Initial Design of a Maze Solving System for the PICO Robot
4SC020 - Embedded motion control

May 10, 2017

Group 5:
T.F. Beernaert (0778690)
R. Estrella Treviño (1035228)
K. Incetan (1037760)
S.C.M. Knippenberg (0738104)
A. Molina Acosta (1036456)
B.C.G. Vercoulen (0747283)

Introduction

This document presents a description of an initial design concept for a control system whose purpose is to guide the PICO mobile robot out of a maze or corridor. The following sections describe the requirements and specifications applicable to this assignment, as well as a preliminary concept for functions, components, and interfaces to be included in the controller.

Requirements and specifications

After analyzing the description of the corridor and maze assignments, an initial set of requirements was derived. These were grouped into five broad categories: Functionality, Usability, Reliability, Performance, and Supportability (usually known by the acronym FURPS, see [1]). The requirements are listed here below. Where applicable, an associated specification is shown in between parenthesis.

Functionality

- The robot shall be able to exit the maze/corridor.
- The robot shall be able to identify walls (the corridors will be between 0.5 and 1.5m wide).
- The robot shall be able to identify doors and dead-ends.
- The robot shall be able to request a door opening (these should be done with the provided command).
- The robot should identify if the door has opened (it should monitor for 2 seconds to detect if the alleged door started opening).
- The robot shall be able to identify corners (corners being two walls at an angle of approximately 90 degrees).
- The system shall be able to plan and follow a path (path following error should not be larger than 10cm).
- The system shall be able to map the path it followed.

Usability

- The robot should provide information on what its actions will do next.
- The system should be able to keep a log of the controller outputs and sensor inputs.
- The system should display the known world map on real-time.

Reliability

- The robot shall not hit any walls.
- The system should take into account the stopping distance that can be achieved with the motors (based on the robot's inertia and motors' capabilities).
- The robot should not stand still for long periods time (at most 30 seconds standing still).
- The motion controller should always be stable.
- The system should never get into a deadlock or get stuck in a infinite loop.
- The robot should wait long enough for the door to open after ringing the bell (it should monitor for 7 seconds at most).

Performance

- The motion controller should not output speeds above the limits (maximum translational speed of 0.5m/s, maximum angular speed of 1.2rad/s).
- The time to exit the maze/corridor should be as short as possible (respecting the speed constraints).
- The robot should exit the maze/corridor in time (at most 7/5 minutes).
- The system should be able to solve the maze within two attempts.

Supportability

- The system should be compatible with the specified starting procedure.

Functions

From the introduced requirements, an early set of functions is extracted. These functions are arranged in a task-skill-motion hierarchy, which discriminates them on levels of abstraction and mutual relations. Figure 1 shows a SysML diagram of the proposed functionality of the PICO controller, including the relationships between these functions. The 'include' arrow indicates that a certain action (at the arrowhead) is a necessary part of another overlapping action (at the tail). The 'extend' relationship depicts that the action at the tail is an optional follow up of the action at the arrowhead.

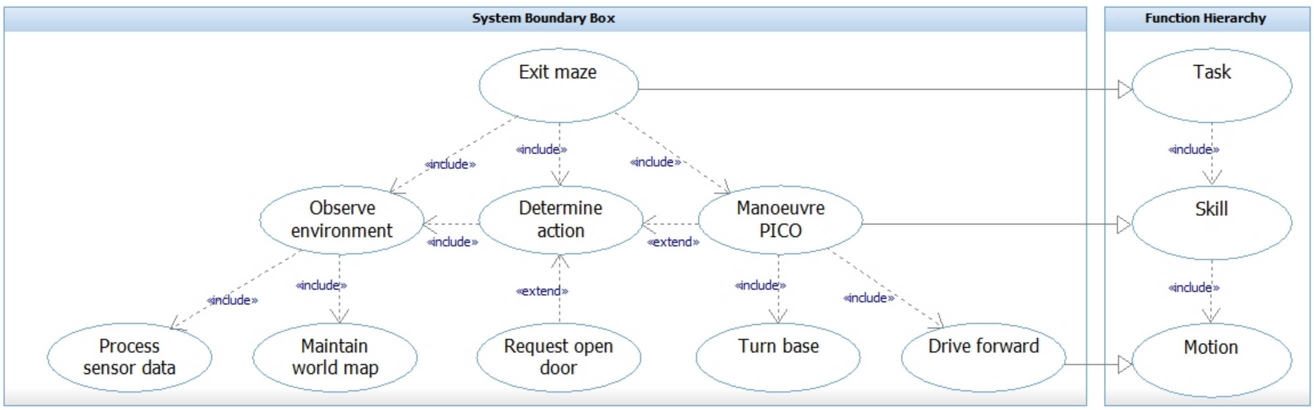

Figure 1: SysML Diagram depicting proposed functionality of the PICO Controller.

The main task of the system is to exit the maze. To do so, three skills are necessary: PICO autonomously decides which action to take (1), based on its observations of the environment (2) and is capable to execute the considered action (3). These skills can be further expanded to find motions, which are shown at the bottom of the diagram.

A function is required to combine the gathered data points from the laser range finder to identify walls. Connecting the identified walls, a world map is generated in which the robot can position itself. This map needs an update every iteration, therefore a mapping function is required to be able to map known walls and extend the map with new walls.

Components

The analyzed functionality leads to a proposed architecture of the system. Data enters the system from the encoders and the Laser Range Finder (LRF). This information is gathered in a world map, containing information about the surroundings of PICO. The current situation is sent to the action manager, which determines the action that will be performed. Finally, the movement controller actually computes the input for the actuator, based on the action from the action manager and the current sensor data. Figure 2 depicts this structure.

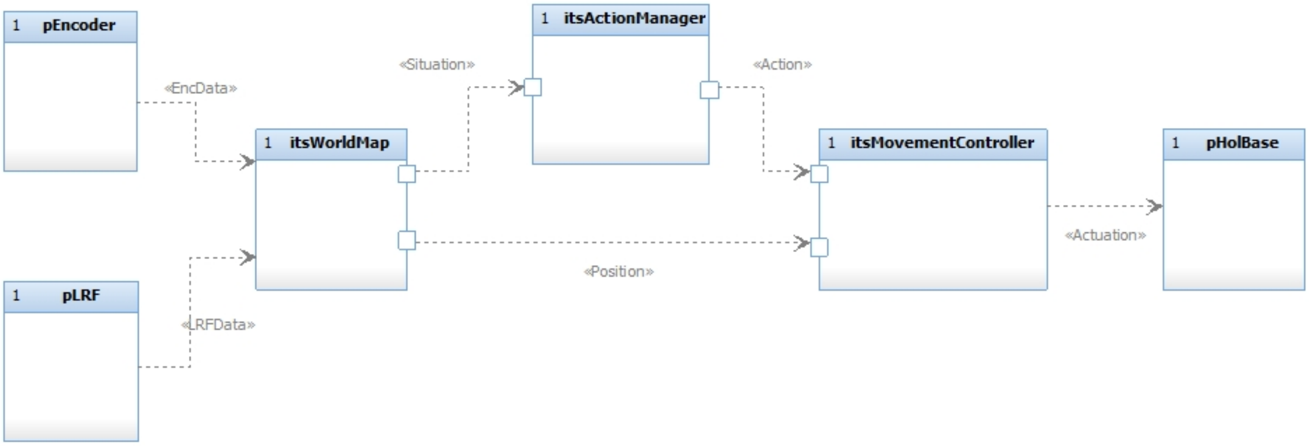

Figure 2: Internal system architecture with data flows.

Interfaces

The point where the robot starts in the maze will be defined as the origin, from which the map is generated. The world map has to be interpreted by the action manager for deciding the next action of the robot. For the feedback to the developers we want to have a map of the world model as the robot knows displayed on the screen, with a list of actions the robot decided to do at each point in time.

Next steps

The upcoming steps that will be performed from now on are:

- Think about how to make abstractions of the real world.
- Think about tactic of decision making and strategies for maze solving.
- Research on path following controllers for holonomic mobile robots.
- Research on techniques for simultaneous localization and mapping.

References

[1] R. B. Grady. (1992) *Practical Software Metrics for Project Management and Process Improvement.* Prentice Hall, Englewood Cliffs