16-350 Spring’23
Planning Techniques for Robotics

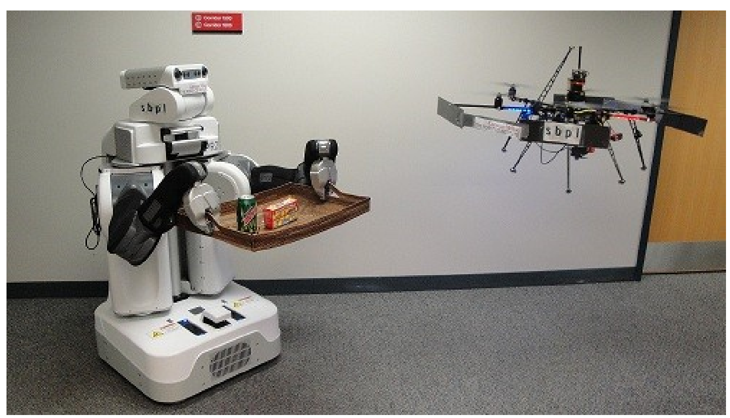
Introduction;
What is Planning for Robotics?

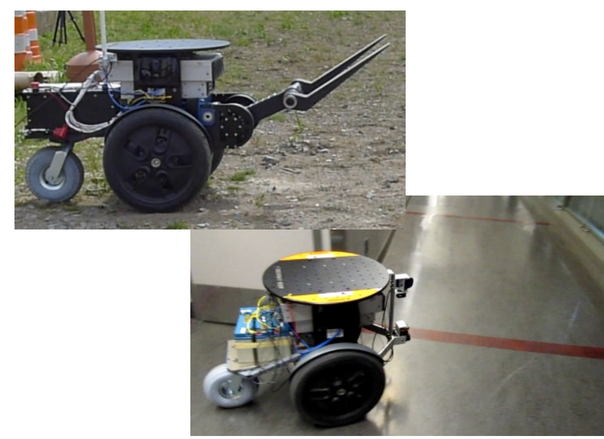
Maxim Likhachev
Robotics Institute
Carnegie Mellon University

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





About Me

- Also, currently split between CMU and Waymo, where I'm heavily involved in planning for self-driving vehicles

Class Logistics

- Instructor:

  Maxim Likhachev – maxim@cs.cmu.edu

- TA:

  Manuj Trehan – mtrehan@andrew.cmu.edu

- Website:

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

- Piazza for Announcements and Questions:

  You should have received an email

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) is preferred

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
18-Jan	Wed	Introduction; What is Planning?		
23-Jan	Mon	planning representations: skeletonization- and grid-based graphs, explicit vs. implicit graphs		
25-Jan	Wed	search algorithms: Uninformed A*		
30-Jan	Mon	search algorithms: A*, Multi-goal A*	HW1	
1-Feb	Wed	heuristics, weighted A*, Backward A*		
6-Feb	Mon	interleaving planning and execution: Anytime heuristic search		
8-Feb	Wed	interleaving planing and execution: Freespace assumption, Incremental heuristic search		
13-Feb	Mon	interleaving planning and execution: Limited Horizon search, LRTA*		HW1
15-Feb	Wed	planning representations: lattice-based graphs		
20-Feb	Mon	case study: planning for autonomous driving		
22-Feb	Wed	planning representations: PRM for continuous spaces	HW2	
27-Feb	Mon	planning representations/search algorithms: RRT, RRT-Connect, RRT*		
1-Mar	Wed	planning representations/search algorithms: RRT, RRT-Connect, RRT* (cont'd)		
6-Mar	Mon	SPRING BREAK; NO CLASSES		
8-Mar	Wed	SPRING BREAK; NO CLASSES		
13-Mar	Mon	case study: planning for mobile manipulation and articulated robots		
15-Mar	Wed	search algorithms: Markov Property, dependent vs. independent variables		HW2
20-Mar	Mon	case study: planning for exploration and surveillance tasks		
22-Mar	Wed	planning representations: state-space vs. symbolic representation for task planning	HW3	
27-Mar	Mon	search algorithms: symbolic task planning algorithms		
29-Mar	Wed	final project proposal presentations		
3-Apr	Mon	planning under uncertainty: Minimax formulation		HW3
5-Apr	Wed	planning under uncertainty: Expected Cost Minimization formulation		
10-Apr	Mon	planning under uncertainty: Solving Markov Decision Processes		
12-Apr	Wed	planning under uncertainty: Solving Markov Decision Processes (cont'd)		
17-Apr	Mon	exam		
19-Apr	Wed	multi-robot planning: centralized planning		
24-Apr	Mon	multi-robot planning: decentralized planning		
26-Apr	Wed	final project presentations		

Class Structure

- Grading

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

- Exam is tentatively scheduled for April 17 (no final exam)

- Late Policy
  - 3 free late days
  - No late days may be used for the final project!
  - Each additional late day incurs 10% penalty with 50% being the upper limit (grade of 90 becomes 81 for one additional late day)

Three Homeworks + Final Project

- All homeworks are individual (no groups)
- Final project are in groups of 2-3 students
- Homeworks are programming assignments
- Final project is a research-like project. For example:
  - to develop a planner for a robot planning problem of your choice
  - to extend an existing or develop a new planning algorithm
  - to prove novel properties of a planning algorithm

  - Get a feel for doing research: Individual meetings with groups, Two class presentations (initial idea and final)

Three Homeworks + Final Project

- Homework assignments for Masters students will have additional scope

- Undergraduate students will have an option to tackle this additional scope and receive bonus points

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)* $M^R = <S^R, A^R>$
  - *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* $a_1, \ldots a_K$

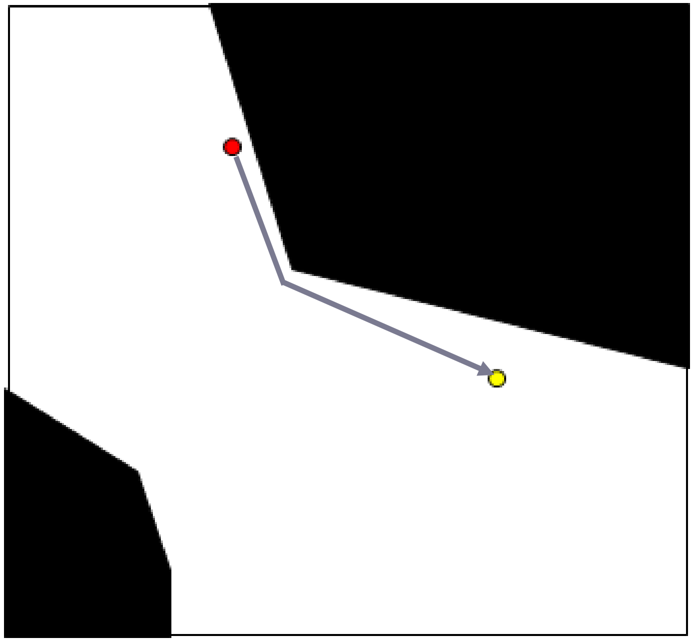
Example

- ***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*

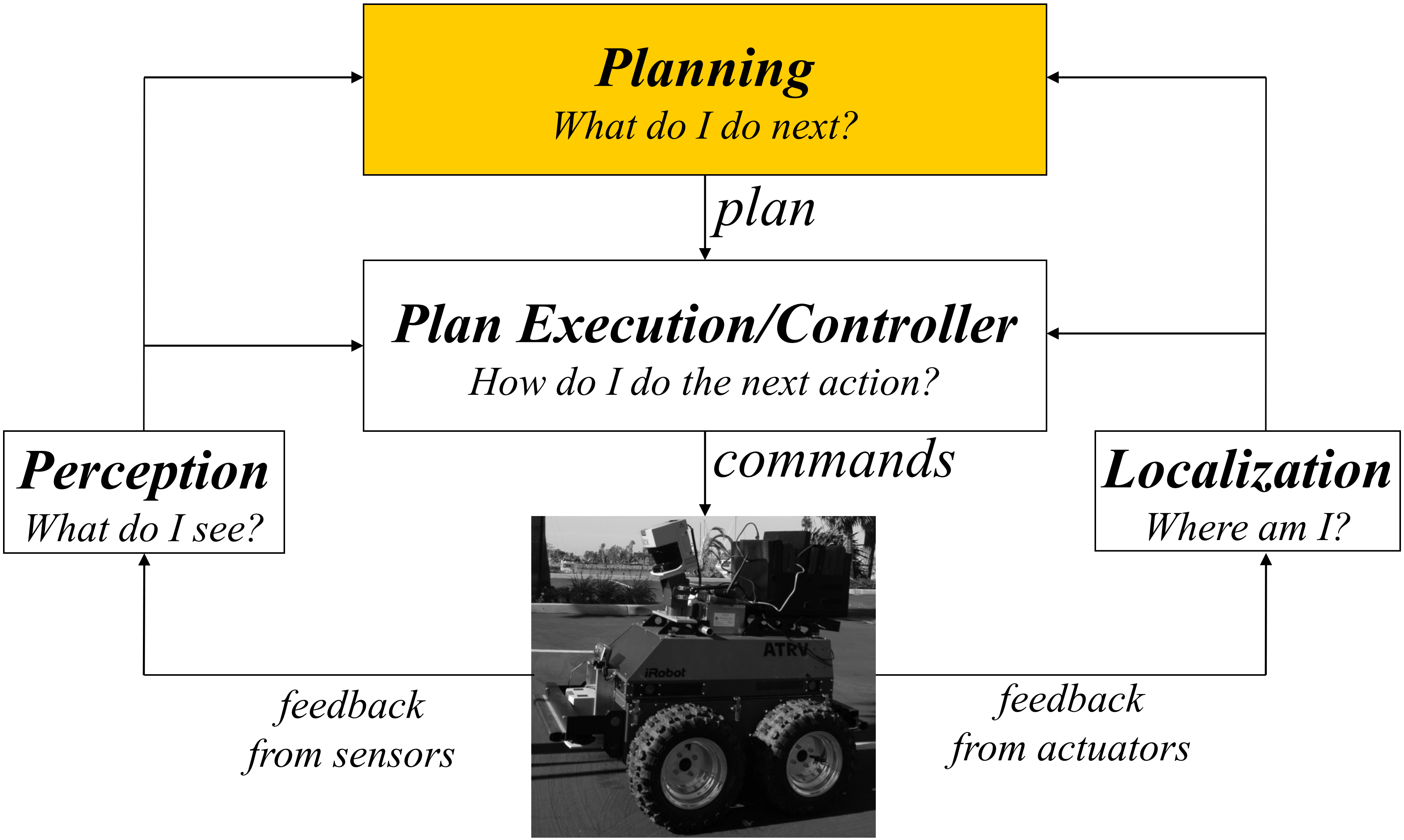
- ***Compute a plan π that***
  - *prescribes a set of actions* $a_1, \ldots a_K$ *in* A^R *the robot should execute*
  - *reaches one of the desired states in G*
  - *(preferably) minimizes the cumulative cost of executing actions* $a_1, \ldots a_K$

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?



Planning within a Typical Autonomy Architecture



Few More 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 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

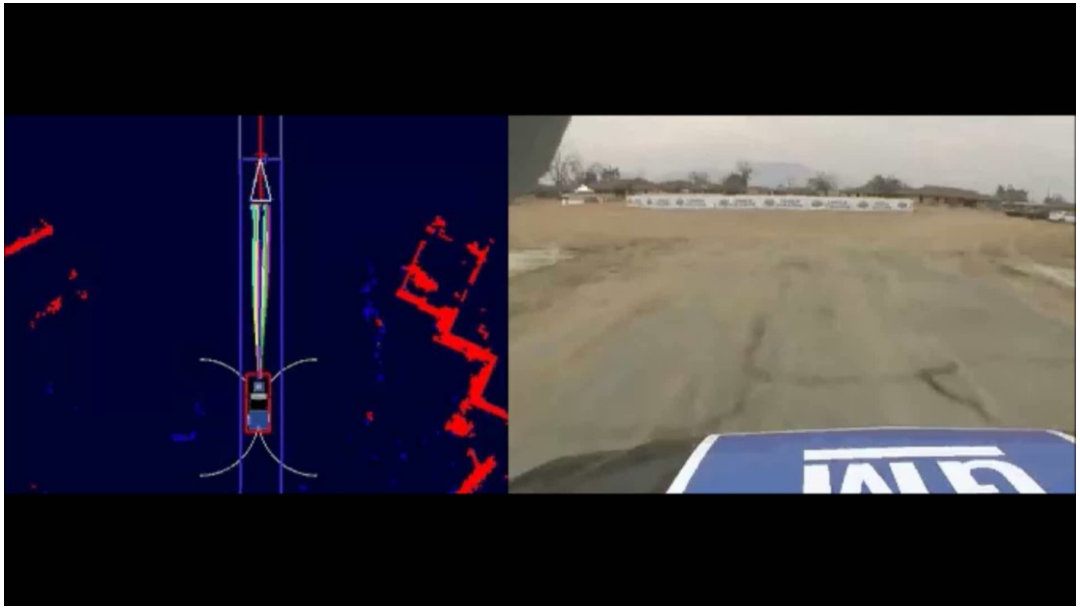
Few More Examples

- ***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* $a_1,\dots a_K$ *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 $s^R_{current}$*?*
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 More 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 More 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?

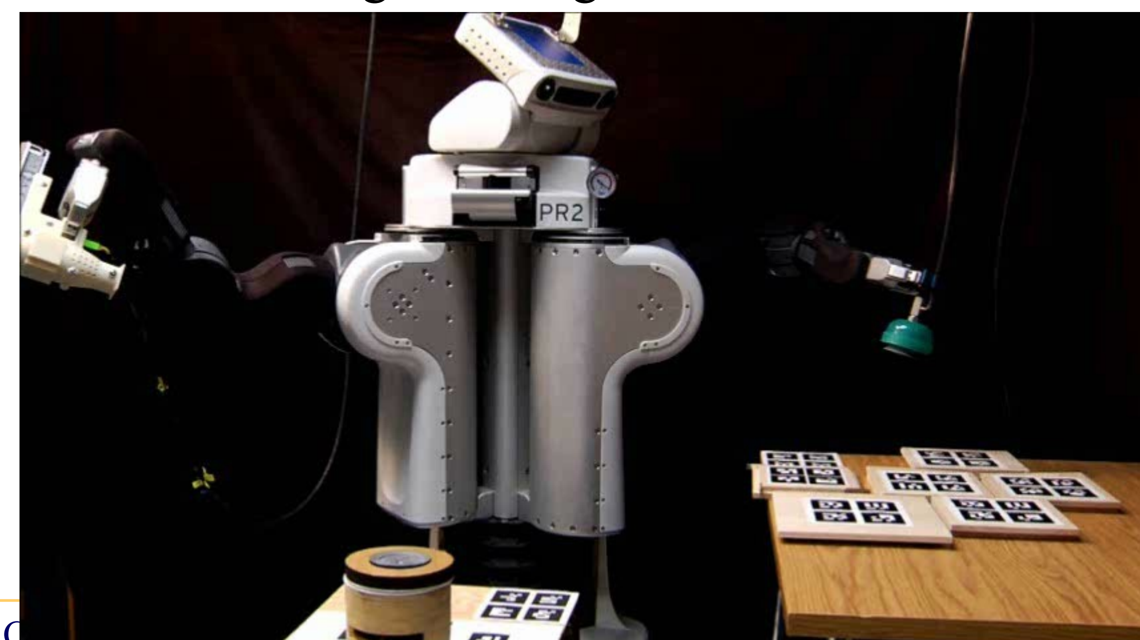
Few More Examples

- ***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 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?



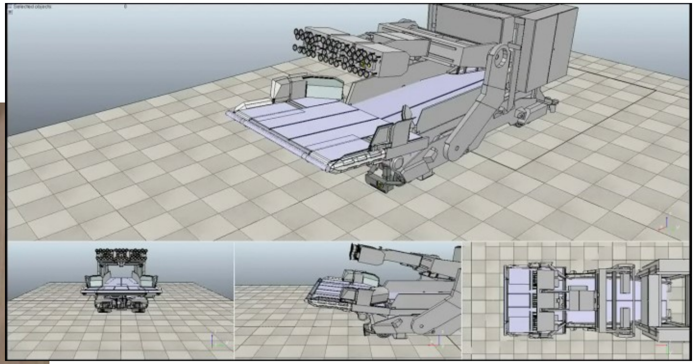
Few More Examples

- ***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 unloading a truck:

What is M^R*?*
What is M^W*?*
What is $s^R_{current}$*?*
What is $s^W_{current}$*?*
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 vs. Learning

Model-based approach

Learning models M^R, M^W and cost function C

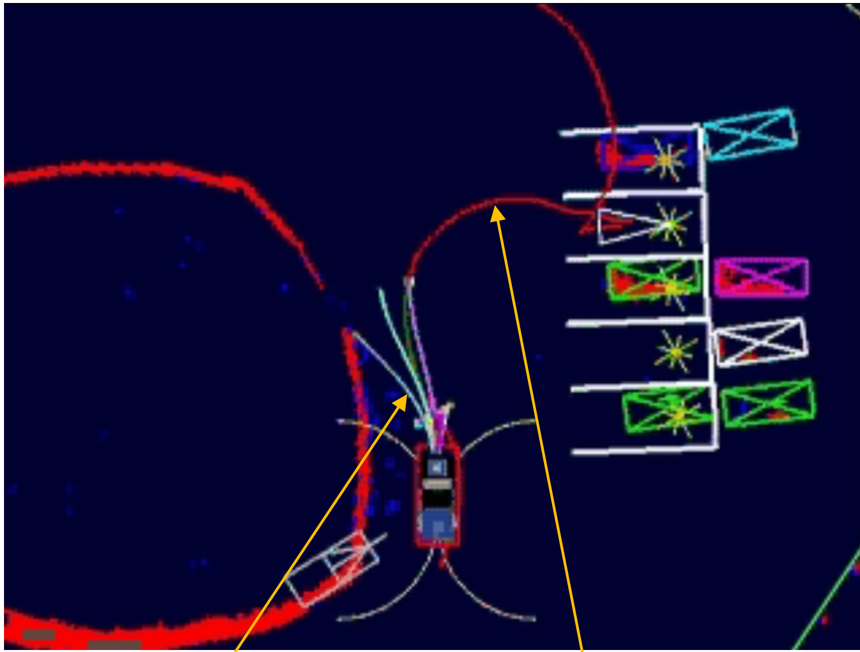
models M^R, M^W and cost function C →

Planning using models M^R, M^W and cost function C

Model-free approach

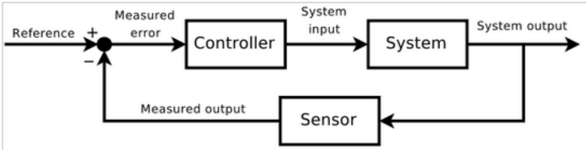
Learning the mapping from "what robot sees" onto "what to do next" using rewards received by the robot (Reinforcement Learning)

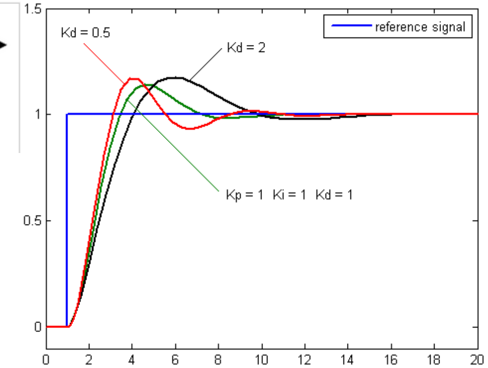
Planning vs. Trajectory Following vs. Control



local planning (trajectory following)

global planning





controller

Images from wikipedia

Questions about the class?