Design Document

4SC020 - Embedded Motion Control 2019

Group 5

Group Members:

Muliang Du	1279874
Shubham Ghatge	1316982
Winston Mendonca	1369237
Yi Qin	1328441
Robert Rompelberg	0905720

Group Tutor:
Hao Liang Chen

Lecturer:
Prof. Dr. René van de Molengraft

Dept. of Mechanical Engineering
Eindhoven University of Technology

May 6, 2019

1 Introduction

The aim of this project is to program the PICO robot to autonomously move around while avoiding walls and other obstacles and thus be able to navigate a closed environment comprising walled rooms, exit/entry points and corridors. In the Escape Room challenge (figure 1) the primary goal would be to exit the room from any given position. In the Hospital challenge, the aim would be to move between rooms to cabinets contained within the rooms. This document briefly describes the functionality and software design plan for PICO to be able to complete the given navigational tasks.

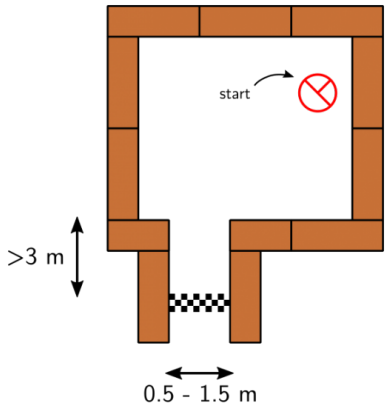

Figure 1: Sample Environment Map for Escape Room Challenge

2 Requirements

The following performance requirements are expected to be met for the Escape Room challenge:

1. PICO must escape the room from any initial position
2. PICO has successfully escaped the room when its rear wheel crosses the finish line at the end of the exit corridor
3. PICO must complete the task within 5 minutes and must not remain idle for more than 30 seconds

To fulfill the above, and keeping the basic aim of the Hospital challenge as stated in the Introduction, the following functional requirements have been identified for the given challenges:

1. Determine its starting position with respect to its surroundings (walls, obstacles, etc.)
2. Translational and Rotational motion within the set constraints (see Section 5)
3. Identification of target position/destination such as entry/exit point or cabinet
4. Trajectory planning to identify the quickest path from current position to destination
5. Obstacle avoidance
6. Environment mapping (if environment map is not already provided)

3 Functions

The tasks that PICO has to perform can be modularised into the following functions:

1. Configuration: Before PICO can begin navigating its environment, its sensors and actuators need to be appropriately initialized and ready to receive instructions or transmit valid sensor data. Any additional variable/data object memory allocations will also take place during the configuration phase. This phase would typically run only when PICO is first powered on.

2. Sensing: PICO's sensors are constantly analysing its environment in terms of its distance to nearby objects, as well as the states of its actuators in terms of the amount translational and rotational motion, control effort for actuation, etc. This data is saved into a "World Model" so that subsequent functions can access these sensor readings for further processing.

3. Environment Mapping: PICO needs to be capable of recognizing and remembering its environment as it progresses through its tasks. This will give it the ability to "know" its position within its operating environment, by recognizing and distinguishing between walls, corners, exit/entry points, etc., and on a higher level, to even identify how to return to a room that it has already visited. In situations where no map is provided before hand, the Environment Mapping function will create an environment map of PICO's surroundings in the World Model using the data previously recorded in the World Model set, and this map could then be used by any path planning or motion output functions.

4. Trajectory Planning: With the generated map and sensor data, PICO can evaluate the control action it needs to take to achieve its current objective. For example, in the Escape Room challenge, this could be the trajectory it would have to traverse to reach the exit of the room. Trajectory planning also ensures that the planned traversal path keeps PICO from bumping into any walls or obstacles.

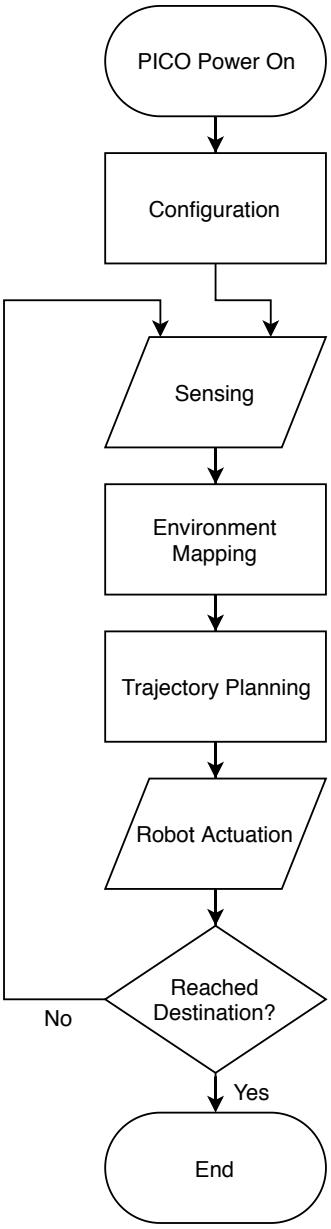

Figure 2: Function Flowchart

5. Robot Actuation: This function will handle all tasks pertaining to PICO's motion. Using the holonomic base, this function will determine the type of motion (translational, rotational or a combination of both) PICO has to undertake to follow the provided path from the Trajectory Planning function. A higher implementation of this function would also include actuating PICOs motors with smooth acceleration profiles to avoid uncontrolled jerks in motion.

The simplest approach to using the above functions to solve the given challenges is by sequentially arranging these functions as shown in figure 2. At a higher level of implementation,

they can also be processed in a manner of parallel execution using multiple software threads providing/receiving data to/from the software's primary loop.

4 Components

PICO Telepresence Robot by Aldebaran:

1. Sensors:
   (a) Proximity measurement with Laser Range Finder (LRF)
   (b) Motion measurement with Wheel encoders (Odometer)
   (c) Control Effort Sensor
2. Actuators:
   (a) Holonomic Base - Omni wheels that facilitate 2D translation and rotation
3. Computer:
   (a) Intel i7 Processor
   (b) OS: Ubuntu 16.04

5 Specifications

1. Laser Range Finder (LRF)
   (a) Linear Range: 0.1 to 10 meters
   (b) Panoramic range: -2 to 2 radians
   (c) Panoramic Resolution: 0.004004 radians
2. Holonomic Base:
   (a) Maximum Linear speed: 0.5 m/s
   (b) Maximum Rotational speed: 1.2 rad/s

6 Interfaces

1. Communication with PICO: All software uploads to the PICO robot will take place through GitLab.
2. Communication within software functions: As described in the Functions section, it is necessary to exchange information from different functions. The main interface between the functions stated above will be the World Model which will contain the raw data recorded by the Sensing function, the processed information from the Environment Mapping and Trajectory Planning functions and the planned trajectories for the Locomotion function.