RRT MP

ECE/CS498
Pulkit Katdare
8th April, 2019
Outline

• Introduction

• Rapidly-Exploring Random Tree

• Controller Design

• Putting it all together
What is Planning?
Motion Planning

Given initial state $x_{init}$ and a goal $X_G$, what is the path or sequence of control inputs that will lead us from start to goal?

Possible Issues:

• Obstacle avoidance

• Nonholonomic systems

• Computationally intensive
  • Nonconvex optimization
  • Large number of samples required in real-time

Two approaches: optimization-based & sampling-based techniques
Rapidly exploring Random Tree (RRT)

Build tree by generating next states through the dynamics by randomly selecting inputs
Rapidly exploring Random Tree (RRT)

Build tree by generating next states through the dynamics by randomly selecting inputs

Generate_RRT($x_{init}, K, \Delta t$)

$\mathcal{T}$.init($x_{init}$)

for $k = 1$ to $K$

\[
\begin{align*}
x_{\text{rand}} & \leftarrow \text{RANDOM\_STATE}() \\
x_{\text{near}} & \leftarrow \text{NEAREST\_NEIGHBOR}(x_{\text{rand}}, \mathcal{T}) \\
u & \leftarrow \text{SELECT\_INPUT}(x_{\text{rand}}, x_{\text{near}}) \\
x_{\text{new}} & \leftarrow \text{NEW\_STATE}(x_{\text{near}}, u, \Delta t) \\
\mathcal{T}.\text{add\_vertex}(x_{\text{new}}) \\
\mathcal{T}.\text{add\_edge}(x_{\text{near}}, x_{\text{new}}, u)
\end{align*}
\]

Return $\mathcal{T}$
Rapidly exploring Random Tree (RRT)

Build tree by generating next states through the dynamics by randomly selecting inputs

\[
\text{Generate\_RRT}(x_{\text{init}}, K, \Delta t)
\]

\[
\mathcal{T}.\text{init}(x_{\text{init}})
\]

for \( k = 1 \) to \( K \)

\[
x_{\text{rand}} \leftarrow \text{RANDOM\_STATE}()
\]

\[
x_{\text{near}} \leftarrow \text{NEAREST\_NEIGHBOR}(x_{\text{rand}}, \mathcal{T})
\]

\[
u \leftarrow \text{SELECT\_INPUT}(x_{\text{rand}}, x_{\text{near}})
\]

\[
x_{\text{new}} \leftarrow \text{NEW\_STATE}(x_{\text{near}}, u, \Delta t)
\]

\[
\mathcal{T}.\text{add\_vertex}(x_{\text{new}})
\]

\[
\mathcal{T}.\text{add\_edge}(x_{\text{near}}, x_{\text{new}}, u)
\]

Return \( \mathcal{T} \)
Controller Design?

• How to move a robot from point A to point B
• We need a model first

\[
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{\theta}
\end{bmatrix} =
\begin{bmatrix}
V_R \cos(\theta) \\
V_R \sin(\theta) \\
\delta
\end{bmatrix}
\]
How to figure out this $V_R$ and $\delta$?

\[
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{\theta}
\end{bmatrix} = \begin{bmatrix}
V_R \cos(\theta) \\
V_R \sin(\theta) \\
\delta
\end{bmatrix}
\]

- $V_R$ : speed of the car
- $\delta$ : steering of the car
Controller algorithm

- $V_R = 0$
- $\delta = K(\arctan(\overrightarrow{AB}) - \theta)$
Controller algorithm

- $V_R =$ Constant
- $\delta = 0.0$
Controller algorithm

- $V_R = \text{Constant}$
- $\delta = 0.0$
Controller algorithm

- $V_R =$ Constant
- $\delta = 0.0$
This MP