16-350 Spring'17
Planning Techniques for Robotics

Introduction;
What is Planning for Robotics?

Maxim Likhachev

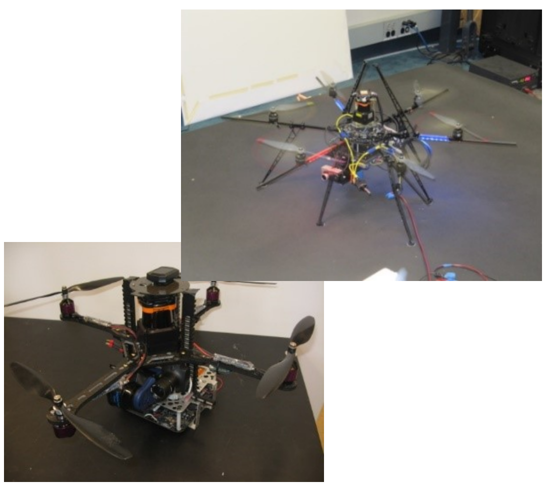
Robotics Institute

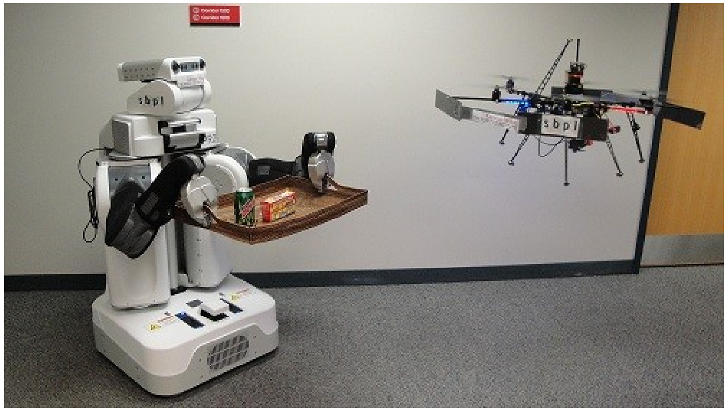
Carnegie Mellon University

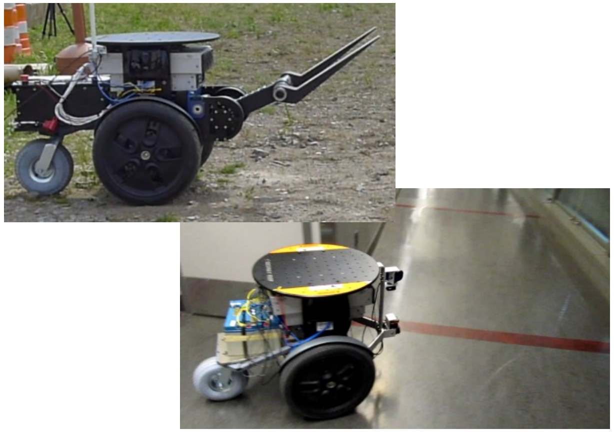
About Me

- My Research Interests:
  - Planning, Decision-making, Learning
  - Applications: planning for complex robotic systems including aerial and ground robots, manipulation platforms, small teams of heterogeneous robots

- More info: http://www.cs.cmu.edu/~maxim
- Search-based Planning Lab: http://www.sbpl.net







What is Planning?

- According to Wikipedia: *"Planning is the process of thinking about an organizing the activities required to achieve a desired goal."*

What is Planning **for Robotics**?

- According to Wikipedia: *"Planning is the process of thinking about an organizing the activities required to achieve a desired goal."*

- ***Given***
  - *model (states and actions) of the robot(s)* $M^R = \langle S^R, A^R \rangle$
  - *a model of the world* M^W
  - *current state of the robot* $s^R_{current}$
  - *current state of the world* $s^W_{current}$
  - *cost function C of robot actions*
  - *desired set of states for robot and world G*

- ***Compute a plan π that***
  - *prescribes a set of actions* $a_1, \ldots a_K$ *in* A^R *the robot should execute*
  - *reaches one of the desired states in G*
  - *(preferably) minimizes the cumulative cost of executing actions* $a_1, \ldots a_K$

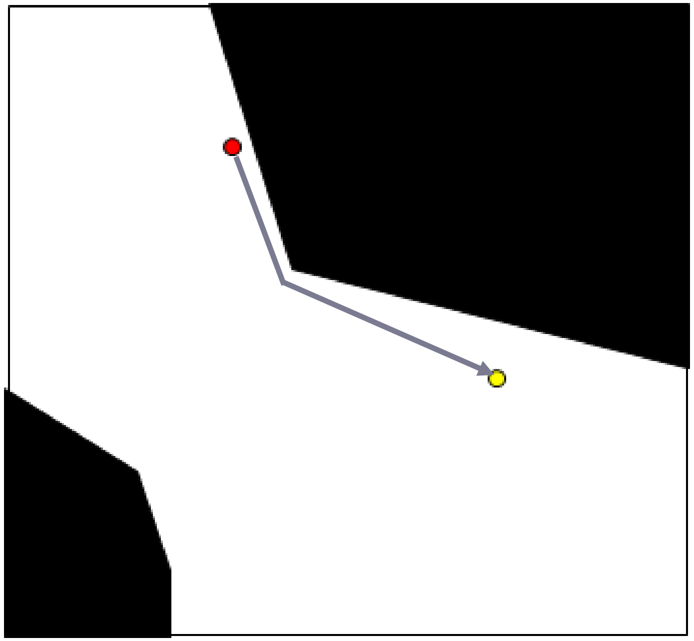
Few Examples

- ***Given***
  - *model (states and actions) of the robot(s)* $M^R = <S^R, A^R>$
  - *a model of the world* M^W
  - *current state of the robot* $s^R_{current}$
  - *current state of the world* $s^W_{current}$
  - *cost function C of robot actions*
  - *desired set of states for robot and world G*

- ***Compute a plan π that***
  - *prescribes a set of actions* $a_1,\ldots a_K$ *in* A^R *the robot should execute*
  - *reaches one of the desired states in G*
  - *(preferably) minimizes the cumulative cost of executing actions* $a_1,\ldots a_K$

Planning for omnidirectional robot:

What is M^R?
What is M^W?
What is $s^R_{current}$?
What is $s^W_{current}$?
What is C?
What is G?



Few Examples

- ***Given***
  - *model (states and actions) of the robot(s)* $M^R = \langle S^R, A^R \rangle$
  - *a model of the world* M^W
  - *current state of the robot* $s^R_{current}$
  - *current state of the world* $s^W_{current}$
  - *cost function C of robot actions*
  - *desired set of states for robot and world G*

- ***Compute a plan π that***
  - *prescribes a set of actions* $a_1, \ldots a_K$ *in* A^R *the robot should execute*
  - *reaches one of the desired states in G*
  - *(preferably) minimizes the cumulative cost of executing actions* $a_1, \ldots a_K$

Planning for omnidirectional drone:

What is M^R*?*
What is M^W*?*
What is $s^R_{current}$*?*
What is $s^W_{current}$*?*
What is C?
What is G?

MacAllister et al., 2013

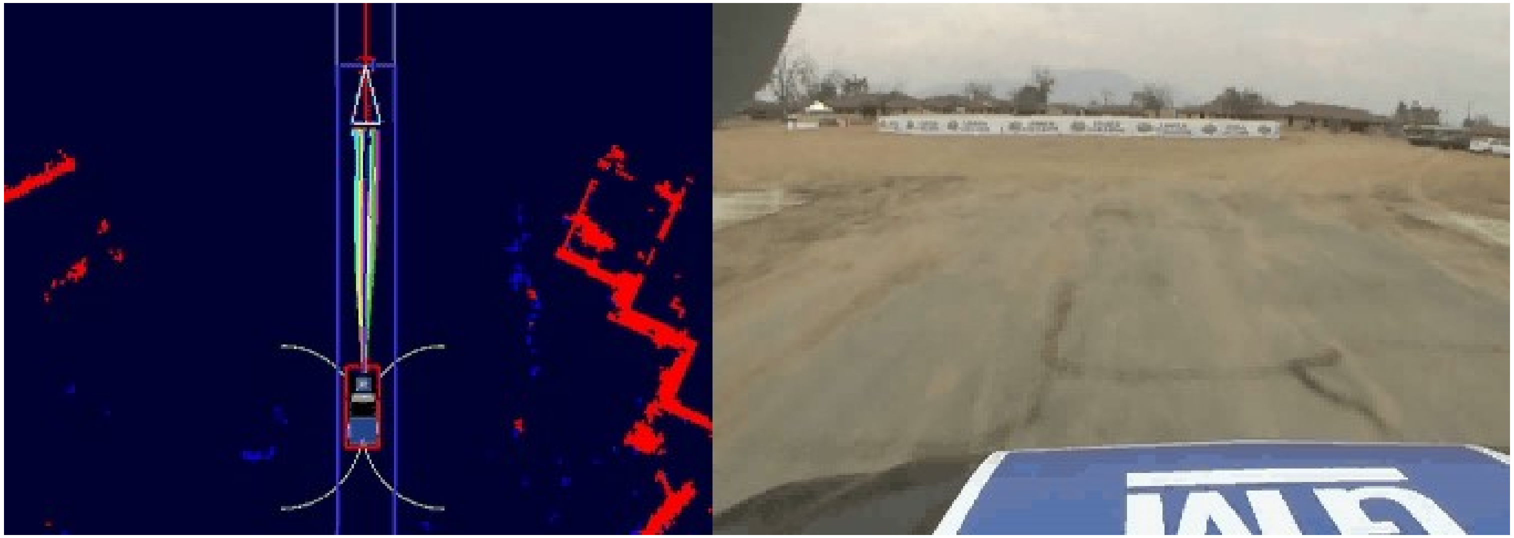
Few Examples

- ***Given***
  - *model (states and actions) of the robot(s)* $M^R = <S^R, A^R>$
  - *a model of the world* M^W
  - *current state of the robot* $s^R_{current}$
  - *current state of the world* $s^W_{current}$
  - *cost function C of robot actions*
  - *desired set of states for robot and world G*

- ***Compute a plan π that***
  - *prescribes a set of actions* $a_1, \ldots a_K$ *in* A^R *the robot should execute*
  - *reaches one of the desired states in G*
  - *(preferably) minimizes the cumulative cost of executing actions* $a_1, \ldots a_K$

Planning for autonomous navigation:

What is M^R*?*
What is M^W*?*
What is $s^R_{current}$*?*
What is $s^W_{current}$*?*
What is C?
What is G?



Likhachev & Ferguson, '09; part of Tartanracing team from CMU for the Urban Challenge 2007 race

Few Examples

- ***Given***
  - *model (states and actions) of the robot(s)* $M^R = <S^R, A^R>$
  - *a model of the world* M^W
  - *current state of the robot* $s^R_{current}$
  - *current state of the world* $s^W_{current}$
  - *cost function C of robot actions*
  - *desired set of states for robot and world G*

- ***Compute a plan π that***
  - *prescribes a set of actions* $a_1, \ldots a_K$ *in* A^R *the robot should execute*
  - *reaches one of the desired states in G*
  - *(preferably) minimizes the cumulative cost of executing actions* $a_1, \ldots a_K$

Planning for autonomous flight among people :

Narayanan et al., 2012

What is M^R*?*

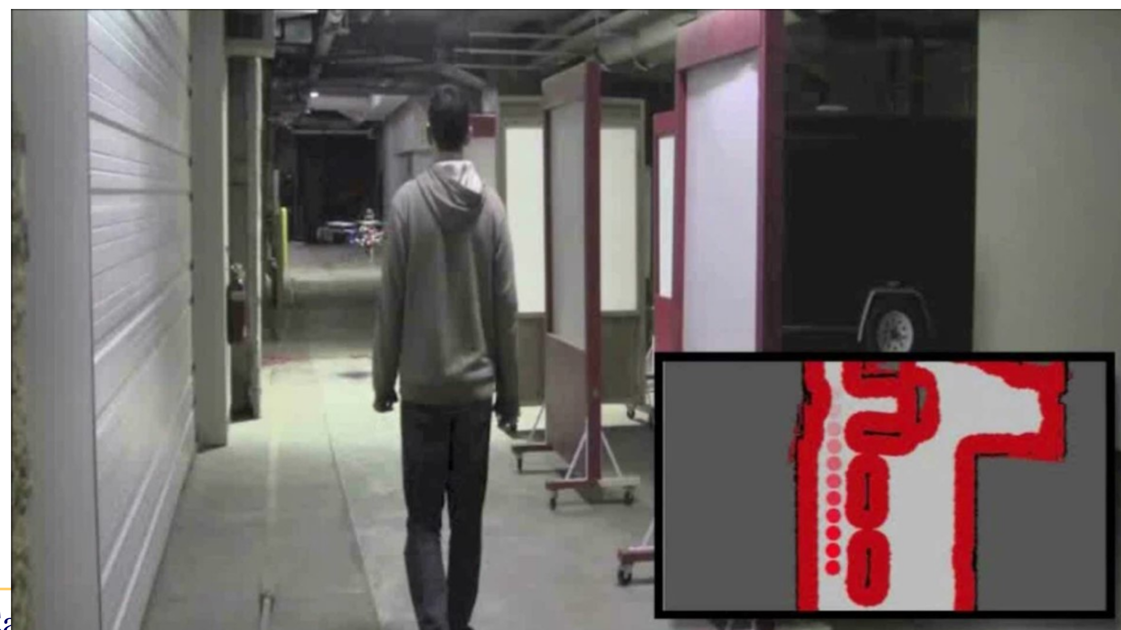
What is M^W*?*

What is $s^R_{current}$*?*

What is $s^W_{current}$*?*

What is C?

What is G?



Few Examples

- ***Given***
  - *model (states and actions) of the robot(s)* $M^R = <S^R, A^R>$
  - *a model of the world* M^W
  - *current state of the robot* $s^R_{current}$
  - *current state of the world* $s^W_{current}$
  - *cost function C of robot actions*
  - *desired set of states for robot and world G*

- ***Compute a plan π that***
  - *prescribes a set of actions* $a_1,\ldots a_K$ *in* A^R *the robot should execute*
  - *reaches one of the desired states in G*
  - *(preferably) minimizes the cumulative cost of executing actions* $a_1,\ldots a_K$

Planning for a mobile manipulator robot opening a door:

Gray et al., 2013

What is M^R*?*
What is M^W*?*
What is $s^R_{current}$*?*
What is $s^W_{current}$*?*
What is C?
What is G?

Few Examples

- ***Given***
  - *model (states and actions) of the robot(s)* $M^R = \langle S^R, A^R \rangle$
  - *a model of the world* M^W
  - *current state of the robot* $s^R_{current}$
  - *current state of the world* $s^W_{current}$
  - *cost function C of robot actions*
  - *desired set of states for robot and world G*

- ***Compute a plan π that***
  - *prescribes a set of actions* $a_1,\ldots a_K$ *in* A^R *the robot should execute*
  - *reaches one of the desired states in G*
  - *(preferably) minimizes the cumulative cost of executing actions* $a_1,\ldots a_K$

Planning for a mobile manipulator robot assembling a birdcage: *Cohen et al., 2015*

What is M^R*?*

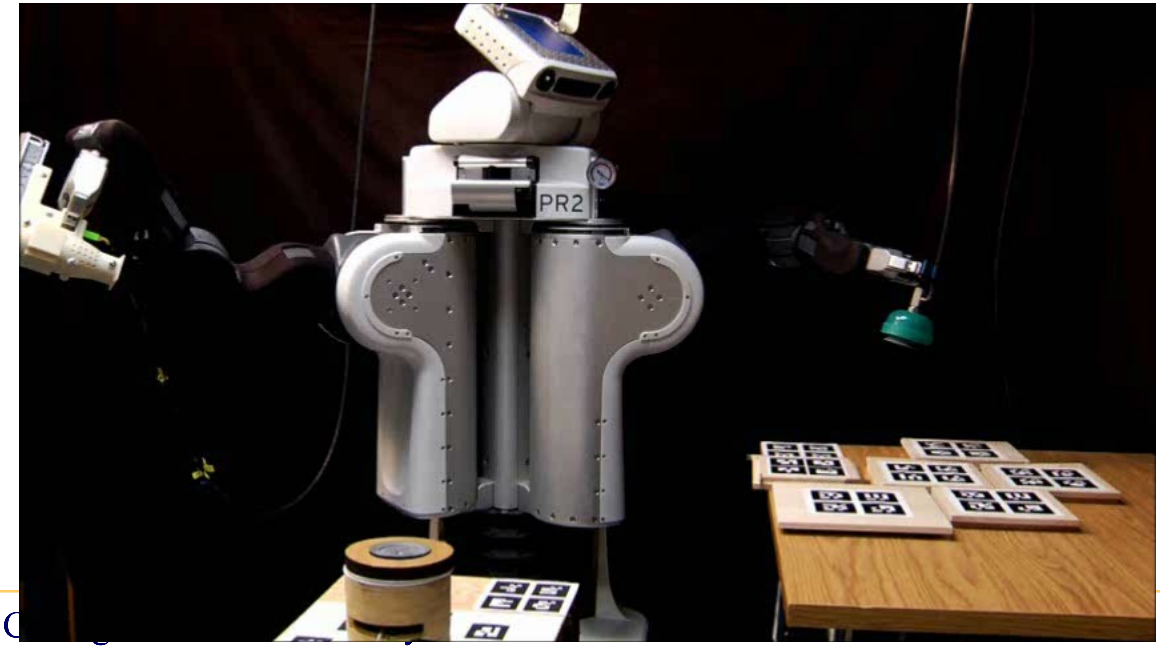
What is M^W*?*

What is $s^R_{current}$*?*

What is $s^W_{current}$*?*

What is C?

What is G?



Planning within a Typical Autonomy Architecture

Planning
What do I do next?

plan

Plan Execution/Controller
How do I do the next action?

commands

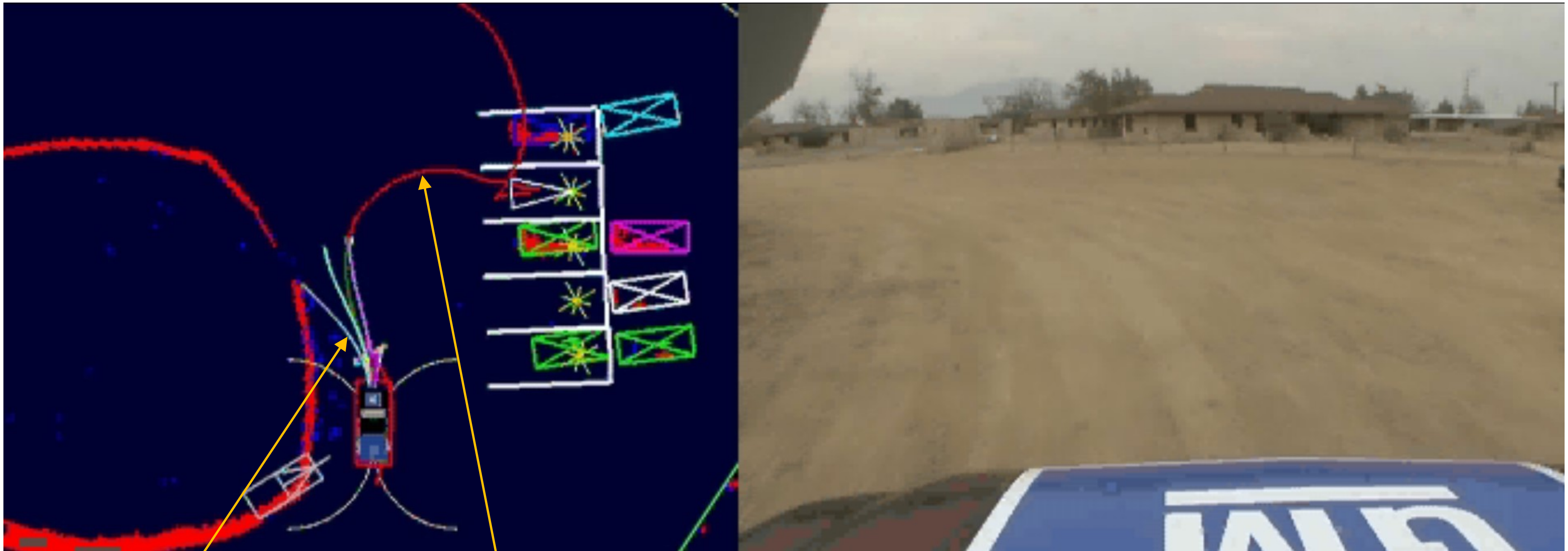
Perception
What do I see?

Localization
Where am I?

feedback from sensors

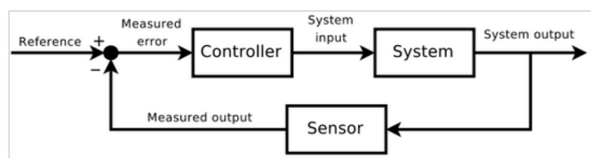
feedback from actuators

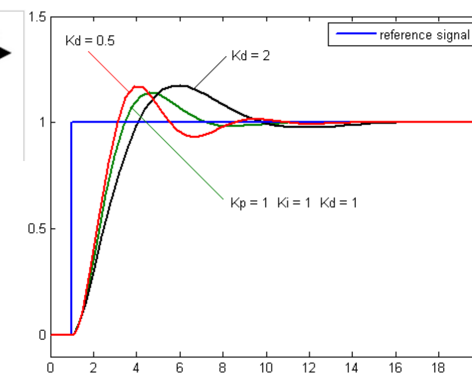
Planning vs. Trajectory Following vs. Control



local planning (trajectory following)

global planning





controller

Images from wikipedia

Class Logistics

- Instructor:

  Maxim Likhachev – maxim@cs.cmu.edu

- Website:

  http://www.cs.cmu.edu/~maxim/classes/robotplanning

- Mailing List for Announcements and Questions:

  I will set it up shortly

Class Logistics

- Books (optional):

- Planning Algorithms *by Steven M. LaValle*

- Heuristic Search, Theory and Applications *by Stefan Edelkamp and Stefan Schroedl*

- Principles of Robot Motion, Theory, Algorithms, and Implementations *by Howie Choset, Kevin M. Lynch, Seth Hutchinson, George A. Kantor, Wolfram Burgard, Lydia E. Kavraki and Sebastian Thrun*

- Artificial Intelligence: A Modern Approach *by Stuart Russell and Peter Norvig*

Class Prerequisites

- Knowledge of programming (e.g., C, C++)
- Knowledge of data structures
- Some prior exposure to robotics (e.g., Intro to Robotics class)

Class Objectives

- Understand and learn how to implement most popular planning algorithms in robotics including heuristic search-based planning algorithms, sampling-based planning algorithms, task planning, planning under uncertainty and multi-robot planning

- Learn basic principles behind the design of planning representations

- Understand core theoretical principles that many planning algorithms rely on and learn how to analyze theoretical properties of the algorithms

- Understand the challenges and basic approaches to interleaving planning and execution in robotic systems

- Learn common uses of planning in robotics

Tentative Class Schedule

Date	Day	Topic	HW out	HW due
17-Jan	Tue	Introduction; What is Planning?		
19-Jan	Thu	planning representations: grid-based graphs		
24-Jan	Tue	search algorithms: A*		
26-Jan	Thu	heuristics, weighted A*, Backward A*		
31-Jan	Tue	interleaving planning and execution: Anytime heuristic search	HW1	
2-Feb	Thu	interleaving planing and execution: Freespace assumption, Incremental heuristic search		
7-Feb	Tue	interleaving planning and execution: Limited Horizon search, LRTA*		
9-Feb	Thu	planning representations: lattice-based graphs, explicit vs. implicit graphs		
14-Feb	Tue	case study: planning for autonomous driving		
16-Feb	Thu	planning representations: PRM for continuous spaces		HW1
21-Feb	Tue	planning representations/search algorithms: RRT, RRT-Connect	HW2	
23-Feb	Thu	planning representations/search algorithms: RRT*		
28-Feb	Tue	case study: planning for mobile manipulation and articulated robots		
2-Mar	Thu	search algorithms: IDA*, Beam Search, Multi-goal A*		
7-Mar	Tue	case study: planning for exploration and surveillance tasks		
9-Mar	Thu	search algorithms: Markov Property, dependent vs. independent variables, Dominant Relationship	HW3	HW2
14-Mar	Tue	SPRING BREAK - NO CLASS		
16-Mar	Thu	SPRING BREAK - NO CLASS		
21-Mar	Tue	planning representations: state-space vs. symbolic representation for task planning		
23-Mar	Thu	search algorithms: symbolic task planning algorithms		
28-Mar	Tue	planning under uncertainty: Minimax formulation		
30-Mar	Thu	planning under uncertainty: Markov Decision Processes		HW3
4-Apr	Tue	planning under uncertainty: VI, RTDP for solving Markov Decision Processes		
6-Apr	Thu	final project proposals		
11-Apr	Tue	case study: planning for landing under uncertainty		
13-Apr	Thu	planning under uncertainty: Rewards version of Markov Decision Processes		
18-Apr	Tue	exam review		
20-Apr	Thu	SPRING CARNIVAL - NO CLASS		
25-Apr	Tue	exam		
27-Apr	Thu	multi-robot planning: centralized planning		
2-May	Tue	multi-robot planning: decentralized planning		
4-May	Thu	final project presentations		

Three Homeworks + Final Project

- All homeworks and the final project are individual (no groups)

- Homeworks are programming assignments based on the material

- Final project is a research-like project. For example:
  - to develop and implement a planner for a robot planning problem of your choice
  - to extend a particular planning algorithm to improve its running time or to handle additional conditions
  - to prove novel properties of a planning algorithm

Class Structure

- Grading

Three homeworks	33%
Exam	20%
In-class pop quizzes	10%
Final project	32%
Participation	5%

- Exam is tentatively scheduled for April 25 (no final exam)

- Late Policy
  - 3 free late days
  - No late days may be used for the final project!
  - Each additional late day will incur a 10% penalty

Questions about the class?