Skoltech

Skolkovo Institute of Science and Technology

L01: Introduction to Planning Algorithms

Planning Algorithms in AI and Robotics

Prof. Gonzalo Ferrer

Skoltech, 29 October 2021

Presentation: Who are we?

Instructor: Prof. Ferrer (name Gonzalo)

(g.ferrer@skoltech.ru)

Teaching Assistants: Aleksandr Gamaiunov

(aleksandr.gamaiunov@skoltech.ru)

Timur Akhtyamov

(timur.akhtyamov@skoltech.ru)

Mobile Robotics Lab: Path planning, Robot Navigation in dynamic environments, Pedestrian Motion prediction, Sensor fusion of Lidar, camera, IMU, etc., SLAM, Localization, Mapping, etc

https://sites.skoltech.ru/mobilerobotics/

What is planning? Robotics

Robot converts high-level specification of tasks into low-level descriptions of how to move.

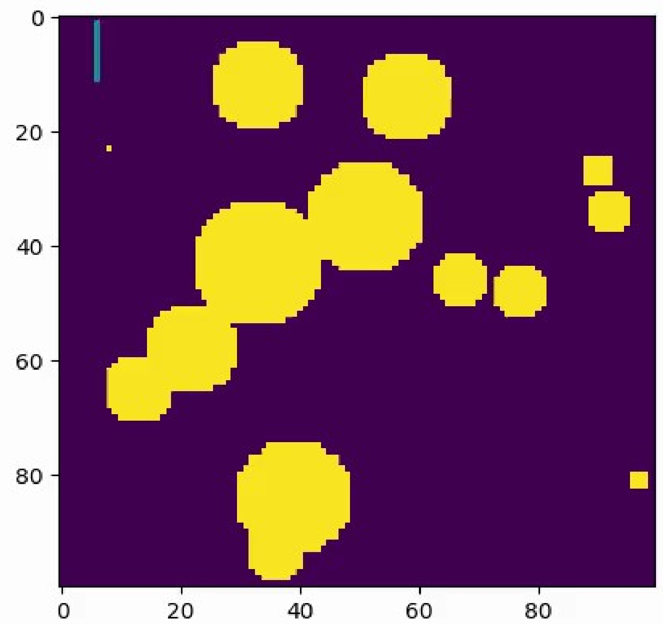
Mostly finding a plan is known as **motion planning** or **path planning**.

Example: The *piano movers problem*: how to move one piano from one room to the next room.

What is planning? Robotics

Other examples, moving an object from a starting configuration to goal configuration while avoiding obstacles.

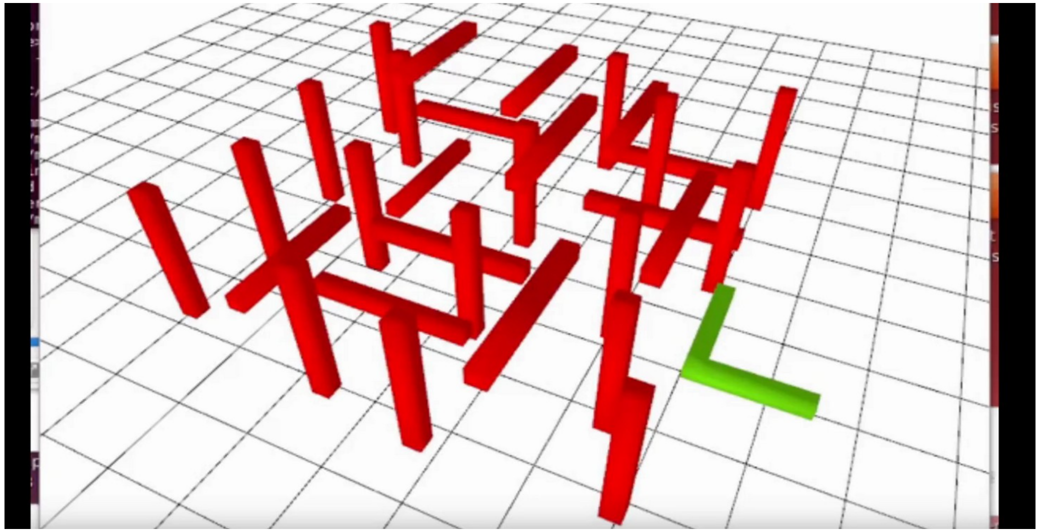
On this first view we only consider **feasibility**, although later we will consider optimality and uncertainty.



What is planning? Robotics

Now, image the "piano" as the green rod in this 3D maze.

Dynamics will not be covered on this course, although it is a natural extension to planning (trajectory planning).

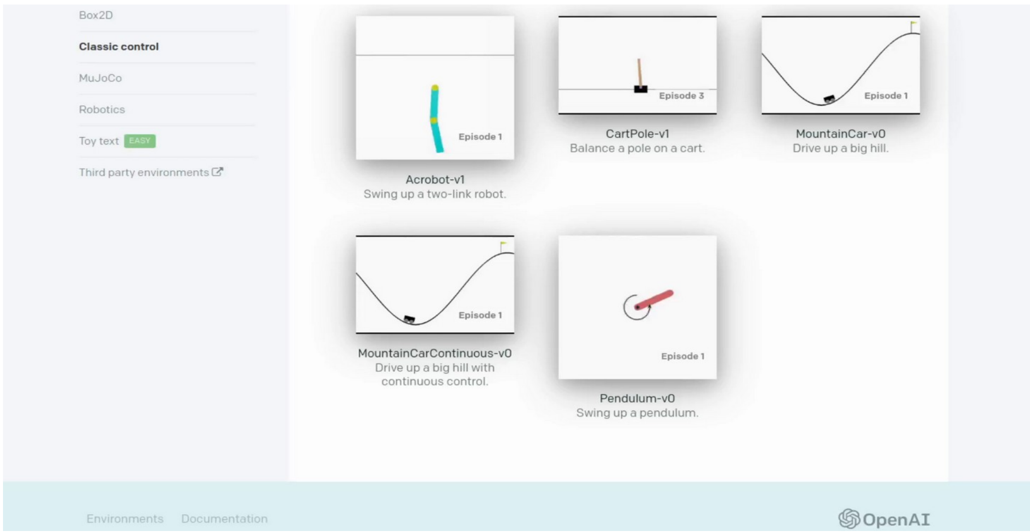


What is planning? Control Theory

Control Theory typically considers physical systems and differential constrains.

The controller designs **feedback policies** or **control laws**.

Controls usually focuses on optimality and stability.



What is planning? Artificial Intelligence

Agents or decision-makers in AI do planning or **Problem Solving.**

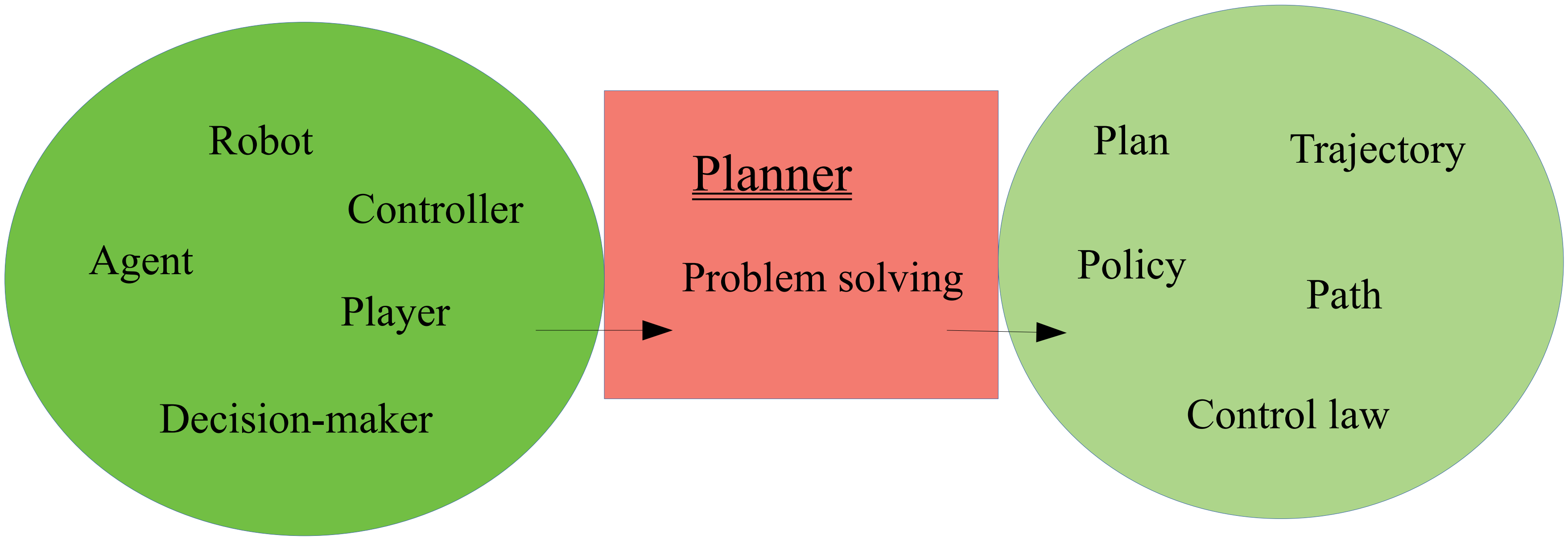
Usually discrete spaces, leading to combinatorial solutions.

Example: Solving the Rubik’s cube.

Planning to plan

Planning means different things in different disciplines: robotics, artificial intelligence and control theory.

In this course we will present a unified view:



This view is from LaValle's book, Ch.1

The ingredients of planning

- **State**: All possible situations that could arise.
- **Action**: Change the states. Also known as *inputs, controls, operations*, etc.
- **Initial and Goal states**: A planning problem involves these 2 states.
- **Criterion**: The desired outcome of a plan:
  - Feasibility: arrival at a goal state, regardless of efficiency.
  - Optimality: Find a feasible plan and optimize some function.
- **Plan**: A specific strategy of behavior. It could be a sequence, a policy, etc.

The ingredients of planning

Example: the Piano movers problem.

State: 3D poses — Actions: 3D (relative) poses

Initial state: the corridor — Goal: my office

Planner: RRT (L04 algorithm) — Plan: sequence of actions

Example: How to arrange the furniture in your new house

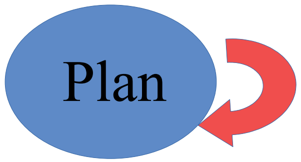
State: Position of all objects — Actions: Move objects

Initial state: Empty space — Goal: Ikea picture

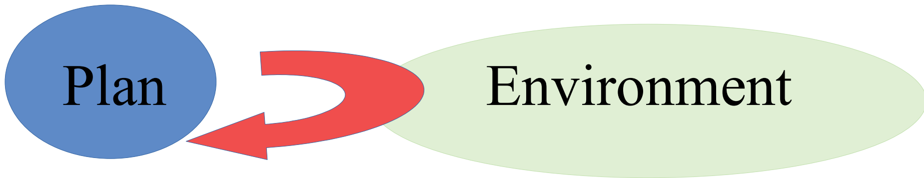
Planner: Yourself — Plan: sequence of actions.

Taxonomy of Planning

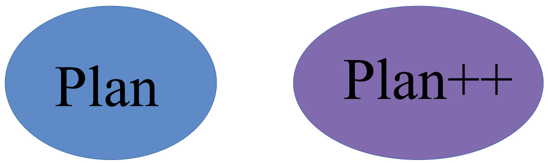
There are different ways to use the plan calculated by the planner



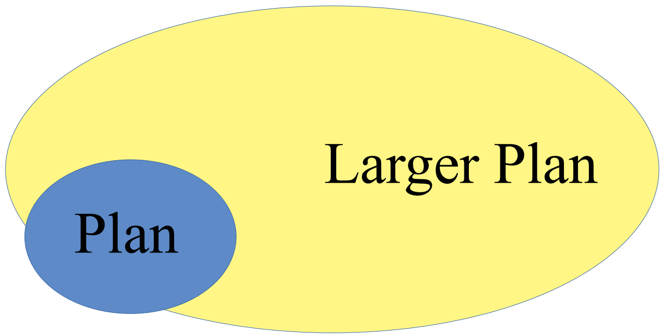
Open loop execution



Feedback



Refinement



Hierarchical

Course Goals

Mastering a set of core planning algorithms, covering a wide variety of planning problems, such as discrete planning, continuous planning, decision-making, planning under uncertainty, learning-based, etc.

For any give task, select the right planning algorithm.

Prerequisites

Basic programming skills in Python

Probability, introductory course.

Next steps -> Perception in robotics in T3

https://github.com/MobileRoboticsSkoltech/Perception-in-Robotics-course-T3-2021-Skoltech

Learning Outcomes

- The student will acquire **theoretical knowledge** and a rich set of **practical skills** to design their own planning and decision-making algorithms, applied to any kind of problem related to AI, Robotics, etc.
- The student will be able to **analyze** problems under the perspective of planning, and provide a more diverse set of tools for problem solving.
- The student will get **experience** on different planning techniques such as discrete planning, continuous planning, decision-making, planning under uncertainty, learning-based. We will prepare materials, seminars and problem sets that will serve as a first step into each of these particular topics.

Course Structure

10 lectures (Online)	Monday 16:00-18:00 Friday 16:00-18:00
70% Problem Sets	PS1: Discrete planning PS2: Sampling-based planning PS3: Value Iteration PS4: Decision-Making
30% Final Group Project	(more in canvas and later in class)

Course summary:

Date	Details	Due
Fri, 29 Oct 2021	L1: Introduction	0:00
Mon, 1 Nov 2021	L2: Discrete Planning	0:00
Mon, 8 Nov 2021	L3: Configuration Space	0:00
Fri, 12 Nov 2021	Seminar 1: Distances	0:00
Sun, 14 Nov 2021	PS1: Discrete Planning	due by 23:59
Mon, 15 Nov 2021	L4: Sampling-Based Planning	0:00
Fri, 19 Nov 2021	Seminar 2: Sampling	0:00
Sun, 21 Nov 2021	PS2: Sampling-based planning	due by 23:59
Mon, 22 Nov 2021	L5: Discrete Optimal Planning	0:00
Fri, 26 Nov 2021	L6: Continuous Optimal Planning	0:00
Sun, 28 Nov 2021	PS3: Value Iteration	due by 23:59
Mon, 29 Nov 2021	L7: Decision Making and Games	0:00
Fri, 3 Dec 2021	L8: Markov Decision Process	0:00
Sun, 5 Dec 2021	PS4: Decision making	due by 23:59
Mon, 6 Dec 2021	Seminar 3: Reinforcement Learning	0:00
Fri, 10 Dec 2021	Seminar 4: General course discussion	0:00
Mon, 20 Dec 2021	Final Project Presentation	0:00

Course Material

- Lecture slides with annotations, uploaded after every class.
- Videos from lectures, upload to canvas/youtube channel.
- Books:
  - S. LaValle, "Planning Algorithms". Cambridge university press, 2006
  - Sutton and Barto "Reinforcement Learning: an Introduction", MIT press 2018
- Canvas, selected papers
- Telegram channel

Class Structure

Lectures will be imparted online via Zoom/offline, blocks of 45 minutes.

Students are encouraged to participate and discuss in class via microphone or chat.

Recommendations:

Participate as much as possible in class!!

Make questions, be engaging, learn more (even in remote mode)

Problem Sets

- Problem Sets (PS) will be written in Python
- PSs are substantial work and should be worked on during the full allotted time period (each is a 17.5% of your grade).
- There will be a penalty in the max possible grade of -1%/hour for the first 3 hours and then a penalty of -20%/day up to 50% of the grade.

  Grade = min (your grade, max possible grade)
- Late submission is based on the last update to canvas.
- Students are encouraged to discuss on PS. Copying code is forbidden. On every PS there will be a section dedicated to Acknowledgments, if any.
- All PS's must be submitted in order to pass the course.

Course Policies

Attendance

Attendance is highly recommended. In the online format of the class, this will be your chance to ask, discuss and learn as we progress. Later it will be too late.

PS Regrade Policy

If you believe we graded a problem-set or an exam of yours incorrectly, you can submit a regrade request no later than one week after the graded work is originally returned.

Academic Integrity

Reference to Skoltech's policy (see canvas)

Final Group Project

- Topic (related to the course): Extend a state of the art algorithm, or paper reproduction or implementation on your own settings. We will upload to canvas a list of selected papers for inspiration.
- 3-5 students / group
- Proposal (December 8): a one page document.

  (ask for feedback from the course team)
- Presentation (December 20) 12’+3’ questions
- Final project document, on an IEEE template.

Past projects

Rapidly-exploring Random Belief Trees for Motion Planning Under Uncertainty

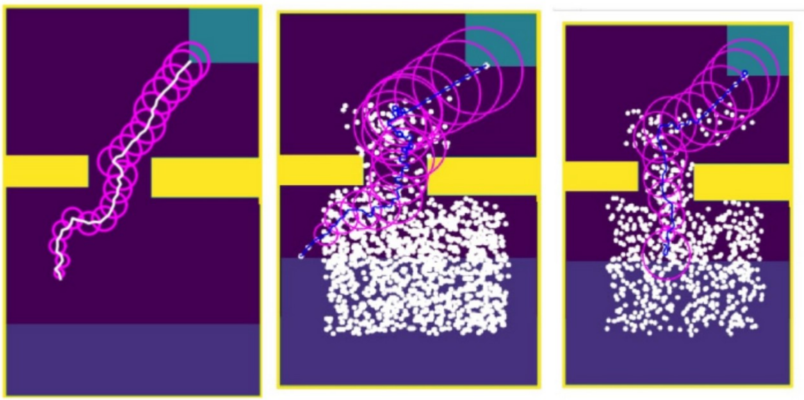


Fig. 4. Preliminary experimental RRBT results with various hyperparameters (size of covariance matrix, num of vertices, initial and goal points.

Bringing the learning to classic motion planning

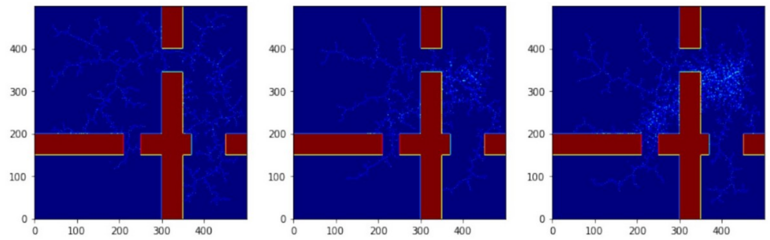


Figure 5: Observed area by RRT applied to learned samples with 0.1,0.5,0.7 probability

Past projects

Evader-pursuer zero-sum game by MCTS

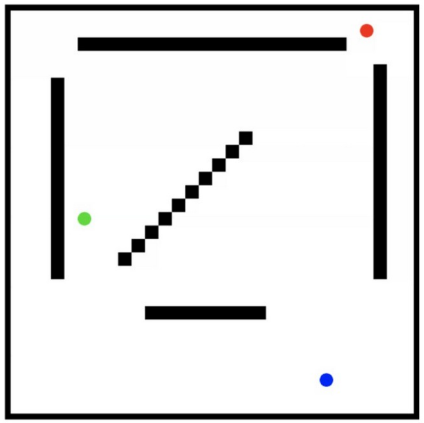


Fig. 2. Environment. Red circle corresponds to the pursuer position, blue to the evader position, green to the goal and black regions correspond to the walls.

Comparison of different algorithms for Pacman environment

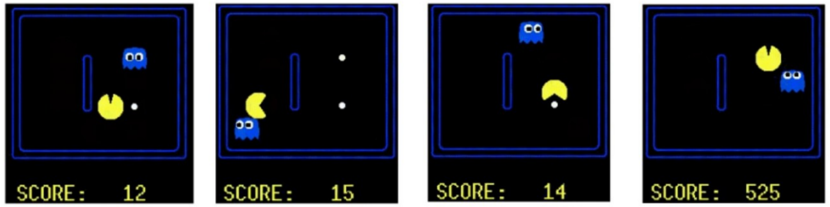


Fig. 6. Experiments with Q-learning on small environment