Problem 1: C-Spaces and Groups

(a) Explain or give a counterexample showing why $SE(2) \neq \mathbb{R}^3$

Special Euclidean Group, whose elements are called rigid motions. And, there are only translation and rotation in SEG. In a 2D case, there are only 2 DOF in translation, so $SE(2) \neq \mathbb{R}^3$.

(b) Explain or give a counterexample showing why $S^1 * S^1 \neq S^2$. Recall that S^1 denotes the unit circle in R^2; S^2 denotes (the surface of) the unit sphere in R^3.

Counterexample: 2-joint planar robot arm. It's C-space is not a sphere in R^3, but a torus in R^3.

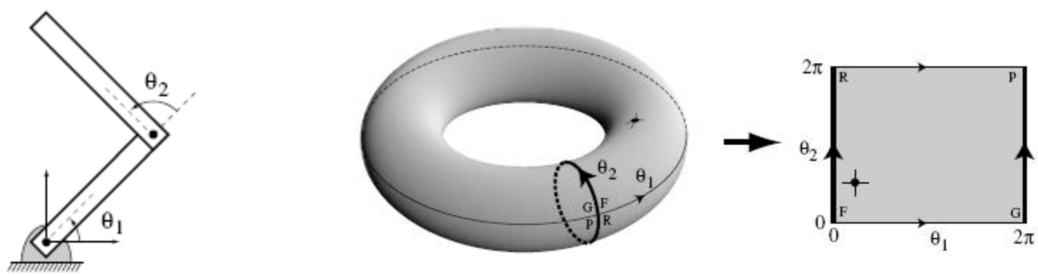

Figure 1: C-Space of Manipulators.

(c) Is $SE(2)$ commutative? In other words, does the ordering of translations and rotations in the plane matter? If yes, give a brief explanation; if no, give a simple counterexample.

The translation and rotation in $SE2)$ are relative to the robot as said in Ed107. Thus, $SE(2)$ is not commutative. For example, the translation result will change depending on the previous orientation of the robot.

Problem 2: C-space Obstacles

Consider the planar RR manipulator shown on the left below with its base fixed at the center. There are four obstacles of varying shapes in its work space. The corresponding "flattened" configuration space is shown on the right, with q_1 being the first joint angle and q_2 being the second (defined relative to the first link). Note that the origin of the C-space is in the lower left corner.

(a) Explain whether it is possible for the manipulator, starting from the shown configuration, toposition its end effector between the two circles. Provide a similar explanation for moving into a position between the lower circle and the rectangle. You should refer to the C-space picture in your explanation.

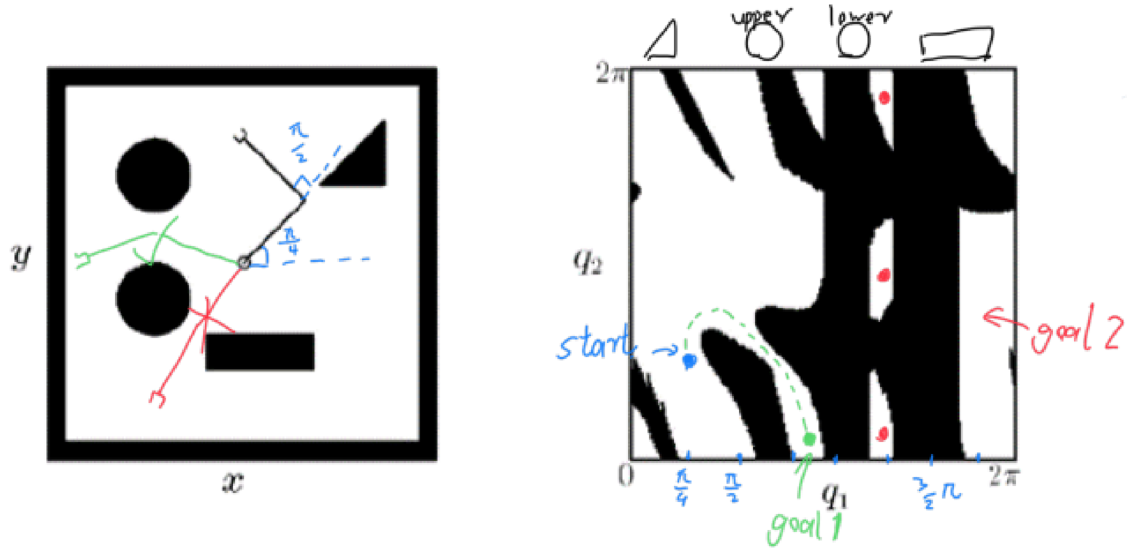

Figure 2: C-Space Obstacles (a)

See figure above as reference.

It is POSSIBLE for the manipulator to position its end effect between the two circles from the shown start configuration. However, it is IMPOSSIBLE to position its end effect between the lower circle and the rectangle. (green sketch refers to "between two circles", and red sketch refers to "between circle and rectangle")

First, the q_1 start configuration is at $\frac{\pi}{4}$, and the angle of q_2 is based on the orientation of q_1, so it's $\frac{\pi}{2}$ as shown in left picture.

Second, in right picture, the start configuration in "flattened" C-space is a blue point now.

Third, to be positioned between the two circles, the end effect should point to the left side as shown in left picture. That means the q_2 need to be around 0°.

Finally, there is one narrow path that can lead to the desired position (green goal1).

Similarly, to be positioned between the lower circle and the rectangle, the end effect should point to the left-bottom as shown in left picture. From the start configuration, there is no path that can reach the space with red dots in right picture. (The Cartesian coordinate is not a complete torus, even if we switch current configuration's direction out of boundary, there is no path available to these red dots.)

(b) See figure below as reference.

When $q_1 = \pi$ as as well as $q_2 = 3\pi/2$, these configurations all intersect an obstacle. It happens when the joint 2 (green sketch in picture) collide the obstacles. In this case, the link 2 will collide no matter the orientation it is.

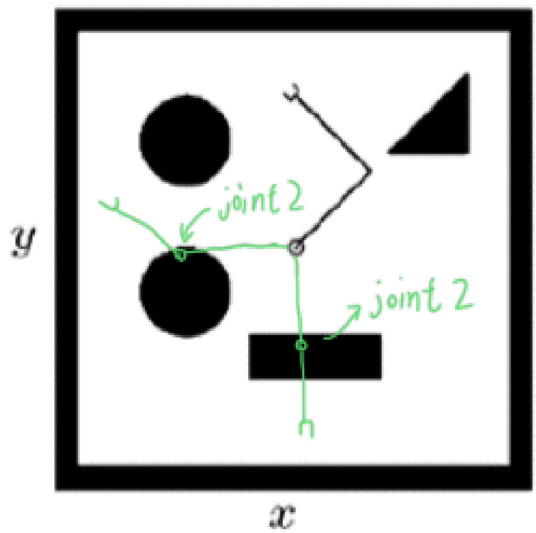

Figure 3: C-Space Obstacles (b)

Problem 3: Bug Algorithms

The environment shown below is similar to the one that was discussed in the Tangent Bug lecture (though somewhat distorted. Here we will consider switching the start and goal configurations, so that the robot travels from T to S.

(a) Make copies of the figure and draw the paths that Bug 1 and Bug 2 would return, assuming that both take right turns upon encountering an obstacle. You may approximate the closest point on the edge of each obstacle to S, within reason.

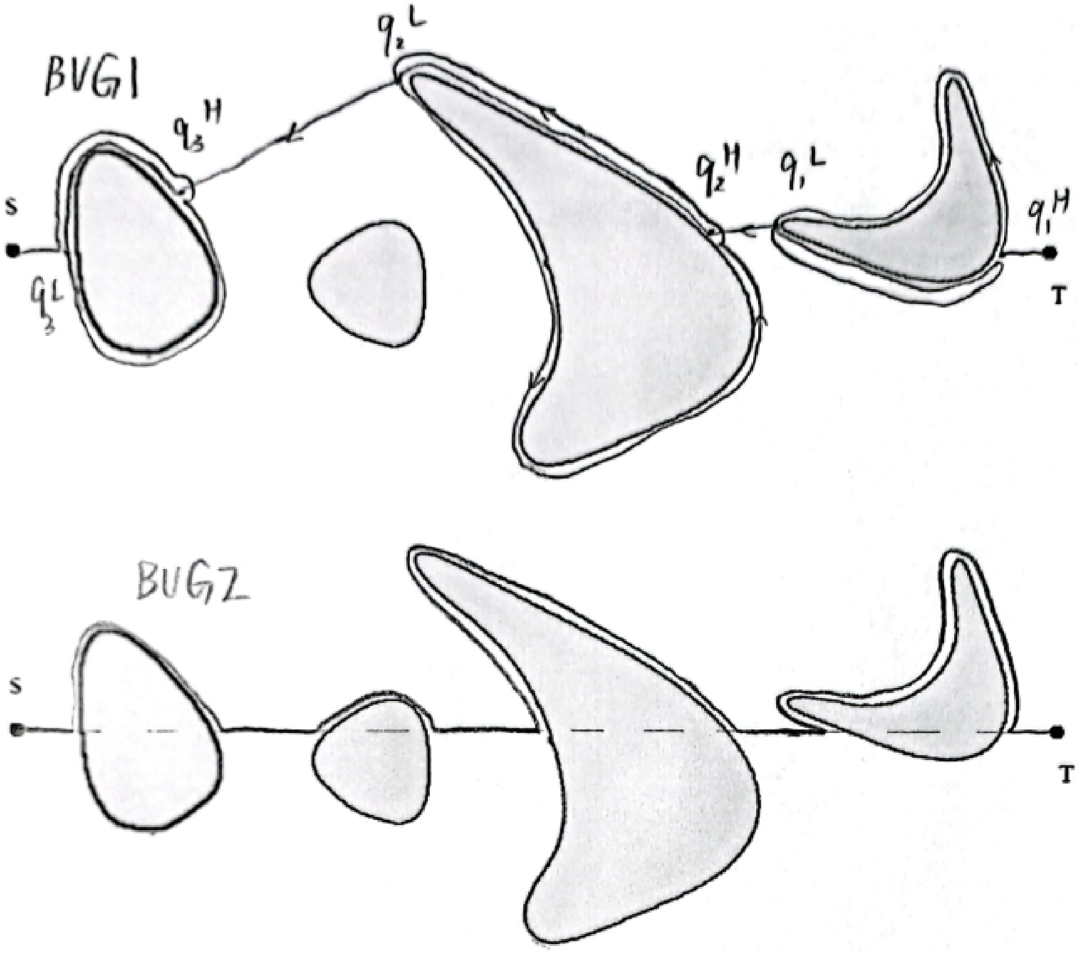

Figure 4: Bug1,bug2 path

Bug1 and Bug2 paths are shown above.

Bug1: The robot heads to the goal initially. If it reaches the obstacle, it circumnavigates the obstacle entirely. In circumnavigation, it will find the closet point to the goal. Then, it will leave from the closet point. Repeat the process until reaching the goal.

Bug2: The robot heads to the goal on it's m-line initially. If it reaches the obstacle, it follows the obstacle boundary until it reaches the m-line again, and it's distance to goal is smaller than the previous hit point. After that, repeat the process until reaching the goal.

(b) Repeat part (a) and come up with the paths for Tangent Bug with zero sensor range and infinite sensor range. Please briefly explain, for each algorithm, the direction of the initial path segment and the path behaviors upon reaching each obstacle

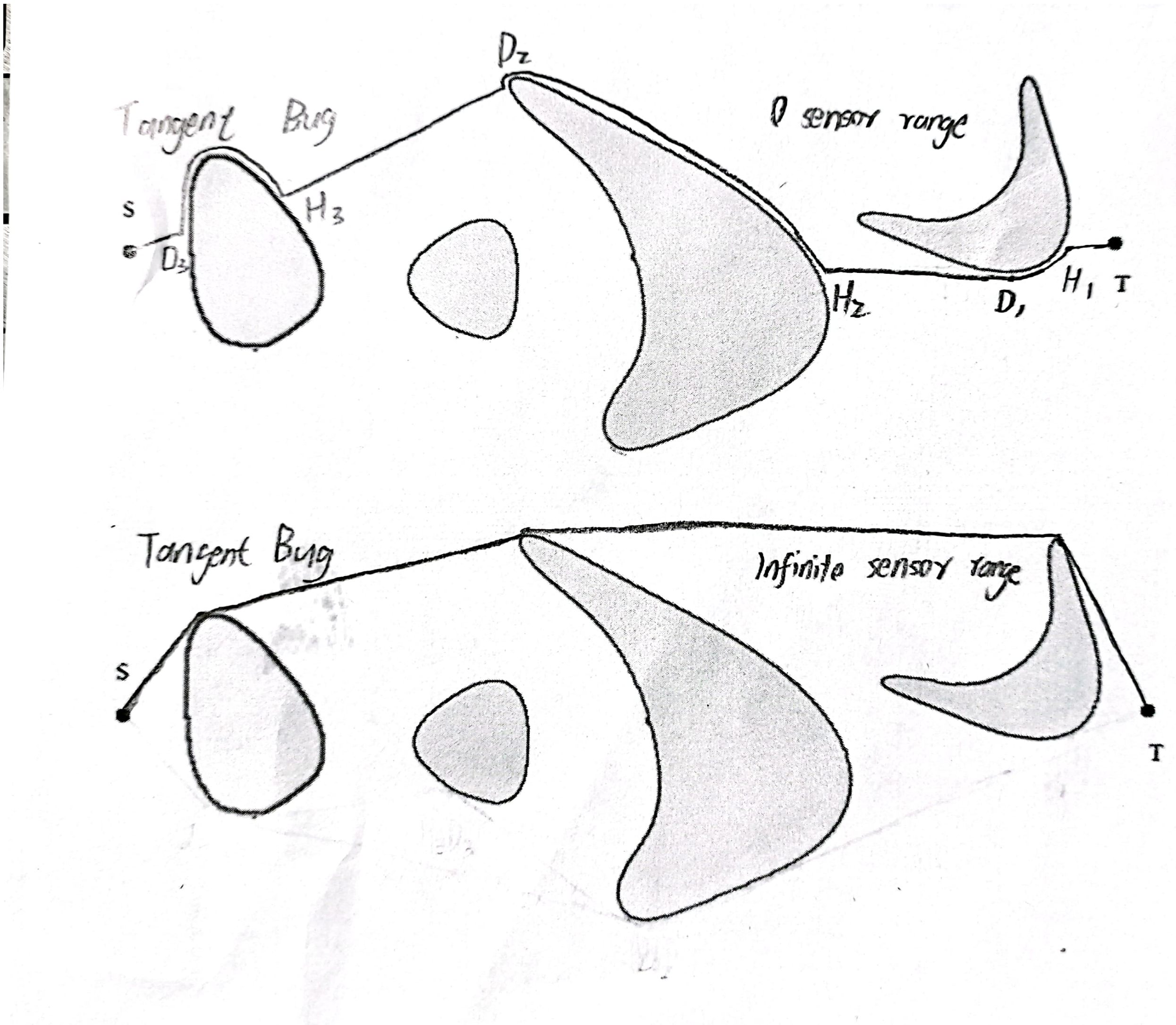

Figure 5: Zero/Infinite tangent bug

Bug1 and Bug2 paths are shown above.

Zero sensor range tangent bug: First, we can assume that this is a tiny sensor field around the robot. The robot heads to the goal initially. During the path, the robot will keep tracking the distance to make sure $d(x, n) + d(n, q_{goal})$ is shortest . It chooses the direction that is closer to goal. If it reaches the obstacle, it follows the boundary until find another closest distance.

Infinite sensor range tangent bug: The robot detects each visible obstacle's boundary extent to get around it. So it doesn't head to the goal initially, instead it heads to the extension where distance $(d(x,n) + d(n, q_{goal}))$ is shortest. (We use parallelogram rule to calculate the shortest distance). If it reaches the obstacle, it keep tracking the shortest distance to decide if it follows the obstacle or not.

In both cases, $d_{reach} < d_{followed}$ during the path, so the distance always decreases.

Problem 4: Potential Functions

Consider an additive attractive/repulsive potential function on $\mathbb{R}^2$. The goal configuration is located at $g_{goal} = (0, 0)$, which induces a combined conic and quadratic attractive potential with parameters $d^*_{goal} = 1$ and $\zeta = 1$. There are two circular obstacles, each with radius $r_1 = 1$ and centered at $q_1 = (2, 0) and q_2 = (2, 0)$, respectively. Each induces a repulsive potential with parameters $Q^* = 1$ and $\eta = 1$. All distances are computed using the Euclidean metric.

(a) Create a streamline plot of the potential function around the origin. A program-generated plot would be awesome, but in this case it would suffice (and probably be less tedious) to draw one by hand. Please include a representative number of lines in each quadrant, showing the effects of the repulsive potentials.

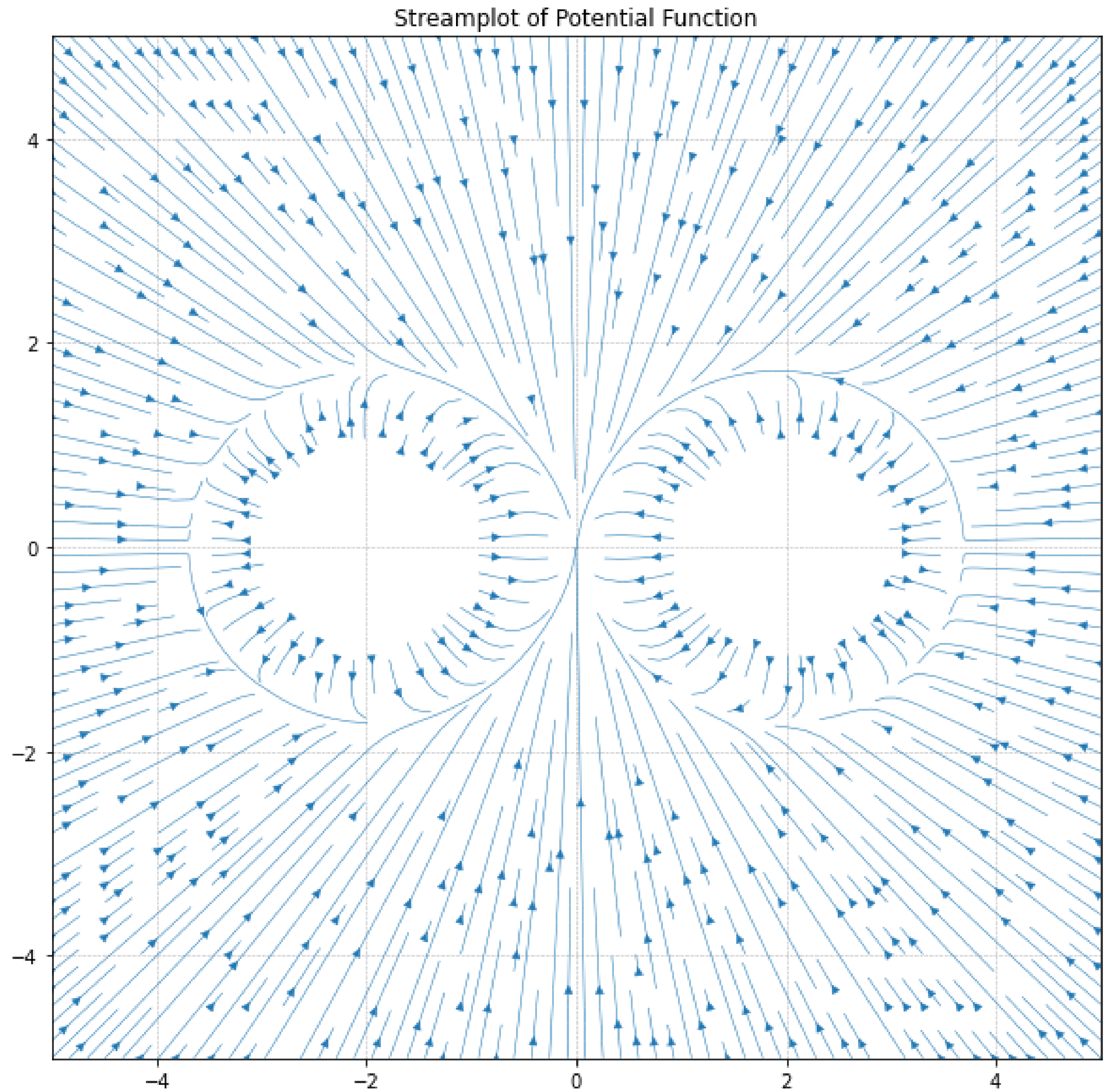

Figure 6: Stream Plot of Potential Function

You can easily see the repulsive potentials boundary in figure above.

(b) Compute the potential and gradient at the configuration $q = (0.5, 0.5)$, Which other configurations have the same potential and gradient? Why you do not need to perform further computation to find out what they are?

We have: $d(q, q_g) \le d^*; d_1(q) \le Q^*$

Potential:

$$
\begin{aligned}
U_t(q) &= U_a(q) + \sum U_{r,i}(q) \\
U_t(q) &= \frac{1}{2}\zeta d(q, q_g)^2 + \frac{1}{2}\eta(\frac{1}{d_1(q)} - \frac{1}{Q^*})^2 \\
U_t(q) &= \frac{1}{2}(\frac{\sqrt{2}}{2})^2 + \frac{1}{2}(\frac{1}{\frac{-2+\sqrt{10}}{2}} - \frac{1}{1})^2 \\
U_t(q) &= 0.51
\end{aligned}
\tag{1}
$$

In above equation, $d(q, q_g) = \frac{\sqrt{10}}{2}$, and $d_1(q) = \frac{-2+\sqrt{10}}{2}$

Gradient:

$$
\begin{aligned}
\bigtriangledown U_t(q) &= \bigtriangledown U_a(q) + \sum \bigtriangledown U_{r,i}(q) \\
\bigtriangledown U_t(q) &= \zeta(q - q_g) + \frac{\eta}{d_i^2(q)}(\frac{1}{Q^*} - \frac{1}{d_i(q)})\frac{q-c}{d(q,c)} \\
\bigtriangledown U_t(q) &= (0.5, 0.5) - (0, 0) + \frac{1}{(\frac{-2+\sqrt{10}}{2})^2}(\frac{1}{1} - \frac{1}{\frac{-2+\sqrt{10}}{2}})\frac{(0.5, 0.5) - (1.05132, 0.31623)}{\frac{-2+\sqrt{10}}{2}} \\
\bigtriangledown U_t(q) &= (2.525, -0.175)
\end{aligned}
\tag{2}
$$

In above equation, c is the closest point on convex obstacle, and $d_1(q) = d(q, c) = \frac{-2+\sqrt{10}}{2}$

$(0.5, 0.5), (-0.5, 0.5), (-0.5, -0.5), (0.5, -0.5)$, they have the same potential and gradient magnitude because the stream plot is symmetric in both x, y axis.

(c) There are two saddle points (orange dots in figure below) other than the goal configuration. The streamlines directly collide here. No local minimum and local maximum.

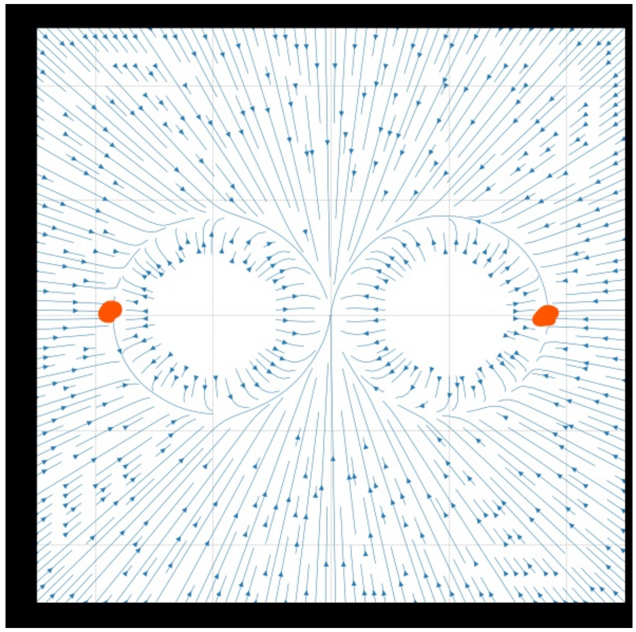

Figure 7: Stream Plot of Potential Function

Problem 5: C-space and Bug Algorithm Implementation

(a) The program firstly put the reference point at the original point and reflect the robot. Then, put robot's reference point at each vertex of the obstacle. Finally, we use ConvexHull function in Scipy to connect points forming the simplical facets of the convex hull.

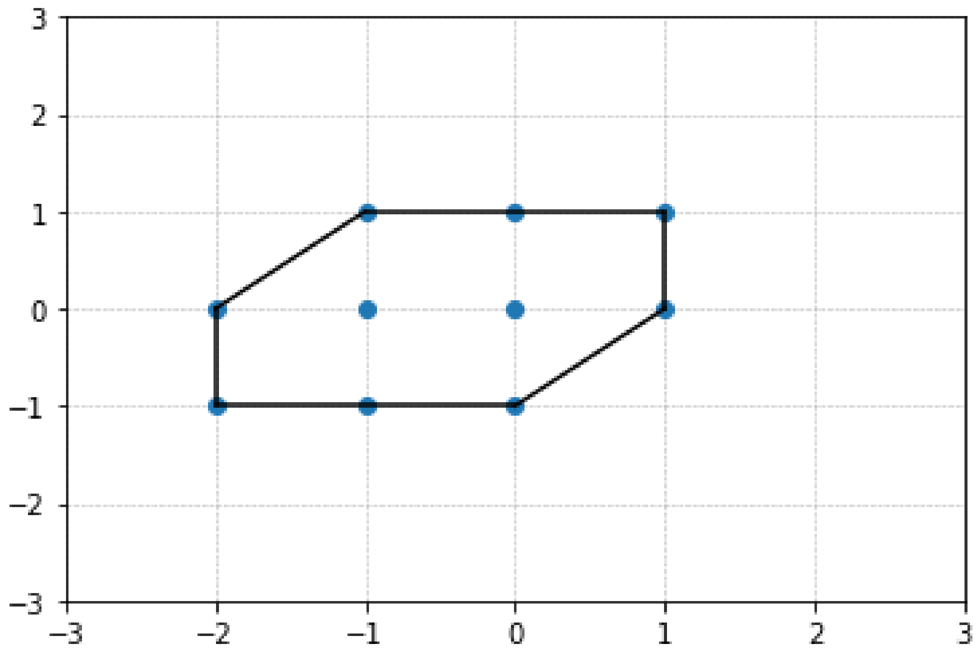

Figure 8: C-space Obstacle - 1

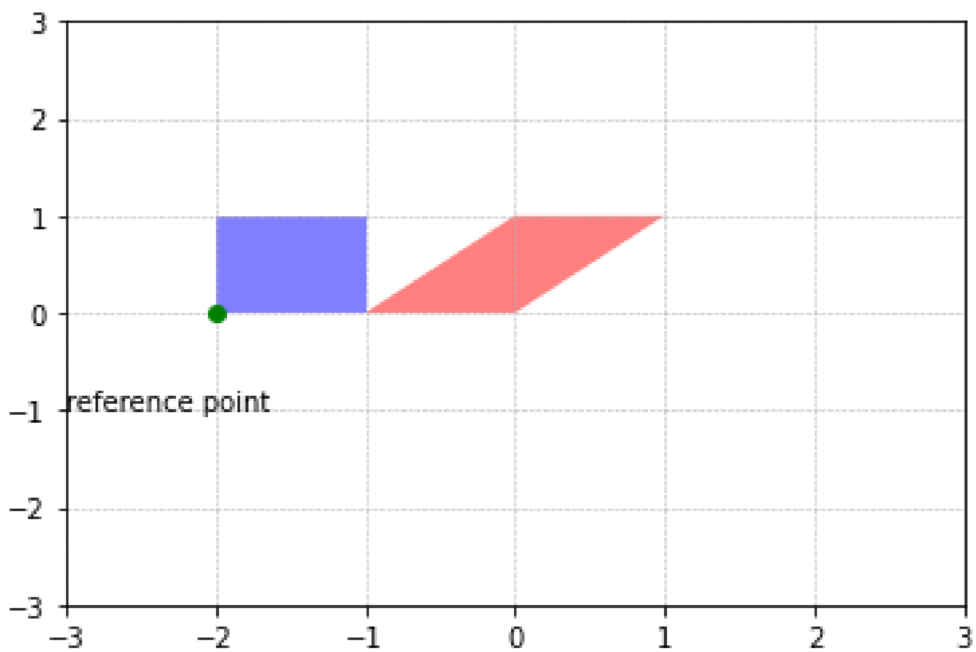

Figure 9: Robot and Obstacle Before - 1

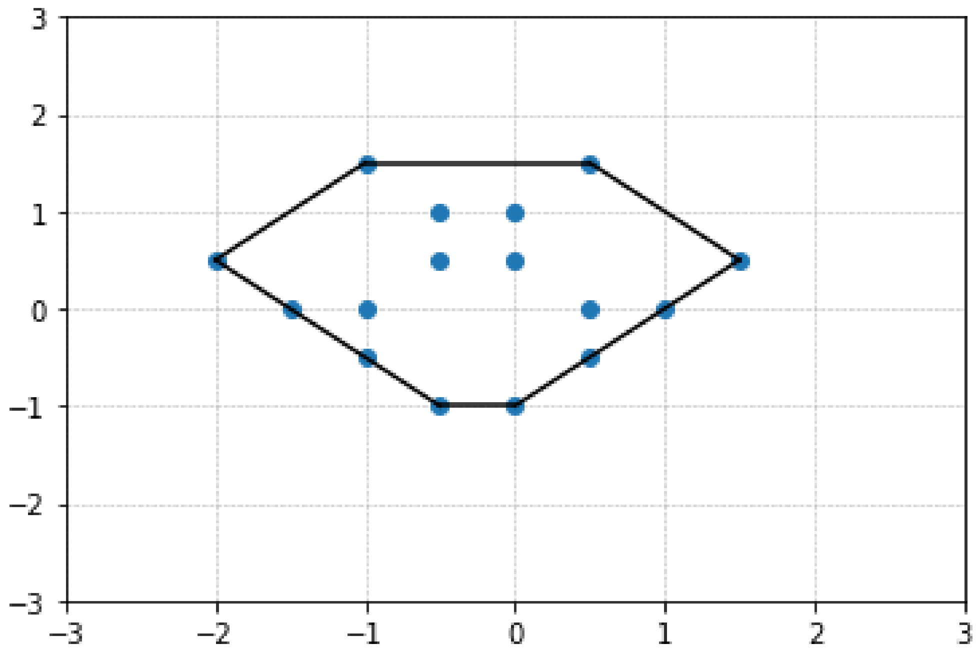

Figure 10: C-space Obstacle - 2

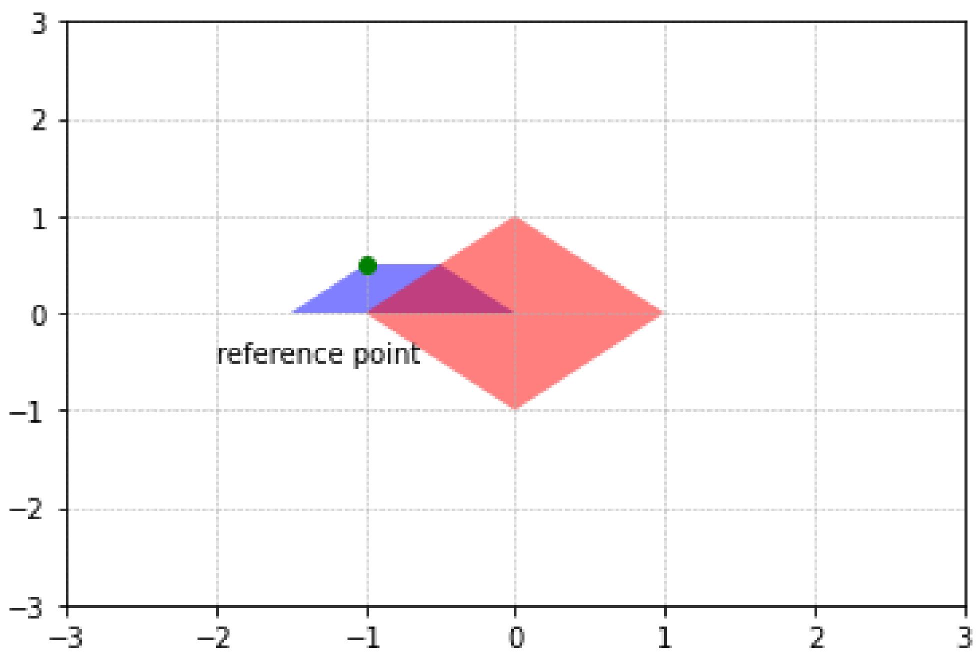

Figure 11: Robot and Obstacle Before - 2

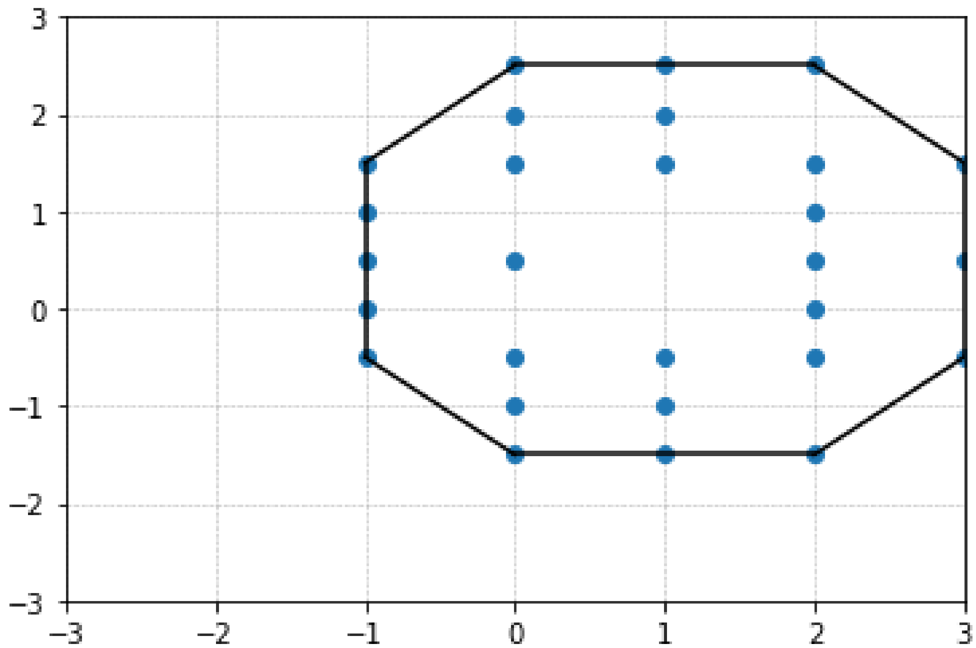

Figure 12: C-space Obstacle - 3

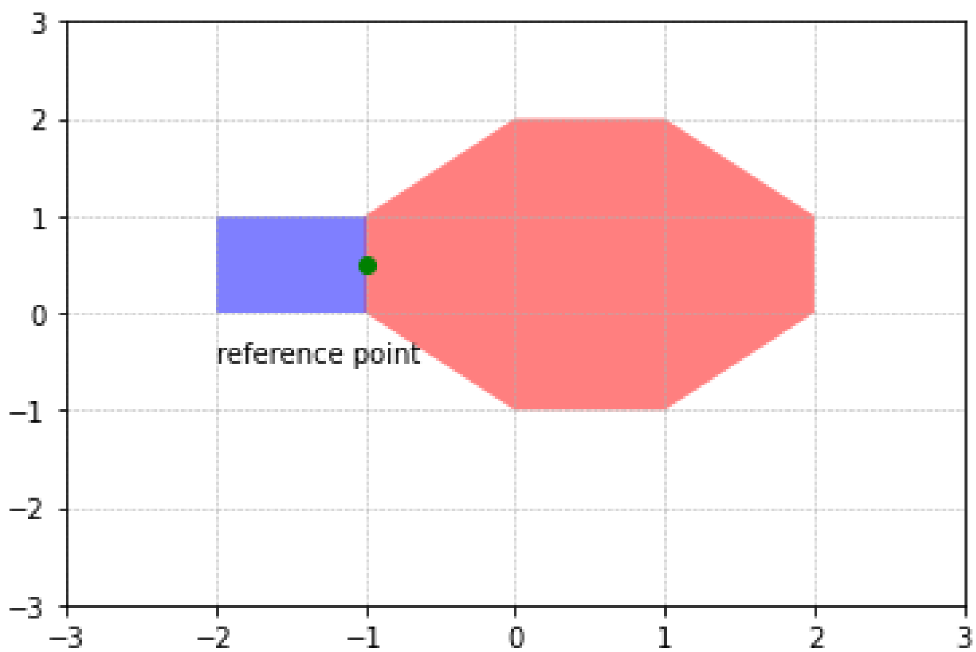

Figure 13: Robot and Obstacle Before - 3

(b) This program firstly defines the start and end points of the robot, the obstacle vertices, and then it makes the robot travel toward the end goal. The moving forward step size and the rotate angle size are input by user. We get the expression of the line between the start and end point. For every step the robot makes, it will first check if the next position toward the leave point will collide with the obstacle or not, if so, it rotates an angle size and check it again until it will not collide. And it will remain this current angle and move one step to get to the next point until it reaches the leave point. If not, it will go one step closer to the leave point remaining the current angle.
The following figures show the three examples we have tried.

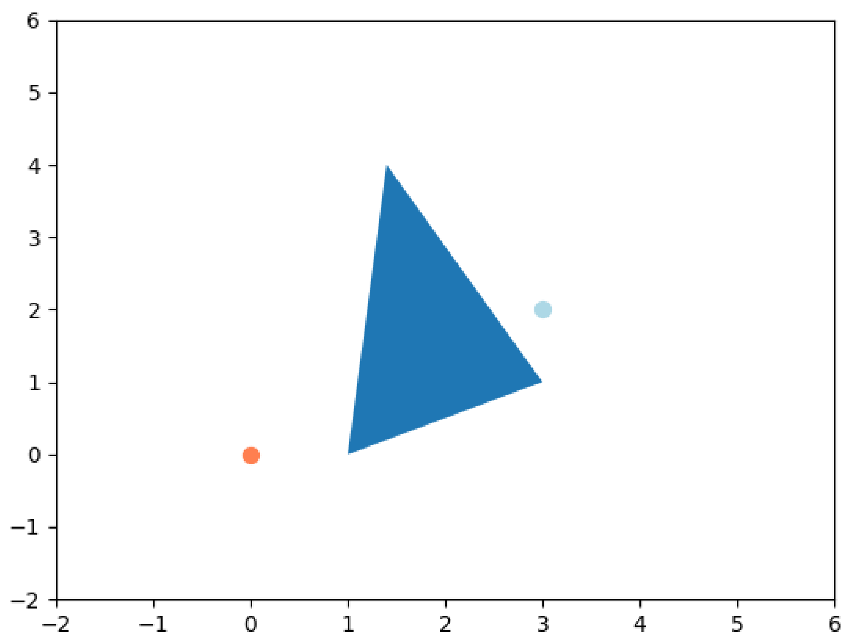

Figure 14: Example 1 obstacle

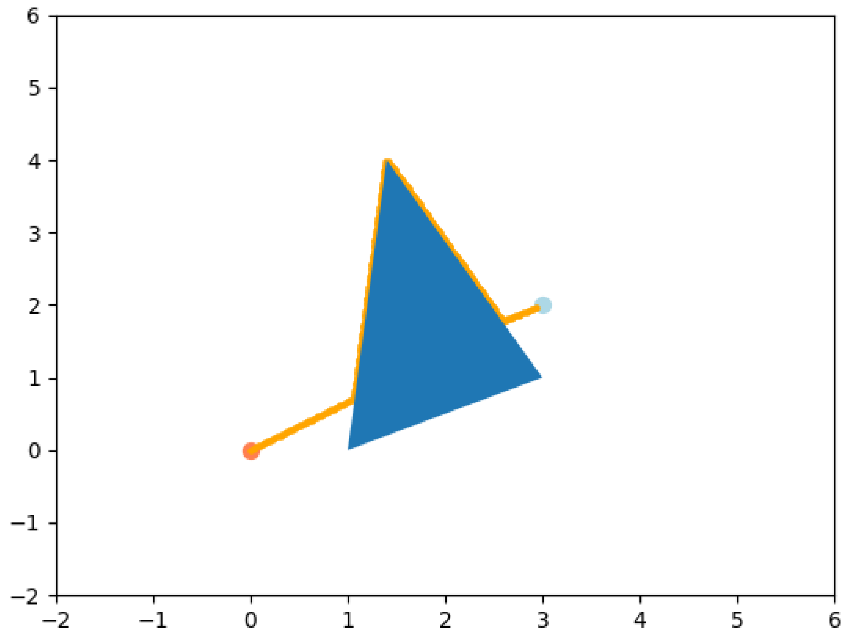

Figure 15: Example 1 path

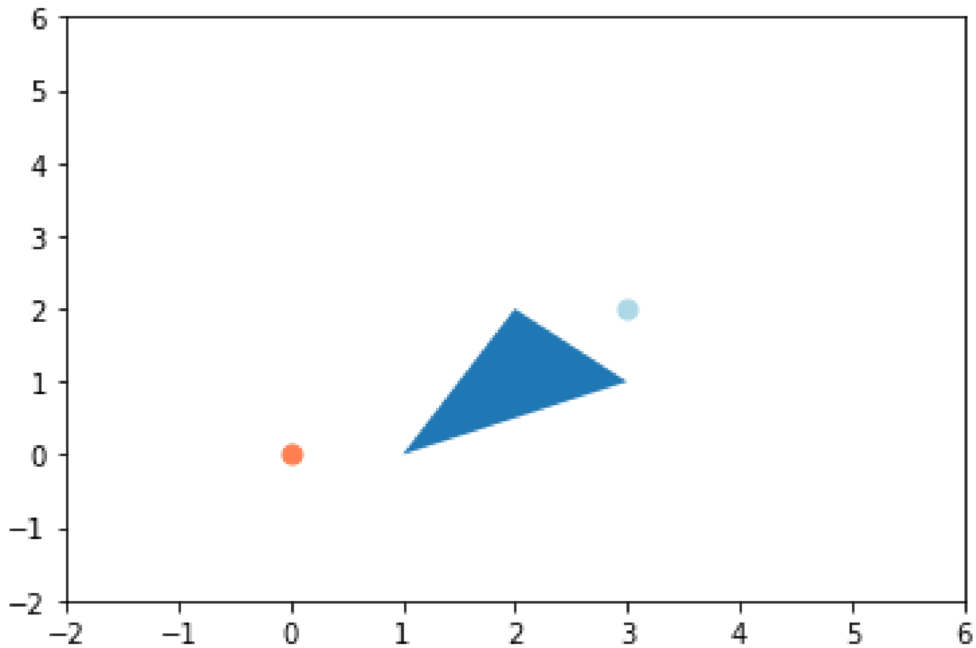

Figure 16: Example 2 obstacle

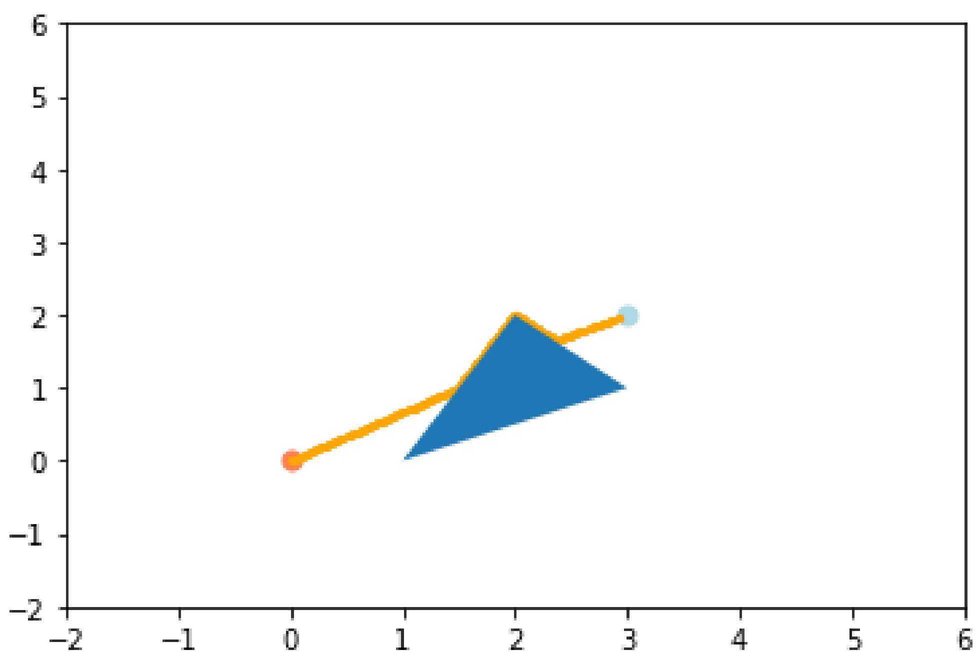

Figure 17: Example 2 path

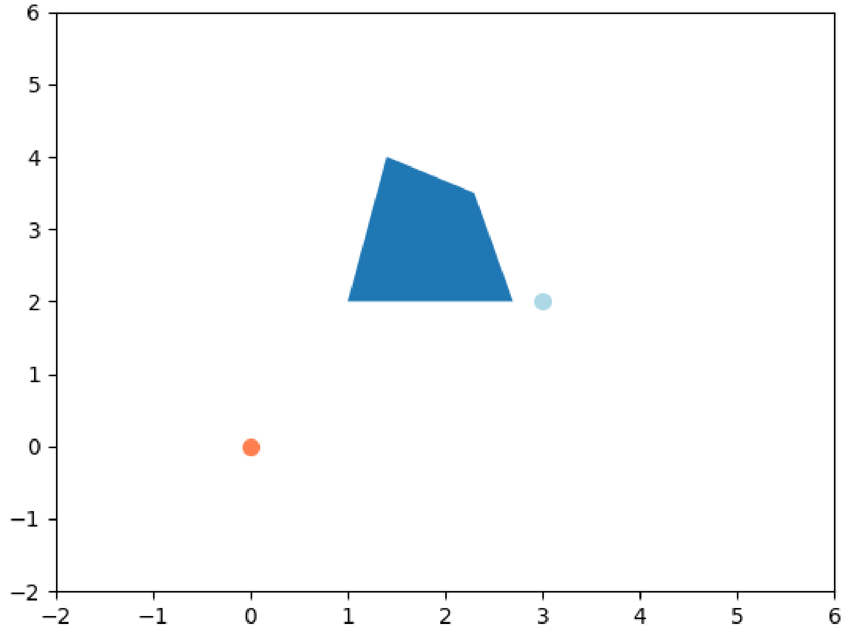

Figure 18: Example 3 obstacle

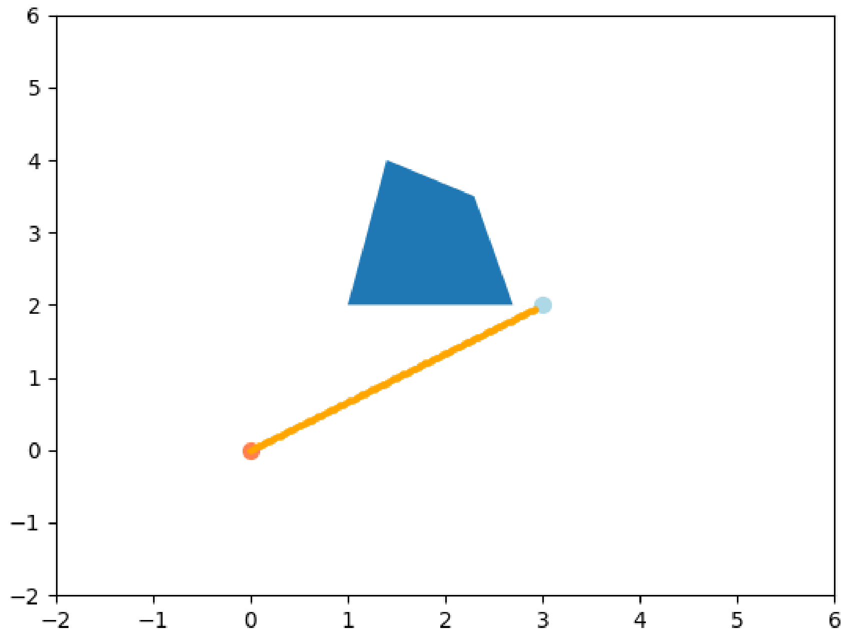

Figure 19: Example 3 path