Department of Mechanical Engineering

4SC020 - Mobile Robot Control

Design Document

Quartile 4: 2020 - 2021

Supervisor:

ir. M. Muñoz Sánchez	m.munoz.sanchez@tue.nl

Students:

Y. Wu (1623141)	y.h.wu@student.tue.nl
D.Y.B.M. van Berlo (1242645)	d.y.b.m.v.berlo@student.tue.nl
V. Reinders (1706837)	v.reinders@student.tue.nl
Tim van de Ven (0948317)	n.c.m.v.d.ven@student.tue.nl
D. Roordink (0863456)	d.roordink@student.tue.nl
M.P.J. van Duijnhoven (1510266)	m.p.j.v.duijnhoven@student.tue.nl

Eindhoven, May 4, 2021

Contents

1 Requirements

All requirements/attributes (table 1.1) are derived from the competition descriptions and categorized into Functionality, Usability, Reliability, Performance and Supportability - FURPS. Each attribute is prioritized with MoSCoW (M = must have, S = should have, C = could have, W = will not have).

Code	Priority	Description
Functionality		
F1	*M*	PICO shall be autonomous.
F2	*M*	PICO shall be able to explore the world/be aware of its surroundings.
F3	*M*	PICO shall avoid collision with static and dynamic objects
F4	*M*	PICO shall never get stuck in a deadlock.
F5	*S*	PICO shall never get stuck in a livelock.
F6	*S*	PICO shall signal when an object (cabinet) is found.
F7	*C*	PICO shall position itself in front of an object, facing the object.
F8	*C*	PICO shall stop when the task is performed successfully.
F8	*C*	PICO shall detect static and dynamic objects and present them in the world model.
F9	*M*	PICO shall save a snapshot of the laser data when in front of a cabinet.
Usability		
U1	*S*	PICO shall be controllable by a single command, that also starts the software.
U2	*C*	When PICO fails to execute a task, a clear error messages shall be given.
Reliability		
R1	*M*	PICO shall map different environments and navigate through them.
R2	*S*	When an error is raised, PICO shall be able to restart at its current location.
Performance		
P1	*S*	PICO shall not exceed speed (0.5 m/s) and rotational speed (1.2 rad/s) limitations.
P2	*M*	PICO shall finish its tasks within the time limit.
P3	*M*	PICO shall not stand still for more than 30 seconds.
Supportability		
S1	*S*	The code shall be clear, structured and documented such that maintenance is simple.

Table 1.1: Requirements for PICO according to FURPS and MoSCoW.

2 Components & Specifications

Code	Object	Specification
PICO		
SP1	Dimensions	0.41 x 0.35 m
SP2	Weight	8 kg
PICO Components		
SP3	Laser range finders (LRF)	Range between 0.1-10 m
SP4	Laser range finders (LRF)	230 degree FOV
SP5	Wheel encoders (odometry)	-
SP6	Holonomic drivetrain	-
Environment		
SP7	Hallways	∼1.5 m
SP8	Doors	0.5-1 m
SP9	Cabinets	Shaped as a rectangular box

Table 2.1: Components and specifications

The most important components and specifications are given in Table 2.1.

2.1 Software

The code is written in C++ and executed on Ubuntu 18.04, which is used as operating system. Ubuntu will run both the simulator and the robot control software. For communication with the simulator additional classes are provided.

3 Interface

World model
The world model displays all information of interest to succeed at the given task. PICO uses the world model together with the strategy to make decisions on performing the task at hand.

1. World map (updating): Static layout of the explored part of the map.
2. Perception data: The current perceptions from the LRF.
3. Positioning: The current position and angle w.r.t. the reference (using odometry).
4. Path: The trajectory from the starting position to the current position.
5. Flags; items of interest: Found cabinets and obstructions.

The competition strategy
The strategy is the method by which the goal is intended to be reached. The goal is to make decisions based upon the objectives, the updated map and the current state. The decisions made will to either continue its current objective or to change it to a new objective. Based upon that decision, PICO can determine a path that will fulfill its objective and follow that path, or interact with a cabinet in order to fulfill its objective.

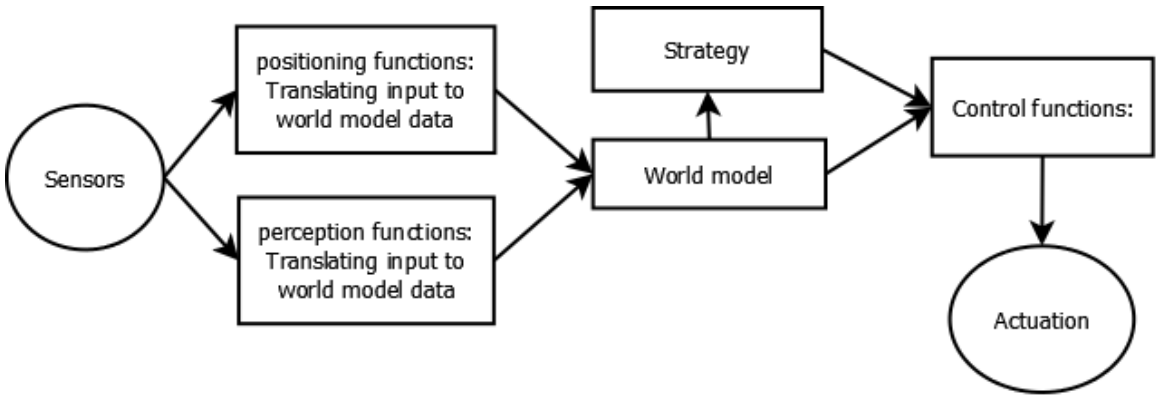

Figure 3.1: Initial interface idea

4 Functions

To achieve the functionality as described in the requirements, a flow diagram (Figure 4.1) is developed to describe the behavior of PICO. The diagram takes into account the provided hardware and applies to both the escape room competition and the hospital competition. Each step is described below and is split up into one or more functionalities where needed.

1. **Initialize:**
   PICO will load sub-modules and configure itself to make sure the system is in a good condition to perform tasks.

2. **Localize:**
   PICO will scan the surroundings, determine its displacement and update the map.

3. **Decision making process:**
   PICO will decide if it should continue its current objective or change it. It does this based on the updated map, its current state, and the objectives. PICO then decides if it should move (path planning and path following), interact with a cabinet, or stop.

4. **Path planning:**
   PICO will determine the path to reach the destination and set up driving parameters. It will consider its environment such as know obstacles and blind spots.

5. **Path following:**
   PICO will move along the generated path towards the destination.

6. **Interact with cabinet - for hospital competition only:**
   PICO will give a clear sound signal and save the laser data after it has positioned itself in front of a cabinet. This step represents picking up and delivering medicine, which is not possible with the available hardware.

7. **Stop:**
   Once all objectives are completed, PICO stops at the final destination and signals that it has executed the objectives.

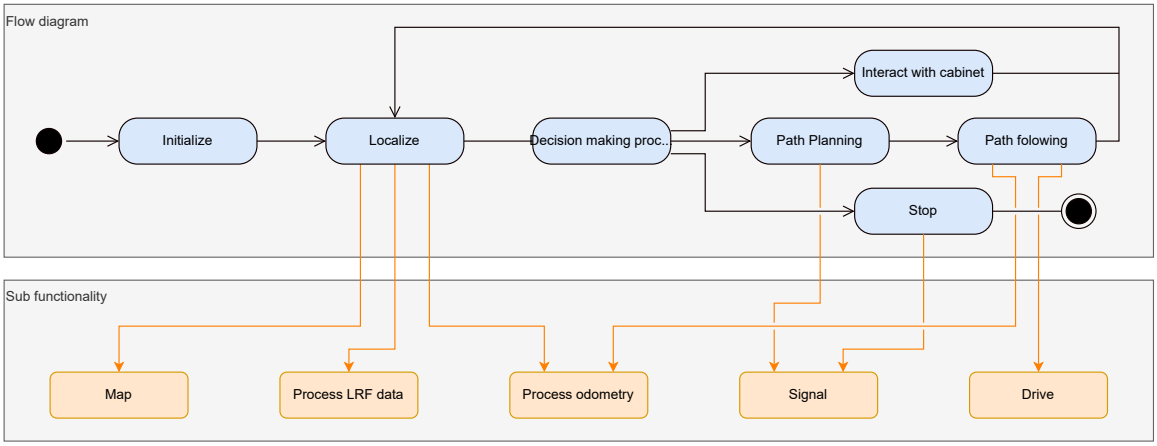

Figure 4.1: Flow diagram for both challenges