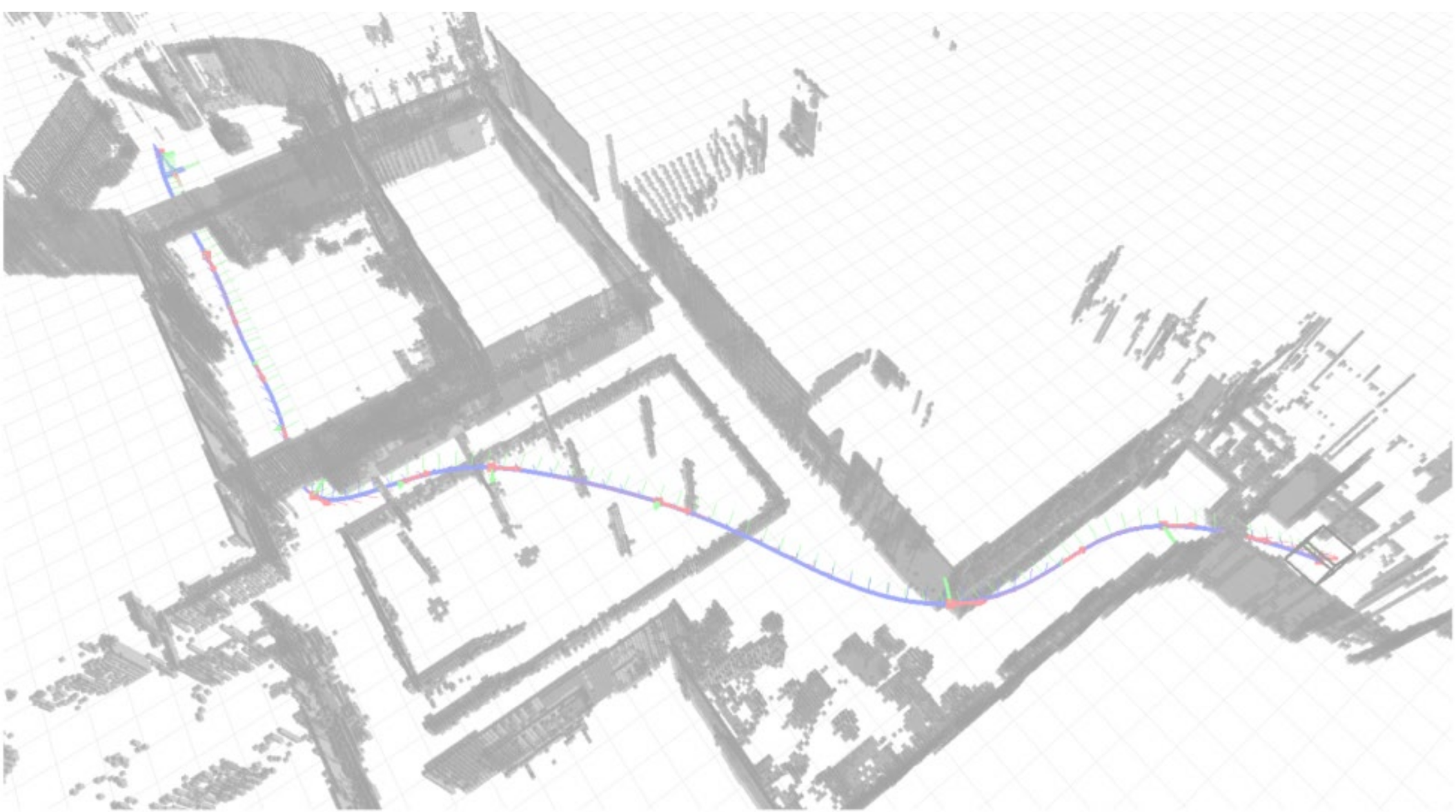
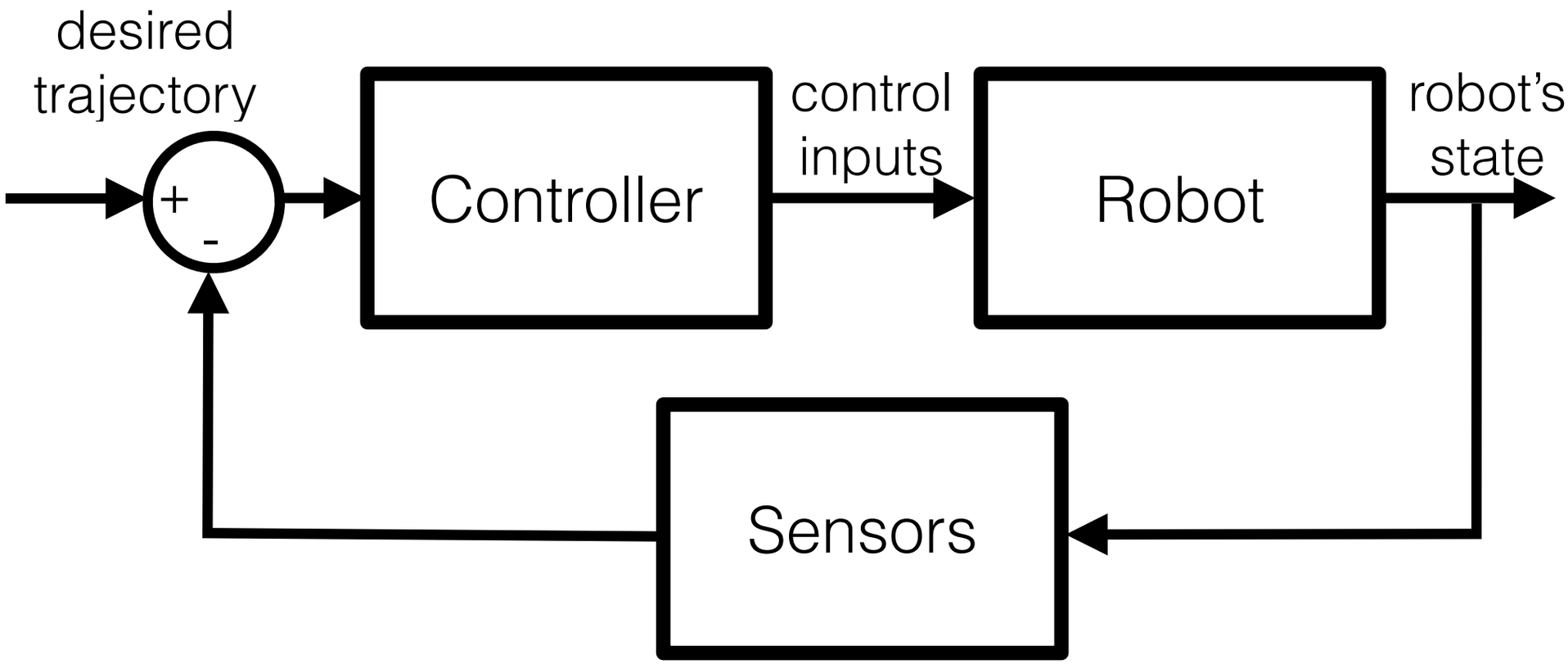


16.485: VNAV - Visual Navigation for Autonomous Vehicles

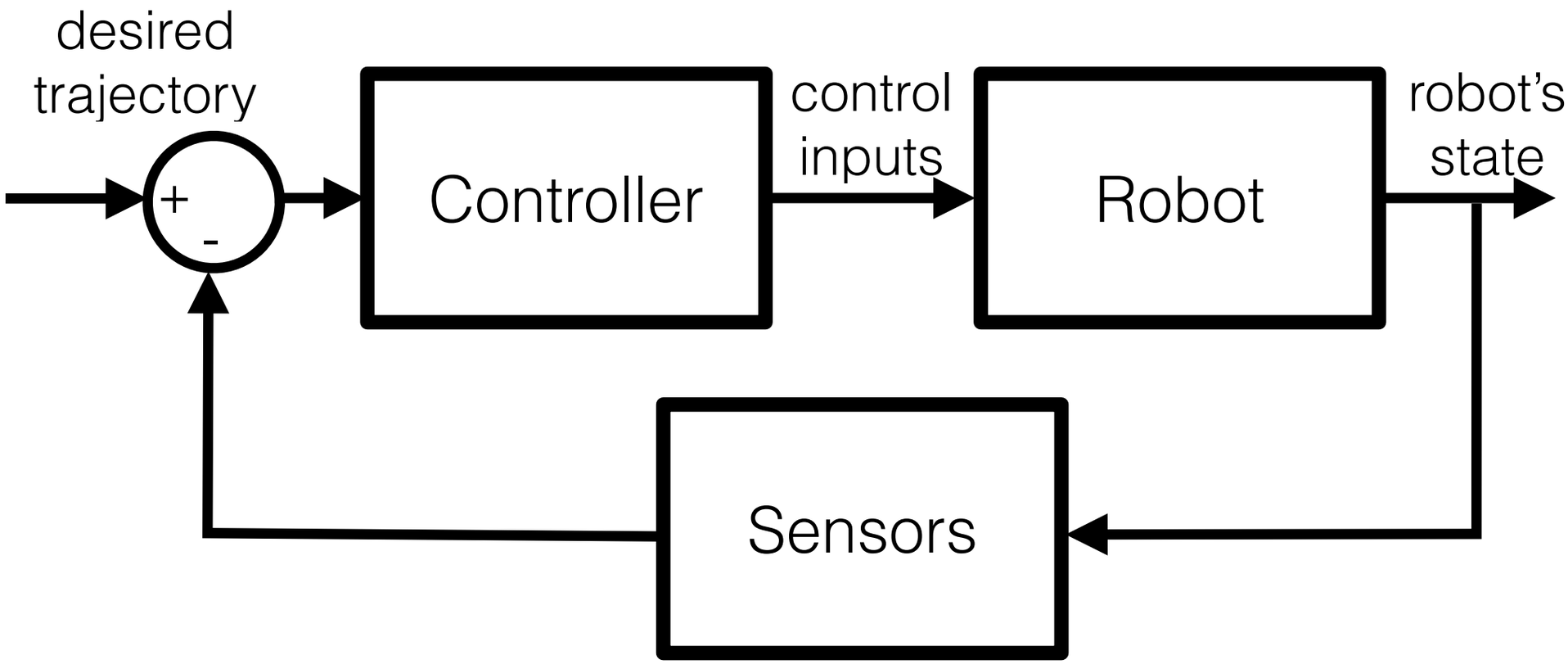
Lecture 8: Trajectory Optimization

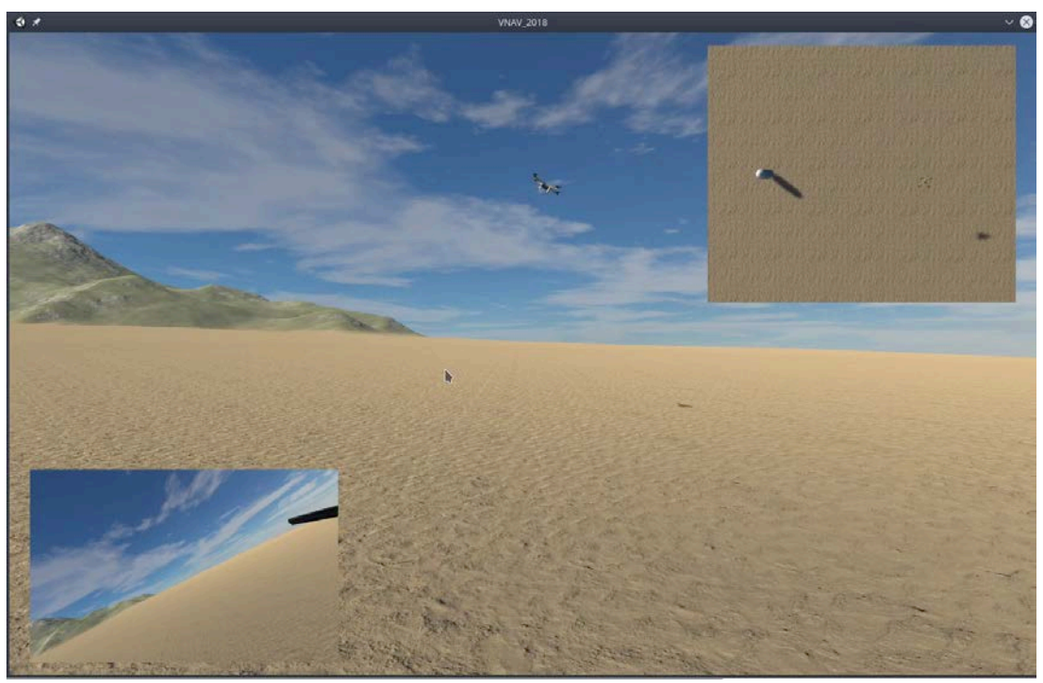
Luca Carlone

Planning vs. Control

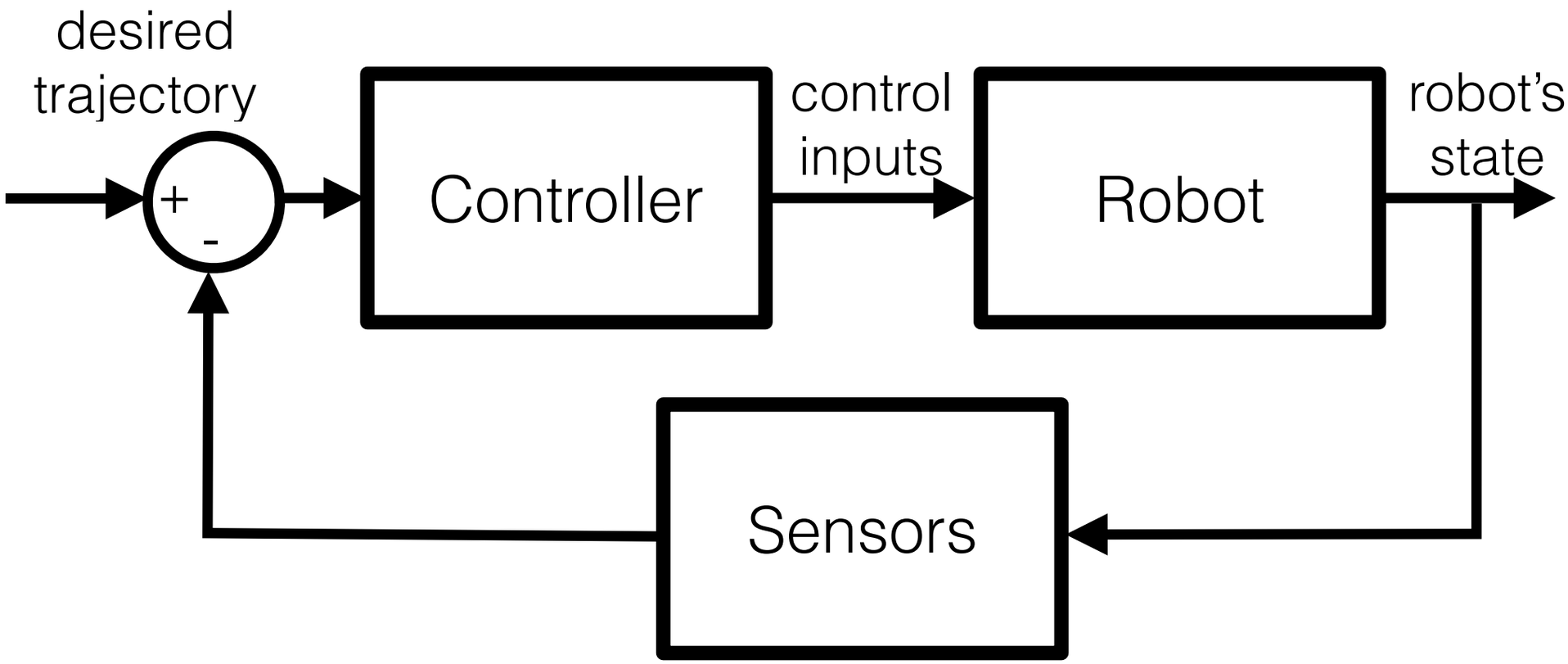


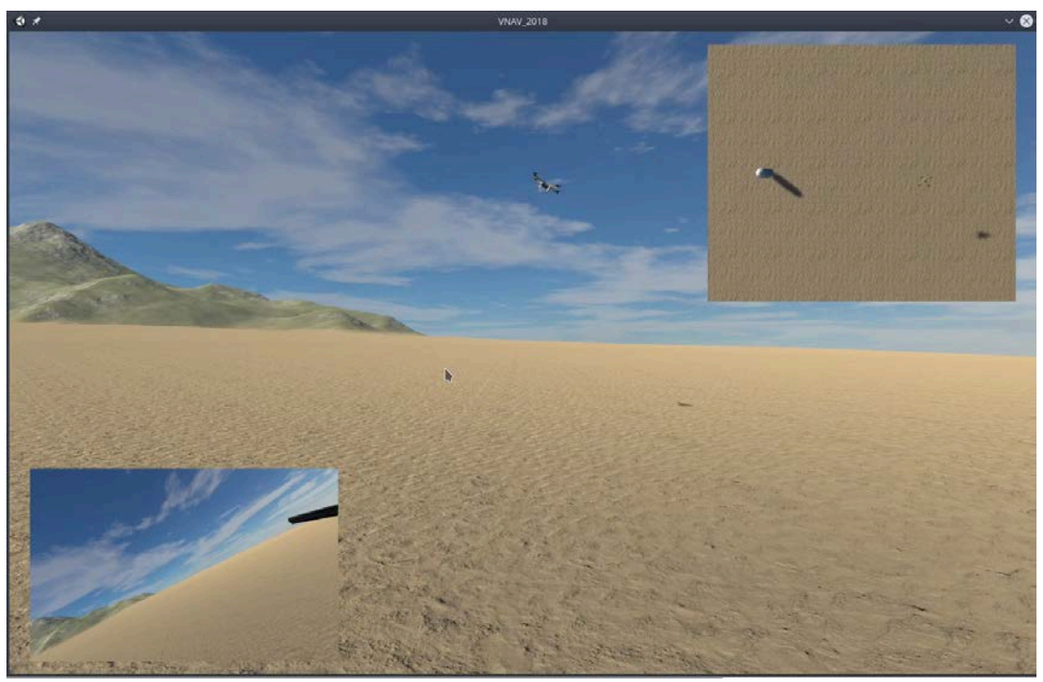
Planning vs. Control

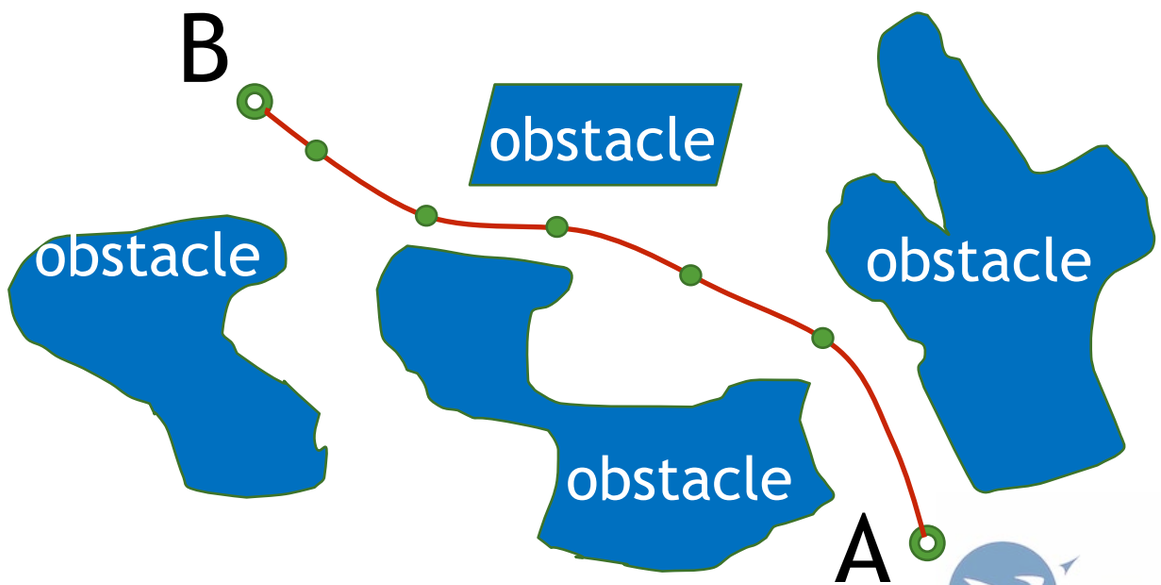




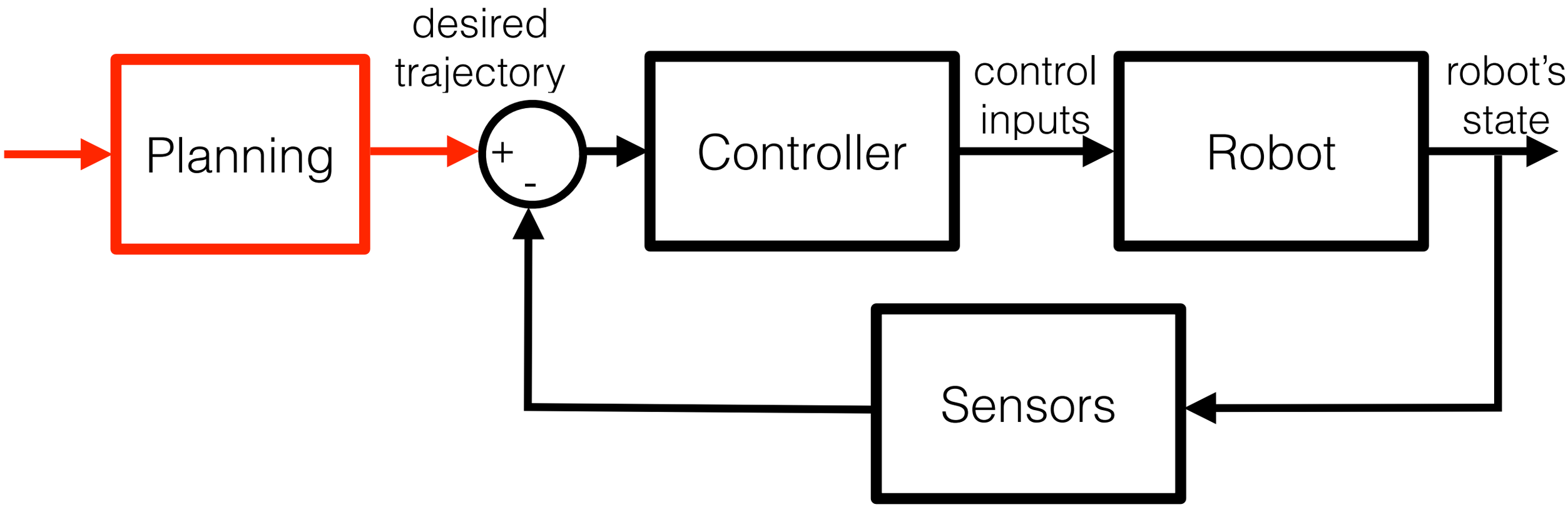
Planning vs. Control

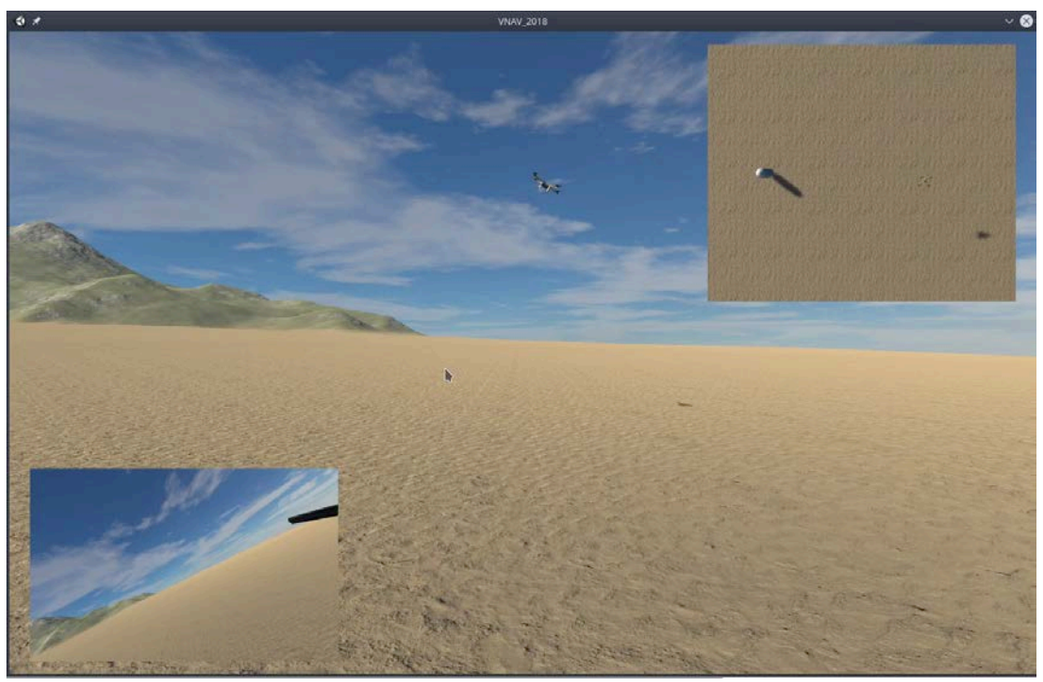


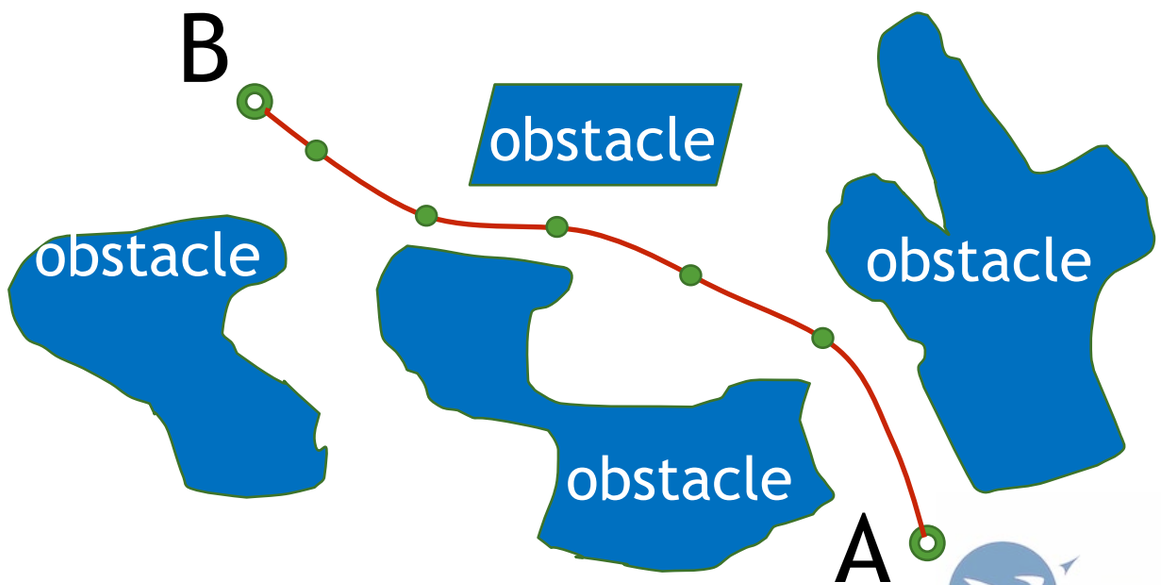




Planning vs. Control



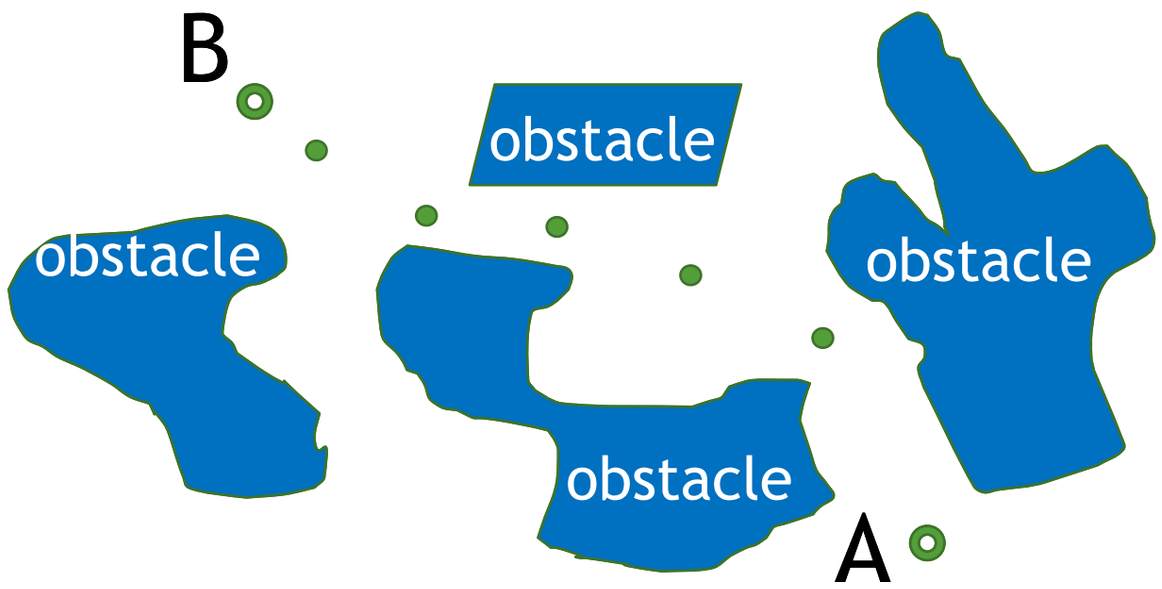




Path Planning vs. Trajectory Planning

- **Path** is a sequence of waypoints (in the obstacle-free space), without *time labels* or *information about velocity or higher order of derivatives*.

DOES NOT ACCOUT FOR DYNAMICS



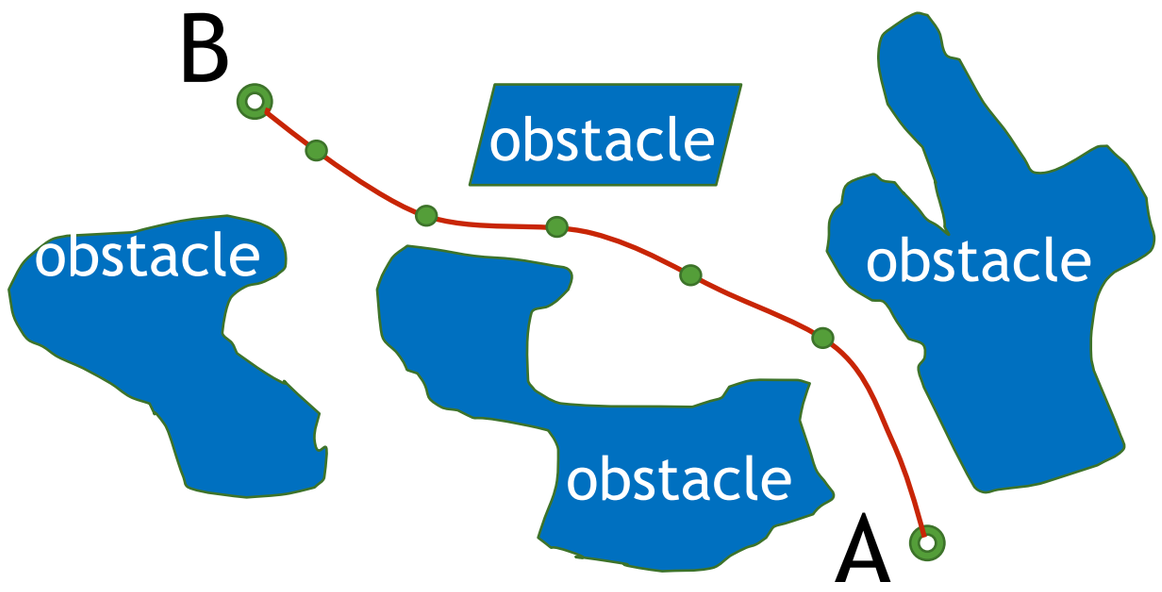
Path Planning vs. Trajectory Planning

- **Path** is a sequence of waypoints (in the obstacle-free space), without *time labels* or *information about velocity or higher order of derivatives*.

DOES NOT ACCOUT FOR DYNAMICS

- **Trajectory** is the sequence of movements the robot should make.

ACCOUNTS FOR DYNAMICS



Path Planning vs. Trajectory Planning

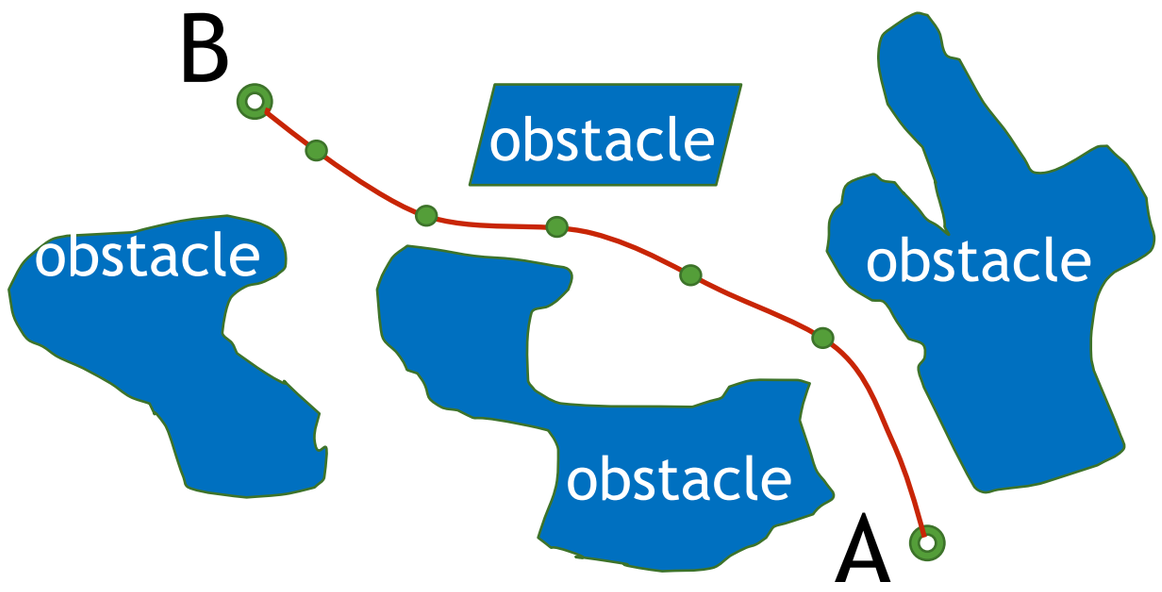
- **Path** is a sequence of waypoints (in the obstacle-free space), without *time labels* or *information about velocity or higher order of derivatives*.

DOES NOT ACCOUT FOR DYNAMICS

- **Trajectory** is the sequence of movements the robot should make.

ACCOUNTS FOR DYNAMICS

MUST BE COLLISION FREE



Path Planning vs. Trajectory Planning

- **Path** is a sequence of waypoints (in the obstacle-free space), without *time labels* or *information about velocity or higher order of derivatives*.

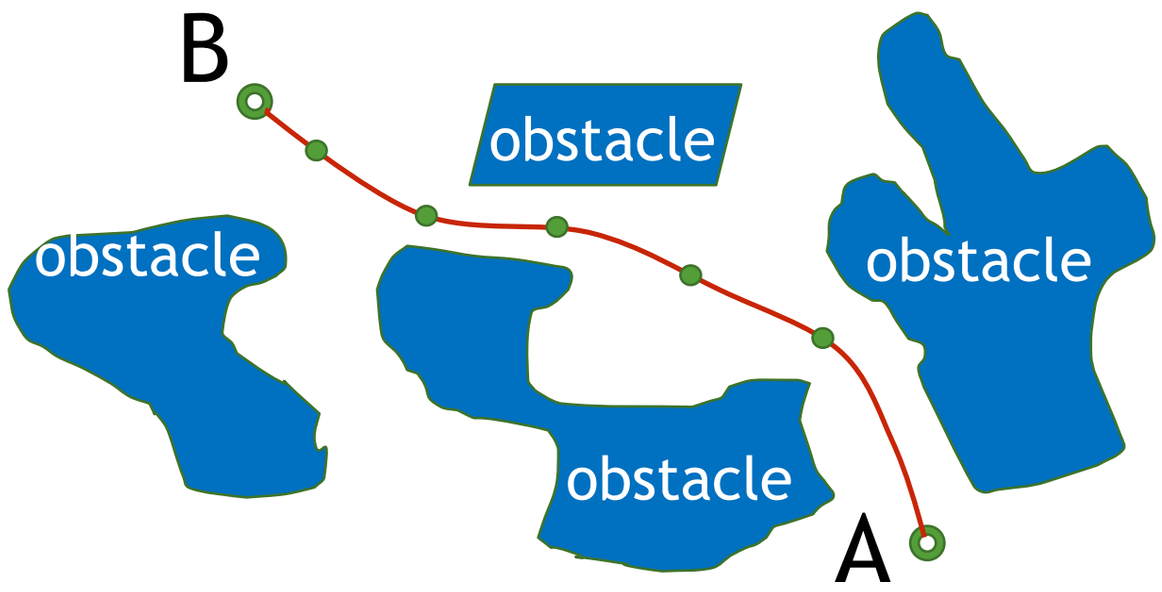
DOES NOT ACCOUT FOR DYNAMICS*

***Can account for dynamics but can be slow** (Bry et al., IJRR '15)

- **Trajectory** is the sequence of movements the robot should make.

ACCOUNTS FOR DYNAMICS

MUST BE COLLISION FREE



Planning algorithms

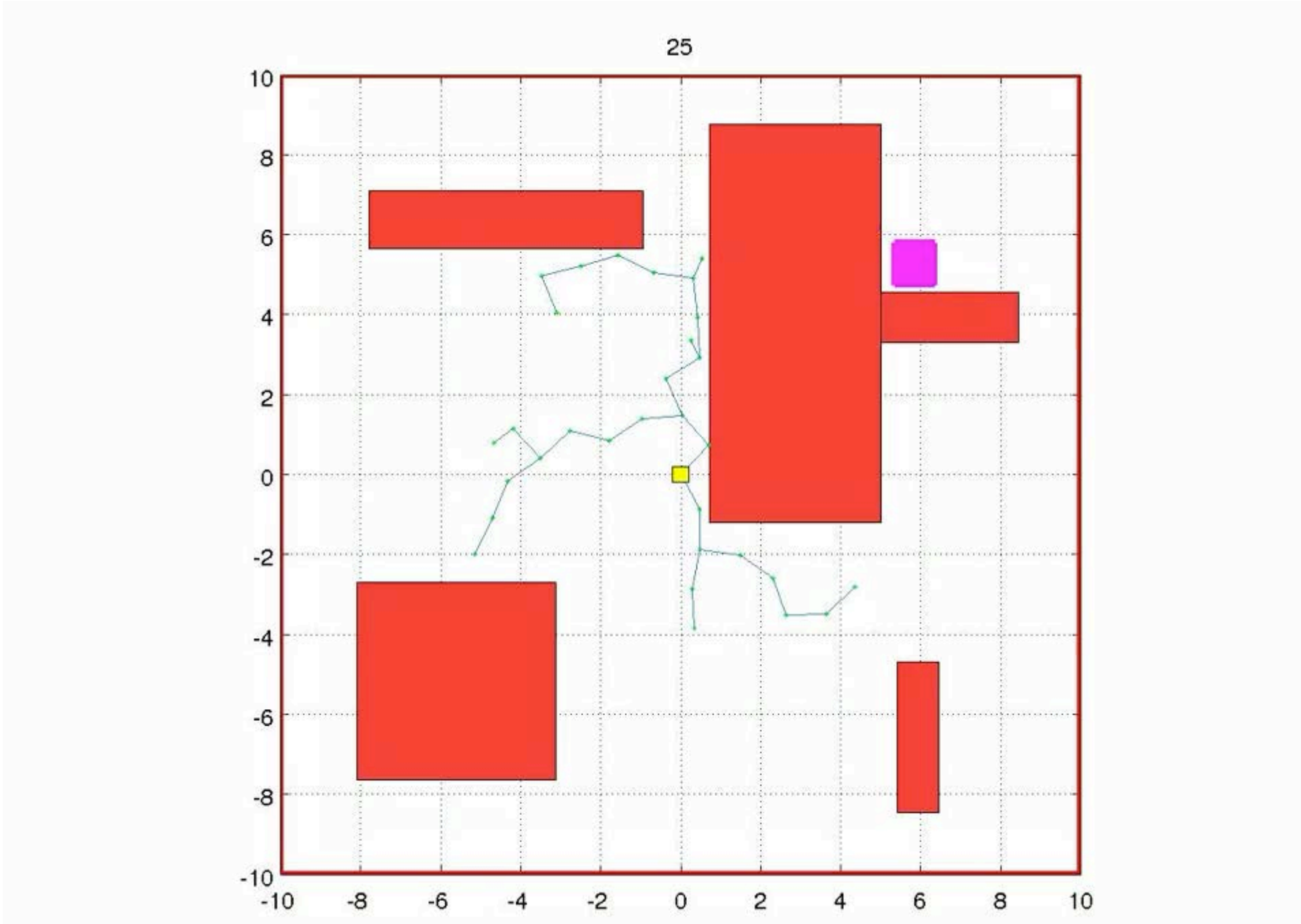
- Open-source libraries
  - Open Motion Planning Library (OMPL)
    - http://ompl.kavrakilab.org/
  - Motion Strategy Library (MSL)
    - http://msl.cs.uiuc.edu/msl/
  - RRT* Library
  - Sampling Based Planning Library

References

- Howie Choset et al., “Principles of Robot Motion,” MIT press, 2005.
- Steven Lavalle, “Planning Algorithms,” Cambridge University Press, 2006.

Path Planning Example

RRT*: Rapidly exploring Random Trees



Path Planning Example

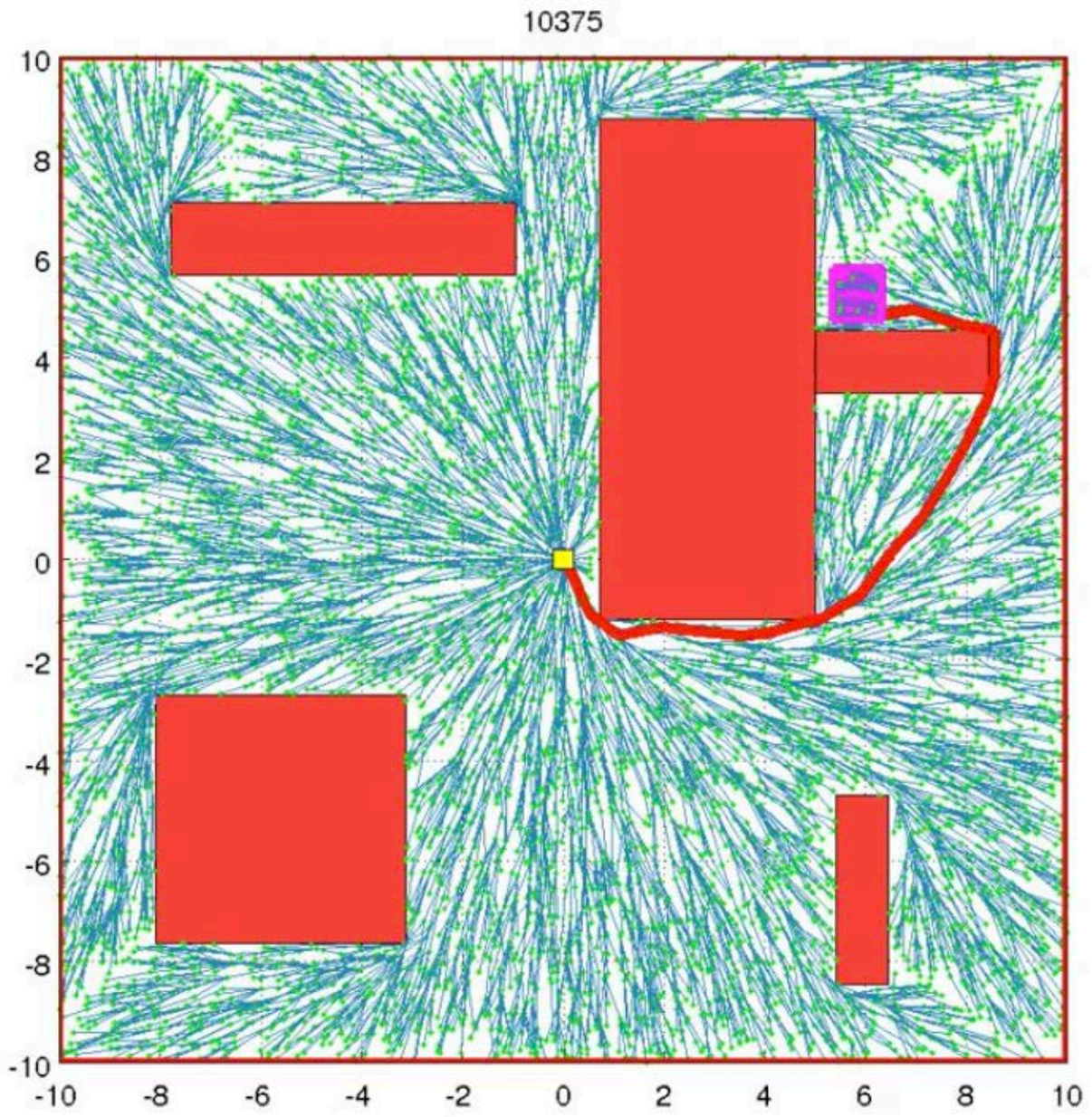
RRT*: Rapidly exploring Random Trees

Pros

- Finds optimal path (if one exists)

Cons

- Impractical running time *if asked for path with smooth trajectory* (Bry et al., IJRR '15)
- Jagged path *otherwise*



Path Planning Example

RRT*: Rapidly exploring Random Trees

Pros

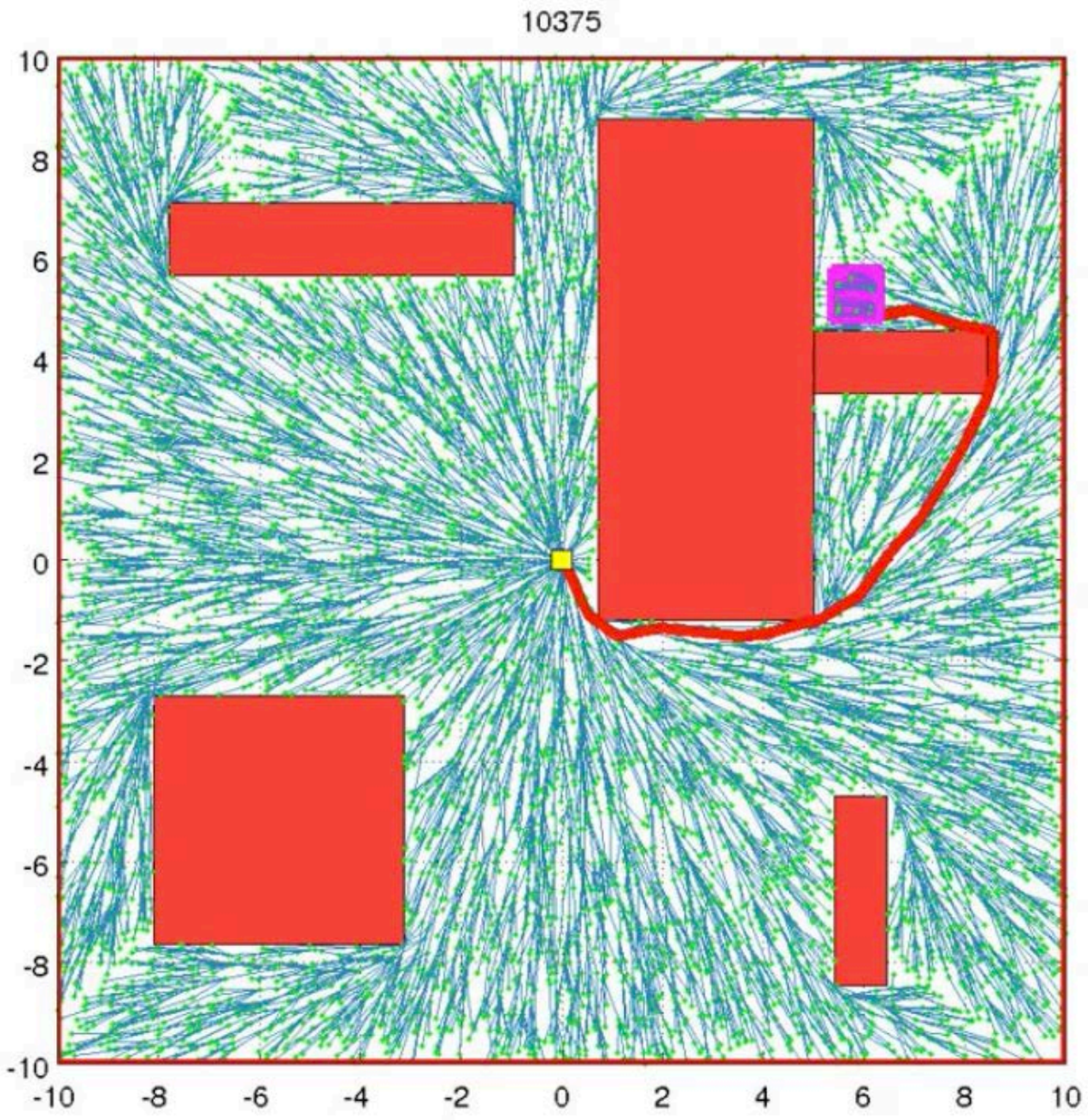
- Finds optimal path (if one exists)

Cons

- Impractical running time *if asked for path with smooth trajectory* (Bry et al., IJRR '15)

Difficult to apply for online planning in unknown/dynamic environments

- Jagged path *otherwise*



Path Planning Example

RRT*: Rapidly exploring Random Trees

Pros

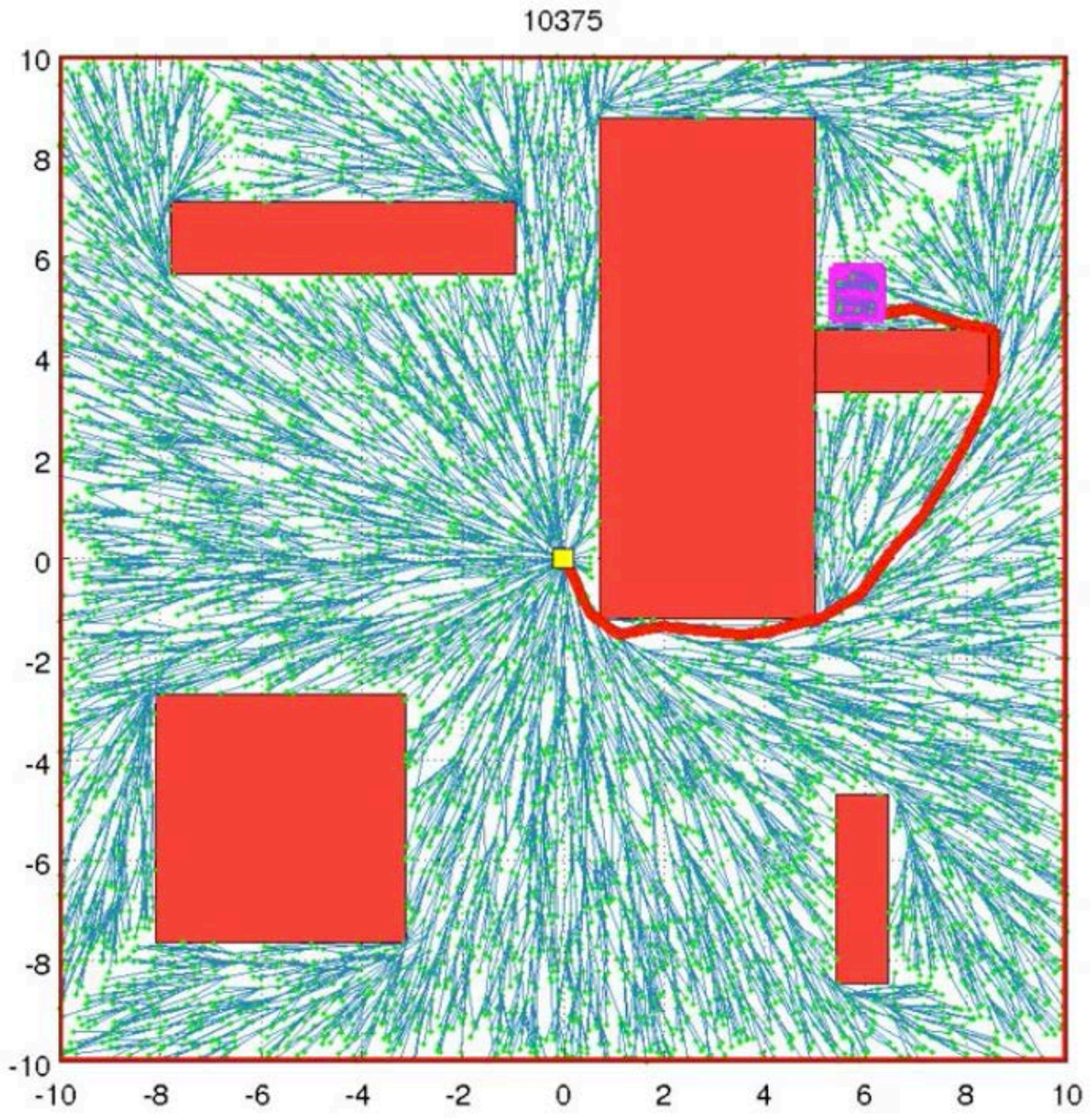
- Finds optimal path (if one exists)

Cons

- Impractical running time *if asked for path with smooth trajectory** (Bry et al., IJRR '15)

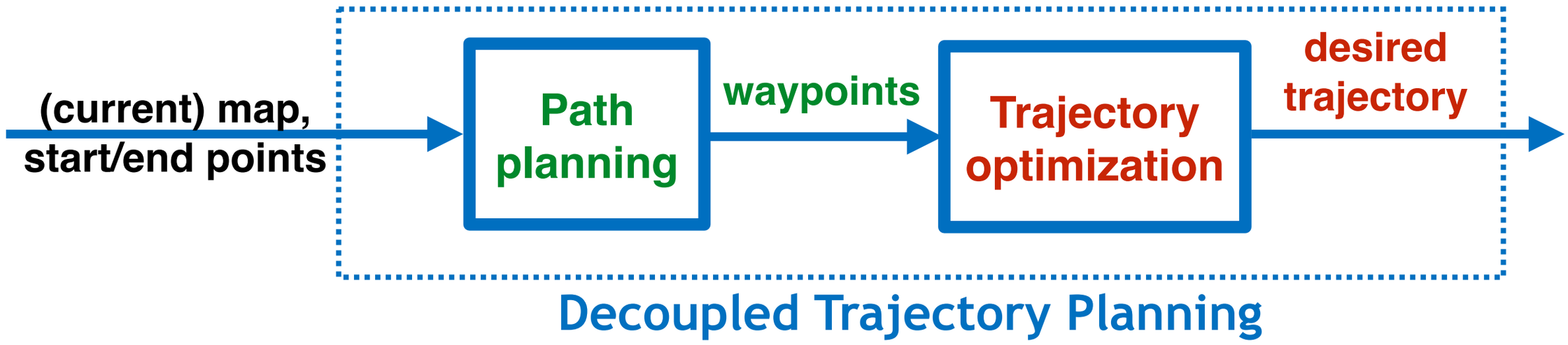
the simultaneous path+trajectory planning is called **direct trajectory planning*

- Jagged path *otherwise*

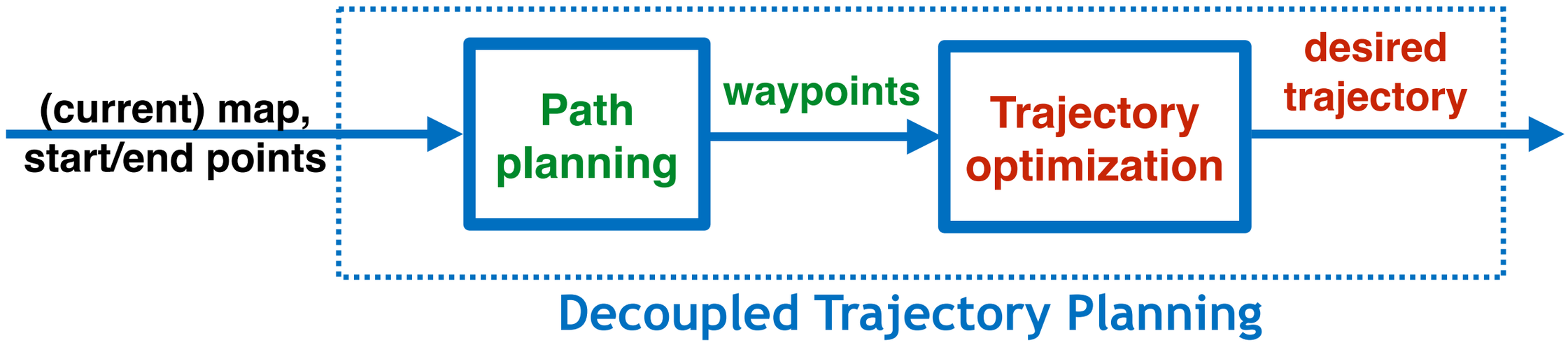


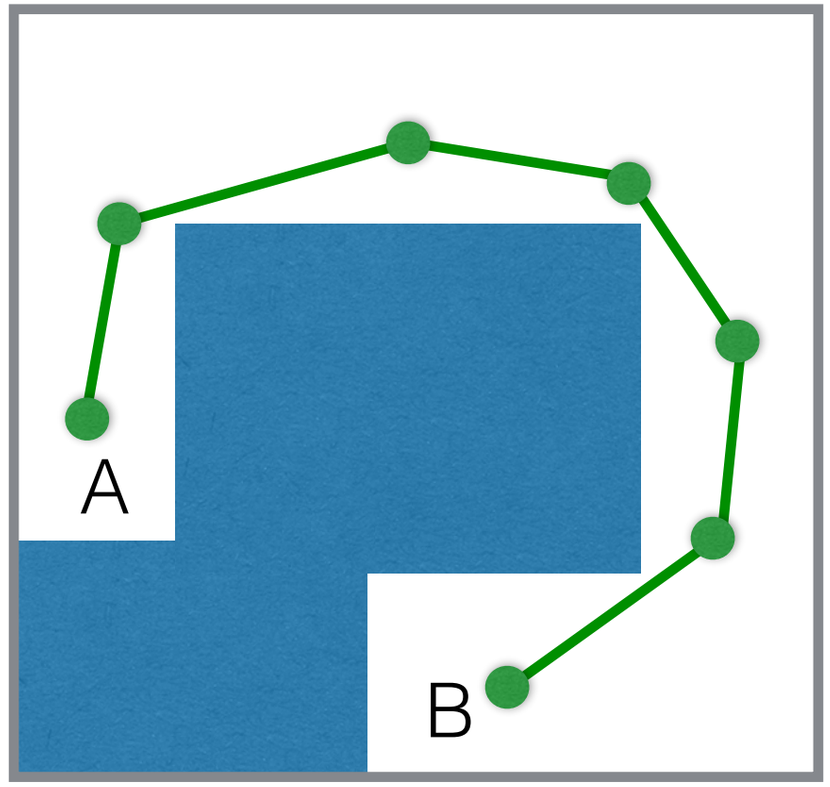
[courtesy of Prof. Sertac Karaman]

Decoupled Trajectory Planning

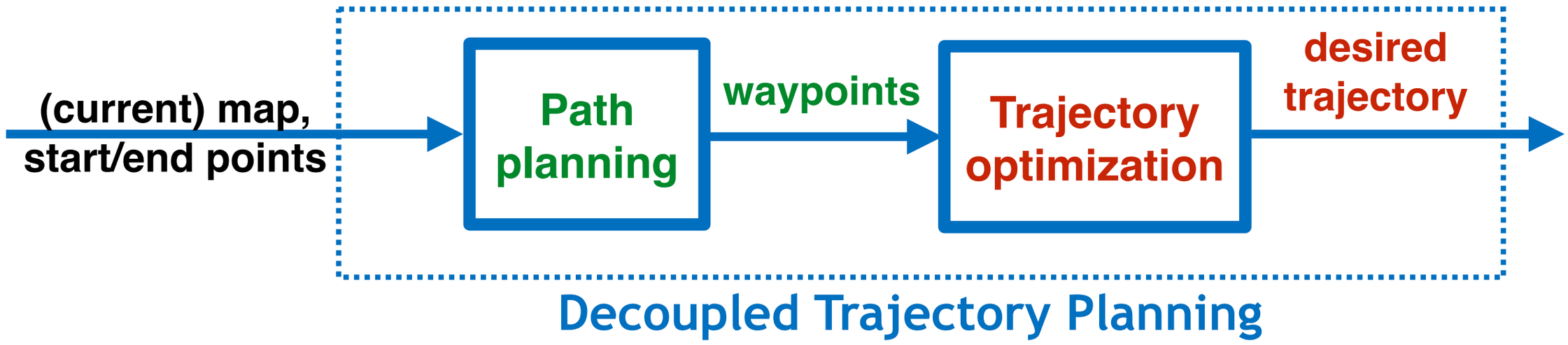


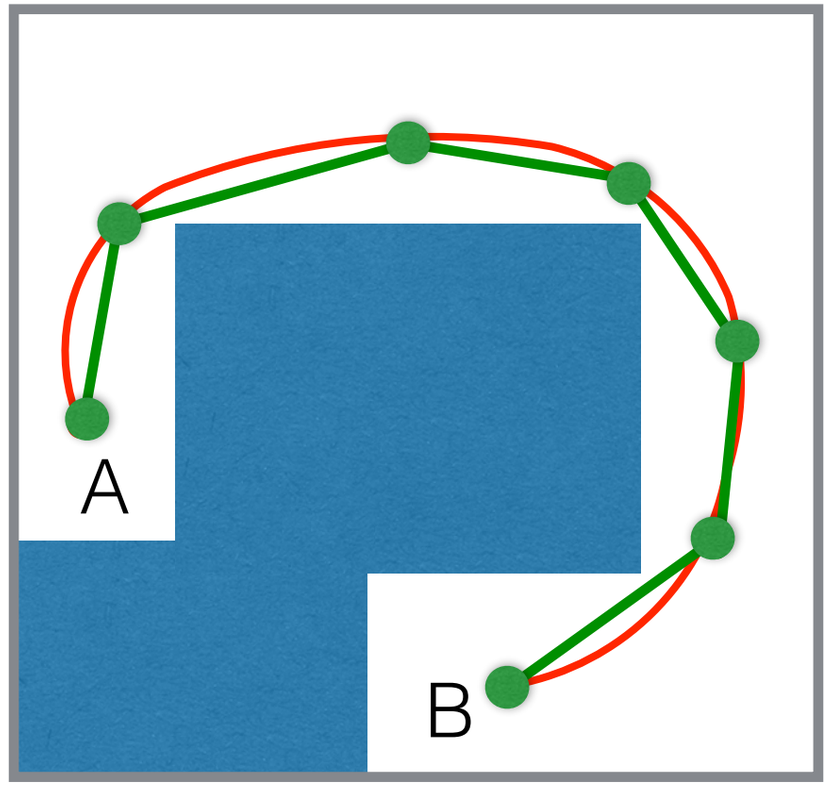
Decoupled Trajectory Planning



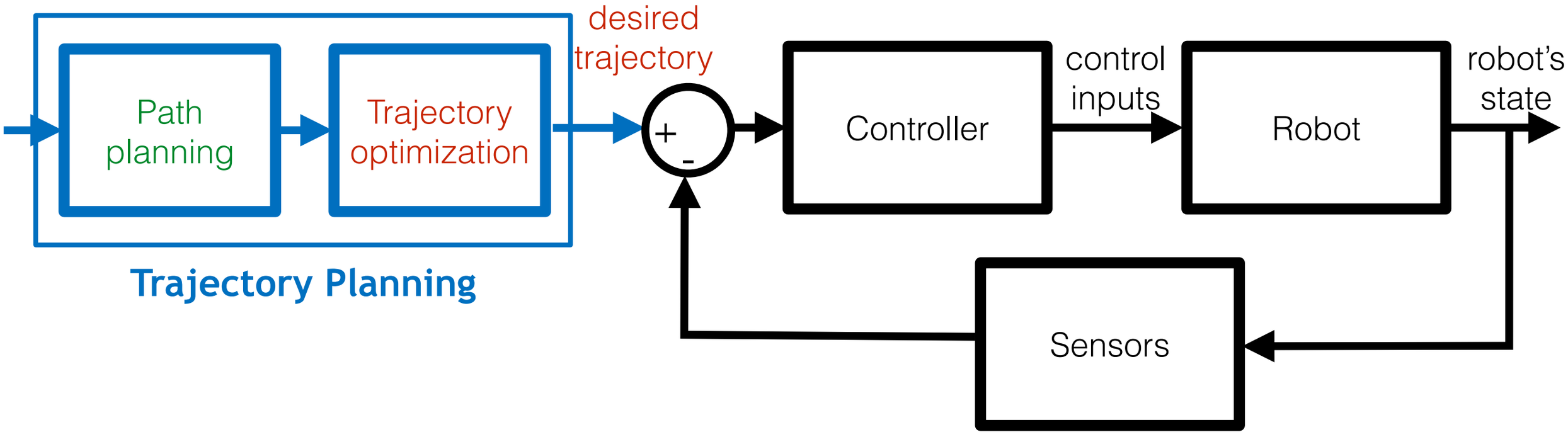


Decoupled Trajectory Planning

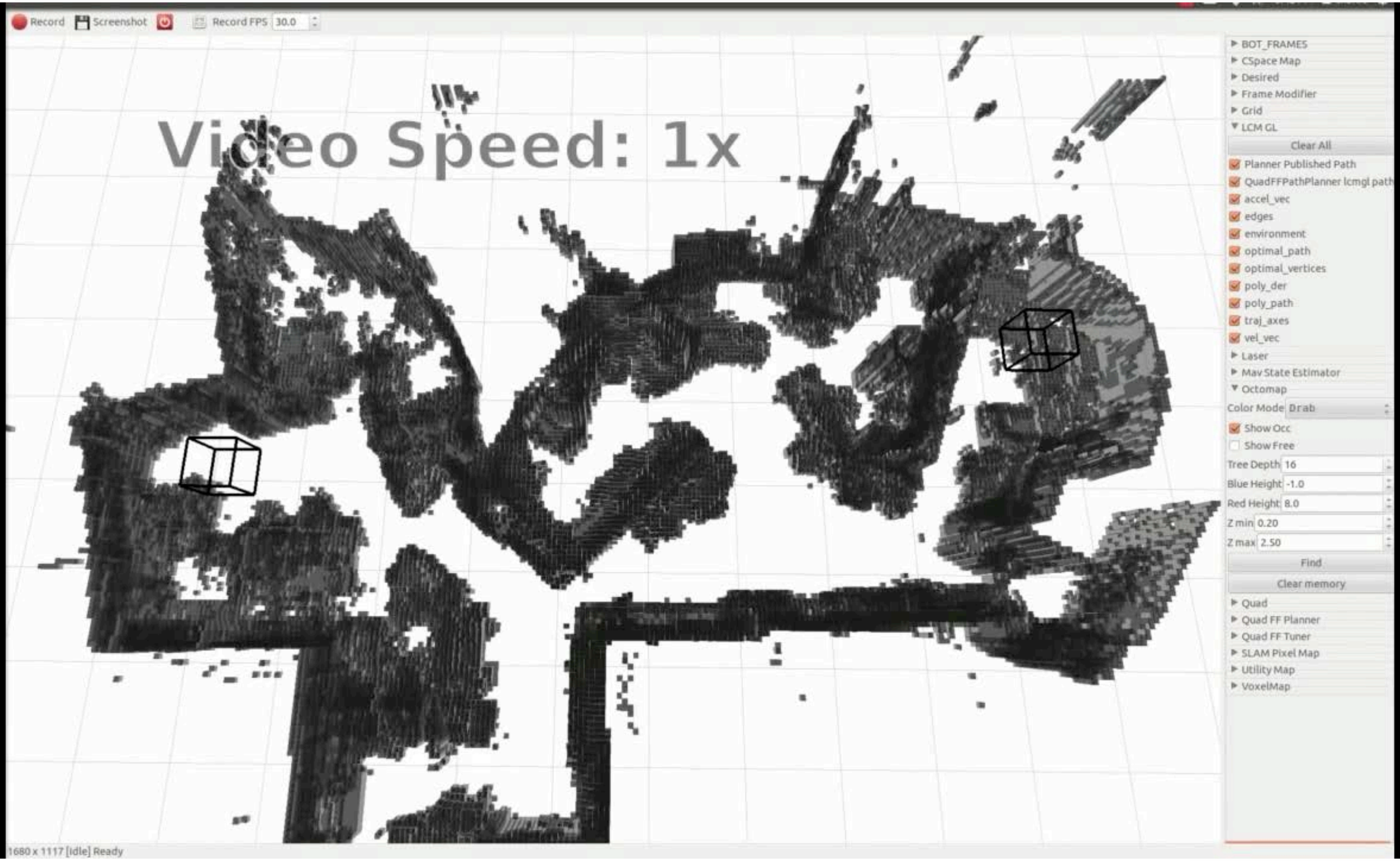




Decoupled Trajectory Planning



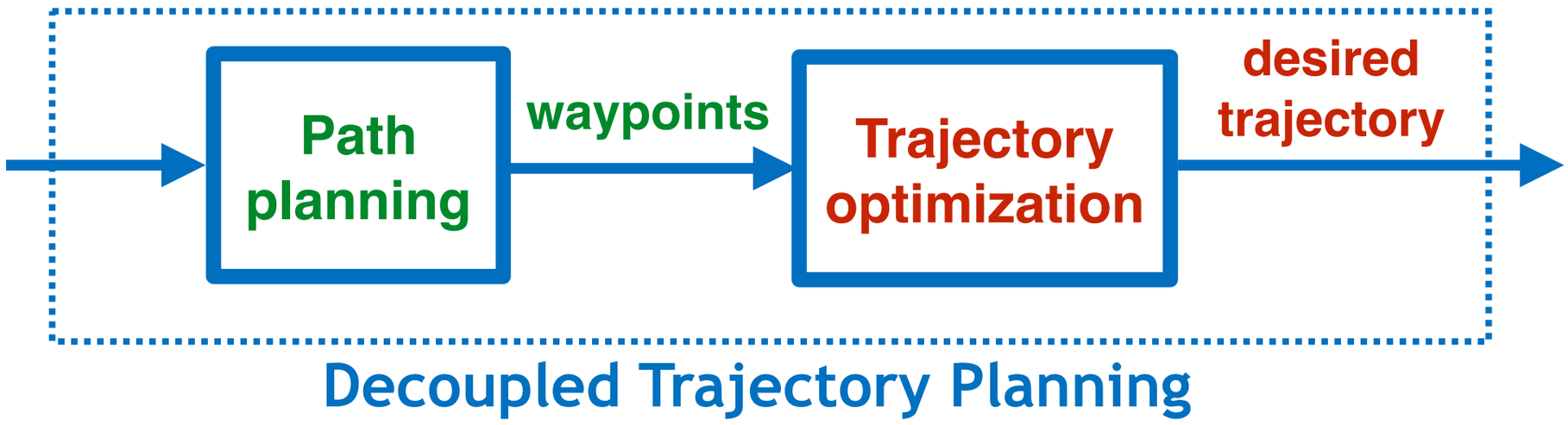
Trajectory Optimization



2013

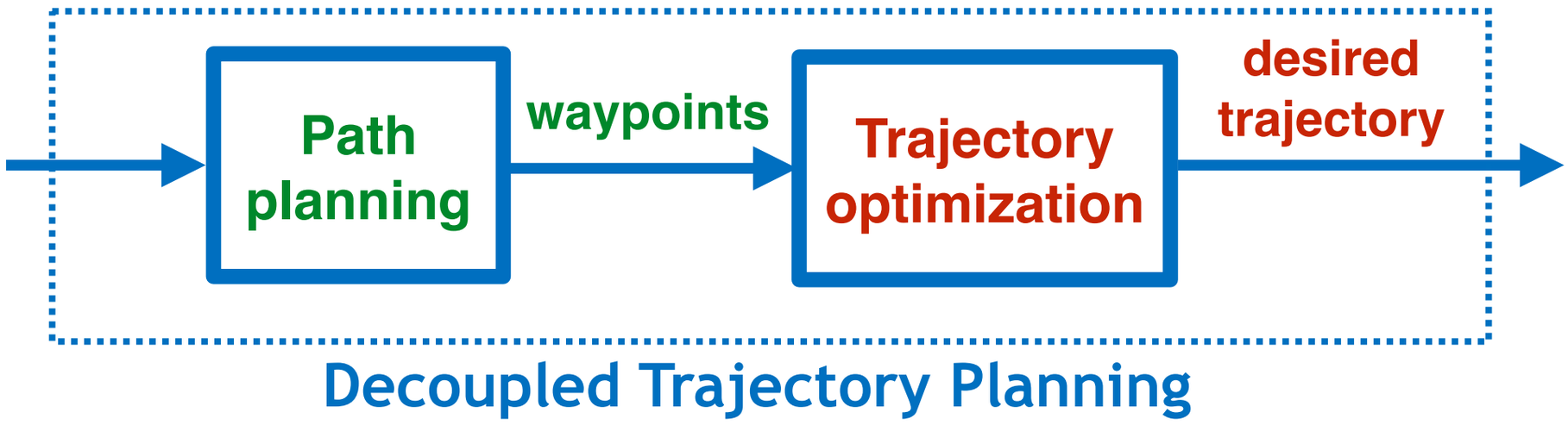
MIT AEROASTRO

Decoupled Trajectory Planning

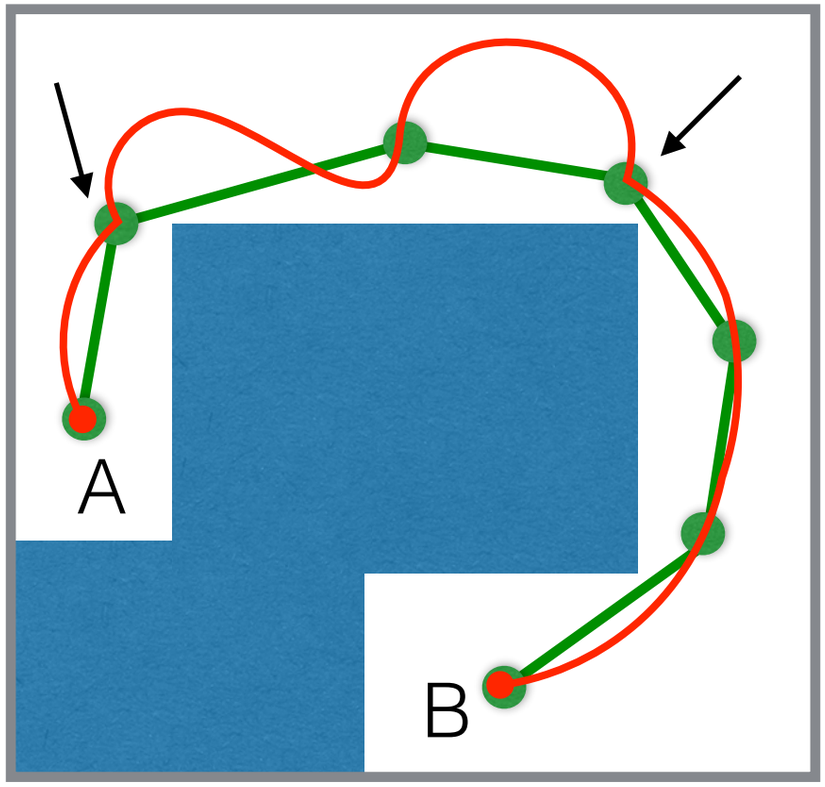


- Need to enforce **“continuity”** between segments for smooth trajectory

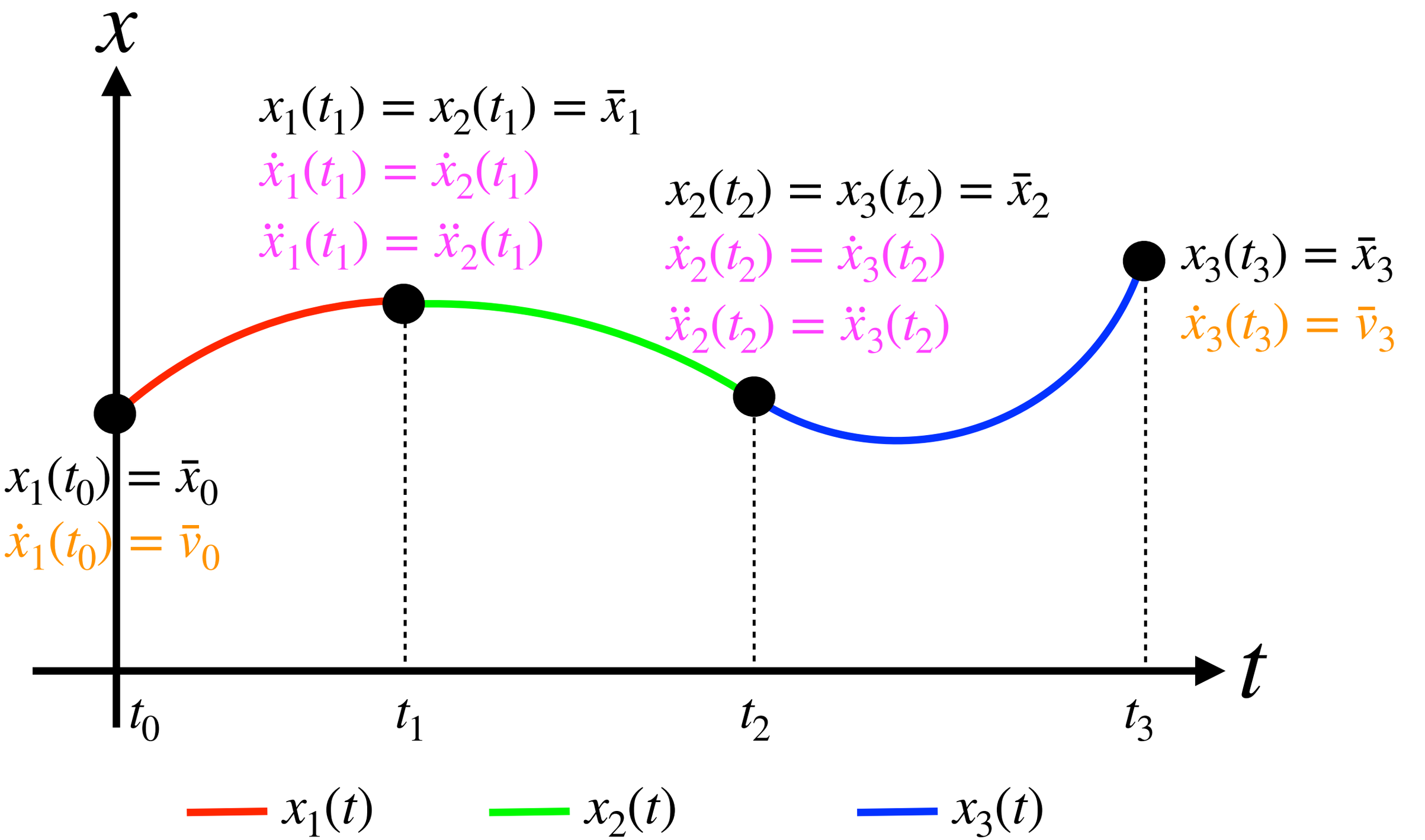
Decoupled Trajectory Planning



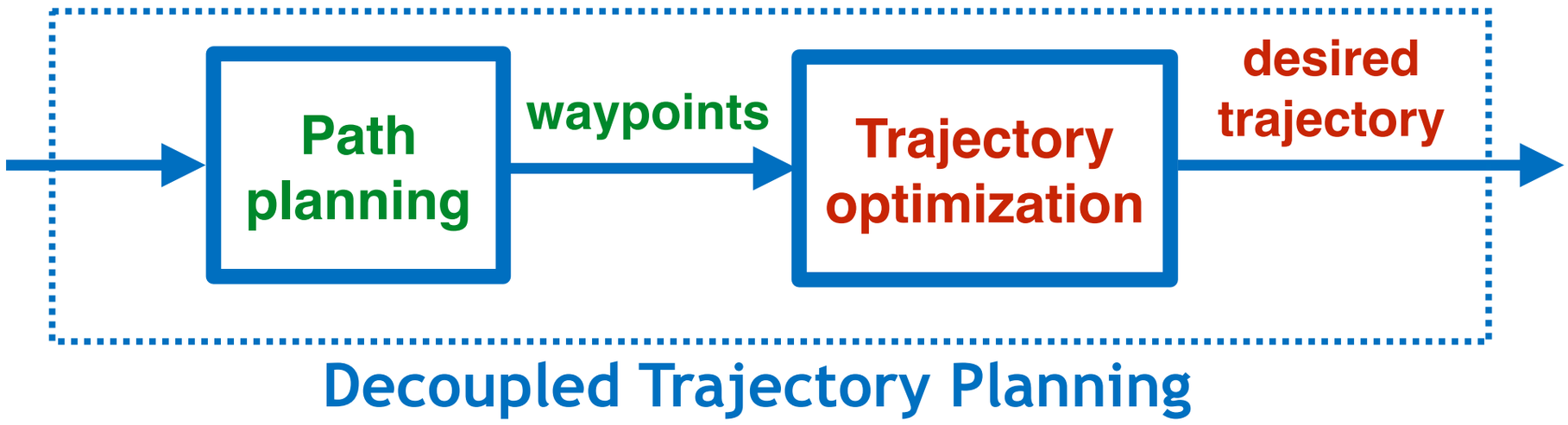
- Need to enforce **“continuity”** between segments for smooth trajectory



Trajectory Optimization

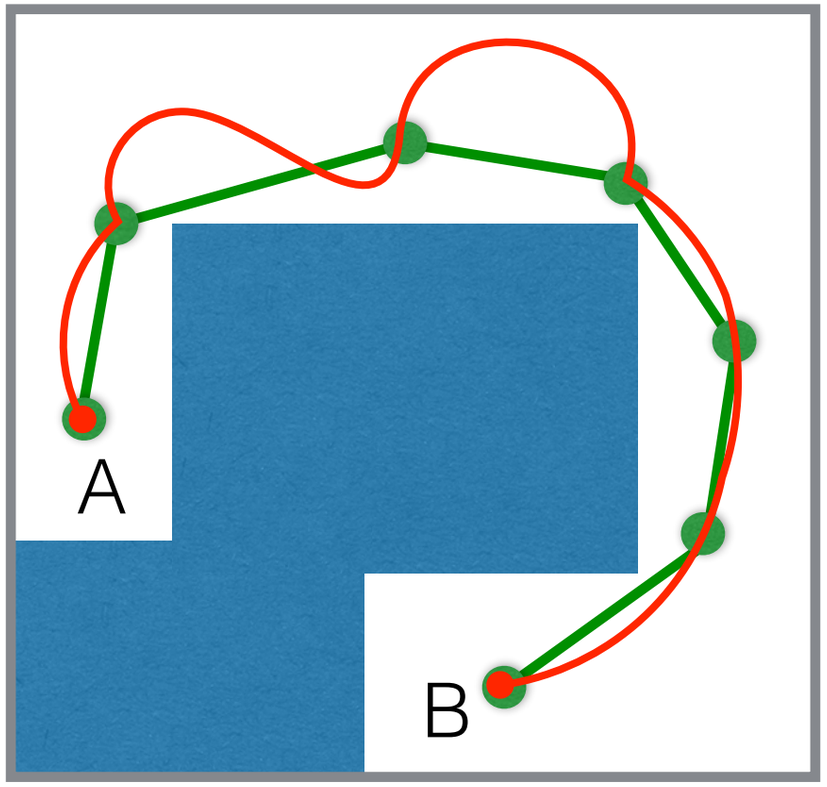


Decoupled Trajectory Planning

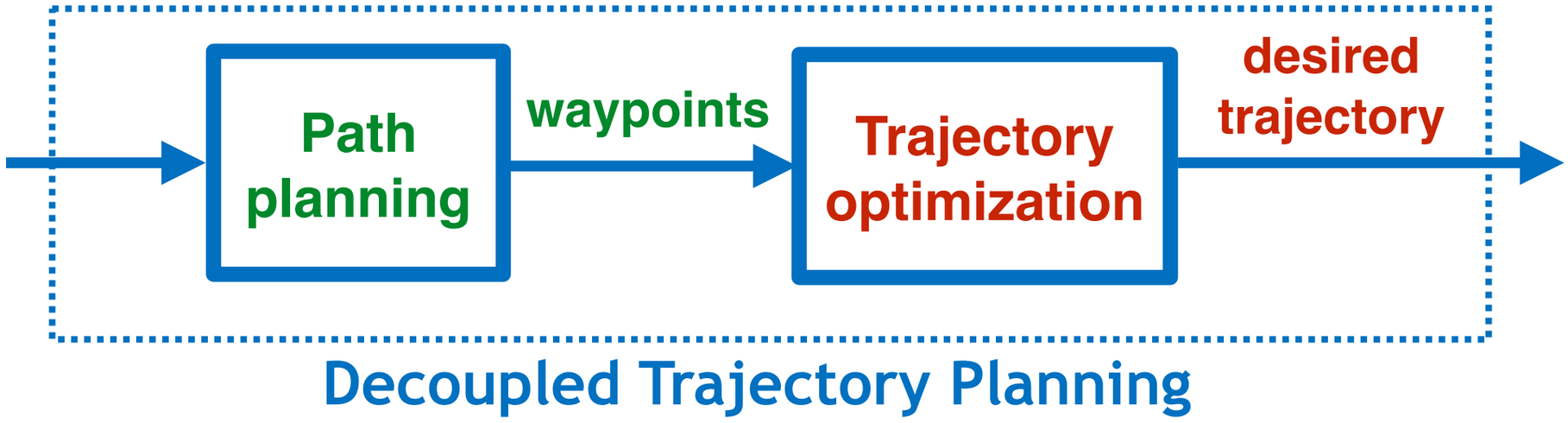


- Need to enforce **“continuity”** between segments for smooth trajectory

- Need to ensure **“minimal” motion**

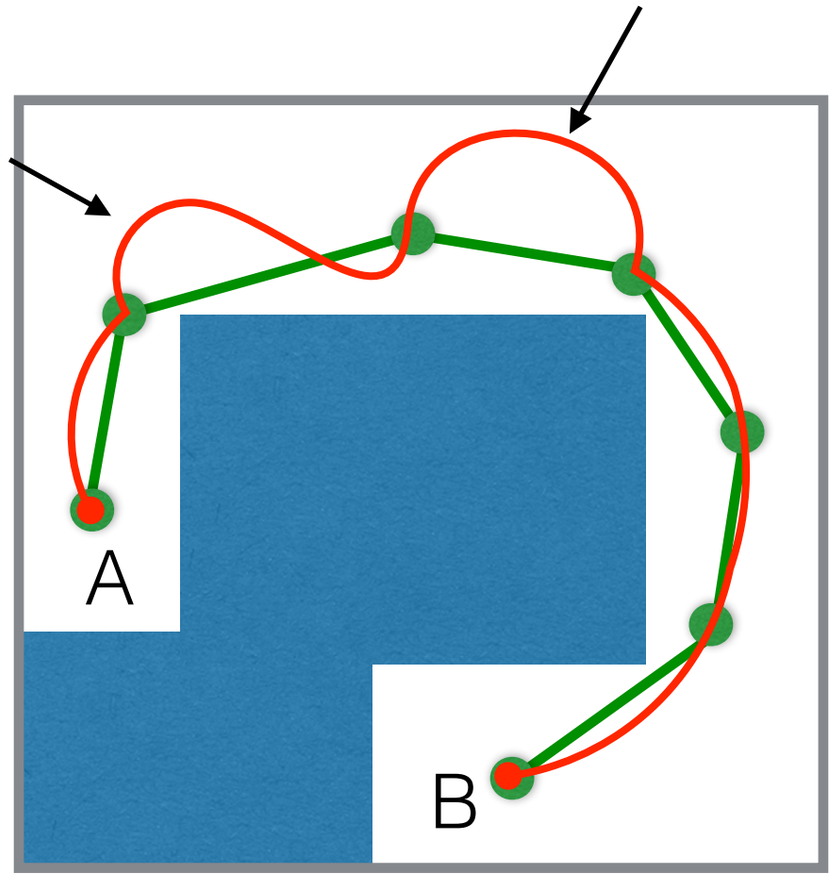


Decoupled Trajectory Planning

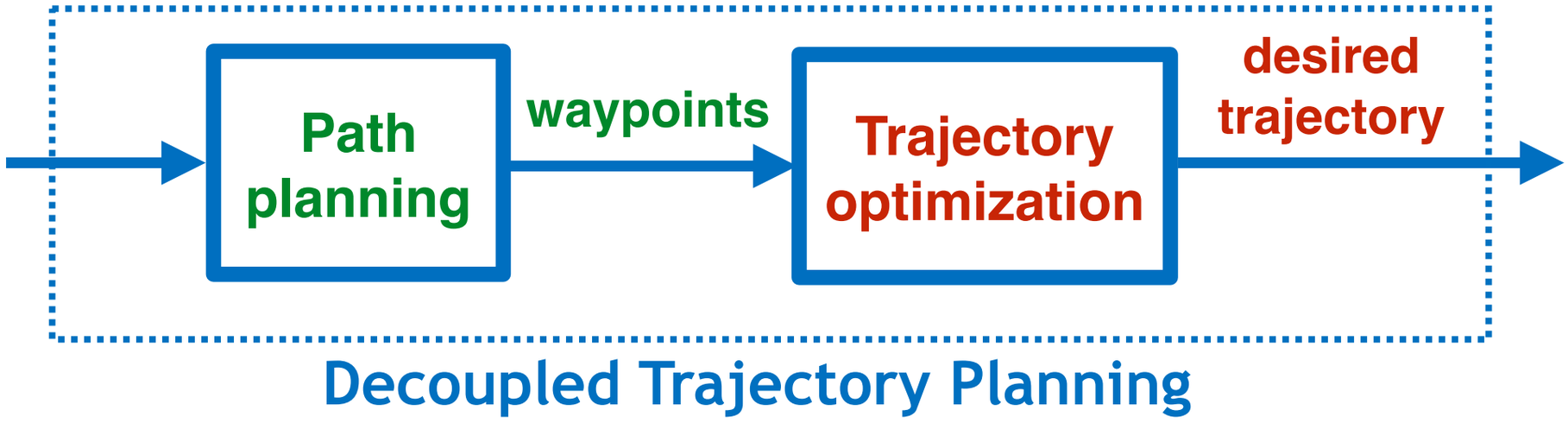


- Need to enforce **“continuity”** between segments for smooth trajectory

- Need to ensure **“minimal” motion**



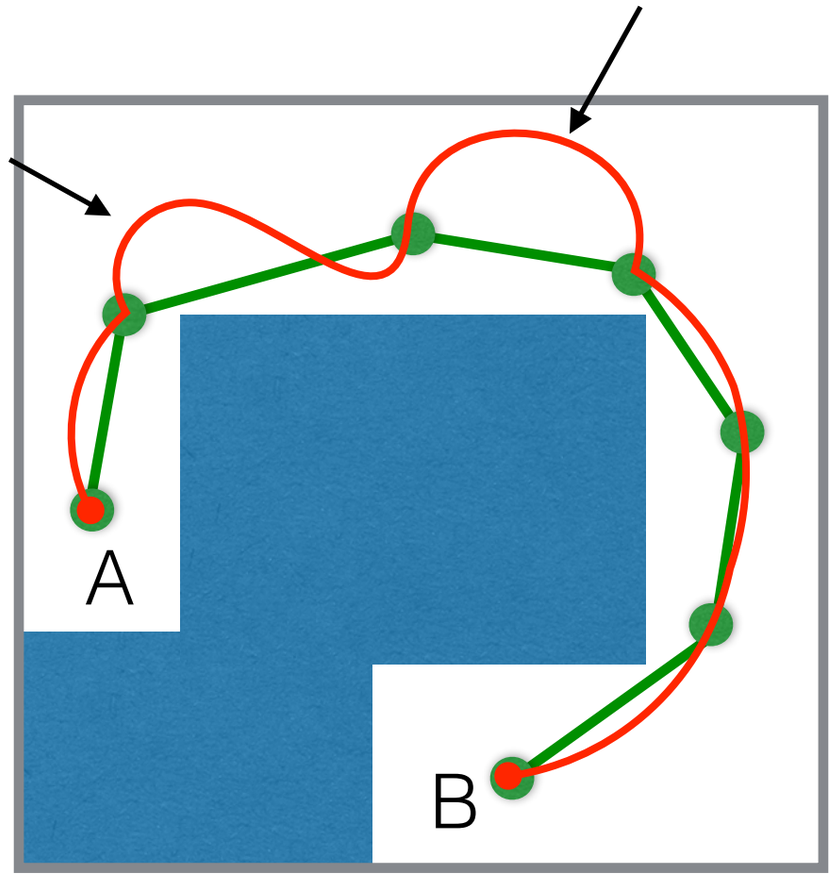
Decoupled Trajectory Planning



- Need to enforce **"continuity"** between segments for smooth trajectory

- Need to ensure **"minimal" motion**

$$\min_{x(t),u(t)} \; J(t_A, t_B, x(t), u(t))$$



Trajectory Optimization

Estimation, Control and Planning for Aggressive Flight with a Small Quadrotor with a Single Camera and IMU

Giuseppe Loianno
Vijay Kumar
Chris Brunner
Gary McGrath

Penn Engineering | GRASP Laboratory
General Robotics, Automation, Sensing & Perception Lab

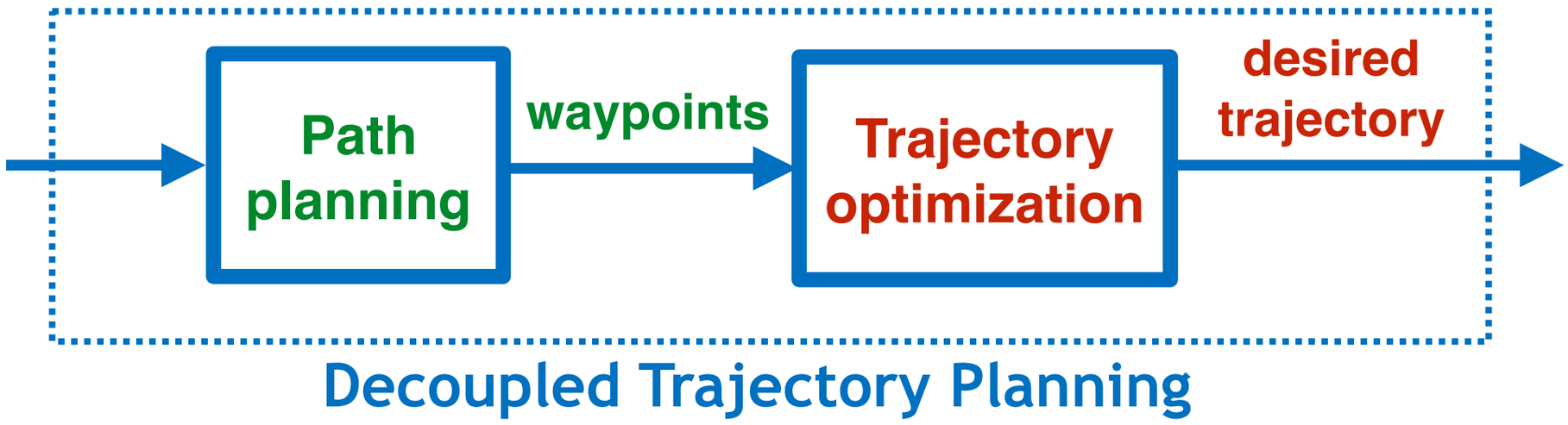
Qualcomm Technologies Inc.
Qualcomm Research is a division of Qualcomm Technologies Inc.

www.kumarrobotics.org

2016/2017

MIT AEROASTRO

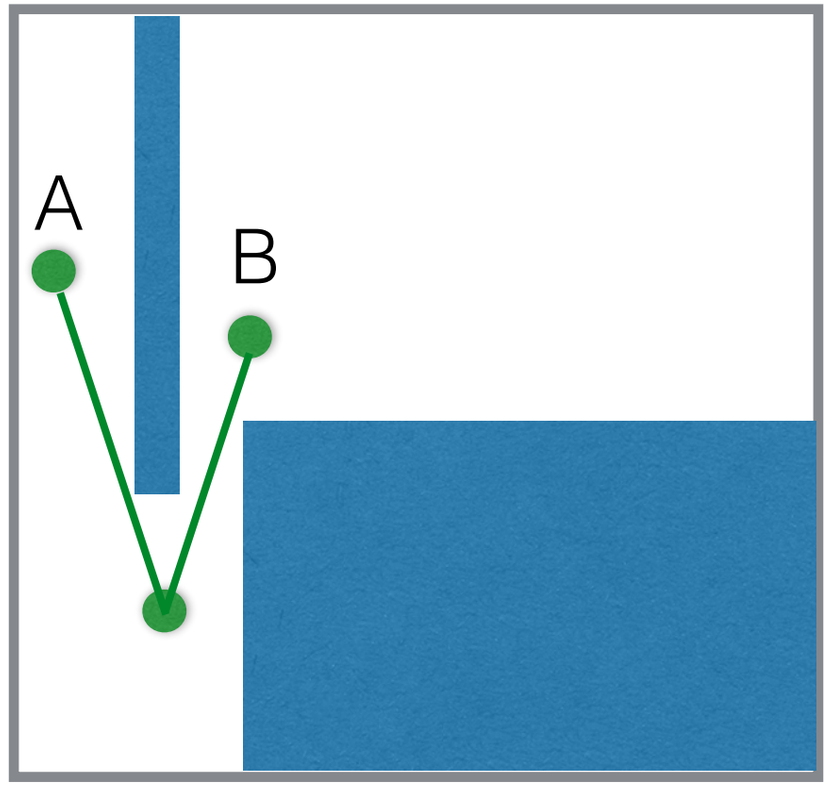
Decoupled Trajectory Planning



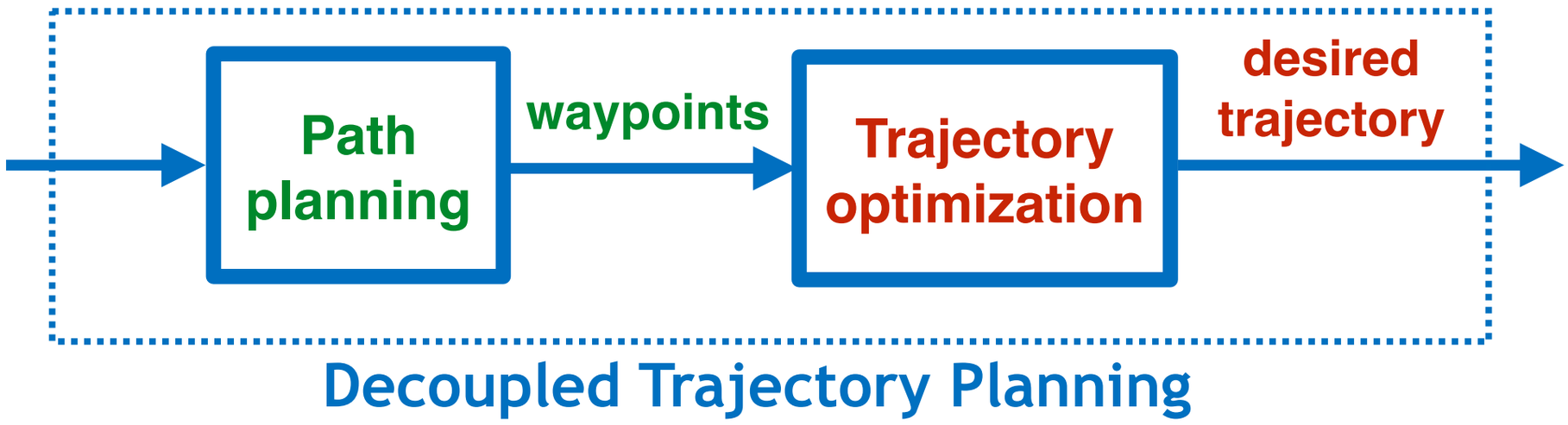
- Need to enforce **“continuity”** between segments for smooth trajectory

- Need to ensure **“minimal” motion**

- Need to ensure **feasibility; e.g.:**
  - Hit no obstacles
  - Don’t saturate your controller



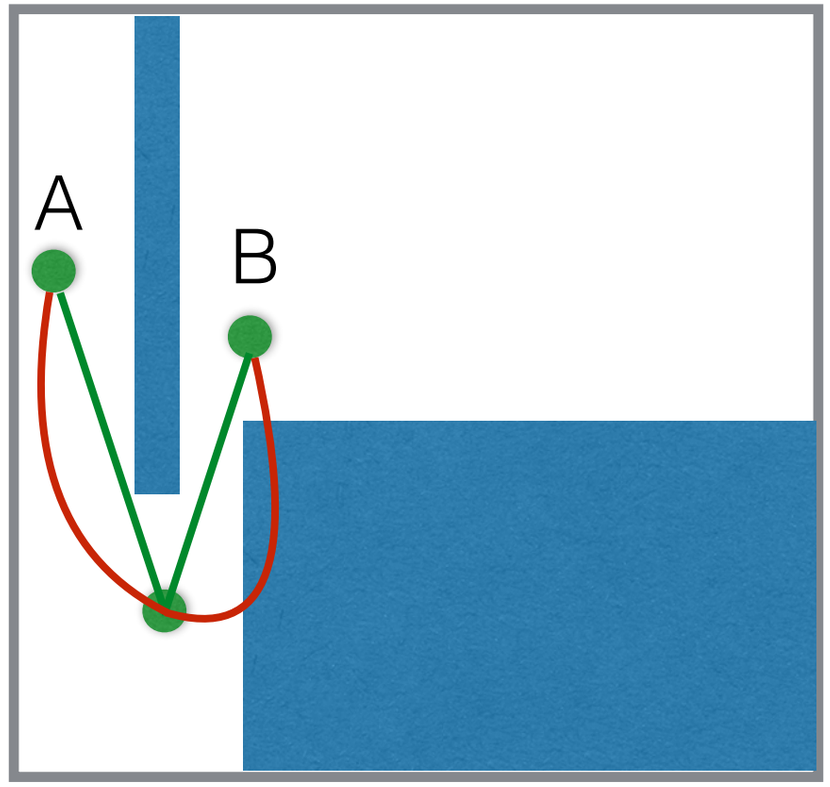
Decoupled Trajectory Planning



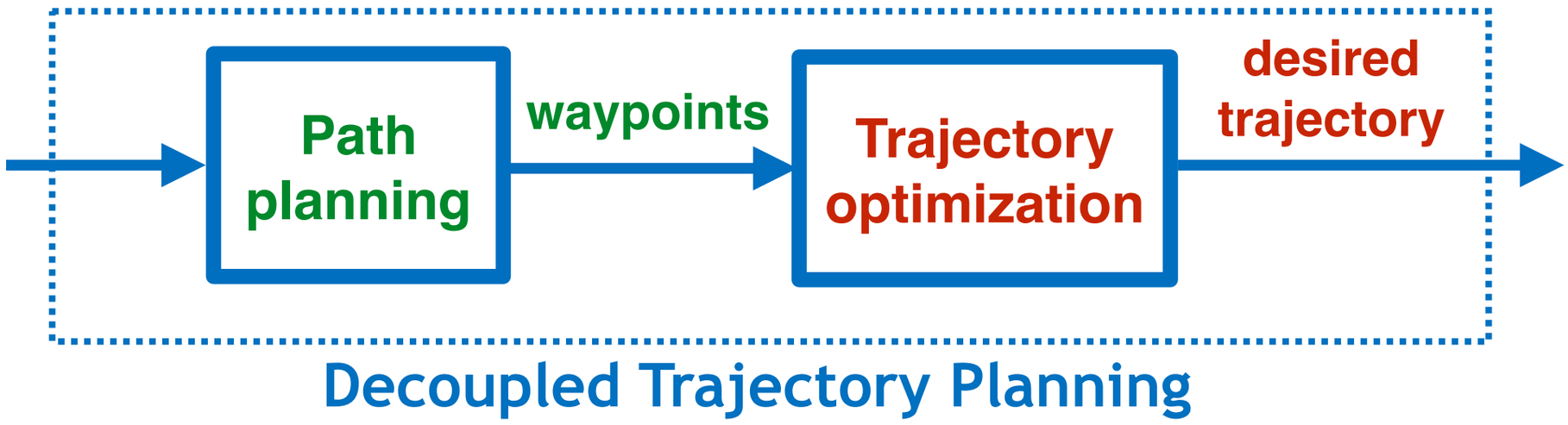
- Need to enforce **“continuity”** between segments for smooth trajectory

- Need to ensure **“minimal” motion**

- Need to ensure **feasibility; e.g.:**
  - Hit no obstacles
  - Don’t saturate your controller



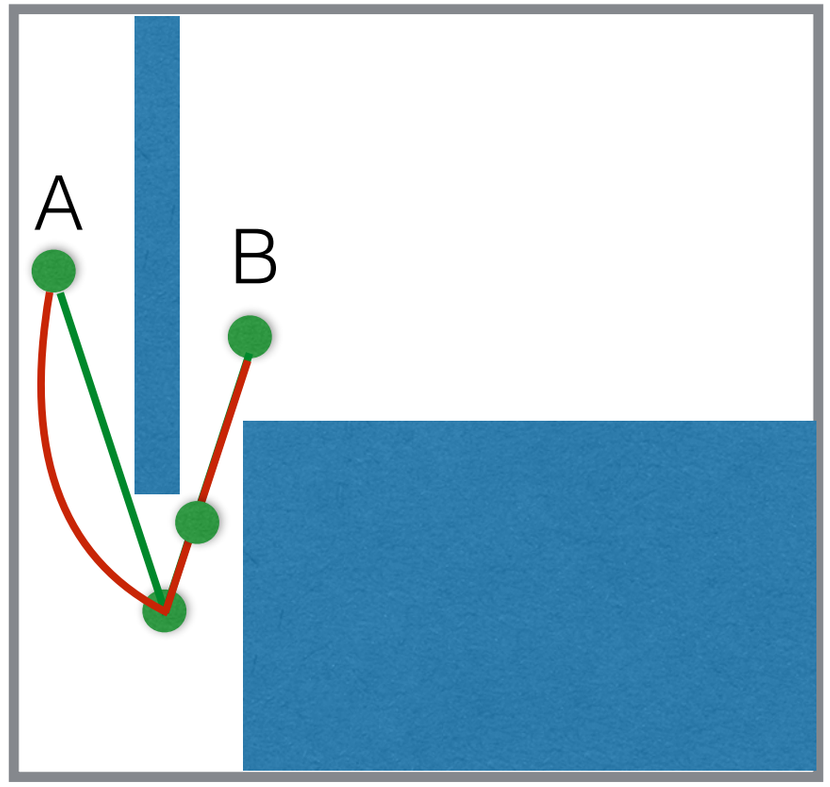
Decoupled Trajectory Planning



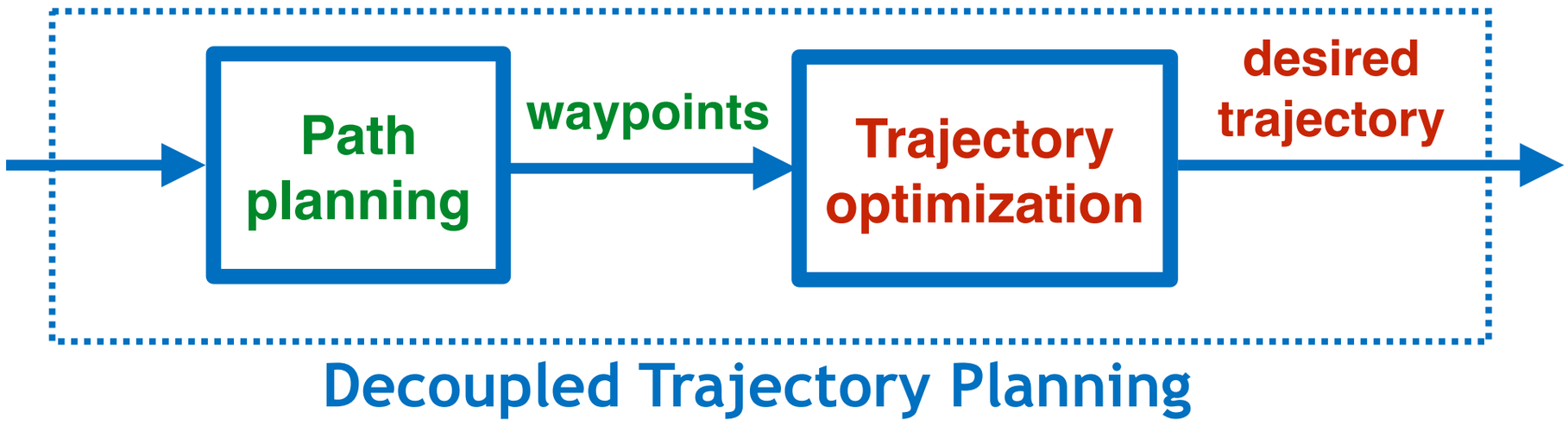
- Need to enforce **“continuity”** between segments for smooth trajectory

- Need to ensure **“minimal” motion**

- Need to ensure **feasibility; e.g.:**
  - Hit no obstacles
  - Don’t saturate your controller



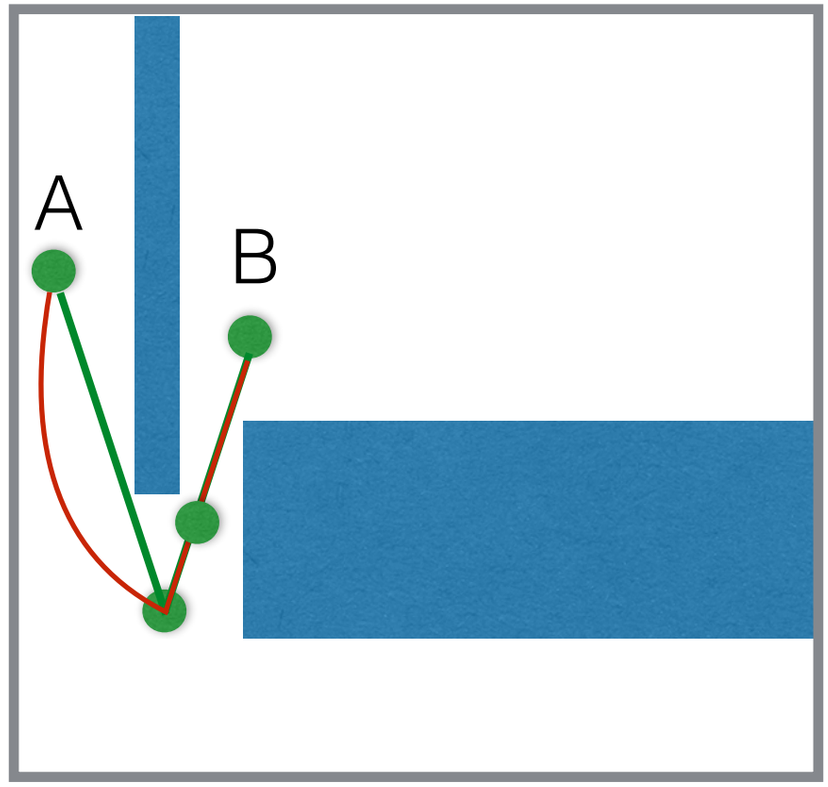
Decoupled Trajectory Planning



- Need to enforce **“continuity”** between segments for smooth trajectory

- Need to ensure **“minimal” motion**

- Need to ensure **feasibility; e.g.:**
  - Hit no obstacles
  - Don’t saturate your controller



16.485 Visual Navigation for Autonomous Vehicles (VNAV)
Fall 2020