Algorithmic Human-Robot Interaction

Wrap-up

CSCI 7000

Prof. Brad Hayes

University of Colorado Boulder

Final Papers

Due Tuesday at 11:59pm

- No presentation required – Online submission only

Mandatory Files

- Zip file titled “Paper.zip”
  - Final PDF
  - LaTeX / Word files
  - Figures
  - Link to a video presentation (up to 10min)
    - Narration over slides is preferred
- Zip file titled “Code.zip”
  - Checkout of your (up-to-date) Git repository
  - README file including instructions to run your project

Highly Encouraged

- Video demonstration of results in your presentation

Most Useful Topics?

- Motion Planning
- Trajectory Optimization
- Learning from Demonstration
- Task Planning (STRIPS, search, GRAPHPLAN, heuristics, etc.)
- Models (MDPs, POMDPs, etc.)
- Project Design (System Architecture / Algorithm Design)
- Experimental Design
- Natural Language Processing (G^3)
- ROS Tutorial (Quadcopter example)
- Inverse Reinforcement Learning
- Reinforcement Learning
- Paper Presentation and Critique

Least Useful Topics?

- Motion Planning
- Trajectory Optimization
- Learning from Demonstration
- Task Planning (STRIPS, search, GRAPHPLAN, heuristics, etc.)
- Models (MDPs, POMDPs, etc.)
- Project Design (System Architecture / Algorithm Design)
- Experimental Design
- Natural Language Processing (G^3)
- ROS Tutorial (Quadcopter example)
- Inverse Reinforcement Learning
- Reinforcement Learning
- Paper Presentation and Critique

Most Interesting Papers?

1. Trajectories and Keyframes for Kinesthetic Teaching: A Human-Robot Interaction Perspective
2. Planning human-aware motions using a sampling-based costmap planner
3. Designing Robot Learners that Ask Good Questions
4. Anticipating Human Actions for Collaboration in the Presence of Task and Sensor Uncertainty
5. Probabilistically Safe Robot Planning with Confidence-Based Human Predictions
6. An Implemented Theory of Mind to Improve Human-Robot Shared Plans Execution
7. Planning for Autonomous Cars that Leverage Effects on Human Actions
8. Game-Theoretic Modeling of Human Adaptation in Human-Robot Collaboration
9. Robust Robot Learning from Demonstration and Skill Repair Using Conceptual Constraints
10. Accurately and Efficiently Interpreting Human-Robot Instructions of Varying Granularities
11. Learning Robot Objectives from Physical Human Interaction
12. Expressing Robot Incapability
13. Balanced Information Gathering and Goal-Oriented Actions in Shared Autonomy
14. Transfer depends on Acquisition: Analyzing Manipulation Strategies for Robotic Feeding
15. Improving Robot Controller Transparency Through Autonomous Policy Explanation
16. Explanation-based Reward Coaching to Improve Human Performance via Reinforcement Learning
17. Shared Autonomy via Deep Reinforcement Learning
18. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning

Least Interesting Papers?

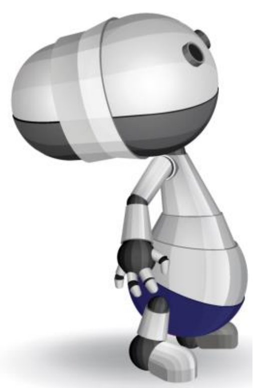
1. Trajectories and Keyframes for Kinesthetic Teaching: A Human-Robot Interaction Perspective
2. Planning human-aware motions using a sampling-based costmap planner
3. Designing Robot Learners that Ask Good Questions
4. Anticipating Human Actions for Collaboration in the Presence of Task and Sensor Uncertainty
5. Probabilistically Safe Robot Planning with Confidence-Based Human Predictions
6. An Implemented Theory of Mind to Improve Human-Robot Shared Plans Execution
7. Planning for Autonomous Cars that Leverage Effects on Human Actions
8. Game-Theoretic Modeling of Human Adaptation in Human-Robot Collaboration
9. Robust Robot Learning from Demonstration and Skill Repair Using Conceptual Constraints
10. Accurately and Efficiently Interpreting Human-Robot Instructions of Varying Granularities
11. Learning Robot Objectives from Physical Human Interaction
12. Expressing Robot Incapability
13. Balanced Information Gathering and Goal-Oriented Actions in Shared Autonomy
14. Transfer depends on Acquisition: Analyzing Manipulation Strategies for Robotic Feeding
15. Improving Robot Controller Transparency Through Autonomous Policy Explanation
16. Explanation-based Reward Coaching to Improve Human Performance via Reinforcement Learning
17. Shared Autonomy via Deep Reinforcement Learning
18. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning

Course Retrospective

We covered **a lot** of material this semester!

Course Content

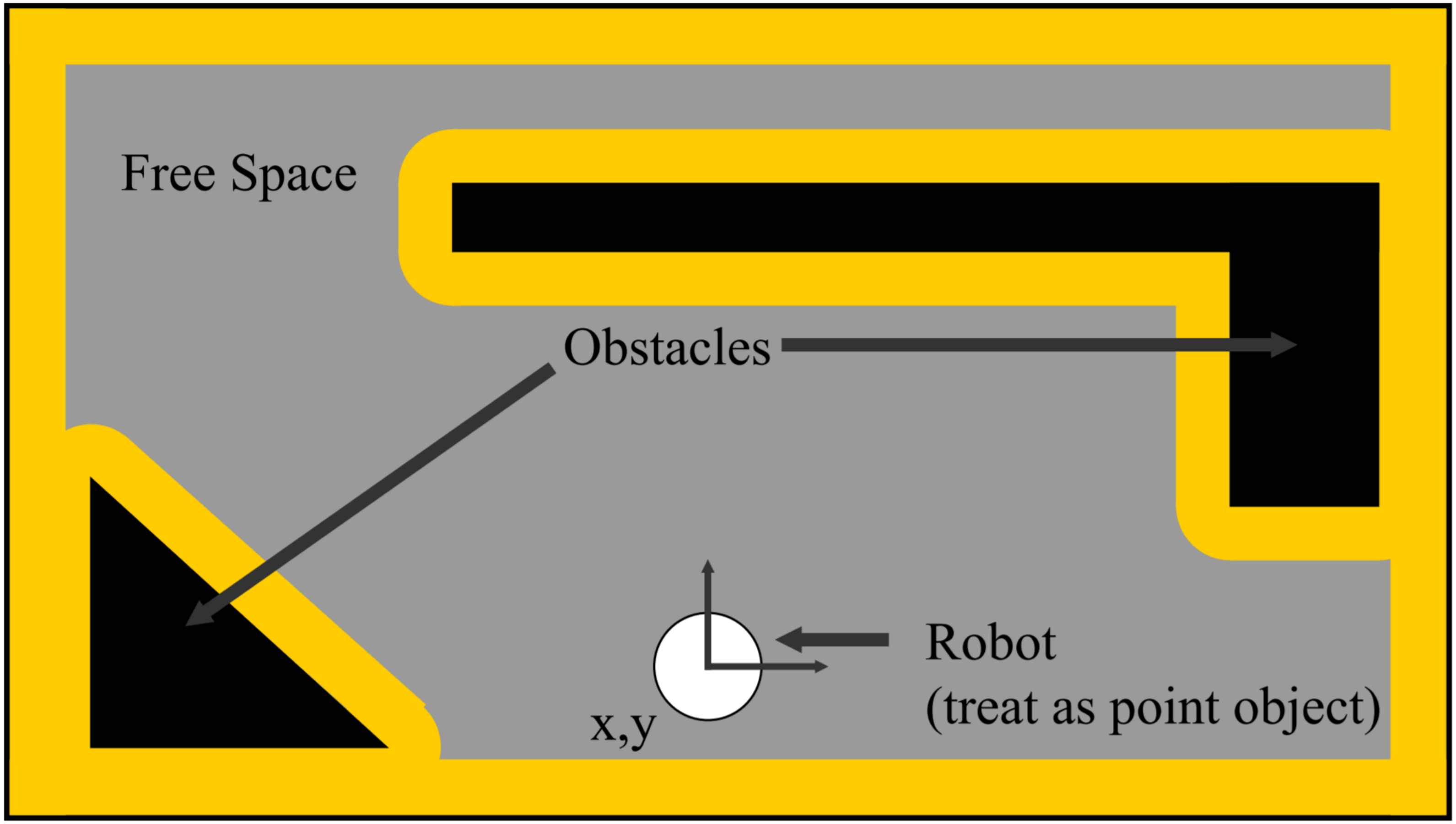
We will study fundamental topics in robotics and psychology/cognitive science with the objective of introducing robustness to human interaction to the former and automation to the latter.



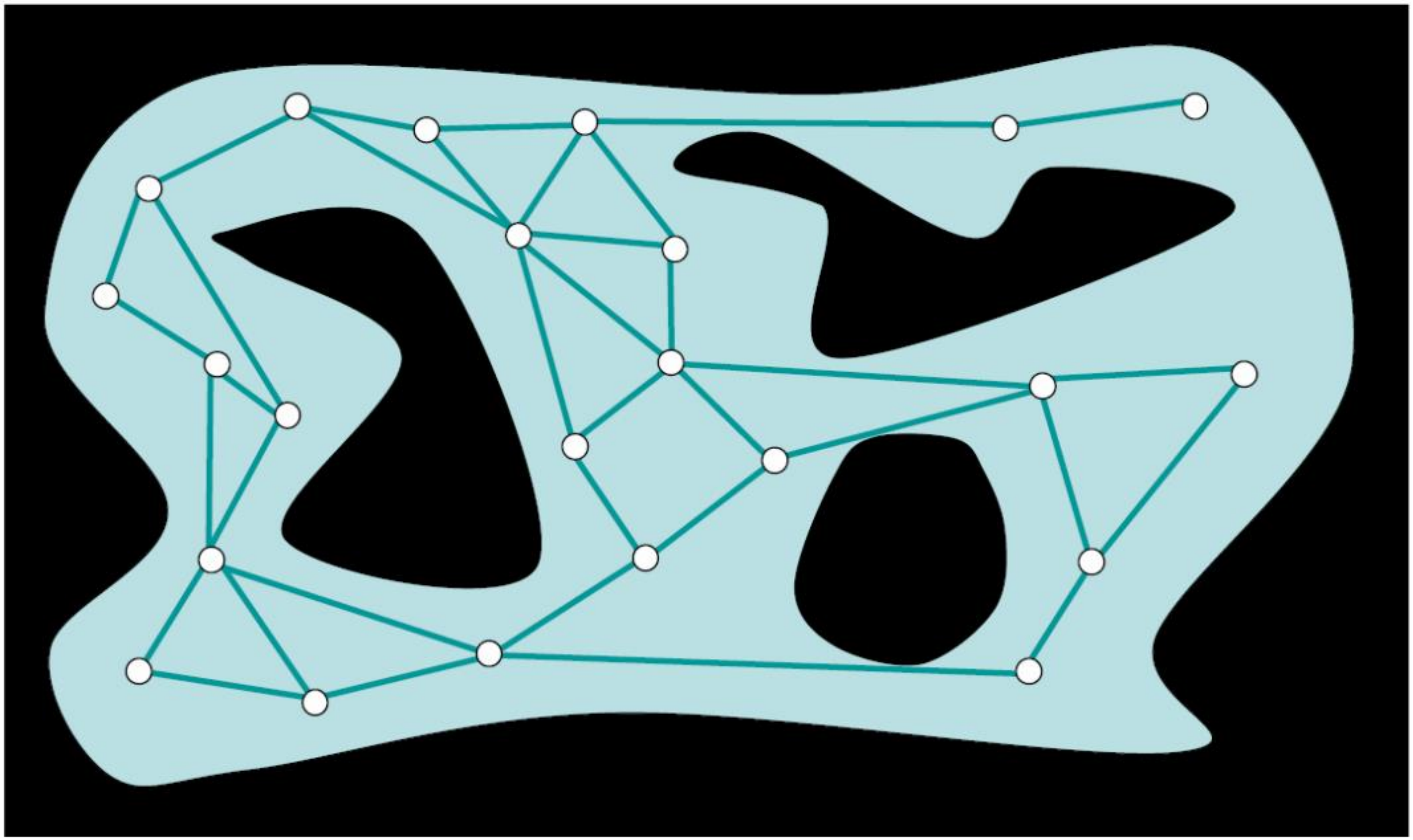
Course Content

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

Configuration Space



Probabilistic Road Map (PRM)

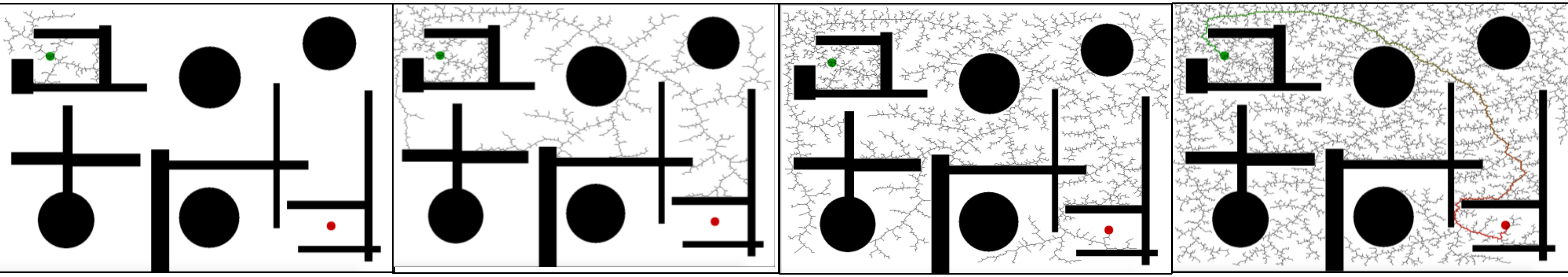


Remove edges crossing forbidden areas

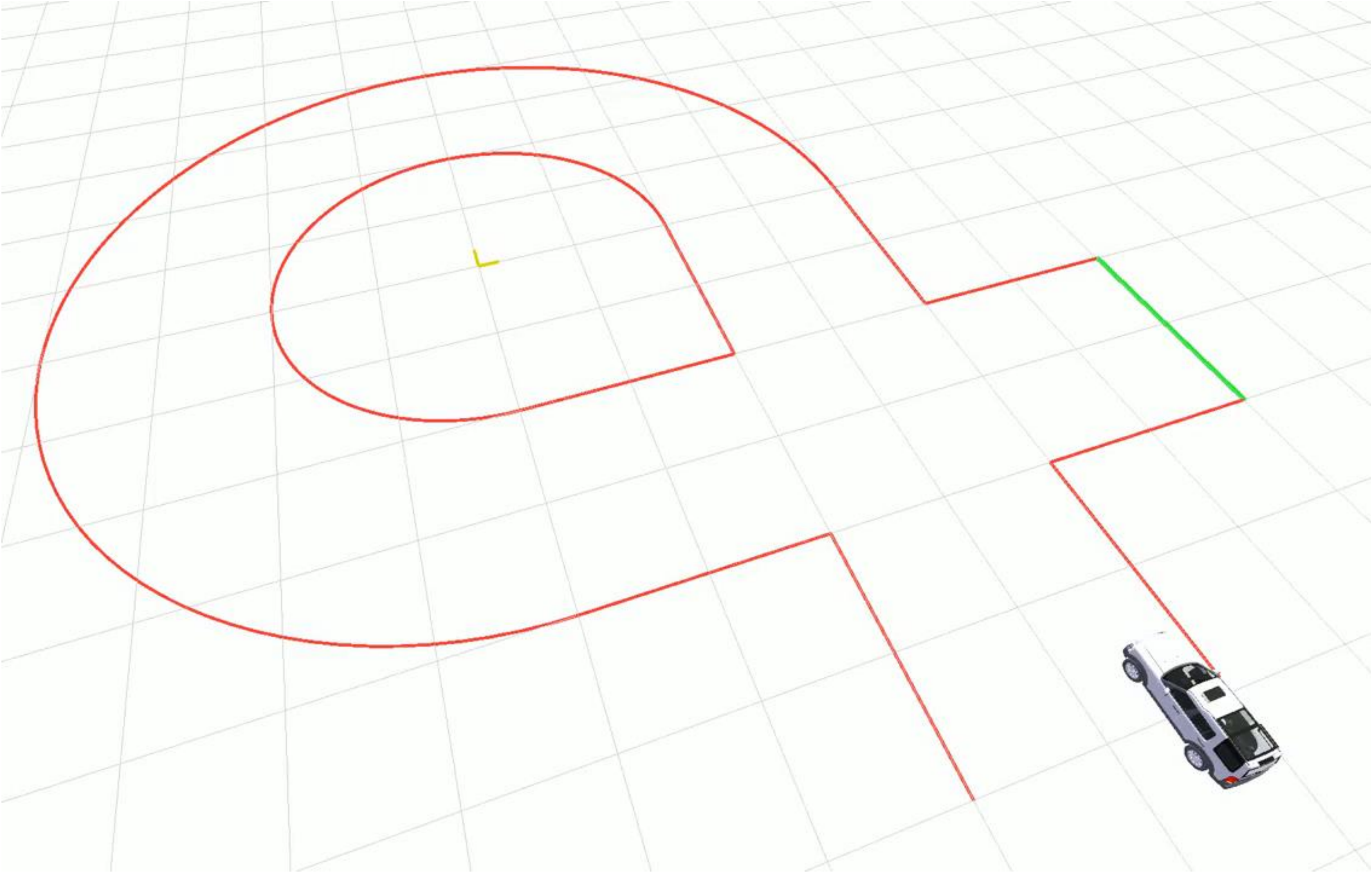
How to sample points?

- Uniformly randomly
- Sample more near places with few neighbors
- Bias samples to exist near obstacles
- Use human-provided waypoints
- Something better?

Rapidly-exploring Random Trees (RRT)



RRT*



Trajectory Optimization:

Problem Statement

- Trajectory $\xi : t \in [0, T] \rightarrow C$ *Maps time to configurations*

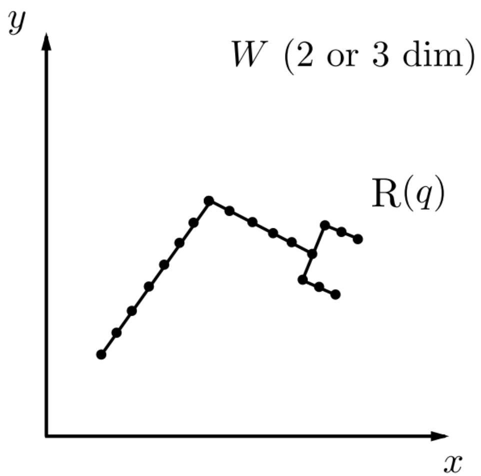
- Objective Functional $U : \Xi \rightarrow \mathbb{R}^{+}$ *Maps trajectories to scalars*

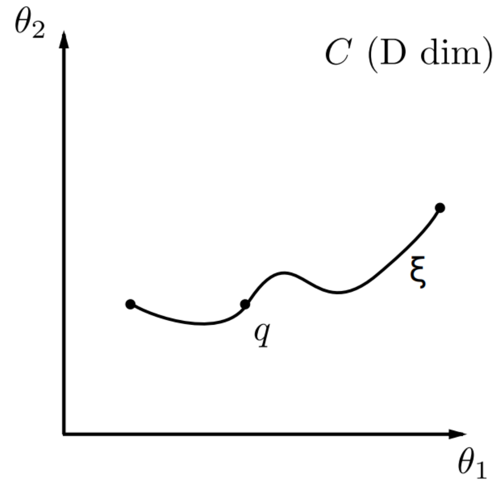
Set of possible trajectories

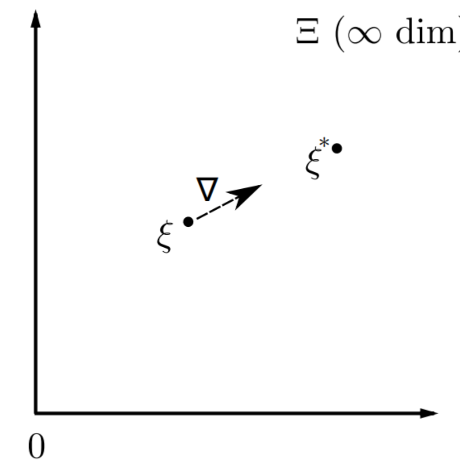
- The objective U encodes traits we want our path to have
  - Path length
  - Efficiency
  - Obstacle avoidance
  - Legibility
  - Uncertainty reduction
  - Human comfort

Goal: Leverage the benefits of randomized sampling with asymptotic optimality

Problem Specification: Spaces







World Space $W(\mathbb{R}^3)$	Configuration Space $C(\#DoF)$	Trajectory Space $\Xi(\infty\ dim)$

Robot pose in World Space (set of points) → Single point in Configuration Space

Trajectory through Configuration Space (set of points) → Single point in Trajectory Space

Making Trajectory Optimization Useful

Need to provide a good choice for $\mathbf{U}[\xi]$.

CHOMP: Covariant Hamiltonian Optimization for Motion Planning

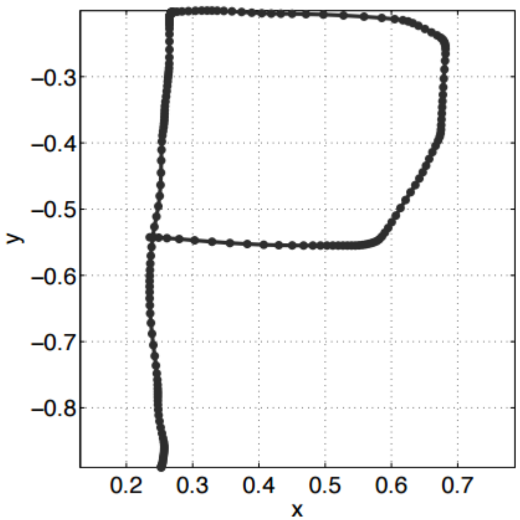
Uses a cost function $\boldsymbol{U[\xi] = U_{smooth}[\xi] + \lambda U_{obs}[\xi]}$

Smoothness cost: $\boldsymbol{U_{smooth}[\xi] = \frac{1}{2}\int_0^T ||\xi'(t)||^2\, dt}$

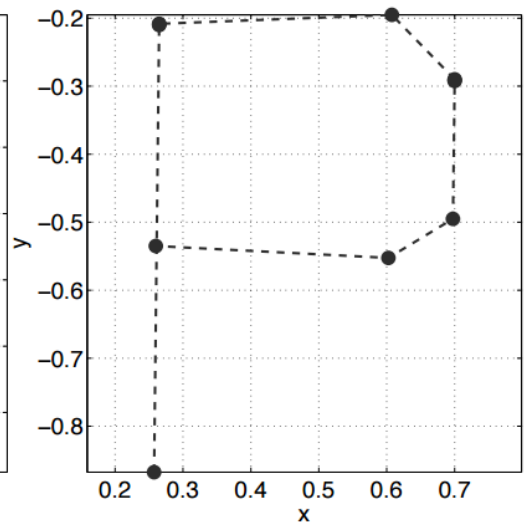
Obstacle cost: $\boldsymbol{U_{obs}[\xi] = \int_t \int_u c\left(\phi_u(\xi(t))\right) * \left\|\frac{d}{dt}\phi_u(\xi(t))\right\| du dt}$

Cost function that computes distance to closest obstacle	Forward Kinematics function that computes location of robot body point u at time t in ξ	Norm of the velocity for body point u at time t in ξ

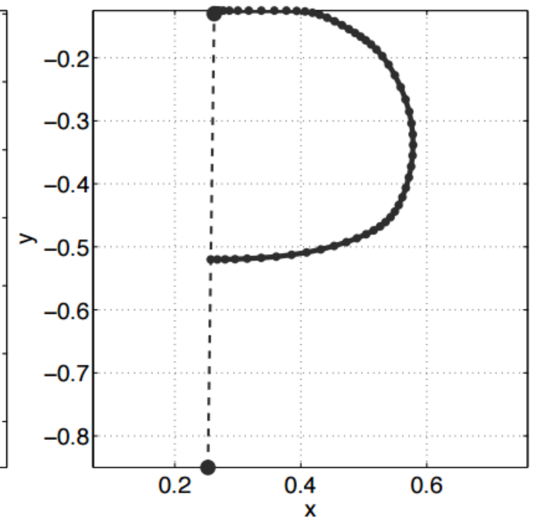
Trajectories and Keyframes for Kinesthetic Teaching: A Human-Robot Interaction Perspective



Trajectory Demonstration



Keyframe Demonstration



Hybrid Demonstration

Sample demonstrations of the letter P in 2D

Planning: High Level

- Thinking before acting
- Determining how to achieve a given goal

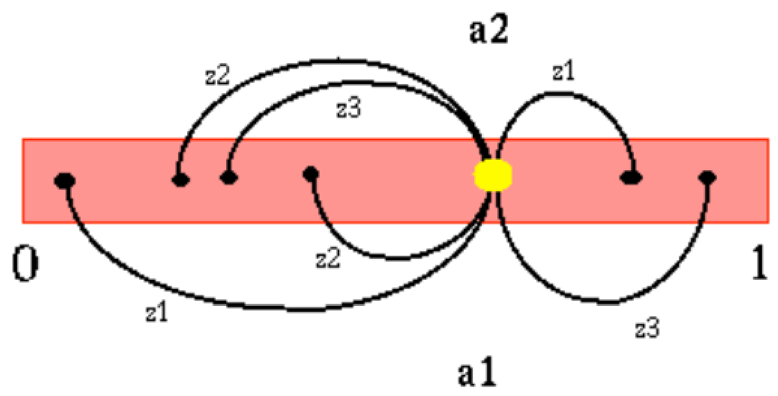
Three approaches to the control problem (what to do next):

1. Programming-based: Specify control by hand
2. Learning-based: Learn control from experience

Inverse Reinforcement Learning (MaxEnt IRL)	Reinforcement Learning (SARSA, REINFORCE)

3. Model-based (Planning): Derive control from a domain model

POMDP: Trivial Example



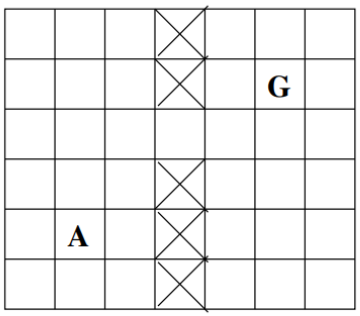
Two states: {0,1}   Two Actions: {a1, a2}

Three Observations: {z1, z2, z3}

- A dot’s position in the red bar indicates our belief over these states. (Yellow is current belief)
- B = [p, 1-p] indicates $p\%$ chance of being in State 0, and $1 - p\%$ chance of being in State 1.
- Executing a1 and observing z3 tells us that we’re very likely to be in State 1
- Executing a1 and observing z1 tells us that we’re very likely to be in State 0

Identifying Types of Planning Problems

Agent **A** must reach **G**, moving one cell at a time in **known** map



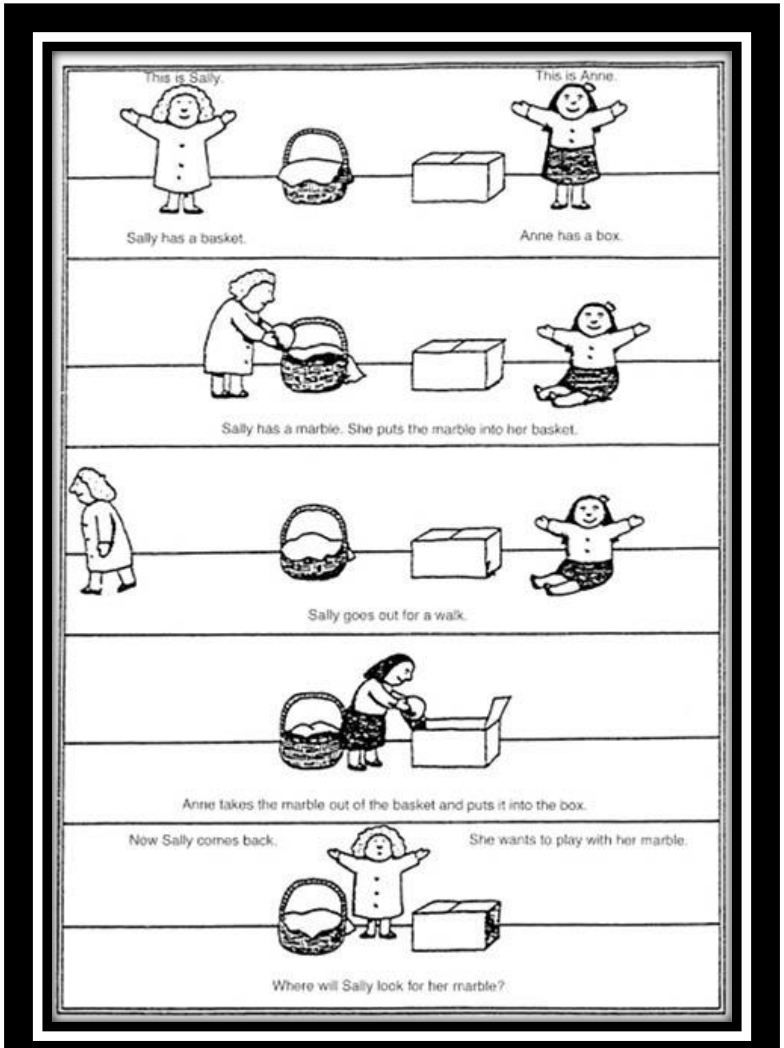
- If actions deterministic and initial location known, planning problem is **Classical**
- If actions non-deterministic and location observable, it's an **MDP** or **FOND**
- If actions non-deterministic and location partially obs, **POMDP** or **Contingent**

Different combinations of uncertainty and feedback: diff problems, diff models

Planner is generic solver for instances of a particular model

Classical planners, MDP Planners, POMDP planners, . . .

Sally-Anne Test of False Belief

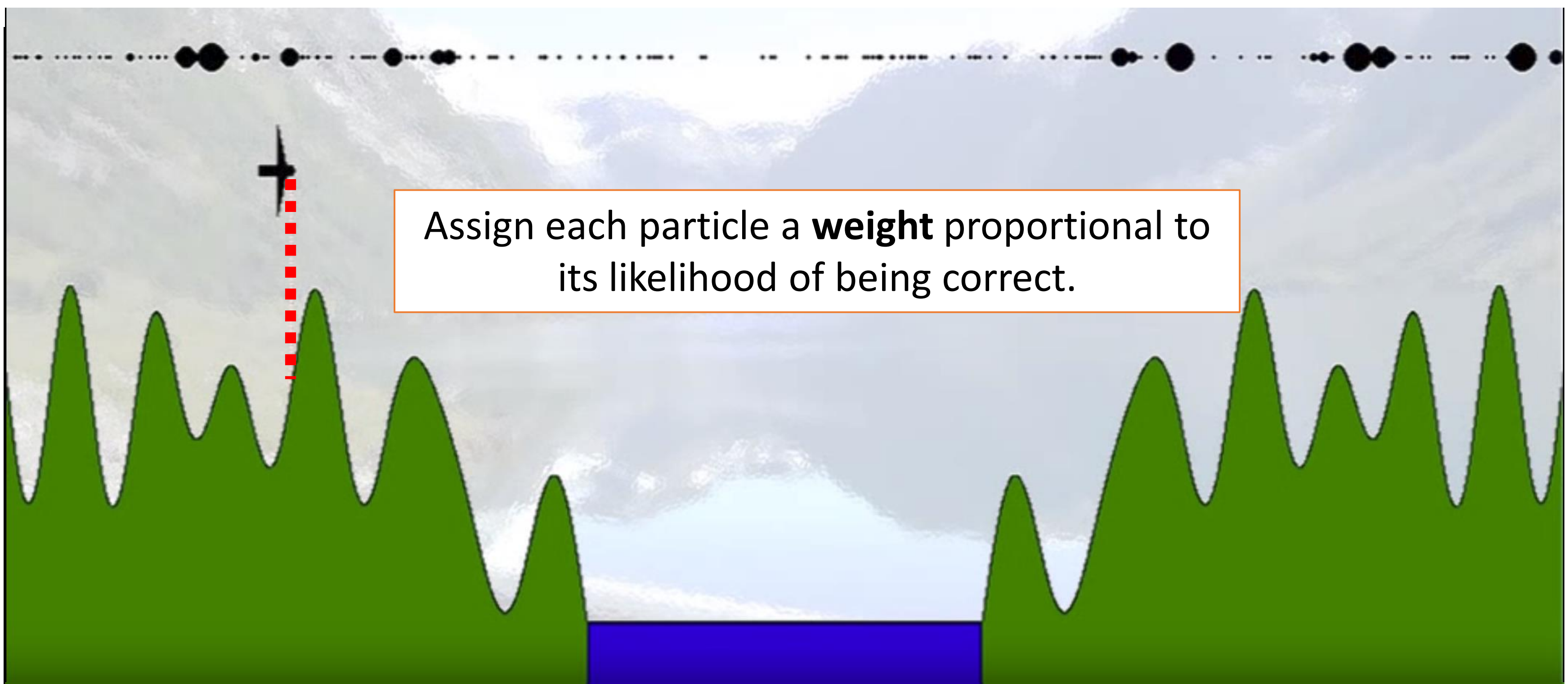


Children under the age of four do not do well on this test!

Particle Filter: Weighting Particles

- **Latent variable:** Horizontal aircraft position
- **Observable variable:** Vertical aircraft position (altitude)
- **Known information:** Airspeed, Map
- **Goal:** Use repeated observations to find true horizontal position

Main Idea:
Generate lots of hypotheses and let the observations determine their likelihood.



Markov Model Chart

Do we have control over the state transitions?
(Are we picking which actions are executed)

Are the states completely observable?

	NO	**YES**
YES	Markov Chain	MDP
NO	HMM	POMDP

Learning an HMM's Parameters

Learning: Given $\boldsymbol{O}$ and $\boldsymbol{S}$ … Determine $\boldsymbol{A}, \boldsymbol{B}$

Challenge: Must simultaneously determine **transition probabilities** AND **emission probabilities**!

Special case of Expectation-Maximization, iteratively improving an initial estimate.

But first, let's solve for a Markov Chain (*fully observable*) given $\boldsymbol{O}, \boldsymbol{S}, \boldsymbol{Q}$

Forward-Backward: Learning B

Now we need to compute observation emission probability:

$$\hat{b}_j(v_k) = \frac{Expected\ \#\ of\ v_k\ seen\ in\ state\ j}{Expected\ \#times\ in\ state\ j} = \frac{\sum_{t=1}^{T} \gamma_t(j) * I(o_t = v_k)}{\sum_{t=1}^{T} \gamma_t(j)}$$

$\gamma_t(j) =$ prob. of being in state j at time t

$$\gamma_t(j) = \frac{\alpha_t(j) * \beta_t(j)}{P(O|A,B)}$$

Forward

$$\boxed{\alpha_t(j) = P(o_1, o_2, \dots, o_t, q_t = j \mid A, B)}$$

Backward

$$\boxed{\beta_t(i) = P(o_{t+1}, o_{t+2}, \dots o_t | q_t = i, A, B)}$$

Expectation-Maximization on A, B

E-Step: Compute state occupancy count γ, expected state transition count ξ using existing A, B probabilities

M-Step: Compute A, B using existing γ and ξ probabilities

$\alpha_t(j)$ = prob. to be in state j at t

$\beta_t(j)$ = prob. of O from state j at t

$\xi_t(i,j)$ = prob. of transition from i to j at time t

$\gamma_t(j)$ = prob. of being in state j at time t

function FORWARD-BACKWARD(*observations* of len T, *output vocabulary* V, *hidden state set* Q) **returns** *HMM=(A,B)*

initialize A and B
iterate until convergence
E-step

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{\alpha_T(q_F)} \quad \forall\, t \text{ and } j$$

$$\xi_t(i,j) = \frac{\alpha_t(i)\, a_{ij} b_j(o_{t+1})\beta_{t+1}(j)}{\alpha_T(q_F)} \quad \forall\, t,\ i, \text{ and } j$$

M-step

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \sum_{k=1}^{N} \xi_t(i,k)}$$

$$\hat{b}_j(v_k) = \frac{\sum_{t=1 s.t.\ O_t=v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

return A, B

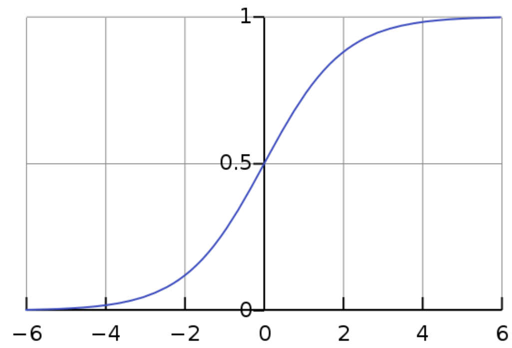
Introduction to Machine Learning

- Regression: How much is this house worth?
- Classification: Is this a photo of a sheepdog or a mop?

Linear Regression to Logistic Regression

- Linear Regression gives us a continuous-valued function approximation
  - Models relationship between scalar dependent variable y and one or more variables X
- Logistic regression allows us to approximate **categorical** data
  - Pick a model function that squashes values between 0 and 1

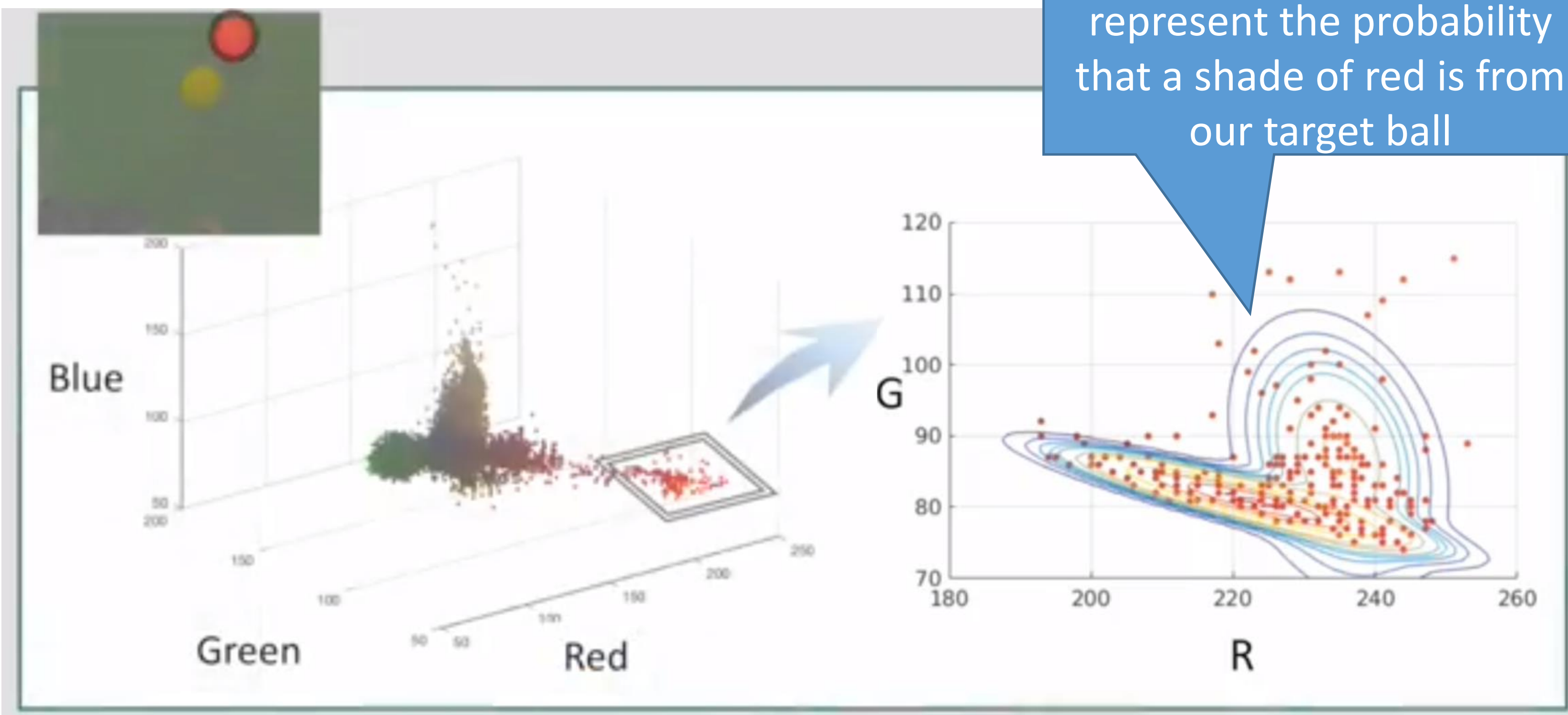
$$F(x) = \frac{1}{1 + e^{-x}}$$



  - Apply it to a familiar function: $g(X) = \beta_0 + \beta_1 x + \epsilon$

$$P(Y=1) = F(g(x)) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 * x)}}$$

Example: Color Filtering

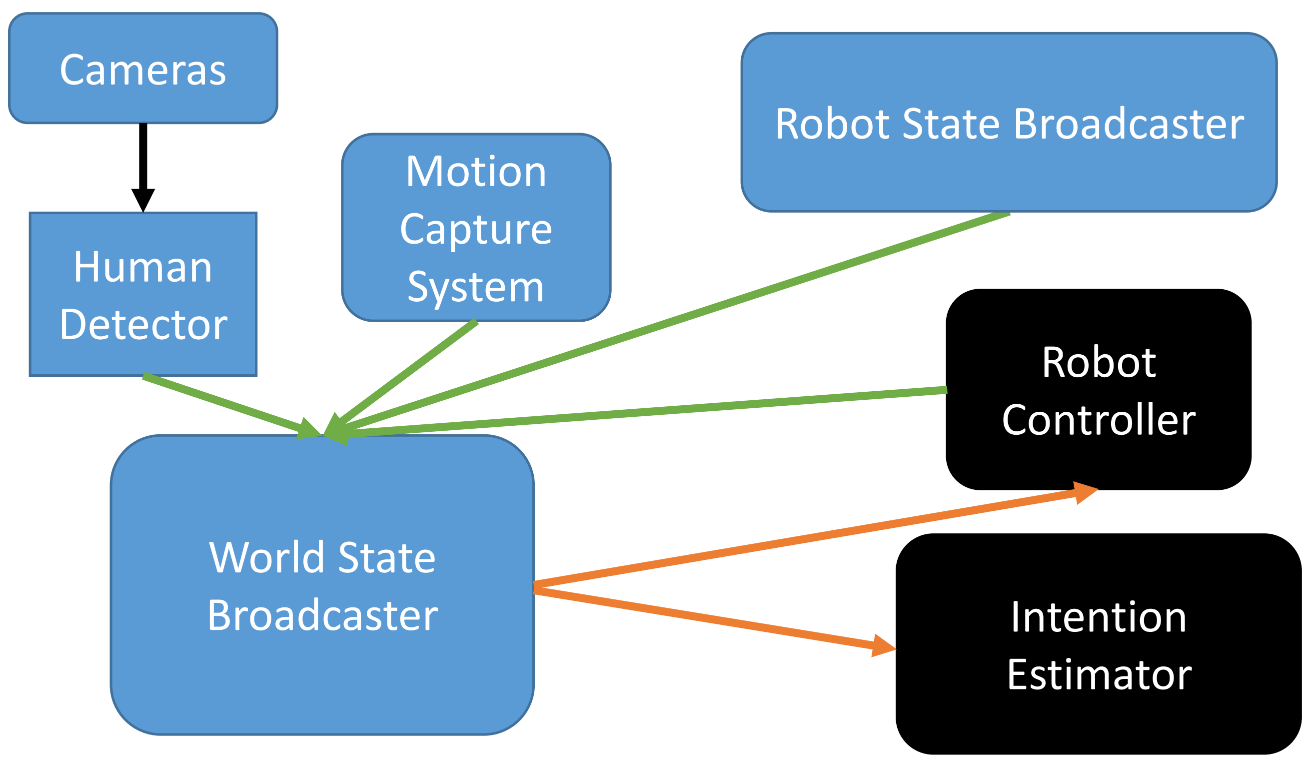


A mixture of two 2D Gaussians can **better** represent the probability that a shade of red is from our target ball

Designing Your System

What is the state your system acts within?
What are the features, and where do they come from?

Modular design is essential!



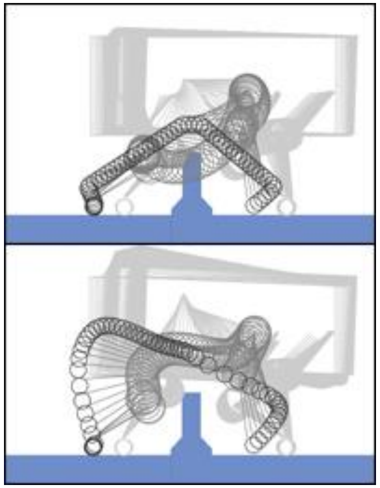
Generally	Evaluate treatment effectiveness Understanding human behavior
HRI	Evaluate a design or interaction strategy Compare algorithm performance

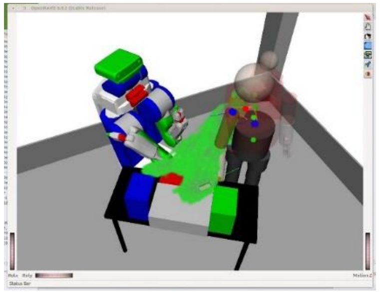
Why Run an Experiment?

Good Experiments have Factorial Design

Unclear if result will be because of constraint type or the optimization order!

Need to run all conditions to isolate independent variable effects.

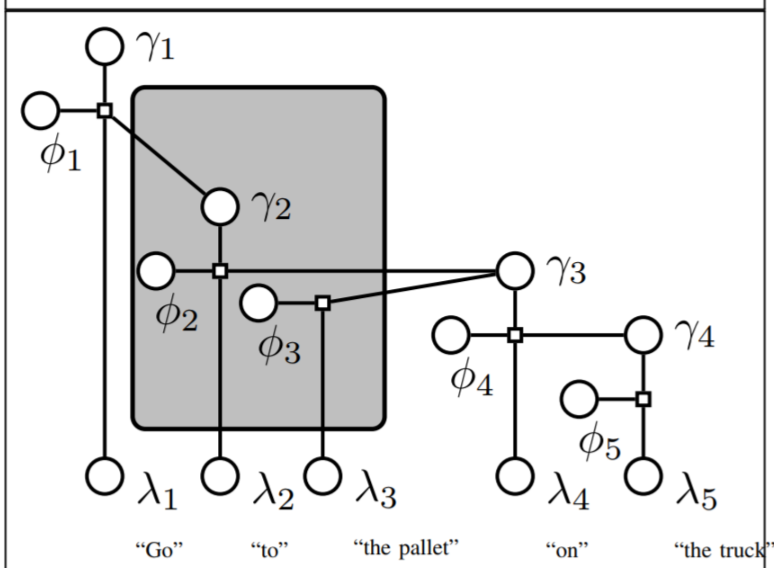




Constraint type/order of optimization	1^{st} order	2^{nd} order
Soft	CHOMP	x
Hard	x	trajOPT

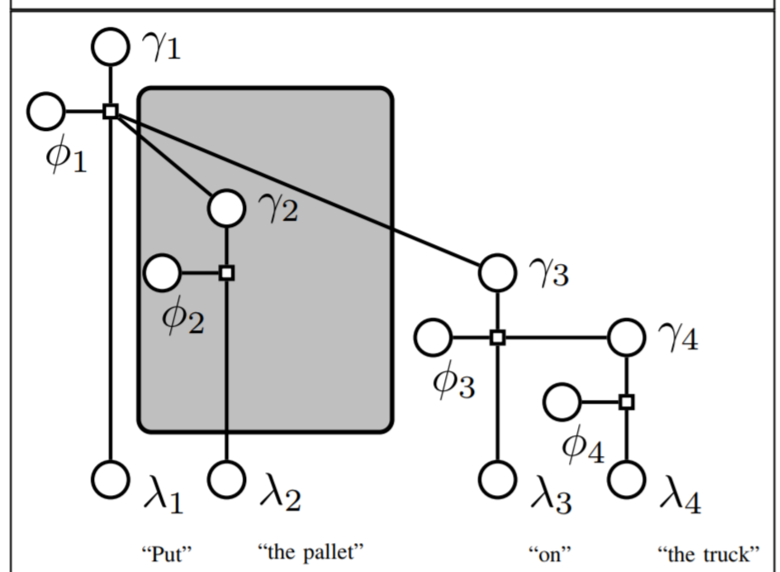
G^3 Examples

```
(ROOT (S (VP (VB Go)
          (PP (TO to)
              (NP (NP (DT the) (NN pallet))
                  (PP (IN on)
                  (NP (DT the) (NN truck))))))
 (.  .)))
```



(b) "Go to the pallet on the truck."

```
(ROOT (S (VP (VB Put)
             (NP (DT the) (NN pallet))
             (PP (IN on)
                 (NP (DT the) (NN truck)))
 (.  .)))
```



(a) "Put the pallet on the truck."

ROS: Robot Operating System

Available at http://www.ros.org/

- Current Version on Lab Machines: Kinetic Kame
- Download Ubuntu 16.04 LTS image and install on VM
- http://wiki.ros.org/kinetic/Installation
- Tutorials will get you up to speed quickly!
  - http://wiki.ros.org/ROS/Tutorials

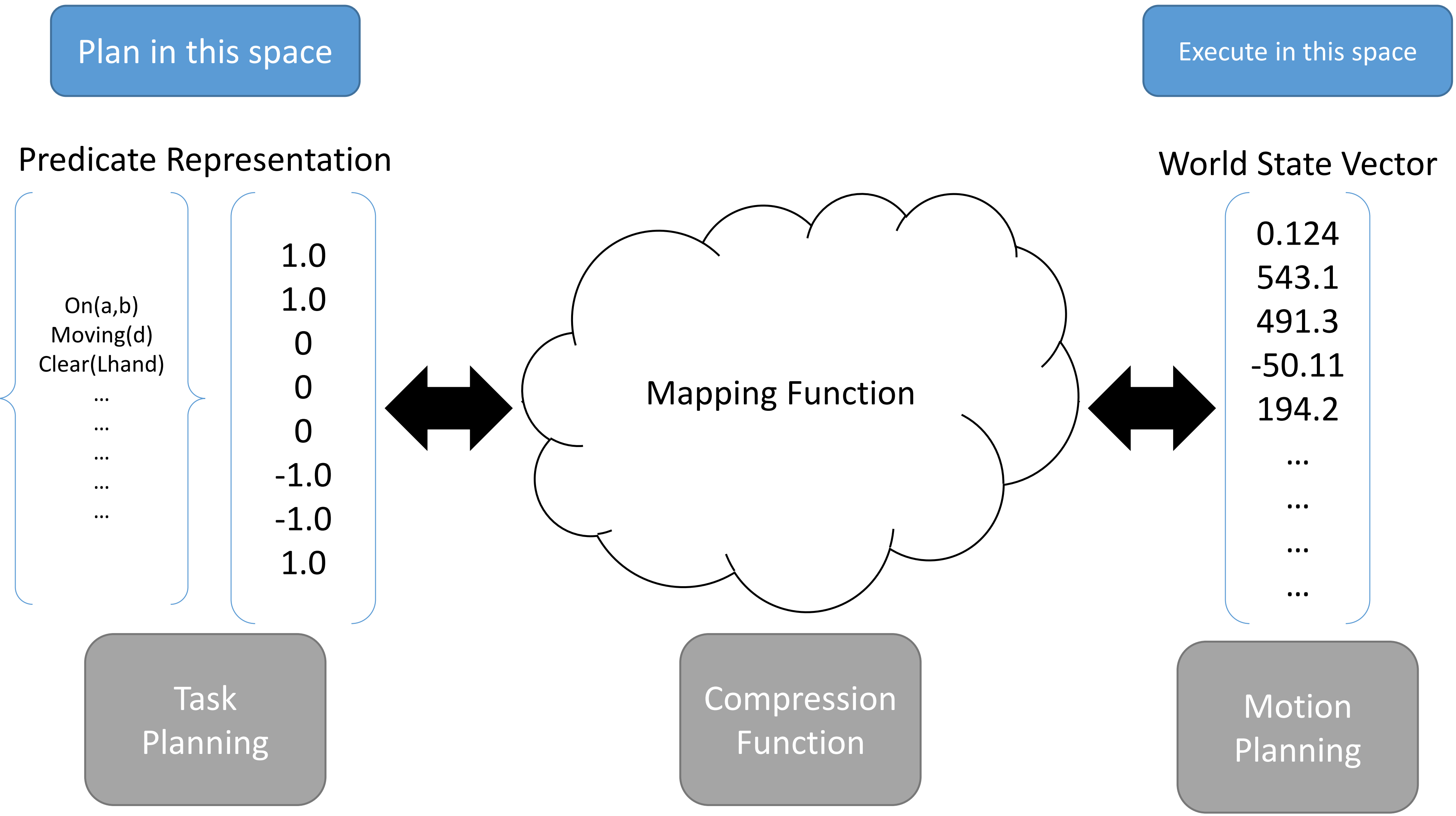
Moving the Camera

> rostopic pub /bebop/camera_control geometry_msgs/Twist
'{linear: {x: 0, y: 0, z: 0}, angular: {x: 0, y: 0, z: 0}}'

Publish a geometry_msgs/Twist message to
/bebop/camera_control to "move" the camera

Angular y: [-90, 90]   -90 = Down, +90 = Forward, (0 = Mostly Forward)

Lifted vs. Grounded Planning



Donald Michie's criteria for Machine Learning (ML)

Weak criterion:

ML occurs whenever a system generates an updated basis building on sample data for improving its performance on subsequent data.

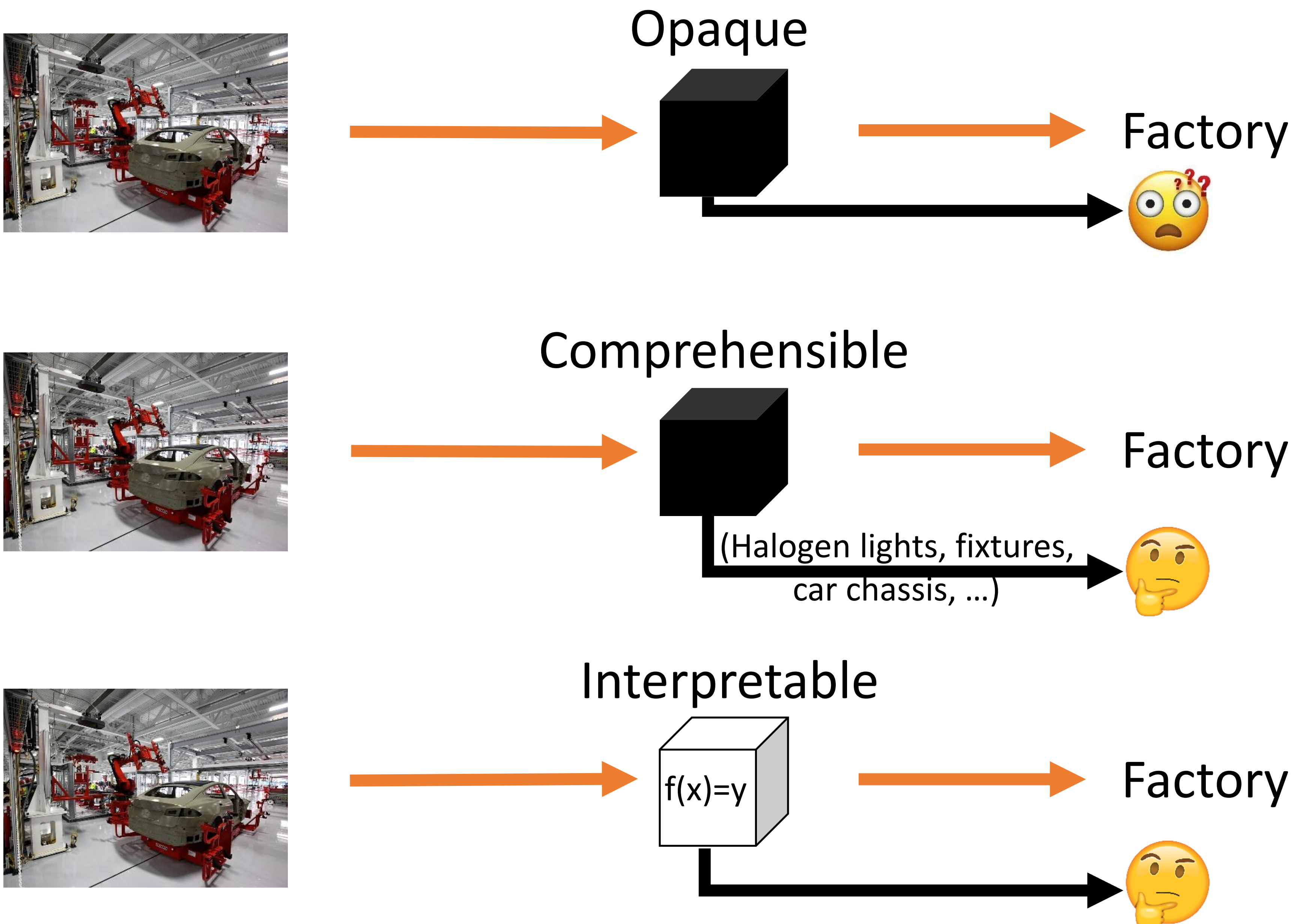
Strong criterion:

Weak criterion + ability of system to communicate internal updates in explicit symbolic form.

Ultra-strong criterion:

Strong criterion + communication of updates must be operationally effective (i.e. user is required to understand updates and consequences should be drawn from it).

Relating Different Types of Systems



Reinforcement Learning

Function: $O \rightarrow A$

$\mathbf{o}$ $\pi_\theta(\mathbf{a}|\mathbf{o})$ $\mathbf{a}$

$\mathbf{s}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{a}_t$ – action

$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ – policy
$\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ – policy (fully observed)

$\mathbf{o}_t$ – observation

$\mathbf{s}_t$ – state

Policy Differentiation

$$\theta^\star = \arg\max_\theta \underbrace{\sum_{t=1}^{T} E_{(\mathbf{s}_t,\mathbf{a}_t)\sim p_\theta(\mathbf{s}_t,\mathbf{a}_t)}[r(\mathbf{s}_t,\mathbf{a}_t)]}_{J(\theta)}$$

$$\underbrace{\pi_\theta(\mathbf{s}_1,\mathbf{a}_1,\ldots,\mathbf{s}_T,\mathbf{a}_T)}_{\pi_\theta(\tau)} = p(\mathbf{s}_1)\prod_{t=1}^{T}\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)p(\mathbf{s}_{t+1}|\mathbf{s}_t,\mathbf{a}_t)$$

$$J(\theta) = E_{\tau\sim\pi_\theta(\tau)}[r(\tau)] = \int \pi_\theta(\tau)r(\tau)d\tau$$

$$\pi_\theta(\tau)\nabla_\theta \log\pi_\theta(\tau) = \pi_\theta(\tau)\frac{\nabla_\theta\pi_\theta(\tau)}{\pi_\theta(\tau)} = \nabla_\theta\pi_\theta(\tau)$$

$$\nabla_\theta J(\theta) = \int \nabla_\theta\pi_\theta(\tau)r(\tau)d\tau = \int \pi_\theta(\tau)\nabla_\theta\log\pi_\theta(\tau)r(\tau)d\tau = E_{\tau\sim\pi_\theta(\tau)}[\nabla_\theta\log\pi_\theta(\tau)r(\tau)]$$

$$\boxed{\nabla_\theta J(\theta) = E_{\tau\sim\pi_\theta(\tau)}\left[\left(\sum_{t=1}^{T}\nabla_\theta\log\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)\right)\left(\sum_{t=1}^{T} r(\mathbf{s}_t,\mathbf{a}_t)\right)\right]}$$

$$\nabla_\theta\overbrace{\left[\cancel{\log p(\mathbf{s}_1)} + \sum_{t=1}^{T}\log\pi_\theta(\mathbf{a}_t|\mathbf{s}_t) + \cancel{\log p(\mathbf{s}_{t+1}|\mathbf{s}_t,\mathbf{a}_t)}\right]}$$

MaxEnt IRL

1. Initialize ψ (reward function params), gather demonstrations D
2. Solve for optimal policy $\pi(a|s)$ with reward r_ψ
3. Solve for state visitation frequencies $p(s|\psi)$
4. Compute gradient: $\nabla_\psi L = -\frac{1}{|D|}\sum_{\tau_d \in D} \frac{dr_\psi}{d\psi}(\tau_d) - \sum_s p(s|\psi)\frac{dr_\psi}{d\psi}(s)$
5. Update ψ with one gradient step using $\nabla_\psi L$
6. GOTO 2

Must solve the whole MDP in the inner loop of findin the reward function!

Syllabus Promises

Articulate	Articulate challenges in building autonomous systems that interact with humans
Apply	Apply machine learning techniques to enable human-robot collaboration
Develop	Develop an understanding of computational models of verbal and non-verbal communication
Learn	Apply learning from demonstration techniques to enable robots to acquire new capabilities and more rapidly generalize existing ones
Communicate	Effectively communicate scientific content
Critique	Critique scientific literature with respect to experimental design and analysis
Implement	Implement control algorithms on real robot hardware to design real autonomous systems

Where do I go from here?