Project: Mobile robot planning, control and estimation

Robotics (B-KUL-H02A4A)

2025

1 Introduction

In this project you will focus on solving a mobile robot navigation problem. The overall application is depicted in Figure 1. The robot has to navigate from a start pose to a goal pose, while avoiding obstacles. For no-collision purposes the robot is modeled as a circle and each of the obstacles as a capsule. The robot is equipped with a GPS sensor, that measures its position (but not orientation). The robot has to estimate its position and orientation from the GPS sensor measurements and known control inputs, and then use this information to control its motion.

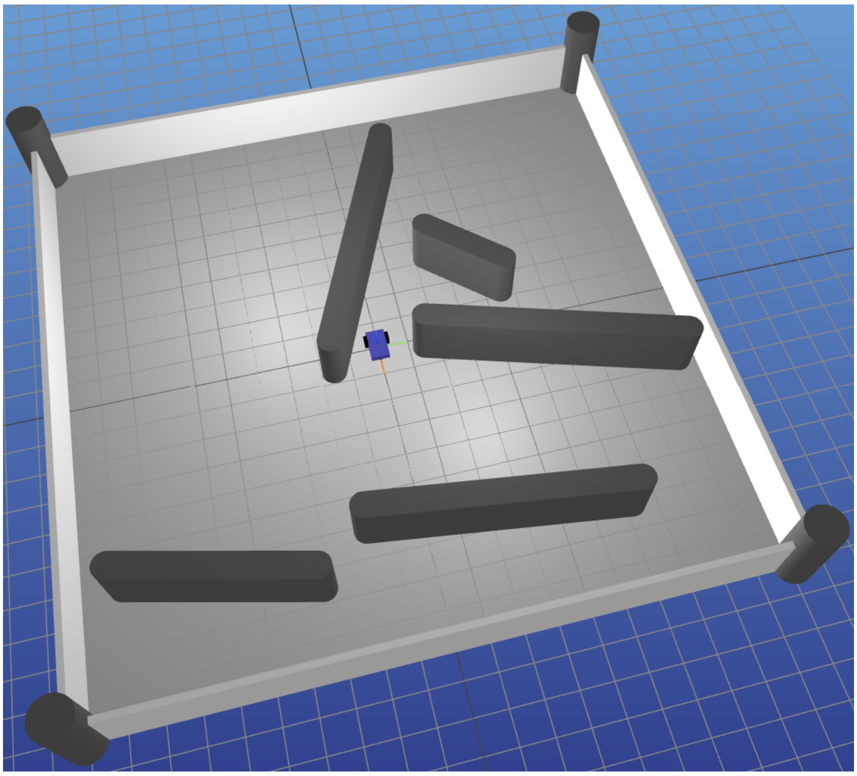

Figure 1: Mobile robot in a 2D environment.

2 What is provided?

The following is provided to help you complete the project:

1. A basic simulation environment simulating the differential-drive robot and laser distance sensors, as seen in Figure 1.

2. A URDF file defining the geometry of the differential-drive robot.

3. A Python notebook `mobile_robot.ipynb` that demonstrates how to interact with the simulation environment and provides a starting point for the project.

3 Tasks

The project consists of four tasks:

1. design a motion planner to generate a reference path or trajectory to move the robot from configuration A to a configuration B, while avoiding obstacles. Choose A and B yourself, and validate your implementation on multiple challenging cases. This motion planner should consist of three parts: 1) implement an optimization-based motion planner for the robot, ignoring obstacles. This planner is called the "local planner". 2) Write a collision check Python function that checks if a solution found by this local planner is effectively collision-free. 3) Implement a global motion planning using the probabalistic roadmap (PRM) algorithm, which uses this local planner as subroutine (hint: refer to section 10.5.2 in the *Modern Robotics* textbook).

2. design a controller that tracks the reference trajectory (hint: refer to section 13.3.4 of the *Modern Robotics* textbook), and

3. design an estimator to localize the robot based on the control inputs and the (noisy) GPS sensor measurements.

4. replace the differential drive robot with a car-like robot. For simplicity, assume a bicycle model of the car, with control inputs as the linear velocity and steering angle velocity. You will have to modify the URDF file of the robot (as well as the motion planner, controller, and estimator) accordingly.

4 Assumptions

1. It can be assumed that the positions of both the obstacles and walls are known beforehand.