

4SC020 EMBEDDED MOTION CONTROL

GROUP 2

Initial design idea

D.J.M Bos *1037038*

C. Butt *1275801*

D. van Dinther *0848638*

L.G.L. Janssen *0895324*

R. Louwers *0843128*



Tutor:

Wouter Houtman

May 11, 2018

1 Introduction

The main goal of this course is to program the PICO robot so that it can map a ‘hospital’ floor and find an object within it. The programming language used is C++ and the software is ran in Ubuntu 16.04, which is the same version of Ubuntu that is running on the robot. For the initial design, the requirements will be discussed as well as the functions and their specifications. Moreover, a list of possible components and a suiting interface is designed.

Note that this is an initial design towards the hospital competition which has overlap with the escape room competition. Some distinction will be indicated.

2 Requirements

- Execute all tasks autonomously.
- Perform all tasks without bumping into a wall.
- Perform all tasks without getting stuck in a loop.
- Finish both challenges as fast as possible, but at most within 5 minutes.
- Escape room competition:
 - Locate the exit and leave the room through this exit.
 - Keep a minimum distance of 20 cm to the walls or any other objects.
- Final competition:
 - Explore and map the hospital rooms.
 - Use the map to get back to the starting position and park backwards into the wall behind it using both the map and the control effort.
 - Use a map to find and stop next to the object that is placed in one of the rooms after the first mapping.
- Nice to have: Use the provided hint to be able to find the placed object directly without searching.

3 Functions

The functions that are going to be implemented are subdivided in three different types of functionality. First of all there is the lowest level functionality, which will be referred to as the motion functions. These functions will be used to interact directly with the robot. This interaction will be specified more precisely in the function overview that can be found below. The mid level functionality, which will be referred to as the skill functions. As the name mid level functionality already suggests these functions act on a more abstract level and are built to be able to perform a specific action. All the actions that will be implemented in the skill functions can also be seen in the overview below. Finally there is the high level functionality, which will be referred to as task functions. These functions are the most abstract and are structured to carry out a specific task, which consists of multiple actions.

To give a full overview: in a task function a specific order of actions will be specified in order to be able to fulfill the task. The actions that the robot can perform, and are therefore available to be called by the task functions, are specified in the skill functions. The motion functions are used in the skill functions to be able to invoke physical behavior by the robot.

- Motion functions
 - Actuation: Using the actuation function PICO is given the command to drive in a specific direction or to turn in a specific direction. The input to this function is a movement vector which specifies in which direction and how far the robot should move. The output is actual robot movement according to this vector and the control effort needed for this movement.

- Obtain information: Using the obtain information function sensor readings can be requested. The input for this function is the request of a (sub)set of sensor data from a specific sensor(set), either the Laser Range Finder (LRF) or the odometry encoders.
- Skill functions
 - Path planning: Using the path planning function a (set of) movement vectors is defined on the basis of the map to reach the defined goal. The input for this function is the current map and the goal that should be reached. The output of this function is a (set of) movement vectors.
 - Object avoidance: Using the object avoidance function it is ensured that PICO does not hit any objects. The input for this function is the LRF sensor data and the movement vector. If the movement vector prescribes a movement that collides with an object that is detected by the LRF the movement vector is adjusted such that PICO keeps a safe distance from the object. The output of this function is therefore an (adjusted) movement vector.
 - Mapping and localization: Using the mapping and localization skill a map of the environment is built up and PICO's location in this environment is determined. The input for this function is sensor information and actuation information that can be obtained from the motion functions. Using the LRF data PICO's environment can be defined in "blocked" and "free" area's, places in which obstacles are present and absent respectively. Using the odometry encoders and the control effort the location of PICO can be tracked in the environment. The output of this function is a visual map, that is expanded and updated every time this function is called.
 - Semantic mapping: Using the semantic mapping function extra information is added to the basic environment map. This function has as input the basic environment map. The surface of this map will be split into different sections which can have one of two different types of identifiers assigned to them: room or hallway. Both the splitting of the sections and the identifications of the sections will be based on shape fitting. If a rectangular shape can be fitted over (a piece of) the basic map, based on the LRF, this (piece of the) map will be defined a (separate) section with the identifier room. Note that each section with the identifier room will be given a separate number to be able to distinguish the different rooms from each other. This is necessary in order to be able to use the hint that will be given for the final challenge. The output of this function is a visual semantic map that is updated every time this function is called.
- Task functions
 - Exit a mapped room: This is the escape room challenge. In this task all motion skills and all skill functions, except the semantic mapping skill, will be used.
 - Semantically map an entire floor and return to the beginning position: This is the first part of the final challenge. In this task all motion and skills functions will be called.
 - Find an item in the semantically mapped floor, preferably using the provided hint on the position of the item: This is the second part of the final challenge. In this task again all motion skills will be used and all the skill functions except the semantic mapping skill.

4 Components

To be able to execute the challenges the PICO robot, already mentioned above, will be used. The following hardware components will be utilized:

- Actuators:
 - Holonomic base with three omni-wheels.
- Sensors:
 - Laser range finder (LRF): To detect the distances to objects in the environment.
 - * Range: To be determined
 - * Field of view: 270°
 - * Accuracy: To be determined

- Wheel encoders: Combined with the control effort these encoders keep track of the robots position relative to its starting point. (This has to be double checked with the position in the map, since the encoders easily accumulate errors.)
 - * Accuracy: To be determined
- Computer
 - Running Ubuntu 16.04.

5 Specifications

The specifications are based on the requirements

- The maximal transitional velocity of PICO in any direction is $0.5 [m/s]$.
- The maximal rotational velocity of PICO is limited to $1.2 [rad/s]$
- PICO should not stand still or make no sensible movements for periods over $30 [seconds]$.
- Escape room competition:
 - PICO has to finish the Escape Room within $5 [min]$
 - The width of the corridor and the openings in the walls are between $0.5 [m]$ and $1.5 [m]$.
 - The shape of the room is rectangular.
 - The exit is located more than $3 [m]$ into the corridor.
- Hospital competition:
 - PICO has to finish the Hospital within $5 [min]$
 - The width of the corridor and the openings in the walls are between $0.5 [m]$ and $1.5 [m]$.

6 Interfaces

The interfaces describe the data that is communicated between the several parts of code that will be developed. The functions that are discussed in Chapter 3 are all part of the interface, however there is one other important piece in this interface: the world model. The world model will contain the most recent (semantic) environment map with the location of PICO included in it. The motion, skill and task functions can monitor and request the information that is stored in this world model and use it to properly execute their function. A graphic showing the interface between the world model and the tasks, skills and motions is shown in Figure 1.

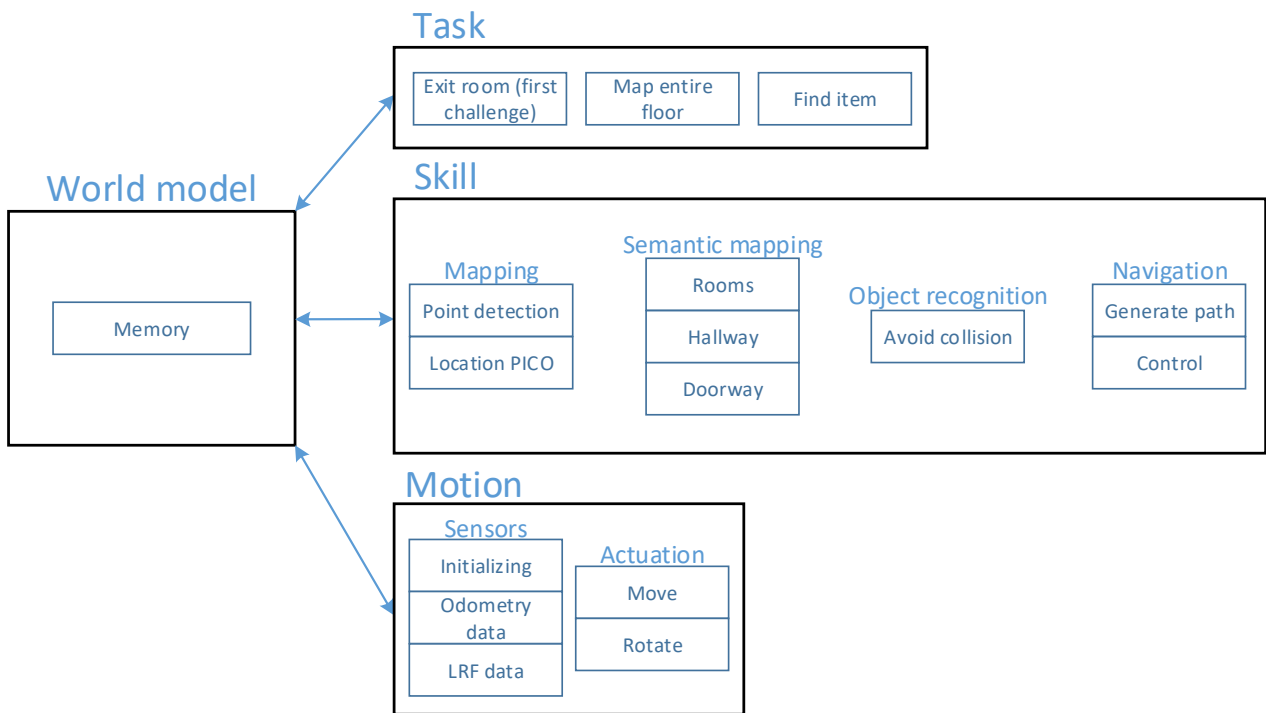


Figure 1: Visual representation of the system's interfaces.