Planning and Decision Making Assignment 3

Zhuoyang Wang (6376770)

1. Definition of the elements of RRT

Answer 1.1

For the **mobile manipulator**, the workspace consists of the 2D plane W_{base} the base can move in and the 3D space W_{arm} that the arm can reach, i.e., $W = W_{\text{base}} \times W_{\text{arm}} \subset \mathbb{R}^3$; the configuration space is $\mathbb{R}^2 \times S^1 \times S^1 \times S^1 \times S^1$, because of the configuration $q = (x, y, \theta, \alpha, \beta, \gamma)$ and the joints have no limits.

For the **toy crane**, the workspace is also the combination of the base workspace $W_{\text{base}} \subset \mathbb{R}^2$ and the crane workspace $W_{\text{crane}} \subset \mathbb{R}^3$, i.e., $W = W_{\text{base}} \times W_{\text{crane}} \subset \mathbb{R}^3$; considering the limits, the configuration space here is $\mathbb{R}^2 \times S^1 \times \left[0, \frac{1}{2}\pi\right] \times [b_{\text{min}}, b_{\text{max}}] \times [c_{\text{min}}, c_{\text{max}}]$.

Answer 1.2

The **mobile manipulator** is nonholonomic because the base is considered a unicycle, which is nonholonomic due to its inability to move sideways.

The **toy crane** is also nonholonomic because he base is considered a car, which is nonholonomic due to its inability to move sideways.

Answer 1.3

For the **mobile manipulator**, denote the configuration as $q = (x, y, \theta, \alpha, \beta, \gamma)$, we can define the distance metric:

$$d(q_1, q_2) = \sqrt{\begin{gathered} k_x(x_1 - x_2)^2 + k_y(y_1 - y_2)^2 + k_\theta(\theta_1 - \theta_2)^2_{[-\pi,\pi]} + k_\alpha(\alpha_1 - \alpha_2)^2_{[-\pi,\pi]} + \\ k_\beta(\beta_1 - \beta_2)^2_{[-\pi,\pi]} + k_\gamma(\gamma_1 - \gamma_2)^2_{[-\pi,\pi]} \end{gathered}} \tag{1}$$

where $(\cdot - \cdot)_{[-\pi,\pi]}$ means wrapping the difference to $[-\pi, \pi]$, $k_{\dots}$ are weights.

For the **toy crane**, denote the configuration as $q = (x, y, \theta, \alpha, b, c)$, we can define the distance metric:

$$d(q_1, q_2) = \sqrt{\begin{gathered} k_x(x_1 - x_2)^2 + k_y(y_1 - y_2)^2 + k_\theta(\theta_1 - \theta_2)^2_{[-\pi,\pi]} + k_\alpha(\alpha_1 - \alpha_2)^2 + \\ k_b(b_1 - b_2)^2 + k_c(c_1 - c_2)^2 \end{gathered}} \tag{2}$$

Answer 1.4

First consider the **mobile manipulator**. For the base (unicycles can turn in place), steer until directly facing the target position, go straight and then turn to the final angle; for the arm, use linear interpolation. Specifically, for the base, $\theta(t) = \theta_1 + \omega_1 t$ until $\theta(t) = \operatorname{atan2}(y_2 - y_1, x_2 - x_1) =: \theta_{\text{target}}$; $\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} x_1 + vt\cos\theta_{\text{target}} \\ y_1 + vt\sin\theta_{\text{target}} \end{pmatrix}$ until reach $\begin{pmatrix} x_2 \\ y_2 \end{pmatrix}$; and finally $\theta(t) = \theta_{\text{target}}\omega_2 t$ until $\theta(t) = \theta_2$. And for the arm, $\begin{pmatrix} \alpha(t) \\ \beta(t) \\ \gamma(t) \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \beta_1 \\ \gamma_1 \end{pmatrix} + \left[\begin{pmatrix} \alpha_2 \\ \beta_2 \\ \gamma_2 \end{pmatrix} - \begin{pmatrix} \alpha_1 \\ \beta_1 \\ \gamma_1 \end{pmatrix}\right]\frac{t}{T}$.

Then consider the **toy crane**. The car base cannot turn in place, so we might solve an optimization problem like

$$\begin{gathered} \min_{v(t),\delta(t)} T \\ \text{where } \dot{x} = v\cos\theta, \dot{y} = v\sin\theta, \dot{\theta} = \frac{v}{L}\tan\delta \\ \text{and } \begin{pmatrix} x(0) \\ y(0) \\ \theta(0) \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \\ \theta_1 \end{pmatrix}, \begin{pmatrix} x(T) \\ y(T) \\ \theta(T) \end{pmatrix} = \begin{pmatrix} x_2 \\ y_2 \\ \theta_2 \end{pmatrix} \end{gathered} \tag{3}$$

to get the base path or use **Reeds-Shepp paths**. Then for the arm still use linear interpolation, like $\begin{pmatrix} \alpha(t) \\ b(t) \\ c(t) \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ b_1 \\ c_1 \end{pmatrix} + \left[\begin{pmatrix} \alpha_2 \\ b_2 \\ c_2 \end{pmatrix} - \begin{pmatrix} \alpha_1 \\ b_1 \\ c_1 \end{pmatrix} \right] \frac{t}{T}$.

2. RRT

Answer 2.1

Step 1. Initialization

Initialize the tree with only the start point.

Step 2. Sampling a random configuration

Randomly sample a configuration q_{rand} in the configuration space. Usually we can set a higher probability to sample near the goal.

Step 3. Finding the nearest neighbor (Use the distance metric from Question 1.3)

Find the node in the tree which is the closest one to q_{rand} as q_{near}.

Step 4. Exact Steering (Use the steering function from Question 1.4)

Derive q_{new} referenced to q_{near} to q_{rand}, and a path $q(t)$ from q_{near} to q_{new} using the steering function.

Step 5. Collision checking

Check if the path is collision-free. If it is then accept the new node into the tree and record the path (new edge).

Step 6. Goal checking

Check if the goal has been reached, or repeat step 2 to 6.

Answer 2.2

RRT is **not optimal** because it only finds the feasible solution but not the shortest one, the tree grows randomly and does not "relax" the early paths.

A globally optimal alternative is RRT* algorithm, which introduces the rewiring step and will

Answer 2.3

RRT is **probabilistic complete** but not complete.

Probabilistic completeness means if a solution exists, the probability that RRT finds it approaches 1 as the number of samples go to infinity; completeness indicates that it guarantee a solution in finite time, which is not satisfied by RRT (if enough unfortunate you will never get the solution).