Introduction

This course aims to present practical skills in the area of robot navigation in a complex task, i.e. a task consisting of many fundamental robotic functionalities. This will be done trough a gradual solving of the exploration task, for which sub-problems like design of a basic robot software architecture, sensor data processing, robot navigation, environment model building, planning, and high-level decision making will be presented and discussed.

By **exploration** we understand a process of autonomous navigation of a mobile robot in an unknown environment in order to build a model of the environment. An exploration algorithm can be defined as an iterative procedure consisting of a selection of a new goal and a navigation to this goal. Such an algorithm is terminated whenever the defined condition (mission objective) is fulfilled. The mission objective in our case is to build a complete map of the environment. The map is built incrementally as sensor measurements are gathered and it serves as a model of the environment for further exploration steps. Besides, the usage of resources (e.g. the exploration time, the length of the trajectory) is optimized.

Task definition

Our main task is to design an algorithm for exploration of an unknown space. But what does this means more precisely? Assume a mobile robot working in an artificial environment (labyrinth) starting in a predefined position somewhere near a boarder of the environment (imagine, we put the robot just near the entrance to the labyrinth). The robot is equipped with a ranging sensor with a fixed, limited range (e.g., a laser range-finder) and field of view, odometry (information about movements performed by the robot). Its goal is to navigate in the environment and build a complete map of it as fast as possible.

The approach

In this section we get acquainted with a frontier based exploration approach introduced by Brian Yamauchi [1]. The approach employs an occupancy grid [2] as an environment model, which is built continuously as new sensor readings are gathered. An occupancy grid represents the working space as a matrix of small rectangular boxes (cells), where each cell stores information about the corresponding piece of the environment in the form of a probabilistic estimate of its state. Occupancy grid cells are consequently segmented into three categories (by application of two thresholds on their probability values): free, occupied, and unknown (unexplored). The frontier based approach detects *frontier cells*, i.e. the reachable free grid cells that are adjacent with at least one cell that has not been explored yet. The *frontier* is a continuous set of frontier cells such

that each frontier cell is a member of exactly one frontier. See Fig. 1 for a graphical visualization of the used terms. Once all frontiers are detected, the most appropriate frontier cell is selected as a new robot goal according to the defined criteria, a path is determined to this cell and the robot is navigated to the goal. This process is executed repeatedly at defined time steps until there is no frontier cell reachable by the robot, which means that all reachable space is already explored. The exploration algorithm is depicted in Alg. [1].

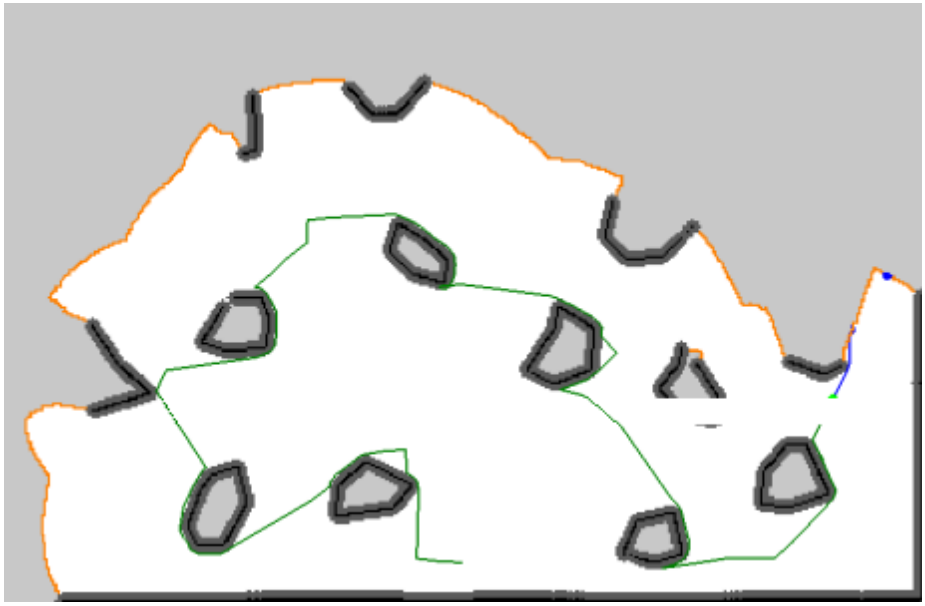

Fig.1.: *Terms used in exploration: the white color represents grid cells known to be free, occupied cells are in black and unknown in light gray. Frontiers are in in brown, while the blue dots stand for goal candidates, and the green circle stands for the current robot position. Finally, the already traversed path is in green, while the plan to the chosen candidate is in blue.*

Alg. 1. Frontier-based exploration

1. read current sensory information
2. update map with the obtained data
3. determine frontier cells
4. determine the next goal
5. plan paths to determined goal
6. move the robot towards the goal

Sensor readings are gathered frequently (units or tens of Hz) so the map is updated with this frequency. On the other hand, a new goal determination and planning is not needed so often (and also not possible due to time complexity of these tasks). It is therefore benefitial to split the processing into two threads (loops): sensor processing, map building, and robot control is processed in one

thread (*control loop*), while frontiers determination, next goal selection and path planning is done in the second one (*exploration loop*). These two threads then communicate via to structures they share:

- **Map**, which is built in the *control loop* and used in the *exploration loop*
- **Plan**, which is filled in the *exploration loop* and then traversed in the *control loop*.

Note also that only the *control loop* communicates with the robot: it reads data from robot sensors (including odomentry, i.e. position information) and sends control commands to robot's engines.

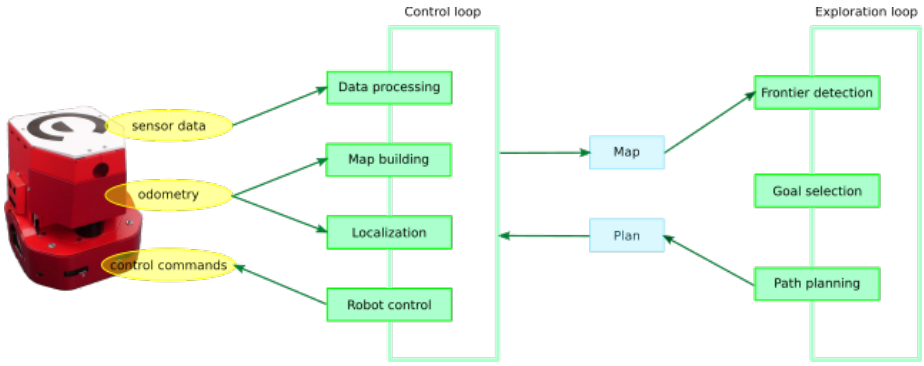

Fig.2.: *Application schema.*

References

[1] B. Yamauchi. A frontier-based approach for autonomous exploration, in: Proc. of IEEE Int. Symposium on Computational Intelligence in Robotics and Automation, IEEE Comput. Soc. Press, 1997, pp. 146–151. [2] A. Elfes A. A Probalistic Framework for Robot Perception and Navigation. Carnegie–Mellon University,1989.