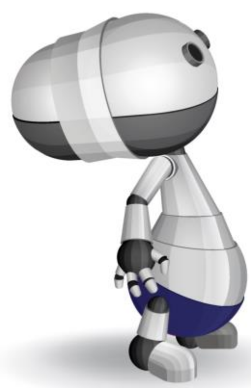
Algorithmic Human-Robot Interaction

Course Content

We will study fundamental topics in robotics and psychology/cognitive science with the objective of introducing robustness to human interaction to the former and automation to the latter.



Expected Background

CS Foundations

- Programming (Python or C++)
- Networking
- Systems-building

Presentation and Written Skills

- LaTeX

<u>Bonus Skills</u>

- Debugging (e.g., pdb or gdb)
- Artificial Intelligence / Machine Learning

Course Content

- Task and Motion Planning
  - Motion Planning
  - Trajectory Optimization
  - Task Planning
  - Human-aware Motion Planning
- Learning from Demonstration
  - Keyframing / Kinesthetic Teaching
  - Imitation Learning
  - Social Scaffolding
- Intent Recognition/Projection
  - Human motion modeling
  - Shared Autonomy
  - Non-verbal Behaviors
    (Gaze, Deictic Gesture)
- Explainable AI
  - Course of Action Justification
  - Anticipatory Explanation
- Coordination
  - Theory of mind
  - Task Modeling
- Communication
  - Requesting assistance
  - Synchronizing Mental Models

Course Format

Paper Presentations:

- Each lecture with assigned reading will include two student presentations per paper-

  1 PRO (10 min) and **1 CON** (5 min)

- Presentations can be done via whiteboard or PowerPoint but are expected to be of high quality

Quizzes and Homeworks:

- A short quiz will be administered at the start of lecture to test high level knowledge and conceptual understanding for assigned readings. They will be graded +, ✓, or -.
- Programming assignments will be given to reinforce algorithmic understanding and familiarity with ROS (Robot Operating System)

Final Project:

- A semester long team-based research effort tackling an unsolved problem in Algorithmic HRI

Final Projects

Final projects will culminate in a research paper suitable for publication at the workshop or symposium level, such as the AI-HRI Symposium at the AAAI Fall Symposium Series (AAAI-FSS)

Project Details

Group Project – Teams of 2 – 4

Explore a problem area in depth
- Theoretical and empirical understanding to create generalizable knowledge

Design and conduct experimental studies

Formal writeup of project and results

Final product:

4-6 page research paper
Publishable at AI-HRI AAAI Fall Symposium

Grading

Class Participation – 10%

Attend regularly, engage in discussions!

Paper Presentations – 30%

Make presentations worth your peers’ time!

Quizzes and Homework – 20%

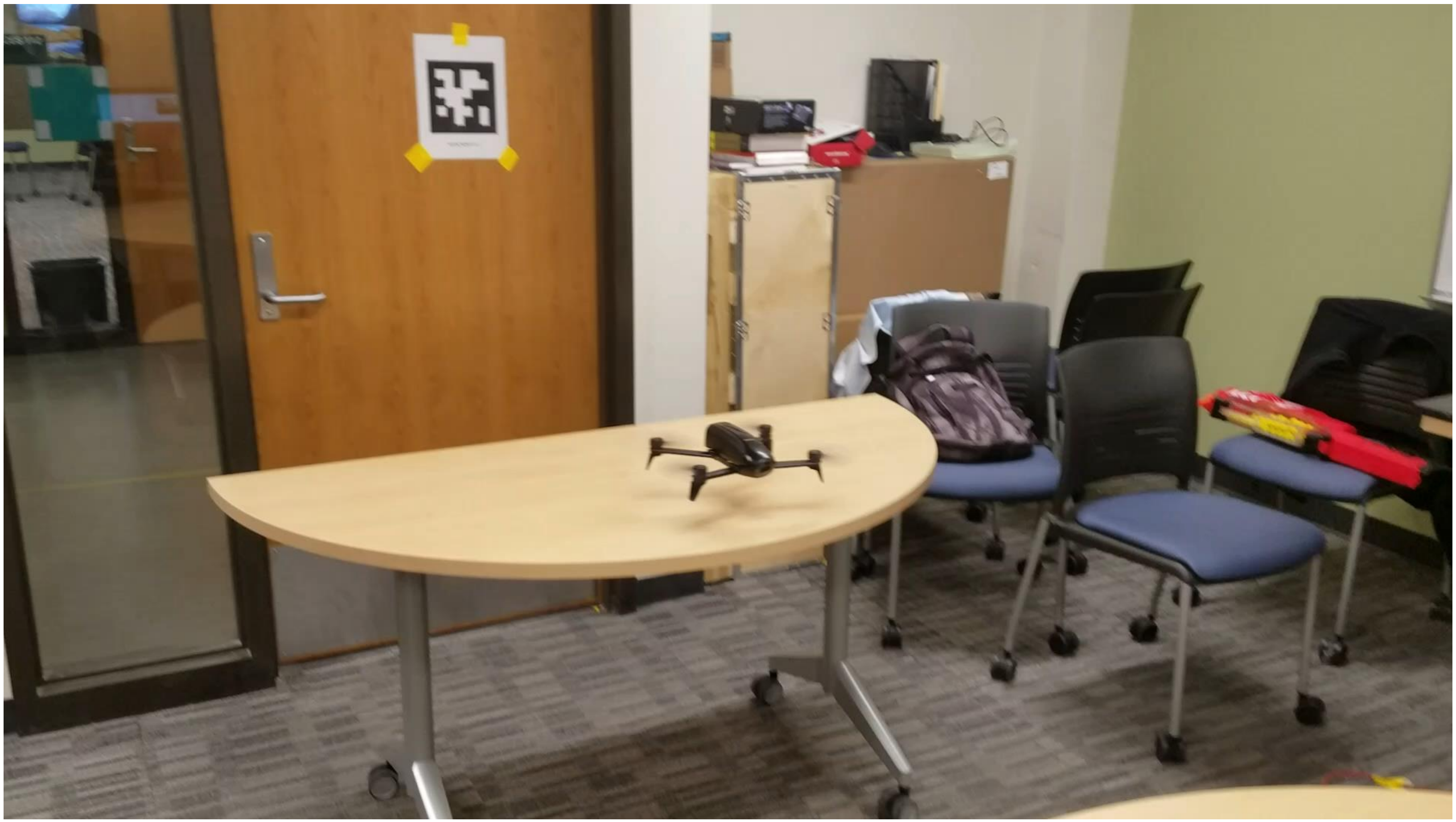
Do the reading, and quizzes will be easy!

Attend class, and homework will be straightforward!

Final Project – 40%

This is why you’re taking the course, make it count!

Robotics is Hard,
Nobody knows everything



Things You Will Know by May

- Robot Operating System
- Control on a real robot platform
- Probabilistic Motion Planning
- Task Planning
- Interaction Design
- Experimental Design / User Study Analysis
- Applied Artificial Intelligence and Machine Learning
    - Modeling
    - Active Learning
    - Learning from Demonstration

Class Schedule

Course Website:

Moodle Link: https://moodle.cs.colorado.edu/course/view.php?id=1124

Slides posted after each lecture

Location:

ECCR 150

Time:

12:30pm – 1:45pm

Office Hours:

By Appointment (ECES 128)

Project Phases

Phase 1: Idea/Hypothesis Development

Phase 2: Planning and Design

Phase 3: Implementation

Phase 4: Validation and Evaluation

Phase 5: Writing

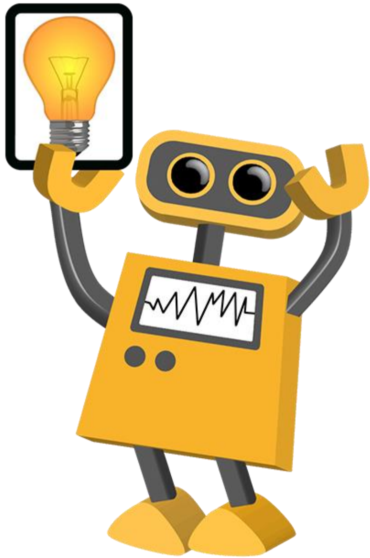
Phase 1: Idea Development

Identifying an area

- Significant but unexplored phenomena / unrealized capability
- Opportunities for impact (Incorporate your strengths into HRI!)

Form a research question

- Contextualize your question w.r.t. prior work



Phase 2: Planning

Goal: Develop plan to answer your research question

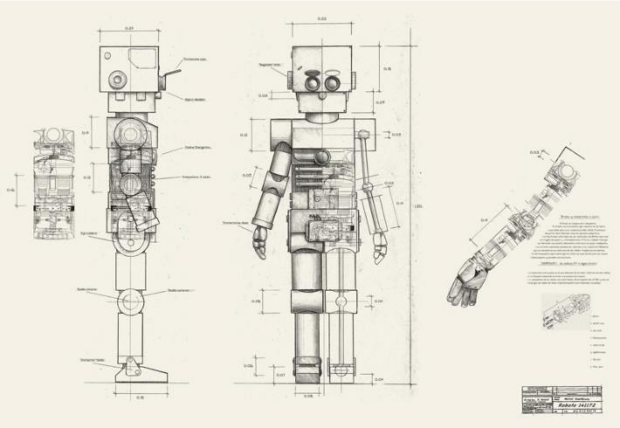
Technical roadmap for new capability

Identify key components, technologies required

Develop an evaluation metric and protocol

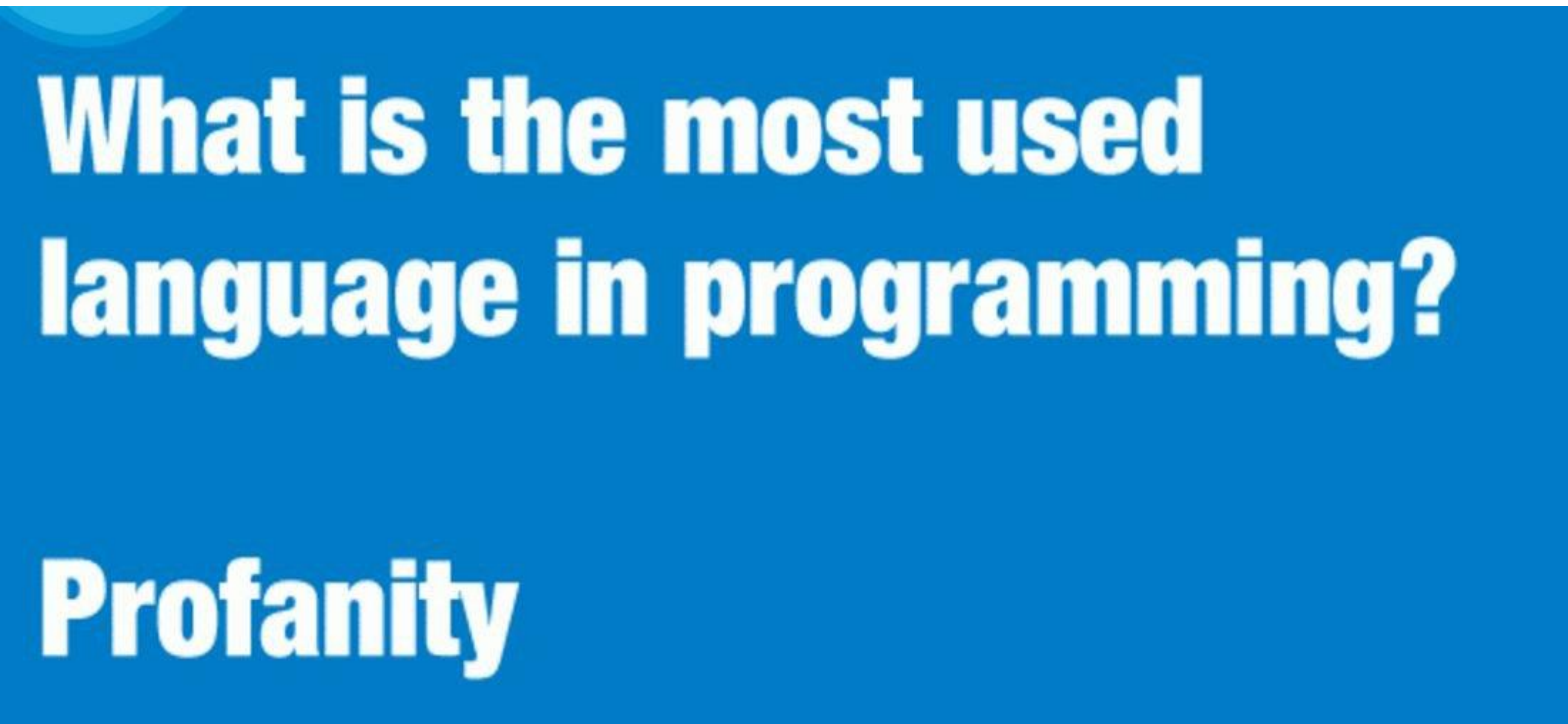
How will you know your contribution works?

If necessary: Seek IRB approval



Phase 3: Implementation

The bulk of your project time will be spent here!



Phase 4: Validation and Evaluation

Design your experimental protocol

Define your metrics for evaluation

Collect data

Analyze data

Draw conclusions

Phase 5: Writing

Project Report

4-8 page account of project progress

Written in ACM/IEEE conference paper format

Project Presentations

15 minute presentations in class

CSCI 7000 :: Roadmap

Week 0: Course Overview

Week 1:
Introduction to Motion Planning

Week 2:
Introduction to Trajectory Optimization

Hardware
Abstraction
Layer

Week 3:
Introduction to Task Planning

Week 4+:
HRI Research Papers

Algorithmic Human-Robot Interaction

Motion Planning I

CSCI 7000

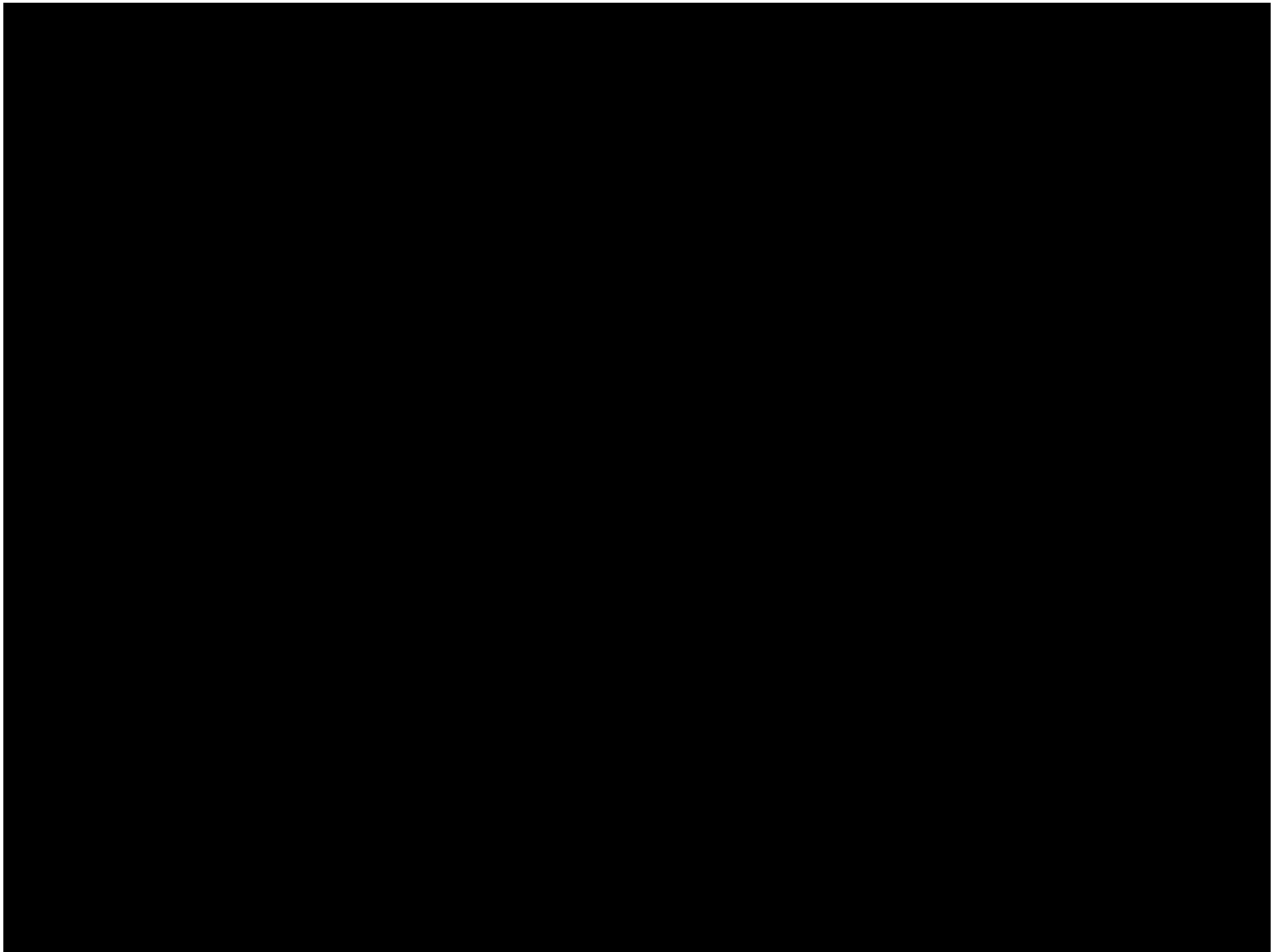
Prof. Brad Hayes

Computer Science Department

University of Colorado Boulder

Motion Planning

Motion Planning



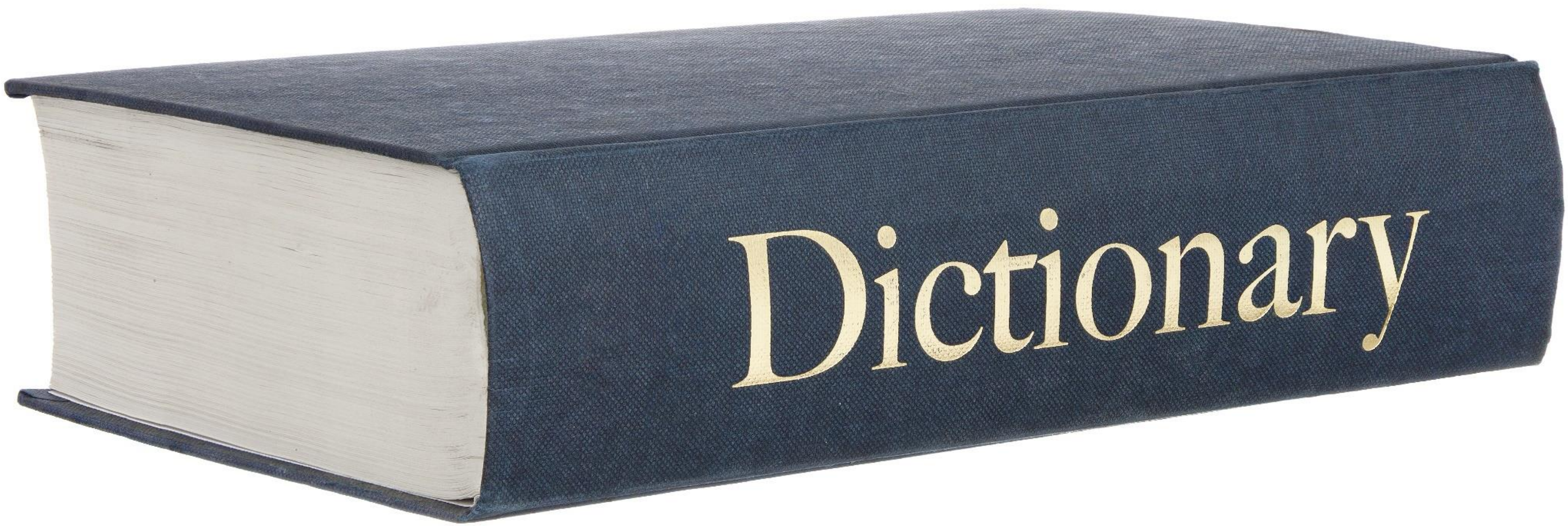
Motion Planning

Robot Motion Planning:
Find a path from point A to point B

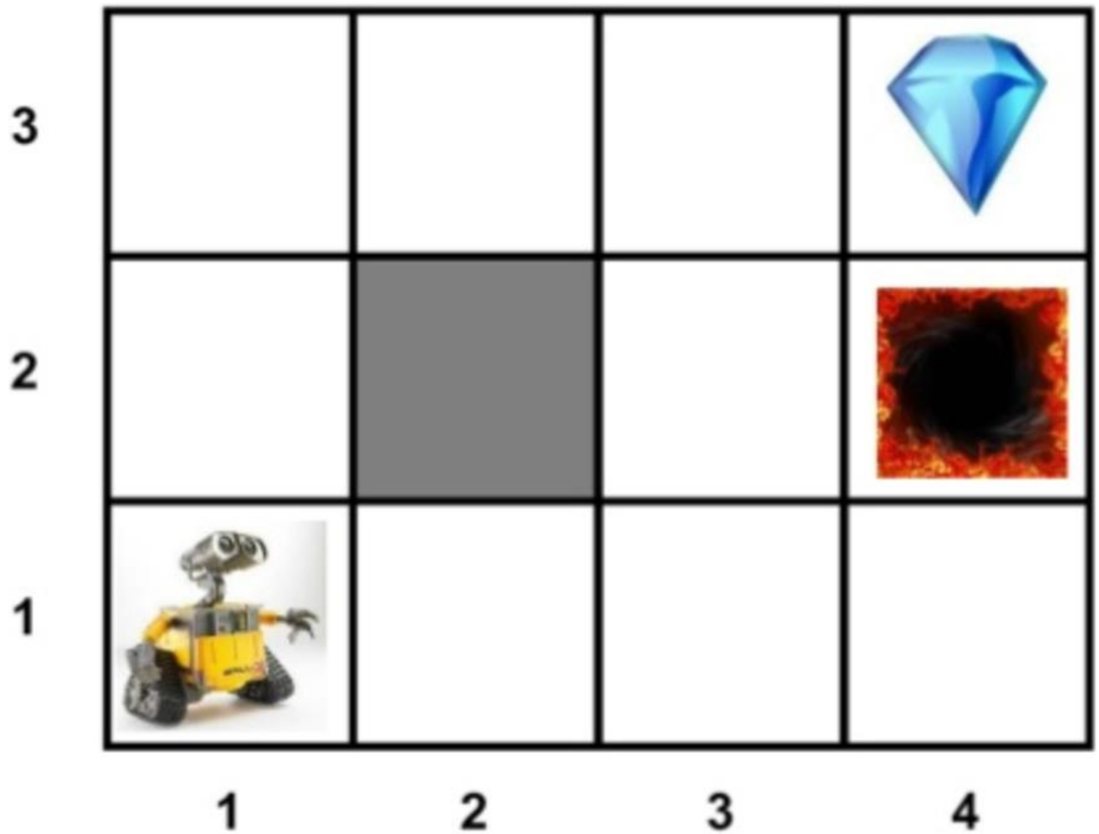
All complete search algorithms scale exponentially

Motion planning problems are high dimensional, e.g., big exponent.

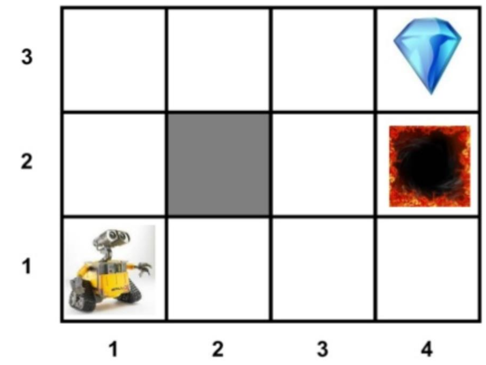
But First…



Sample Problem



Sample Problem Terminology



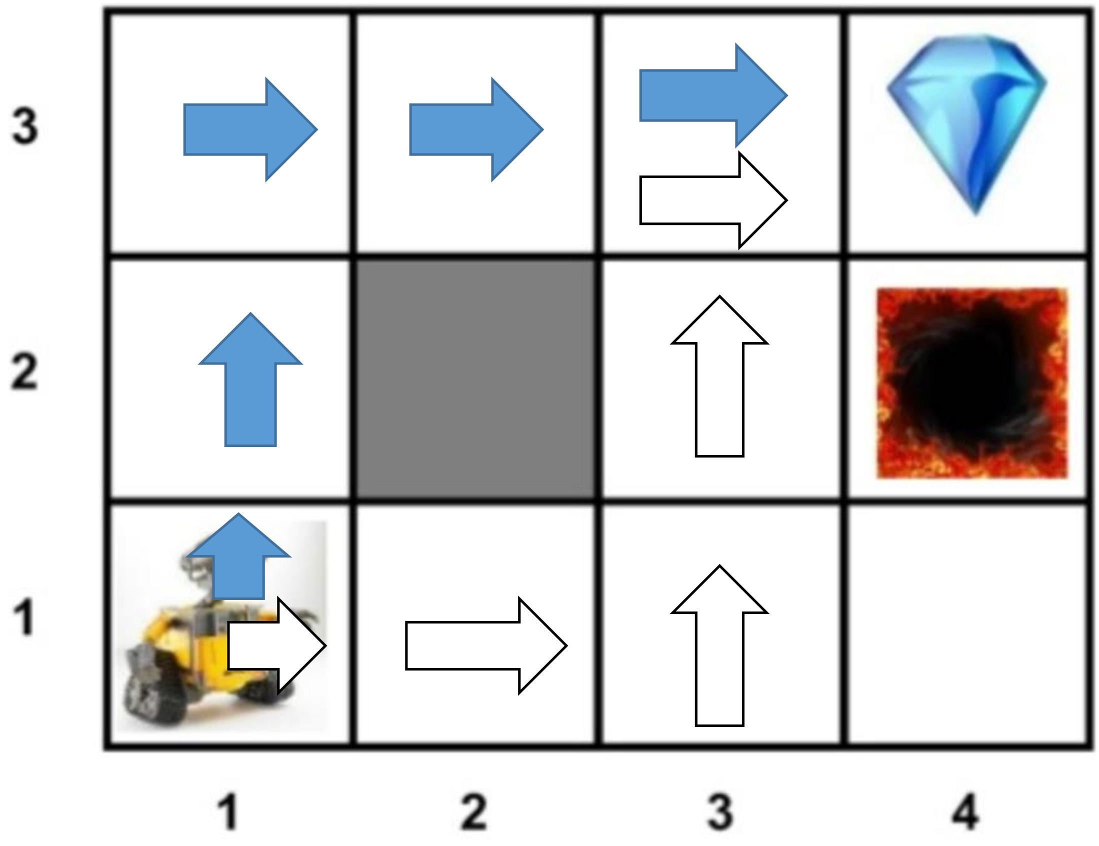
A **state** is a representation of the world

An **action** is something that transitions you from one state to another (can also be a self-transition!)

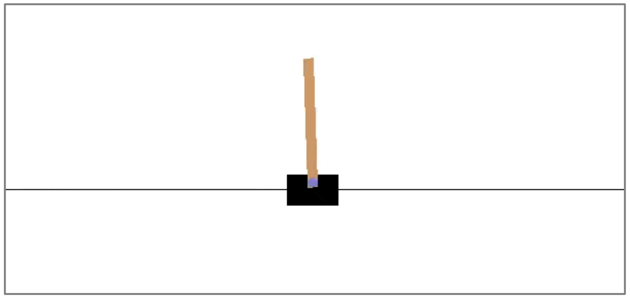
A **transition function** T(s,a,s') provides the probability that a particular action **a** taken in a particular state **s** will bring the system to state **s'**

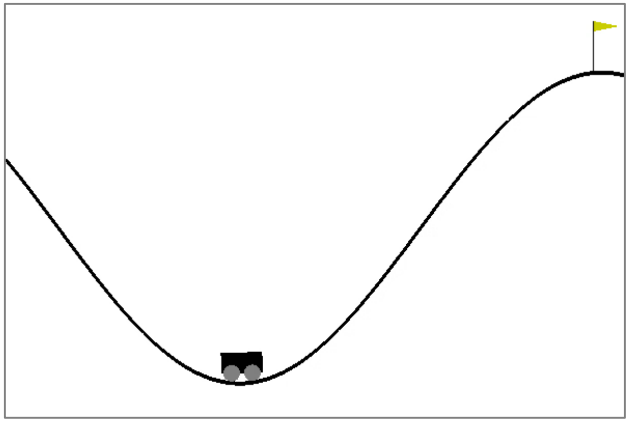
A **reward function** R(s, a) provides the value of taking a particular action **a** in state **s**

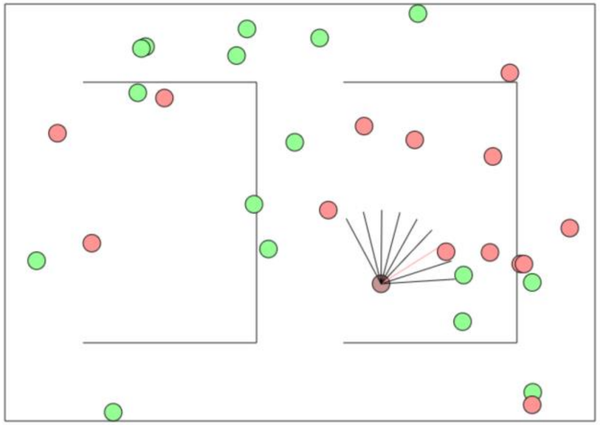
Sample Solutions

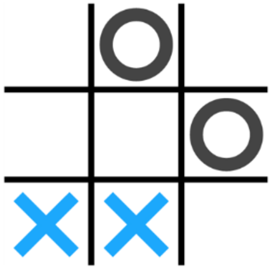


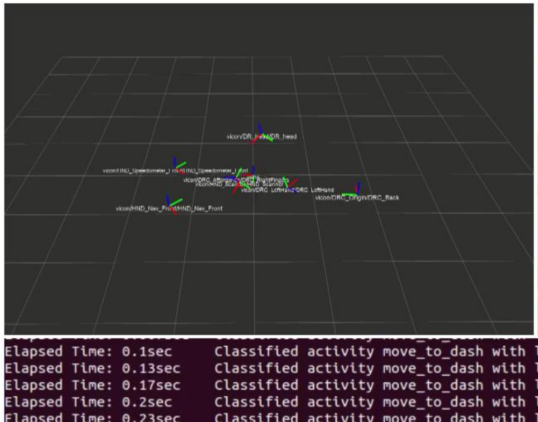
State Representation is Critical

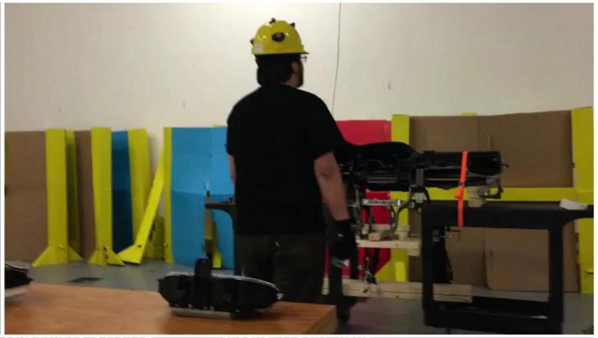












The Real World Brings Uncertainty

“All models are wrong but some are useful.”

- George Box

Recurring Theme: Robotics is Challenging

Actions in the world must be coordinated with perception of and models of the world

Physical world is continuous, dynamic, and accessible only through sensing

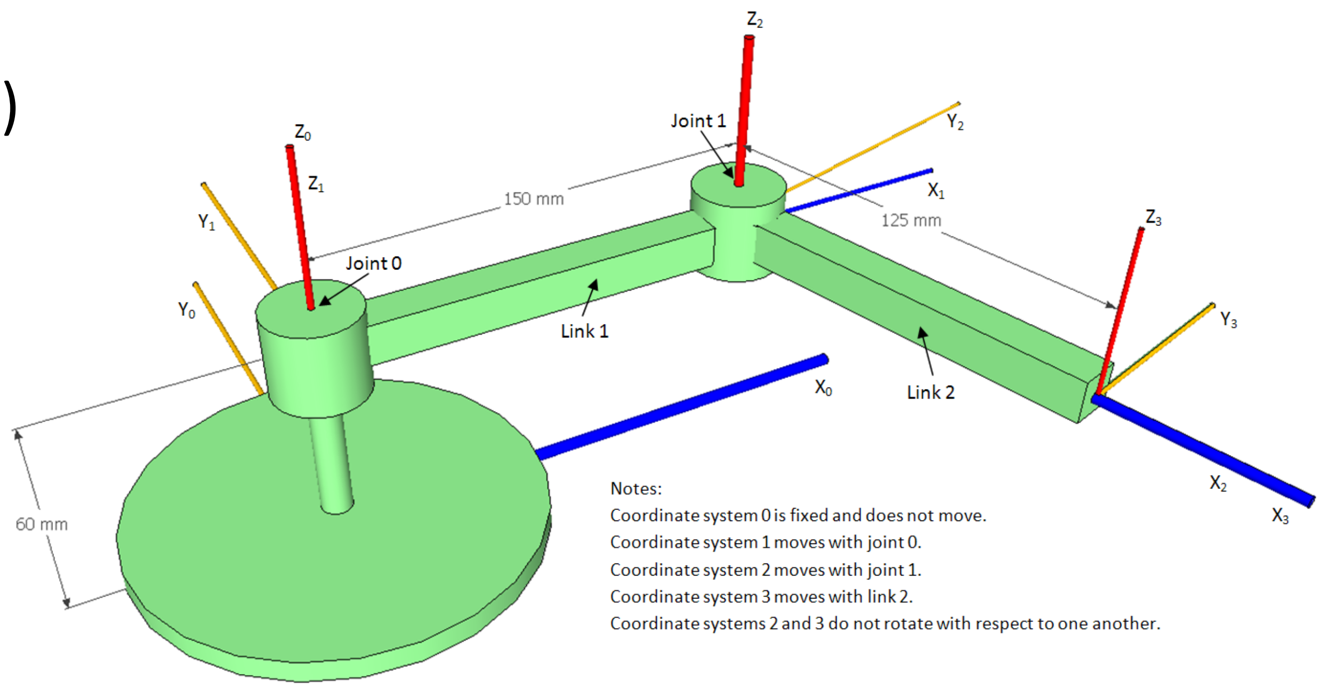
Sensors and actuators are uncertain, exhibiting noise, and are subject to error

Communicating intent often requires comprehensive knowledge about the world

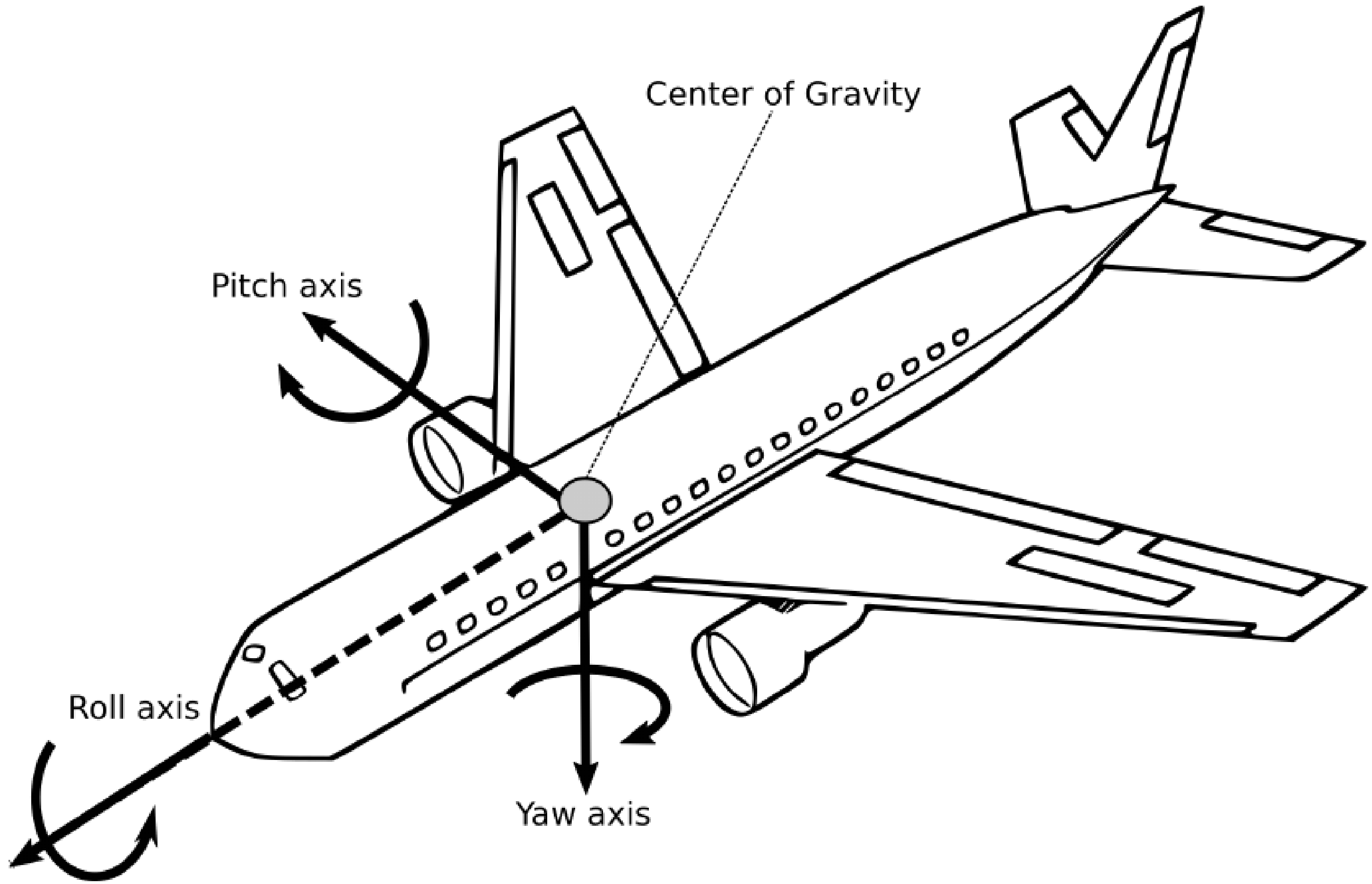
It’s not enough for robots to be functional, they must also be *tolerable*

Kinematics

- Kinematics
  - The way parts of a robot move with respect to each other and the environment
  - Position (x)
  - Speed (x')
- Dynamics
  - Acceleration (x'')
  - Jerk (x''')



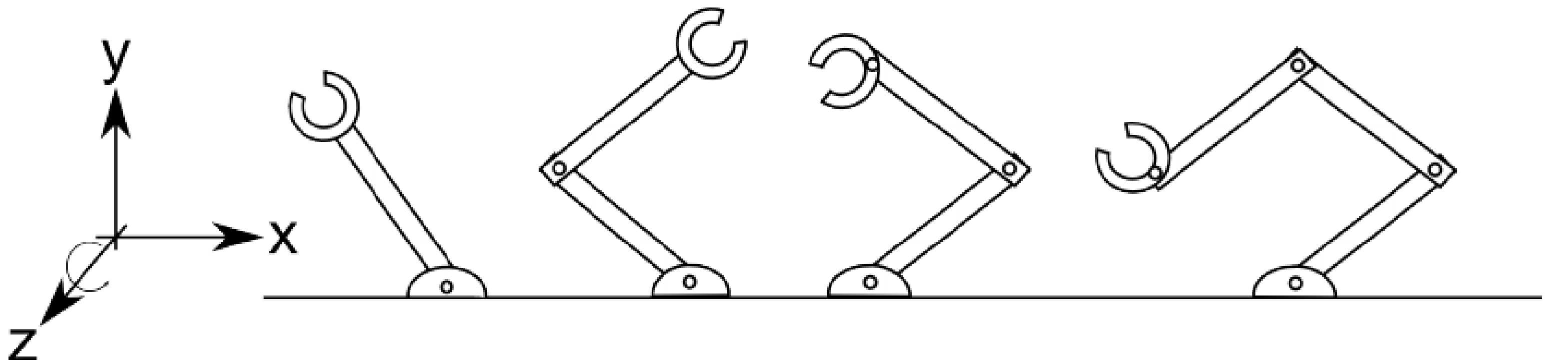
Degrees of Freedom:
X, Y, Z, Yaw, Pitch, and Roll



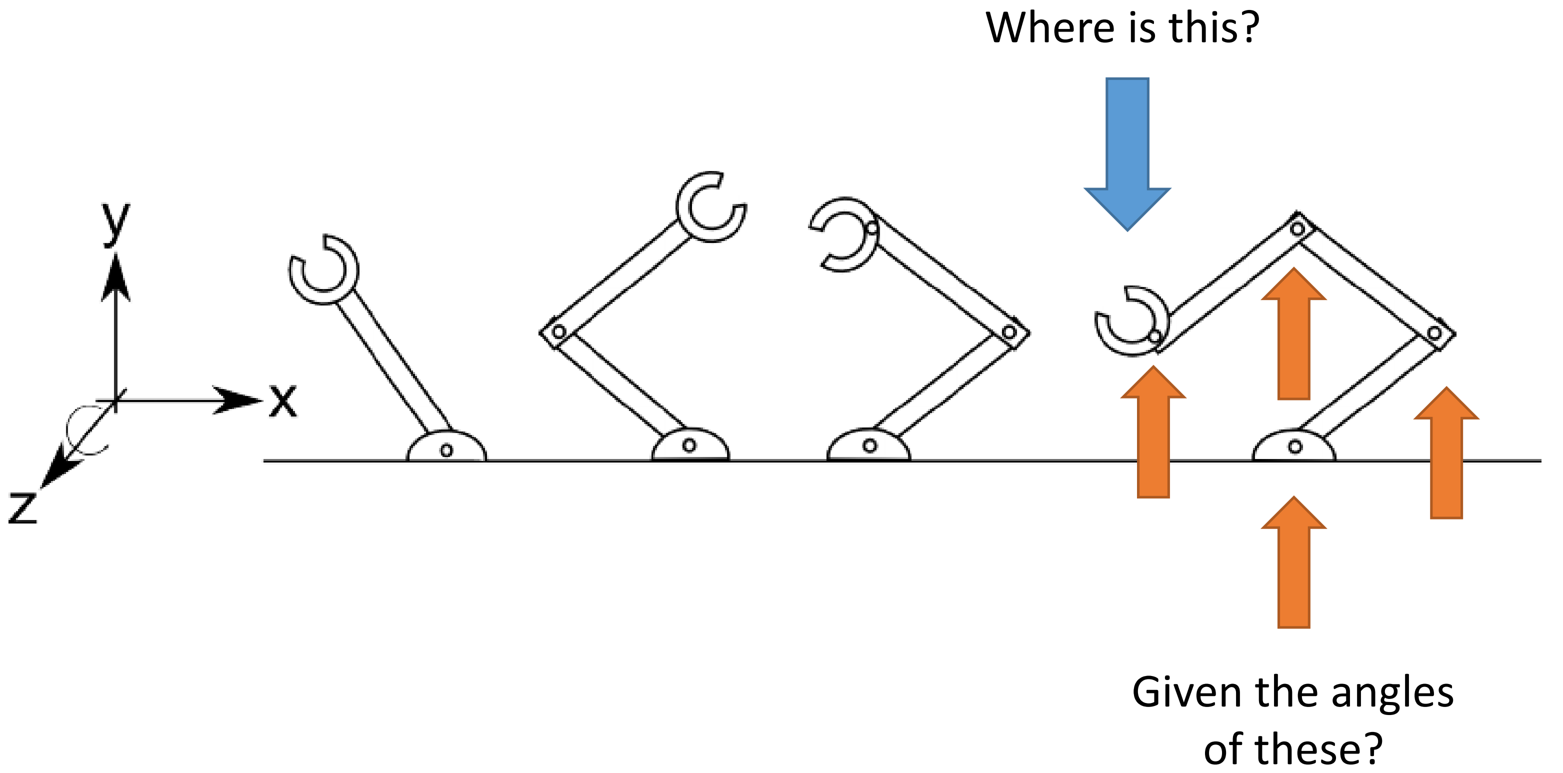
Degrees of Freedom (cont.)

- Most robot arms have 6-DoF at their end effector
- Can also measure DoF at the robot's base
  - DoF = Number of controllable points of actuation
  - Can have **redundant** degrees of freedom (Robot DoF > Environment DoF)
    - Why would you want this?
- Control paradigms
  - Position control
  - Torque control
  - Force control

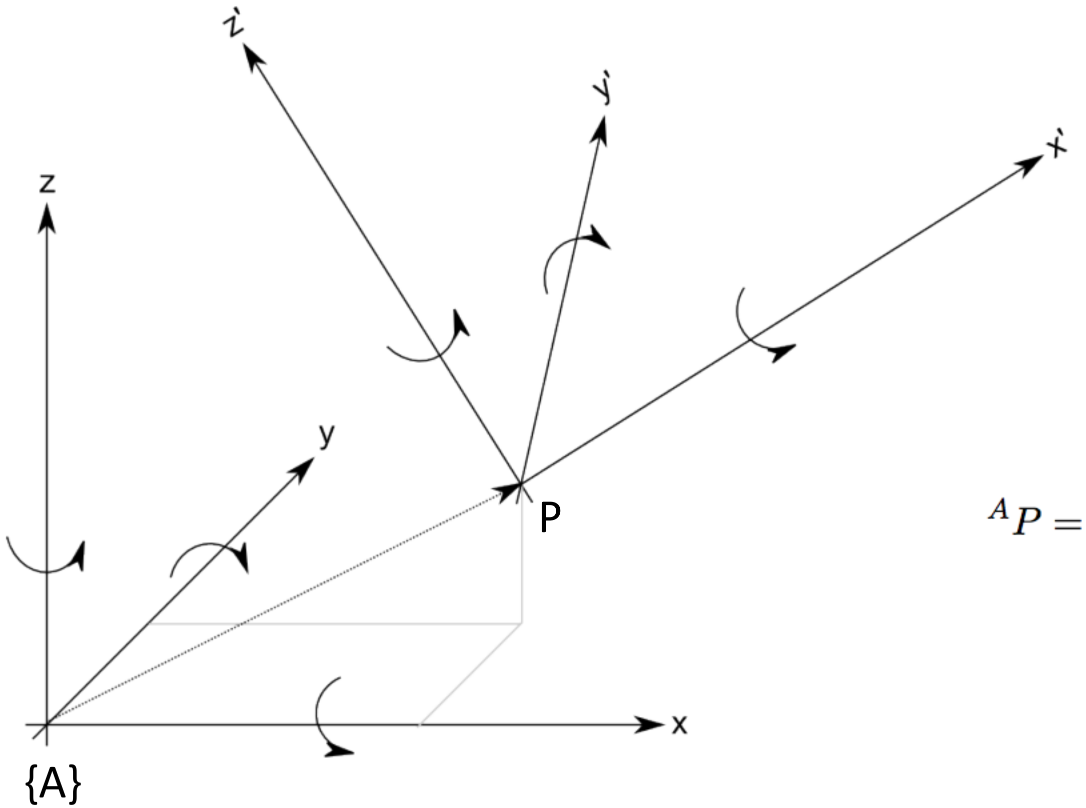
Manipulator DoF



Forward Kinematics

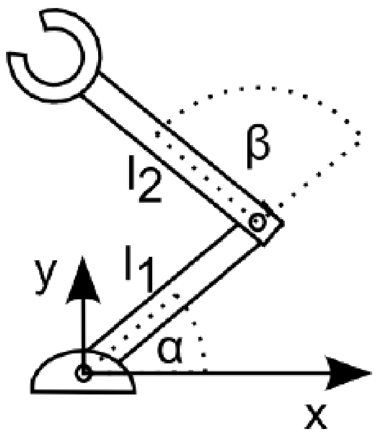


Nested Coordinate Systems



$$^{A}P = p_x \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + p_y \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + p_z \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Transformations in Practice: Forward Kinematics (Arm)



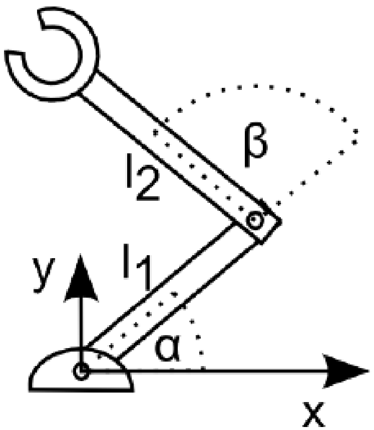
$$
\begin{aligned}
x_1 &= \cos \alpha l_1 \\
y_1 &= \sin \alpha l_1
\end{aligned}
$$

$$
\begin{aligned}
x_2 &= \cos(\alpha + \beta) l_2 + x_1 \\
y_2 &= \sin(\alpha + \beta) l_2 + z_1
\end{aligned}
$$

$$
\begin{aligned}
x &= \cos(\alpha + \beta) l_2 + \cos \alpha l_1 \\
y &= \sin(\alpha + \beta) l_2 + \sin \alpha l_1
\end{aligned}
$$

Inverse Kinematics

Known: l_1, l_2



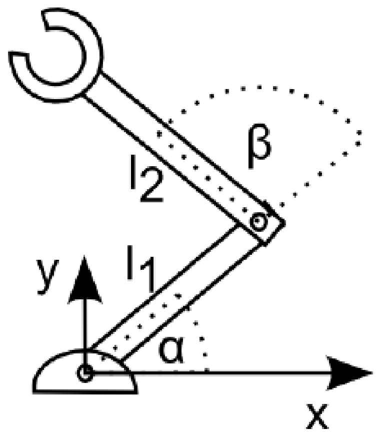
$$
\begin{aligned}
x &= \cos(\alpha+\beta)l_2 + \cos\alpha l_1 \\
y &= \sin(\alpha+\beta)l_2 + \sin\alpha l_1
\end{aligned}
$$

Forward Kinematics
(Given α, β find x, y)

What angles do I need to set my joints to reach a desired pose?

Inverse Kinematics
(Given x, y find α, β)

Inverse Kinematics of a 2-link Arm



$$x_1 = \cos\alpha l_1$$

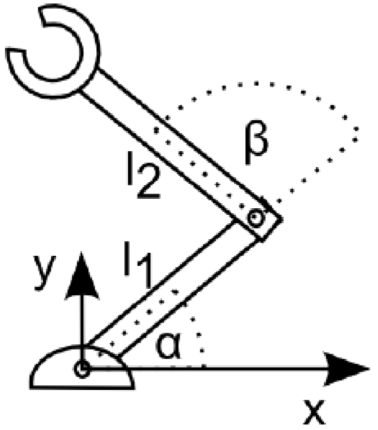
$$\left[\cos^{-1}\frac{x_1}{l_1}, -\cos^{-1}\frac{x_1}{l_1}\right]$$

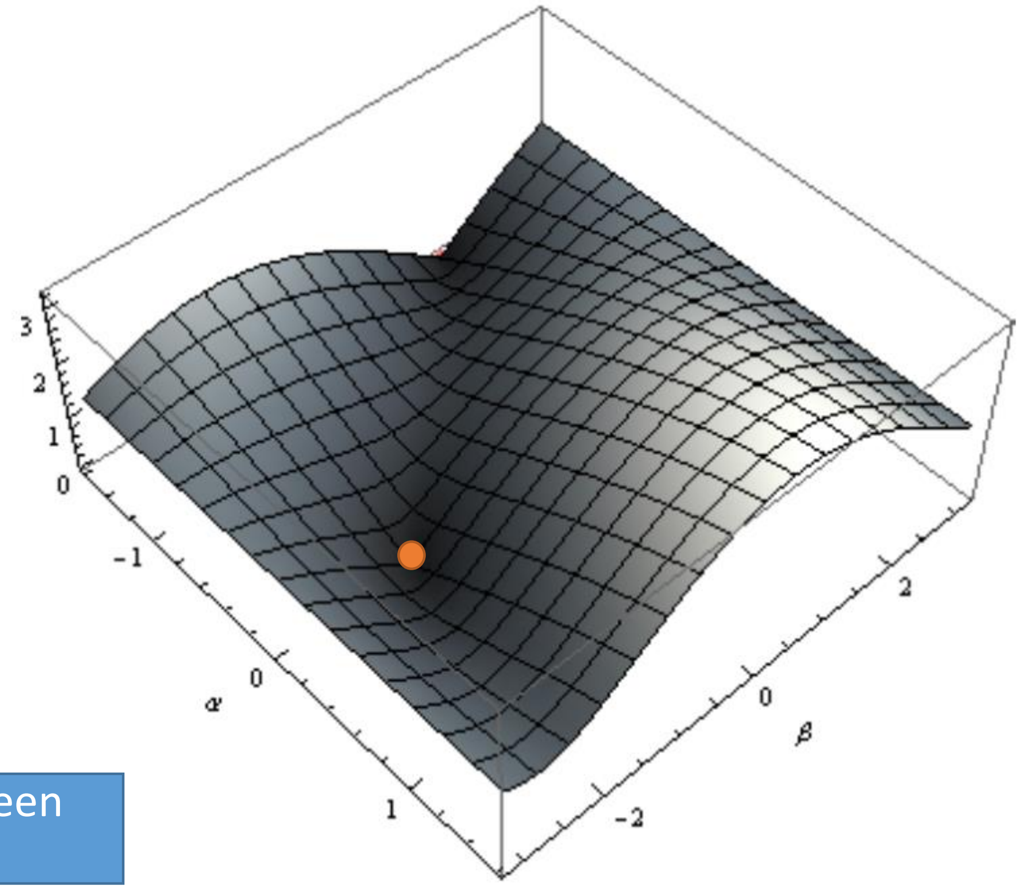
$$\begin{aligned} x &= \cos(\alpha+\beta)l_2 + \cos\alpha l_1 \\ y &= \sin(\alpha+\beta)l_2 + \sin\alpha l_1 \end{aligned}$$

$$\alpha \to \cos^{-1}\left(\frac{x^2 y + y^3 - \sqrt{4x^4 - x^6 + 4x^2y^2 - 2x^4y^2 - x^2y^4}}{2(x^2+y^2)}\right)$$

$$\beta \to -\cos^{-1}\left(1/2(-2 + x^2 + y^2)\right)$$

Easier ways to solve the IK problems





Just the Euclidean distance between two vectors!

$$f_{x,y}(\alpha,\beta) = \sqrt{(\sin(\alpha+\beta)l_2 + \sin(\alpha)l_1 - y)^2 + (\cos(\alpha+\beta)l_2 + cos(\alpha)l_1 - x)^2}$$

Configuration Space

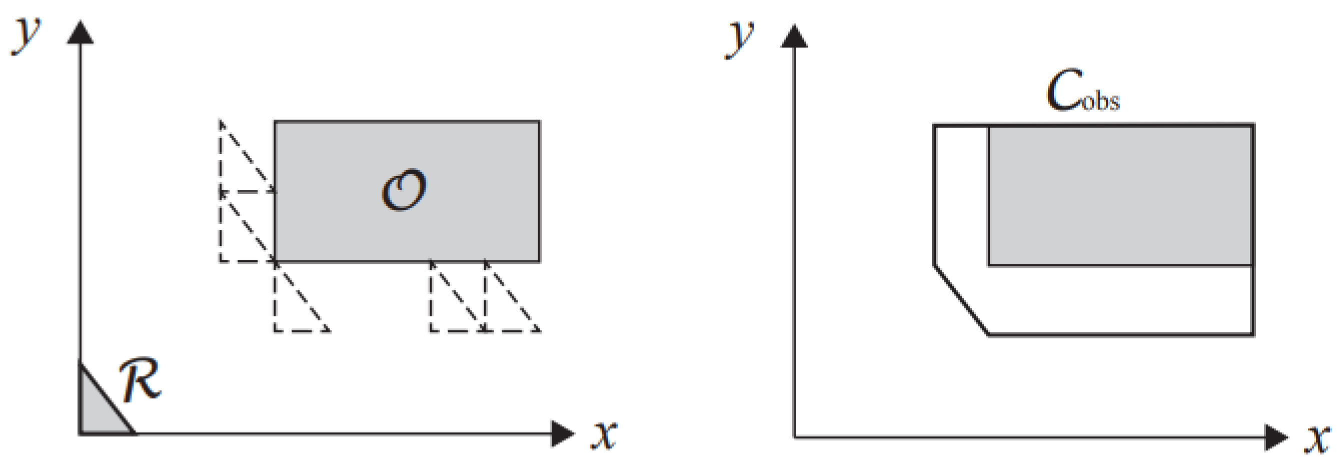
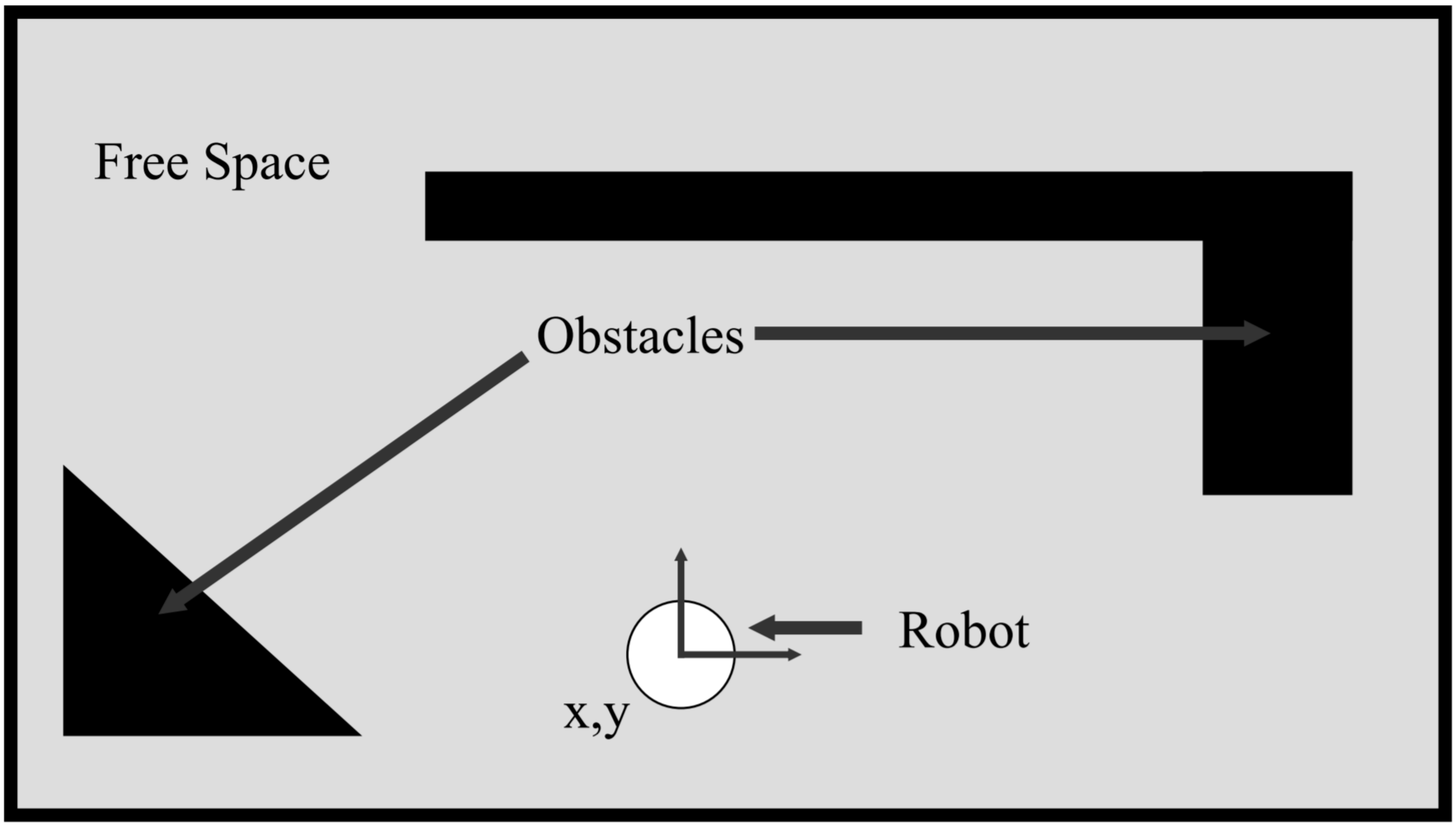
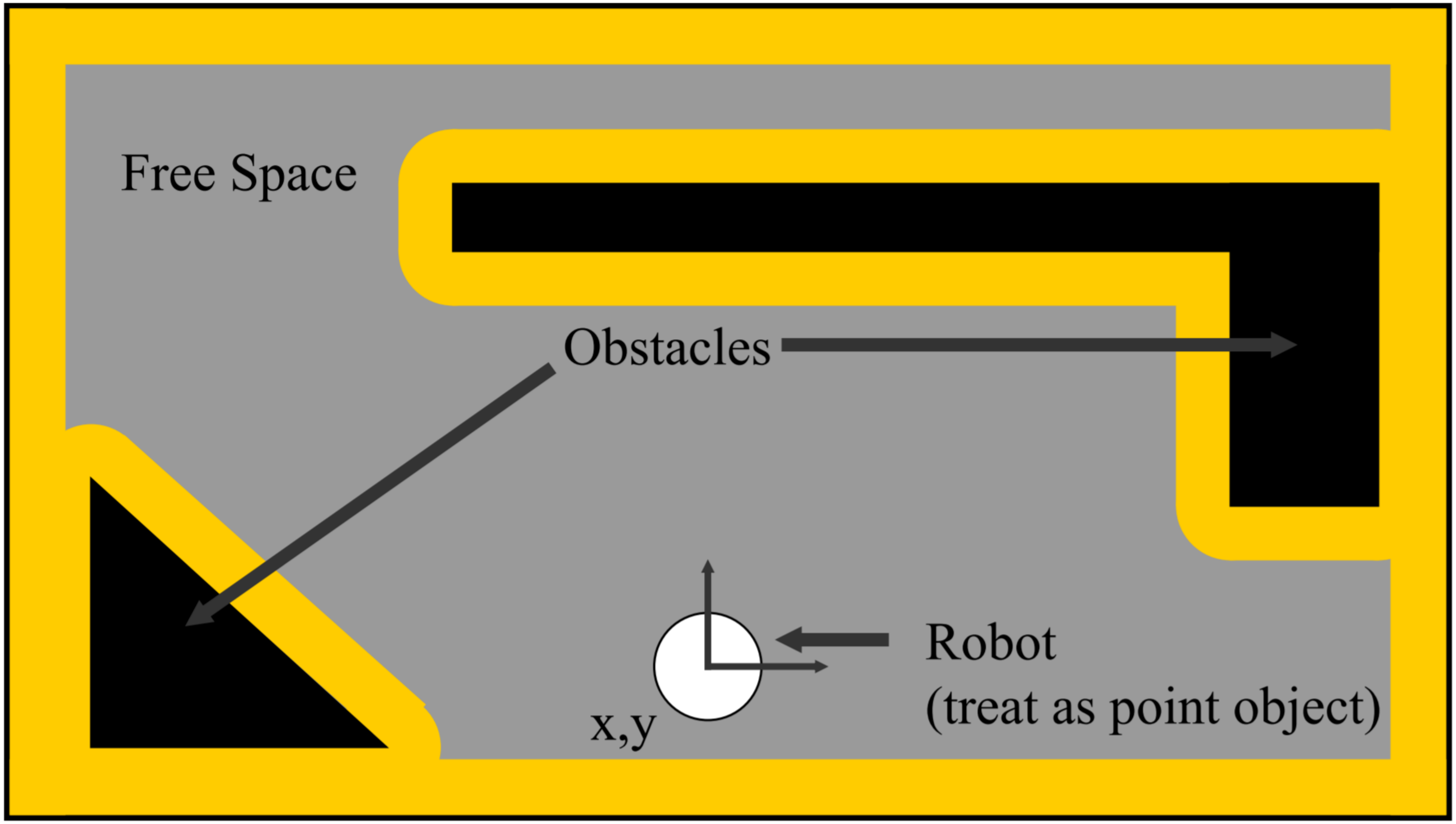


Figure 2.4: C_{obs} for a robot R that translates in x-y with a rectangle obstacle O

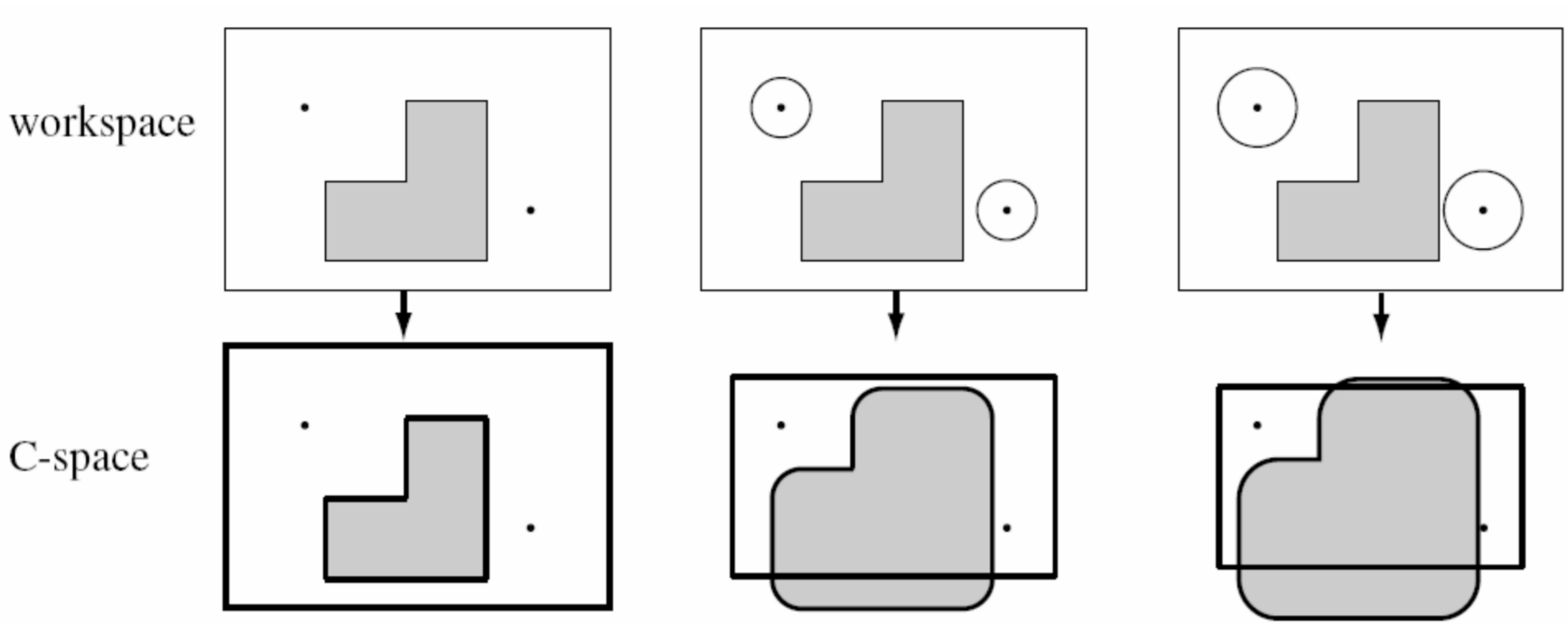
Configuration Space



Configuration Space



Configuration Space



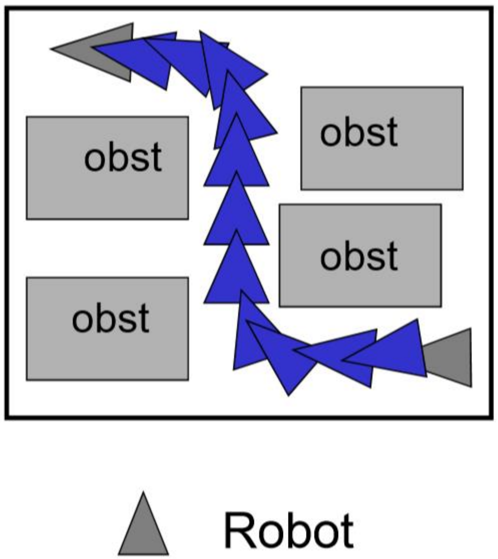
Configuration Space

- Allows us to reduce robot positions to a single point
  - Very convenient for planning!

- One axis of configuration space for each degree of freedom of the robot
  - Again, very convenient for planning as it can occur in the same space as the robot's controllers

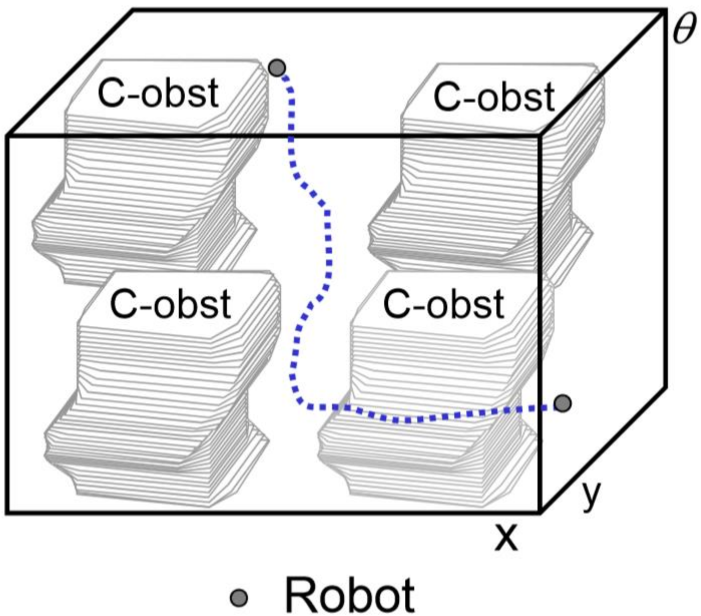
Non-circular Robots in C-Space

Workspace
(*x, y*)



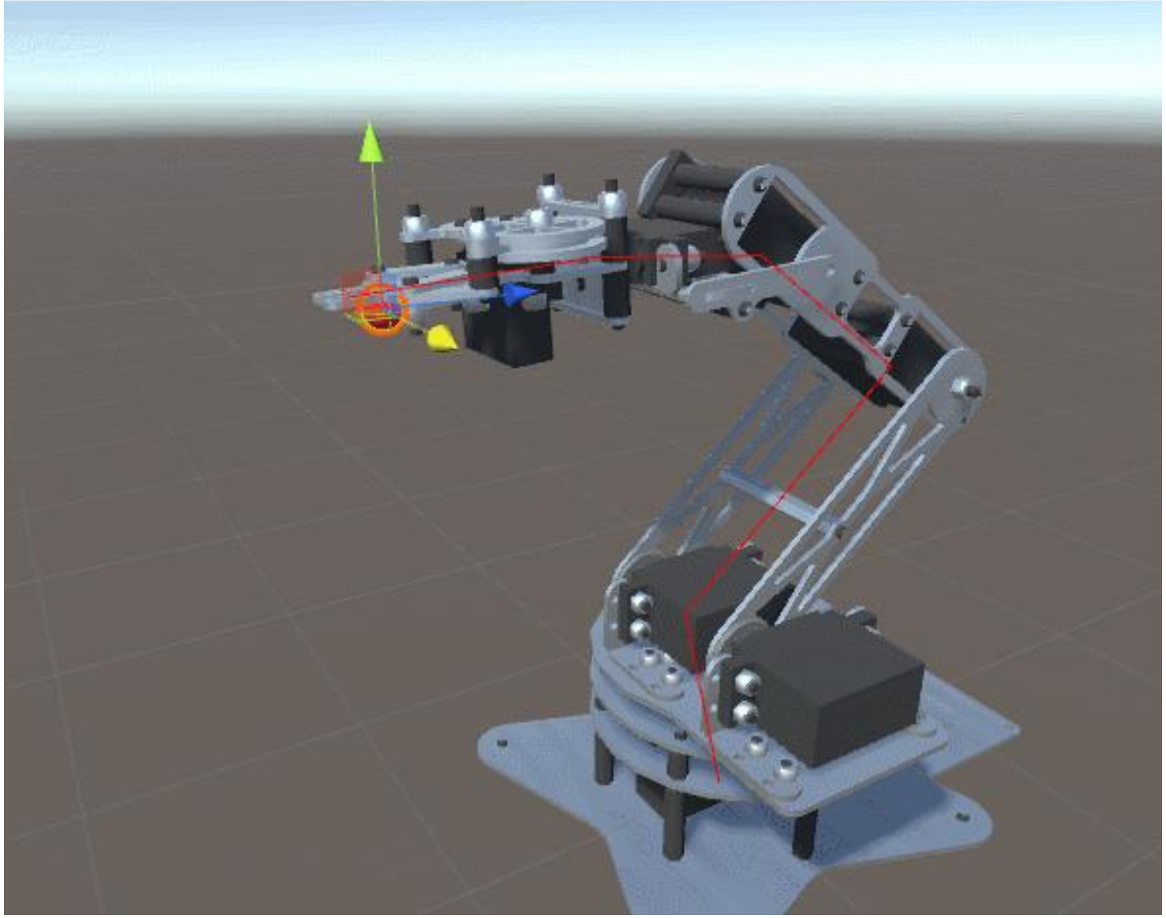
Path is hard to express

C-space
(*x, y, θ*)



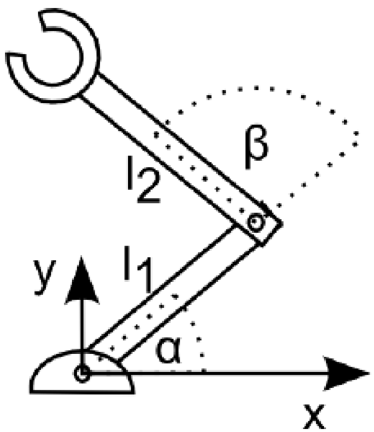
Path is just a space curve

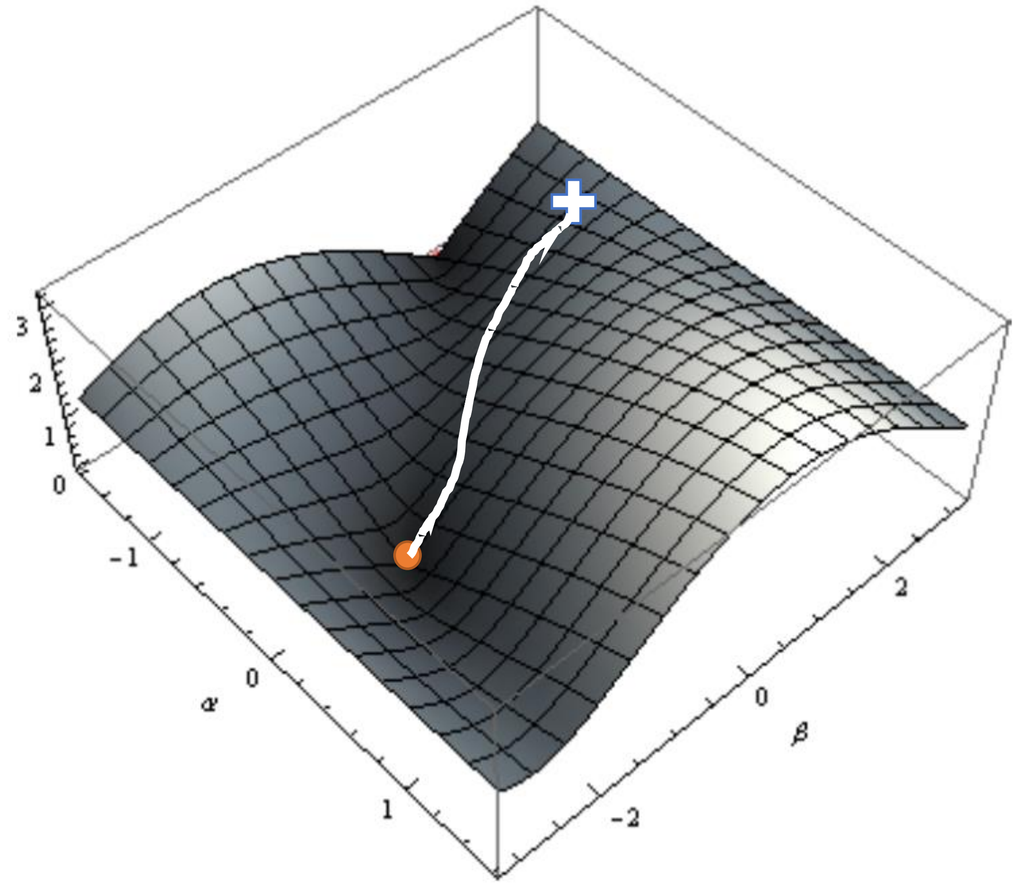
End-effector Position Control



https://www.alanzucconi.com/2017/04/10/robotic-arms/

Motion Planning in EE Space





$$f_{x,y}(\alpha,\beta) = \sqrt{(\sin(\alpha+\beta)l_2 + \sin(\alpha)l_1 - y)^2 + (\cos(\alpha+\beta)l_2 + cos(\alpha)l_1 - x)^2}$$

Gradient Descent for Solving IK

Given distance-from-goal function f:

$$\nabla f(\alpha_0, \alpha_1, \alpha_2) = [\nabla f_{\alpha_0}(\alpha_0, \alpha_1, \alpha_2), \nabla f_{\alpha_1}(\alpha_0, \alpha_1, \alpha_2), \nabla f_{\alpha_2}(\alpha_0, \alpha_1, \alpha_2)]$$

$$\nabla f_{a_0} = (\alpha_0, \alpha_1, \alpha_2) = \frac{f(\alpha_0 + \Delta_x, \alpha_1, \alpha_2) - f(\alpha_0, \alpha_1, \alpha_2)}{\Delta x}$$

$$\nabla f_{a_1} = (\alpha_0, \alpha_1, \alpha_2) = \frac{f\left(\alpha_0, \alpha_1 + \Delta_y, \alpha_2\right) - f(\alpha_0, \alpha_1, \alpha_2)}{\Delta y}$$

$$\nabla f_{a_2} = (\alpha_0, \alpha_1, \alpha_2) = \frac{f(\alpha_0, \alpha_1, \alpha_2 + \Delta_z) - f(\alpha_0, \alpha_1, \alpha_2)}{\Delta z}$$

Gradient Descent for Solving IK

Gradient Definition:

$$\nabla f(\alpha_0, \alpha_1, \alpha_2) = [\nabla f_{\alpha_0}(\alpha_0, \alpha_1, \alpha_2), \nabla f_{\alpha_1}(\alpha_0, \alpha_1, \alpha_2), \nabla f_{\alpha_2}(\alpha_0, \alpha_1, \alpha_2)]$$

$$\nabla f_{a_0} = (\alpha_0, \alpha_1, \alpha_2) = \frac{f(\alpha_0 + \Delta_x, \alpha_1, \alpha_2) - f(\alpha_0, \alpha_1, \alpha_2)}{\Delta x}$$

Update Rule:

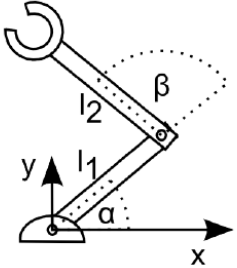
$$\begin{aligned} \alpha_0 &\leftarrow \alpha_0 - L\nabla f_{\alpha_0}(\alpha_0, \alpha_1, \alpha_2) \\ \alpha_1 &\leftarrow \alpha_1 - L\nabla f_{\alpha_1}(\alpha_0, \alpha_1, \alpha_2) \\ \alpha_2 &\leftarrow \alpha_2 - L\nabla f_{\alpha_2}(\alpha_0, \alpha_1, \alpha_2) \end{aligned}$$

Distance from goal
$f: \mathbb{R}^3 \rightarrow \mathbb{R}$

Joint angles
α_i

Learning rate
L

Jacobian



Linear equations dictate end-effector position:

$$x_e(\alpha,\beta) = l_1 \cos(\alpha) + l_2 \cos(\alpha+\beta)$$
$$y_e(\alpha,\beta) = l_1\, sin(\alpha) + l_2 \sin(\alpha+\beta)$$

Relationship between position change and angle change:

$$dx_e = \frac{\delta x_e(\alpha,\beta)}{\delta\alpha} d\alpha + \frac{\delta x_e(\alpha,\beta)}{\delta\beta} d\beta$$
$$dy_e = \frac{\delta y_e(\alpha,\beta)}{\delta\alpha} d\alpha + \frac{\delta y_e(\alpha,\beta)}{\delta\beta} d\beta$$
$$dx = J \cdot d\boldsymbol{q}$$

$\frac{dx_e}{dt} = \boldsymbol{J}\frac{d\boldsymbol{q}}{dt}$ or in other terms: $v_e = \boldsymbol{J} \cdot \dot{\boldsymbol{q}}$

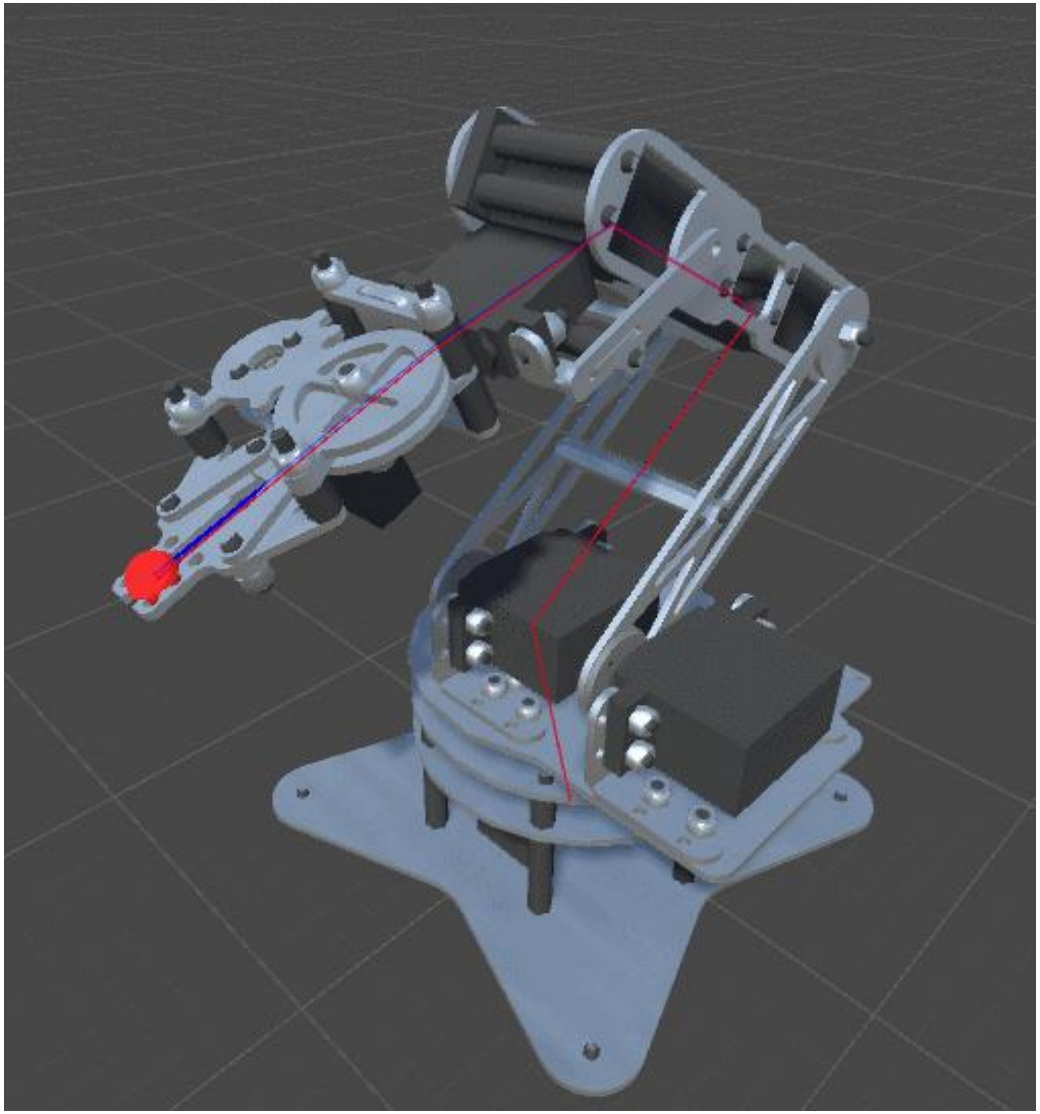
Jacobian

$$\frac{dx_e}{dt} = \boldsymbol{J}\frac{d\boldsymbol{q}}{dt}$$ or in other terms: $$v_e = \boldsymbol{J} \cdot \dot{\boldsymbol{q}}$$

$$\dot{q} = \boldsymbol{J}^{-\mathbf{1}} \cdot (\boldsymbol{v}_{\boldsymbol{e},\boldsymbol{des}} + \boldsymbol{K}(\boldsymbol{x}_{\boldsymbol{e},\boldsymbol{des}} - \boldsymbol{x}))$$

$\dot{q}$: Change in C-space
K: gain
$v_{e,des}$: Desired velocity
$x_{e,des}$: Desired position

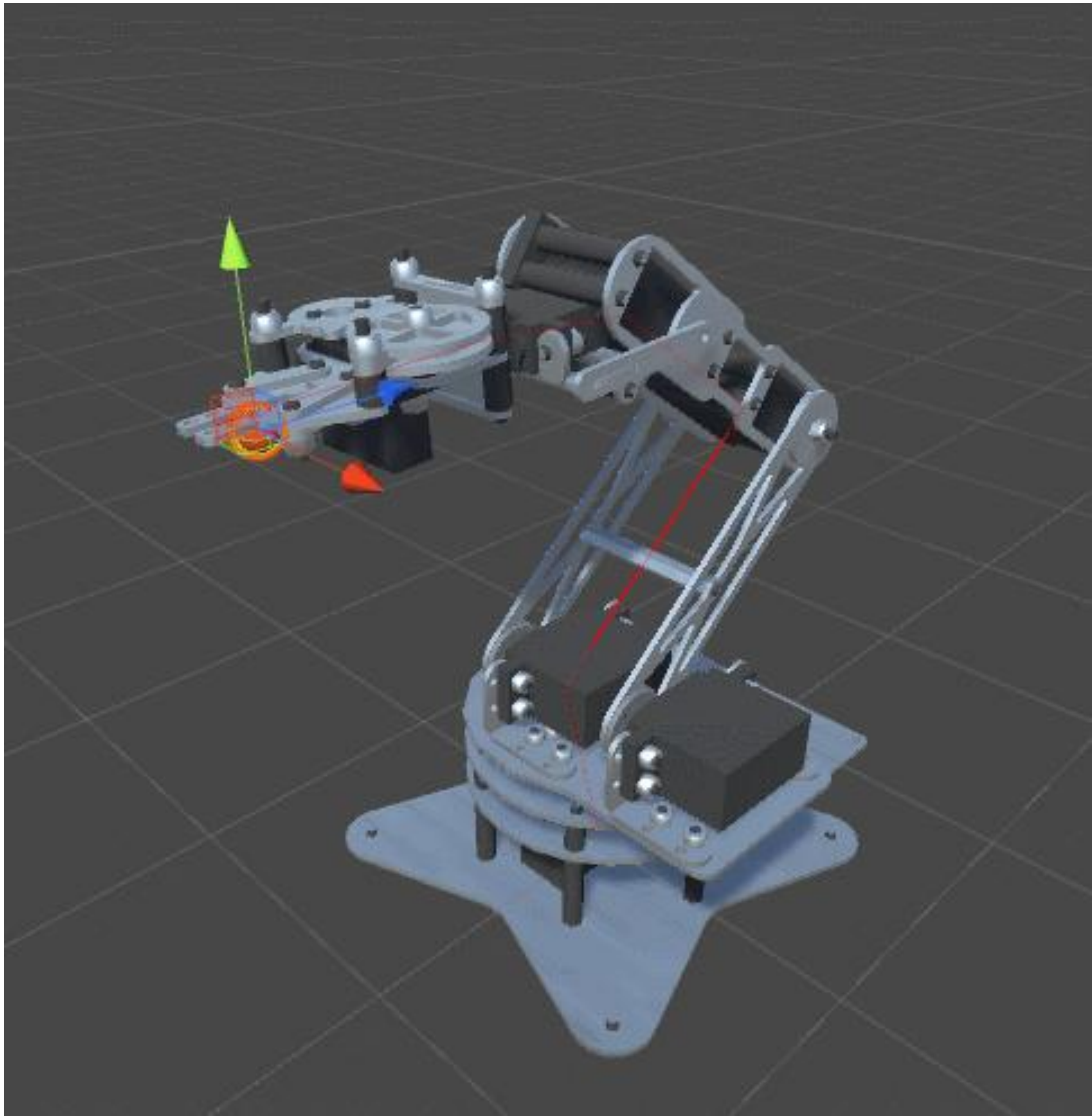
Convergence Problems!



$$\nabla f_{a_0} = (\alpha_0, \alpha_1, \alpha_2) = \frac{f(\alpha_0 + \Delta_x, \alpha_1, \alpha_2) - f(\alpha_0, \alpha_1, \alpha_2)}{\Delta x}$$

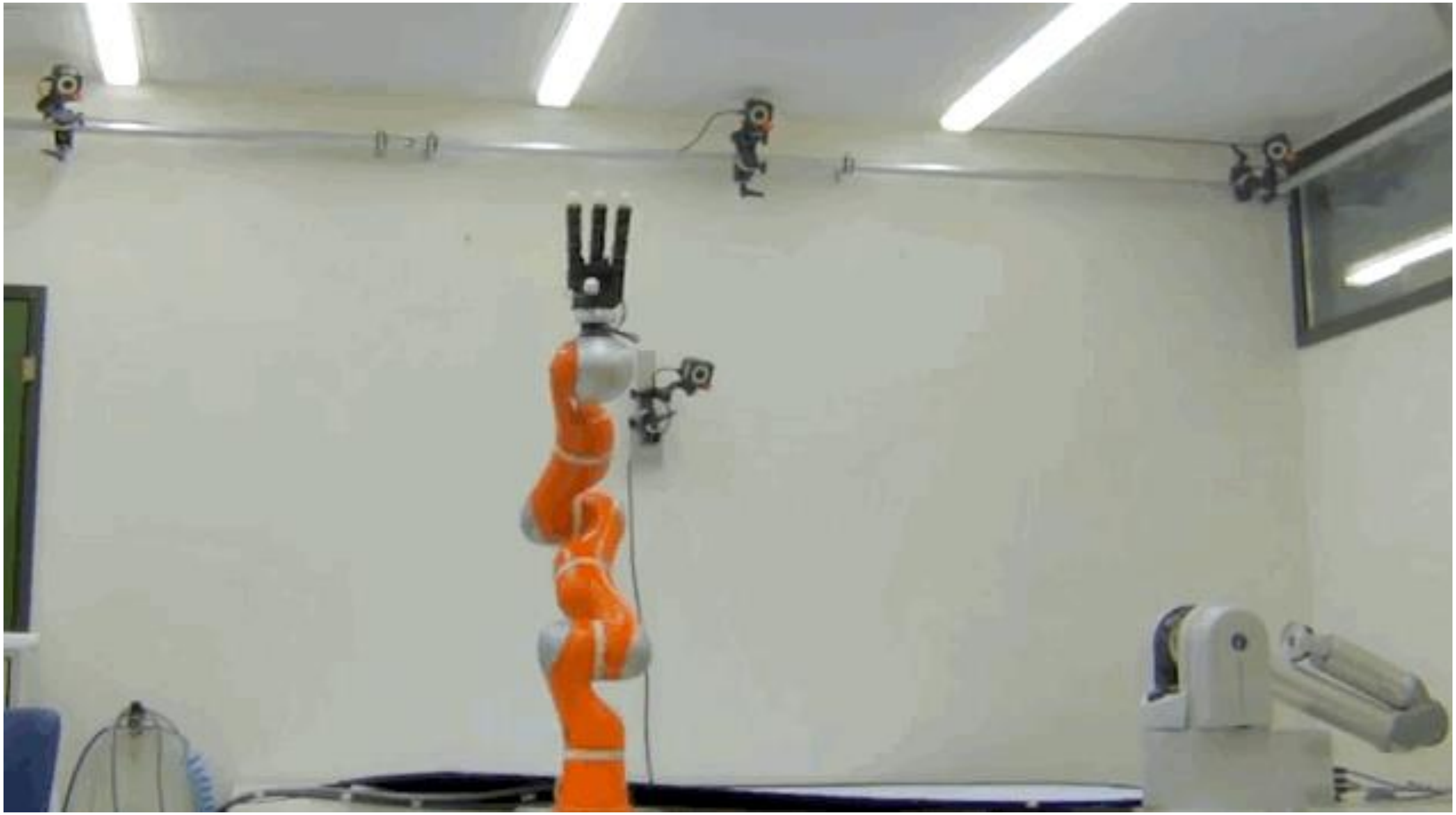
$$\alpha_0 \leftarrow \alpha_0 - L\nabla f_{\alpha_0}(\alpha_0, \alpha_1, \alpha_2)$$

Convergence Problems!



How do we fix this?

Fast IK Planning



YCB Task:
HERB Stacks Cups

Vinitha Ranganeni
Jennifer King
Rachel Holladay
Siddhartha Srinivasa

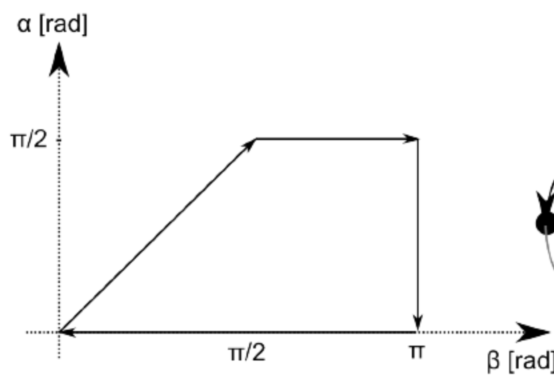
The Robotics Institute
Carnegie Mellon University

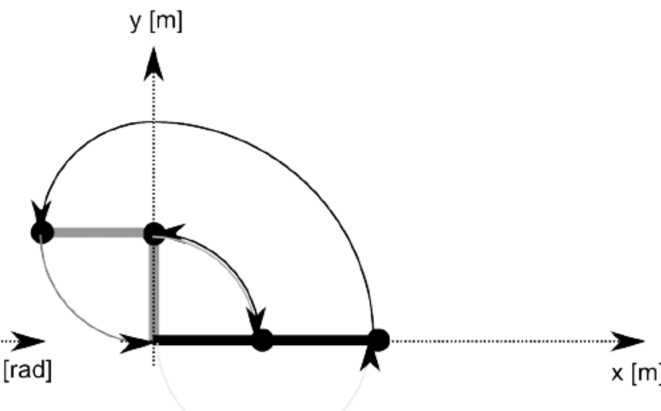
THE ROBOTICS INSTITUTE

Motion Planning

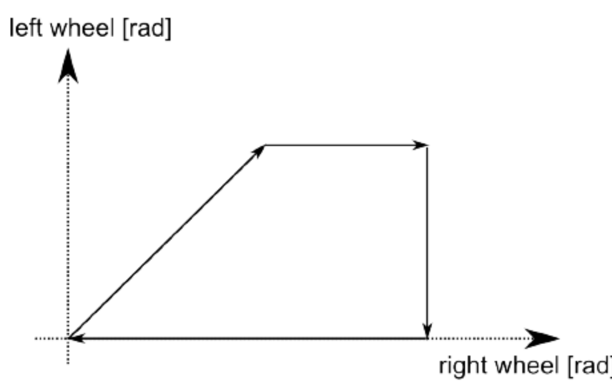
Holonomic vs. Non-Holonomic

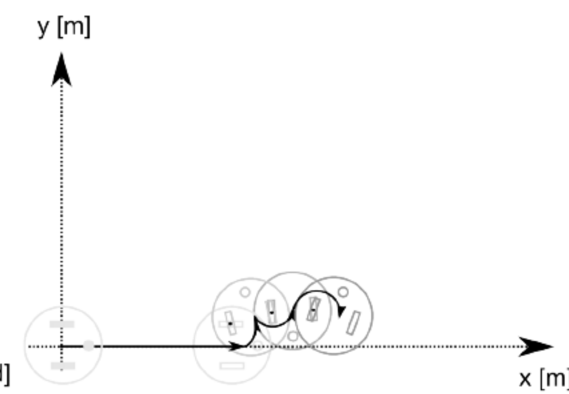
Manipulator





Diff. Wheels





Configuration Space

Workspace

Holonomic or Non-Holonomic on the 2D plane?

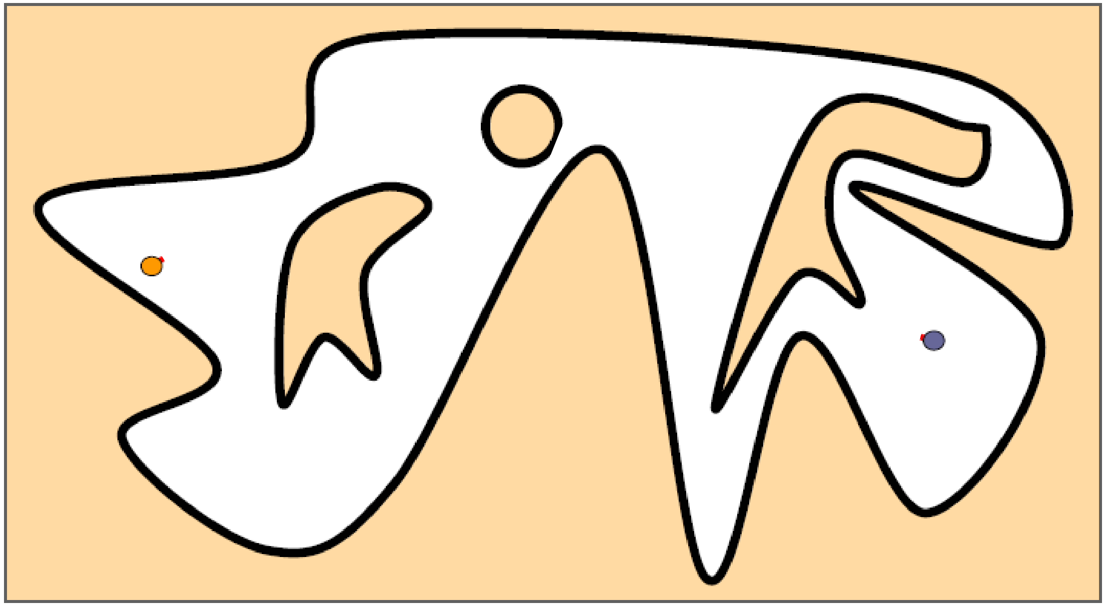
Steering wheel is rotated 90 degrees then acceleration is applied for 1 second

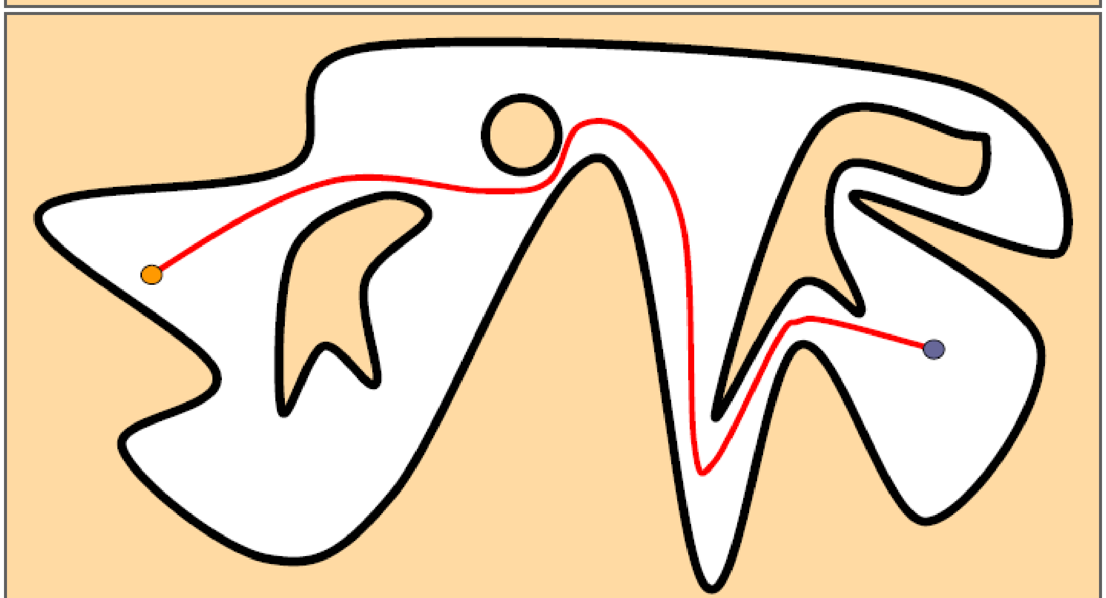
vs.

Acceleration is applied for 1 second then steering wheel rotated 90 degrees!

Different ending configuration = Non-holonomic!

Moving from start to goal state

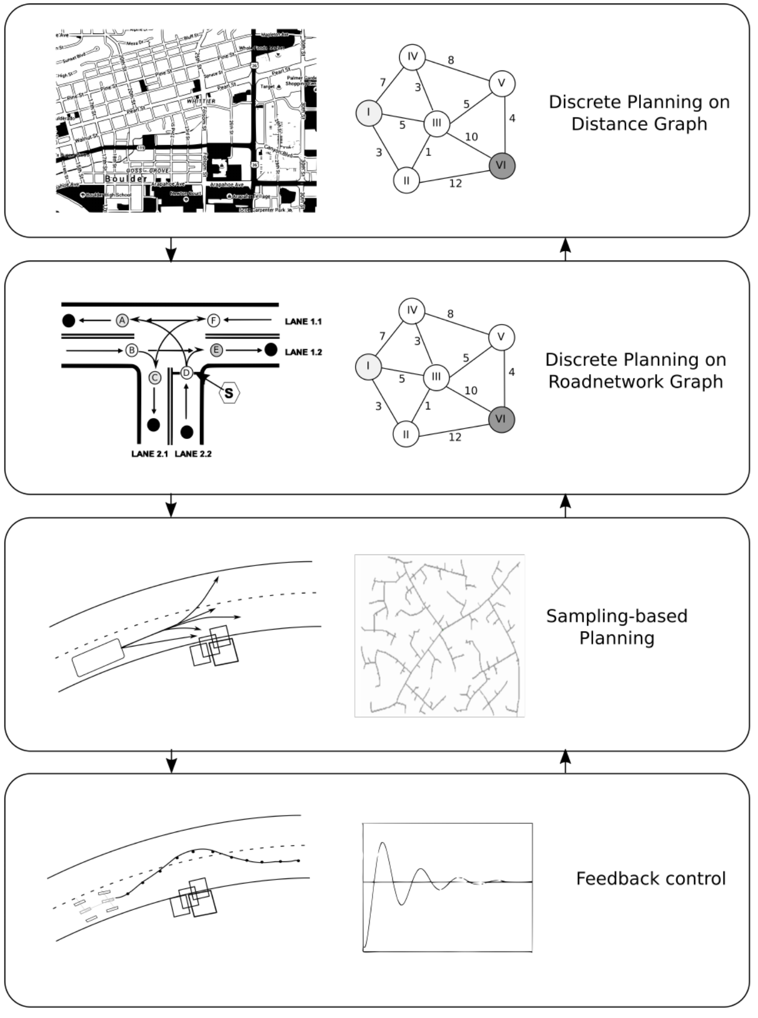




General Approach:
Reduce what amounts to an intractable problem in continuous C-space to a tractable problem in a discrete space.

Tools:
- Environment Representations
- Search Algorithms

Next Time:
Planning across length scales



Project Inspiration Reading Assignment

Computational Human-Robot Interaction

http://guyhoffman.com/publications/ThomazHoffmanCakmak16.pdf

Survey of recent papers in the field:

Serve as inspiration for final projects

Catalyst for choosing special topics to cover

For Next Week:

- Install **Ubuntu 16.04:** http://releases.ubuntu.com/16.04/
- Install **ROS Kinetic:** http://wiki.ros.org/kinetic/Installation
- (Optional) Go through the **ROS Beginner Tutorials**: http://wiki.ros.org/ROS/Tutorials