EINDHOVEN UNIVERSITY OF TECHNOLOGY

DEPARTMENT OF MECHANICAL ENGINEERING

4SC020 - Embedded Motion Control

Design Document

Tutor:	Hao Liang Chen	
Course:	Mobile Robot Control	
Student:	A.L. Nooren	1553356
	G.J.L. Creugers	0886409
	M.K. Salahuddin	1425986
	M.N. de Boer	0907480
	P. Leonavicius	1471333
	S. Narla	1507125

Eindhoven, June 23, 2020

Contents

1 | Design Document

In the design document the initial design ideas are elaborated for the topics requirements, functions, components, specifications and interfaces. In the next sections these topics are further explained, both the Escape Room and Hospital Challenge are taken into account.

1.1 Requirements

The requirements of a system are defined as the things the system should do. The requirements are divided into task performance, safety and software concepts. Where the task performance requirements include the requirements which needs to be resolved to be able to finish the Escape Room and Hospital Challenge. The safety requirements assure a safe execution of the challenges and the software requirements satisfy the an easy use of the software. The following requirements regarding the three topics should be satisfied by the simulated PICO robot:

- Task performance
  - The robot must be able to recognize target cabinets;
  - The robot is capable of planning a path and is able to adapt its planned path to unexpected circumstances, for instance a closed door;
  - PICO can rotate in place, in order to re-position when in front of a cabinet;
  - Must be able to announce the completion of the current objective;
  - The robot should not be inactive for more than 25 seconds;
  - The robot has to be able to detect static and dynamic object and present them in the world model.
- Safety
  - The robot avoids collisions with static and dynamic obstacles;
  - PICO must obey the limits on translation and rotation velocity;
  - PICO should maintain a 5cm Stopping distance from the obstacle.
- Software
  - The software is started by a single executable;
  - The software can be easily updated;
  - The User-interaction should be minimal and User-friendly.

1.2 Functions

The functions give a rough overview of how a successful execution of the challenges is break down into several sub tasks. The functions are divided into three subareas: input data processing, localization and control. The purpose of the input data processing functions is to process the raw data from the sensors (Laser Range Finder (LRF) and Odometer). After this processing, the raw data is converted to usable data for the other functions. In the localization function, the usable data is further processed towards a belief of the environment and is compared with what should be the actual environment. Also an actual position of the robot in the room is obtained. Finally, the control functions take care of the movement and positioning of the robot. The specific functions for a successful execution of the challenges are described in the next sections.

Input data processing

- Laser range finder interpretation
  The LaserData contains (range_min, range_max), which define what the smallest and largest measurable distances are. Furthermore, (angle_min, angle_max) determine the angle of the first and last beam in the measurement, (angle_increment) is the angle difference between two beams. This data is interpreted in this function and forwarded to the environment mapping functions.

- Odometer interpretation
  The OdometryData contains the displacement (x,y) and rotation (a) of the robot since its start, according to the wheel rotations. This displacement has to be interpreted and converted towards a position value of the robot, which is forwarded to the environment mapping functions.
- Sensor Fusion
  Combining sensory information from the LRF and Odometer sensors can result in a conflicting outcome, for example there can be a conflict about the exact location of the robot. This function can help to have reliable information flow to correlate and deconstruct data.
- Vector map data interpretation
  A function used for structuring data obtained from the provided map of the testing area. To be used as inputs for position estimation and path planning functions.

Localization

- Environment comparison and localization
  This function will make a features based comparison between the expected surroundings based on the vector map and the output of the LRF interpretation function. The function will return the location of the robot (x,y,theta) coordinates on the map. A specific declaration of the to be used coordinate system needs to be further defined.
- Obstacle recognition
  Together with the environment comparison, obstacles has to be recognized. This function determines if a feature obtained from the LRF interpretation function is either a known wall or a unknown obstacle. The comparison will be done in the same manner as the environment comparison, feature based. If an unknown obstacle is determined, it has to be marked as such.
- Position estimation
  Once a location is obtained from the Environment comparison and localization function, the location has to be compared with the OdometryData. After this comparison the robot has the final belief of where it is in the room and the control functions can take care of the movement and positioning of the robot.

Control

- Path planning:
  A function based on a Dijkstra's/ A* / Dynamic programming algorithm. Uses data from the provided vector map and outputs from LRF and Odometry interpretation functions. Constantly recalculates the optimal path based on detected obstacles or changes in the environment such as closed doors. However the interpretation functions return a feature based interpretation of the environment there is chosen to not do a feature based navigation. On the other hand, there is chosen to implement a path planning algorithm to have a more efficient routing through the rooms (in case of the Hospital Challenge). The specific algorithm for path planning has to be considered.
- Movement functions:
  This function is used for describing routines to be sent as inputs for the base controller of the robot.
- Final re-positioning: After the objective position is reached, the rotation of the robot is compared to the required value obtained from the vector map data. Furthermore the function contains a robustness check for the Hospital Challenge, if the robot has the belief it is aligned towards the cabinet there is a check with use of the LRF data if it is actually correctly aligned towards the cabinet. For example, there can be a check if the feature in front of the robot is perpendicular to the front beam of the LRF.
- Signaling function:
  The decision making process of completing a objective is printed on the screen. In this way there can be tracked in which phase of decision making the robot is in, which makes ambiguities in the process more visible to the user. Furthermore. a print output marks the completion of an objective: called once the final state of the path planning algorithm is reached and the correct orientation of the robot is achieved.
- Safety function:
  Constantly running in the background (as a separate process/thread) in order to detect anomalous behavior of the mobile robot and interrupt the operation of the robot if necessary.

1.3 Specifications

The specifications of a system are defined as the specific things the system really can do. In this case specifications are the things PICO is capable of regarding speed limits and specific functionalities which are imposed by both the Escape Room and Hospital challenge.

- Maximum speed of $0.5m/s$ translational, $1.2rad/s$ rotational;
- Maintain a distance of $20cm$ between walls and PICO to make sure the walls will not be bumped, stop and reroute if distance is less than $20cm$;
- Maintain a distance of $80cm$ from any dynamic object to consider the movement of those objects;
- Move at a maximum speed of $0.3m/s$ while tracking a dynamic object;
- Move forward slightly, if robot has been standing stationary for $30s$;
- Position PICO $\pi\ rad$ with respect to the cabinet upon arrival;
- Visit the cabinets in the required order.

1.4 Components

The components give a overview of the hardware of PICO, containing the sensors, actuators and computation unit, a global outline is given in this section.

Sensors

- Laser range Finder
  - Provides a set of points in polar coordinates relative to the PICO robot.
- Wheel encoders
  - With this distance translated and rotated can be measured, is however highly sensitive to noise and will require filtering.

Actuators

- Holonomic base wheel
  - It can realise the required degrees of freedom - translation in x and y and rotate about the z-axis without any position level constraints.

Computation unit

- Containing the software module that drives the robot and lets all other components communicate

1.5 Interfaces

In this section the interface design for the robotic system is displayed. The interface shows the boundary between the components of PICO (Laser Range Finder and Odometer) and the exchange between the software components Localization, Control and Actuation. The interface design is shown in [FIGURE XX].

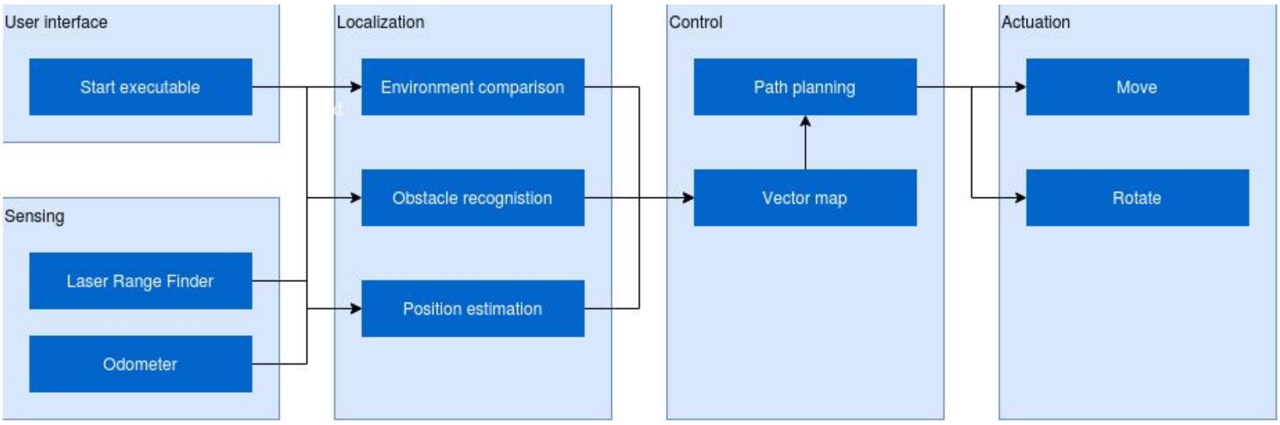

Figure 1.5.1: The interfaces for the robotic system.