COM S 476/576 Project 2: C-Space and C-Space Obstacles

Due Mar 9 at 11:59pm

This project is an extension of Project 1 and problem 3.14 in Homework 1, with the special case of $m = 2$.

Consider a robot consisting of 2 links, $\mathcal{A}_1$ and $\mathcal{A}_2$. Each link has width W and length L. The distance between the two points of attachment is D. $\mathcal{A}_2$ is attached to $\mathcal{A}_1$ while $\mathcal{A}_1$ is attached to the origin. Each link is allowed to rotate about its point of attachment. The configuration of the robot is expressed with 2 angles (θ_1, θ_2). The first angle, θ_1, represents the angle between the segment drawn between the two points of attachment of $\mathcal{A}_1$ and the x-axis. The second angle, θ_2, represents the angle between $\mathcal{A}_2$ and $\mathcal{A}_1$ ($\theta_2 = 0$ when they are parallel).

The world is $\mathcal{W} = \mathbb{R}^2$. The obstacle region $\mathcal{O} \subset \mathcal{W}$, the link's parameters, and the initial and goal configurations are described in a json file, which contains the following fields.

- `"O"`: a list $[\mathcal{O}_1, \ldots, \mathcal{O}_n]$, where $\mathcal{O}_i$ is a list $[(x_{i,0}, y_{i,0}), \ldots, (x_{i,m}, y_{i,m})]$ of coordinates of the vertices of the i^{th} obstacle.
- `"W"`: the width of each link.
- `"L"`: the length of each link.
- `"D"`: the distance between the two points of attachment on each link
- `"xI"`: a list $[i, j]$ specifying the initial configuration $x_I = (i, j) \in X$, and
- `"XG"`: a list of $[i, j]$'s, each corresponding to a goal configuration $x_G \in X_G$.

Task 1 (C-space and C-space obstacles) [10 points]: Discretize the C-space into 1-degree by 1-degree grid. So you'll have 360×360 grid, centered at $\{(i, j) \in \mathbb{Z} \times \mathbb{Z} \mid 0 \leq i \leq 359, 0 \leq j \leq 359\}$. The center of each grid cell represents a configuration. For example, the grid cell centered at $(0, 0)$ represents configuration $(0, 0)$, which corresponds to a configuration in which the two links lay flat horizontally, pointing to the right.

Compute the resulting 2D grid configuration similar to that in Project 1

$$G = \begin{bmatrix} O_{0,0} & O_{1,0} & \cdots & O_{M,0} \\ O_{0,1} & O_{1,1} & \cdots & O_{M,1} \\ \vdots & \vdots & \vdots & \vdots \\ O_{0,N} & O_{1,N} & \cdots & O_{M,N} \end{bmatrix}, \tag{1}$$

where $O_{i,j} \in \{0, 1\}$ indicates whether configuration (i, j) is in collision with the obstacles, i.e., $O_{i,j} = 1$ if and only if the robot at configuration (i, j) is in collision with the obstacles. Notice that when writing G as a 2-dimensional list, its indices are such that $G[i][j]$ corresponds to configuration (j, i).

Hint: For each of the 360×360 grid cells, you need to compute if the robot at the corresponding configuration is in collision with the obstacles. So you will need a function to detect the collision. If it is in collision, give the corresponding cell a value of 1, otherwise 0. Feel free to use any external library to check whether 2 rectangles overlap.

Task 2 (Path planning) [10 points]: The output from Task 1 is a grid configuration G similar to an input of Project 1. Use your favorite search algorithm that you implemented for Project 1 to compute a path from x_I to X_G.

Generate a json file containing the following fields:

- `"G"`: a 2-dimensional list representing the grid configuration G from Task 1.
- `"path"`: the list of cells specifying the path from x_I to X_G.

For example, if your code is `project2.py`, running

```
python project2.py project2_desc.json --out project2_out.json
```

should output `project2_out.json`, containing `"G"` and `"path"` for the problem described in `project2_desc.json`. Example of `project2_out.json` and `project2_desc.json` can be found on the course github repo.

Submission: Please submit a single zip file on Canvas containing the followings

- your code (with comments, explaining clearly what each function/class is doing), and
- a text file explaining clearly how to compile and run your code.