Design Architecture

This documents regards the requirements, system specifications, functions, components and interfaces of the PICO robot and the course embedded motion control 4SC020.

Requirements

The requirements for these challenges were found and grouped using the FURPS method (Functionality Usability Reliability Performance Supportability). An overview of the categories and requirements can be found in Table 1.

1. The Pico/Taco robot must solve the corridor and maze challenge.
    a. The corridor challenge must be solved within two attempts of maximum 5 minutes total.
    b. The maze challenge must be solved within two attempts of maximum 7 minutes total.
    c. Must not be at a standstill for more than 30 seconds.
    d. Maze solving algorithm must be implemented. It must be taken into consideration that the robot might start/be inside a loop or encounter a dead-end.
    e. Entire rear wheels must be across the finish line.
2. The Pico/Taco robot must take the desired turns, not drive into walls and recognize doors.
3. The Pico/Taco robot must operate autonomously.
4. The Pico/Taco robot must be able to send a signal to open the doors.
5. Only one executable has to be called to start the software.
6. Use given hardware
7. Design must be reliable.
    a. To validate the design it will be tested and simulated.

Table 1 Overview of the requirements and their FURPS category

Requirement	F	U	R	P	S
1. Solve corridor/maze	x			x	
2. Navigate appropriately	x			x	
3. Operate autonomously	x			x	
4. Send signals				x	
5. Execution of software		x			
6. Use given hardware					x
7. Reliable design			x		

Specifications

The specifications can be divided over different contexts, this can been seen in Figure 1. The following contexts were used:

Challenge: Challenges are Corridor and Maze Challenges. In both challenges PICO have to calculate and follow true trajectory for exit without any collision to walls within 5 minutes for Corridor and 7 minutes for Maze Challenge. In addition to a corridor, Maze Challenge contains loops, door and complicated maps.

Environment: For a better representation of the environment both geometric and semantic maps are made. Furthermore the geometric map is required to create the semantic map and detect doors.

Skills: Skills of the PICO robot that are satisfactory/sufficient in order to find its way out of the maze can be categorized into 4 groups: Movement skills provided by omni-wheels allow PICO to change its position and orientation along the maze, which is the most important capability to complete the challenges. Observational skills provided by the sensors allow PICO to detect its surroundings (corridors, corners, doors, etc.) which provides awareness to make other skills useful. Navigational skills come from the interpretation of the detected data through the selected maze solving algorithm, that provides a dynamically forming path (a global path is not possible since we don't have access to the entire maze at a given time) to make a way out of the maze. Other skills worth noting are enabled by the software running in real-time: Interacting with the environment when a relevant observation is made and the autonomous operation enabled (also limited) by the finite-state-machine.

Tasks: This contains the tactical actuation, operational actuation and detection. The tactical actuation defines tasks in order to complete the goal, while the operational actuation defines tasks to control the robot based on the detection task. The detection task converts received data into useful information and sends it to the operational actuation which decides upon a new control action.

Hardware: This contains the physical hardware of the robot. These are the sensors and actuators that have previously been discussed as well as the computer.

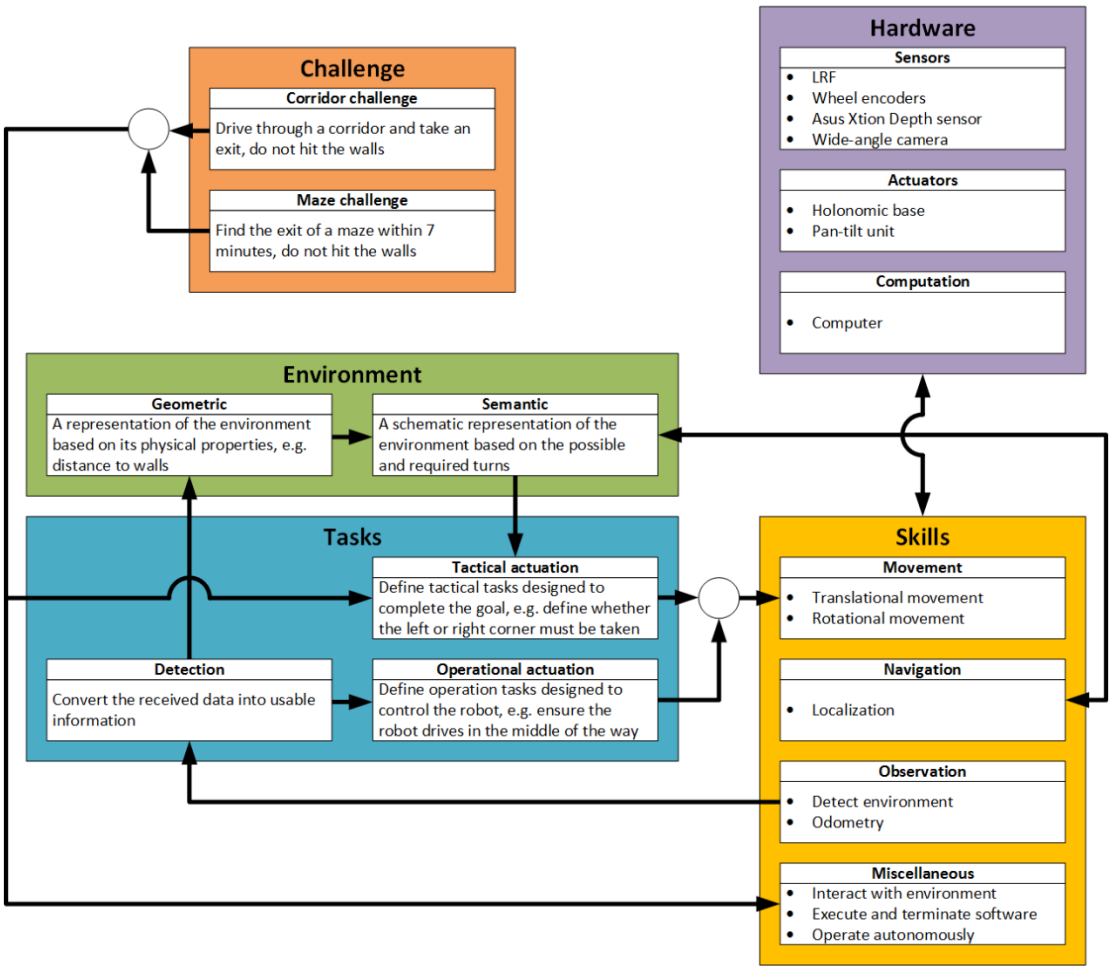

Figure 1 Schematic overview of the system specifications

Functions

The following functions have been specified for the PICO robot.

- Basic movements in order to navigate through the corridor and maze challenge. These movements involve:
  - Forward and backward movement
  - Left and right rotation
  - Wait in front of doors or dead end
- Reading sensor data
- Autonomous decision making
- Create a mapping and localization
  - Include odometry
- Upon competition of the corridor and maze challenge the robot should terminate autonomously

Components

The Pico/Taco robot is consisted of the following sensors and actuators:

- Sensors:
  - **Laser Range Finder**: By measuring the time of flight of the laser pulses (sent in a narrow beam) that reflect off the targets, LRF is used to detect the walls and the doors. Laser data is represented in polar coordinates, giving the distance and the angle to the obstacles in the range of the sensor. Note that the mapping capabilities of the robot is highly related to the head-on direction of PICO.
  - **Wheel encoders**: The wheel encoders will allow the robot to estimate its position relative to the starting location. Odometry alone is not accurate enough, but together with the LRF it can determine the traveled distance and direction. For the odometry to be used effectively, rapid and accurate data collection, equipment calibration and processing is required.
  - **Asus Xtion Depth sensor**: Xtion enables color image sensing and two lenses which provide depth detection. The distance of use is between 0.8 and 3.5 meters. This range specification provides us to avoiding waiting whole sliding duration of the door for detecting the door. Although the time duration of door sliding is 3 seconds, Xtion notices starting of the sliding and understand whether it is a door or a dead-end at the beginning. Therefore we will save approximately 3 seconds for each dead-ends that the PICO faces.
  - **170° wide-angle camera**: The wide range camera allows the robot to have a better perception of the environment. This will help the robot decide what direction is available to take.
- Actuators:
  - **Holonomic base:** Three omni-wheels are used to move the robot. The robot can achieve speeds up to 0.5 m/s in translational direction and a rotational speed of 1.2 rad/s. If the robot rotates and translates simultaneously the turning radius will be just above 40 cm. It is however currently unknown if the hardware and preprogrammed software allows this.
  - **Pan-tilt unit**: A pan-tilt unit is attached to the head; this allows the depth sensor to be rotated over two axes and aimed in a desired direction.

It also consists of a computer with an Intel i7 processor which is running on Ubuntu 14.04.

Interfaces

The following interfaces were defined between the different contexts:

Challenge – Tasks: Completing the challenges is the biggest picture. In order to achieve that goal, challenges need to be divided into smaller tasks that need to be fulfilled one step at a time.

Challenge – Skills: Challenges must be solvable with the existing skill set of the robot at hand (feasibility/existence of a solution)

Task – Environment: provides the information for building the geometric structure of the environment.

Environment – Environment: map geometric representation of the environment onto schematic representations that are proper for applying maze solving algorithms.

Environment – Tasks: sends the schematic representation of the environment data for deciding true movement in order to complete the challenge.

Environment – Skills: This interface provides information from the semantic model of the environment to the navigation skills which allows the robot to determine the location on the map.

Tasks – Skills: This interface selects the appropriate movement skills according to the information from the tactical and operational actuation and also sends information from the observation skills to the detection task.

Tasks – Tasks: The task of detection is linked to the task of operational actuation. When the surroundings are detected the robot can be operated accordingly; avoid bumping into walls and driving in the middle of the pathway.

Hardware – Skills: The hardware enables the skills of the robot. For example, movement skills are not possible without the holonomic base.