Albert-Ludwigs-Universität Freiburg
Lecture: Introduction to Mobile Robotics
Summer term 2011

Institut für Informatik

Prof. Dr. Wolfram Burgard
PD Dr. Cyrill Stachniss
Juniorprof. Dr. Maren Bennewitz
Juniorprof. Dr. Kai Arras

Sheet 12

Topic: Motion Planning

Submission deadline: August 2, 2011
Submit to: `mobilerobotics@informatik.uni-freiburg.de`

General Notice

In the following exercises you will implement motion planners for a mobile robot. We assume a point-like robot which can freely move in the obstacle free region of a 2-dimensional environment. Consequently, its *continuous* configuration space is $\mathcal{C} = \mathbb{R}^2$ and its *continuous* action space is $\mathcal{U} = \mathbb{R}^2$ where each action $\mathbf{u} = [v_x \; v_y]^T$ defines the velocity in x and y direction.

A framework containing the environment definition and visualization tools is provided to you so you can concentrate on the implementation of the planning techniques itself. The framework can be obtained from the web site of this lecture.

To run the planners launch the *Octave* program. Inside *Octave*, type `test_planner` to run the example which should be extended step by step in the following exercises.

Exercise 1: Basic Techniques

(a) Complete the file `collisioncheck.m` by implementing collision checking for a single configuration $\mathbf{v} \in \mathcal{C}$ given the map represented by all obstacle cells.

(b) Complete the file `localplanner.m` by implementing a local planner checking whether a configuration $\mathbf{v}_2$ is reachable from another configuration $\mathbf{v}_1$ on a straight line.
Note: For simplicity you could check a discrete set of points on the straight line for collision.

(c) Complete the file `rejectionsample.m` which should uniformly draw samples from the free space. Therefore, draw samples from the complete configuration space uniformly and reject samples which are in obstacle cells.

Exercise 2: Probabilistic Road Maps

Implement the sampling of a probabilistic road map (PRM). Complete the file `sampleroadmap.m` by using the functions implemented in Exercise 1.

Note: Try to connect a new node only to the nodes which are in its vicinity. These nodes can be defined by a maximum Euclidean distance (e.g. 6*).*

Optional: Add the start and the goal configuration to the road map and apply your favorite graph search algorithm (e.g., depth-first search, breadth-first search, A*, ...) to find a path from the start to the goal.

Exercise 3: Rapidly-exploring Random Trees

Explore the configuration space using a rapidly-exploring random tree (RRT). You can use the Euclidean distance as a metric when searching for the nearest neighbor to the sampled configuration. Use the goal instead of a uniform sample of the free space in every 50th iteration. Complete the file `buildrrt.m`.

Exercise 4: Challenges

Define an environment which is very challenging for PRM and RRT planners. How could you modify the PRM and RRT planners to cope with such environments?