COMS 576 - Motion Planning for Robotics and Autonomous Systems, Homework 3

Mobina Amrollahi

Deadline: 10/11/2024, 11:59 PM

1 (10 points for COM S 4760, 7 points for COM S 5760) Configuration Space:

1.1 Answer

(a) Since the robot is circular base with radius r, we can assume it's a circle entered at (x_i, y_i), and has a radius r. Therefore, the perimeter, as well as the interior points within the corresponding circle, represent the semi-algebraic model of the robot, which is the following:

$$\mathcal{H} = \{(x, y) \in \mathcal{W} | (x - x_i)^2 + (y - y_i)^2 - r^2 \leq 0\}$$

(b) Based on the statement, first a rotation by θ is performed and then its followed by translation by $T(d)$. Hence, to find the region in the world occupied by the robot, we need to figure out the following homogeneous transformation matrix where:

$$S = \begin{bmatrix} R & v \\ 0 & 1 \end{bmatrix}$$

- The R part corresponds to the rotation. Since the robot is operating in a 2D world, the rotation matrix is:

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

- The v part corresponds to the translation. Suppose, the intial point is (x_i, y_i), and the robot moves within the distance d to some (x_g, y_g). Then, we know:

$$\sqrt{(x_g - x_i)^2 + (y_g - y_i)^2} = d$$

  Hence, if the translation along x-axis is Δx and translation along y-aixs is Δy, the following holds:

$$\Delta x = d\cos\theta, \Delta y = d\sin\theta$$

  Therefore, the translation can be represented as:

$$q = (d\cos\theta, d\sin\theta) \rightarrow h(q, (x, y)) = (x + d\cos\theta, y + d\sin\theta)$$

The translation vector is the following:

$$v = \begin{bmatrix} d\cos\theta \\ d\sin\theta \end{bmatrix}$$

Finally, the homogeneous transformation matrix describing the robot which performs the rotation $R(\theta)$ followed by a translation $T(d)$ is the following:

$$S = \begin{bmatrix} \cos\theta & -\sin\theta & d\cos\theta \\ \sin\theta & \cos\theta & d\sin\theta \\ 0 & 0 & 1 \end{bmatrix}$$

To determine the region in the world occupied by the robot after this transformation, we will take (x_i, y_i) to see where it ends up after this transformation and use the new center to update the robot's semi-algebraic model. Thus, we have:

$$\begin{bmatrix} x_{new} \\ y_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & d\cos\theta \\ \sin\theta & \cos\theta & d\sin\theta \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} x_{new} \\ y_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta x_i - \sin\theta y_i + d\cos\theta \\ \sin\theta x_i + \cos\theta y_i + d\sin\theta \\ 1 \end{bmatrix}$$

$$\Rightarrow (x_{new}, y_{new}) = (\cos\theta x_i - \sin\theta y_i + d\cos\theta, \sin\theta x_i + \cos\theta y_i + d\sin\theta)$$

Finally, we will update the semi-algebraic model by replacing (x_i, y_i) with (x_{new}, y_{new}). Hence, we will get:

$$\mathcal{H}^{'} = \{(x, y) \in \mathcal{W} | (x - (\cos\theta x_i - \sin\theta y_i + d\cos\theta))^2 + (y - (\sin\theta x_i + \cos\theta y_i + d\sin\theta))^2 - r^2 \leq 0\}$$

(c) Based on the statement, the robot needs to do the following sequence of actions:

1. $R(\theta_1)$: Rotate from θ_i to the goal, where it is directly facing the goal,
2. $T(d)$: Translate from (x_i, y_i) to (x_g, y_g) by going forward as much of the Euclidean distance of these two points,
3. $R(\theta_2)$: Rotate from the goal angle to θ_g.

We will be calculating $R_1(\theta)$, $T(d)$, and $R_2(\theta)$ respectively.

1. $R(\theta_1)$: Assume θ_{goal} is the direct angle from the initial point to the goal point, which is measured with respect to the world frame, and the rotation direction is counterclockwise. Therefore, $\tan(\theta_{goal})$ represents the ratio between the vertical and horizontal distances from the robot's current position to the goal. These distances form a right triangle, with one leg along the horizontal direction (difference in x-coordinates) and the other leg along the vertical direction (difference in y-coordinates). Thus:

$$\tan(\theta_{goal}) = \frac{y_g - y_i}{x_g - x_i}$$

We need to know the value of θ_{goal} to rotate from θ_i to this angle, facing the goal. Therefore to calculate θ_{goal}:

$$\tan(\theta_{goal}) = \frac{y_g - y_i}{x_g - x_i} \Rightarrow \theta_{goal} = \arctan(\frac{y_g - y_i}{x_g - x_i})$$

Finally, the robot needs to rotate $\theta_1 = \theta_{goal} - \theta_i$ to face the goal. Thus:

$$R(\theta_1) = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 \\ \sin\theta_1 & \cos\theta_1 \end{bmatrix} = \begin{bmatrix} \cos(\theta_{goal} - \theta_i) & -\sin(\theta_{goal} - \theta_i) \\ \sin(\theta_{goal} - \theta_i) & \cos(\theta_{goal} - \theta_i) \end{bmatrix}$$

2. $T(d)$: After facing the goal, the robot needs to translate forward as much of the distance (x_i, y_i) and (x_g, y_g) have, which is the following:

$$d = \sqrt{(x_g - x_i)^2 + (y_g - y_i)^2}$$

Thus:

$$T(d) = \begin{bmatrix} \sqrt{(x_g - x_i)^2 + (y_g - y_i)^2}\cos\theta \\ \sqrt{(x_g - x_i)^2 + (y_g - y_i)^2}\sin\theta \\ 1 \end{bmatrix}$$

3. $R_2(\theta)$: Lastly, the robot needs to rotate from θ_{goal} to θ_g. This means the robot needs to rotate $\theta_2 = \theta_g - \theta_{goal}$. Hence:

$$R(\theta_2) = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 \\ \sin\theta_2 & \cos\theta_2 \end{bmatrix} = \begin{bmatrix} \cos(\theta_g - \theta_{goal}) & -\sin(\theta_g - \theta_{goal}) \\ \sin(\theta_g - \theta_{goal}) & \cos(\theta_g - \theta_{goal}) \end{bmatrix}$$

(d) The configuration space $\mathcal{C}$ of the robot is a manifold which is the set of all possible transformations of a robot with n degree of freedom. Since the robot is operating in a 2D workspace, then its homogeneous transformation matrix is:

$$S = \begin{bmatrix} \cos\theta & -\sin\theta & d\cos\theta \\ \sin\theta & \cos\theta & d\sin\theta \\ 0 & 0 & 1 \end{bmatrix}$$

And we write:

$$\left\{ \begin{pmatrix} R & v \\ 0 & 1 \end{pmatrix} \;\middle|\; R \in SO(n) \text{ and } v \in \mathbb{R}^2 \right\}$$

1. v: corresponds to translation.
2. R: corresponds to the direction.

This group which describes the possible rigid transformations that can be applied to the robot's configuration in the 2D plane, is called the special Euclidean group or $SE(2)$. $SE(2)$ is **the mathematical structure** associated with the robot's configuration space $\mathcal{C}$.

- Rotation and translation are independent $\Rightarrow SE(2)$ is homeomorphic to $\mathbb{R}^2 \times SO(2)$
- $SE(2)$ is homeomorphic to $\mathbb{R}^2 \times \mathbb{S}^1 \Rightarrow$ The C-space of a 2D rigid body is homeomorphic to $\mathcal{C} = \mathbb{R}^2 \times \mathbb{S}^1$

Thus, the configuration of the robot in this 2D space is completely **represented** by the tuple (x, y, θ), where it performs a rotation by θ, followed by a translation by (x_t, y_t):

- Any $x_t, y_t \in \mathbb{R}$ can be selected for translation $\Rightarrow$ Manifold $M_1 = \mathbb{R}^2$
- Any $\theta \in [0, 2\pi)$ can be selected for rotation $\Rightarrow$ Manifold $M_2 = \mathbb{S}^1$

Hence, the **3D manifold** for all rigid-body motions is a cylinder:

$$\mathcal{C} = M_1 \times M_2 = \mathbb{R}^2 \times \mathbb{S}^1 = \mathbb{R}^2 \times [0, 2\pi)$$

Each configuration $q \in \mathcal{C}$ of the robot is **uniquely** represented by the three parameters x, y, and θ. Thus, a configuration is expressed as the triple:

$$q = (x, y, \theta) \in \mathbb{R}^2 \times [0, 2\pi)$$

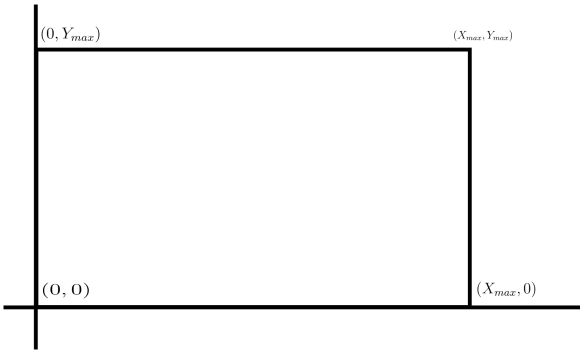

Figure 1: Rectangle Representing the 2D World

2 (10 points for COM S 4760, 7 points for COM S 5760) C-Space Obstacles:

2.1 Answer

(a) Based on the statement, the task is to define a set of algebraic primitives that can be combined to define $\mathcal{C}_{obs}$. $\mathcal{C}_{obs}$ is the set of configurations $q \in \mathcal{C}$ where $\mathcal{A}(q)$ collides with the $\mathcal{O}$ where q represents a point such as (x, y) in the configuration space. Thus, $0 \leq x \leq X_{max}$ and $0 \leq y \leq Y_{max}$.
To ensure the entire robot wouldn't collide with the obstacle, we have to make sure that the center of both circular entities, the robot, and the obstacle, is more the summation of entities radius, which in this case is $r + r_0$. Suppose the robot centers at (x, y). Since the obstacle is centered at (x_0, y_0), then the following should hold:

$$\sqrt{(x - x_0)^2 + (y - y_0)^2} \leq r + r_0$$

Hence:

$$(x - x_0)^2 + (y - y_0)^2 \leq (r + r_0)^2$$

And:

$$(x - x_0)^2 + (y - y_0)^2 - (r + r_0)^2 \leq 0$$

Thus $f(q, x_0, y_0, r_0)$ is:

$$f(q, x_0, y_0, r_0) = (x - x_0)^2 + (y - y_0)^2 - (r + r_0)^2$$

And $\mathcal{C}_{obs,o}$ is the set of points where the robot collides with the obstacle:

$$\mathcal{C}_{obs,o} = \{(x, y) \in \mathcal{C} | (x - x_0)^2 + (y - y_0)^2 - (r + r_0)^2 \leq 0\}$$

(b) First, we need to figure out the configuration space that represents the area of the rectangle, and then we will calculate the area outside of this rectangle. Please note that the area that the robot operates in is a closed set of $[0, X_{max}] \times [0, Y_{max}]$, which contains $(0, 0), (0, Y_{max}), (X_{max}, 0)$, and (X_{max}, Y_{max}). Hence the edges and vertices of this rectangle are part of the configuration space as well. Based on the figure 1, this rectangle consists of four following edges:

$$
\begin{aligned}
&l_1 : y = Y_{max} \\
&l_2 : y = 0 \\
&l_3 : x = X_{max} \\
&l_4 : x = 0
\end{aligned}
$$

Therefore, l_1, l_2, l_3 and l_4 are the functions of the lines passing through the edges of the rectangle. To completely represent the area within the rectangle, we will pick a point inside the rectangle, for instance, $\mathcal{P} = (\frac{X_{max}}{2}, \frac{Y_{max}}{2})$, and if we plug this point into l_1 and l_3 the result shouldn't be positive, and if we plug it into l_2 and l_4, the result shouldn't be negative. Thus:

$$
\begin{aligned}
&S_1 : \frac{Y_{max}}{2} - Y_{max} \leq 0 \\
&S_2 : 0 - \frac{Y_{max}}{2} \leq 0 \\
&S_3 : \frac{X_{max}}{2} - X_{max} \leq 0 \\
&S_4 : 0 - \frac{X_{max}}{2} \leq 0
\end{aligned}
$$

Hence, the following represents the area inside the rectangle:

$$
\begin{aligned}
&S_1 : Y_{max} - y \geq 0 \\
&S_2 : y \geq 0 \\
&S_3 : X_{max} - x \geq 0 \\
&S_4 : x \geq 0
\end{aligned}
$$

Therefore, the following represents the workspace area outside:

$$
\begin{aligned}
&f_1 = Y_{max} - y < 0 \\
&f_2 = y < 0 \\
&f_3 = X_{max} - x < 0 \\
&f_4 = x < 0
\end{aligned}
$$

Hence, if assuming H_1, H_2, H_3 and H_4 are the algebraic primitives of f_1, f_2, f_3 and f_4, then they're the following:

$$
\begin{aligned}
&H_1 = \{(x, y) \in \mathcal{W} | Y_{max} - y < 0\} = \{(x, y) \in \mathcal{W} | f_2 < 0\} \\
&H_2 = \{(x, y) \in \mathcal{W} | y < 0\} = \{(x, y) \in \mathcal{W} | f_2 < 0\} \\
&H_3 = \{(x, y) \in \mathcal{W} | X_{max} - x < 0\} = \{(x, y) \in \mathcal{W} | f_3 < 0\} \\
&H_4 = \{(x, y) \in \mathcal{W} | x < 0\} = \{(x, y) \in \mathcal{W} | f_4 < 0\}
\end{aligned}
$$

Thus, if we consider the area outside the workspace boundaries as $\mathcal{O}$, then:

$$
\mathcal{O} = H_1 \cup H_2 \cup H_3 \cup H_4 \tag{1}
$$

Finally, to find the complete $\mathcal{C}_{obs}$ we must take union of all $\mathcal{C}_{obs,o}$ and $\mathcal{O}$.

$$
\mathcal{C}_{obs} = \mathcal{O} \cup \left(\bigcup_{o \in \mathcal{O}_{circles}} \mathcal{C}_{obs,o} \right)
$$

Where $\mathcal{O}_{circles}$ is the set of all circular obstacles in the workspace.

$$\Rightarrow \mathcal{C}_{obs} = H_1 \cup H_2 \cup H_3 \cup H_4 \cup \left(\bigcup_{o \in \mathcal{O}_{circles}} \{(x,y) \in \mathcal{W} \mid (x - x_o)^2 + (y - y_o)^2 \leq (r + r_o)^2\} \right)$$

3 (COM S 5760 only, 6 points)

(a) It is possible to formulate a discrete motion planning problem for the robot. We will be given the resulting state space X, the set of actions U, the action space $U(x)$ for each state $x \in X$, and the state transition function $f : X \times U \to X$.

1. State space X: In this problem, the state space is finite and countable, which represents all the possible configurations of the robot. Since the robot can only move between these points, transitioning only to directly neighboring points in the grid—either horizontally or vertically X can be defined as a set of all pairs $(x, y, \theta)\backslash\mathcal{O}$, where $\mathcal{O}$ refers to the set of obstacles, (x, y) are the coordinates of the grid points, with $x \in \{0, 1, \ldots, X_{\max}\}$ and $y \in \{0, 1, \ldots, Y_{\max}\}$. Finally, $\theta \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$, since the robot is facing along positive and negative x-axis and y-axis, and can only move horizontally and vertically, not diagonal. Thus, the state space is:

$$X = \{(x, y, \theta) | x \in \{0, 1, \ldots, X_{\max}\}, y \in \{0, 1, \ldots, Y_{\max}\}, \theta \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}\}\backslash\mathcal{O}$$

2. Set of actions U: Based on the description and problem 1, in order for the robot to move to a specific grid, first, it has to rotate to a certain amount θ (θ can be zero if the robot is already facing the goal grid), and then move forward. Thus, there are two sets of actions the robot can do. Rotating counter-clockwise by angle θ, and then moving forward by $d = 1$ unit, which unit is the distance between neighboring grids. First, we will show moving up, down, left, and right is possible by using a sequence of transformations derived from problem 1c:
   - Up: rotate to $\theta = \frac{\pi}{2}$. Then, take another action of moving forward by 1 unit.
   - Down: rotate to $\theta = \frac{3\pi}{2}$. Then, take another action of moving forward by 1 unit.
   - Left: rotate to $\theta = \pi$. Then, take another action of moving forward by 1 unit.
   - Right: rotate to $\theta = 0$. Then, take another action of moving forward by 1 unit.

   Thus, the action space U contains two sets of action:

$$U = \{(0, 0, \theta), (1, 0, 0), (0, 1, 0)\}$$

3. Action space $U(x)$: For each state $x = (x, y, \theta) \in X$:

$$U(x) = \{u \in U \mid f(x, u) \in X\} \text{ and } U = \bigcup_{x \in X} U(x)$$

4. State transition function $f : X \times U \to X$: For each state $x = (x, y, \theta)$ and action $u \in U(x)$, the state transition function f is defined as follows:

$$f(x, u) = (x + u_x, y + u_y, \theta + \theta_u), \forall x = (x, y, \theta) \in X, u = (u_y, u_x, \theta_u) \in U(x)$$

(b) To determine the grid points that the robot can safely occupy without causing a collision with the workspace boundaries or circular obstacles, we need to take into account the robot's radius $r = 0.5$ and the obstacles, as well as their positions.

- Workspace boundaries: The workspace is a grid defined by $0 \leq x \leq 6$ and $0 \leq y \leq 6$. The robot must be positioned such that its radius does not extend beyond the workspace boundaries. Since the robot has a radius of 0.5, thus:

$$0.5 \leq x \leq 5.5 \quad \text{and} \quad 0.5 \leq y \leq 4.5$$

  This implies that the valid x-values are: $x = \{1, 2, 3, 4, 5\}$ and the valid y-values are: $y = \{1, 2, 3, 4\}$.

- Not having a collision with obstacles: The robot's center must remain at least $r + r_{\mathcal{O}}$ away from each obstacle, where $r_{\mathcal{O}}$ is the radius of the obstacle.

  1. Obstacle $o_1 = (3.0, 1.5, 1.12)$: The robot's center must remain at least $0.5 + 1.12 = 1.62$ units away from the obstacle's center.
     For each grid point (x, y), we compute the distance d between the point and the center of o_1. A point is considered safe if $d > 1.62$ (since obstacles have a closed form expression):

$$d = \sqrt{(x - 3.0)^2 + (y - 1.5)^2}$$

     We will only check those points that the position of the robot in those points won't extend the boundaries. Hence $(x = 1, 2, 3, 4, 5)$ and $(y = 1, 2, 3, 4)$:

$$d_{(1,1)} = \sqrt{(1 - 3.0)^2 + (1 - 1.5)^2} = \sqrt{(-2)^2 + (-0.5)^2} = \sqrt{4 + 0.25} = \sqrt{4.25} = 2.06 > 1.62$$

$$d_{(1,2)} = \sqrt{(1 - 3.0)^2 + (2 - 1.5)^2} = \sqrt{(-2)^2 + (0.5)^2} = \sqrt{4 + 0.25} = \sqrt{4.25} = 2.06 > 1.62$$

$$d_{(1,3)} = \sqrt{(1 - 3.0)^2 + (3 - 1.5)^2} = \sqrt{(-2)^2 + (1.5)^2} = \sqrt{4 + 2.25} = \sqrt{6.25} = 2.50 > 1.62$$

$$d_{(1,4)} = \sqrt{(1 - 3.0)^2 + (4 - 1.5)^2} = \sqrt{(-2)^2 + (2.5)^2} = \sqrt{4 + 6.25} = \sqrt{10.25} = 3.20 > 1.62$$

$$d_{(2,1)} = \sqrt{(2 - 3.0)^2 + (1 - 1.5)^2} = \sqrt{(-1)^2 + (-0.5)^2} = \sqrt{1 + 0.25} = \sqrt{1.25} = 1.12 < 1.62$$

$$d_{(2,2)} = \sqrt{(2 - 3.0)^2 + (2 - 1.5)^2} = \sqrt{(-1)^2 + (0.5)^2} = \sqrt{1 + 0.25} = \sqrt{1.25} = 1.12 < 1.62$$

$$d_{(2,3)} = \sqrt{(2 - 3.0)^2 + (3 - 1.5)^2} = \sqrt{(-1)^2 + (1.5)^2} = \sqrt{1 + 2.25} = \sqrt{3.25} = 1.80 > 1.62$$

$$d_{(2,4)} = \sqrt{(2 - 3.0)^2 + (4 - 1.5)^2} = \sqrt{(-1)^2 + (2.5)^2} = \sqrt{1 + 6.25} = \sqrt{7.25} = 2.69 > 1.62$$

$$d_{(3,1)} = \sqrt{(3 - 3.0)^2 + (1 - 1.5)^2} = \sqrt{(0)^2 + (-0.5)^2} = \sqrt{0 + 0.25} = \sqrt{0.25} = 0.50 < 1.62$$

$$d_{(3,2)} = \sqrt{(3 - 3.0)^2 + (2 - 1.5)^2} = \sqrt{(0)^2 + (0.5)^2} = \sqrt{0 + 0.25} = \sqrt{0.25} = 0.50 < 1.62$$

$$d_{(3,3)} = \sqrt{(3 - 3.0)^2 + (3 - 1.5)^2} = \sqrt{(0)^2 + (1.5)^2} = \sqrt{0 + 2.25} = \sqrt{2.25} = 1.50 < 1.62$$

$$d_{(3,4)} = \sqrt{(3 - 3.0)^2 + (4 - 1.5)^2} = \sqrt{(0)^2 + (2.5)^2} = \sqrt{0 + 6.25} = \sqrt{6.25} = 2.50 > 1.62$$

$$d_{(4,1)} = \sqrt{(4 - 3.0)^2 + (1 - 1.5)^2} = \sqrt{(1)^2 + (-0.5)^2} = \sqrt{1 + 0.25} = \sqrt{1.25} = 1.12 < 1.62$$

$$d_{(4,2)} = \sqrt{(4 - 3.0)^2 + (2 - 1.5)^2} = \sqrt{(1)^2 + (0.5)^2} = \sqrt{1 + 0.25} = \sqrt{1.25} = 1.12 < 1.62$$

$$d_{(4,3)} = \sqrt{(4 - 3.0)^2 + (3 - 1.5)^2} = \sqrt{(1)^2 + (1.5)^2} = \sqrt{1 + 2.25} = \sqrt{3.25} = 1.80 > 1.62$$

$$d_{(4,4)} = \sqrt{(4-3.0)^2 + (4-1.5)^2} = \sqrt{(1)^2 + (2.5)^2} = \sqrt{1+6.25} = \sqrt{7.25} = 2.69 > 1.62$$

$$d_{(5,1)} = \sqrt{(5-3.0)^2 + (1-1.5)^2} = \sqrt{(2)^2 + (-0.5)^2} = \sqrt{4+0.25} = \sqrt{4.25} = 2.06 > 1.62$$

$$d_{(5,2)} = \sqrt{(5-3.0)^2 + (2-1.5)^2} = \sqrt{(2)^2 + (0.5)^2} = \sqrt{4+0.25} = \sqrt{4.25} = 2.06 > 1.62$$

$$d_{(5,3)} = \sqrt{(5-3.0)^2 + (3-1.5)^2} = \sqrt{(2)^2 + (1.5)^2} = \sqrt{4+2.25} = \sqrt{6.25} = 2.50 > 1.62$$

$$d_{(5,4)} = \sqrt{(5-3.0)^2 + (4-1.5)^2} = \sqrt{(2)^2 + (2.5)^2} = \sqrt{4+6.25} = \sqrt{10.25} = 3.20 > 1.62$$

Thus, safe grids are:

$$\{(1,1),(1,2),(1,3),(1,4),(2,3),(2,4),(3,4),(4,4),(5,1),(5,2),(5,3),(5,4)\}$$

2. Obstacle $o_2 = (2.5, 3.0, 0.5)$: The obstacle is centered at (2.5, 3.0) with a radius of 0.5. The robot's center must remain at least $0.5 + 0.5 = 1.0$ units away from the obstacle's center.
   For each grid point (x, y), we compute the distance d between the point and the center of o_2. A point is considered safe if $d > 1.0$ (since obstacles have a closed form expression):
   $$d = \sqrt{(x-2.5)^2 + (y-3.0)^2}$$
   We will only check those points so that the position of the robot in those points won't extend the boundaries or collide with the first obstacle.
   $$d_{(x,y)} = \sqrt{(x-2.5)^2 + (y-3.0)^2}$$

$$d_{(1,1)} = \sqrt{(1-2.5)^2 + (1-3.0)^2} = \sqrt{(-1.5)^2 + (-2.0)^2} = \sqrt{2.25+4} = \sqrt{6.25} = 2.50 > 1.0$$

$$d_{(1,2)} = \sqrt{(1-2.5)^2 + (2-3.0)^2} = \sqrt{(-1.5)^2 + (-1.0)^2} = \sqrt{2.25+1} = \sqrt{3.25} = 1.80 > 1.0$$

$$d_{(1,3)} = \sqrt{(1-2.5)^2 + (3-3.0)^2} = \sqrt{(-1.5)^2 + (0)^2} = \sqrt{2.25+0} = \sqrt{2.25} = 1.50 > 1.0$$

$$d_{(1,4)} = \sqrt{(1-2.5)^2 + (4-3.0)^2} = \sqrt{(-1.5)^2 + (1.0)^2} = \sqrt{2.25+1} = \sqrt{3.25} = 1.80 > 1.0$$

$$d_{(2,3)} = \sqrt{(2-2.5)^2 + (3-3.0)^2} = \sqrt{(-0.5)^2 + (0)^2} = \sqrt{0.25+0} = \sqrt{0.25} = 0.50 < 1.0$$

$$d_{(2,4)} = \sqrt{(2-2.5)^2 + (4-3.0)^2} = \sqrt{(-0.5)^2 + (1.0)^2} = \sqrt{0.25+1} = \sqrt{1.25} = 1.12 > 1.0$$

$$d_{(3,4)} = \sqrt{(3-2.5)^2 + (4-3.0)^2} = \sqrt{(0.5)^2 + (1.0)^2} = \sqrt{0.25+1} = \sqrt{1.25} = 1.12 > 1.0$$

$$d_{(4,4)} = \sqrt{(4-2.5)^2 + (4-3.0)^2} = \sqrt{(1.5)^2 + (1.0)^2} = \sqrt{2.25+1} = \sqrt{3.25} = 1.80 > 1.0$$

$$d_{(5,1)} = \sqrt{(5-2.5)^2 + (1-3.0)^2} = \sqrt{(2.5)^2 + (-2.0)^2} = \sqrt{6.25+4} = \sqrt{10.25} = 3.20 > 1.0$$

$$d_{(5,2)} = \sqrt{(5-2.5)^2 + (2-3.0)^2} = \sqrt{(2.5)^2 + (-1.0)^2} = \sqrt{6.25+1} = \sqrt{7.25} = 2.69 > 1.0$$

$$d_{(5,3)} = \sqrt{(5-2.5)^2 + (3-3.0)^2} = \sqrt{(2.5)^2 + (0)^2} = \sqrt{6.25+0} = \sqrt{6.25} = 2.50 > 1.0$$

$$d_{(5,4)} = \sqrt{(5-2.5)^2 + (4-3.0)^2} = \sqrt{(2.5)^2 + (1.0)^2} = \sqrt{6.25+1} = \sqrt{7.25} = 2.69 > 1.0$$

Thus, the final list of safe grids is:

$$\{(1,1),(1,2),(1,3),(1,4),(2,4),(3,4),(4,3),(4,4),(5,1),(5,2),(5,3),(5,4)\}$$

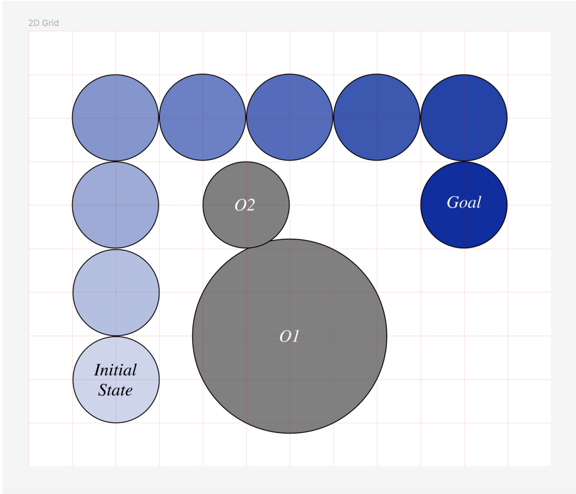

Figure 2: The optimal path based on BFS algorithm

(c) A motion planning algorithm for a circular robot in a 2D grid with obstacles is developed. The algorithm first finds a path using only (x, y) coordinates through Breadth-First Search (BFS), then adds necessary rotations to ensure proper orientation (See figure 2). A StateSpacefor2D class derived from StateSpace was implemented to handle valid states, including collision detection with circular obstacles and boundaries. The ActionSpacefor2D class defines possible rotations and translations, ensuring staying in the action space. The final output provides an ordered robot's path, including only positions.