CS 294-115 Algorithmic Human-Robot Interaction **Fall 2016**

Lecture 2: Motion Planning - Part I

Scribes: Andrea Bajcsy, Kush Bhatia

2.1 Administrative

Readings which are not marked as mandatory are, in general, more useful after the lectures. Mandatory readings are supposed to be read before class.

2.2 Overview

The lectures on *Motion Planning* will broadly cover the following topics:

1. Configuration Spaces
2. Problem Statement
3. Algorithms

Recently, Motion Planning problems have been looked at through the lens of optimization. This has not been the case classically and through these lectures we will attempt to look at the problem through a more classical robotics lens.

2.3 Configuration Space

A configuration can be informally understood as everything that we need to describe where the robot is (assuming known kinematics/geometry). Mathematically, we represent a configuration space by the set C and a configuration is any element $q \in C$.

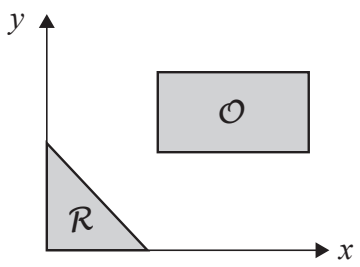
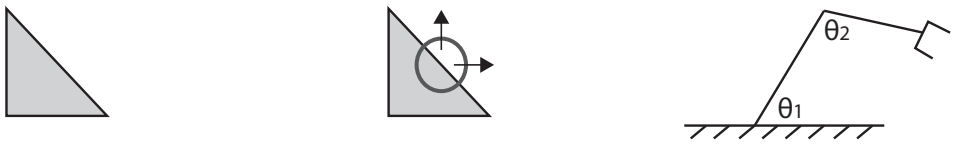


Figure 2.1: World W denoting Robot R and Obstacle O

We denote the world space by $W(\mathbb{R}^2$ or $\mathbb{R}^3)$. We represent robot R as a function:

$$R(q) : C \to \mathbb{P}(W)$$

where $\mathbb{P}(A)$ represents the powerset of the input set A. In effect, the function R maps a given configuration $q \in C$ to a set of points $R(q)$ representing the part of the world occupied by the robot.



(a) Robot R - translate (b) Robot R - translate + rotate (c) A robotic arm R with 2 dof

Figure 2.2: Examples robots for illustrating Configuration Spaces

Example 1: For the robot R in figure 2.2a, the configuration space is given by $q = (x, y)$, i.e., the x and y coordinates are sufficient to describe the robot since it is restricted to translation motion.

Example 2: For the robot R in figure 2.2b, the configuration space will be represented by $q = (x, y, \theta)$. In addition to the (x, y) coordinates, we require an additional θ coordinate to specify the rotation.

Example 3: For the robotic arm R in figure 2.2c, we require two parameters θ_1, θ_2 to completely specify the position of the arm in the world. Therefore, $q = (\theta_1, \theta_2)$.

Fun Fact: Human arms have 7 degrees of freedom, which is one more than the 6 required to specify a general configuration in the $3D$ space. This allows for infinitely many configurations for solving a particular problem and thus making it easier to perform our tasks.

2.4 Forward & Inverse Kinematics

Forward Kinematics (ϕ) for a given point on the robot will take a configuration q and point it to a point in the world.

$$\phi : C \to W$$

For example, in figure 2.2c, $\phi_{gripper}$ will take as input a configuration (θ_1, θ_2) and return a point (x, y) in the world which would indicate the position of the gripper under the current configuration.

Inverse Kinematics (IK) is a function which given a point in the world x, will map it to the set $\{q|\phi(x) = x\}$. While it is possible to compute the inverse kinematics for some robots analytically, for many of the robotic designs it is infeasible and we must use optimization based approaches.

2.5 Obstacles

The real world contains obstacles. These *obstacles* are part of the set $O \subset W$ and affect the configuration space:

$$C_{obs} = \{q \in C \mid R(q) \cap O \neq \emptyset\}$$

$$C_{free} = C \setminus C_{obs}$$

C_{obs} is the set of configurations where the robot and the obstacle are in an intersecting position in the world, while C_{free} is the set of configuration on which the robot can move freely.

Technical Note: a *closed set* is a set that contains it's limit points (e.g. [0,1]). C_{obs} is a closed set with limit points being the points that are touching the robot. C_{free} is an *open set*.

Figures 2.3, 2.4, and 2.5, show several examples of different robots, obstacles, and the resulting configuration space. Aside: In grasping problems, $R(q)$ will change when the robot eventually grasps the target object. This makes manipulation interesting because touching is allowed (unlike in traditional motion planning) and we now have closure of C_{free}.

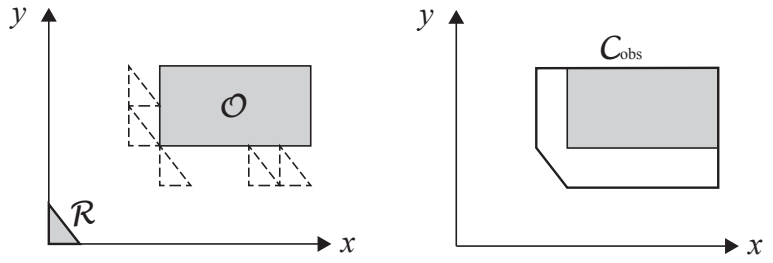
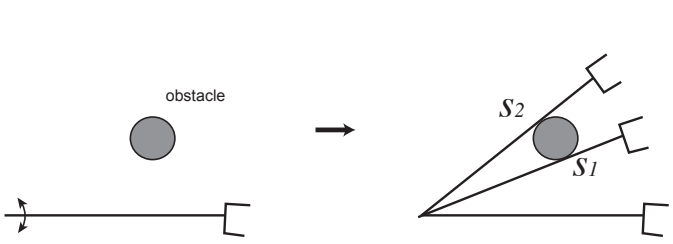
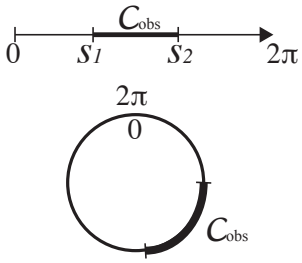


Figure 2.3: C_{obs} for a robot R that translates in x-y with a rectangle obstacle O



(a) A 1DOF robotic arm and 1 obstacle.



(b) C_{obs} for a 1DOF arm and 1 obstacle.

Figure 2.4: Example of C_{obs} for a robotic arm with 1DOF.

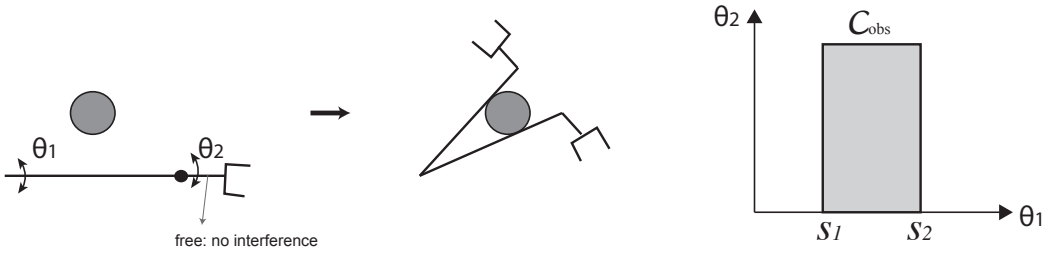
2.6 Minkowski Difference

Now that we know obstacles are present in our environment, how do we compute C_{obs}? When dealing with 1D or 2D translations, we can find C_{obs} using the *Minkowski Difference*:

$$C_{obs} = O \ominus R(\emptyset) = \{o - r \mid o \in O,\ r \in R(\emptyset)\}$$

The Minkowski Difference takes all the points in the obstacle, all the points the robot occupies in the $\emptyset$ configuration, and subtracts them. See Figure 2.6 for a 1D example of the Minkowski Difference.

THEOREM 1 *If O and $R(\emptyset)$ are both convex, then C_{obs} is also convex given that C_{obs}, O and $R(\emptyset)$ all belong to the same space.*



(a) A 2DOF robotic arm and 1 obstacle.

(b) C_{obs} for a 2DOF arm and 1 obstacle.

Figure 2.5: Example of C_{obs} for a robotic arm with 2DOF. Note: Link-to-link collisions are part of C_{obs} because we don't want to collide with ourselves.

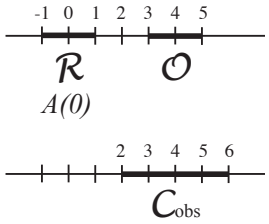


Figure 2.6: Example of Minkowski difference

PROOF 1 *We need to show that if we take any two points in C_{obs} and linearly interpolate between them, then the result is still in C_{obs}.*

Let $t_1, t_2 \in C_{obs}$. This implies that $\lambda t_1 + (1-\lambda)t_2 \in C_{obs}$.

Define $R' = -R$ as the reflection of R where R' is convex and $C_{obs} = O \oplus R'$. Since $t_1, t_2 \in C_{obs}$, we can write:

$$t_1 = O_1 + R_1$$

$$t_2 = O_2 + R_2$$

We know that O, R' are both convex, and can be written in terms of O_1, O_2 and R_1, R_2 respectively:

$$\lambda O_1 + (1-\lambda)O_2 \in O$$

$$\lambda R_1 + (1-\lambda)R_2 \in R'$$

Adding these together, we get:

$$\lambda(O_1 + R_1) + (1-\lambda)(O_2 + R_2) \in O \oplus R'$$

$$\lambda t_1 + (1-\lambda)t_2 \in C_{obs}$$

Thus, C_{obs} is convex, which was to be shown.