Advanced robotics (ARO)
Course Assignment 1
Total: 10 Marks

Steve Tonneau ©University of Edinburgh
30 September 2024 (homework handout)

This coursework counts for 10% of the final mark. Each point is equivalent to 1%.

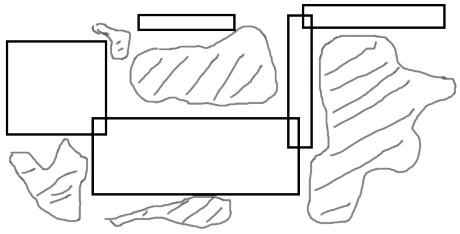
Before starting, please consider the following points:

- This should not take more than a few hours. If you feel like a question requires a dramatic amount of research / work, you probably misunderstood it.
- I am not expecting you to program anything. Paper and pen / latex report if you want to train for the lab report is all you need. Of course you are allowed to use code if that helps you.
- The "Discuss" questions do not require any calculations. I am more interested in assessing your understanding of the consequences of given design choices.
- You need to work alone on this homework. Collaboration is not allowed. You can ask for clarifications on EdStem, but any form of request for a solution will be sanctioned.

Motion planning - 7 points

The problem described in this exercise is similar to last year's exam, but the questions are completely different. The previous exam does not provide any help in solving this exercise, but you can take a look at it for an illustration of different approaches to a given problem.

For this exercise we assimilate our robot to a 2D point. Let's consider this figure:



The obstacles are represented by the grey shapes. The union of all the black rectangles provides an approximation of the free space $\mathcal{C}^{free}$. Any path such that our robot lies inside one of the rectangles at all time is thus guaranteed to be free of collision. A rectangle $\mathcal{S}_i$ is defined by

its 4 extreme points $\mathbf{s}_i^j, j \in \{1,2,3,4\}$. We will now assume that $\mathcal{C}^{free} = \bigcup_{i=1}^{m} \mathcal{S}_i$, with m the total number of rectangles. With this representation to check that the robot is not in collision is equivalent to verifying that the robot belongs to one of the rectangles.

We aim at finding a continuous path from a position $\mathbf{x}^{start} \in \mathcal{C}^{free}$ to any position $\mathbf{x}^{end} \in \mathcal{C}^{free}$.

Part 1 - evaluating whether a sample x belongs to $\mathcal{C}^{free}$

question 1 (1 pt): We can easily verify that a point lies within one given rectangle through a set of linear inequalities. For instance, think of a unit rectangle described by the vertices coordinates (0,0), (0,1), (1,0), (1,1). Any point that belongs to the rectangle is such that $-x \leq 0$, $x \leq 1$, $-y \leq 0$ and $y \leq 1$. Assuming that the $\mathbf{s}_i^j, j \in \{1,2,3,4\}$ are sorted in a clockwise order, give the general form of the inequality that determines if a point belongs to $\mathcal{S}_i$, ie:

$$\mathcal{S}_i : \{\mathbf{x} \in \mathbb{R}^2, \mathbf{A}_i\mathbf{x} \leq \mathbf{b}_i\}$$

with $\mathbf{A}_i$ and $\mathbf{b}_i$ respectively a matrix and a vector of appropriate size. To be explicit, you are expected to give the expression of $\mathbf{A}_i$ and $\mathbf{b}_i$ with respect to the $\mathbf{s}_i^j$.

Important note: Do **NOT** assume the rectangle edges are aligned with the coordinate axes.

Part 2 - computing a path in $\mathcal{C}^{free}$

question 2 (2 pt): Write a sampling-based algorithm that will solve this problem. Linear interpolation can be used as a steering method. Discuss the parametrisation of the algorithm, in particular the definition of the boundaries.

Part 3 - sample efficiency

Quantitative analysis of your algorithm indicates that a very large proportion of the samples do not belong to $\mathcal{C}^{free}$. We thus propose to rewrite the sampling method to ensure that all generated $\mathbf{x}$ are collision free in the hope of improving the performance. We will use two approaches to achieve this.

question 3 (1 pt): Use an optimisation-based method to project a sample $\mathbf{x} \in \mathcal{C}$ into $\mathbf{x}^* \in \mathcal{C}^{free}$. Discuss the properties that you are looking for in such a projector. Discuss also the completeness of the resulting algorithm and the distribution of the samples.

question 4 (1 pt): Since a rectangle is a convex shape, it verifies the following:

$$\mathbf{x} \in \mathcal{S}_i \Leftrightarrow \exists \boldsymbol{\alpha} = [\alpha_1, \alpha_2, \alpha_3, \alpha_4] \in \mathbb{R}^{4+}, \|\boldsymbol{\alpha}\| = 1, \mathbf{x} = \sum_{i=1}^{4} \alpha_i \mathbf{s}_i$$

Using this property, write a function that directly samples $\mathcal{C}^{free}$.

question 5 (1 pt): Discuss the odds of sampling points that lie exactly on the edge of a rectangle using each approach.

question 6 (1 pt): Discuss the completeness of the resulting algorithms in each approach, as well as the effect of the sampling method on the distribution of the samples.

Inverse geometry - 3 points

The generalisation of a polygon to abitrary dimension is called a polytope (Examples of convex 3D polytopes are a cube or a pyramid). If the polytope is convex, we can use the same concepts to determine whether a point in dimension n lies within a polytope $\mathcal{P}$:

$$\mathcal{P} : \{\mathbf{q} \in \mathbb{R}^n, \mathbf{P}\mathbf{q} \leq \mathbf{p}\}$$

with again, $\mathbf{P}$ and $\mathbf{p}$ a matrix and vector of appropriate size. As a result, the algorithm you proposed can straightforwardly be extended to a manipulator arm for instance. We now consider the issue of interpolating between two sample configurations. For simplicity we will assume that $\mathcal{C}^{free}$ is approximated with a single polytope $\mathcal{P}$ and that $\mathbf{P}$ and $\mathbf{p}$ have been somehow calculated and are known.

question 7 (1 pt): Given two configurations $\mathbf{q}_1$ and $\mathbf{q}_2$, both inside the same polytope $\mathcal{P}$, prove that the convex combination

$$\mathbf{q}_\gamma = \gamma\mathbf{q}_1 + (1-\gamma)\mathbf{q}_2$$

is such that $\mathbf{q}_\gamma \in \mathcal{P}$, $\forall\gamma \in [0, 1]$.

However, when interpolating between $\mathbf{q}_1$ and $\mathbf{q}_2$, we would like the end-effector to follow a straight line. Make sure you understand why it is not the case here (this is not assessed).

question 8 (2 pt): Write the pseudo-code that generates such a path while ensuring that all configurations in the resulting path belong to $\mathcal{P}$. You can assume that a method $forwardgeometry(\mathbf{q})$ gives you the end effector position of a given configuration.