Technische Universiteit Eindhoven University of Technology

Embedded Motion Control - Design Document

Group 7
Piyush George , 1279637
Sam Roovers, 0833049
Siebrand Keja, 0886654
Milan Haverlag, 1022624
Tutor: Wouter Kuijpers

May 11, 2018

1 Requirements and Specifications

The requirements are derived based on description of the robot, Escape room and Hospital mission description. The FURPS principle has been adopted here, whereby they have been categorized into Functionality, reliability, Usability, Performance and Supportability.

The requirements have been described below and the associated specifications wherever applicable have been placed in parenthesis alongside. For referring later, requirements are classified as A for general, B for escape room and C for hospital specific.

Functionality

- PICO must be autonomous
  - PICO must be able to move around:
    - Translational movement forward, sideways or a combination of both. (maximum allowable translational speed is 0.5m/s)
    - Rotational movement .(maximum allowable speed is 1.2 rad/s).
  - PICO should be able to detect corners (the corners between two walls are approximated at 90 degrees).
  - PICO must be able to detect its surroundings, walls and exits.(Horizontal view for LRF is 230 degrees, range is 0.1 to 30m).
- PICO should not be standing still for a long period of time (PICO should not stand still longer than 30 seconds).
- PICO must be able create a map by exploring the room and park itself backwards at the starting position .(The path following should be done with an error less than 5cm).
- PICO must drive through the opening/exit in the room (Width of the corridor will be between 0.5 - 1.5 m).
- PICO must stop when it has completely crossed the finish line.(At the exit, the finish line is located more than 3 m into the corridor. The walls that can be used to align PICO will be a little bit longer)

Usability

- The system shall keep a log of the sensor inputs and control outputs that are given.
- System shall give a real time representation of the map that is being generated.
- System shall be able to show the next actions it's going to take next in advance.

Reliability

- The system should account for the stopping distance for PICO that can be achieved on braking to avoid bumping. (based on the mass and inertia of PICO).
- PICO must not hit any object in its surroundings. (Slight toughing of wall is permissible, but no head on collision).
- The motion control should be always stable.

Performance

- The system should be able to finish the challenge with the least time. (Maximum allowable time is 5 minutes)
- PICO must be able to detect an object that is put in the room. (based on the map generated it should be able to detect the object, which here would be the additional object),(PICO should stand close to the identified object and stop moving, defined the end of the hospital challenge).
- PICO has 2 attempts to solve the challenges.

Supportability

- Software must be be written in C++.
- System should be compatible to the starting condition, in whatever orientation PICO starts, it should adjust itself and drive through the challenge.
- System should avoid deadlocks, should not get stuck in an infinite loop.

2 Functions

The functions of PICO are derived from the requirements. The structuring of the functions is based on slide 9 of the presentation by Herman Bruyninckx on the 4th of may 2018 and the Task-Skill-Motion Framework.

Skill functions (Monitoring & Control)
First the basic *skills* functions, these are the functions directly controlling the components of PICO, which is low level control.

1. Actuate functions (Control)
   - A function for Translational Motion of PICO.
     - move the robot translationaly in any direction at speeds up to 0.5m/s
   - A function for rotational motion of PICO.
     - rotate the robot at speeds up to 1.2rad/s
2. Detect functions (Monitoring) : There should be a function to detect the surroundings with help of the LRF
   - Measure the relative distance between PICO and objects in a cone of 230 degrees in front of PICO

World model functions (Perception) Functions that can process the obtained data from the detect functions into a *world model*.For example, PICO identifies the entrance to a door.

- A function determining the relative position of PICO:
  - The position of PICO relative to an object
  - (**C:** Hospital specific) The position of PICO relative on the map
- (**C :** Hospital specific) A function that distincts between the object which should be detected and walls.
- (**C :** Hospital specific) A function to build a map
  - Function to determine the position of a wall
  - Function that clasifies a combination of walls as a room
  - Function that determines exits in a room

Task functions (Plan) From a higher level of control, depending on the current task (escape room or hospital challenge), and the status of the world model, PICO should take certain actions.

- Function keeping track of the motion of PICO, preventing Pico from not moving for 30 seconds.
  - Compare the control effort and odometry signals to determine whether PICO is moving as desired.
- Function to avoid a bump into the wall or object of PICO.
  - Use the laser scanner to determine if PICO is close to a wall and steer PICO away from a wall if it comes to close.
- Function to scan the room
  - Use the laser scanner to look for objects in the room.
- Function to determine an exit in the room
- Function to move PICO to a particular exit (There are multiple exits in the hospital contest)
- A mediate function, which can set functions to a lower proirity by limiting its computational efforts
  - This function could influence the accuracy of the world model construction, by tracking how much time is left to complete the challenge
- (**C :** Hospital specific) A function to drive to specific points on the map

3 Components

PICO Robot has the following components :

- **Sensors** : Laser Range Finder(LRF), determines the distance to an object. This is done by estimating the time taken for a laser pulse in a narrow beam to reflect from an object and reach back to the sender.
- **Wheel encoders (Odometry)** : The wheel encoder enables one to measure displacement and the rotation of the robot with respect to it's initial position, from which the translational and rotational velocity can be estimated.
- **Actuators** : PICO has omni wheels that allows omnidirectional translational and rotational motion.
- **Computer** : PICO uses Ubuntu 16.04 as an operating system. PICO is powered by an Intel i7 processor.

4 Interfaces

Figure 1 shows the basic design for the software architecture. The colored backgrounds represent classes, the white text blocks within these backgrounds are the functions present in each class. The *Decision maker* class contains the main function, with the so called *even loop*. The main calls functions of the *Task functions* class, based on information(data) provided by provided by world model class. The main function will be different for the hospital and escape room challenge since different tasks need to be performed. It is convenient to make use of such a software architecture, because only the main function needs to be altered to let PICO perform different challenges.

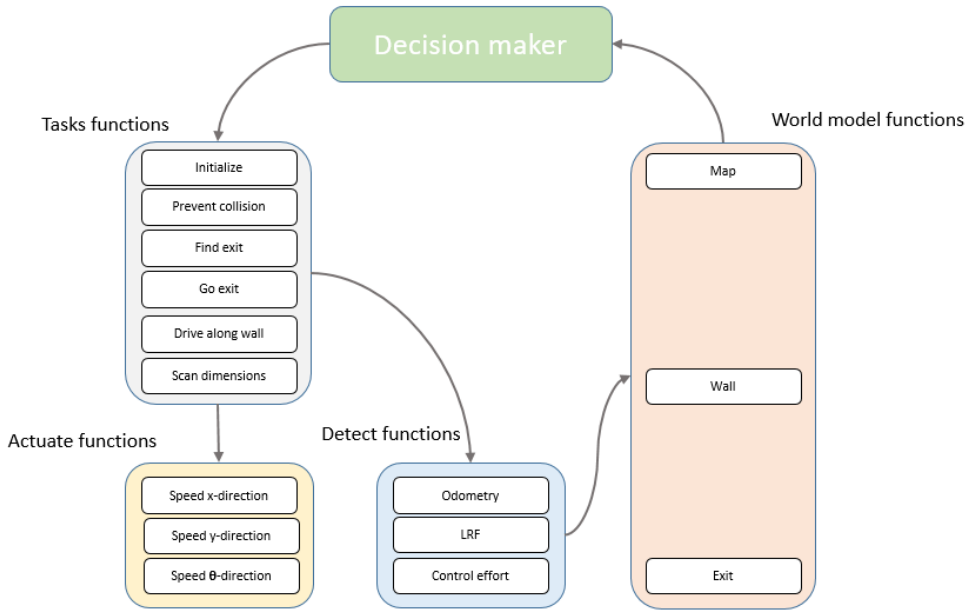

Figure 1: The interfaces between the Skill, World and task functions as discussed in Section 2.