4SC020 Design Document MRC

Group 7: Tim van Esch (1235917), Aron Prinsen (1243943),
Thom Samuels (1267566), Naomi de Vos (1233610), Joey Wouters (0813063)

April 30, 2021

1 Requirements

Requirements specify what the system should do. Some requirements are challenge dependent, as the environment and task at hand differ per challenge, while others are general. These general requirements are both given by the client and decided upon by the team. The following requirements are applicable to both challenges, where the first three are given by the client and the others are preferences of the group:

1. Wall clearance: The robot should stay $(0.5 - 2r)/2$ meters away from the wall, where r is the radius of the robot and 0.5 is the minimum width of the corridor of the escape room challenge.
2. Speed limit: 0.5 m/s translational, 1.2 rad/s rotational.
3. Sensible movement: The robot should make a sensible move within 30 seconds.
4. Local path planning: The ability to create a path which needs to be followed, within the same room.
5. Effective path: Used time to complete a challenge should be minimized.
6. Scalability: The solutions should work for different environments.
7. Communication: The robot needs to be able to communicate what it is doing.

For the escape room challenge, there are only two extra needed requirement, both given by the client, which are the following:

1. Challenge completion: The robot needs to cross the finish line, at the end of the corridor, completely to pass the escape room challenge.
2. Maximum time: Efficient computing and control speed is required to be able to complete the escape room challenge within 5 minutes.

The requirements needed specifically for the hospital challenge are listed below. In this list, the first four requirements come from the client, while the others are decided upon by the group, in order to improve efficiency.

1. Challenge completion: The robot needs to visit the cabinets in the given order.
2. Maximum time: Efficient computing and control speed is required to be able to complete the hospital challenge within 10 minutes.
3. Object detection: Unknown and possibly moving objects should be detected. If unexpected objects and/or people are in the way, the robot needs to go by them, avoiding collisions at all time.
4. Orientation with respect to cabinets: The robot should approach the cabinets with the right orientation.
5. Door detection: The robot needs to recognize door openings. If a door is locked the robot needs to look for another entry.
6. Global path planning: The ability to create a sequence of consecutive intermediate points which needs to be visited in order, divided over multiple rooms.
7. Keeping right: In hallways, the robot should stay on the right side if possible.

2 Functions

Functions are requirements that the software needs to meet. These functions can be split up in two types, namely functions that improve the quality of the code and functions that specify what the robot needs to be able to do. The functions that the code needs to meet are as follows:

- Readability: The software needs to be well readable, this includes using indents and comments and best practices.
- Structure: The software needs to have a logical structure, such as dividing the code in multiple files. This also helps improve readability.
- Maintainability: It needs to be possible to maintain the software, this includes that all team members need to be able to work on the software.

The functions that specify what the software needs to enable the robot to do are as follows:

- Perception: Process the LRF data and based on the found distances between the robot and objects, walls and people at a certain angle, which is used to perceive what static and dynamic objects are surrounding the robot.
- Localization: Pinpointing the location of the robot, using the given map of the environment and perception module.
- Path planning: Planning a needed path, based on the given general layout of the environment, together with the given target position and its current position.
- Navigation: Enabling the robot to follow a route from its current location to a target location.
- Collision avoidance with dynamic objects: Detection and reaction to motion of other objects and at all time try to avoid collision with these.
- Alignment: Making sure the robot has the right orientation with respect to the cabinets and hallways.
- Communication: The software needs to be able to communicate with the user.

3 Components

The software consists of multiple components. These components are as follows:

- Localization: The robot needs to localize itself with respect to the environment.
- Perception: The robot needs to perceive its surroundings, including the detection of objects.
- Navigation: The robot needs to be able to navigate through the environment.
- Path planning: The robot needs to plan its movement towards its target position.
- World modeling: The robot needs to make a model of the world for itself, including the dynamic objects.
- System planning: The robot needs to decide which components to use at what time, in order to come up with a strategy for completing the task.

4 Specifications

Specifications specify what the system can do. The following specifications can be set up:

- The robot can detect (the distance to) walls/objects using its Laser Range Finder (LRF).
- The robot can estimate its change in position over time, using the wheel encoders.
- The robot can move in all directions, using its holonomic base with omni-wheels.
- The robot can use speech to communicate with its environment.

5 Interfaces and strategy

Interfaces link the different components of the software. The interfaces can be seen in Figure 1.

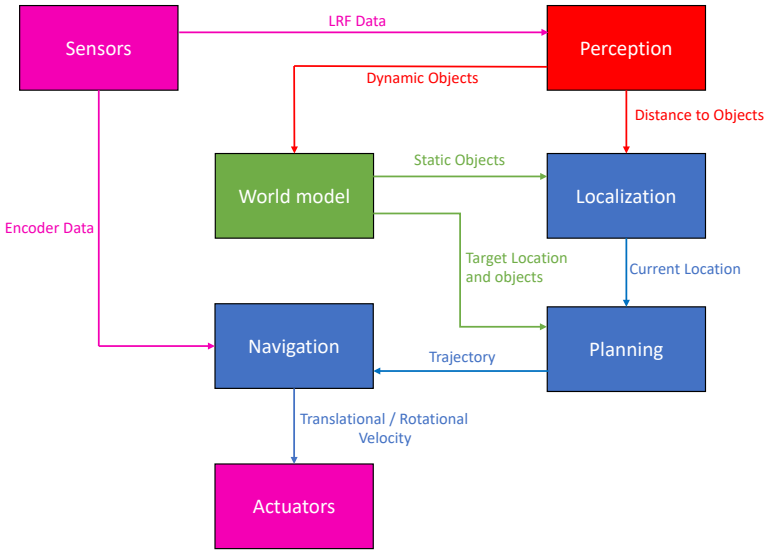

Figure 1: Interfaces between components

The strategy used in order to complete tasks is shown in the state machine, which looks as follows:

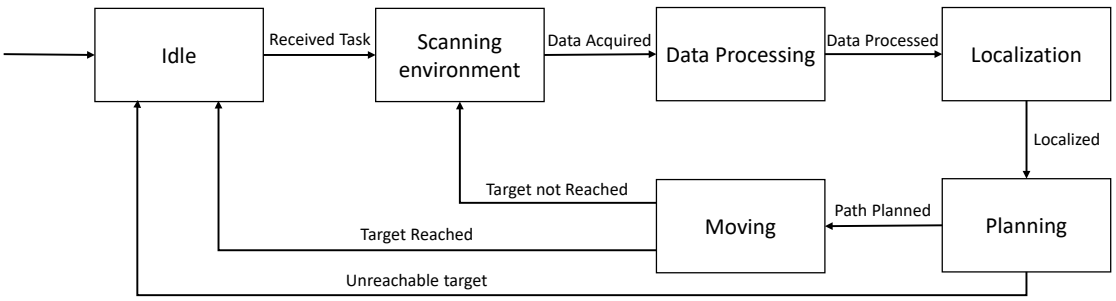

Figure 2: State machine of the robot