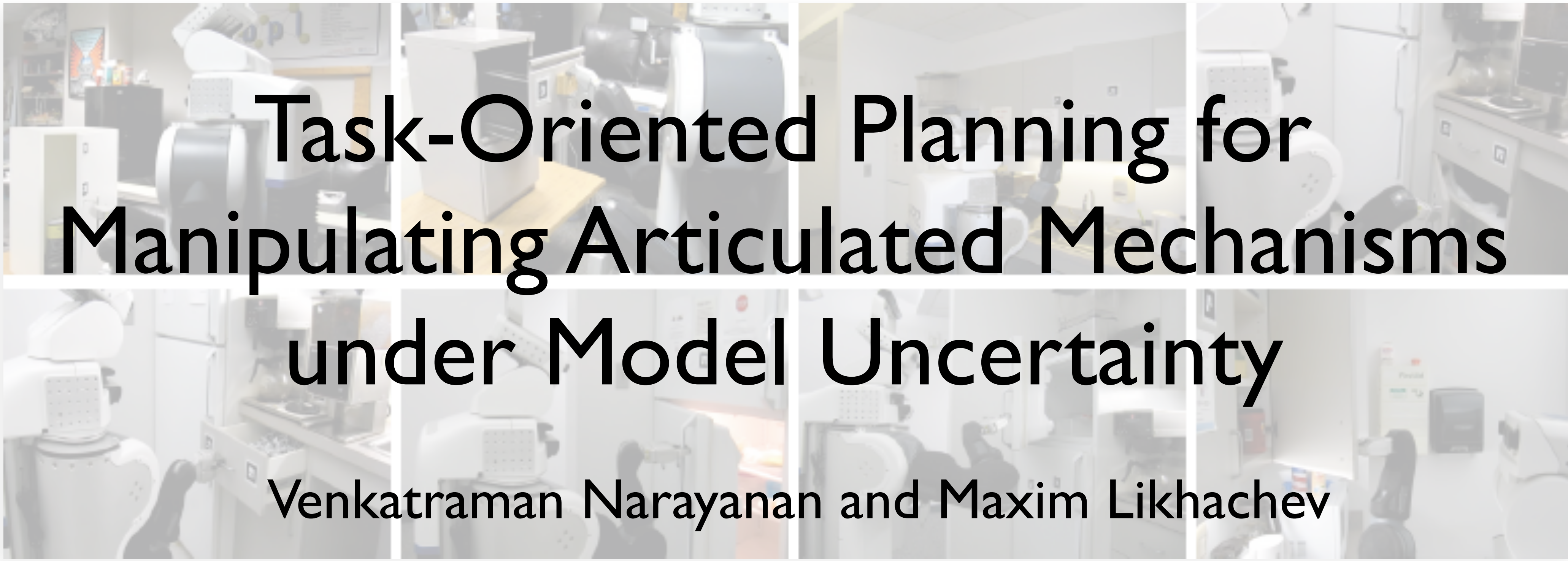


Task-Oriented Planning for Manipulating Articulated Mechanisms under Model Uncertainty

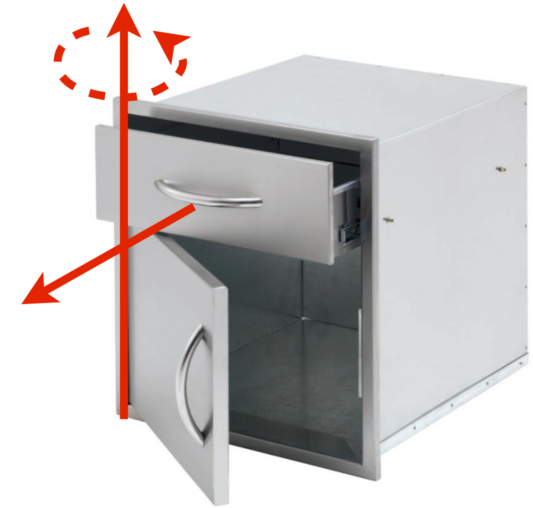
Venkatraman Narayanan and Maxim Likhachev

THE ROBOTICS INSTITUTE

Carnegie Mellon
THE ROBOTICS INSTITUTE

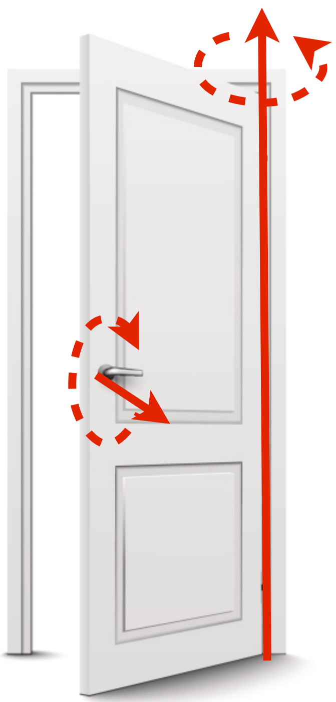
Motivation

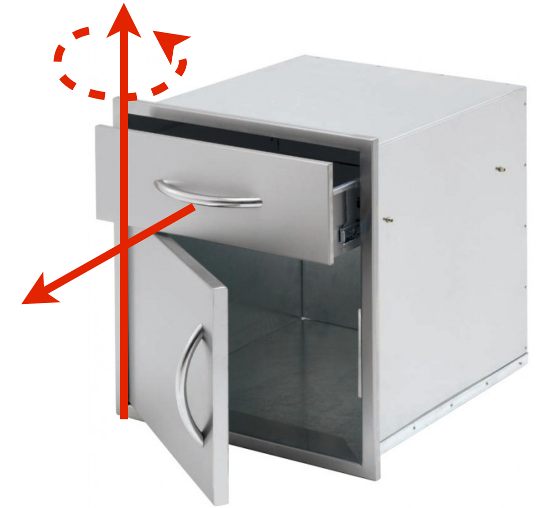
Motivation



A lot of household objects are

ARTICULATED





A lot of household objects are

ARTICULATED

Robot manipulation is typically

TASK-DRIVEN

Outline

Representation

Problem Formulation

Related Work

Planning under Model Uncertainty

Perceptual Grounding

Experiments

Outline

Representation

Problem Formulation

Related Work

Planning under Model Uncertainty

Perceptual Grounding

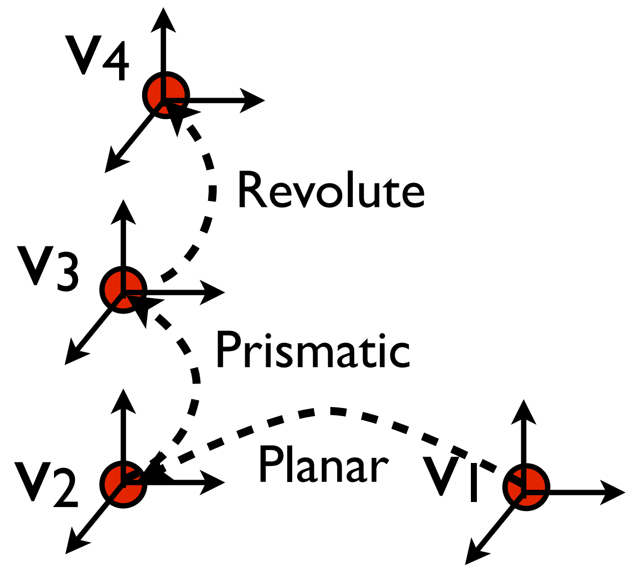
Experiments

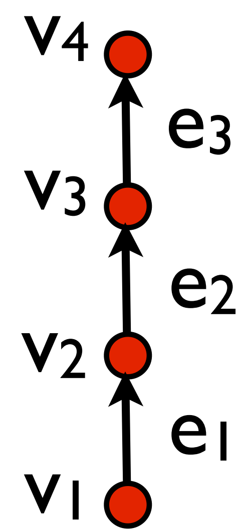
Representation

Kinematic Graph

Representation

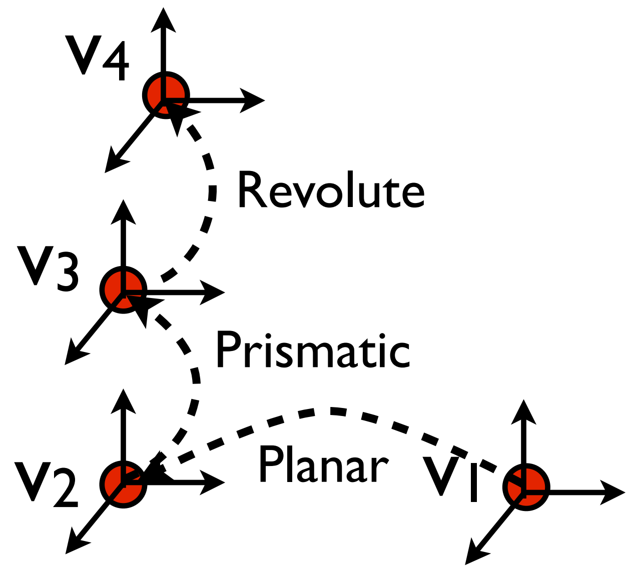
Kinematic Graph

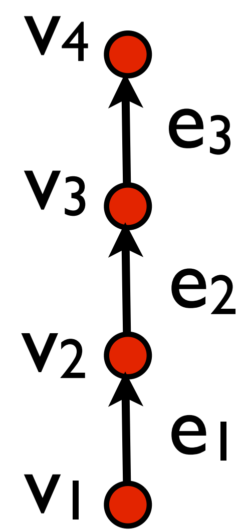




Representation

Kinematic Graph





Vertex v : 6 DoF pose

Edge e : Tuple $\langle m, \beta \rangle$

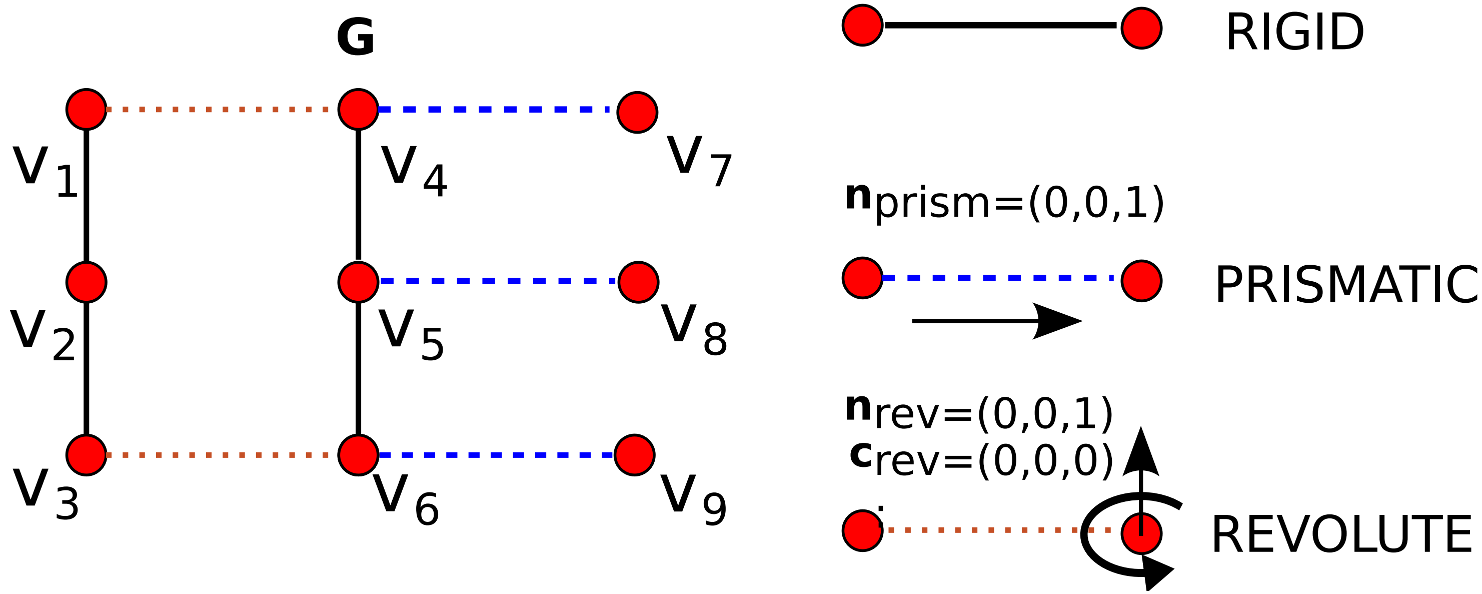
$$G = (V, E)$$

model parameters: axis direction, joint limits, ...

model: {prismatic, revolute, spherical, ...}

Representation

Kinematic Graph



Vertex v : 6 DoF pose

Edge e : Tuple $\langle m, \beta \rangle$

$$G = (V, E)$$

model parameters: axis direction, joint limits, ...

model: {prismatic, revolute, spherical, ...}

Representation

Kinematic Graph

$\text{Vertex } v : 6 \text{ DoF pose}$

$\text{Edge } e : \text{Tuple } \langle m, \beta \rangle$

$$G = (V, E)$$

model parameters: axis direction, joint limits, ...

model: {prismatic, revolute, spherical, ...}

Representation

Kinematic Graph

Vertex v : 6 DoF pose

Edge e : Tuple $\langle m, \beta \rangle$

$$G = (V, E)$$

β → model parameters: axis direction, joint limits, ...

m → model: {prismatic, revolute, spherical, ...}

Generalized Kinematic Graph (GK-Graph)

Allow edges to be a function of vertices: $e = f(V)$

Expressive representation captures complex articulations

Representation

Generalized Kinematic Graph (GK-Graph)

Allow edges to be a function of vertices: $e = f(V)$

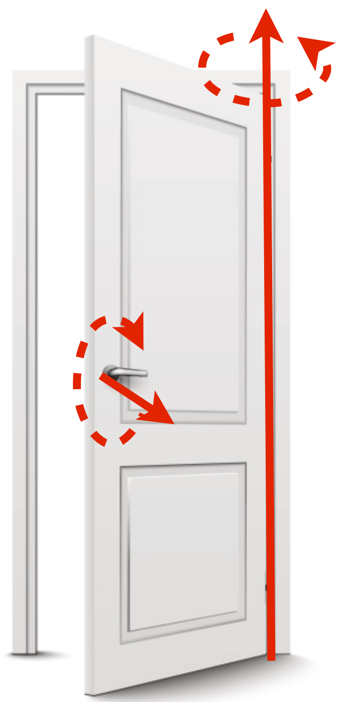
Expressive representation captures complex articulations

Representation

Generalized Kinematic Graph (GK-Graph)

Allow edges to be a function of vertices: $e = f(V)$

Expressive representation captures complex articulations



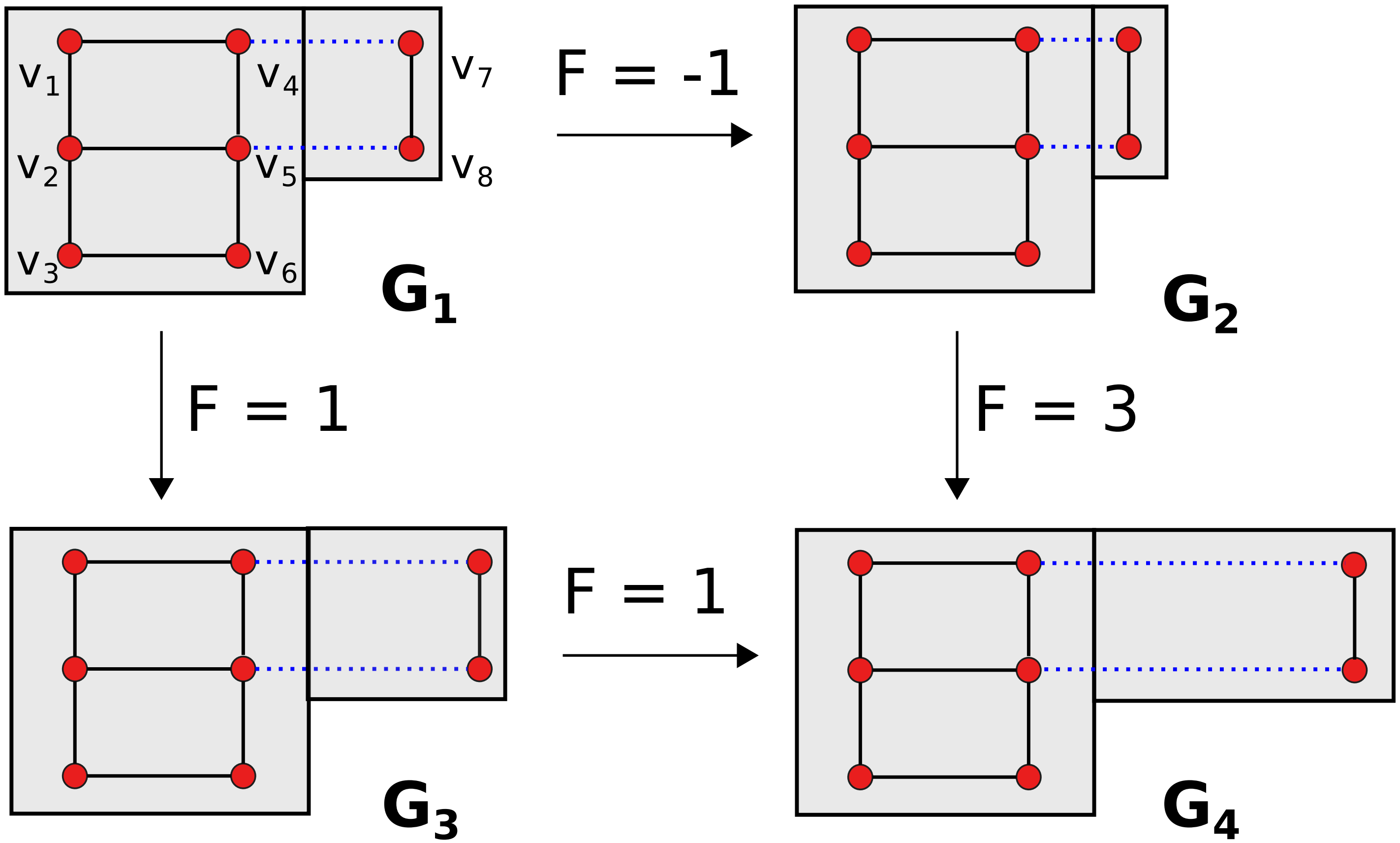
Able to represent conditions such as

Door-Wall joint is rigid only when handle is
a) unturned, and
b) in the plane of the door frame

Otherwise, it is revolute

Representation

Planning with the GK-Graph



Outline

Representation

Problem Formulation

Related Work

Planning under Model Uncertainty

Perceptual Grounding

Experiments

Outline

Representation

Problem Formulation

Related Work

Planning under Model Uncertainty

Perceptual Grounding

Experiments

Problem Formulation

Given N candidate articulation models, find a cost-minimal policy to achieve the goal

Problem Formulation

Given N candidate articulation models, find a cost-minimal policy to achieve the goal

Each candidate model is a hypothesis of how the object operates

“the door opens if you turn the handle and push”
“the door opens if you turn the handle and pull”
“the door opens if you slide it across”

Problem Formulation

Given N candidate articulation models, find a cost-minimal policy to achieve the goal

Each candidate model is a hypothesis of how the object operates

The goal is some function of the kinematic graph

"the object's joint limits have been reached"
"the handle has moved X cm from where it was"
"the camera can see what is inside"

Problem Formulation

Given N candidate articulation models, find a cost-minimal policy to achieve the goal

Each candidate model is a hypothesis of how the object operates

The goal is some function of the kinematic graph

User-defined cost, e.g, get to the goal as quickly as possible

Policy: mapping from what the robot sees and its uncertainty over candidate models to an action it can execute

Problem Formulation

Given N candidate articulation models, find a cost-minimal policy to achieve the goal

Each candidate model is a hypothesis of how the object operates

The goal is some function of the kinematic graph

User-defined cost, e.g, get to the goal as quickly as possible

Policy: mapping from what the robot sees and its uncertainty over candidate models to an action it can execute

Outline

Representation

Problem Formulation

Related Work

Planning under Model Uncertainty

Perceptual Grounding

Experiments

Outline

Representation

Problem Formulation

Related Work

Planning under Model Uncertainty

Perceptual Grounding

Experiments

Related Work

Learning Articulation

Katz and Brock, ICRA ’08
Katz et al. ICRA ‘13
Sturm et al. IJCAI ’09, Springer ’13

Learn Articulation Quickly

Katz et al. RSS ’08
Barragan et al. ICRA ’14
Otte et al. IROS ’14

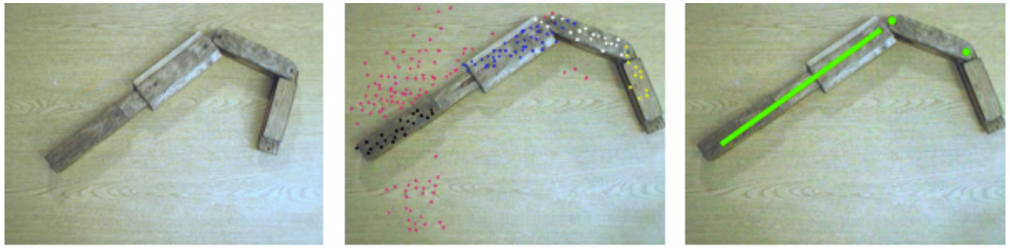
Controller-based Approaches

Jain and Kemp, AURO ‘13
Ruhr et al. ICRA ‘12
Sturm et al. IROS ‘10

Related Work

Learning Articulation

Katz and Brock, ICRA ’08
Katz et al. ICRA ‘13
Sturm et al. IJCAI ’09, Springer ’13



Learn Articulation Quickly

Katz et al. RSS ’08
Barragan et al. ICRA ’14
Otte et al. IROS ’14

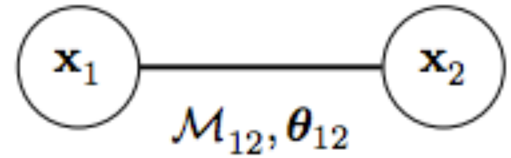
Controller-based Approaches

Jain and Kemp, AURO ‘13
Ruhr et al. ICRA ‘12
Sturm et al. IROS ‘10

Related Work

Learning Articulation

Katz and Brock, ICRA '08
Katz et al. ICRA '13
Sturm et al. IJCAI '09, Springer '13



$$(\hat{\mathcal{M}}_{ij}, \hat{\boldsymbol{\theta}}_{ij}) = \underset{\mathcal{M}_{ij}, \boldsymbol{\theta}_{ij}}{\arg\max}\, p(\mathcal{M}_{ij}, \boldsymbol{\theta}_{ij} \mid \mathcal{D}_{\mathbf{z}_{ij}}).$$

Learn Articulation Quickly

Katz et al. RSS '08
Barragan et al. ICRA '14
Otte et al. IROS '14

(a) Cabinet door

Controller-based Approaches

Jain and Kemp, AURO '13
Ruhr et al. ICRA '12
Sturm et al. IROS '10

(b) Dishwasher door

Related Work

Learning Articulation
Katz and Brock, ICRA ’08
Katz et al. ICRA ‘13
Sturm et al. IJCAI ’09, Springer ’13

Learn Articulation Quickly
Katz et al. RSS ’08
Barragan et al. ICRA ’14
Otte et al. IROS ’14

Controller-based Approaches
Jain and Kemp, AURO ‘13
Ruhr et al. ICRA ‘12
Sturm et al. IROS ‘10

Theme: Select actions to minimize entropy in distribution over model parameters/degrees of freedom

Related Work

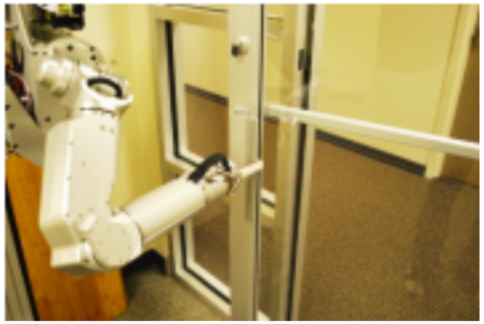
Learning Articulation
Katz and Brock, ICRA ’08
Katz et al. ICRA ‘13
Sturm et al. IJCAI ’09, Springer ’13

Learn Articulation Quickly
Katz et al. RSS ’08
Barragan et al. ICRA ’14
Otte et al. IROS ’14

Controller-based Approaches
Jain and Kemp, AURO ‘13
Ruhr et al. ICRA ‘12
Sturm et al. IROS ‘10

$$(c, r)^* = argmin(\sum_t (\|x_{ee}[t] - c\| - r)^2)$$

Equilibrium point control couple with continuous mechanism estimation



Related Work

Learning Articulation

Katz and Brock, ICRA '08
Katz et al. ICRA '13
Sturm et al. IJCAI '09, Springer '13

Learn Articulation Quickly

Katz et al. RSS '08
Barragan et al. ICRA '14
Otte et al. IROS '14

Controller-based Approaches

Jain and Kemp, AURO '13
Ruhr et al. ICRA '12
Sturm et al. IROS '10

Related Work

Learning Articulation

Katz and Brock, ICRA ’08
Katz et al. ICRA ‘13
Sturm et al. IJCAI ’09, Springer ’13

Learn Articulation Quickly

Katz et al. RSS ’08
Barragan et al. ICRA ’14
Otte et al. IROS ’14

Controller-based Approaches

Jain and Kemp, AURO ‘13
Ruhr et al. ICRA ‘12
Sturm et al. IROS ‘10

Related Work

Learning Articulation

Katz and Brock, ICRA ’08
Katz et al. ICRA ‘13
Sturm et al. IJCAI ’09, Springer ’13

Learn Articulation Quickly

Katz et al. RSS ’08
Barragan et al. ICRA ’14
Otte et al. IROS ’14

Controller-based Approaches

Jain and Kemp, AURO ‘13
Ruhr et al. ICRA ‘12
Sturm et al. IROS ‘10

Related Work

Learning Articulation

Katz and Brock, ICRA ’08
Katz et al. ICRA ‘13
Sturm et al. IJCAI ’09, Springer ’13

Learn Articulation Quickly

Katz et al. RSS ’08
Barragan et al. ICRA ’14
Otte et al. IROS ’14

Controller-based Approaches

Jain and Kemp, AURO ‘13
Ruhr et al. ICRA ‘12
Sturm et al. IROS ‘10

Key Contributions of this Work:

1. Task-oriented: approach as a planning problem as opposed to a learning problem

2. Novel (and perceptually grounded) representation for articulated objects: GK-Graph

Outline

Representation

Problem Formulation

Related Work

Planning under Model Uncertainty

Perceptual Grounding

Experiments

Outline

Representation

Problem Formulation

Related Work

Planning under Model Uncertainty

Perceptual Grounding

Experiments

Planning under Model Uncertainty

Given N candidate articulation models, find a cost-minimal policy to achieve the goal

Planning under Model Uncertainty

Given N candidate articulation models, find a cost-minimal policy to achieve the goal

Some notation:

Set of vertices in GK-Graph: $x \in \mathcal{X}$

Candidate models: $f_\theta(x), \theta = \{1, 2, \ldots, N\}$

Action: $a \in A$

State: $s \in \mathcal{S} : \langle x, \theta \rangle$

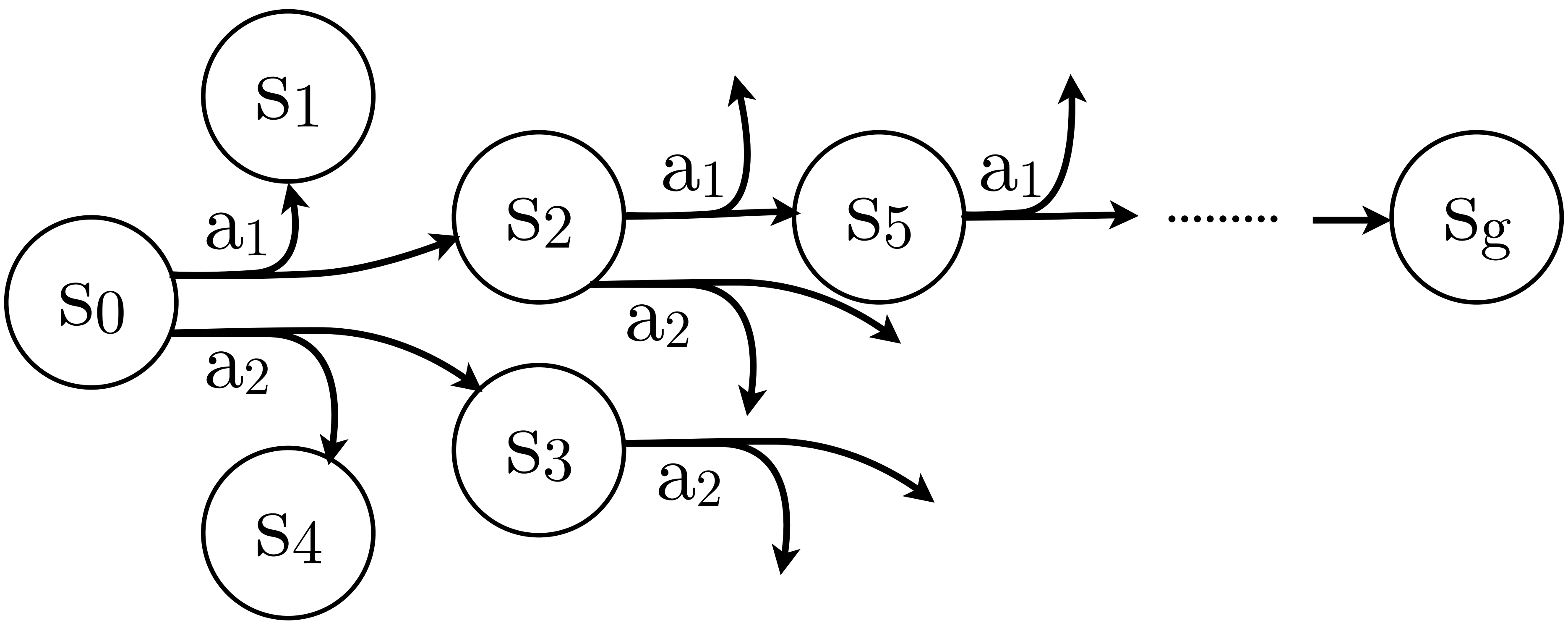
Planning under Model Uncertainty

Set of vertices in GK-Graph: $x \in \mathcal{X}$

Candidate models: $f_\theta(x), \theta = \{1, 2, \ldots, N\}$

Action: $a \in A$

State: $s \in \mathcal{S} : \langle x, \theta \rangle$



Planning under Model Uncertainty

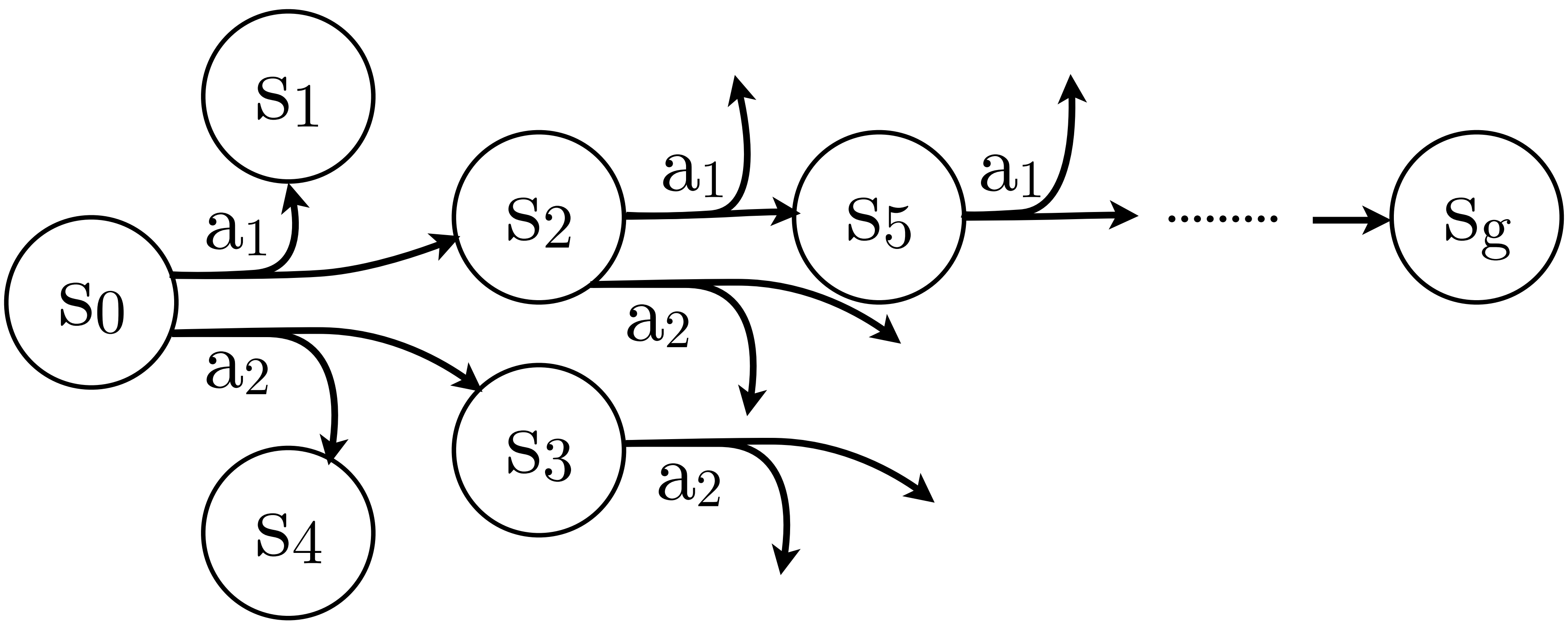
Set of vertices in GK-Graph: $x \in \mathcal{X}$

Candidate models: $f_\theta(x), \theta = \{1, 2, \ldots, N\}$

Action: $a \in A$

State: $s \in \mathcal{S} : \langle x, \theta \rangle$

Underlying 'true' model is unobserved
POMDP



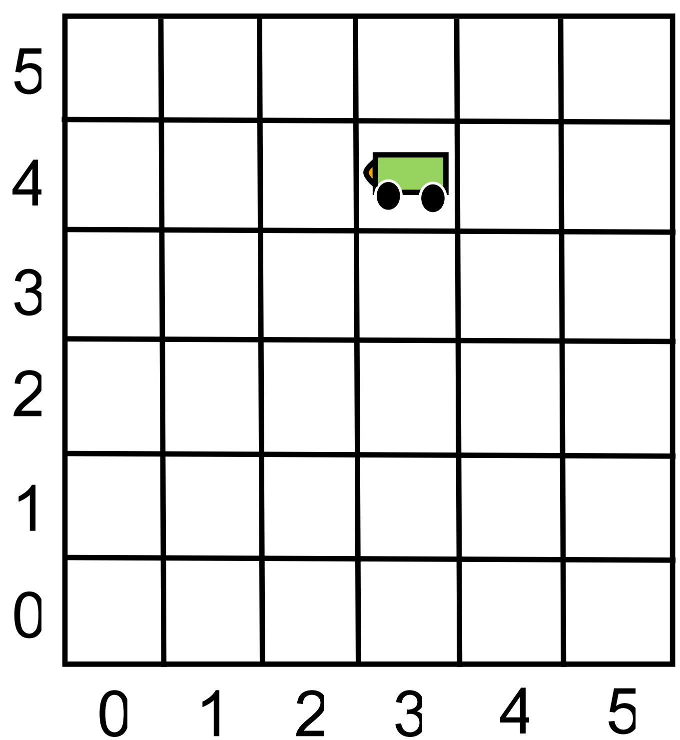
Planning under Model Uncertainty

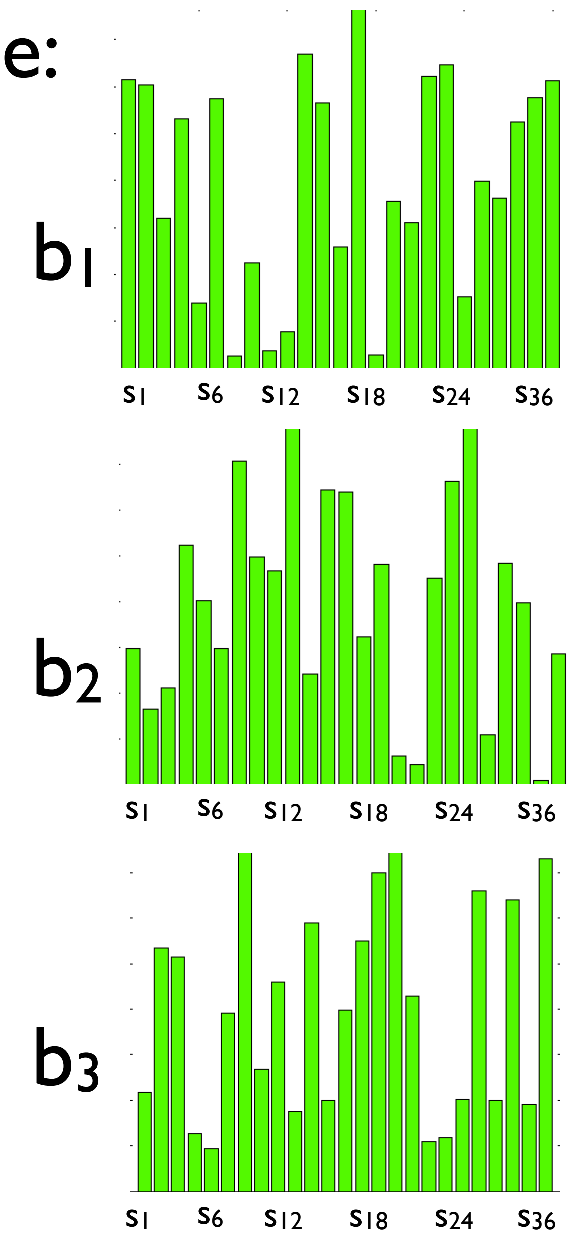
POMDPs:

- Defined by <S,A,T,C,O> (state, action, transition, cost, observation)
- For given partially observed state, what is the optimal action to take (policy)?
- Optimal action should minimize sum of future costs (optionally discounted in time)
- Hard to solve exactly (PSPACE-complete)

Planning under Model Uncertainty

POMDPs and belief space:





Belief space

Belief: $b \in \mathcal{B}$

Planning under Model Uncertainty

The belief MDP (b-MDP)

- POMDPs are equivalent to an MDP on the belief space
- POMDP: <S,A,T,C,O>
- b-MDP: <B,A,T',C'> (need to define T' and C')
- Use MDP solver of your choice (value iteration, policy iteration) and be done
- Alas, not so simple--infinitely many belief states, infinite branching factor

Planning under Model Uncertainty

Back to our problem...

Set of vertices in GK-Graph: $x \in \mathcal{X}$

Candidate models: $f_\theta(x), \theta = \{1, 2, \ldots, N\}$

Action: $a \in A$

State: $s \in \mathcal{S} : \langle x, \theta \rangle$

Key assumptions:

x is fully observed--no noise in observing GK-graph vertices

GK-graph transitions are deterministic: $x' = \textsc{Sim}(x, \theta, a)$

Planning under Model Uncertainty

Key assumptions:

x is fully observed--no noise in observing GK-graph vertices

GK-graph transitions are deterministic: $x' = \textsc{Sim}(x, \theta, a)$

Planning under Model Uncertainty

Key assumptions:

x is fully observed--no noise in observing GK-graph vertices

Don't need belief over all states $b(s)$. Sufficient to maintain $b_x(\theta)$

GK-graph transitions are deterministic: $x' = \textsc{Sim}(x, \theta, a)$

$b_x(\theta)$ is simply an N-vector. We have one of these for every x.

Planning under Model Uncertainty

Key assumptions:

x is fully observed--no noise in observing GK-graph vertices

Don't need belief over all states $b(s)$. Sufficient to maintain $b_x(\theta)$

GK-graph transitions are deterministic: $x' = \textsc{Sim}(x, \theta, a)$

Belief transitions don't have infinite branching factor!

$b_x(\theta)$ is simply an N-vector. We have one of these for every x.

Planning under Model Uncertainty

The belief MDP

For this special case where part of the state is fully observed (MOMDP[1]), we can write the belief transition update as

$$b'_{x'}(\theta') = \eta \sum_{\theta} p(x'|x,\theta,a)p(\theta'|x,\theta,a,x')b_x(\theta)$$

[1] Ong et al., POMDPS for Robotics Tasks with Mixed Observability, RSS '05

Planning under Model Uncertainty

The belief MDP

$$b'_{x'}(\theta') = \eta \sum_{\theta} p(x'|x,\theta,a)p(\theta'|x,\theta,a,x')b_x(\theta)$$

Planning under Model Uncertainty

The belief MDP

$$b'_{x'}(\theta') = \eta \sum_{\theta} p(x'|x,\theta,a)p(\theta'|x,\theta,a,x')b_x(\theta)$$

$$\begin{aligned} b'_{x'}(\theta') &= \eta \sum_{\theta} \mathbb{1}_{x'}(\textsc{Sim}(x,\theta,a))\mathbb{1}_{\theta'}(\theta)b_x(\theta) \\ &= \eta \mathbb{1}_{x'}(\textsc{Sim}(x,\theta',a))b_x(\theta') \\ &= \begin{cases} \eta b_x(\theta') & \text{if } x' = \textsc{Sim}(x,\theta',a) \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

Planning under Model Uncertainty

The belief MDP

$$b'_{x'}(\theta') = \eta \sum_{\theta} p(x'|x,\theta,a)p(\theta'|x,\theta,a,x')b_x(\theta)$$

$$\begin{aligned} b'_{x'}(\theta') &= \eta \sum_{\theta} \mathbb{1}_{x'}(\textsc{Sim}(x,\theta,a))\mathbb{1}_{\theta'}(\theta)b_x(\theta) \\ &= \eta \mathbb{1}_{x'}(\textsc{Sim}(x,\theta',a))b_x(\theta') \\ &= \begin{cases} \eta b_x(\theta') & \text{if } x' = \textsc{Sim}(x,\theta',a) \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

Key result: an action in the belief space can produce at most N successor belief states (as opposed to infinitely many in the general case)

Planning under Model Uncertainty

Belief MDP is now tractable--got rid of infinite branching, uncertainty only over models

State space is still large (can’t run value iteration)

Value iteration examines every state in the MDP
Can we get away without doing so?

Planning under Model Uncertainty

Belief MDP is now tractable--got rid of infinite branching, uncertainty only over models

State space is still large (can't run value iteration)

Value iteration examines every state in the MDP
Can we get away without doing so?

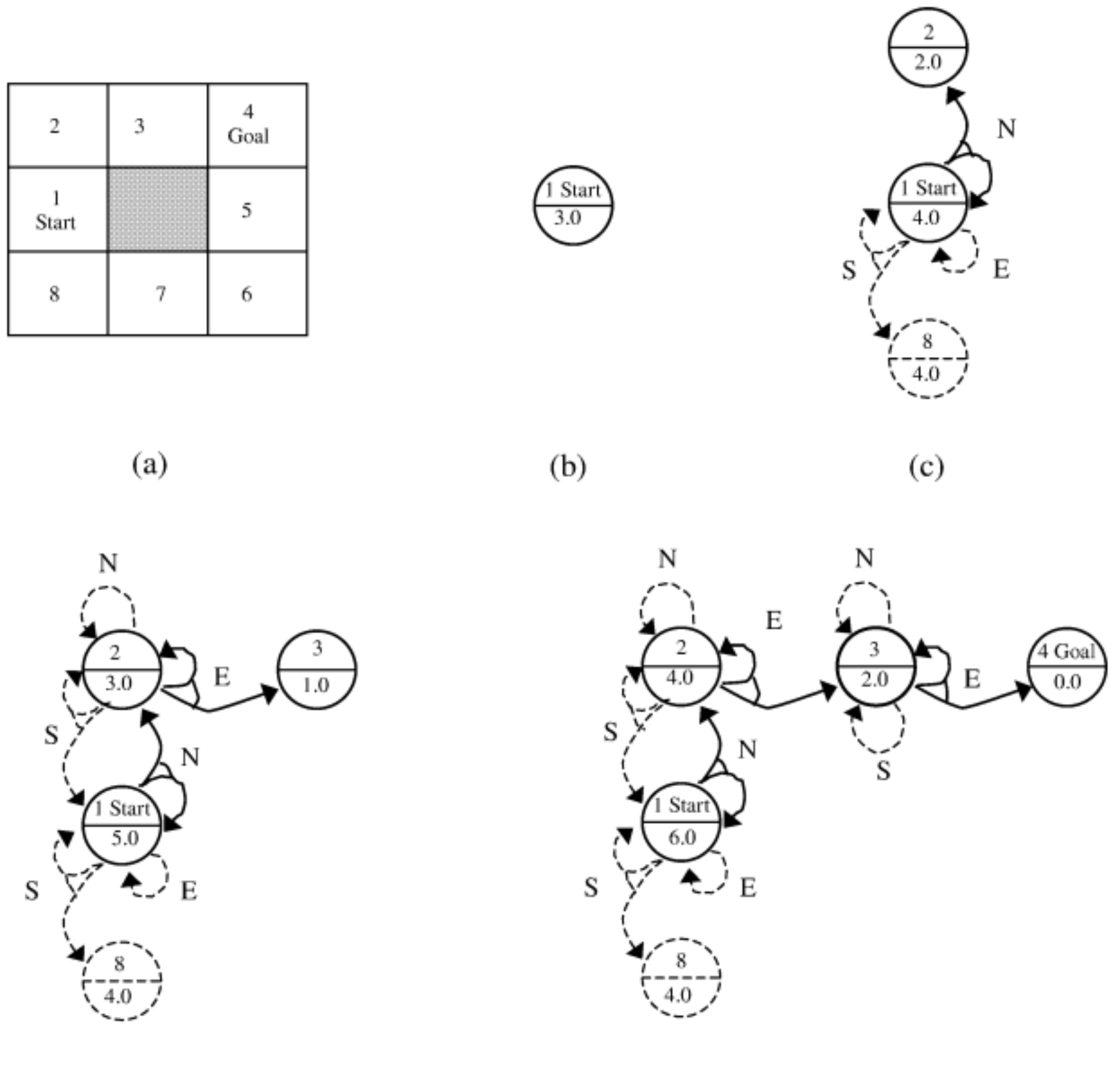
Yes! Use heuristics (a la A*) to prune the search space

Key idea: we will never reach certain parts of the state space from the start state, under the optimal policy

LAO*: Hansen and Zilberstein, Artificial Intelligence, 2001

Planning under Model Uncertainty

LAO*



(a) (b) (c)

(d) (e)

Planning under Model Uncertainty

Plan-execute-replan-repeat

```
procedure MAIN()
    while not SATISFIESGOAL(b_start) do
        π ← COMPUTEPOLICY(b_start)
        BEGINPOLICYEXECUTION(π)
        wait for new observation z
        b_start ← UPDATEBELIEF(z, b_start, π_executed)
```

Belief update

$$b'_z(\theta') = \eta \mathcal{N}(z|\textsc{Sim}(x, \theta', a), \Sigma_{motion}) b_x(\theta')$$

Outline

Representation

Problem Formulation

Related Work

Planning under Model Uncertainty

Perceptual Grounding

Experiments

Outline

Representation

Problem Formulation

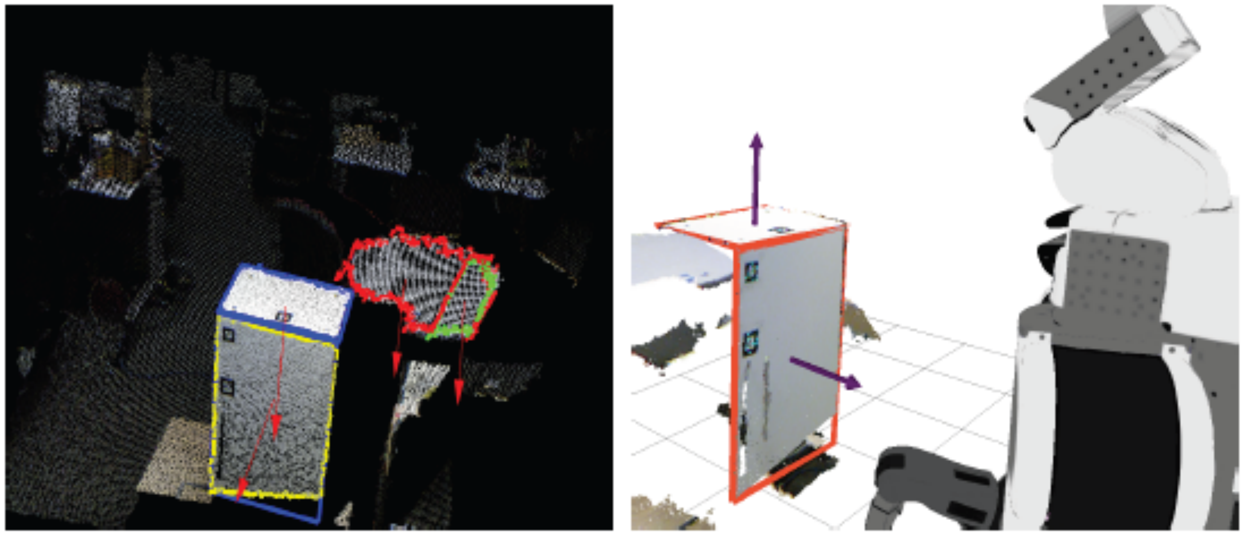
Related Work

Planning under Model Uncertainty

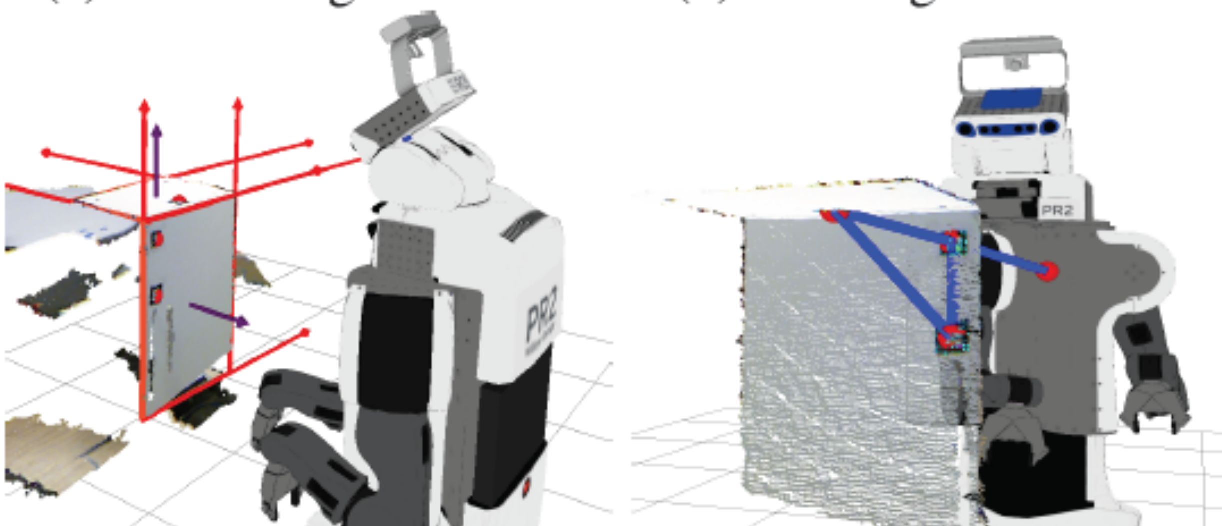
Perceptual Grounding

Experiments

Perceptual Grounding



(a) Planar segmentation (b) Rectangle detection

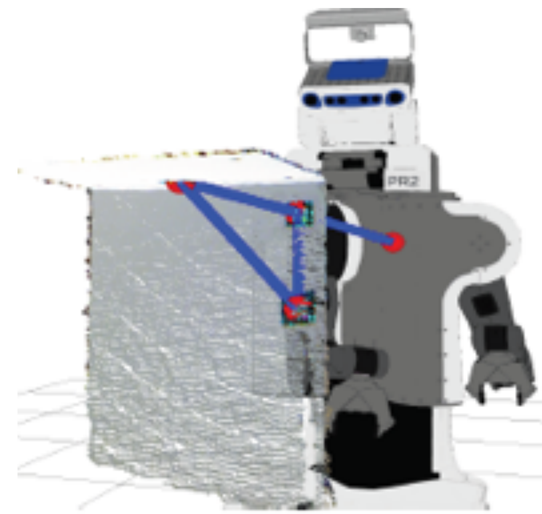


(c) Generating candidate axes (d) Assigning edge tuples to the GK-Graph

Perceptual Grounding

Use mincut for GK-graph segmentation

$$w_{ij} = \begin{cases} \exp(-(\alpha \cdot d_{ij} + \beta \cdot \cos(\theta))) & \text{if prismatic} \\ \exp(-(\alpha \cdot d_{ij} + \beta \cdot \sin(\theta))) & \text{if revolute} \end{cases}$$



For prismatic model, theta is angle between prismatic axis and x_i-x_j

For revolute model, theta is angle between the lever arms from x_i and x_j

Outline

Representation

Problem Formulation

Related Work

Planning under Model Uncertainty

Perceptual Grounding

Experiments

Outline

Representation

Problem Formulation

Related Work

Planning under Model Uncertainty

Perceptual Grounding

Experiments

Experiments

Actions: forces on unit sphere discretized into 20 directions

Heuristic: $h(b) = \min\left(0, d_{goal} - \|b.x[v_{grasp}] - b_{start}.x[v_{grasp}]\|\right).$

Inverse kinematics controller for executing action

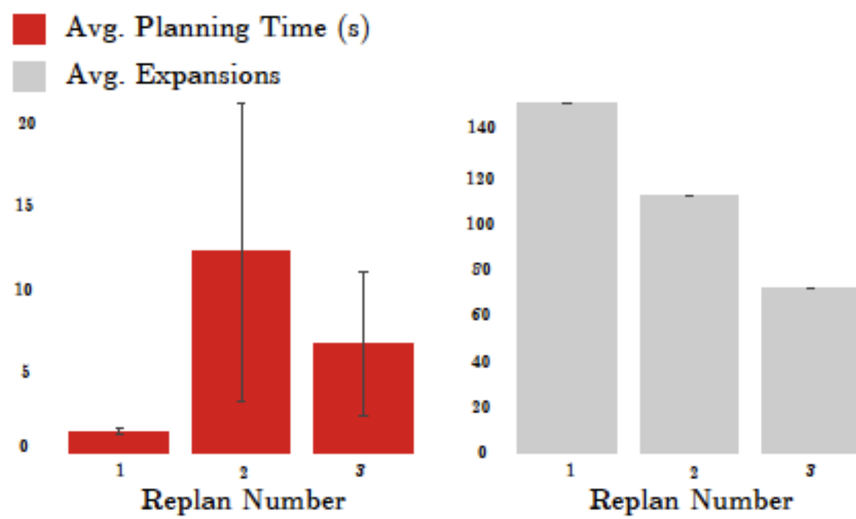
Experiments

Video

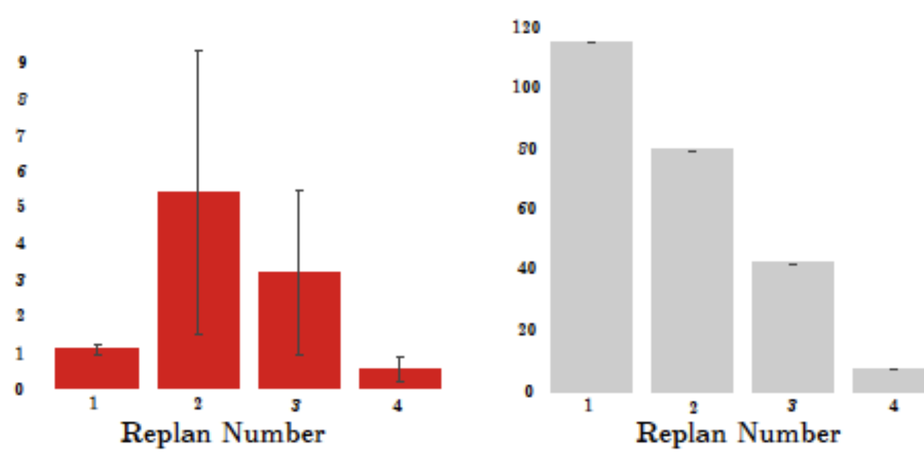
https://www.youtube.com/watch?v=E7xFtzC8ycc

Experiments

Planner Efficiency Tests



(a) Statistics for opening drawers



Summary

Novel representation (GK-Graph) for articulated objects

Planning for task-oriented manipulation

Efficient LAO* based planner for solving the belief MDP

Key insights: belief MDP tractable when transitions are deterministic, and when part of state is fully observed

Perception system for auto-generating candidate models

Extensive experiments on the PR2 robot

Questions?