

# ME 599/699 Robot Modeling & Control

## Fall 2021

### **Motion Planning Overview**

Hasan A. Poonawala

Department of Mechanical Engineering  
University of Kentucky

Email: [hasan.poonawala@uky.edu](mailto:hasan.poonawala@uky.edu)

Web: <https://www.engr.uky.edu/~hap>

# Motion Planning

**Plan:** "A detailed proposal for doing or achieving something."

# Motion Planning

**Plan:** "A detailed proposal for doing or achieving something."

**Motion Plan:** In robotics, a motion plan is a sequence of states and/or controller actions that achieves a task.

# Motion Planning

**Plan:** "A detailed proposal for doing or achieving something."

**Motion Plan:** In robotics, a motion plan is a sequence of states and/or controller actions that achieves a task.

Two types:

- ▶ Path Plan
- ▶ Trajectory Plan

# Paths and Trajectories

## Definition (Path)

A path is a sequence of configurations.

# Paths and Trajectories

## Definition (Path)

A path is a sequence of configurations.

## Definition (Continuous Path)

A path in  $\mathbb{R}^n$  is a continuous function  $\gamma$  from the unit interval  $I = [0, 1]$  to  $\mathbb{R}^n$ .

# Paths and Trajectories

## Definition (Path)

A path is a sequence of configurations.

## Definition (Continuous Path)

A path in  $\mathbb{R}^n$  is a continuous function  $\gamma$  from the unit interval  $I = [0, 1]$  to  $\mathbb{R}^n$ .

## Definition (Trajectory)

A trajectory  $q(t)$  in  $\mathbb{R}^n$  is a continuous function  $q$  from the an interval of time  $[t_0, t_f]$  to  $\mathbb{R}^n$ .

# Paths and Trajectories

## Definition (Path)

A path is a sequence of configurations.

## Definition (Continuous Path)

A path in  $\mathbb{R}^n$  is a continuous function  $\gamma$  from the unit interval  $I = [0, 1]$  to  $\mathbb{R}^n$ .

## Definition (Trajectory)

A trajectory  $q(t)$  in  $\mathbb{R}^n$  is a continuous function  $q$  from the an interval of time  $[t_0, t_f]$  to  $\mathbb{R}^n$ .

Some robots will execute motion described either as a path (waypoints) or a trajectory.



# Motion Planning

- ▶ Optimization over parametrized paths/trajectories

# Motion Planning

- ▶ Optimization over parametrized paths/trajectories
  - ▶ Splines / Bezier curves

# Motion Planning

- ▶ Optimization over parametrized paths/trajectories
  - ▶ Splines / Bezier curves
  - ▶ Polynomial trajectories

# Motion Planning

- ▶ Optimization over parametrized paths/trajectories
  - ▶ Splines / Bezier curves
  - ▶ Polynomial trajectories
- ▶ CS Search-based algorithms

# Motion Planning

- ▶ Optimization over parametrized paths/trajectories
  - ▶ Splines / Bezier curves
  - ▶ Polynomial trajectories
- ▶ CS Search-based algorithms
  - ▶ We may be given a graph to plan over,

# Motion Planning

- ▶ Optimization over parametrized paths/trajectories
  - ▶ Splines / Bezier curves
  - ▶ Polynomial trajectories
- ▶ CS Search-based algorithms
  - ▶ We may be given a graph to plan over,
  - ▶ or we build one using sampling (PRM, RRT)

# Motion Planning

- ▶ Optimization over parametrized paths/trajectories
  - ▶ Splines / Bezier curves
  - ▶ Polynomial trajectories
- ▶ CS Search-based algorithms
  - ▶ We may be given a graph to plan over,
  - ▶ or we build one using sampling (PRM, RRT)
- ▶ Optimal Control

# Motion Planning

- ▶ Optimization over parametrized paths/trajectories
  - ▶ Splines / Bezier curves
  - ▶ Polynomial trajectories
- ▶ CS Search-based algorithms
  - ▶ We may be given a graph to plan over,
  - ▶ or we build one using sampling (PRM, RRT)
- ▶ Optimal Control
  - ▶ LQR, iLQR, LQR Trees (Linear Quadratic Regulator)



# Motion Planning

- ▶ Optimization over parametrized paths/trajectories
  - ▶ Splines / Bezier curves
  - ▶ Polynomial trajectories
- ▶ CS Search-based algorithms
  - ▶ We may be given a graph to plan over,
  - ▶ or we build one using sampling (PRM, RRT)
- ▶ Optimal Control
  - ▶ LQR, iLQR, LQR Trees (Linear Quadratic Regulator)
  - ▶ Model Predictive Control (MPC)

# Motion Planning

- ▶ Optimization over parametrized paths/trajectories
  - ▶ Splines / Bezier curves
  - ▶ Polynomial trajectories
- ▶ CS Search-based algorithms
  - ▶ We may be given a graph to plan over,
  - ▶ or we build one using sampling (PRM, RRT)
- ▶ Optimal Control
  - ▶ LQR, iLQR, LQR Trees (Linear Quadratic Regulator)
  - ▶ Model Predictive Control (MPC)
  - ▶ Markov decision processes (MDPs), POMDPs

# Motion Planning

- ▶ Optimization over parametrized paths/trajectories
  - ▶ Splines / Bezier curves
  - ▶ Polynomial trajectories
- ▶ CS Search-based algorithms
  - ▶ We may be given a graph to plan over,
  - ▶ or we build one using sampling (PRM, RRT)
- ▶ Optimal Control
  - ▶ LQR, iLQR, LQR Trees (Linear Quadratic Regulator)
  - ▶ Model Predictive Control (MPC)
  - ▶ Markov decision processes (MDPs), POMDPs
  - ▶ Reinforcement Learning

# Motion Planning

- ▶ Optimization over parametrized paths/trajectories
  - ▶ Splines / Bezier curves
  - ▶ Polynomial trajectories
- ▶ CS Search-based algorithms
  - ▶ We may be given a graph to plan over,
  - ▶ or we build one using sampling (PRM, RRT)
- ▶ Optimal Control
  - ▶ LQR, iLQR, LQR Trees (Linear Quadratic Regulator)
  - ▶ Model Predictive Control (MPC)
  - ▶ Markov decision processes (MDPs), POMDPs
  - ▶ Reinforcement Learning
  - ▶ Generates the trajectory *and* the control

# Graphs

A Graph  $G = (V, E)$  Consists of two lists (sets):

- Vertices  $V$

On the right,  $V = \{1, 2, 3, 4\}$

- Edges  $E \subseteq V \times V$

On the right,  $E =$

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

