

16-782 Fall'17

Planning & Decision-making in Robotics

Introduction;

What is Planning, Role of Planning in Robots

Maxim Likhachev

Robotics Institute

Carnegie Mellon University

Class Logistics

- Instructor:

Maxim Likhachev – maxim@cs.cmu.edu

- TA:

Fahad Islam – fi@andrew.cmu.edu

- Website:

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

- Mailing List for Announcements and Questions:

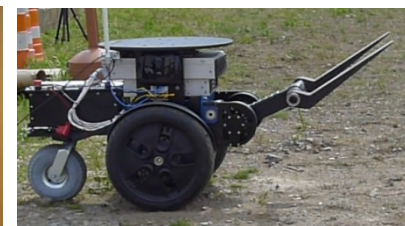
- 16782-pdr-request@lists.andrew.cmu.edu
- TA should have sent a “welcome” email to everyone

For those on the waitlist

- **Consider** taking the undergraduate (basic) version:
 - 16-350 in Spring'18
 - basic version of this course
 - Master students should be able to register for it
 - see syllabus from Spring'17:
<http://www.cs.cmu.edu/~maxim/classes/robotplanning>

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 and 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 $s_{current}^R$
- current state of the world $s_{current}^W$
- 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, \dots, 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, \dots, a_K

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 $s_{current}^R$
- current state of the world $s_{current}^W$
- 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, \dots, 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, \dots, a_K

Planning for omnidirectional robot:

What is M^R ?

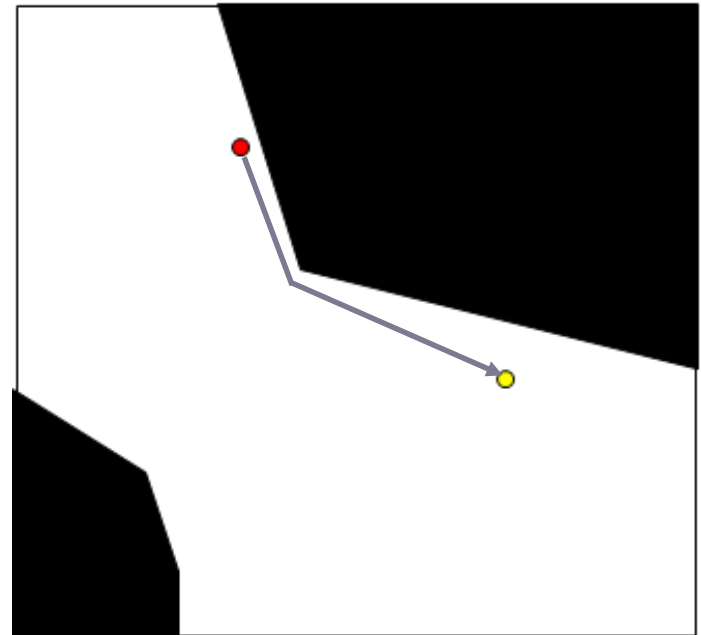
What is M^W ?

What is $s_{current}^R$?

What is $s_{current}^W$?

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 $s_{current}^R$
- current state of the world $s_{current}^W$
- 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, \dots, 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, \dots, a_K

Planning for omnidirectional drone:

What is M^R ?

What is M^W ?

What is $s_{current}^R$?

What is $s_{current}^W$?

What is C ?

What is G ?



MacAllister et al., 2013

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 $s_{current}^R$
- current state of the world $s_{current}^W$
- 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, \dots, 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, \dots, a_K

Planning for autonomous navigation:

What is M^R ?

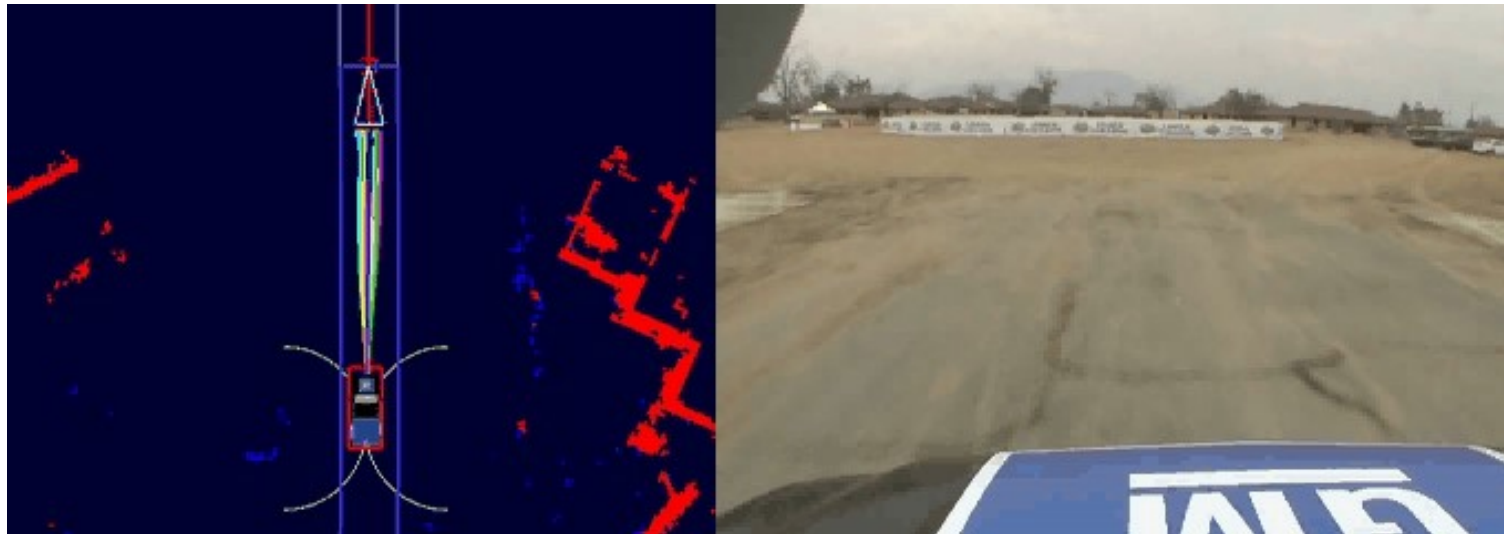
What is M^W ?

What is $s_{current}^R$?

What is $s_{current}^W$?

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) $M^R = \langle S^R, A^R \rangle$
- a model of the world M^W
- current state of the robot $s_{current}^R$
- current state of the world $s_{current}^W$
- 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, \dots, 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, \dots, a_K

Planning for autonomous flight among people :

Narayanan et al., 2012

What is M^R ?

What is M^W ?

What is $s_{current}^R$?

What is $s_{current}^W$?

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 $s_{current}^R$
- current state of the world $s_{current}^W$
- 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, \dots, 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, \dots, 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 $s_{current}^R$?

What is $s_{current}^W$?

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 $s_{current}^R$
- current state of the world $s_{current}^W$
- 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, \dots, 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, \dots, a_K

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

What is M^R ?

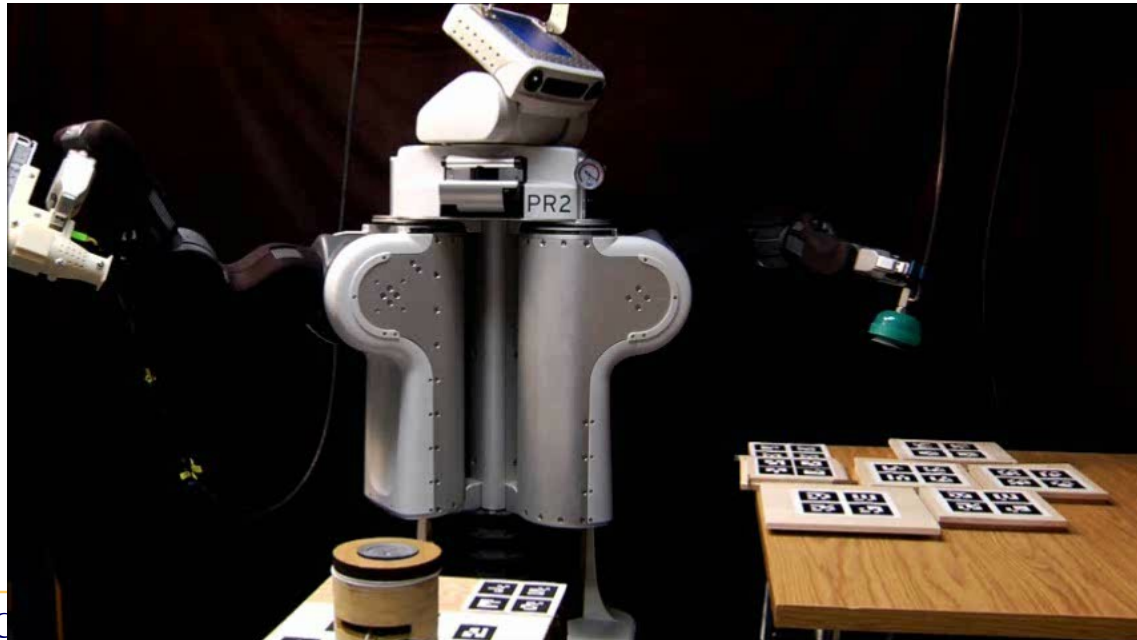
What is M^W ?

What is $s_{current}^R$?

What is $s_{current}^W$?

What is C ?

What is G ?



Assuming Infinite Computational Resources...

Where does Planning break?

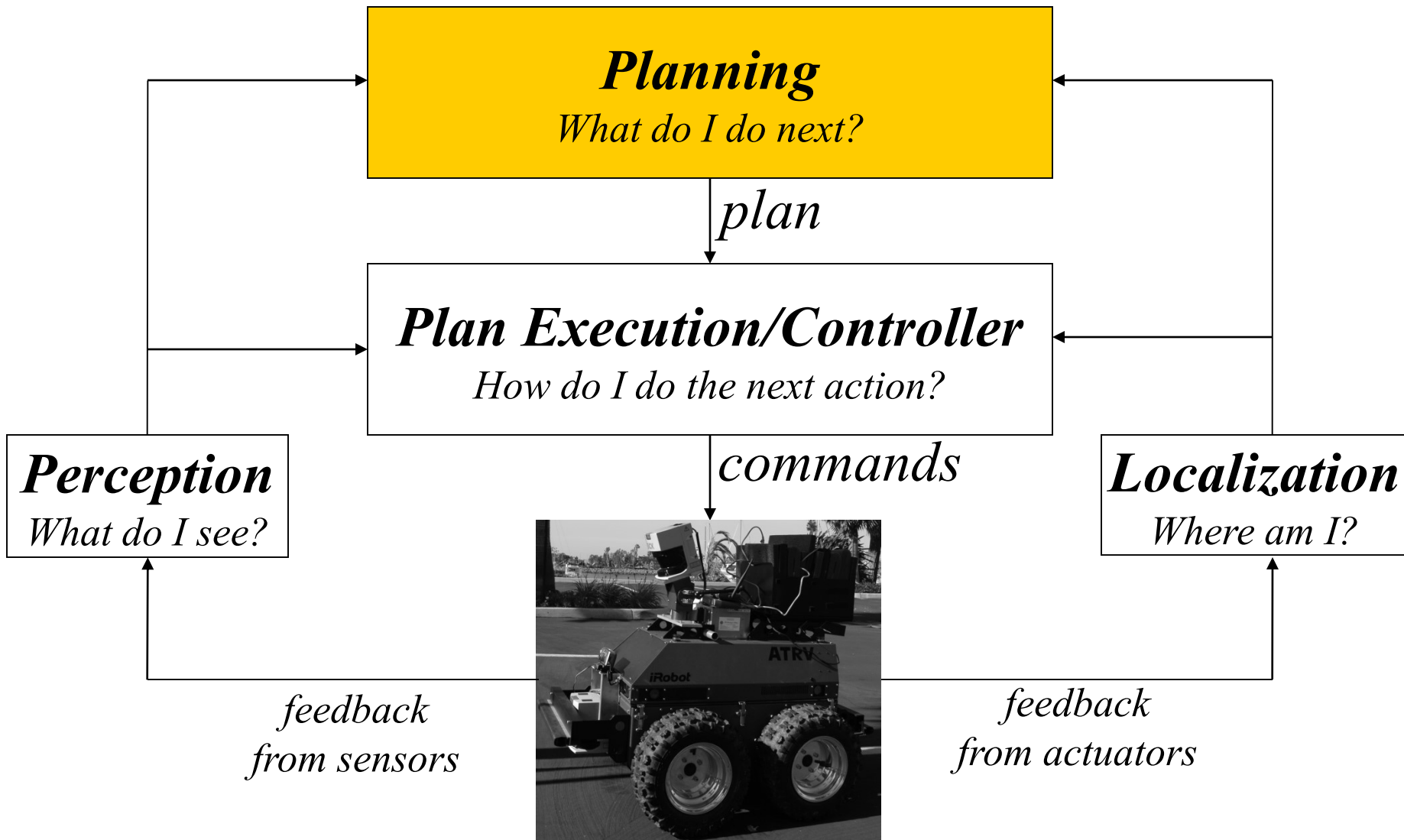
Assuming Infinite Computational Resources...

Where does Planning break?

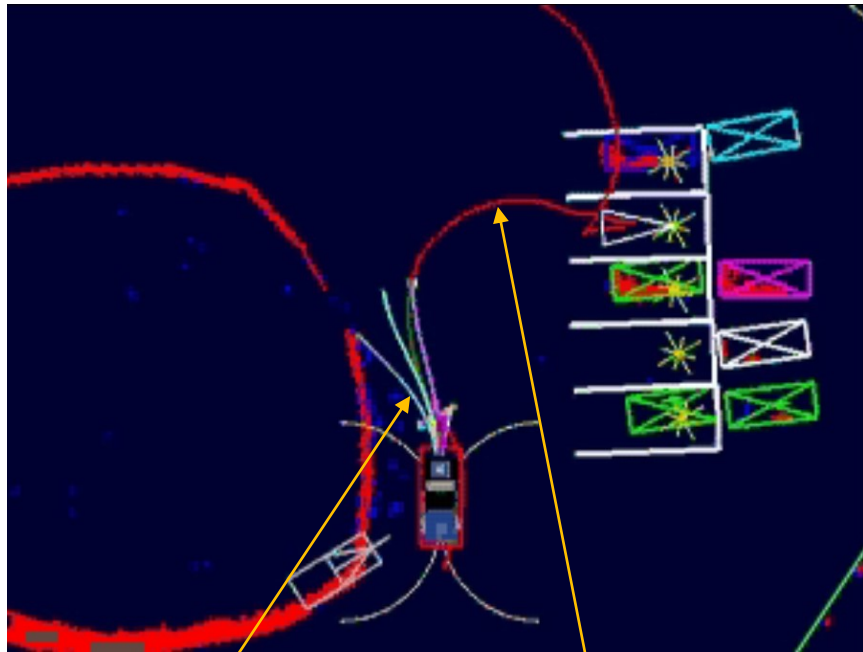
Reliance on the knowledge/accuracy of the model!

Role of Learning in Planning?

Planning within a Typical Autonomy Architecture

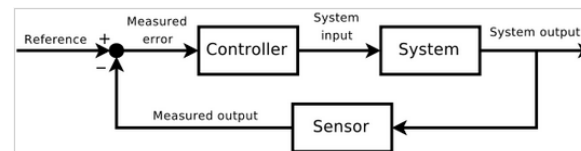


Planning vs. Trajectory Following vs. Control

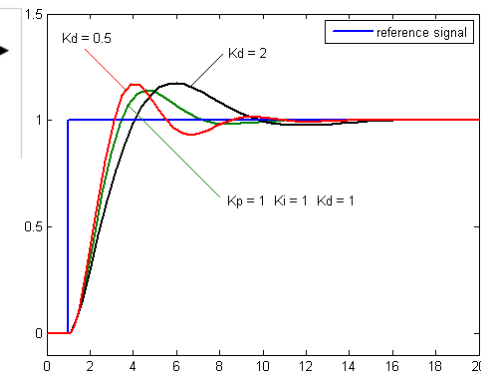


*local planning
(trajectory following)*

global planning



controller



Images from wikipedia

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++)
- Working knowledge of data structures & basic Computer Science algorithms (e.g., graphs, linked lists, priority queues, BFS/DFS, etc.)
- Prior exposure to robotics

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
28-Aug	Mon	Introduction; What is Planning?		
30-Aug	Wed	planning representations: explicit vs. implicit graphs, skeletonization, cell decomposition & lattice-based graphs		
4-Sep	Mon	LABOR DAY - NO CLASS		
6-Sep	Wed	search algorithms: A*, Backward A*, Weighted A*		
11-Sep	Mon	search algorithms: Heuristic functions, Multi-Heuristic A*	HW1	
13-Sep	Wed	interleaving planning and execution: Anytime heuristic search, Incremental heuristic search		
18-Sep	Mon	interleaving planning and execution: Freespace assumption, Limited Horizon search, LRTA*		
20-Sep	Wed	case study: planning for autonomous driving		
25-Sep	Mon	planning representations: PRM for continuous spaces		
27-Sep	Wed	planning representations/search algorithms: RRT, RRT-Connect		HW1
2-Oct	Mon	planning representations/search algorithms: RRT*	HW2	
4-Oct	Wed	case study: planning for mobile manipulation and articulated robots		
9-Oct	Mon	search algorithms: IDA*, Multi-goal A*		
11-Oct	Wed	case study: planning for exploration and surveillance tasks		
16-Oct	Mon	search algorithms: Markov Property, dependent vs. independent variables, Dominant Relationship		
18-Oct	Wed	planning representations: state-space vs. symbolic representation for task planning	HW3	HW2
23-Oct	Mon	search algorithms: symbolic task planning algorithms		
25-Oct	Wed	dealing with uncertainty: Planning sequence of behaviors, Planning with position uncertainty estimate		
30-Oct	Mon	planning under uncertainty: Minimax formulation, Minimax Backward A*		
1-Nov	Wed	planning under uncertainty: Markov Decision Processes, Value Iteration, RTDP		
6-Nov	Mon	planning under uncertainty: Partially-Observable Markov Decision Processes		HW3
8-Nov	Wed	planning under uncertainty: Partially-Observable Markov Decision Processes (cont'd)		
13-Nov	Mon	final project proposal presentations		
15-Nov	Wed	multi-robot planning		
20-Nov	Mon	multi-robot planning (cont'd)		
22-Nov	Wed	THANKSGIVING - NO CLASS		
27-Nov	Mon	exam review		
29-Nov	Wed	exam		
4-Dec	Mon	TBD		
6-Dec	Wed	final project presentations		

Three Homeworks + Final Project

- All homeworks are individual (no groups)
- Final projects is a group project (3-5 people per group)
- 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

Class Structure

- Grading

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

- Exam is tentatively scheduled for Nov. 29 (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?