ASEN 5519 - Algorithmic Motion Planning
Fall 2021

Homework 4

Assigned September 17; Due September 24

Exercise 1. Consider workspace $W \subseteq \mathbb{R}^2$ with one obstacle in the shape of a triangle with vertices $(0,0)$, $(1,2)$, and $(0,2)$.

(a) Determine the coordinates of all the vertices of the C-space obstacle for a robot with the same shape as the obstacle that is limited to translational motions in W. Choose the local reference point of the robot to be at its lower-left vertex.

(b) Plot the C-space obstacle for the same robot with the additional capability of rotational motion in W. Use a view angle that clearly displays the full dimensionality and shape of the obstacle. (Hint: use a discretization of the additional parameter and the implementation of the algorithm used in part (a). Use only 12 equally-spaced values of the additional parameter in your plot.)

Exercise 2. Implement a kinematic model for a planar 3-link manipulator with the following capabilities:

- The user can specify the length of each link as well as the configuration, in which case the output is the visual display of the robot configuration as well as the exact location of the end point on the final link.
- The user can specify the length of each link as well as the desired location of the end point, in which case the output is the visual display of the robot and the numeric values of the angles. If you make any assumptions, they need to be clearly explained.

Illustrate the performance of your implementation by plotting the configuration(s) of the manipulator for the following cases:

(a) Links: $a_1 = 0.5$, $a_2 = 1$, $a_3 = 0.5$, and joint angles: $\theta_1 = \frac{\pi}{6}$, $\theta_2 = \frac{\pi}{3}$, and $\theta_3 = \frac{7\pi}{4}$.

(b) Links: $a_1 = 1$, $a_2 = 0.5$, $a_3 = 1$, and end-effector at $(2,0)$.

Exercise 3. Implement a C-space constructor for a 2-link manipulator in a 2-dimensional workspace with polygonal obstacles (hint: use a fine grid). Assume the base of the robot is at the origin of the workspace. The program should have the following capabilities:

- The user can specify the length of each link, number of obstacles, and vertices of each obstacle.
- The program displays the C-space with its obstacles.

Illustrate the performance of your implementation for a robot with link lengths of 1 in the following three workspaces:

(a) a workspace with a triangular obstacle with vertices $(0.25, 0.25)$, $(0, 0.75)$, and $(-0.25, 0.25)$.

(b) a workspace with two large rectangular obstacles with vertices:

$$\begin{aligned} O_1: &\quad (-0.25, 1.1),\ (-0.25, 2),\ (0.25, 2),\ \text{and}\ \ (0.25, 1.1), \\ O_2: &\quad (-2, -2),\ (-2, -1.8),\ (2, -1.8),\ \text{and}\ \ (2, -2). \end{aligned}$$

(c) a workspace with two obstacles: O_1 from part (b) and

$$O_2': \quad (-2, -0.5),\ (-2, -0.3),\ (2, -0.3),\ \text{and}\ \ (2, -0.5).$$

Plot both the workspace and C-space for each case. The C-space plots must include only positive values displaying the whole space.