Department of Mechanical Engineering

4SC020 - Embedded Motion Control

Information architecture design document

Group 7

Guus Bauwens	0958439
Ruben Beumer	0967254
Ainse Kokkelmans	0957735
Johan Kon	0959920
Koen de Vos	0955647

Tutor: Y.G.M. Douven

Eindhoven, May 6, 2019

1 Requirements and specifications

To successfully navigate a hospital through multiple rooms with different subsequent waypoints, PICO

- should have a sense of where it is located with respect to its target (e.g. by means of an orientation and a 2D distance),
- should autonomously be able to decide what actions to carry out to reach its targets (e.g. how to move, when to stop, playing an audio fragment) in a partially observable environment,
- and should have some sense of its progress towards its current target, e.g. the distance to go.

While doing so, PICO

- must stay clear of static objects (walls) with a safety margin of 50 mm,
- and safely interact with its environment (i.e. do not hit humans),
- preferably completing its targets as fast and efficiently as possible,
- while communicating its intents with its surroundings, e.g. a certain stage in a roadmap by speaking.

2 Information Architecture

Figure 2.1 depicts the proposed information architecture, from incoming information resources (LRF, odometry, control effort, map) to outgoing actuation capabilities. It illustrates the multiple components, grouped into five relevant containers and the interfaces between these components. The functionality and specifications of each component are explained in Figure 2.1 and the text below serves as a motivation for the different components.

2.1 Planning

The planning component is responsible for defining the current target of the robot. This component consists of a finite state machine, a path planning algorithm and a trajectory planner. The world model provides the planner with the current position of the robot and the world map. The component returns a reference position and velocity for the robot to follow.

The finite state machine switches between the different discrete behavioural states of the robot. When the software is started, it will first wait for initialisation of the sensors. Then it will explore to ensure convergence of localisation by the Monte Carlo particle filter (MCPF). Afterwards, it can use this localisation to visit the different cabinets. In each state, the finite state machine will provide the path planning algorithm with a target position based on the available monitors. In addition, the finite state machine will generate audio to make its intentions clear to bystanders.

The path planning algorithm will be based on the A* algorithm: a grid based planning technique for navigating occupancy maps. The world model provides the algorithm with the current position and a grid map of the world. Note that although grids scale badly, only finite precision is required for generating a path so also a coarse grid suffices. The path planning algorithm will then generate a path which the robot needs to follow. The trajectory planner takes this path information and converts it into a reference position and velocity. This reference velocity needs to be within the constraints as posed in the problem description.

2.2 Control

The control component is responsible for converting the reference positions and velocities of the planning algorithm to an actual velocity control signal. This is done in two ways. Firstly, the velocity setpoint can be applied as an open loop feed forward signal. Secondly, the position setpoint is applied in a feedback situation, in which actuation is performed on the basis of the error between desired (planned) and actual (MCPF pose) movement along the trajectory and steering angle. To fully utilise the LRF in the driving direction, it is beneficial to drive forward. However, this does not exploit PICO's holonomicity. To combine the two, the provided path in (x,y) is mapped into local references using the current orientation while the orientation is simultaneously controlled into the driving direction. This ensures that its holonomicity is exploited for small distances in narrow environments, and also that PICO will converge to driving forward.

A deadzone can be added to the feedback controller to incorporate the concept of guarded motion, such that the controller will only react outside a tube around the reference.

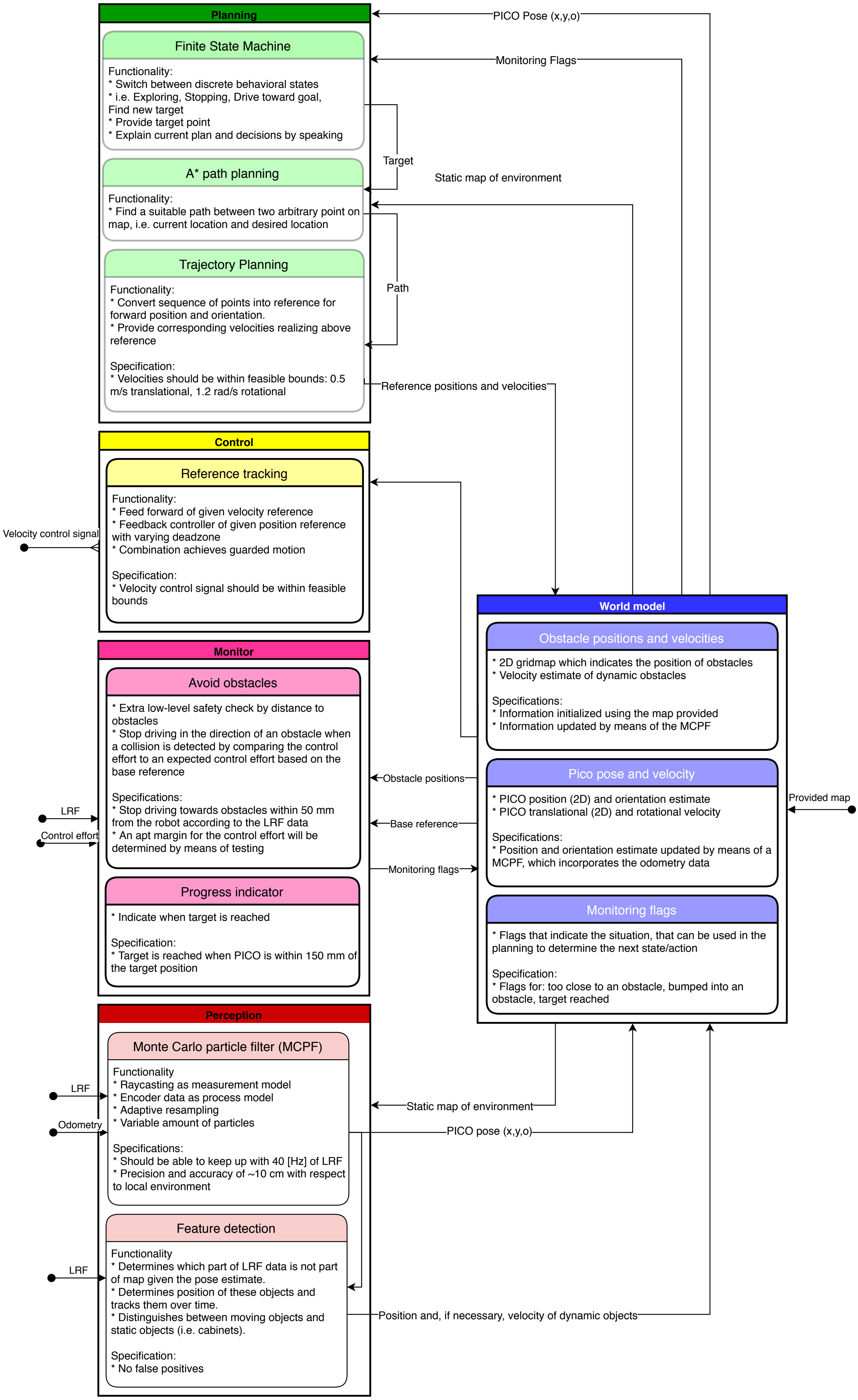

Figure 2.1: Planned software architecture for the hospital challenge

2.3 Monitor

Global positioning and path planning are covered by the perception and correspondingly the planning. However, PICO's low level software should have specific safe behaviour in case of for example a (incoming) collision, too high control effort and cabinet detection.

As dynamic objects (i.e. humans) are present in a hospital environment, these can bump into PICO and likewise PICO could bump into them. As PICO does not have a 360 wide view range, obstacles (static and dynamic) cannot always be noticed. This can mostly be prevented by manoeuvring PICO only in a direction visible by the LRF domain.

To avoid damage and injuries, PICO must stop when it has to apply too high control effort for the task it is performing (or when an obstruction is measured by the LRF). As this control effort is not known for specific movement speeds, normal operating values should be gathered and stored for reference. For redundancy, the LRF data can be checked for overall changes when control effort is applied to an actuator of PICO. In case of too high control effort, PICO should wait and check its surroundings (which also gives passing by dynamic objects a chance to leave) and move in a safe direction. If the control effort is still too high, PICO should try to turn and move into a (significant) other direction if the LRF data confirms this direction is safe to travel to.

From this point on wards the path planning should compensate for the obstruction found and reroute PICO to its target. In all cases the monitoring software should overrule the path planning to avoid the mentioned damage to the environment and PICO itself. If for any reason PICO becomes closed in, it should stop and make it known to its environment that it cannot manoeuvre by for example its speak engine.

2.4 Perception

A Monte Carlo particle filter (MCPF) is the main component of the perception: given Laser Range Finder (LRF) and odometry measurements and a map, it will keep track of a non-parametric probability distribution representing the robots belief about its position. A particle filter is well suited for this, as it is able to handle the complex, non-linear LRF sensor model and is able to recover the robot location on the map from every initial position.

Furthermore, a particle filter allows for a memory in location: a lack of information will not result in a loss of information about its position on the map. Local mismatch of distance measurements and the provided map is tolerable due the large amount of LRF measurements, all providing information about the robots position through other areas of the map. It also fully exploits all information provided by the LRF: even if part of the rays is irrelevant for the current task, they still provide information about the robots pose. Although the position is expressed with respect to the map (i.e. global), the localisation provides local accuracy due to the nature of the LRF.

The output of the filter is a robot pose as median of the particles (in case of an approximately unimodal non parametric distribution) or multiple robot poses (multimodal), which can then be used by the state machine to determine a plan (e.g. explore to determine position or plan path from current, well-known position).

Given a converged robot pose and a map, the different rays of the LRF can be associated either with components on the static map (walls) or components that are not present on the static map (dynamic, e.g. humans, or static, e.g. cabinets). This allows for tracking these objects and distinguishing between static and dynamic ones, such that cabinets can be recognised.

2.5 World Model

In the world model, all relevant information is grouped. It serves as a knowledge base for all other activities, but does not manipulate this information itself.