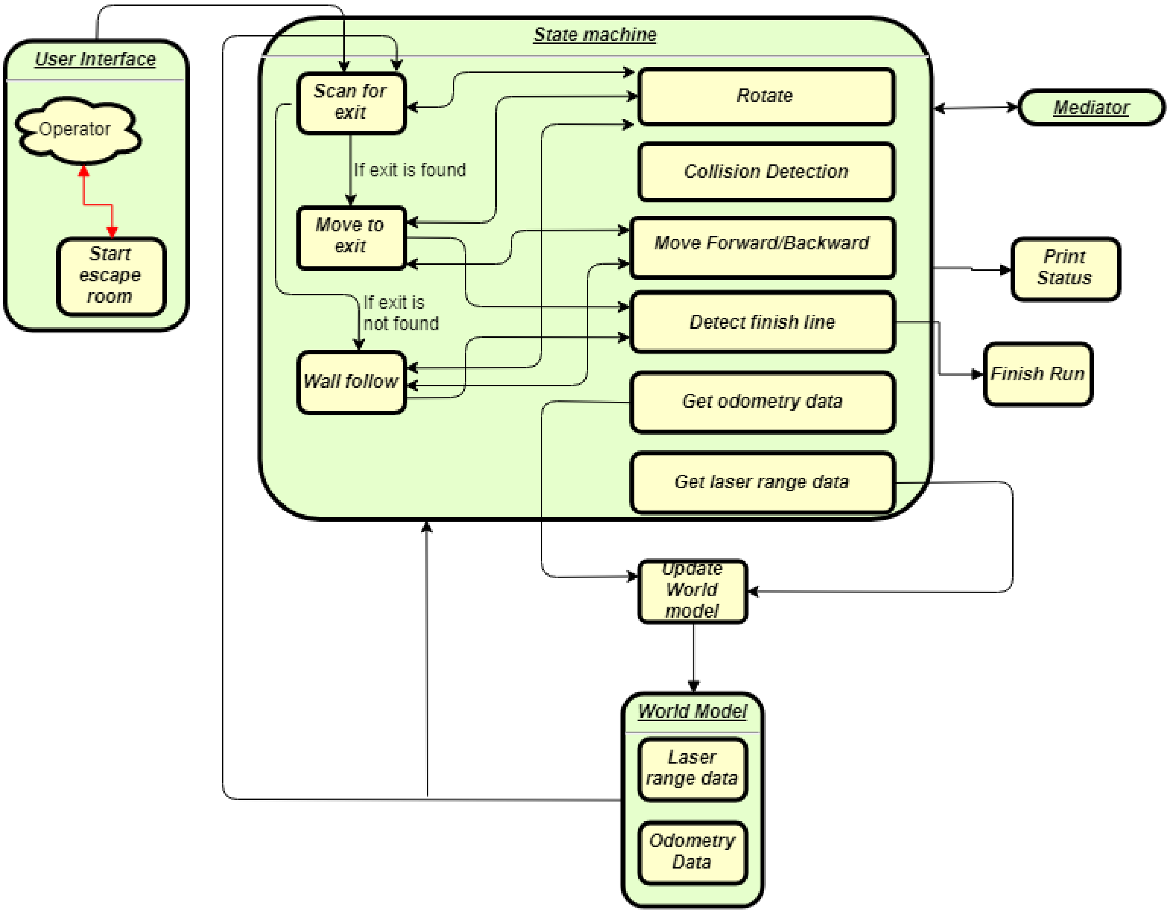

Figure 3.1: *An overview of the software design of the robot*

3.2 Sensors

Sensors are extremely important to monitor the environment, e.g. (R.B). Therefore, (S.4).

Getting sensor data: Functions (*producer*) are built in, which will transfer the raw sensor data to a format which can easily be read by the required functions. With this data, the world model is to be updated to the most up-to-date data. The sensor data will be a range of indices, whereas the data written to WM could correspond with the right values. The data in WM can be accessed by e.g. the wall follow, collision detection and scan functions (*consumer*).

Collision detection: The robot will be able to detect collisions, by interpreting the data gathered from laser range data. A safety margin will be implemented to ensure no collisions will happen.

3.3 Planning

The ultimate goal of the robot is to autonomously exit the 'Escape Room (R.A) (S.2) (S.3), so some planning is essential to ensure that the robot correctly switches states (S.10). Exiting implies that the rear wheel of the robot must be entirely across the finish line (S.1). (The next subsection elaborates on the meaning of a trial (S.3).)

Wall follow: A function is made, which ensures that the robot has the capabilities to follow a wall. This will ensure that the robot will not collide with the wall, while still making progress to find the exit.

Scan for exit: The robot is required to scan for the exit at an unknown location within a room. This function will scan the sensor data for possibilities of an exit, and if no exit can be detected, rotate and scan the other half. If no exit is detected then, the wall following function will be called upon.

Detect finish line: The finish line is required to be detected, which can be done in a multitude of ways. This function will determine when the robot can shut off as it has accomplished its task.

3.4 User interface

A trial is ended straight away if the robot bumps into a wall (R.B), but is also ended if the robot does not make any progress towards its goal for 30 seconds (R.D). Therefore, it is important for the robot to communicate how it is progressing in its strategy plan to the user (S9). Furthermore, the software of the robot must be easy to set-up (R.E), meaning (S.11), (S.12), (S.13).

Print status: A status update of the progress of the robot is given, once required, as well as at every new task it is performing. This function will print the status as information for the user to update to its current status.

Start up and shut down: Start up and shut down functions are to be implemented, in order for the robot to have a start up and shut down protocol, upon, respectively, placement in and exiting the room.

3.5 Miscellaneous

Overall robustness of the robot's behaviour must be guaranteed (R.F). Scalable functions (S.14) and explicit definition of data ownership (S.15) are therefore very important.

4 Interfaces

First, at a high level, the interfaces can be seen as the communication between the user and the robot, and the robot and its environment. The interface between the user and robot works as the user giving the input to the robot, as software, and receiving updates by means of status printing by the robot. The interface between the robot and its environment can be seen as the influence of the environment on the robot and vice versa. The influence of the environment on the robot is done through the sensors, and for the robot on the environment by the actuators present.

Next, within the software there are multiple interfaces, which can be seen in Figure 3.1. Starting with the influence of the environment on the robot, the interface between the sensors and the world model has to pass the information retrieved by the sensors into readable information in order for the world model to be updated. Next, the interface between the world model and the planning has to retrieve the information from the world model, in order for the planning to continue and adjust according to the new information. Then, the interface between the state machine and the planning has to find the information from the planning, and act accordingly, thus fulfilling its task. The interface between the state machine and the printing of the status is required in order to fulfill the higher level interface between user and machine. Lastly, the interface between the state machine and the actuators should take the given information by the state machine and transform it into readable information for the actuators to influence the environment.