DESIGN DOCUMENT GROUP #2

Das, A.

Dávalos Segura, S.

Hristov, G.S.

Irigoyen Perdiguero, N.

Kuijpers, W.J.P.

Padilla Cazar, G.P.

Singla Lezcano, G.

Wechgelaer, C.A.

This document describes the initial design idea for the software which controls the operation of the PICO robot during the "*A-Maze-ing challenge.*" This challenge is part of the course [4K450] Embedded Motion Control at the Eindhoven University of Technology.

GOAL:

The goal of the "*A-Maze-ing challenge*" is:

Design and implement software for the PICO robot, such that the robot is able to find the exit of the maze as quickly as possible and in a fully autonomous way.

1 REQUIREMENTS

1. The PICO robot has to be used to solve the "*A-Maze-ing challenge.*"
2. The PICO robot should operate fully autonomously; after starting, the robot it should make all decisions about navigating autonomously.
3. The operation of the PICO robot should be independent of the maze configuration.
4. During operation, the PICO robot should not collide with anything, e.g. walls, doors, robots and humans.
5. The PICO robot should, upon completion of the maze, terminate without human intervention.

2 FUNCTIONS

1. The PICO robot performs basic movements in navigating through the maze, as it is equipped with a holonomic base (omni-wheels), the following movements are possible
   - move forward or backward
   - move sideways,
   - change orientation of the robot

   of course any combination of the above-mentioned movements is also possible.

   The odometry data from the holonomic base of the robot will be used to control and monitor the robot basic movements.
2. During navigation the PICO robot should keep a minimum distance to the walls which define the maze. The distance to the walls will be measured using the laser-range-finder.
3. During navigation the PICO robot should take autonomous decisions about which way to head on an intersection.
4. Upon encountering a dead-end the PICO robot should execute an algorithm which tests whether the dead-end is a door or not; *Test for doors*.
5. During navigation the PICO robot should build a semantic maze model from the perceived environment using the laser-range-finder.
6. Upon completion of the maze the PICO robot should autonomously terminate, meaning that it will not move any further from that point onwards.
7. During operation the PICO robot should respond, in a safe way, to the kill-switch command which can be triggered by the operator from a software-interface.

3 COMPONENTS

A schematic overview of the software for the PICO robot is given in Figure 2, the specifications of the components in this overview are given in *Section 4: Specifications*.

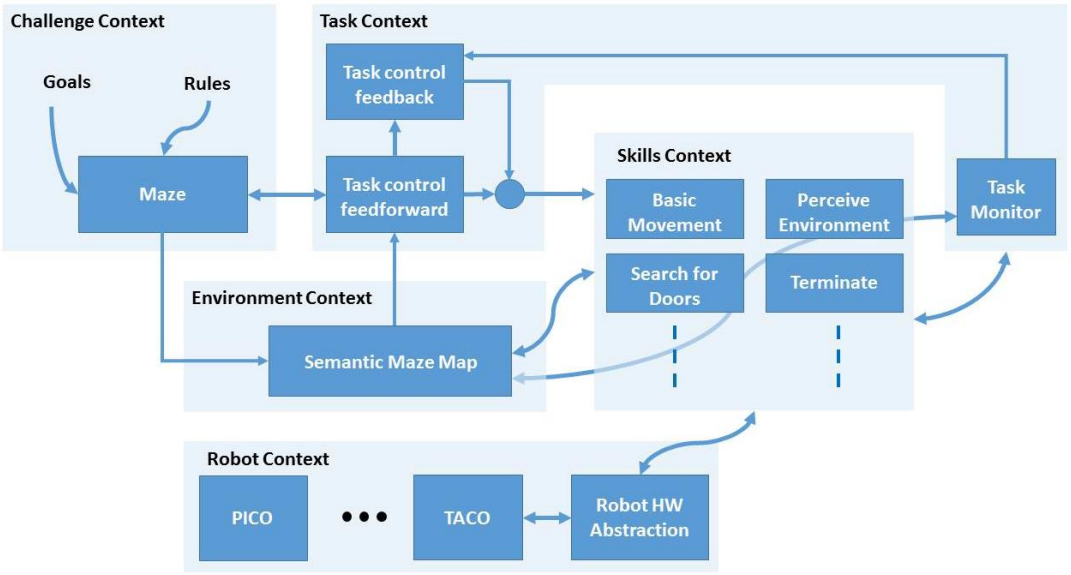

Figure 1: Schematic view of components making up the software of the PICO Robot

4 SPECIFICATIONS

- **Task Context:** Controls the execution of the robot's skills depending on the challenge and the environment context. The task context is divided into three major parts
  - **Task Control Feedback:** Depending on the information received from the task monitor contributes in controlling the robot skills. This includes
    - Keeping a minimum distance to the walls.
    - Taking an autonomous decision if the robot encounters a special situation: intersection, T-junction or a corner.
  - **Task Control Feedforward:** Depending on the state and the goal of the challenge contribute in controlling the robot skills. This includes,
    - Control of the robot when no special situation is encountered; drive forward.
    - Use the information gathered using the laser-range finder to construct a semantic maze model.
  - **Task Monitor:** Monitors the execution of the skills and sends this information to the task control feedback which in turn decides upon a new control action.
- **Environment Context:** Semantic maze model.
- **Challenge Context:** Contains all the information about the maze-challenge at hand such as the rules of the game and goal of the game. The following characteristics of the maze are given
  - The location of both entrance and exit is unknown, but will be on the boundaries of the maze. The maze does not contain loops.
  - The walls will always be approximately parallel to each other.
  - The walls will be placed at an approximately constant distance apart; allowing the PICO robot to perform a complete rotation without hitting the wall.
  - The following situations exist: corners, T-junctions, crossings and straights.
- **Robot Context:** Low-level specifications of the PICO Robot. To navigate through the maze the robot is equipped with a holonomic base (omni-wheels). In navigating through the maze the following sensors will be used
  - Laser-range-finder (LRF), the data from this sensor is represented in polar coordinates (corresponding distance at a certain angle).
  - Wheel encoders (odometry), the data from this sensor provides information about angular displacement of the wheels; allowing to calculate the distance travelled.

  The robots hardware abstractions layer takes care of all low-level sensing- and actuation-capabilities of the robot. The robots embedded processing is handled by an Intel I7 processor running Ubuntu 14.04.
- **Skill Context:** Contains the functions which are already mentioned, these can be categorized according:
  - Navigate through the maze
  - Map the maze

- Terminate
- Perceive the environment
- Test for doors

5 INTERFACES

- **Skill context – Robot context:** This interface takes care of sending commands to low-level hardware and returns sensor signals.
- **Challenge context – Environment context:** This interface takes care of assumptions about the maze and goal approach.
- **Task context – Environment context:** This interface takes care of providing the correct information to the task context in order to make decisions based on the maze map.
- **Task context – Skill context:** This interface takes care of the skill selection based on both the contribution of the task control feedback and task control feedforward.
- **Challenge context – Task context:** This interface takes care of providing the goal of the challenge context to task context, in order to make decisions to aim for the goal of the challenge while adhering to the rules of the game.
- The **Challenge context** also provides the interface to control the robot from the outside of the field, this involves commands to start and stop the robot and provide information about the current status of the robot in the maze.