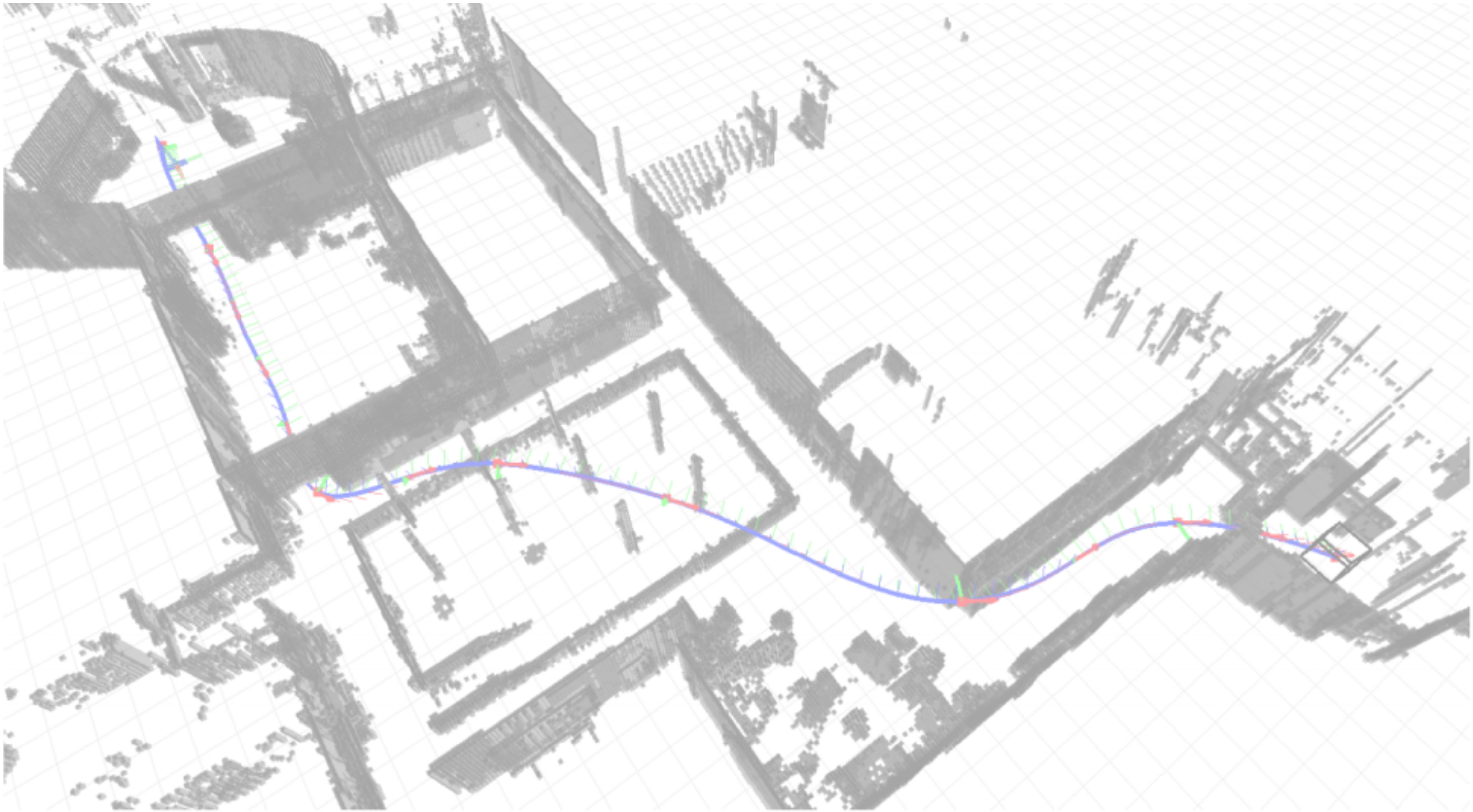
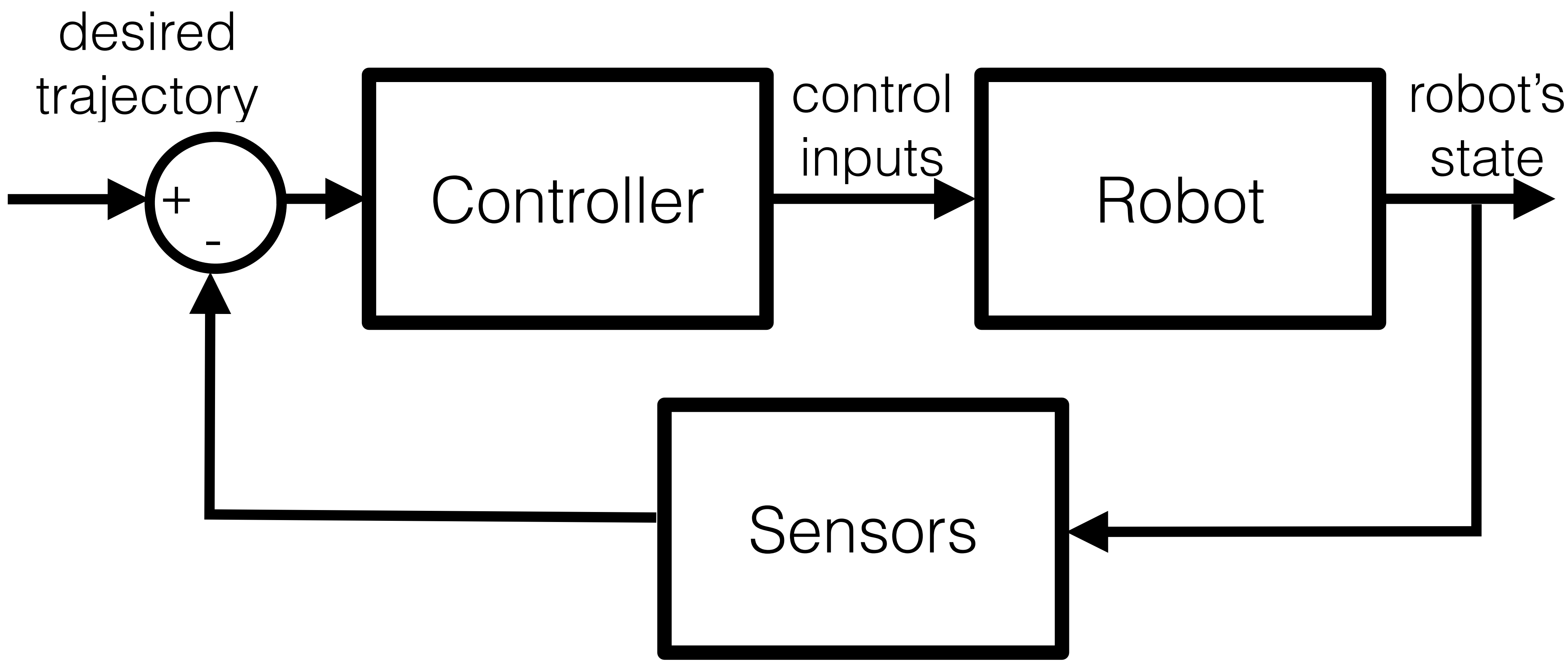


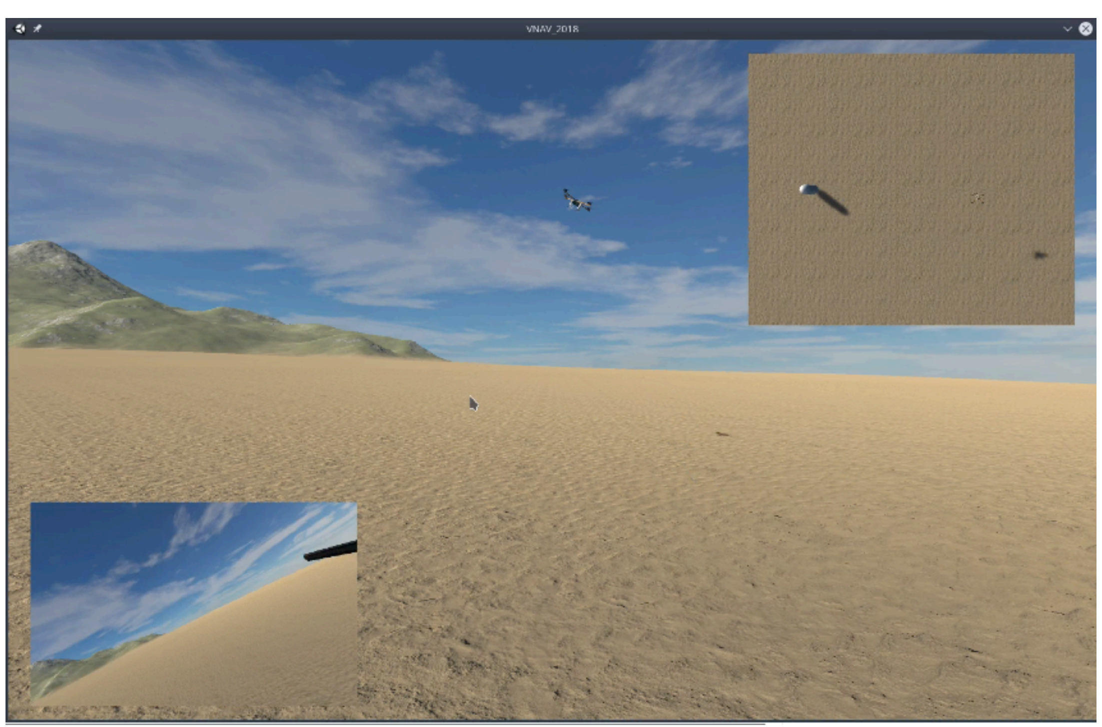
16.485: VNAV - Visual Navigation for Autonomous Vehicles

Lecture 8: Trajectory Optimization

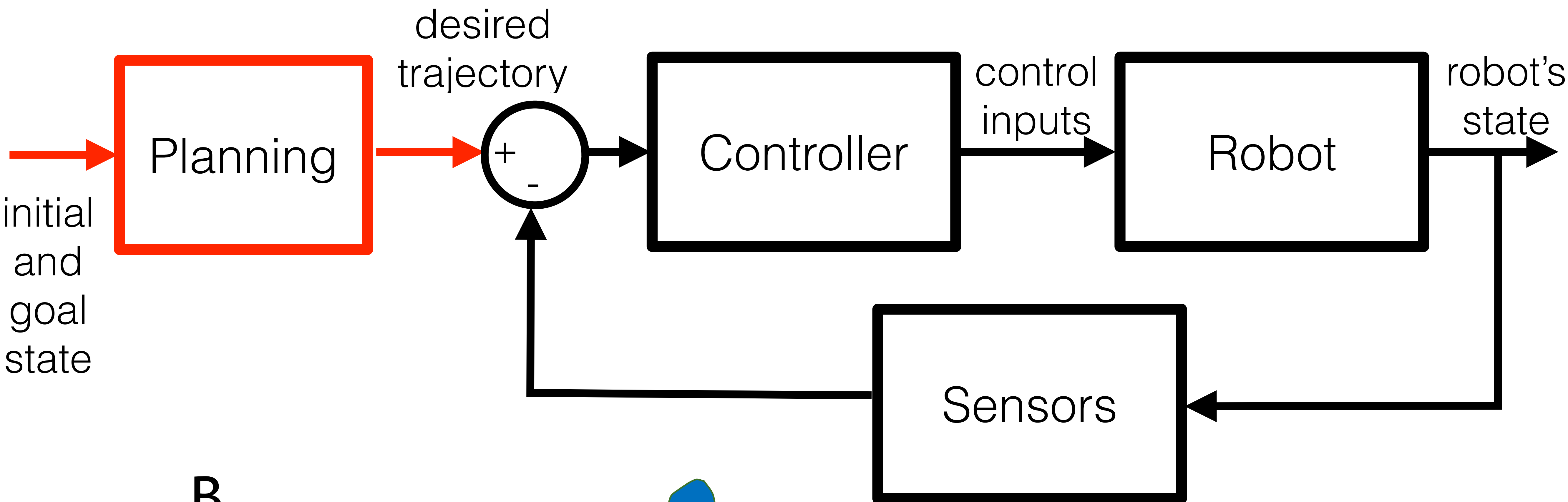
Luca Carlone

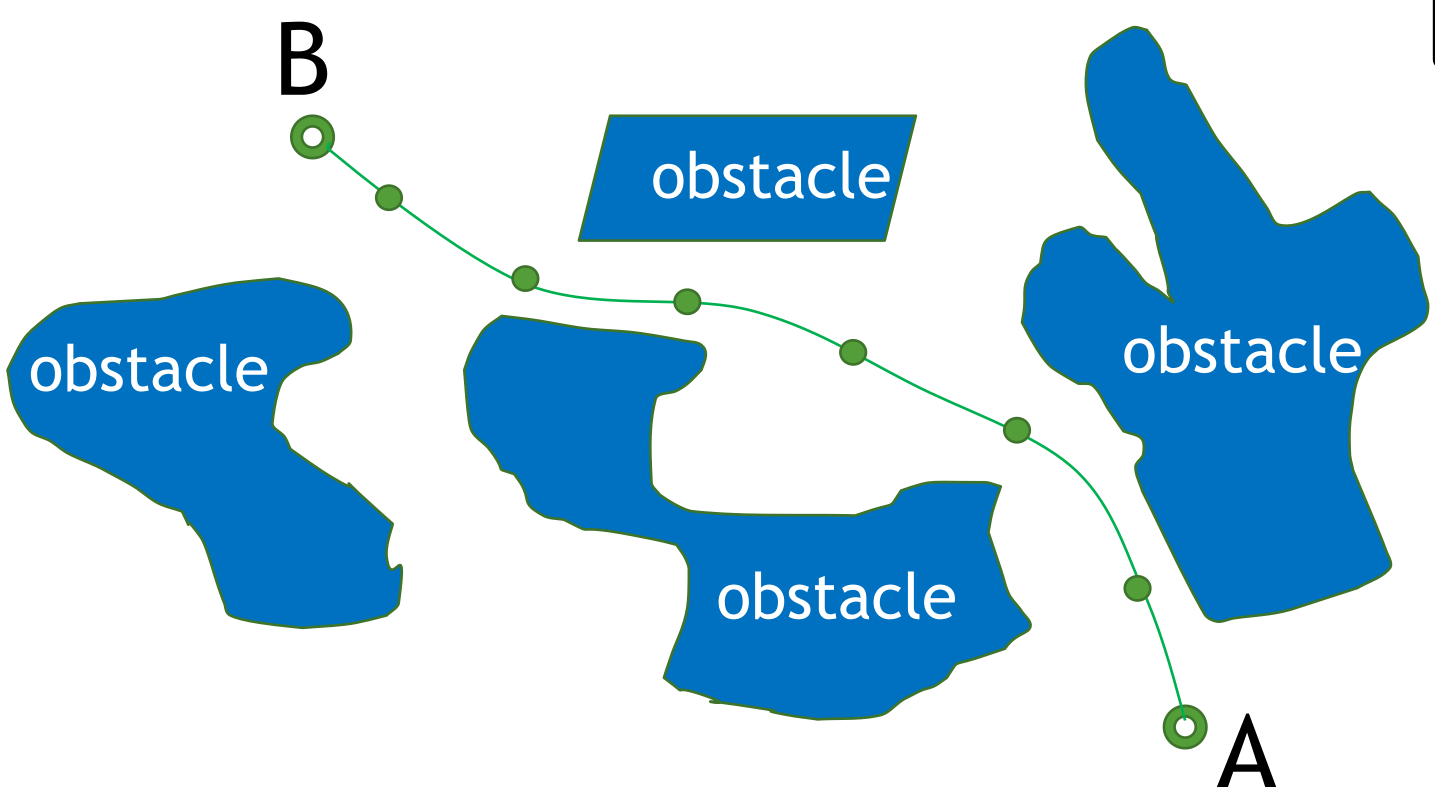
Planning vs. Control





Planning vs. Control

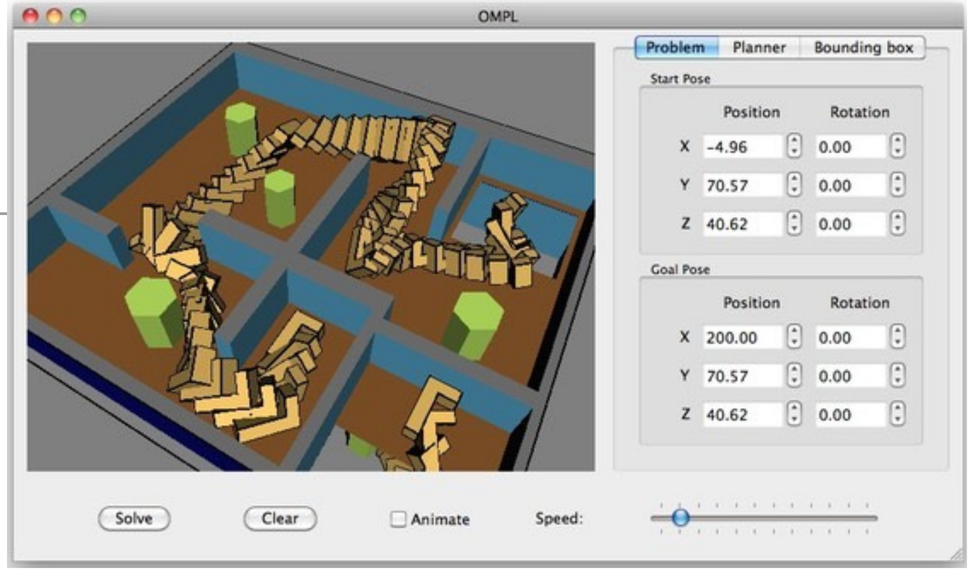




Trajectory Planning vs. Path Planning

- Path vs. trajectory:
  - A **path** usually contains a sequence of waypoints, without time label or information about velocity or higher order of derivatives.
  - A **trajectory** defines the path the robot should navigate as a function of time.
- Path planning:
  - "slow" systems
  - Many approaches: graph-based algorithms (PRM, A* and variants), sampling-based methods (RRT, RRT* and variants)
- Trajectory planning:
  - Key for high-speed maneuvers on drones

Path / Motion Planning

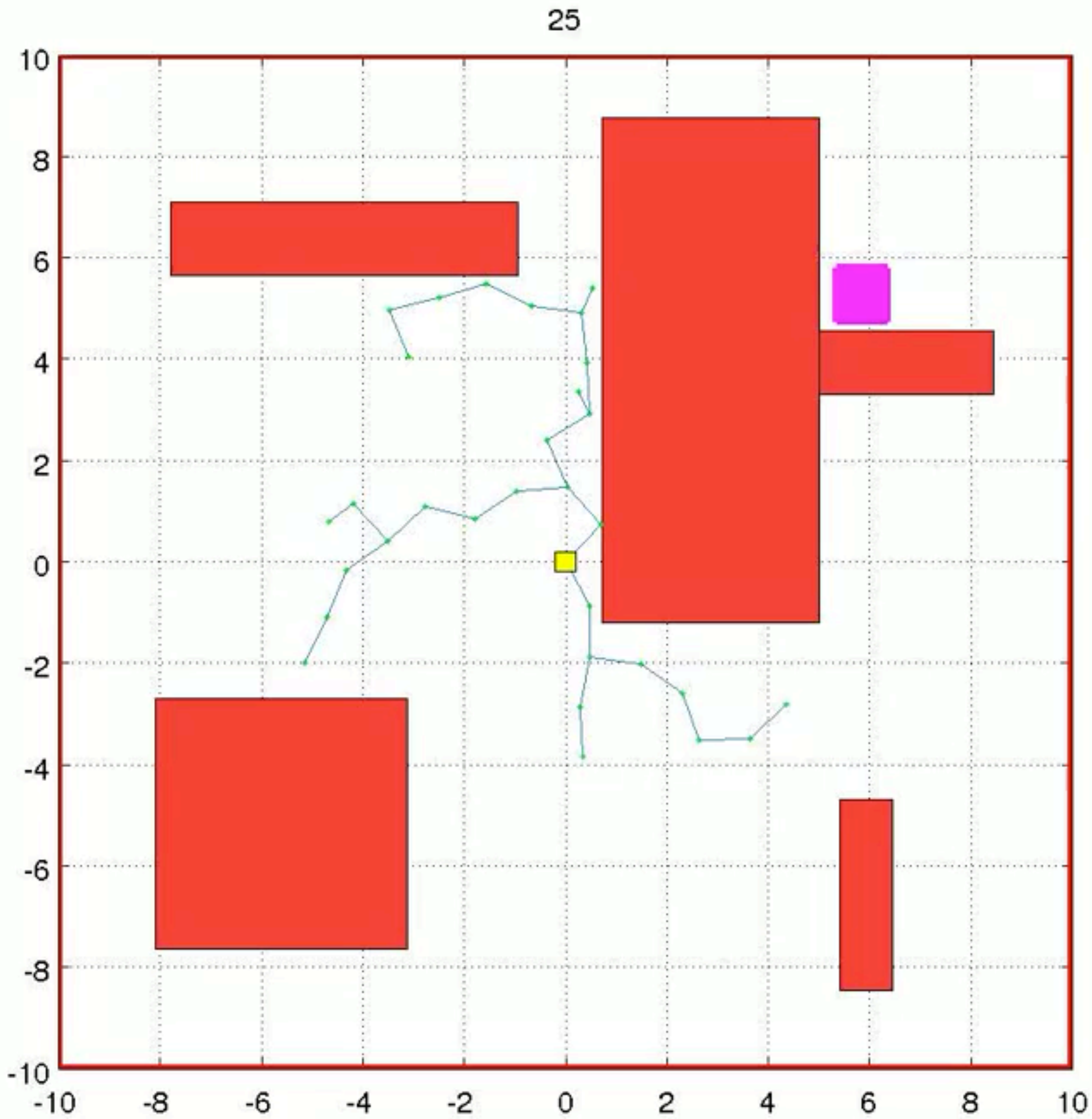


- Open-source libraries
  - The Open Motion Planning Library (OMPL)
    - http://ompl.kavrakilab.org/
  - Motion Strategy Library (MSL)
    - http://msl.cs.uiuc.edu/msl/
  - RRT* Library
    - https://svn.csail.mit.edu/rrtstar/
  - Sampling Based Planning Library
    - https://svn.csail.mit.edu/smp

- References:
  - Howie Choset , Kevin Lynch et al, " Principles of Robot Motion" MIT press, 2005.
  - Steven Lavalle, " Planning Algorithms", Cambridge University Press, 2006.

Path / Motion Planning Example

- RRT*: Rapidly exploring Random Trees



Path / Motion Planning

COMBINATORIAL PLANNING

- (resolution) complete
  - they find the solution if exists, otherwise they report failure
- No need for collision checking, since is explicitly defined
- Expensive computation, impractical for higher dimensional systems
  - Exploding number of cells
- The output path is usually jagged and needs smoothing

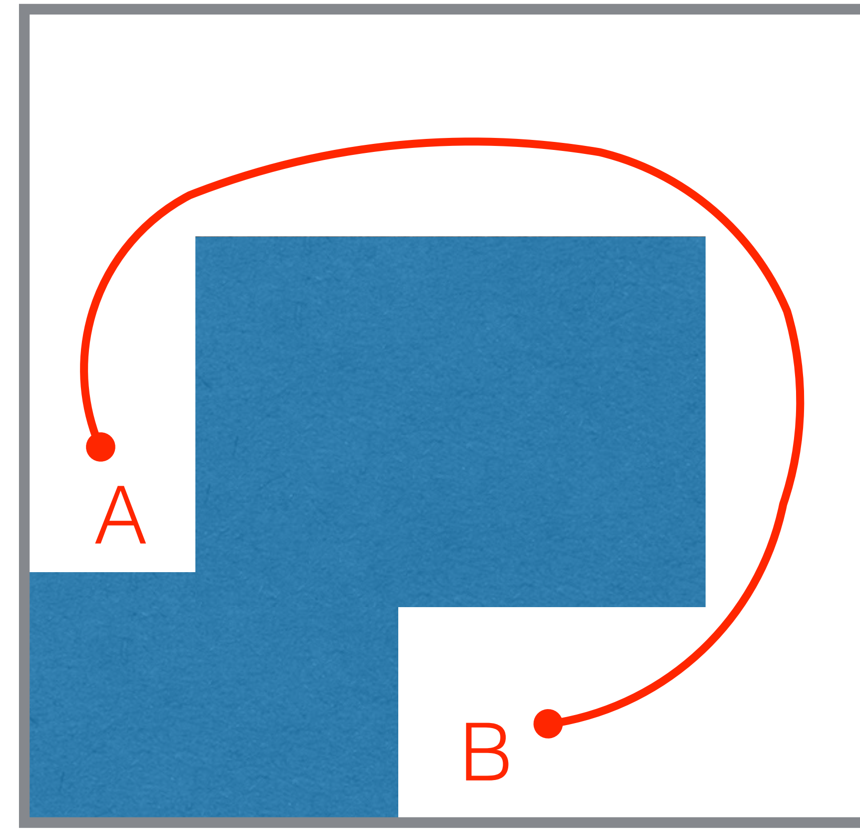
SAMPLING BASED PLANNING

- Probabilistic Complete
  - The probability of finding a solution if it exists tends to one, otherwise it may run forever.
- More practical
- Narrow passage challenge
- The output path is usually jagged and needs smoothing

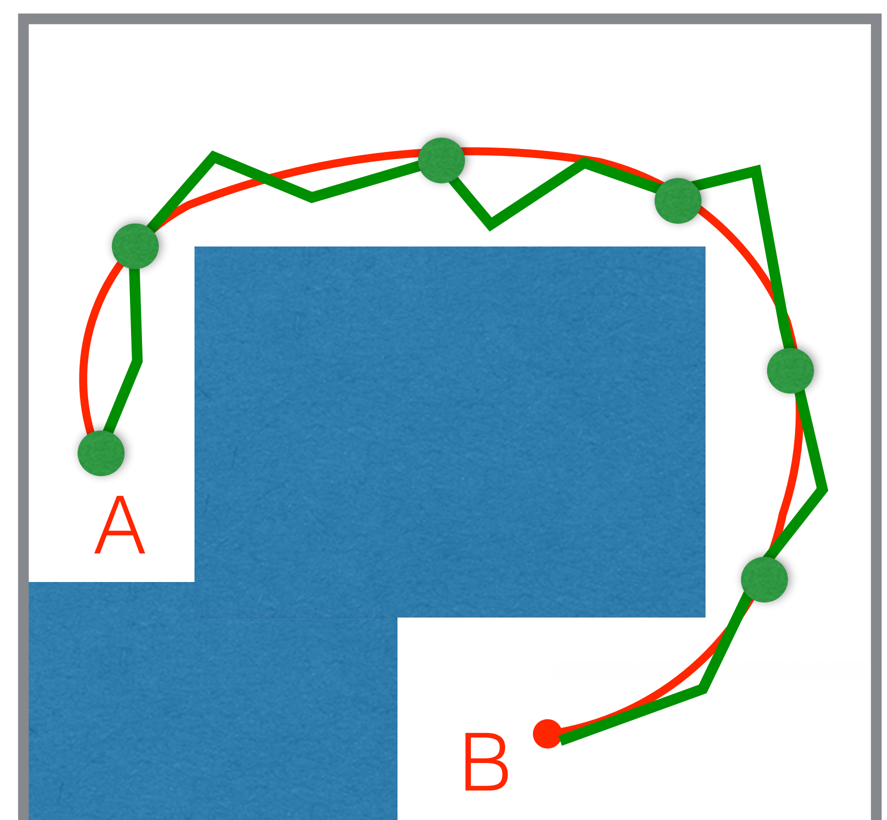
MIT AEROASTRO

Trajectory Planning: Two Approaches

- **Direct trajectory planning**: plan the path and optimize trajectory simultaneously
  - Typically slow for online planning and fast dynamics (difficult to account for obstacles)
  - Fast heuristics (next slide): motion primitives

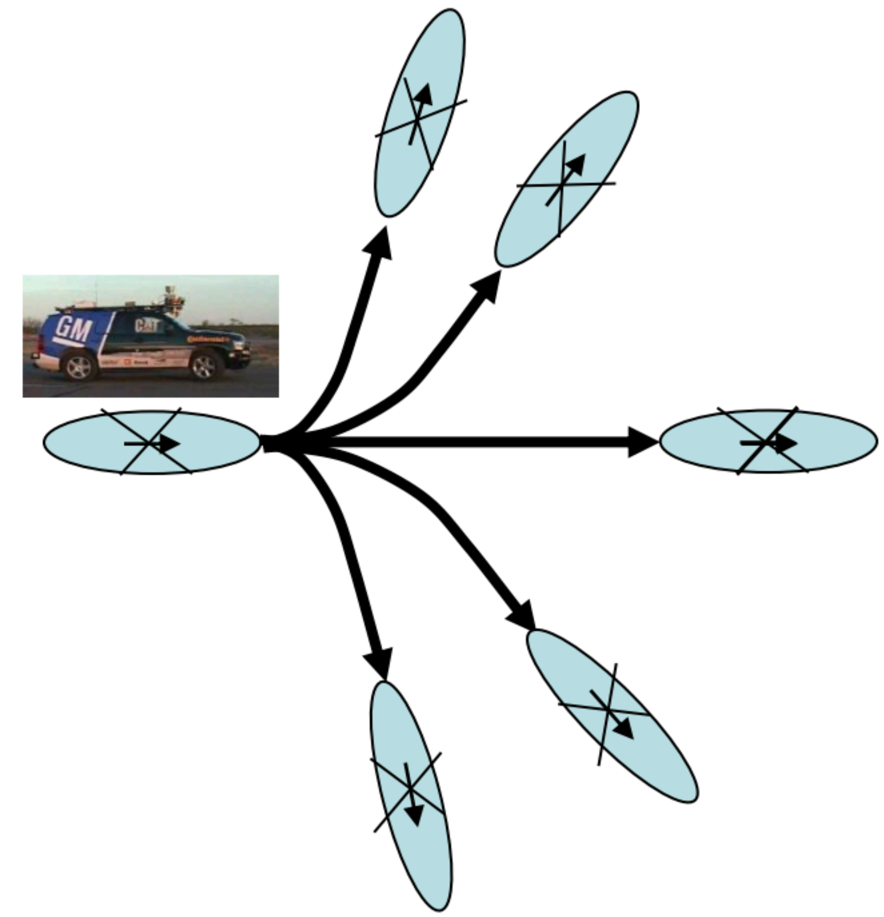


- **Decoupled trajectory planning**: first path planning then trajectory generation
  - Fast computation (only partially accounts for obstacles)
  - Polynomial trajectory optimization

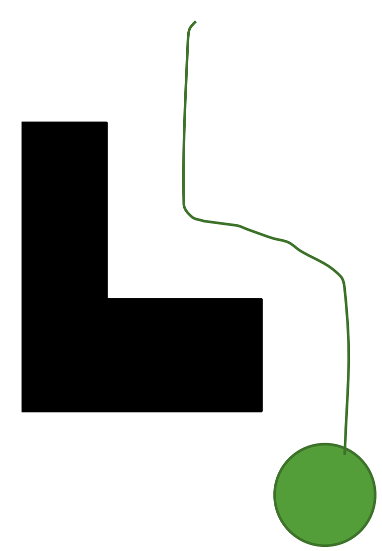


Motion Primitives

- **Input**:
  - A library of motion primitives: each motion primitive is generated by forward simulation of the system for a given control sequence
  - The current robot state
- **Output**: a choice of motion primitive that minimizes a cost function while avoiding obstacles.



- The final trajectory is usually not smooth
- More complex instantiations: https://www.cs.cmu.edu/~maxim/files/tutorials/robschooltutorial_oct10.pdf



Number	Description
1	Straight
2	Climb
3	Takeoff (no throttle)
4	Gentle left
5	Gentle right
6	Left jog
7	Right jog

Output: 1->1>4->1->1->6

Motion Primitives For Aggressive Flight

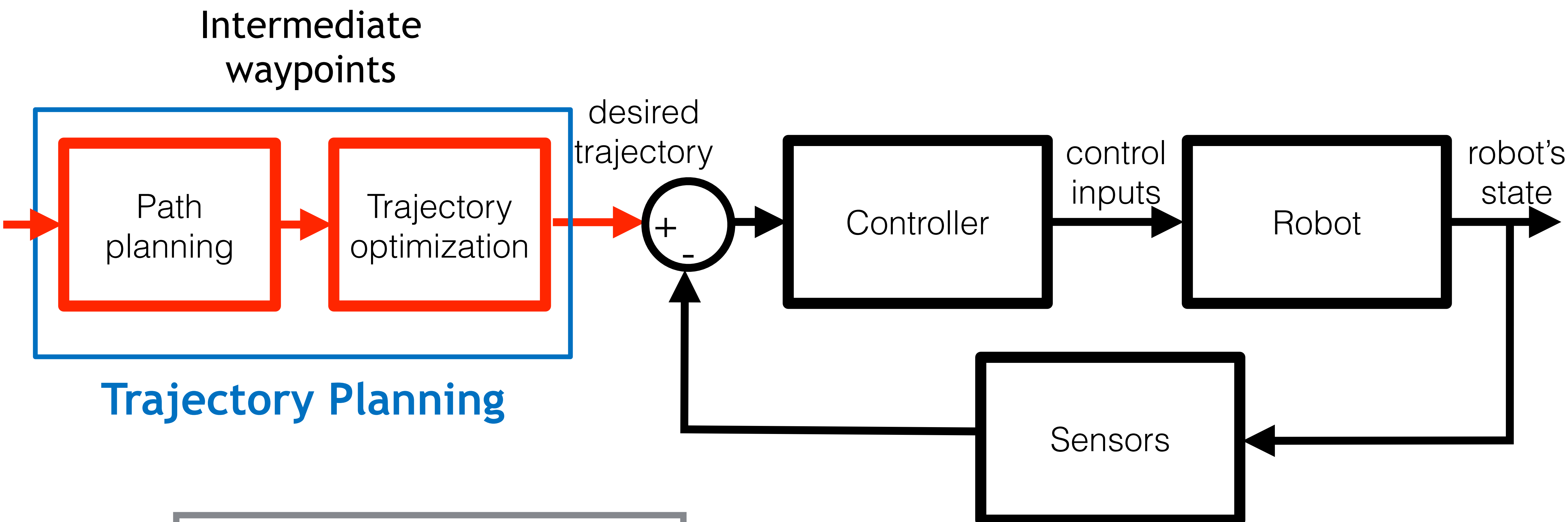
2017

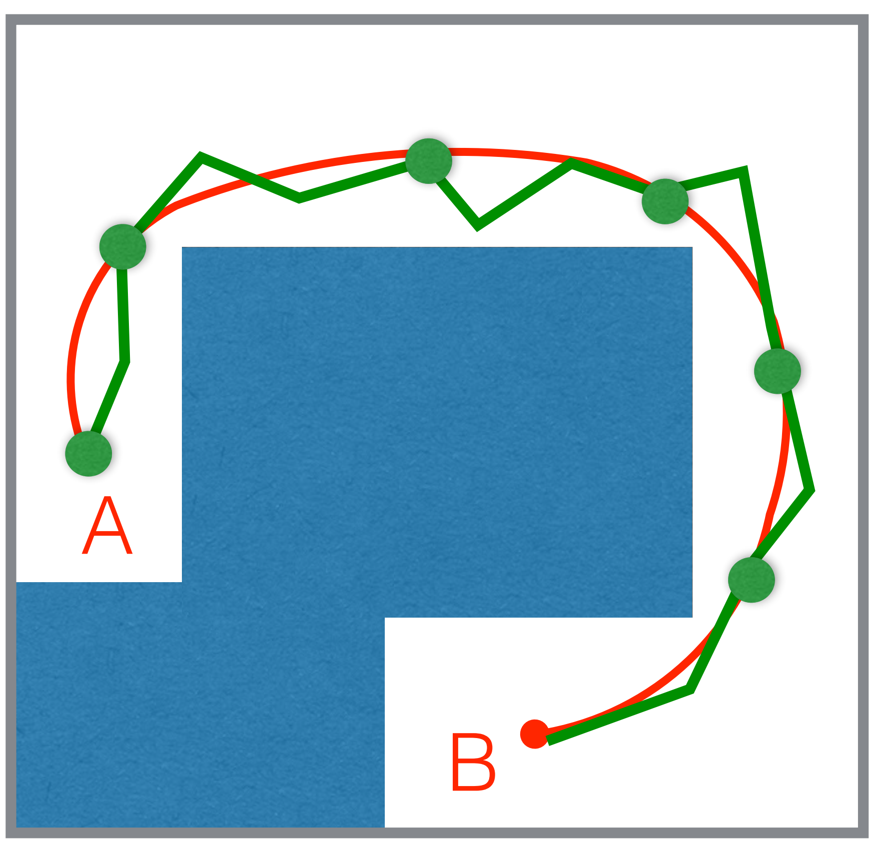
Motion Primitives For Aggressive Flight

2019

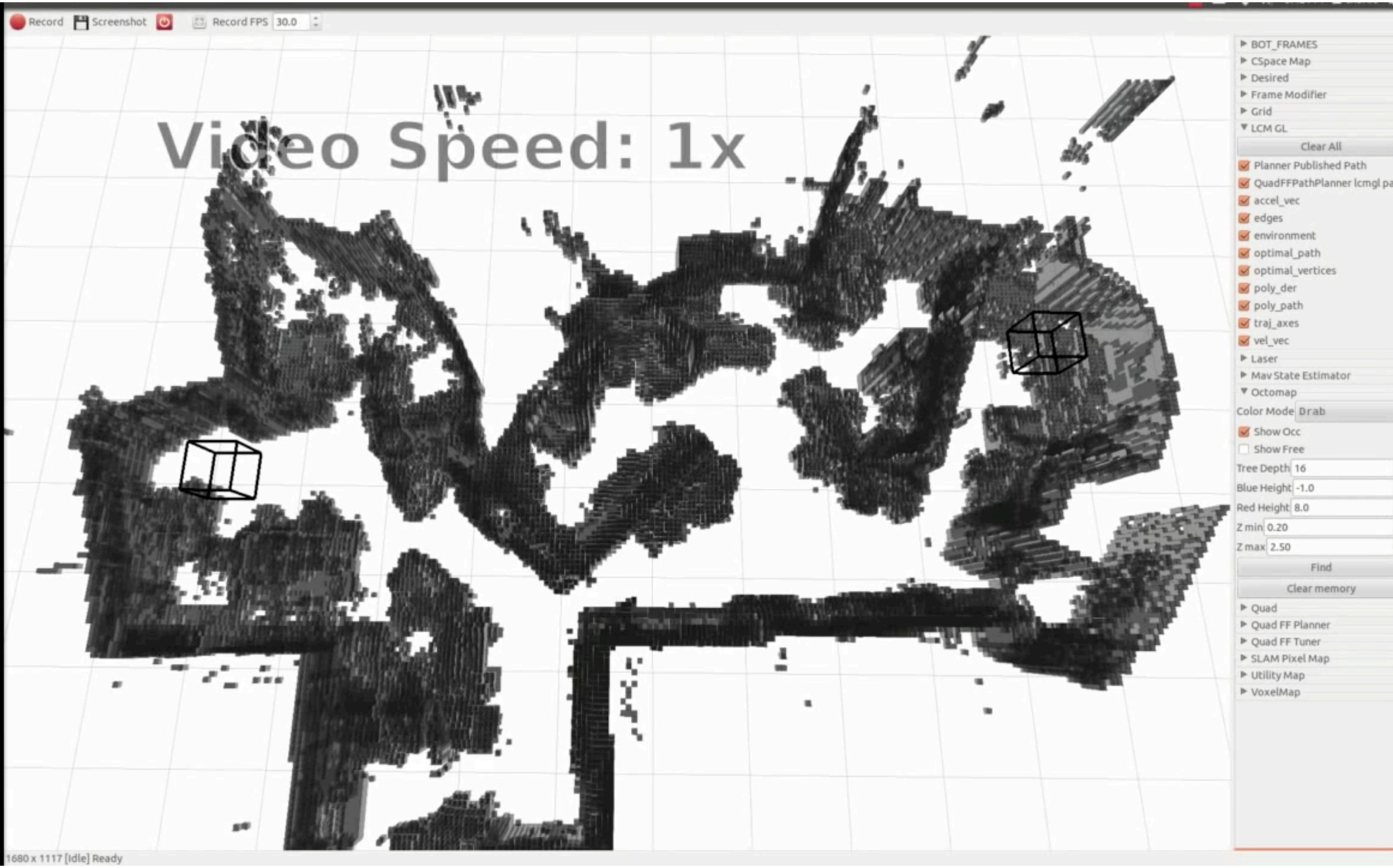
MIT AEROASTRO

Decoupled Trajectory Planning





Trajectory Optimization



2013

MIT AEROASTRO

Trajectory Optimization

2016-2017

Trajectory Optimization