Pretty good behavior in pretty big domains

or

How to stop worrying and love being wrong

Leslie Pack Kaelbling
Tomas Lozano-Perez
MIT CSAIL

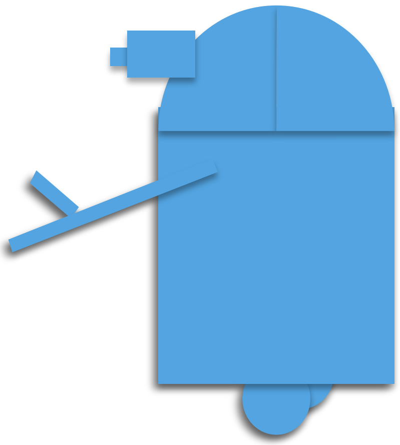
Flexible intelligence in complex domains

[1] Where's The Fork?

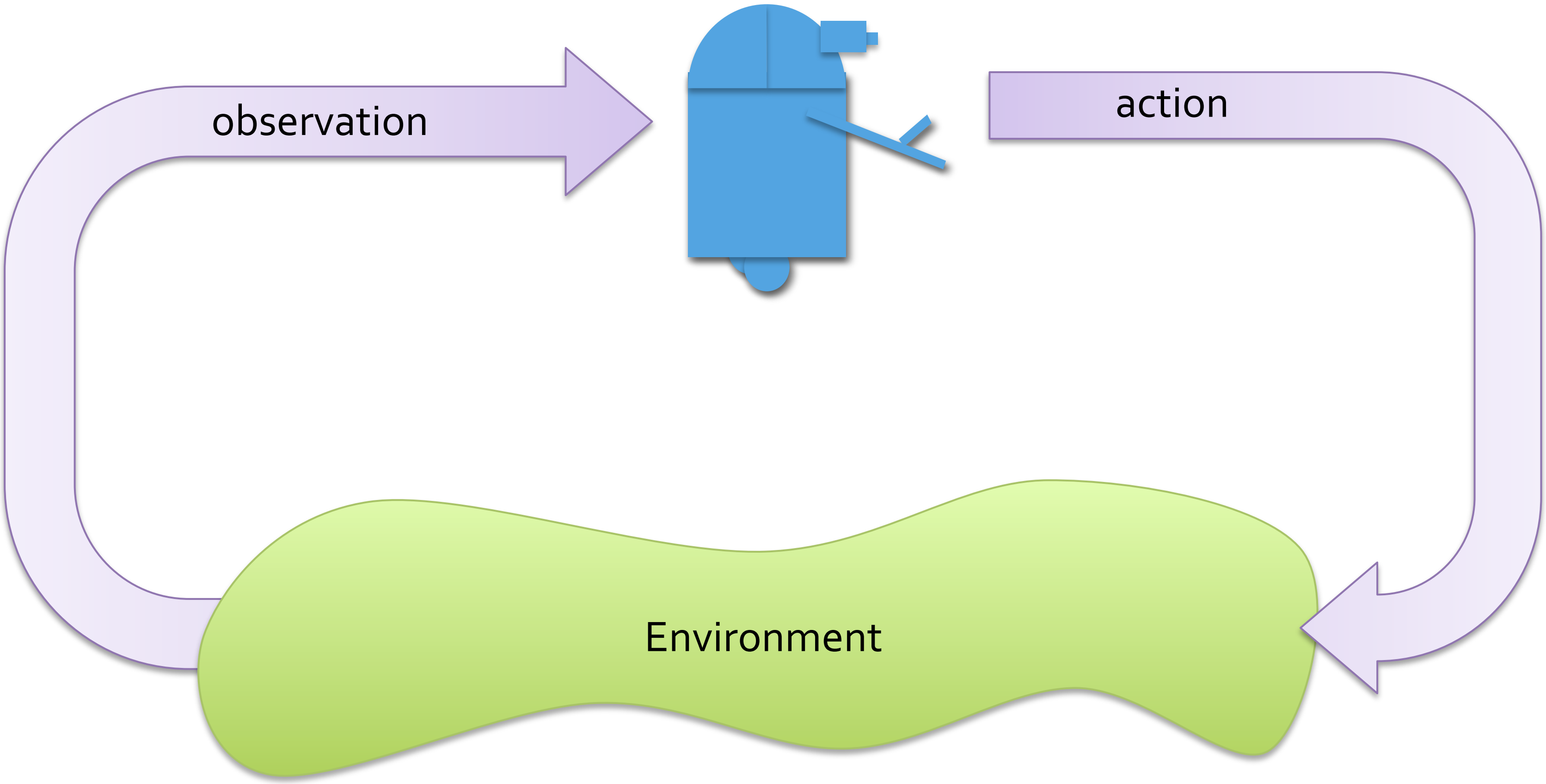
Our problem

How to build the **'central' computational mechanisms** for

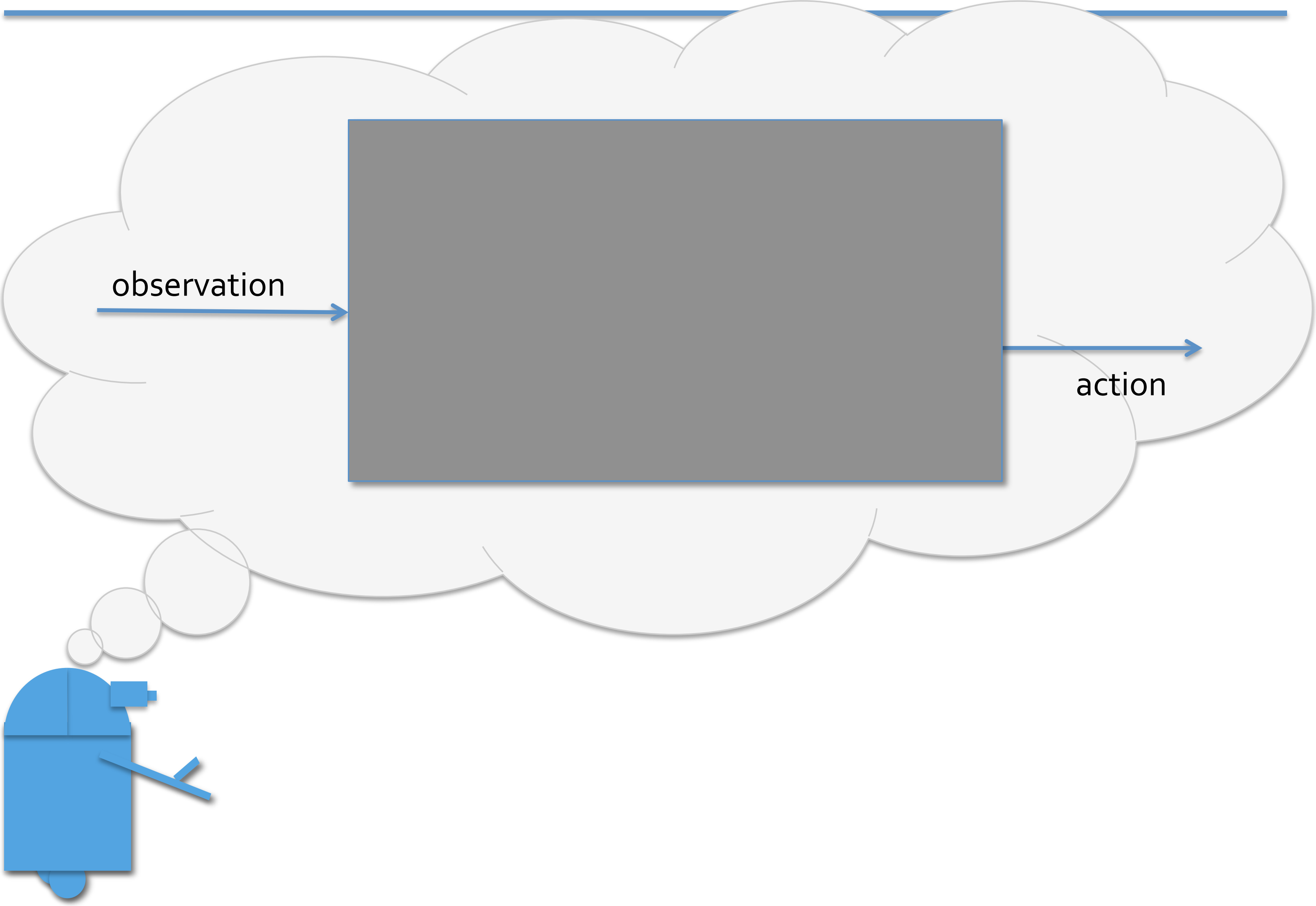
- **closed-loop control** of a system with
- **sensors and actuators** that has
- **long-term goal-directed** interactions with
- a **complex**
- **imperfectly predictable** external environment



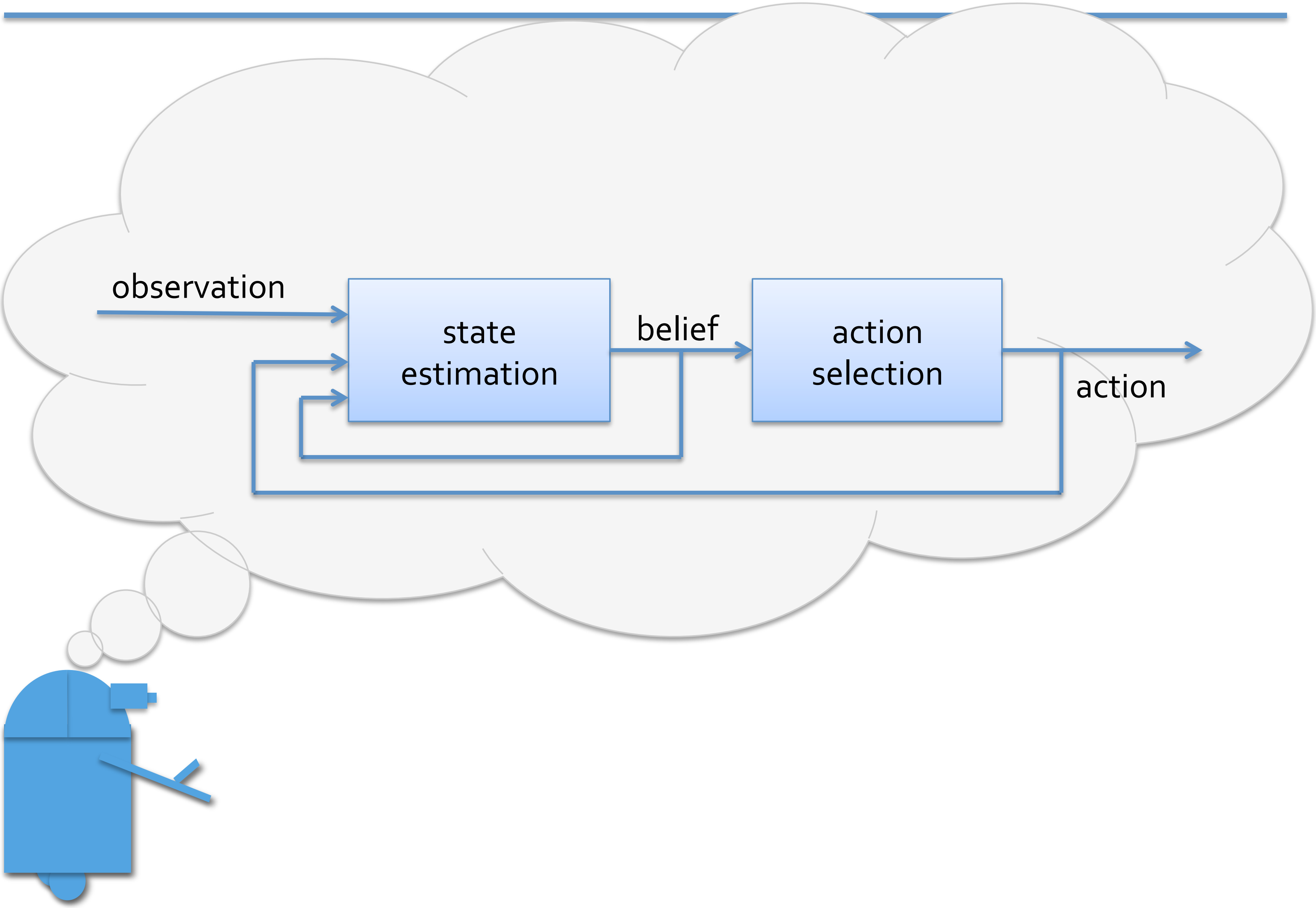
Interaction with an external environment



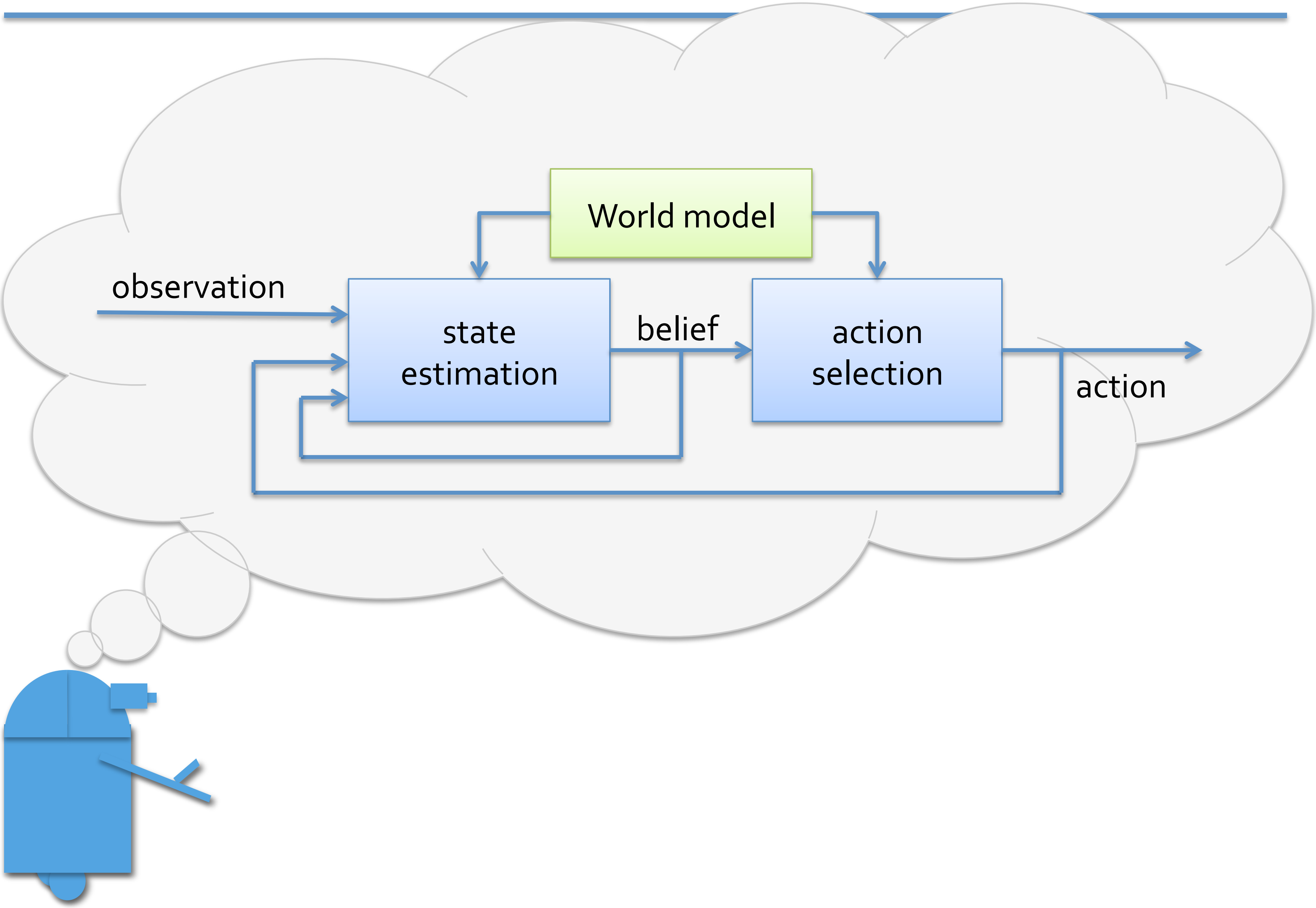
Mapping observation histories to actions



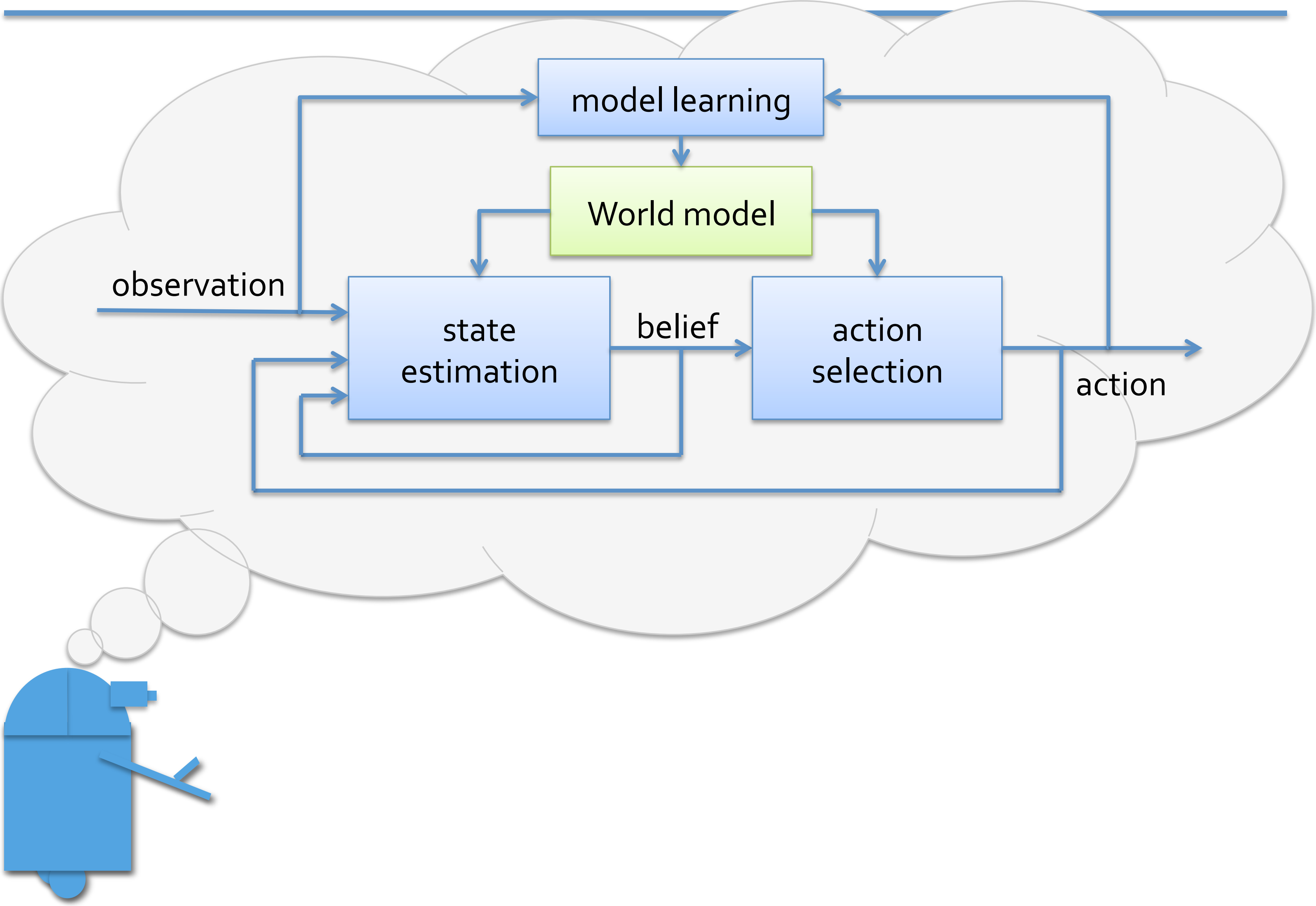
Internal architecture



Internal architecture



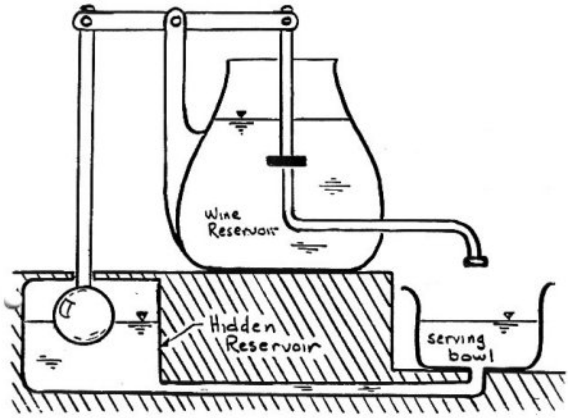
Internal architecture



Feedback control

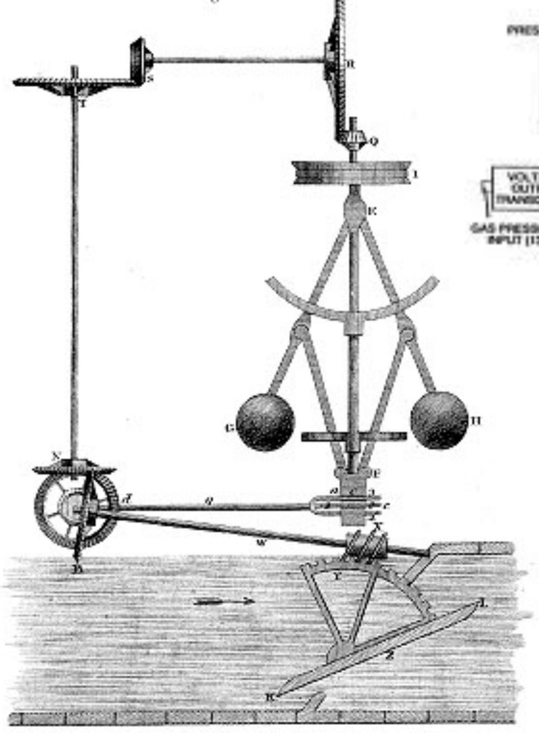
Loop:

- select an action based on (estimated) world state
- see what effect it has in the world



HERO'S SELF-LEVELING BOWL
ca. 30 B.C.

Fig. 5.



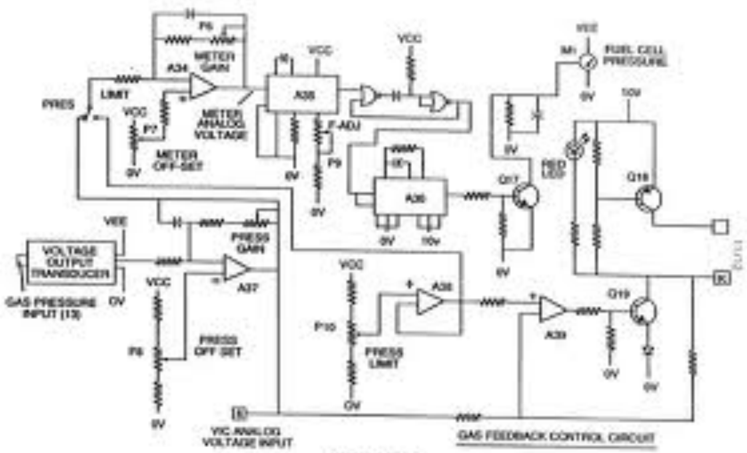
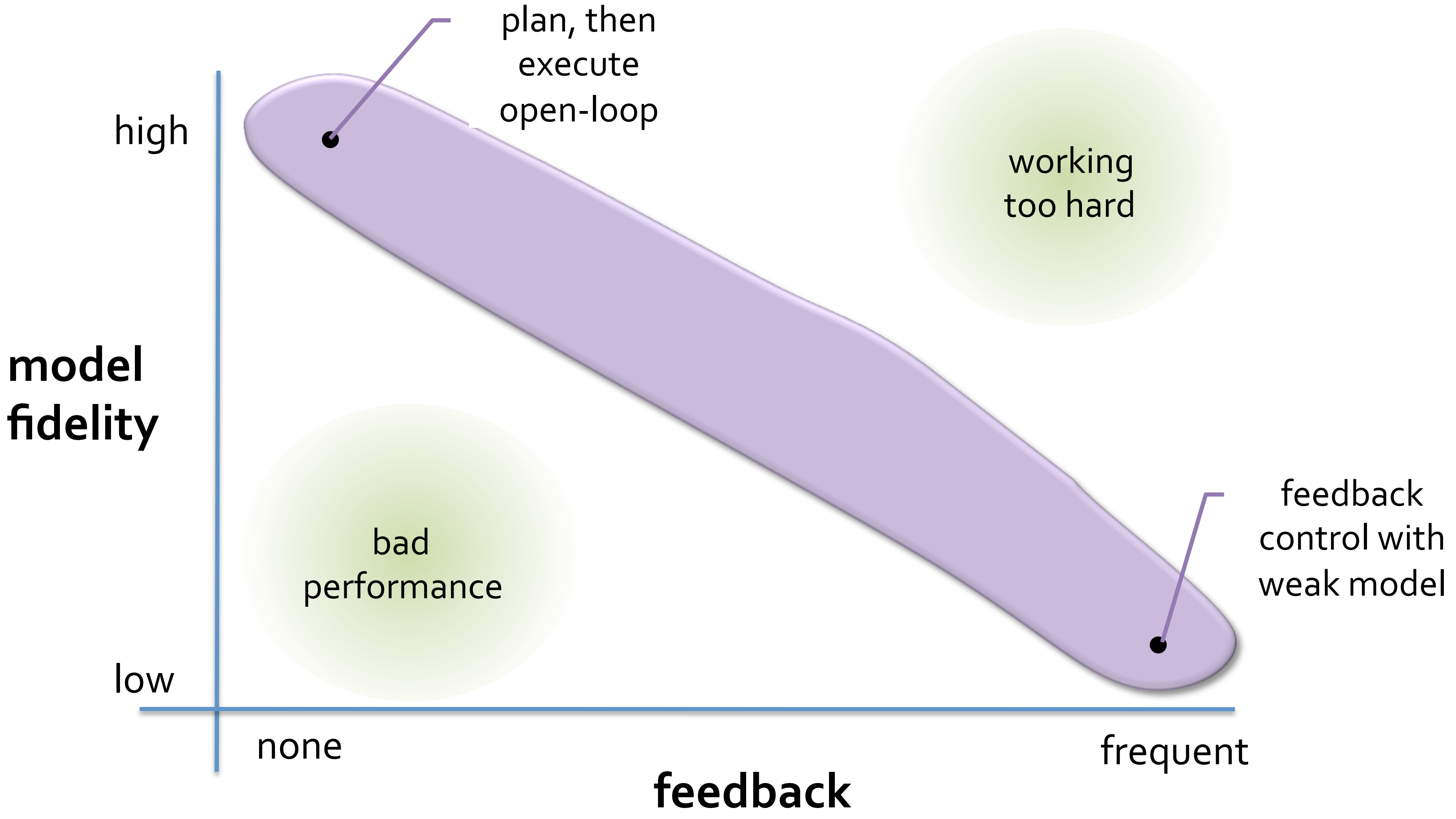


FIGURE 11

All models are wrong; but some are useful. - Box



Using simplified models for action selection

ICAPS 2004: First probabilistic planning competition

Entries: Many sophisticated MDP planning algorithms

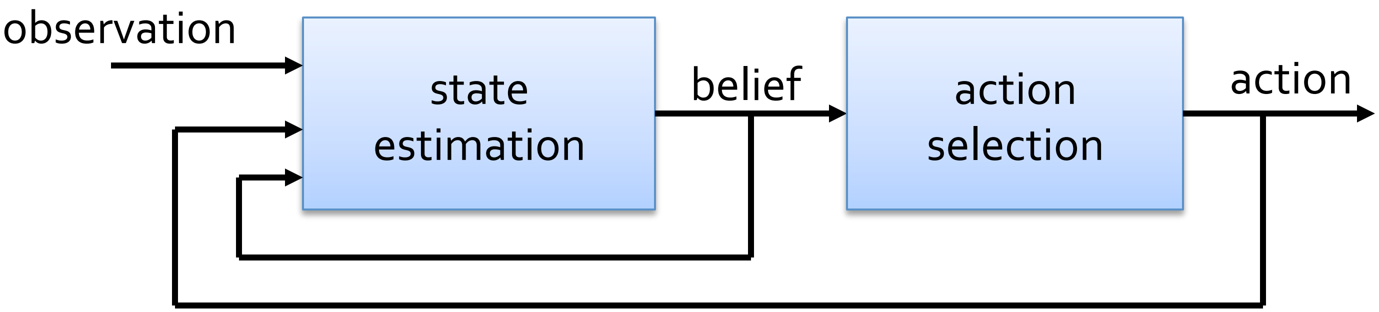
Winner: FF-Replan

- determinimized model + classical forward planner
- replan on unexpected outcomes

Result: New definition of probabilistically interesting problems

- can't be solved effectively by FF-Replan

Action selection with partial observability



Plan in belief space:

- every action gains information and changes the world
- changes are reflected in new belief via estimation
- goal is to believe that the environment is in a desired state

Using simplified models for action selection

Three examples:

Continuous control with state-dependent observation noise:

- deterministic dynamics
- most likely observation

Robot grasping with tactile sensing:

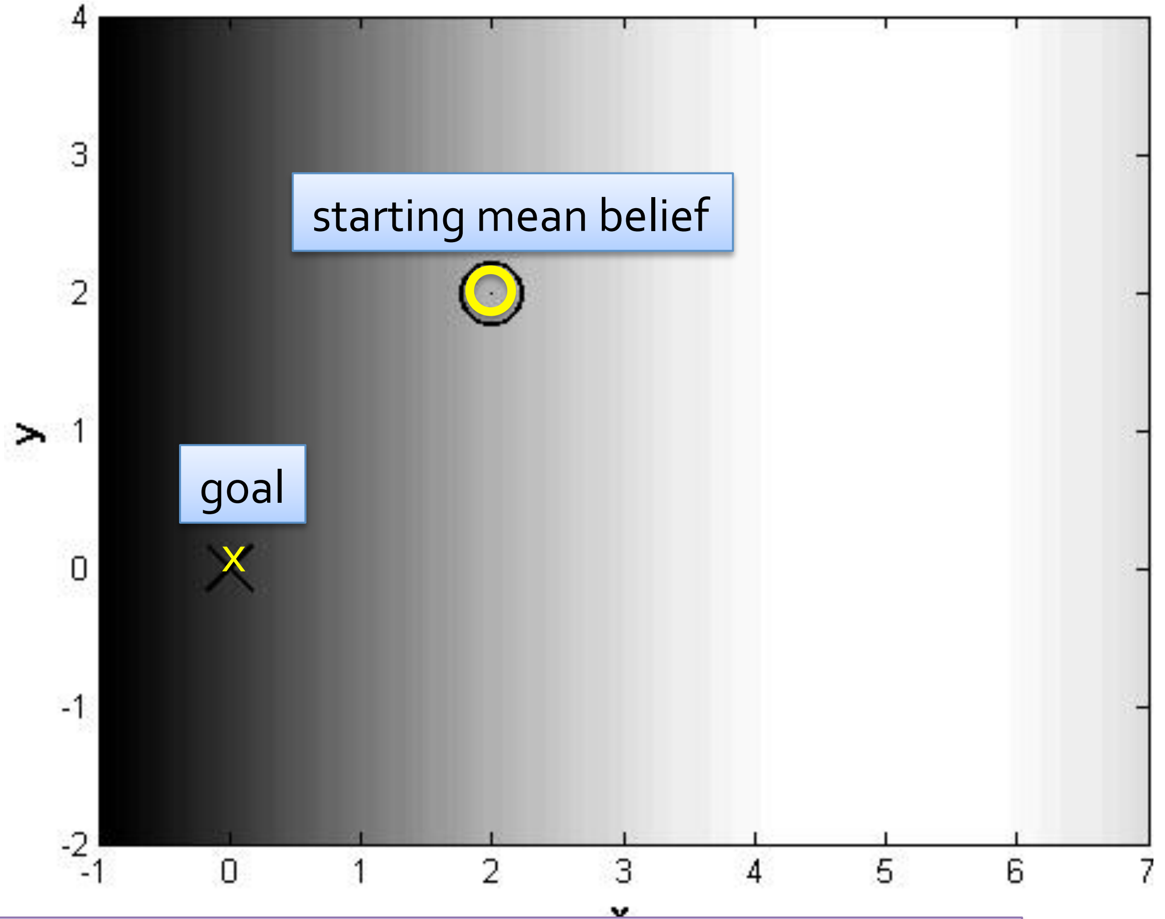
- shortened horizon
- reduced action space

Household robot with local sensing:

- assume subtask serializability
- assume desired observations

State-dependent observation noise

- robot in x, y space
- good position sensing in light regions; poor in dark

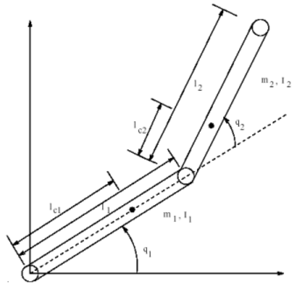


starting mean belief

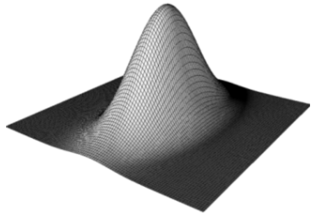
goal

Joint work with Rob Platt, Russ Tedrake and Tomás Lozano-Pérez

Control in belief space: underactuated



Acrobot



Gaussian belief:

State space: $x = \begin{pmatrix} \theta \\ \dot{\theta} \end{pmatrix}$ $b = \begin{pmatrix} m \\ \Sigma \end{pmatrix}$

Planning objective: $x_g = \begin{pmatrix} \pi \\ 0 \end{pmatrix}$ $b_g = \begin{pmatrix} x_g \\ 0 \end{pmatrix}$

Underactuated dynamics: $\ddot{\theta} = f(\theta, \dot{\theta}, u)$???

Belief space dynamics

Dynamics specify next belief state, as a function of previous belief state and action

- state update: generalized Kalman filter

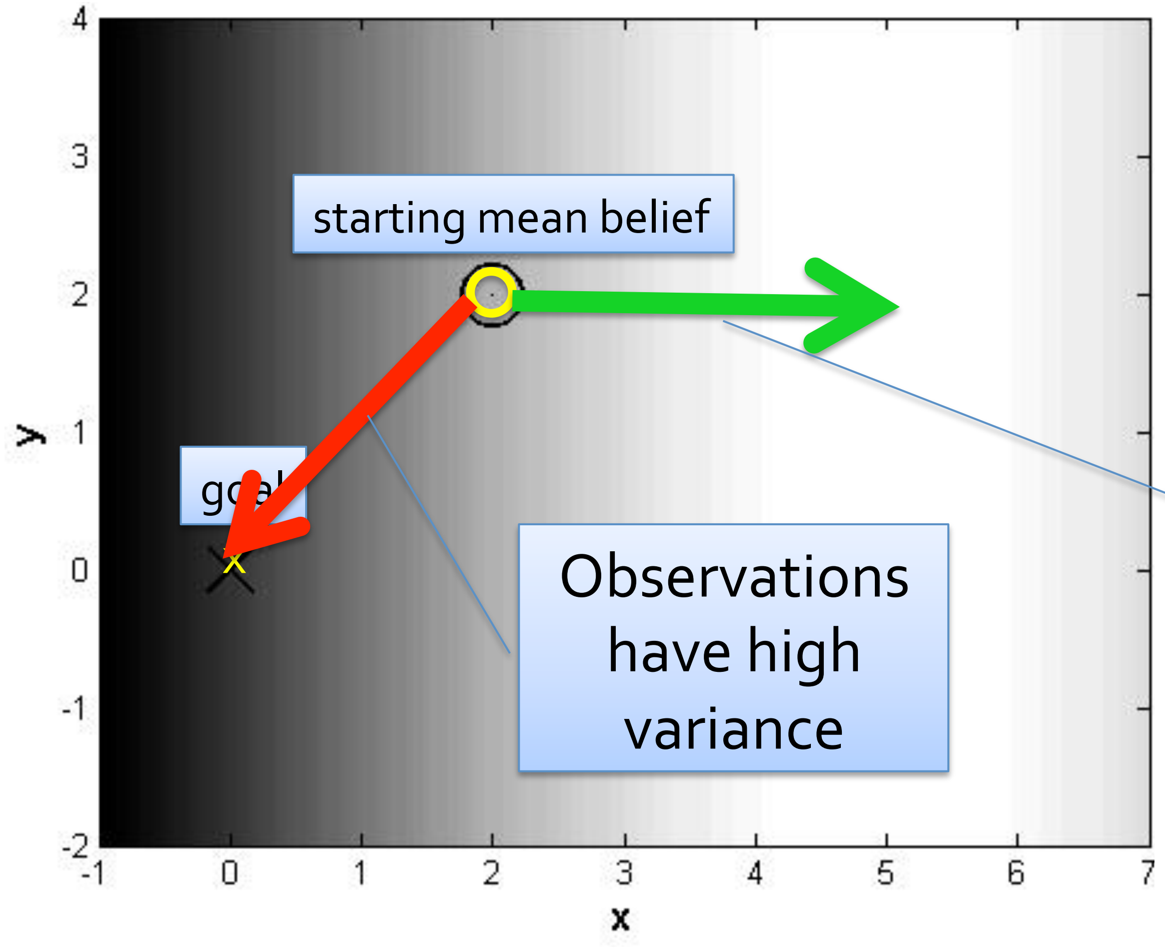
$$(\mu_{t+1}, \Sigma_{t+1}) = \mathrm{GKF}(o_t, a_t, \mu_t, \Sigma_t)$$

- **substitute expected observation in for actual one** add Gaussian noise

$$\begin{aligned}(\mu_{t+1}, \Sigma_{t+1}) &= F(a_t, \mu_t, \Sigma_t) + N \\ &= \mathrm{GKF}(\bar{o}(\mu_t), a_t, \mu_t, \Sigma_t) + N\end{aligned}$$

- continuous Gaussian non-linear dynamics: apply tools from control theory

Information gathering



$$\sigma_z^2 = \frac{1}{2}(5 - x)^2$$

Observations have high variance

Observations have low variance

Belief space planning

Find path $a_1 \ldots a_T$ that minimizes the cost function:

$$J = \sum_{i=i}^{k} n_i^T \Sigma_T n_i + \sum_{i=1}^{T} a_t^T R a_t$$

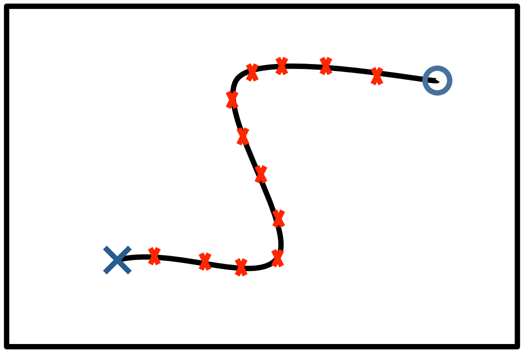
covariance at final state

action cost along path

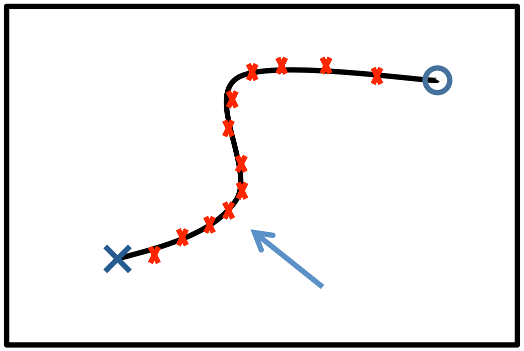
subject to $\mu_T = x_{goal}$

Planning by local optimization

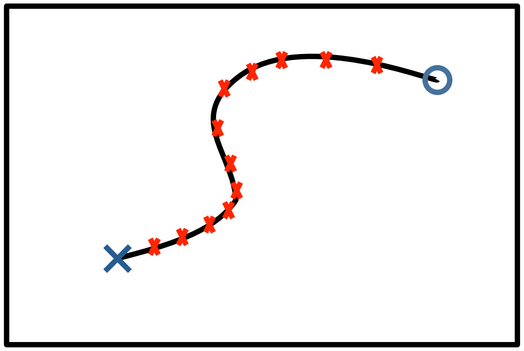
1. Parameterize initial trajectory by “via” points



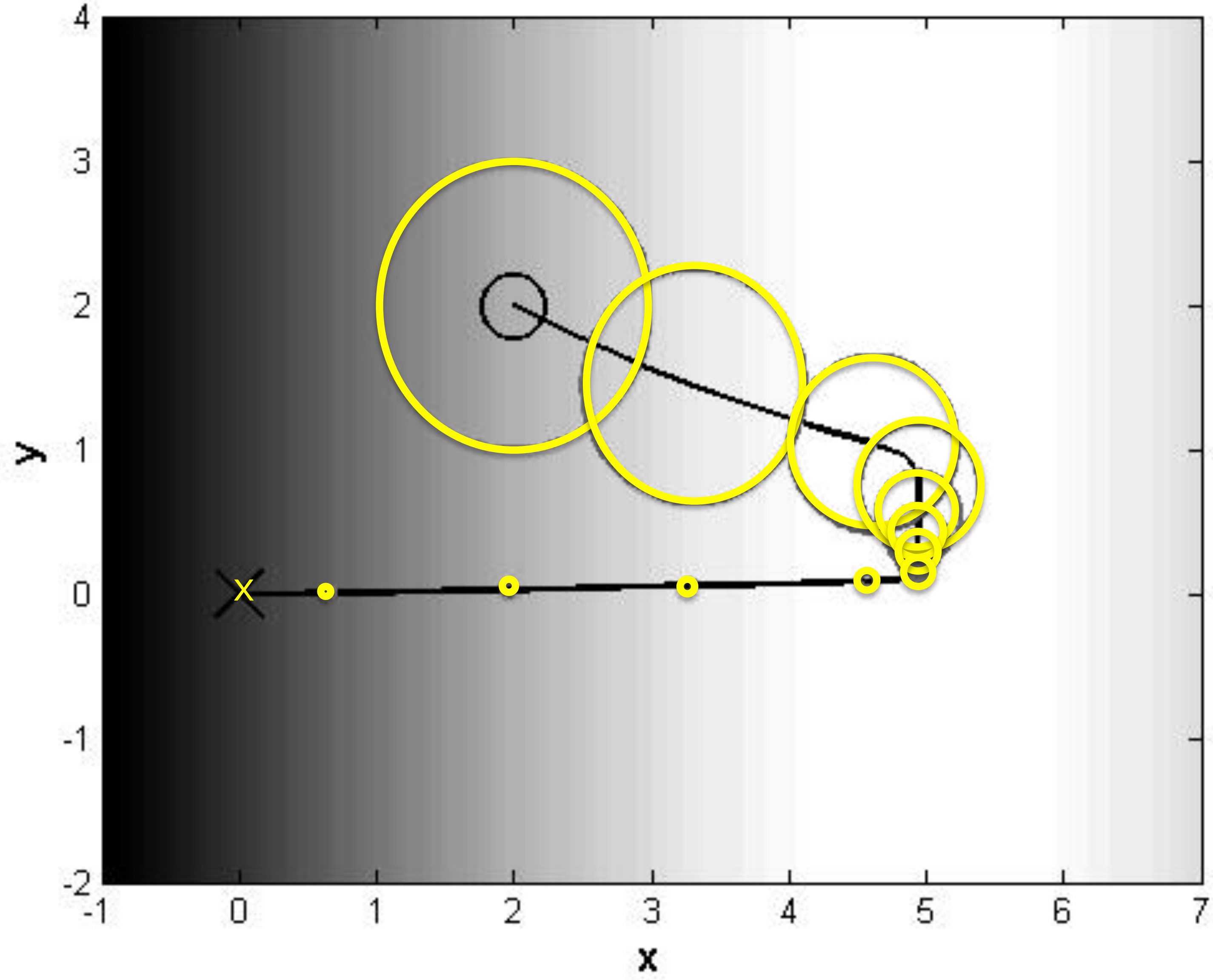
2. Shift “via” points while enforcing dynamic constraints



3. Stop when local minimum is reached

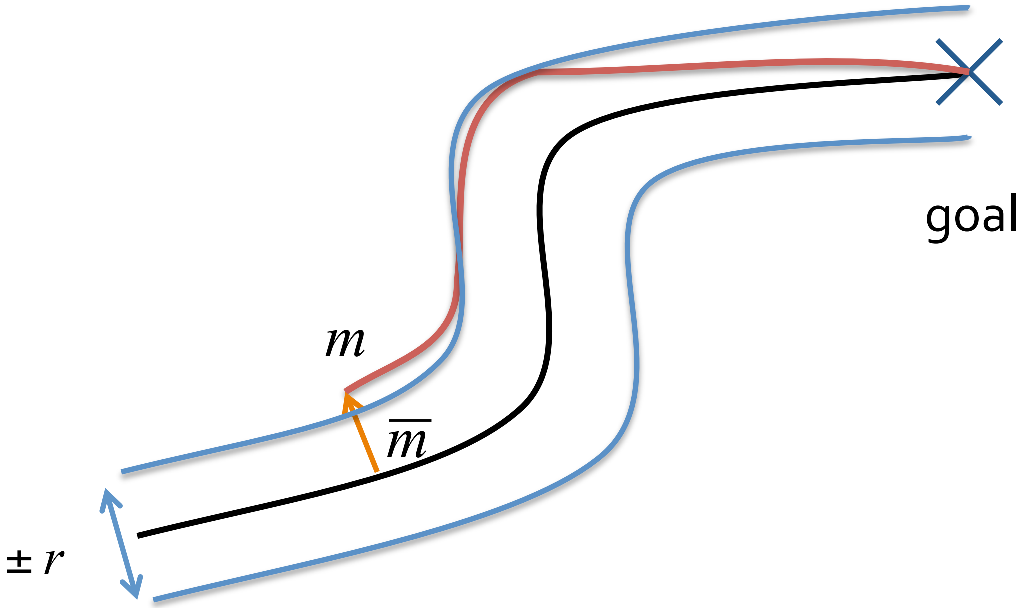


Light-dark plan

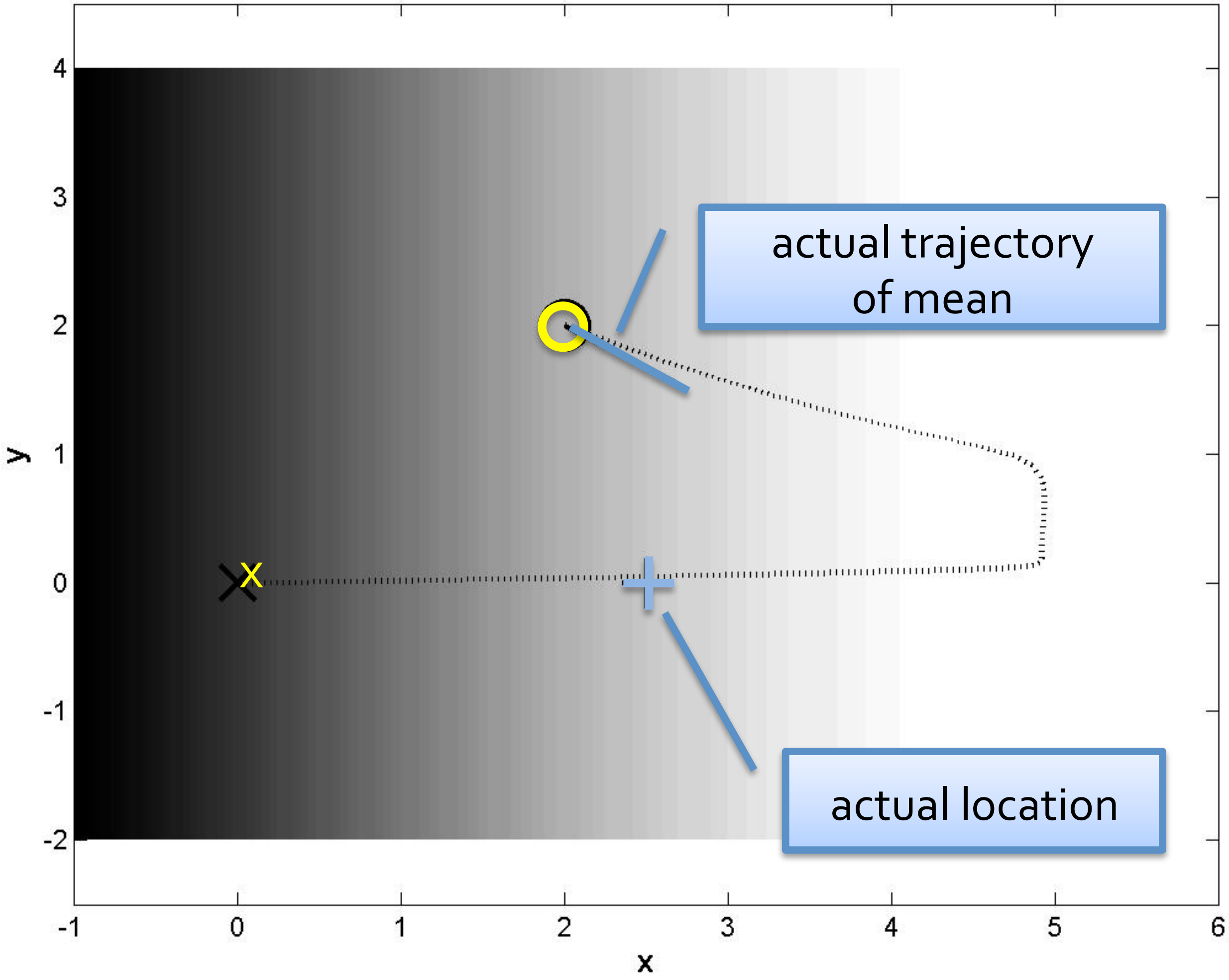


Replanning

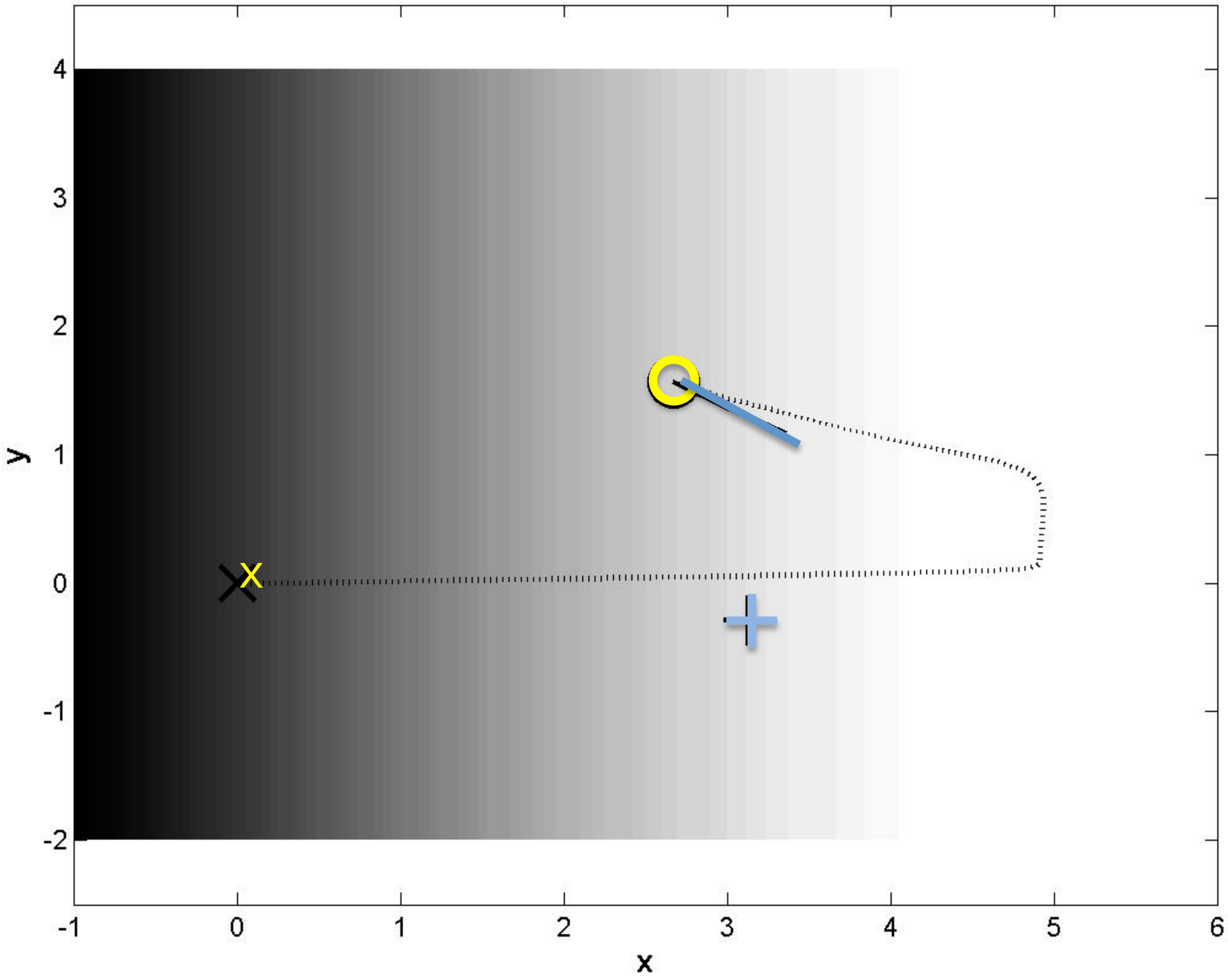
Replan when new belief state deviates too far from planned trajectory



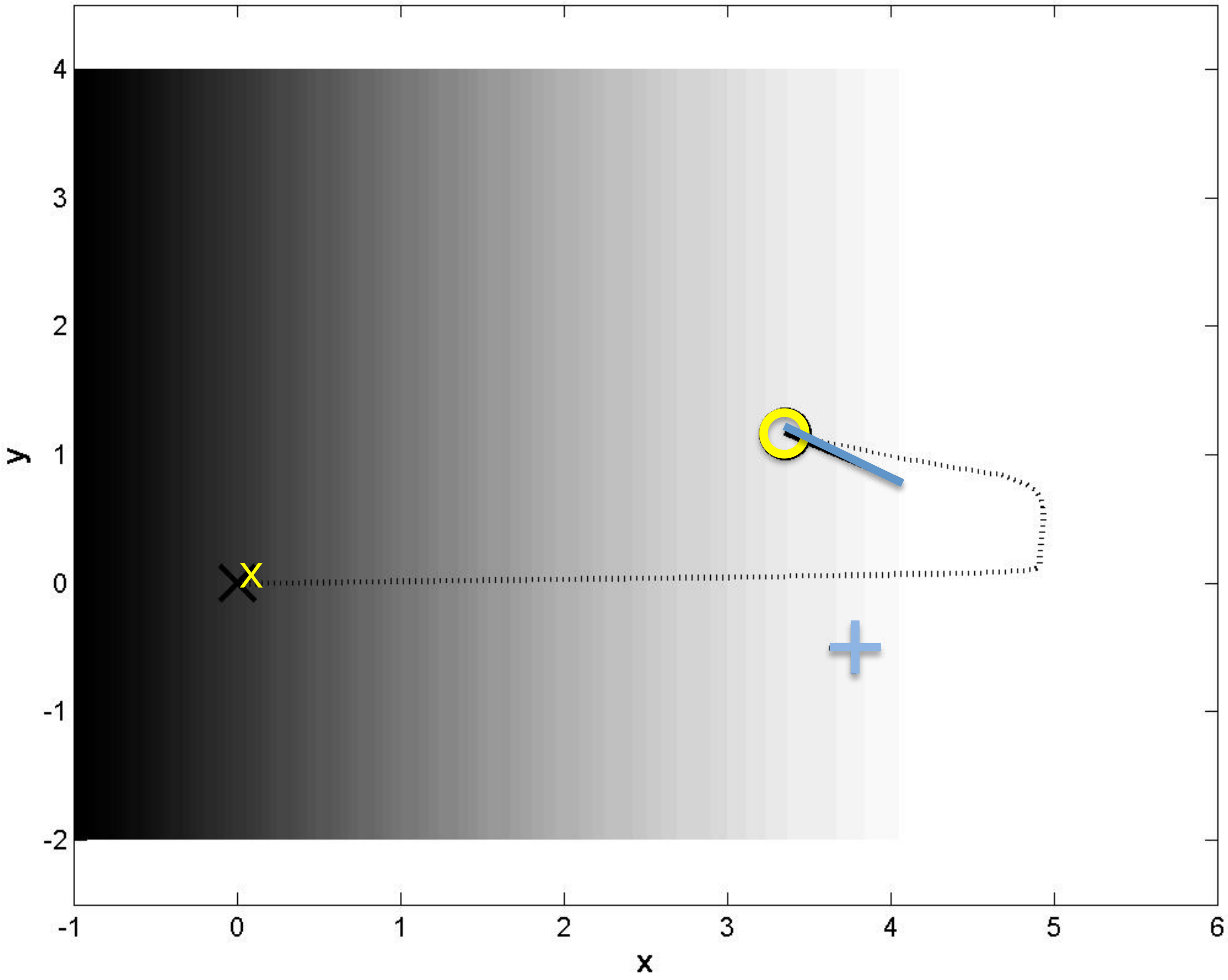
Replanning: light-dark problem



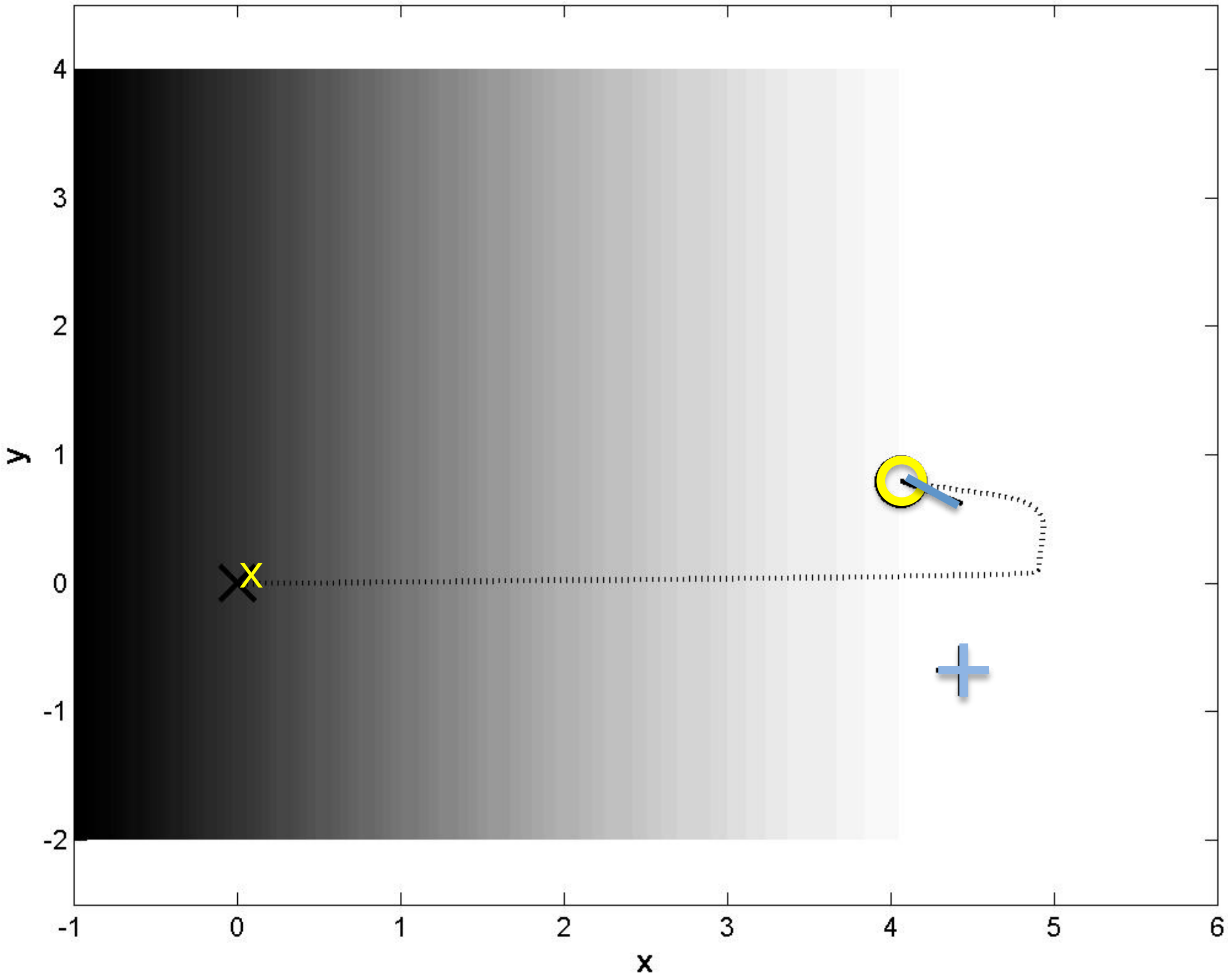
Replanning: light-dark problem



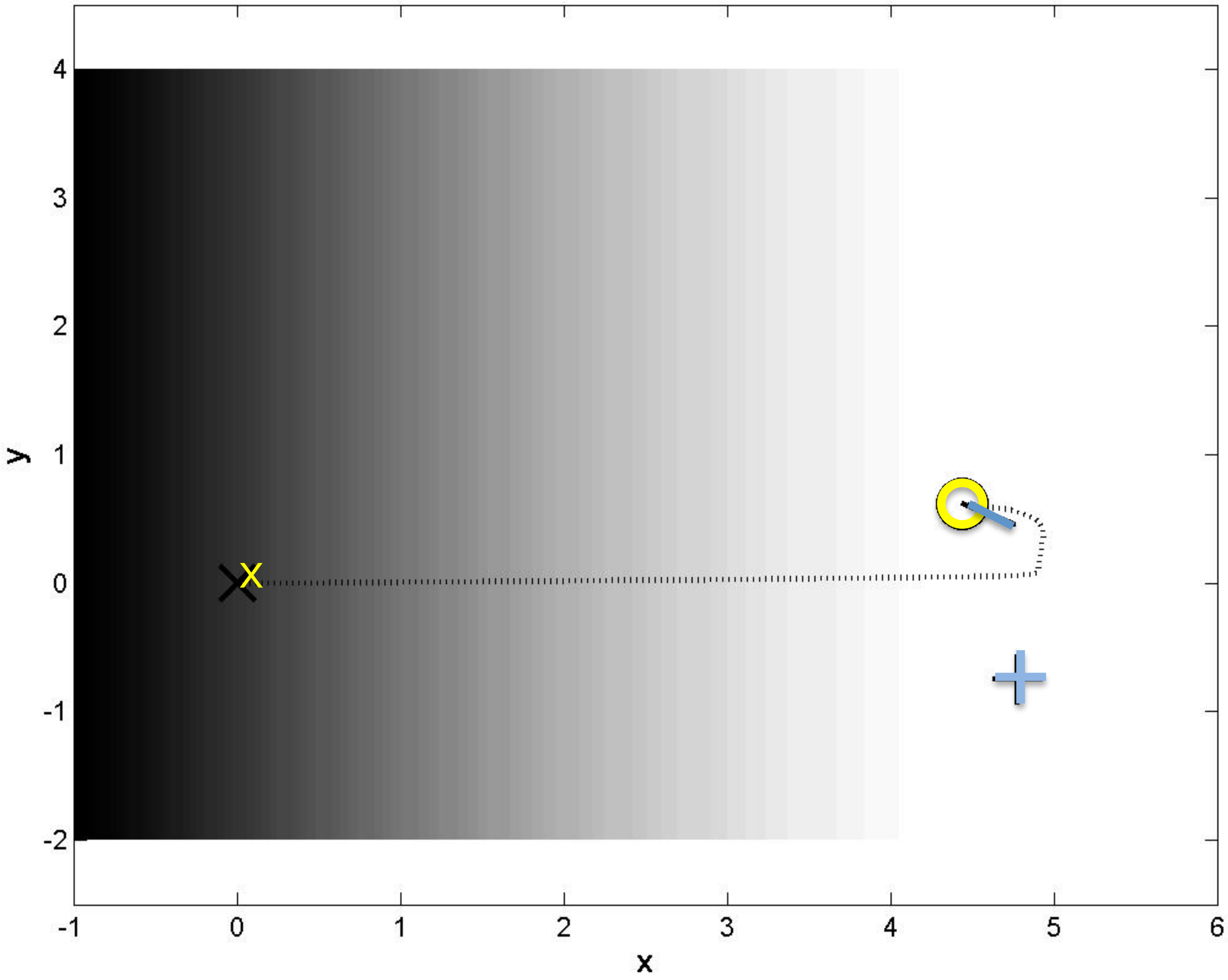
Replanning: light-dark problem



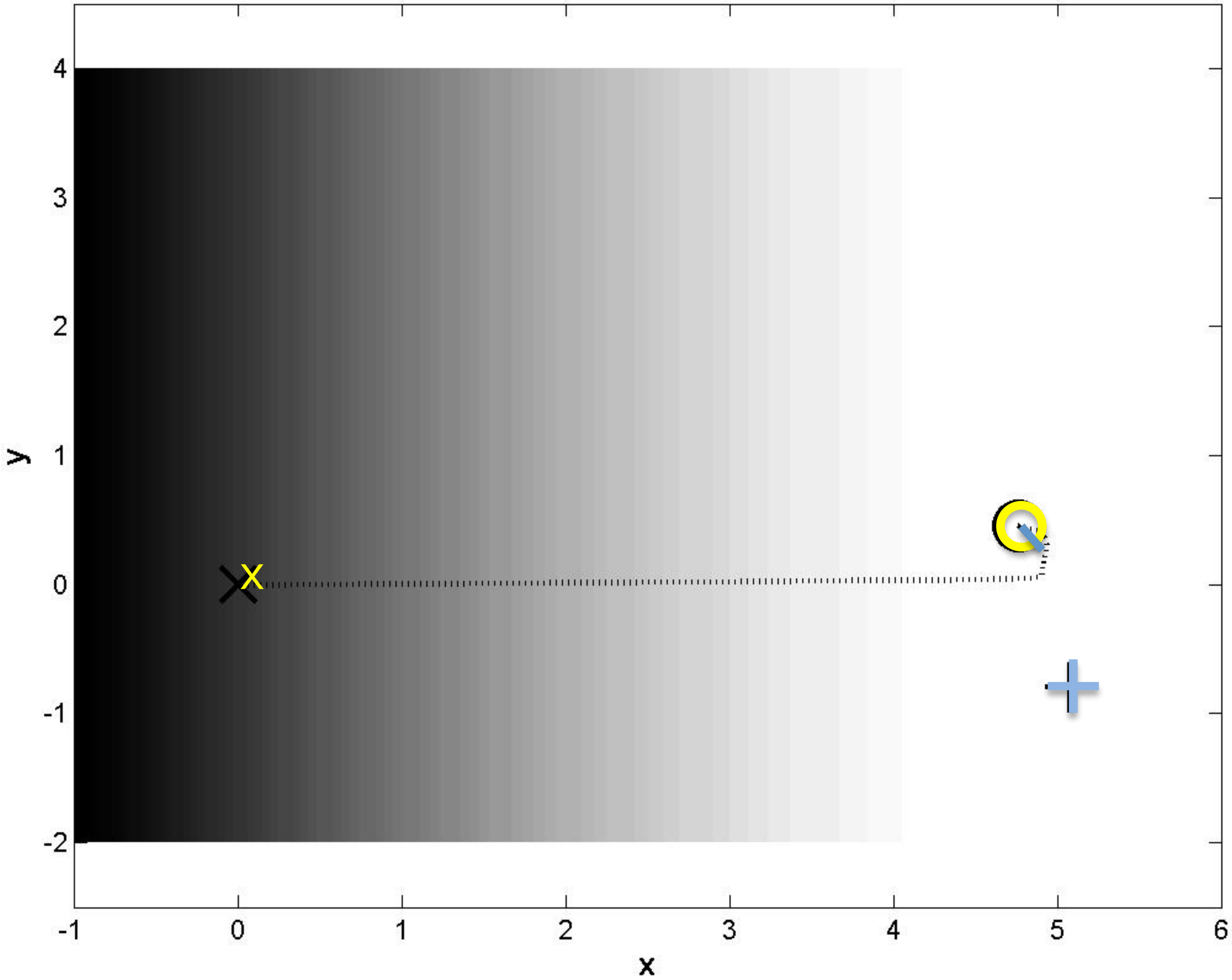
Replanning: light-dark problem



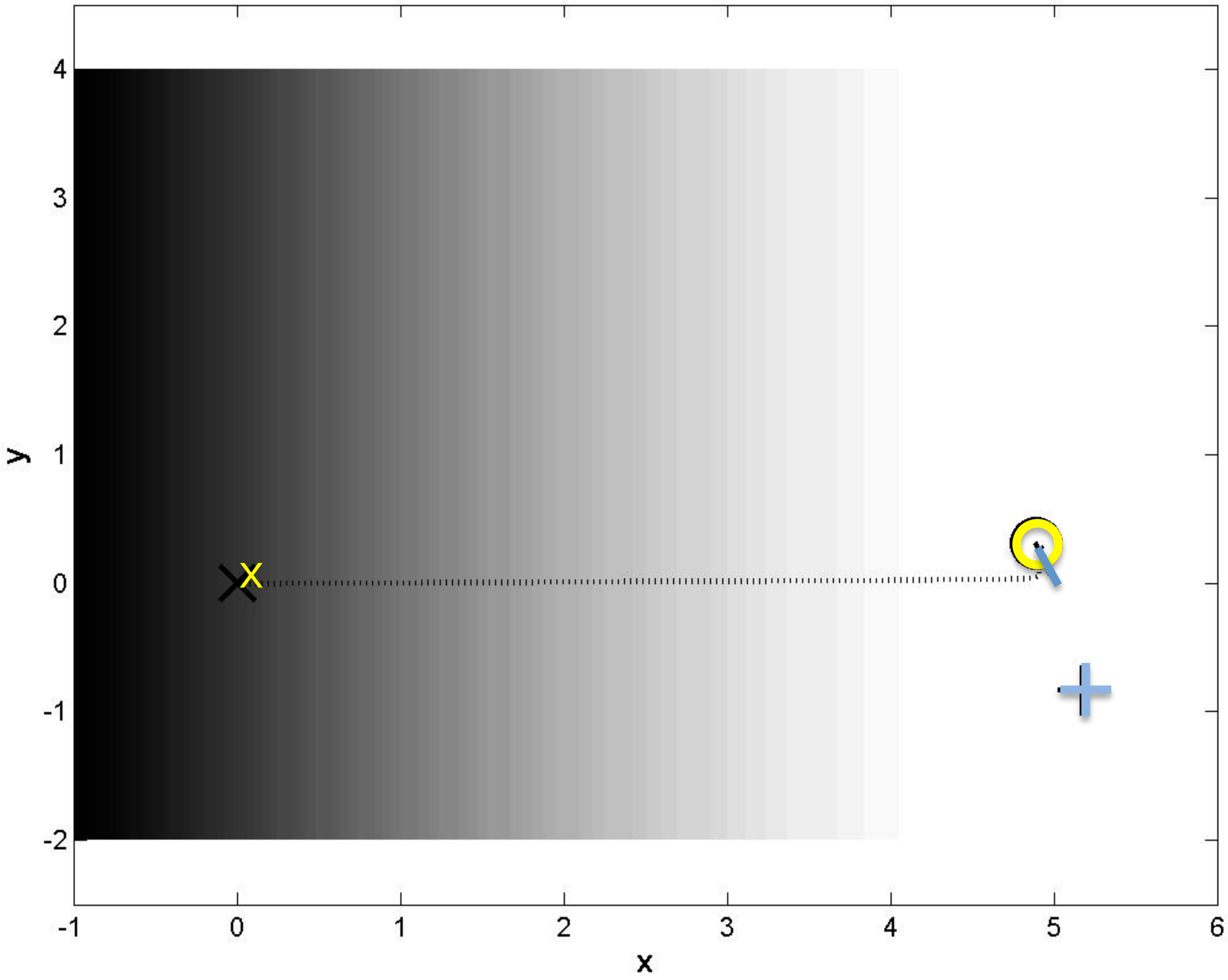
Replanning: light-dark problem



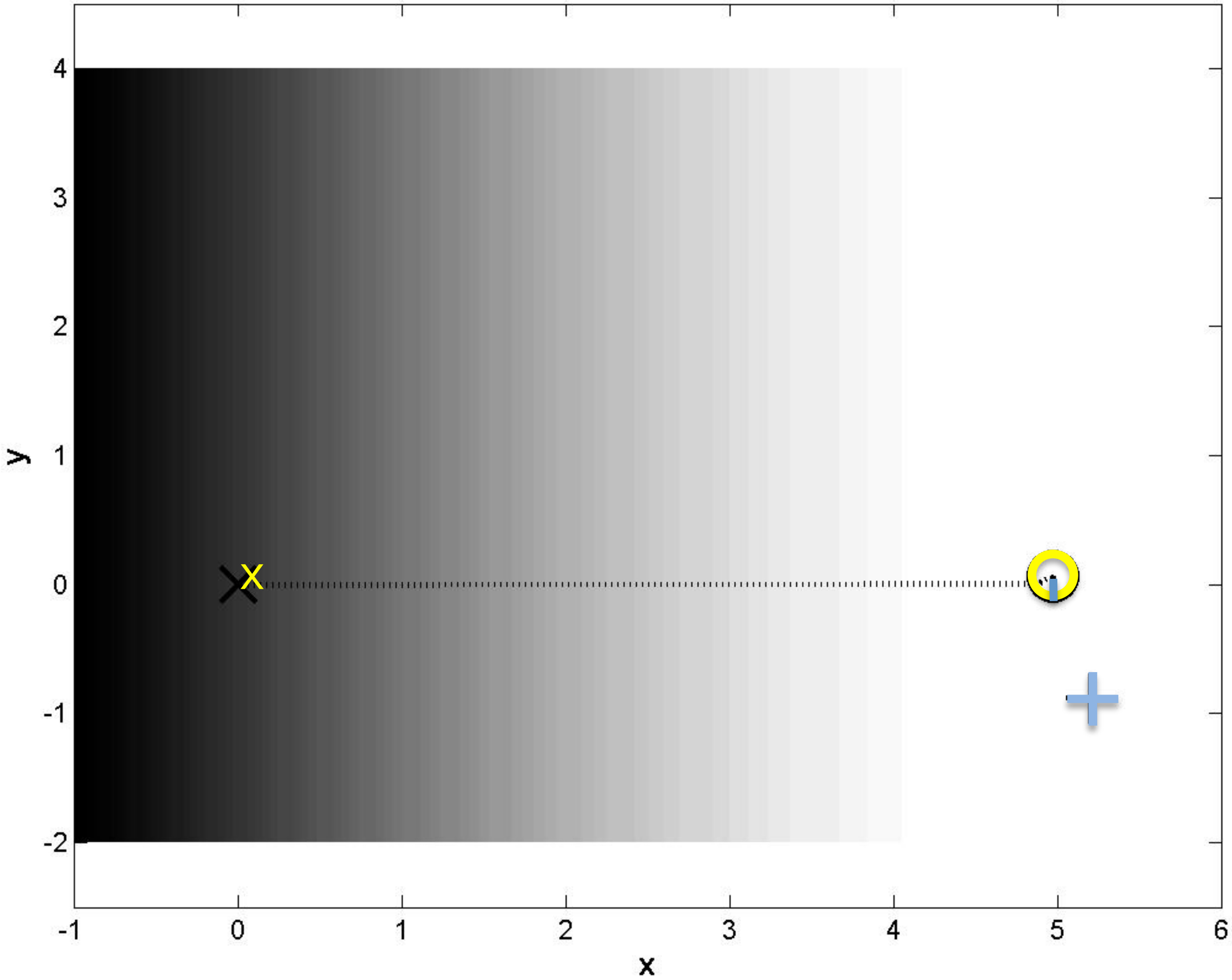
Replanning: light-dark problem



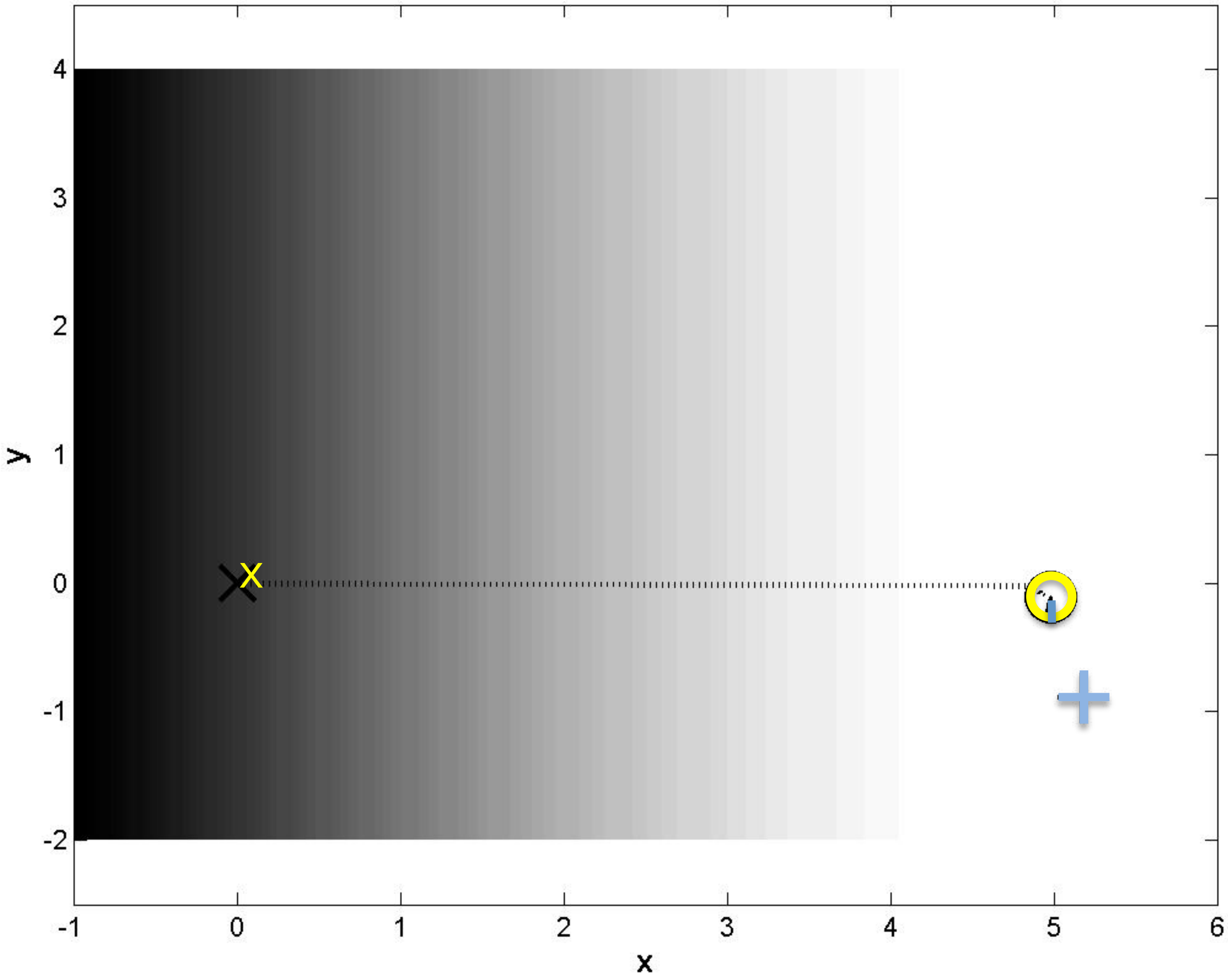
Replanning: light-dark problem



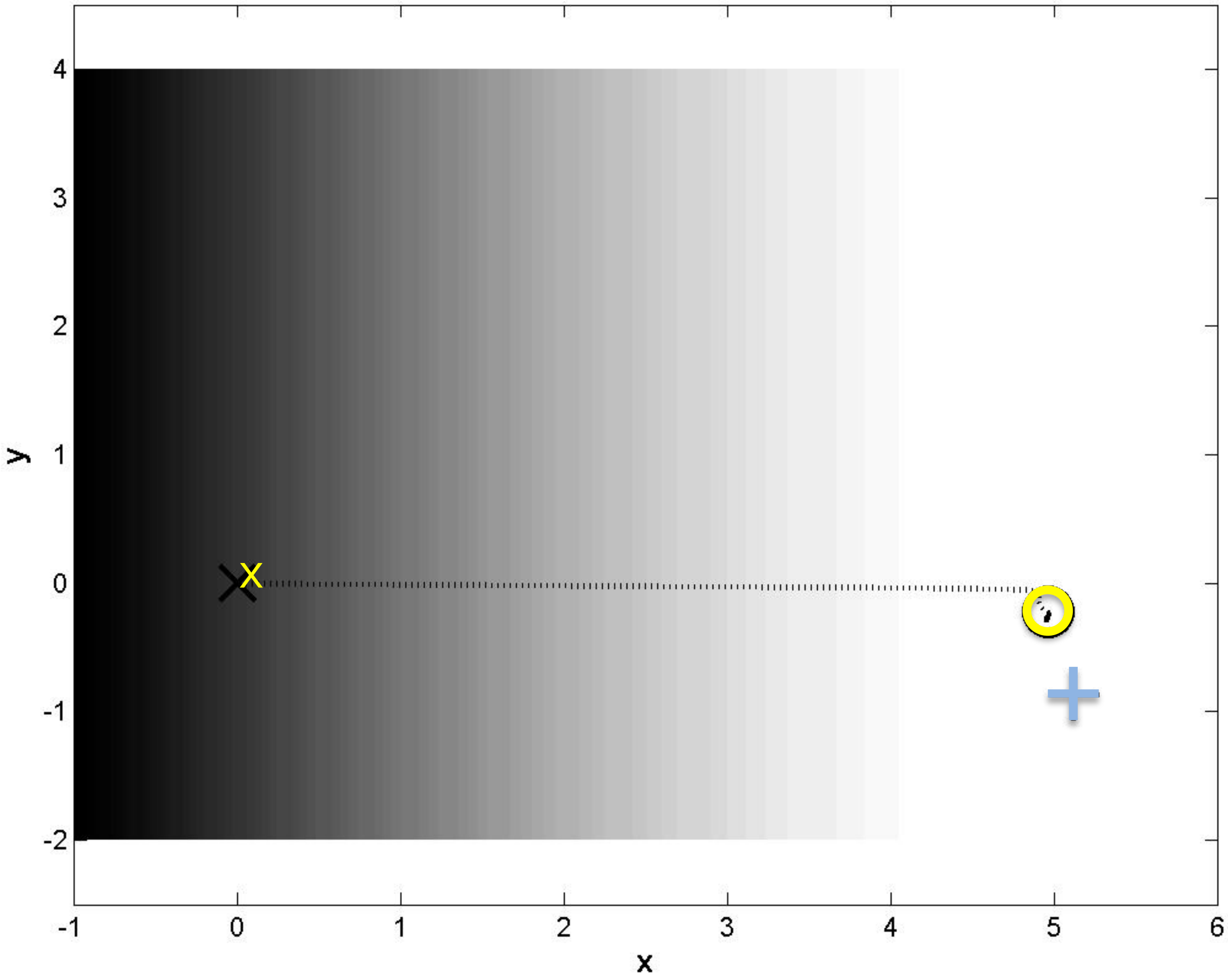
Replanning: light-dark problem



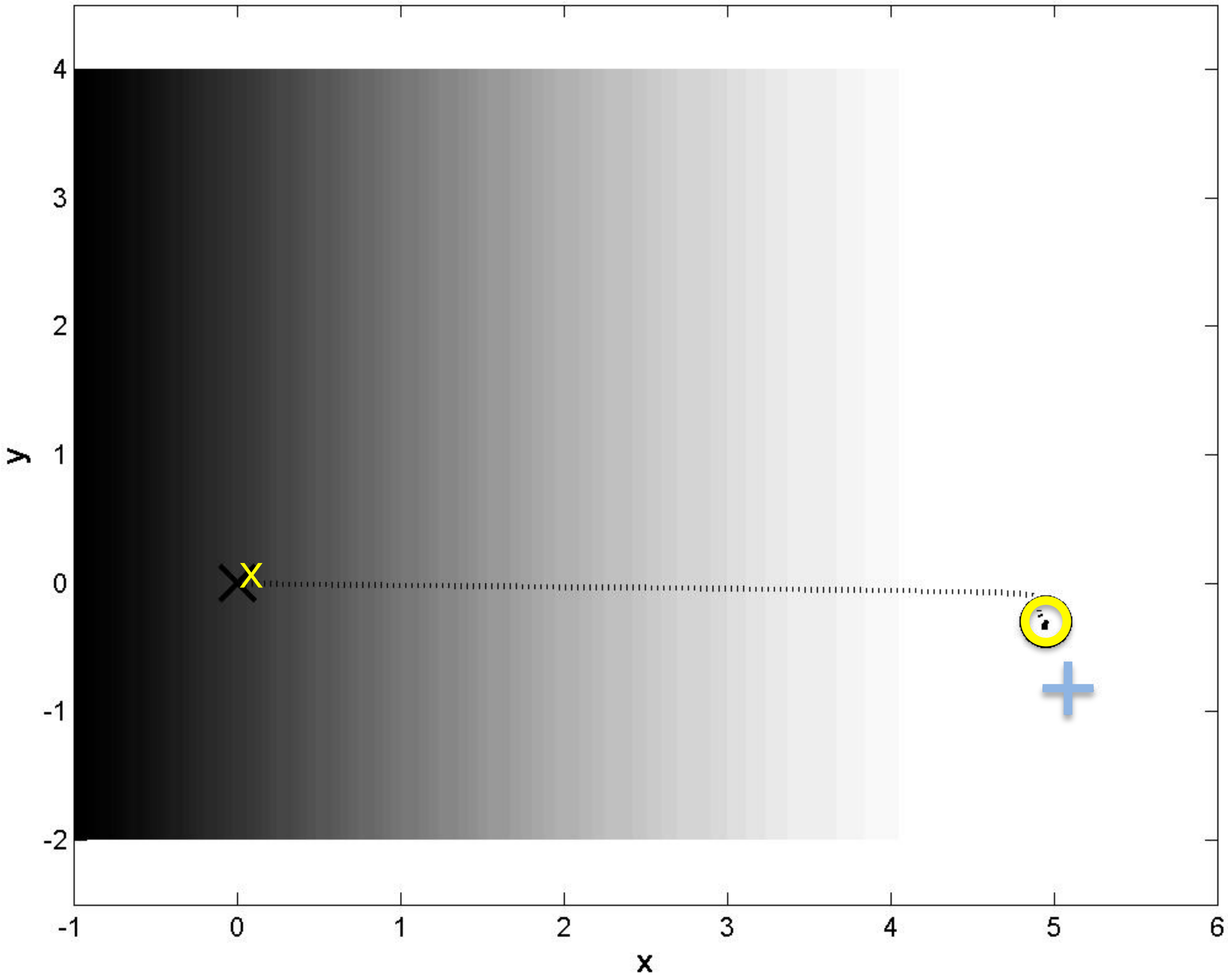
Replanning: light-dark problem



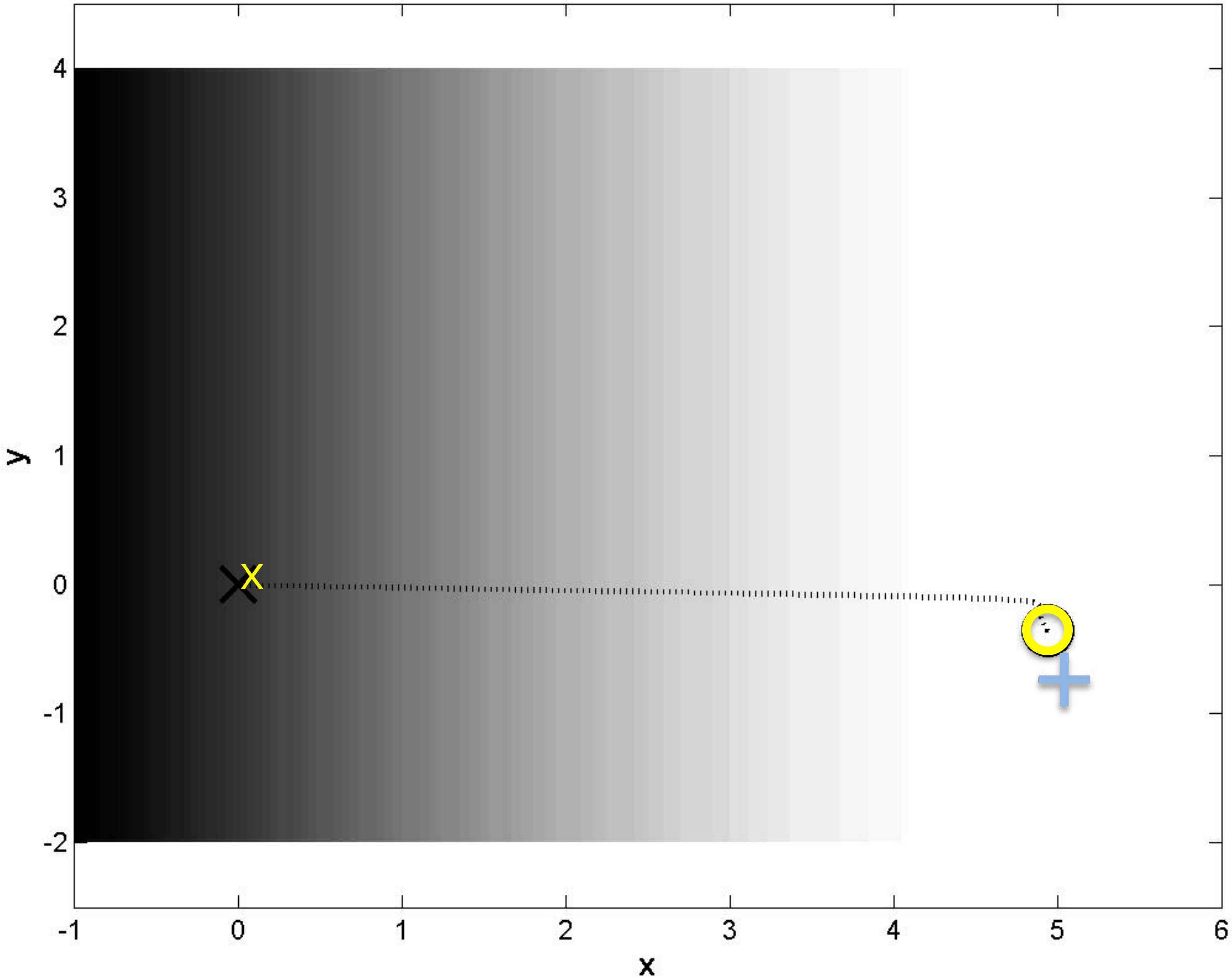
Replanning: light-dark problem



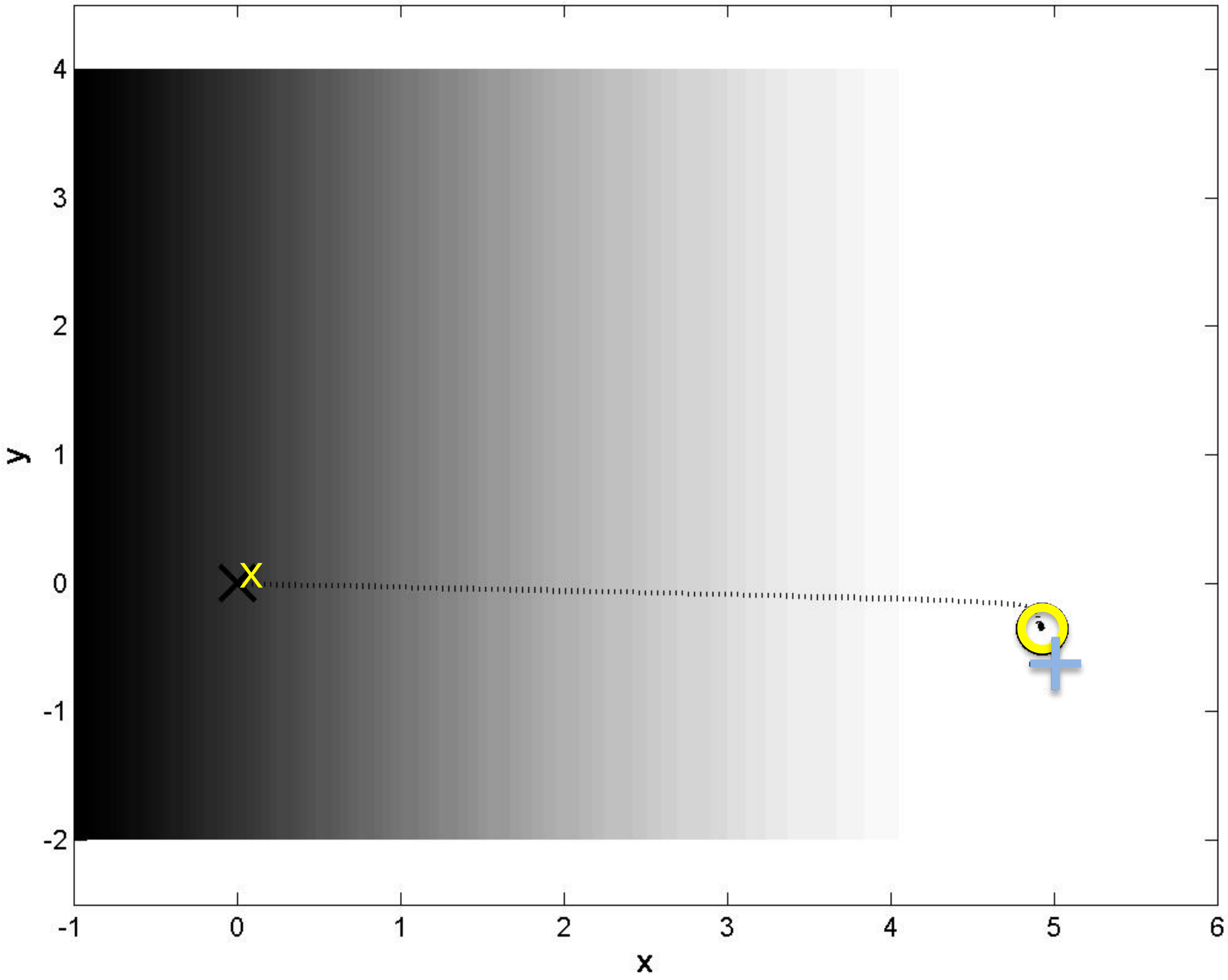
Replanning: light-dark problem



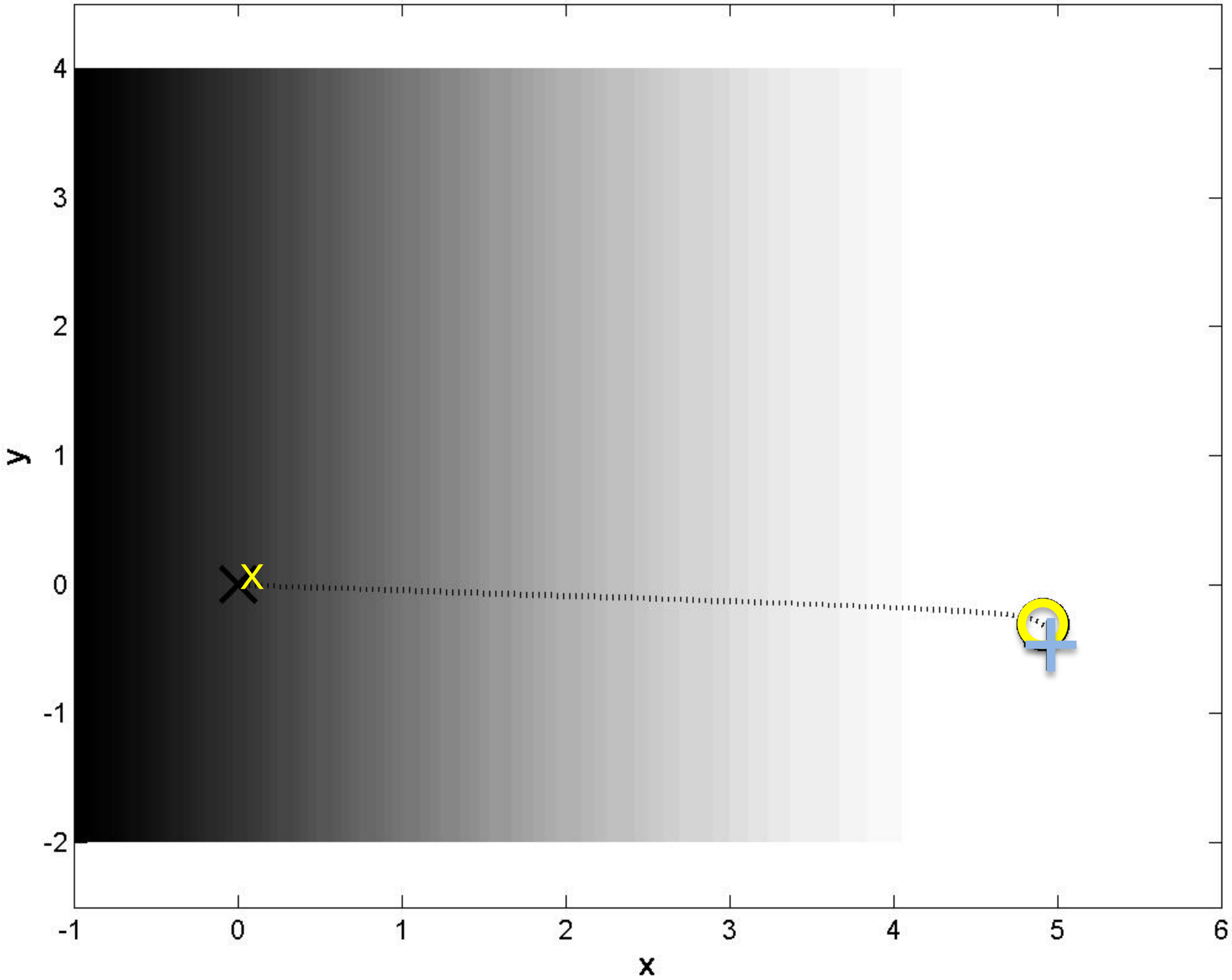
Replanning: light-dark problem



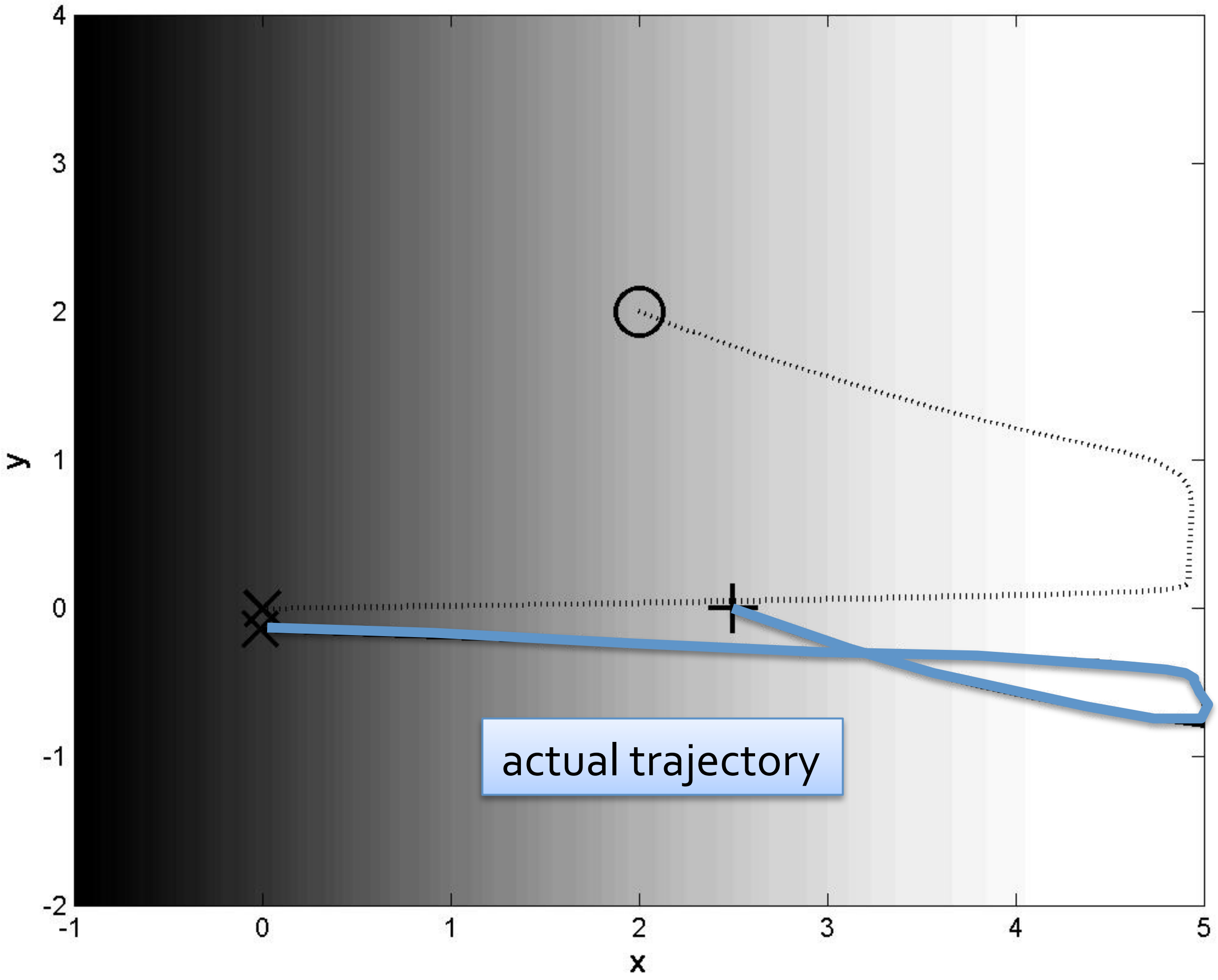
Replanning: light-dark problem



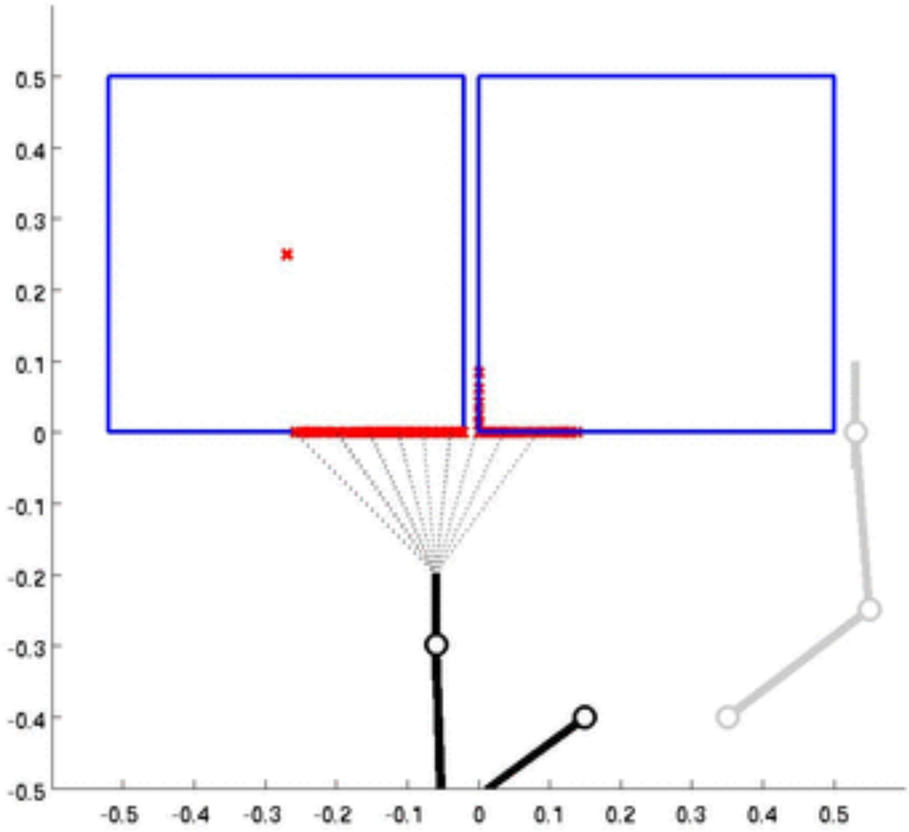
Replanning: light-dark problem



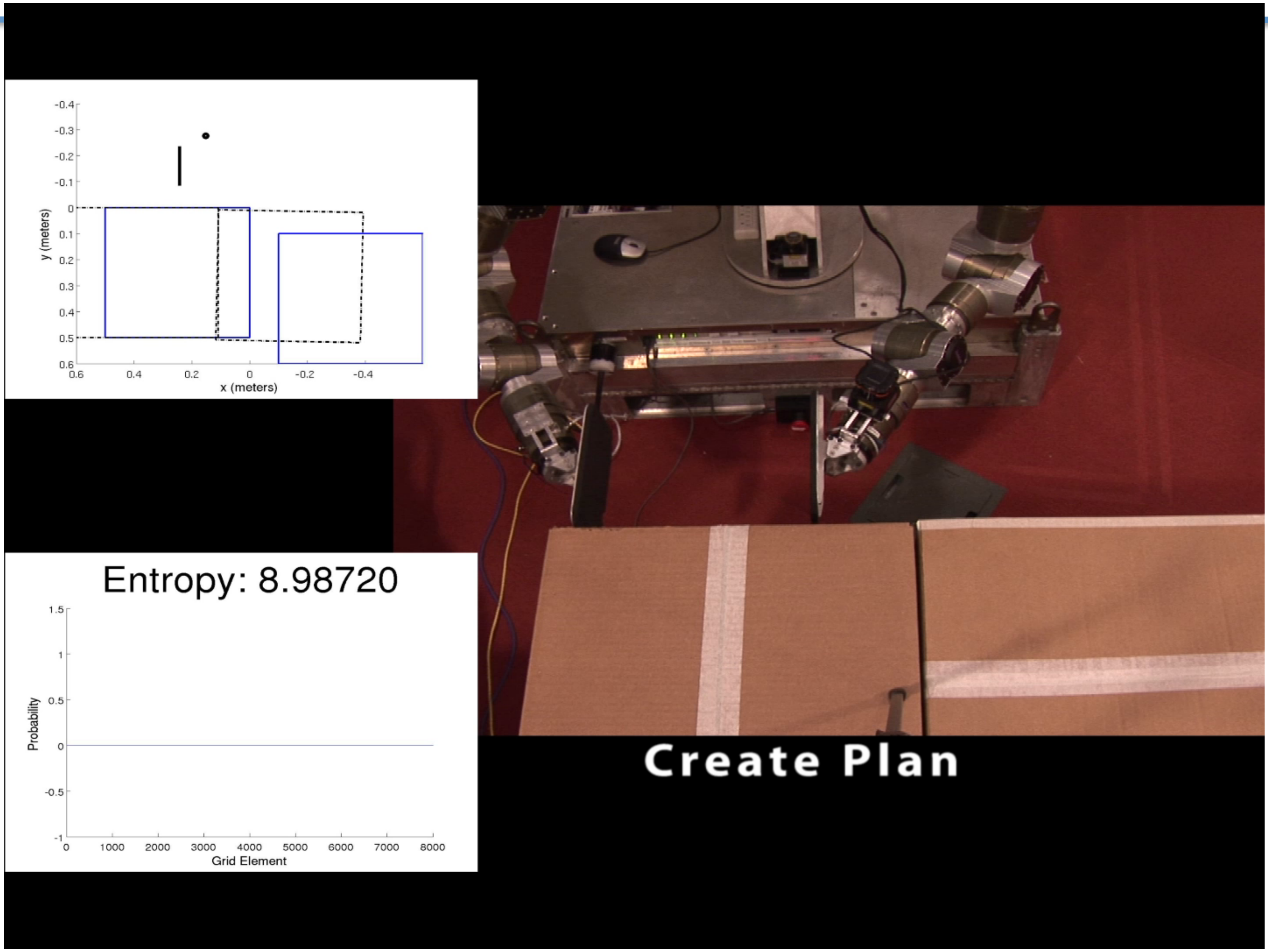
Replanning: light-dark problem



Box Pushing



Paddles



-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5
0.6
y (meters)
0.6
0.4
0.2
0
-0.2
-0.4
x (meters)
Entropy: 8.98720
1.5
1
0.5
0
-0.5
-1
Probability
0
1000
2000
3000
4000
5000
6000
7000
8000
Grid Element
Create Plan

Using simplified models for action selection

Three examples in partially observable domains

Continuous control with state-dependent observation noise:

- deterministic dynamics
- most likely observation

Robot grasping with tactile sensing:

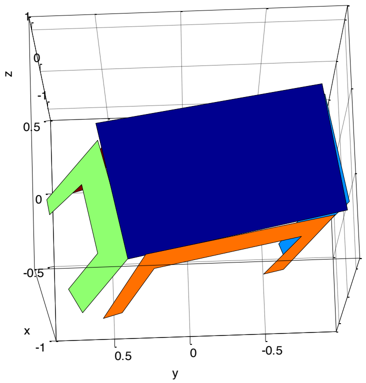
- shortened horizon
- reduced action space

Household robot with local sensing:

- assume subtask serializability
- assume desired observations

Goal: pick up object of known shape with specific grasp

Visual localization and detection works moderately well…



Joint work with Kaijen Hsiao and Tomás Lozano-Pérez

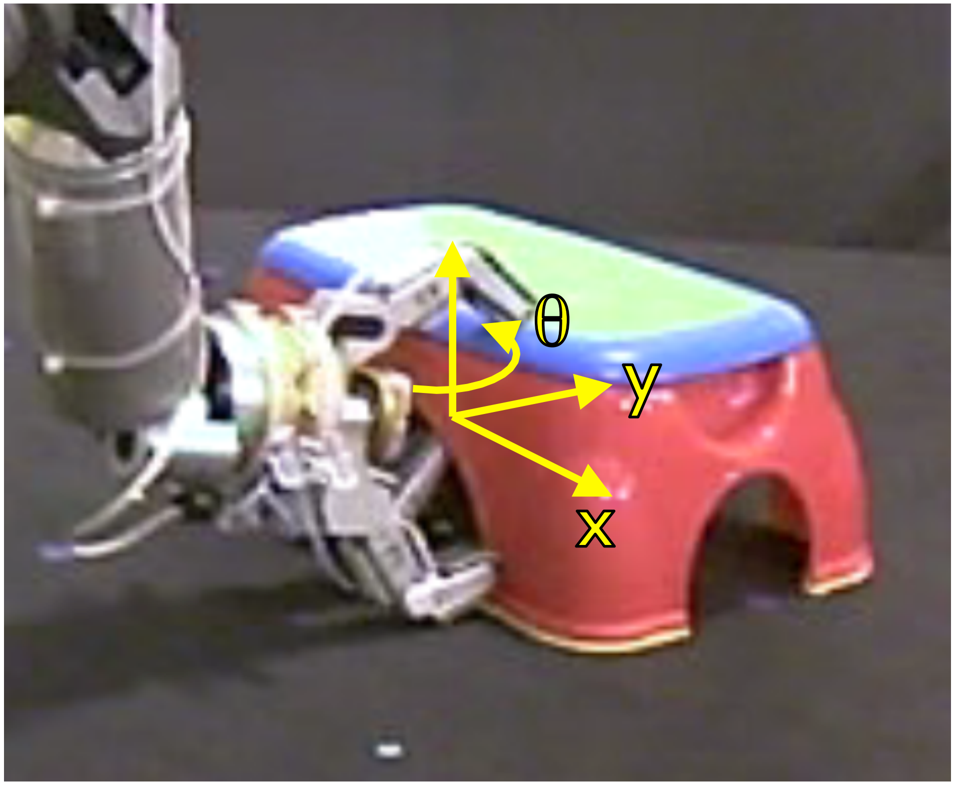
Hypothesis space

Robot pose:

- 11 DOF
- model as fully observable

Object pose:

- 3 DOF
- model as partially observable

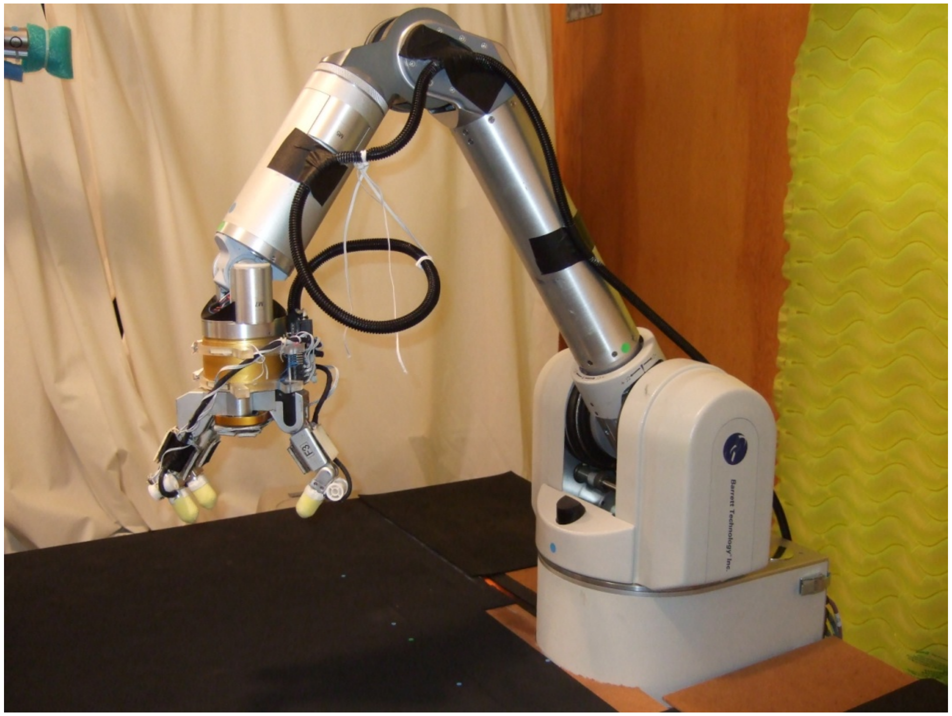


State estimate: probability distribution over object pose

Macro actions

Execute a trajectory:

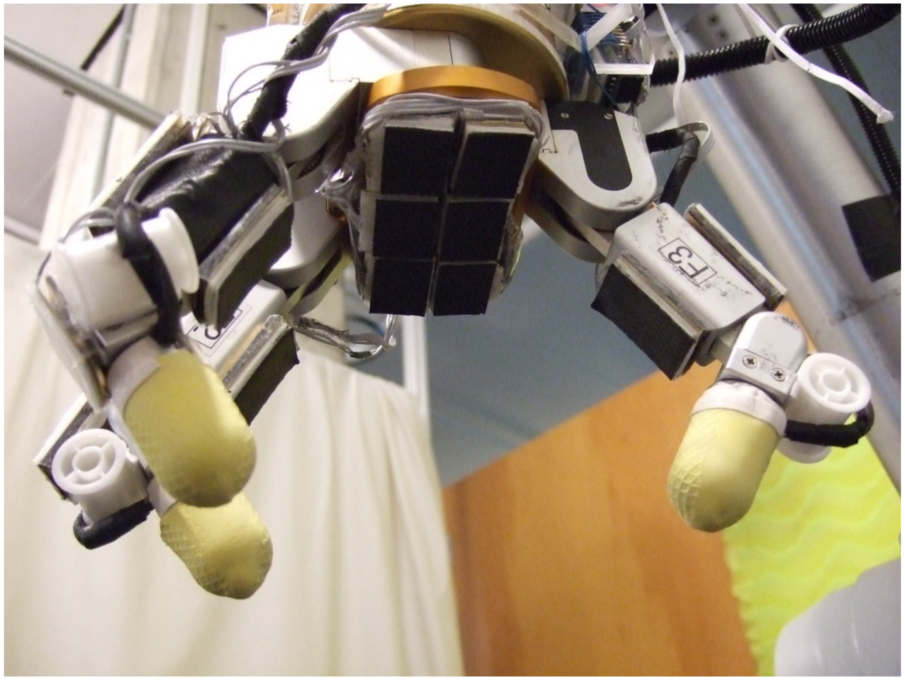
- stop moving arm if any contact is felt
- close each finger until it makes contact



Fixed set of parameterized trajectories, always executed with respect to most likely state

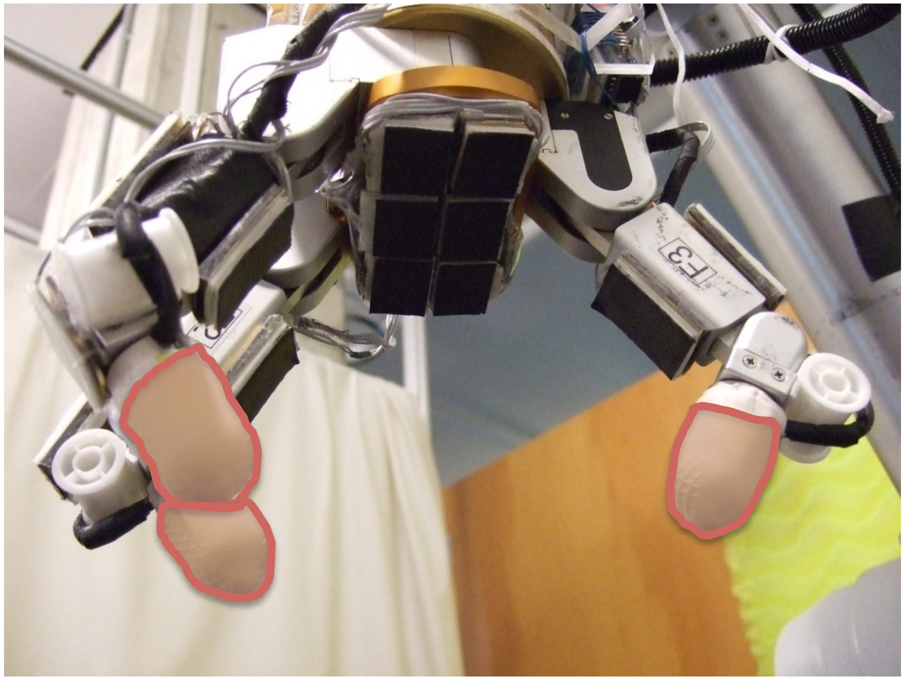
Observations

- Arm trajectory according to proprioception



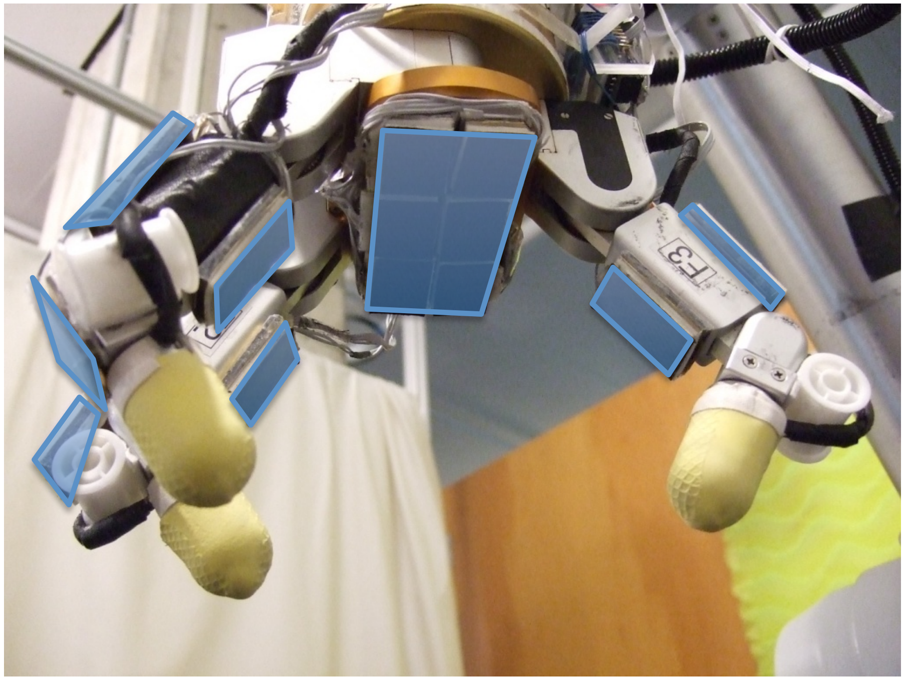
Observations

- Arm trajectory according to proprioception
- 6-axis force-torque sensors on fingertips



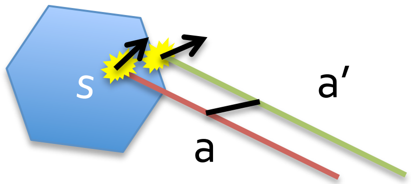
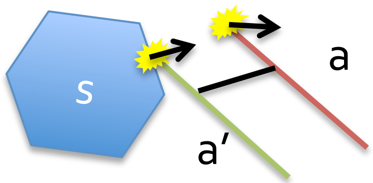
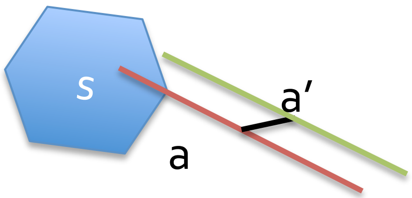
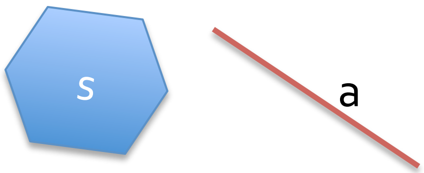
Observations

- Arm trajectory according to proprioception
- 6-axis force-torque sensors on fingertips
- Binary contact sensors



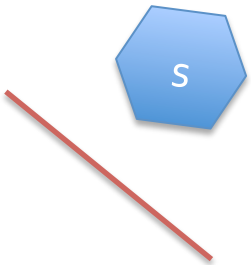
Observation model: $\Pr(o \mid s, a)$

Nominal observation for s, a: o*

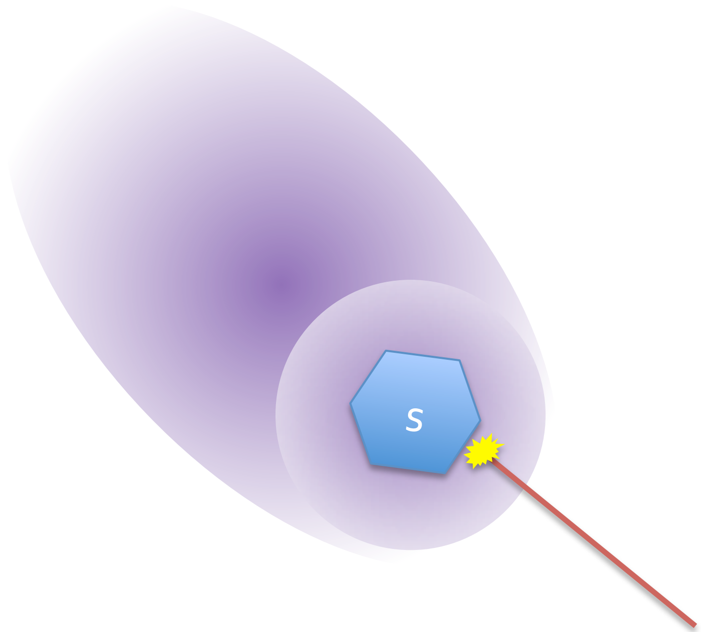
		Contact	no contact
Actual o	Contact	Gaussian density on dist to closest a' that would not have caused interpenetration X Gaussian density on dist between contact positions and normals 	Gaussian density on dist to closest a' that would have caused contact X Gaussian density on dist between contact positions and normals 
	no contact	Gaussian density on dist to closest a' that would not have caused contact 	Max value of Gaussian density used for nominal contact case 

Transition model: $\Pr(s_{t+1} \mid s_t, a_t)$

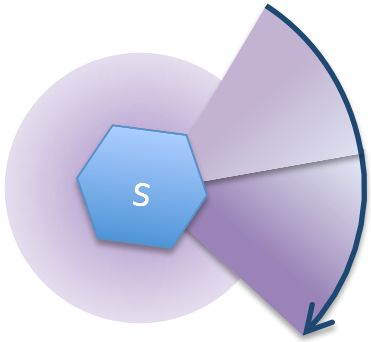
No contact: no change



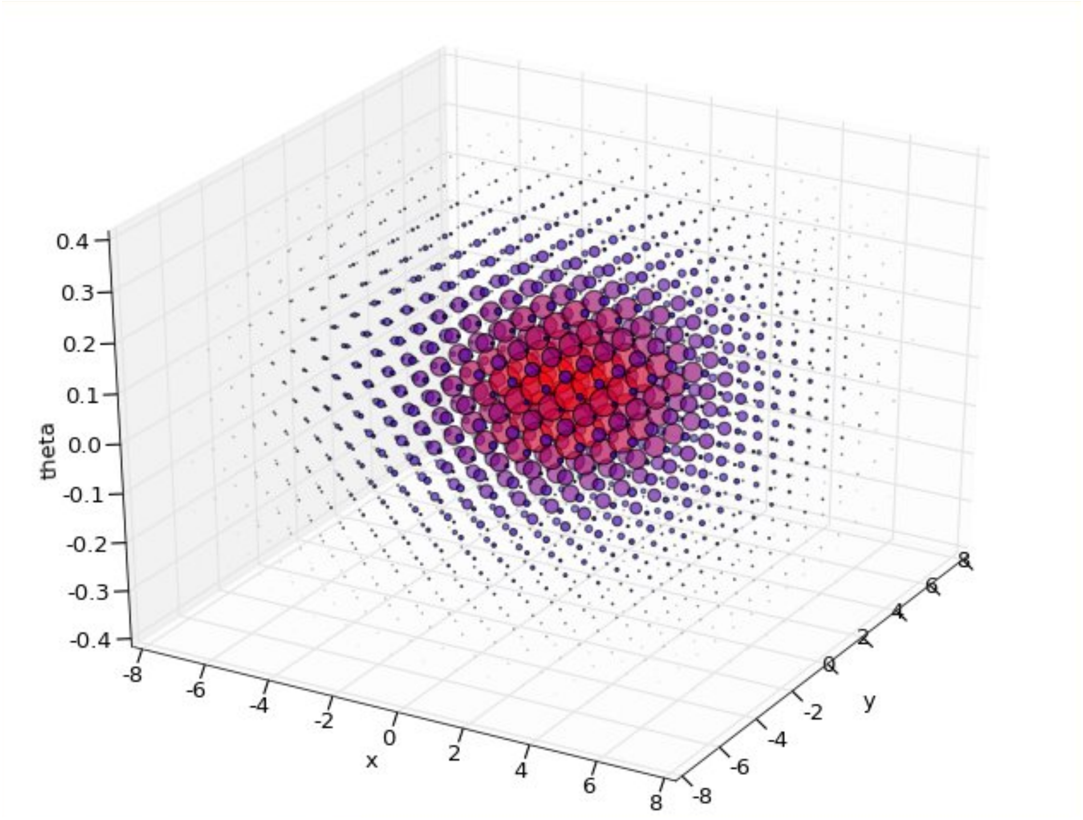
Contact: probability of being bumped depends on observation



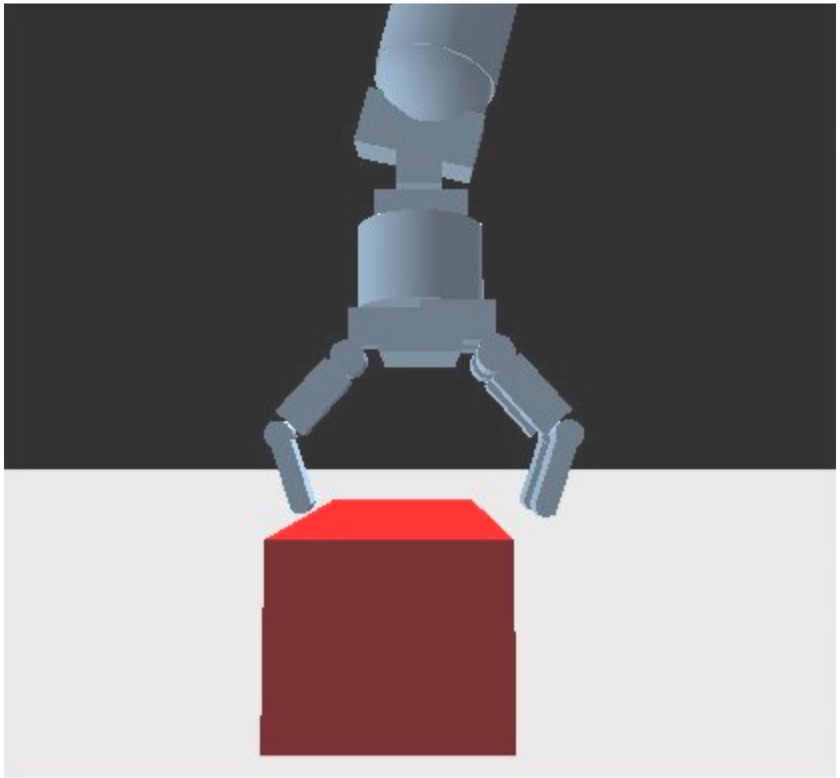
Reorientation: similar to contact with large rotational variances



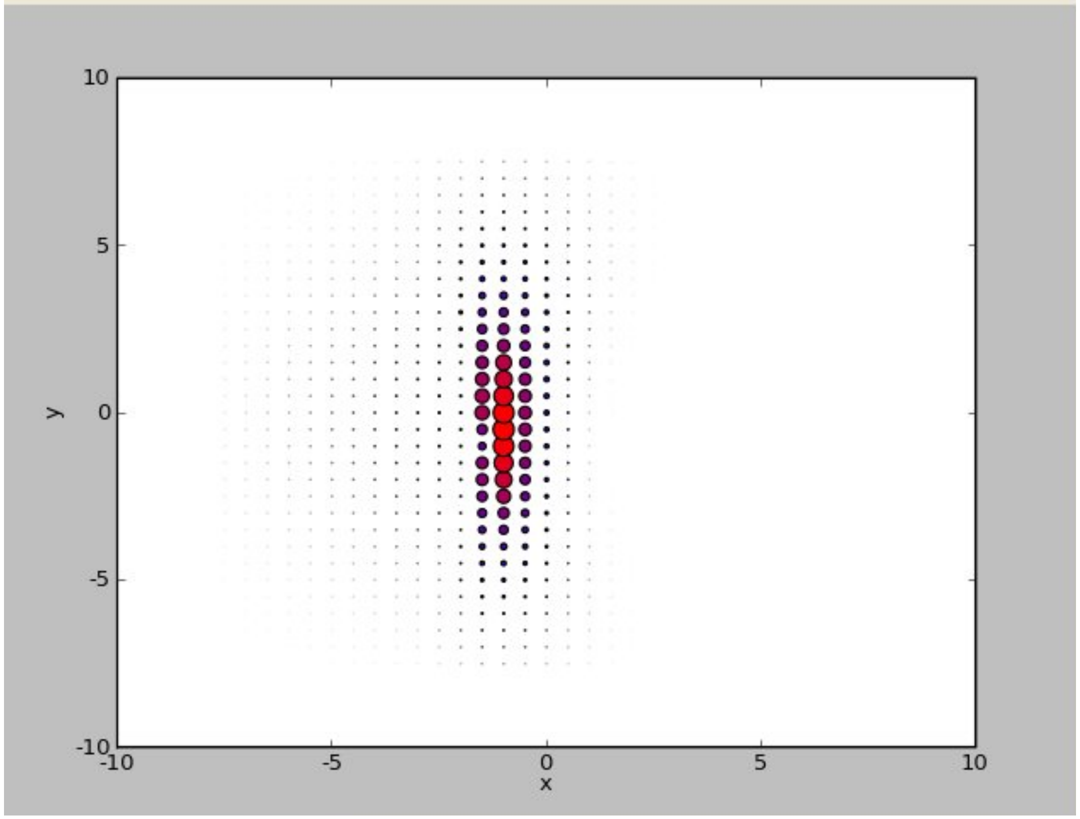
Initial belief state



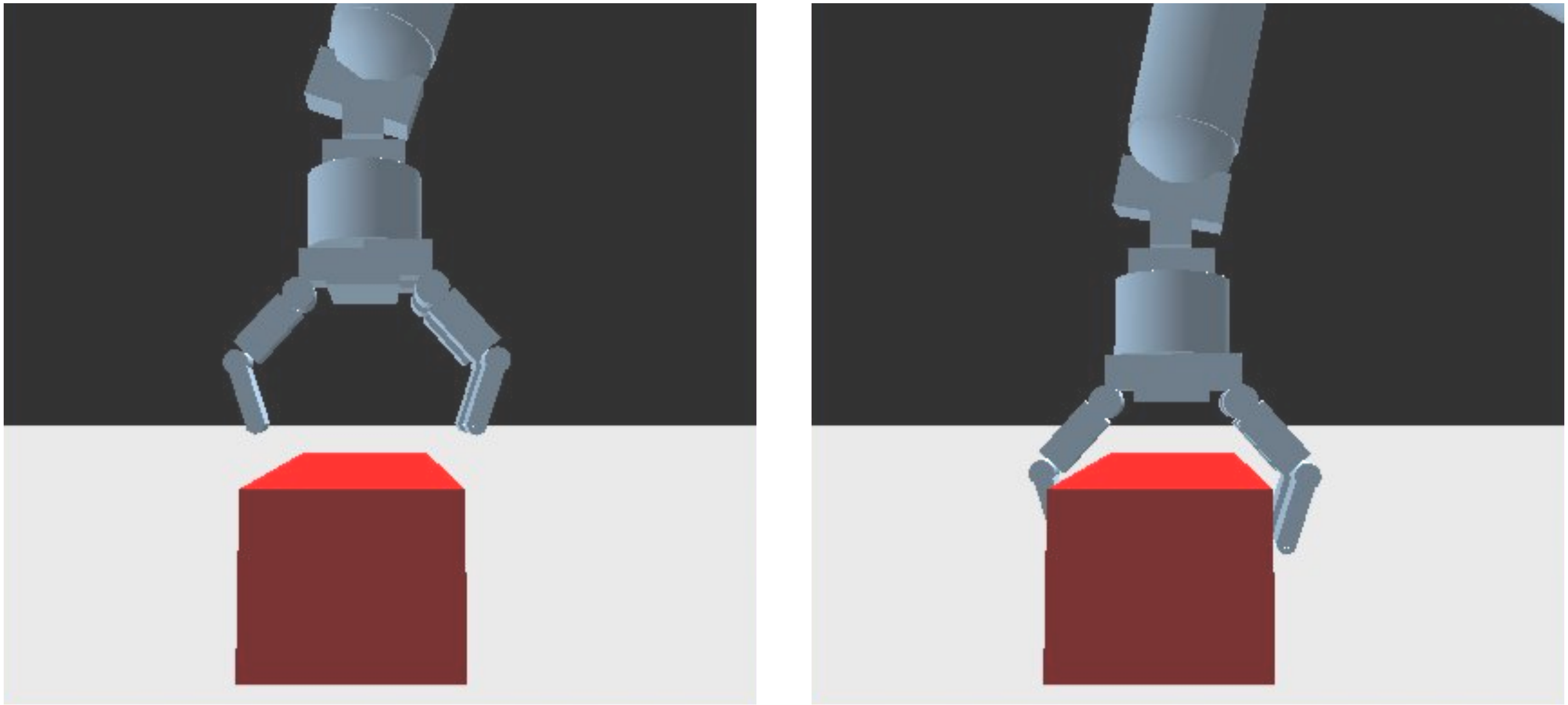
Tried to move down — finger hit corner



Updated belief

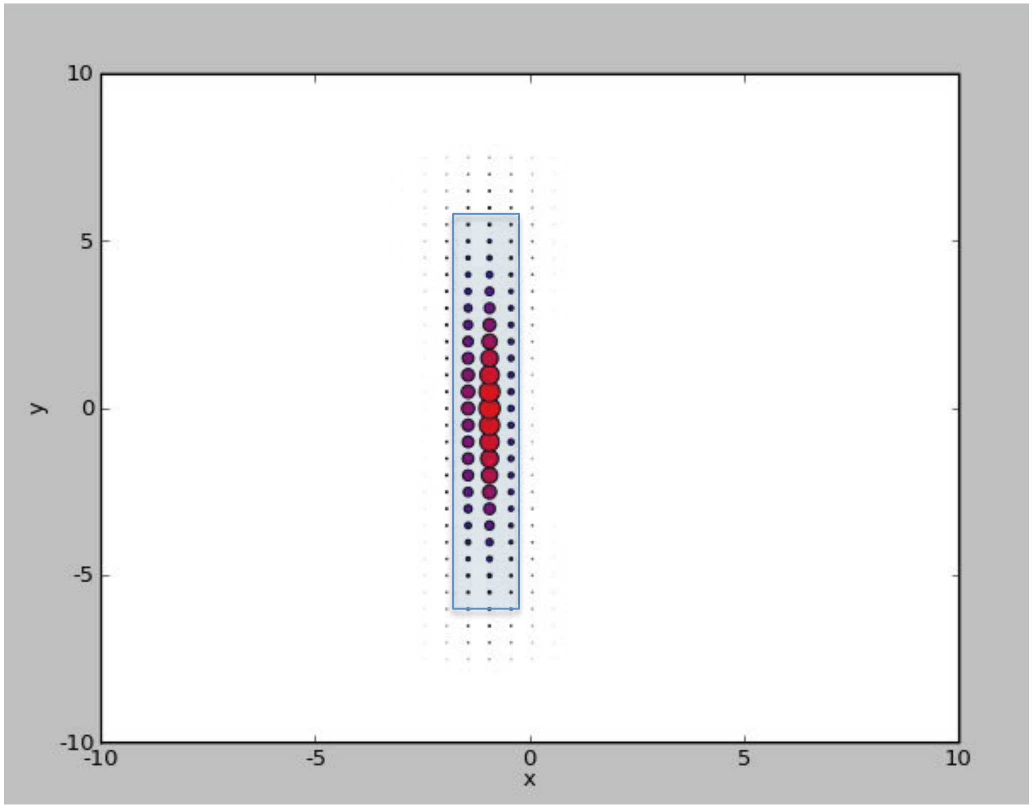


Another grasp attempt

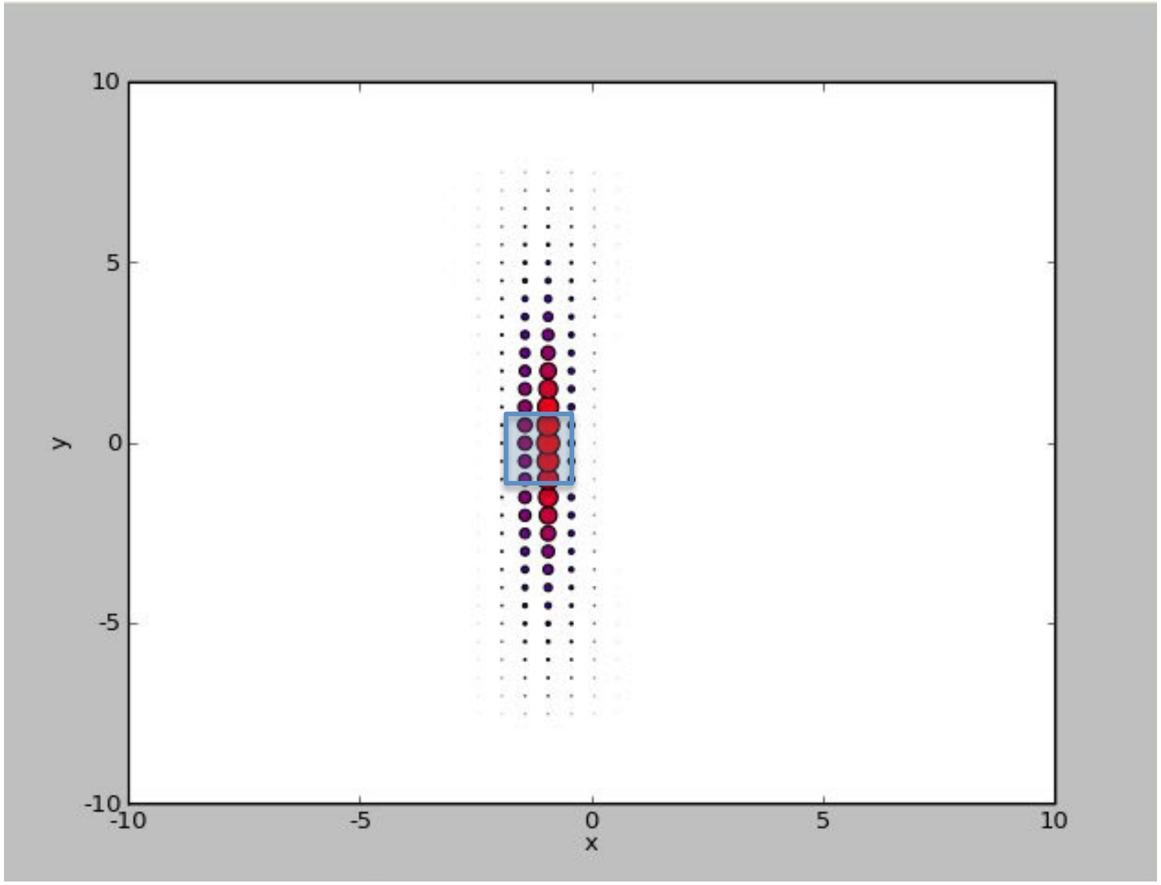


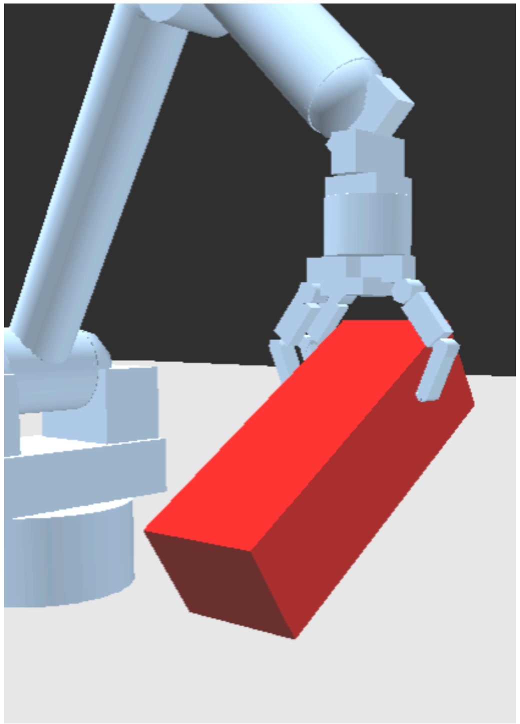
Goals in belief space

- Specify set of desirable ranges in X, Y, ϴ
- Satisfied if probability that the pose is in that set is high



What if Y coordinate of grasp matters?





Action selection

How to select among the actions?

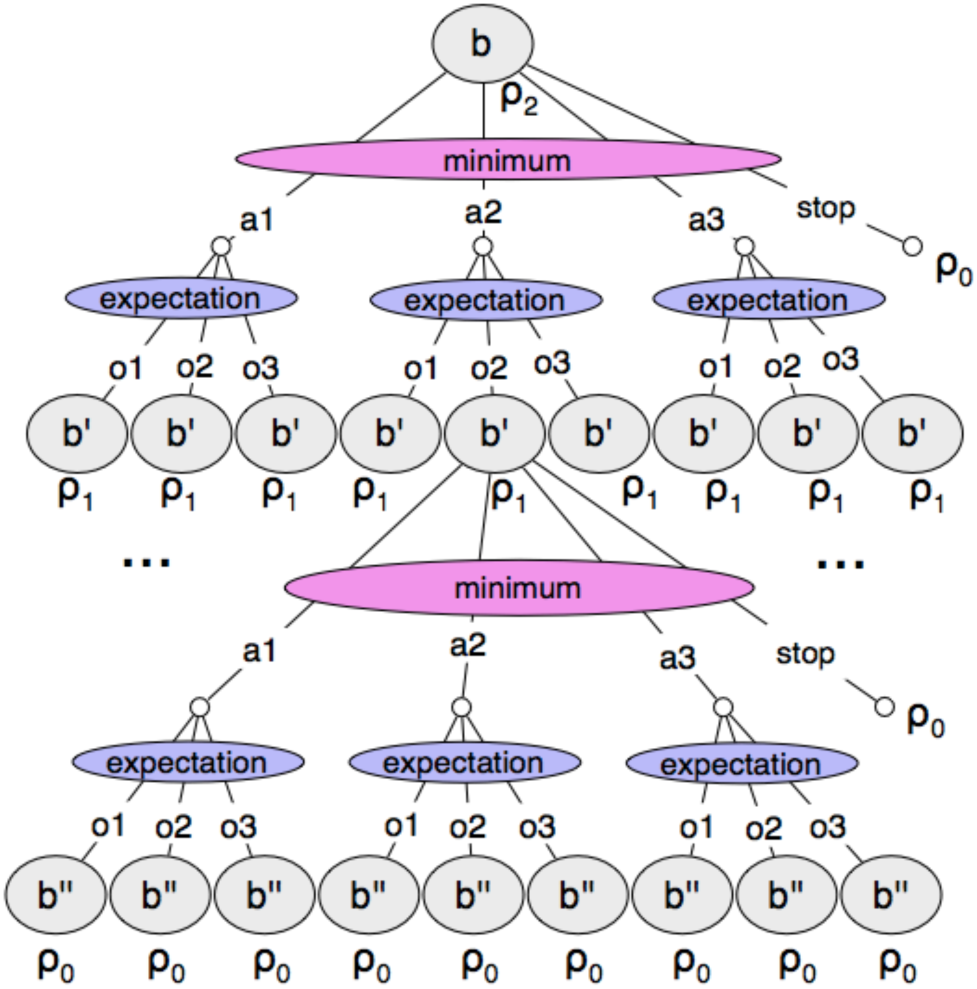
- Until probability of failure given belief is < eps
  - Select WRT by searching forward from belief
  - Execute WRT, and get observations o
  - Update belief

WRTs include:

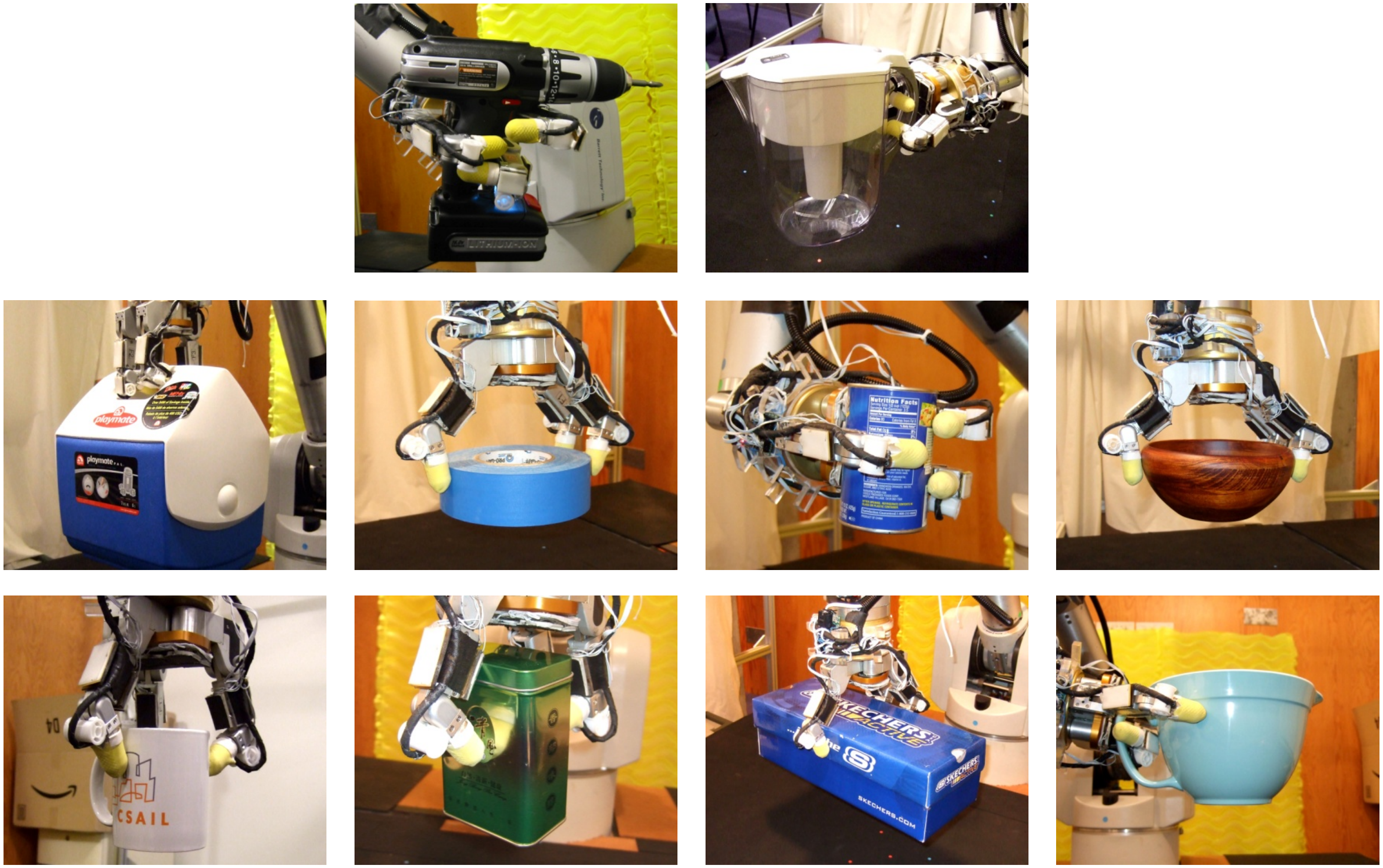
- target grasp
- information-gain trajectories
- re-orientation

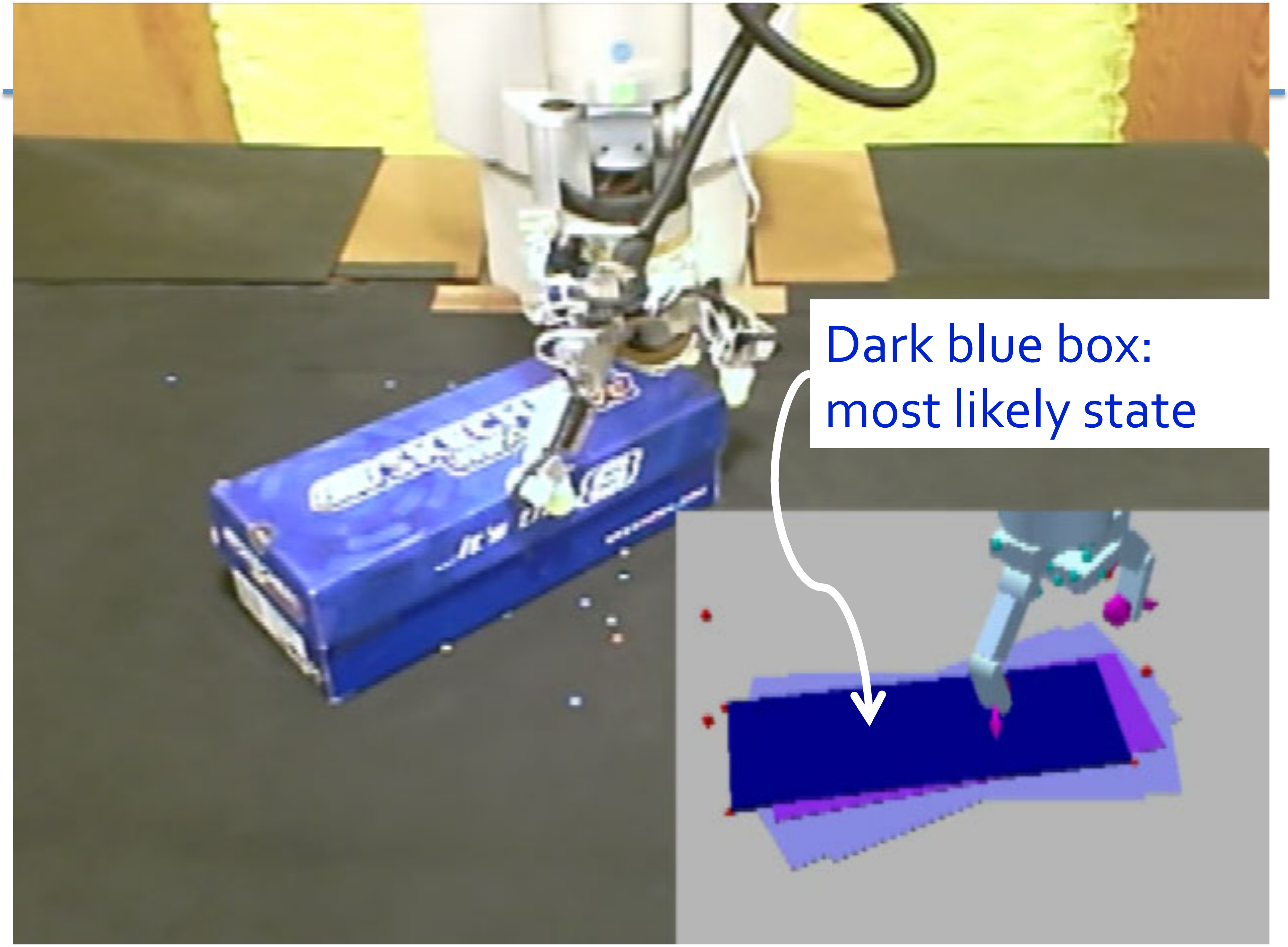
Forward search

- Compute k-step risk using backward induction
- **Prune and cluster to decrease observation branching**
- Depth 2 sufficed in our problems
- Risk at leaves is likelihood of failure of target grasp

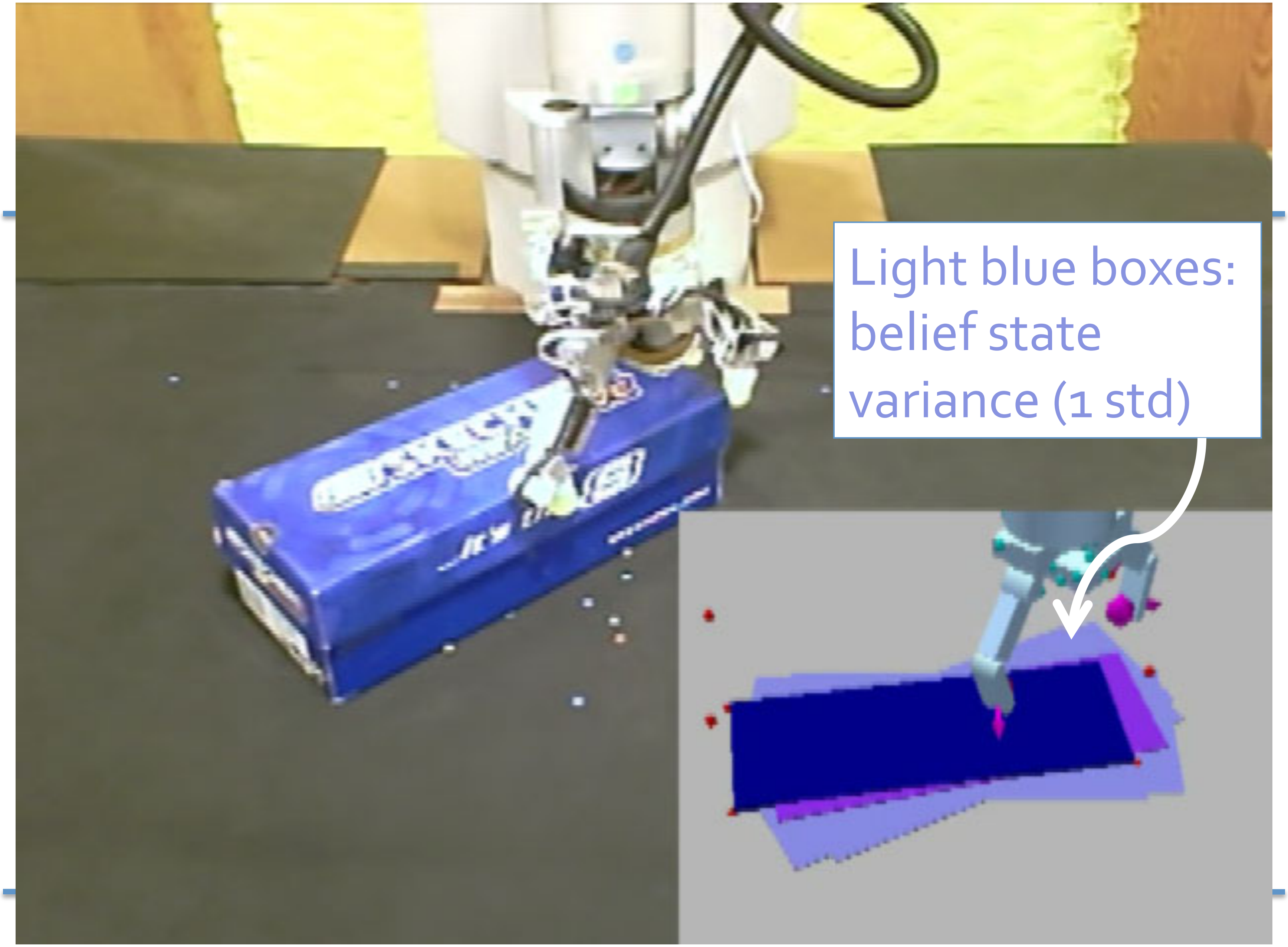


Objects and desired grasps





Dark blue box: most likely state



Light blue boxes: belief state variance (1 std)

Brita results: 10 / 10 successful grasps

Grasping a
Brita Pitcher
50x
Low deviation

Powerdrill: 10 / 10 successful grasps

25x
speed

Using simplified models for action selection

Three examples in partially observable domains

Continuous control with state-dependent observation noise:

- deterministic dynamics
- most likely observation

Robot grasping with tactile sensing:

- shortened horizon
- reduced action space

Household robot with local sensing:

- assume subtask serializability
- assume desired observations

Real robot meets real world: large spaces

Configuration space

- joint angles of robot
- base pose
- positions and orientations of all objects in house
- are the dishes clean?
- is the stove on?
- are the leftovers edible?

Joint work with Tomás Lozano-Pérez

Real robot meets real world: long horizon

Steps to clean kitchen:

- put away food (10 items)
- wash dishes (40 items)
- put away dishes (40 items)
- clean surfaces (10 items)

Or:

- 1,000 pick, place, or wipe

Or:

- 100,000 linearly interpolated joint motions

Real robot meets real world: uncertainty

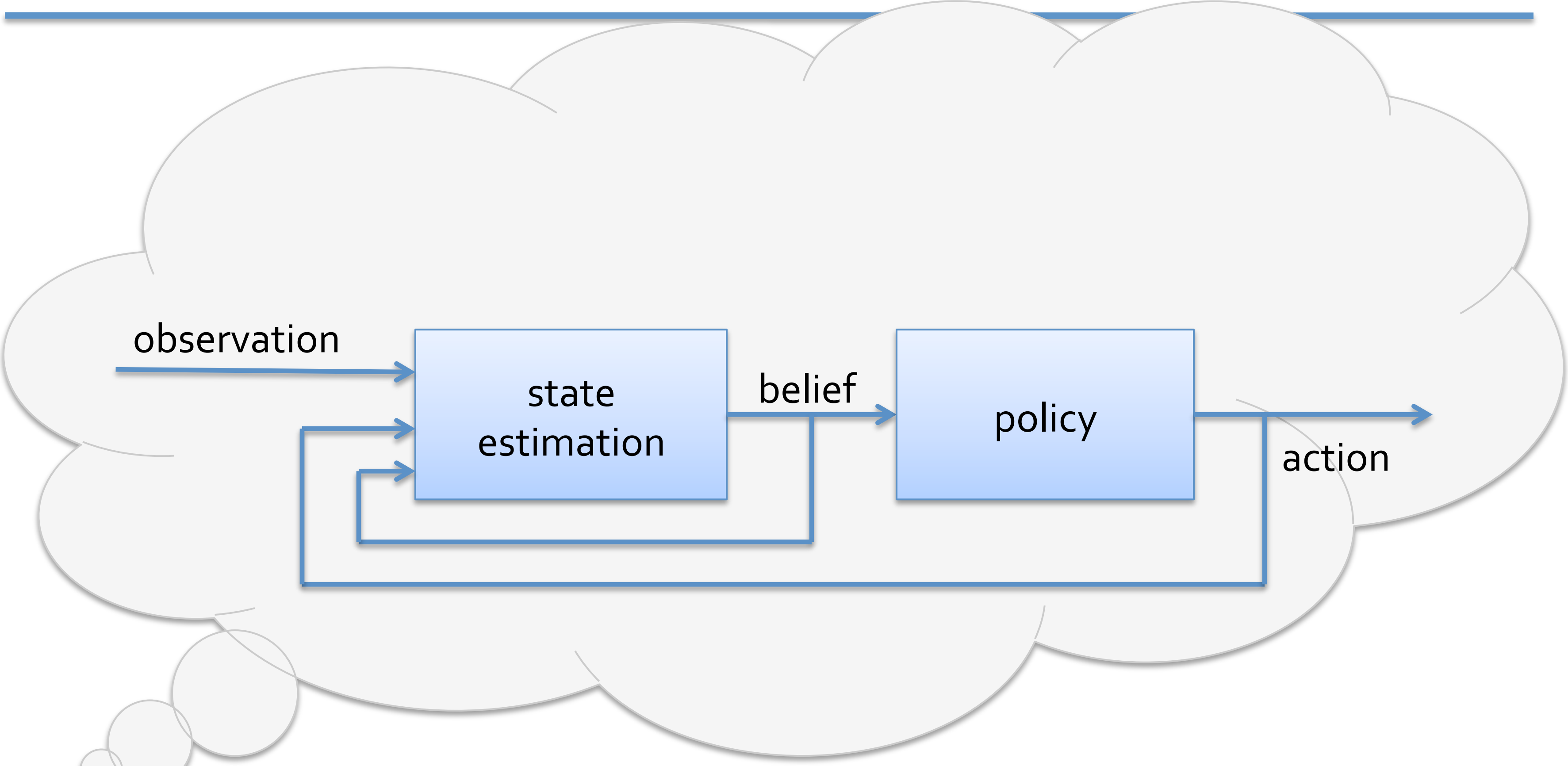
Current-state uncertainty

- What is inside the tupperware?
- Is the dishwasher clean?
- What is the exact pose of the pot?
- What's the friction of a wet dish?

Predictive uncertainty

- What will happen when the robot lifts the cookie sheet?
- What is the error in the motor control?
- When will the inhabitants come home?

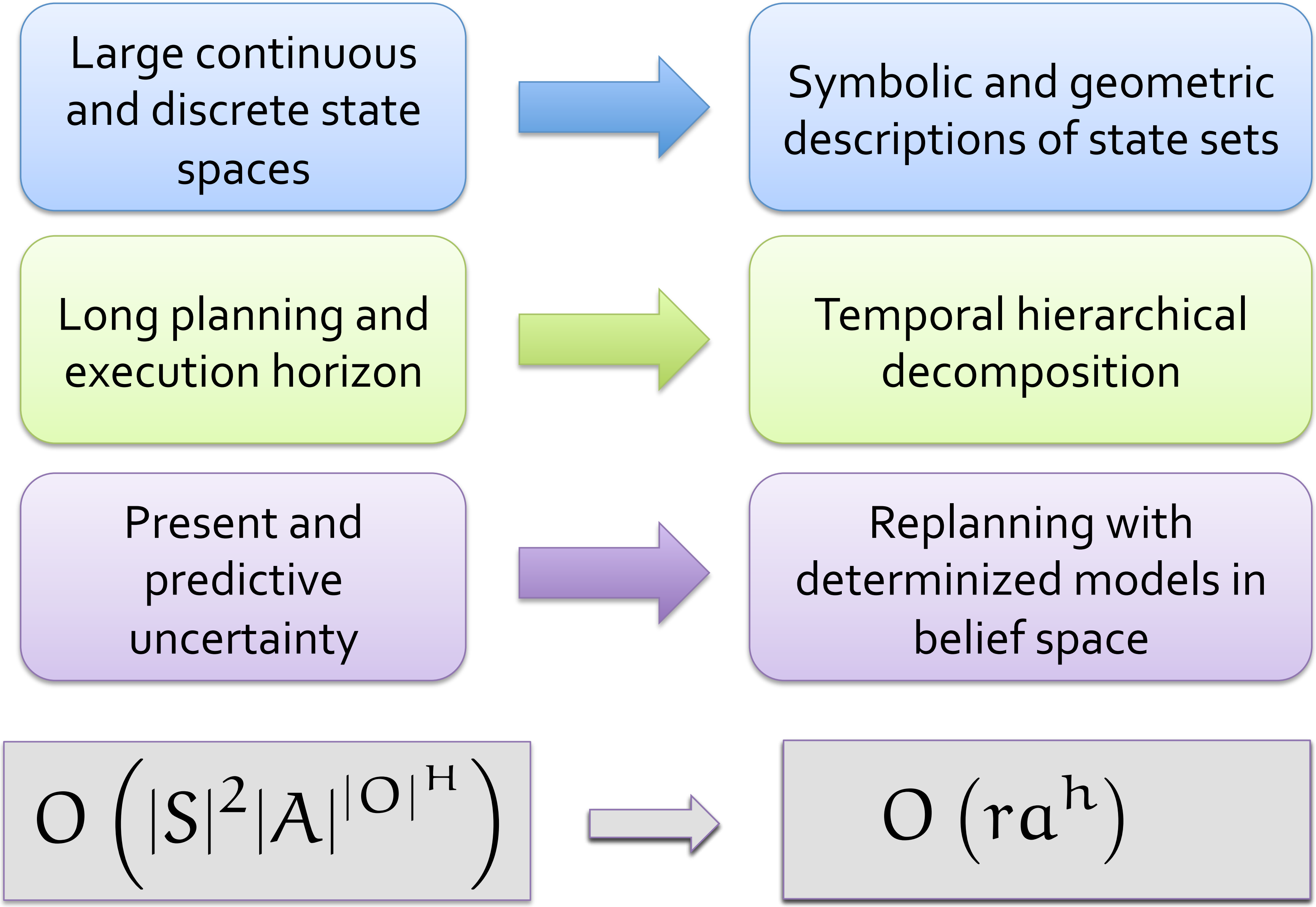
POMDP: Optimal solution is complex…



Policy is a total function
Solution to MDP in B space

$$O\left(|S|^2|A|^{|O|^H}\right)$$

Addressing real world challenges



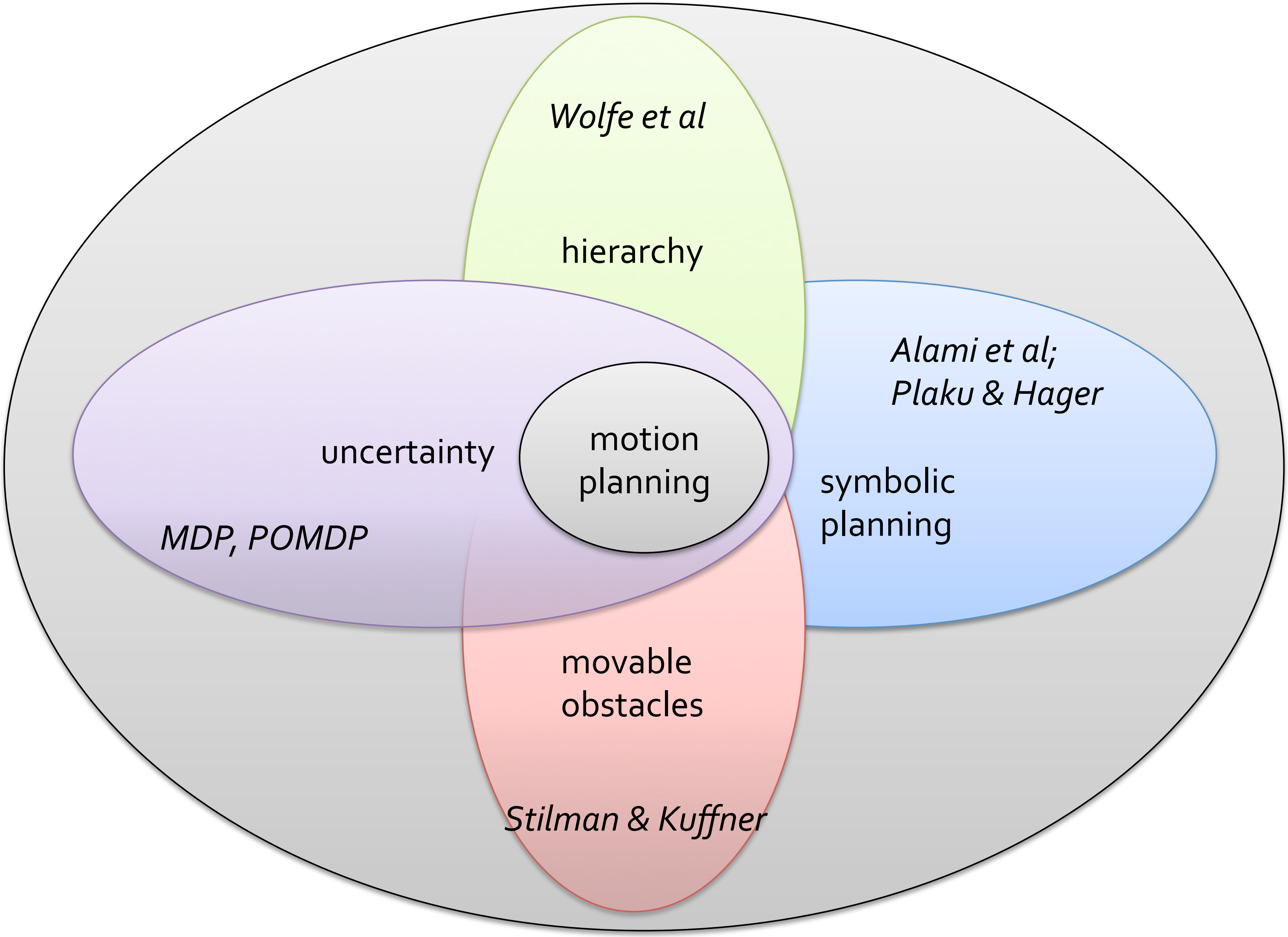
Large continuous and discrete state spaces → Symbolic and geometric descriptions of state sets

Long planning and execution horizon → Temporal hierarchical decomposition

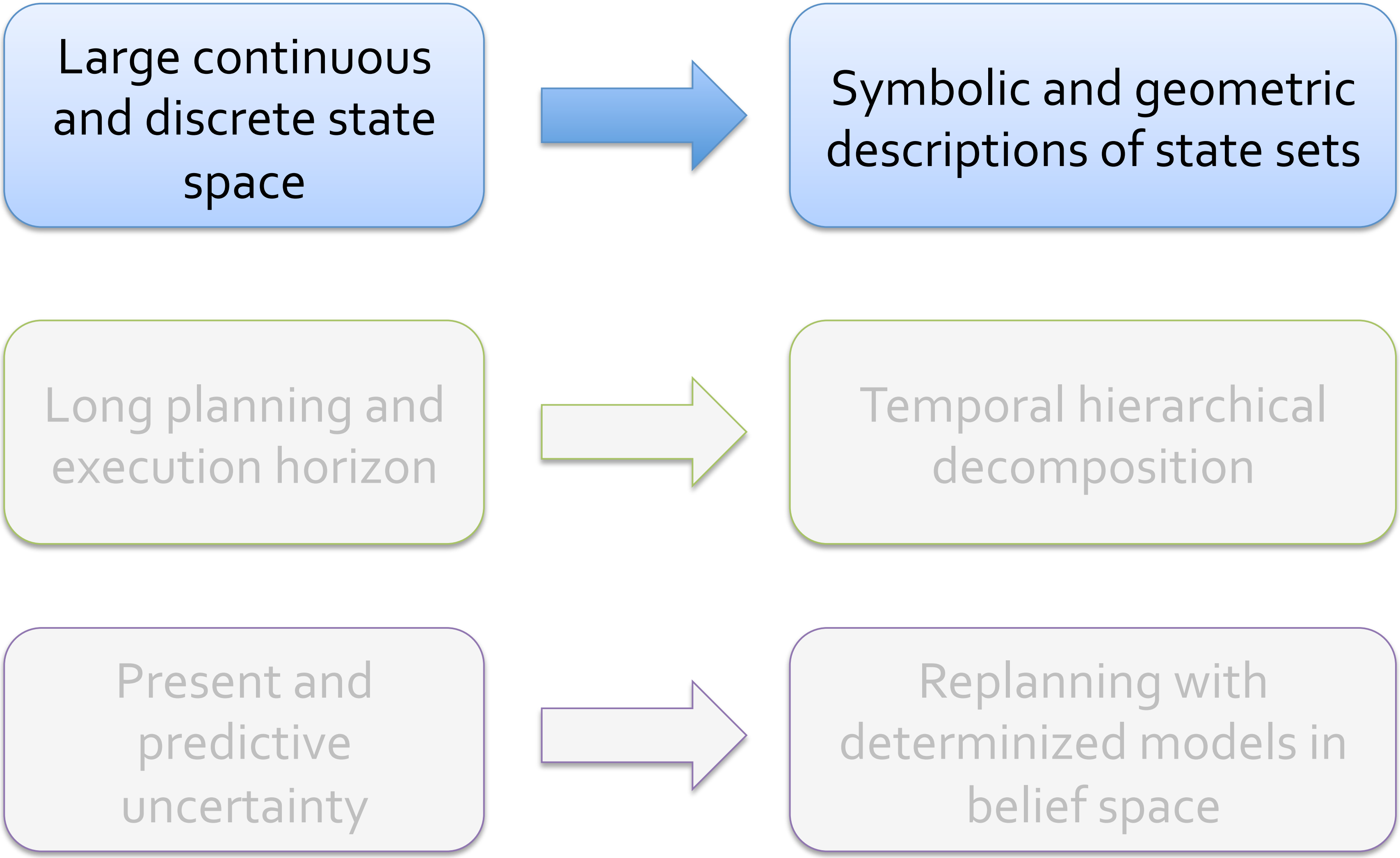
Present and predictive uncertainty → Replanning with determinized models in belief space

$O\left(|S|^2|A|^{|O|^H}\right)$ → $O\left(ra^h\right)$

Some related approaches



Addressing real world challenges



Large continuous and discrete state space → Symbolic and geometric descriptions of state sets

Long planning and execution horizon → Temporal hierarchical decomposition

Present and predictive uncertainty → Replanning with determinized models in belief space

Symbolic and geometric descriptions of state sets

Symbols for non-geometric properties

- are the dishes clean? ***Clean(dish24)***
- is the stove on? ***On(stove)***
- are the leftovers edible? ***Edible(lasagna)***

Symbols for geometric and physical abstractions

- is the beer in the refrigerator? ***In(beer, refrigerator)***
  - region containment
- is the robot holding the pot? ***Holding(pot)***
  - contacts, force closure, …

Specific start state; abstract goal

Initial state known in geometric detail

Cannot represent all geometric and continuous aspects of the current state symbolically

Goal set is abstract, symbolic

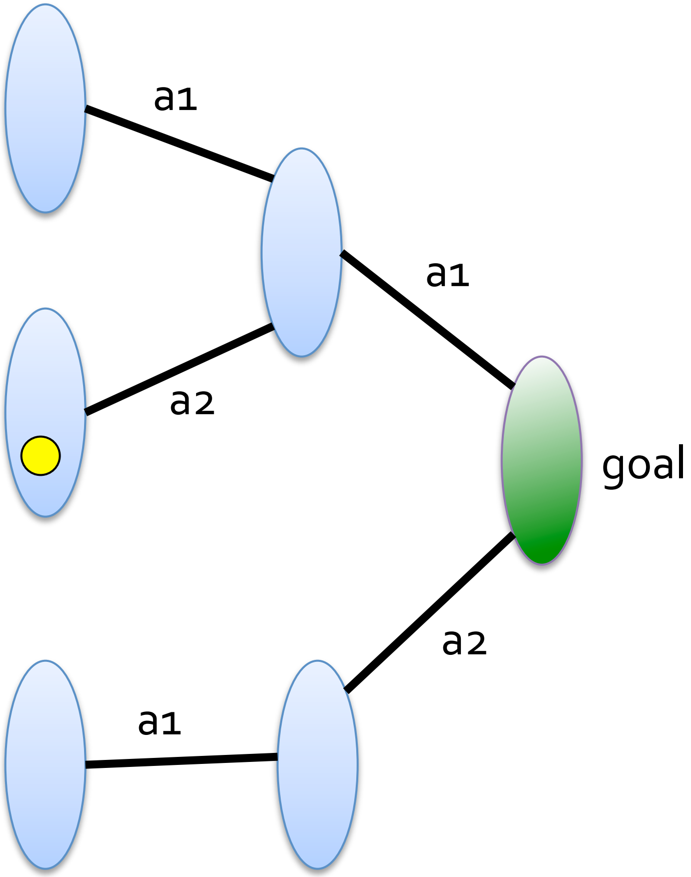
tidy(house) ^ *charged(robot)*

Goal Regression / Pre-image backchaining

Weakest precondition of goal set under each action sequence

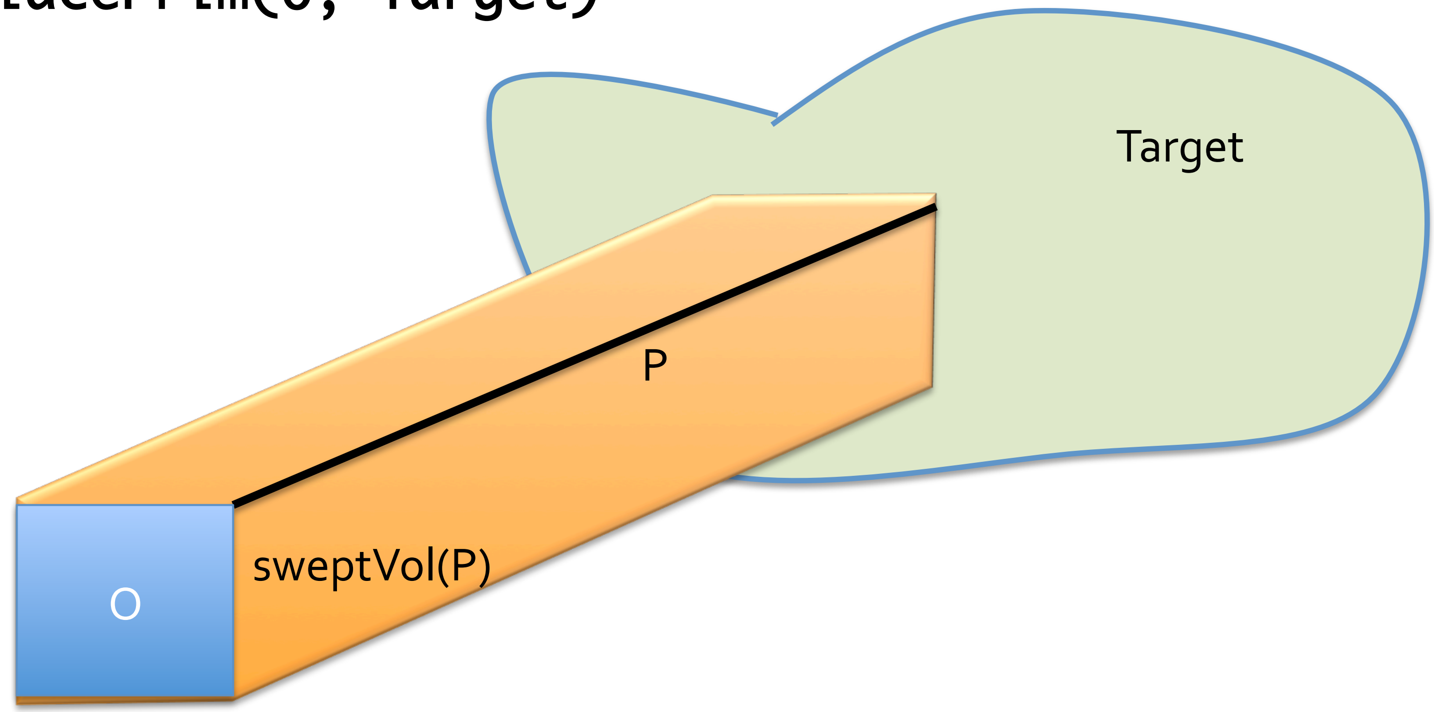
Test whether start state is in a pre-image

Represent goal and pre-images as conjunctions of fluents



Planning operator: Place

```
Place(O, Target):
   exists: P ∈ generatePlacePaths(O, Target)
   pre: ClearX(sweptVol(P), O), Holding(O)
   result: In(O, Target)
   prim: PlacePrim(O, Target)
```

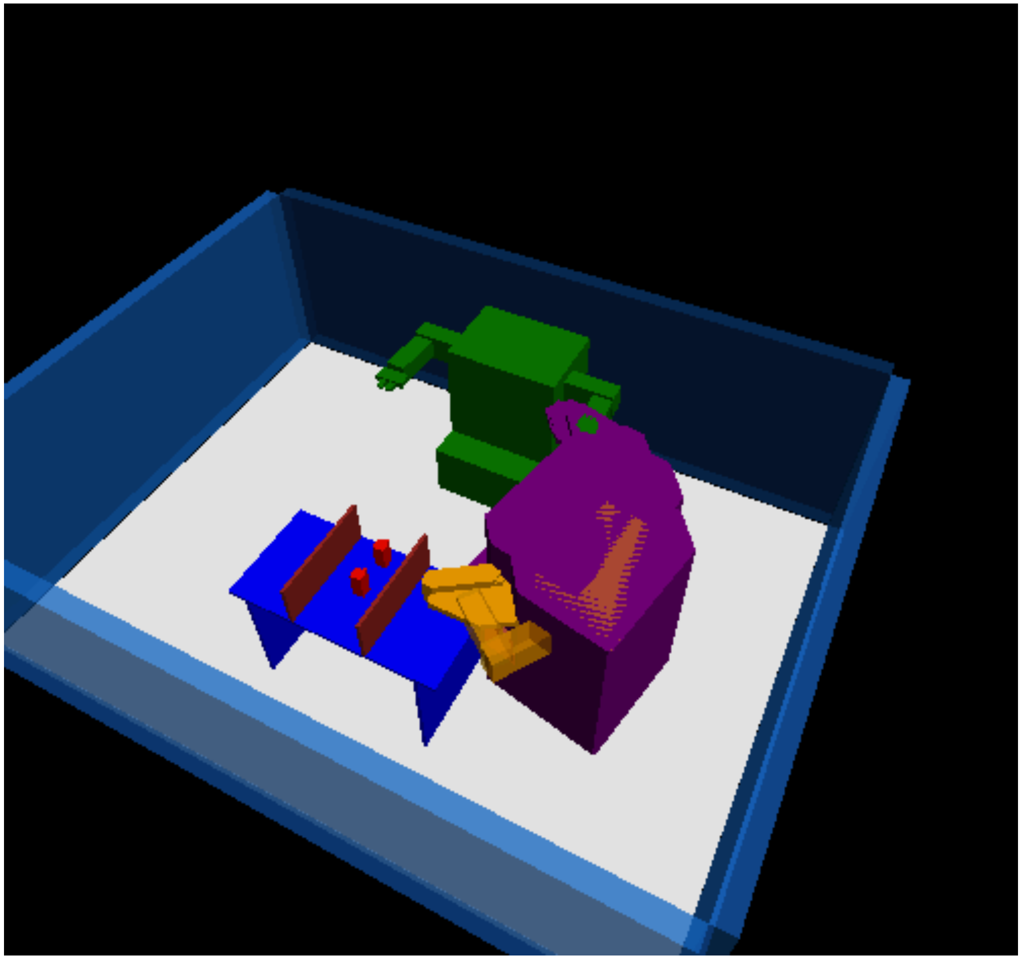


Stilman and Kuffner

Generators

```
Place(O, Target):
   exists: P ∈ generatePlacePaths(O, Target)
```

- visibility graph planner
- fail fast
- conservative
- not the path we'll use: certificate that one exists



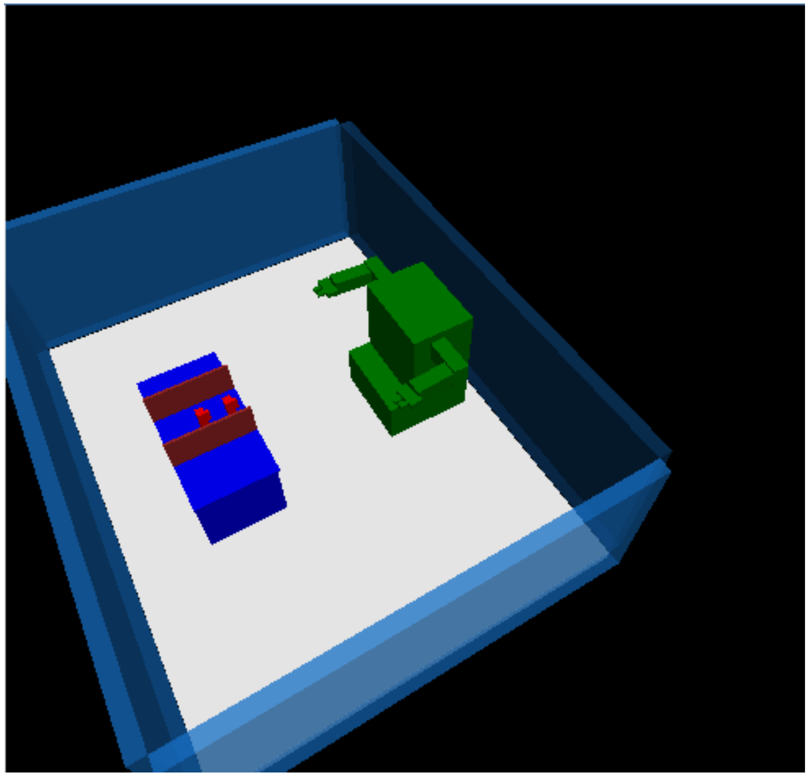
Generators Primitives

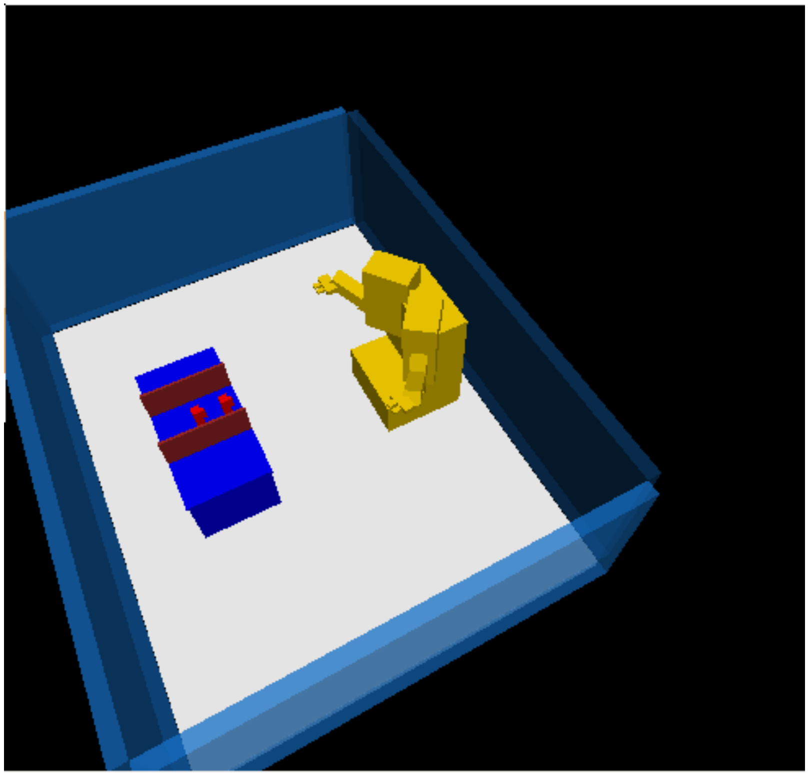
Efficient conservative planner

- Simplified robot model
- Objects: x, y, z, th
- Grasp selection

Your favorite motion planner

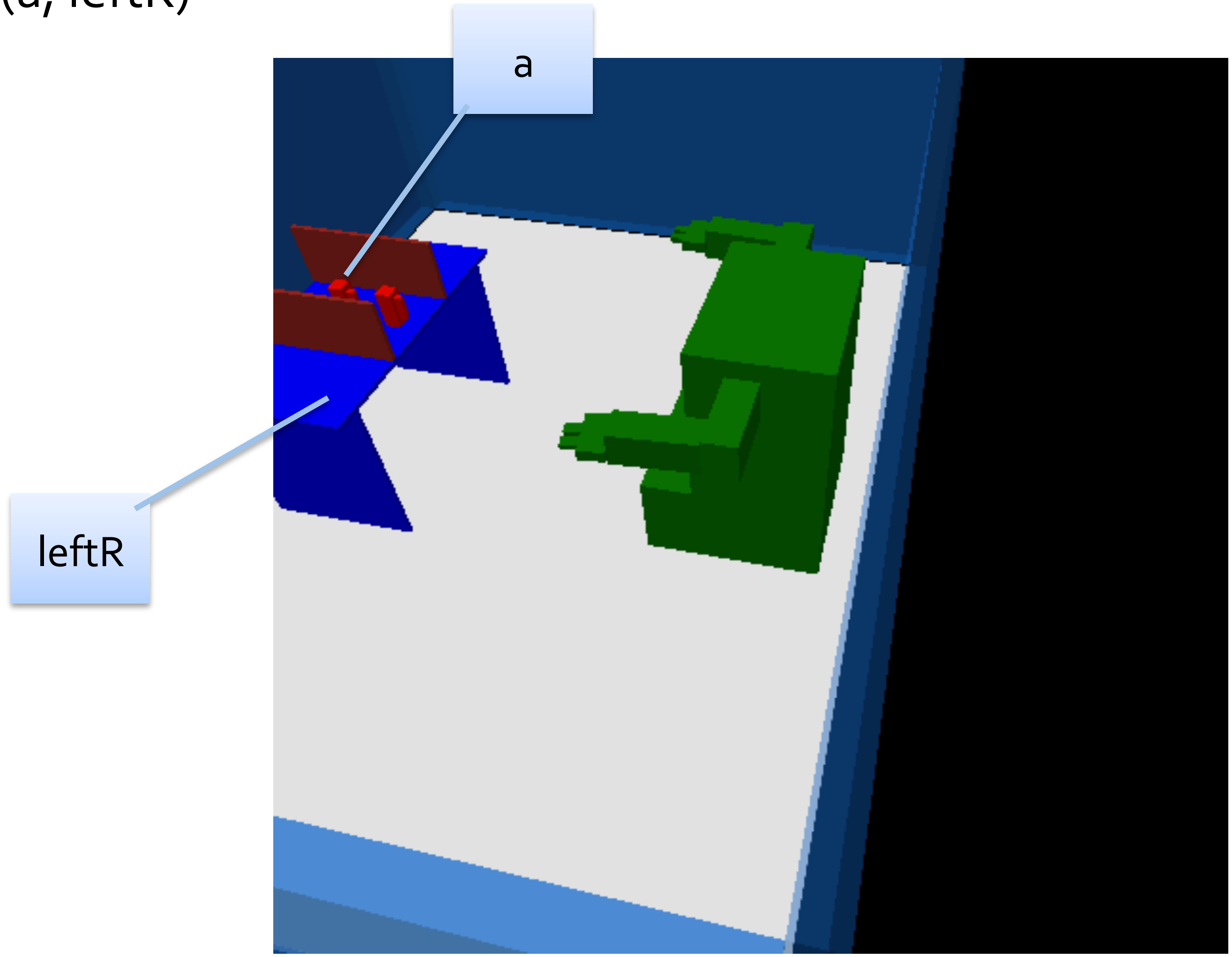
- Accurate robot model
- No reasoning about objects
- Grasps from generator





Regression and generation

Goal: In(a, leftR)

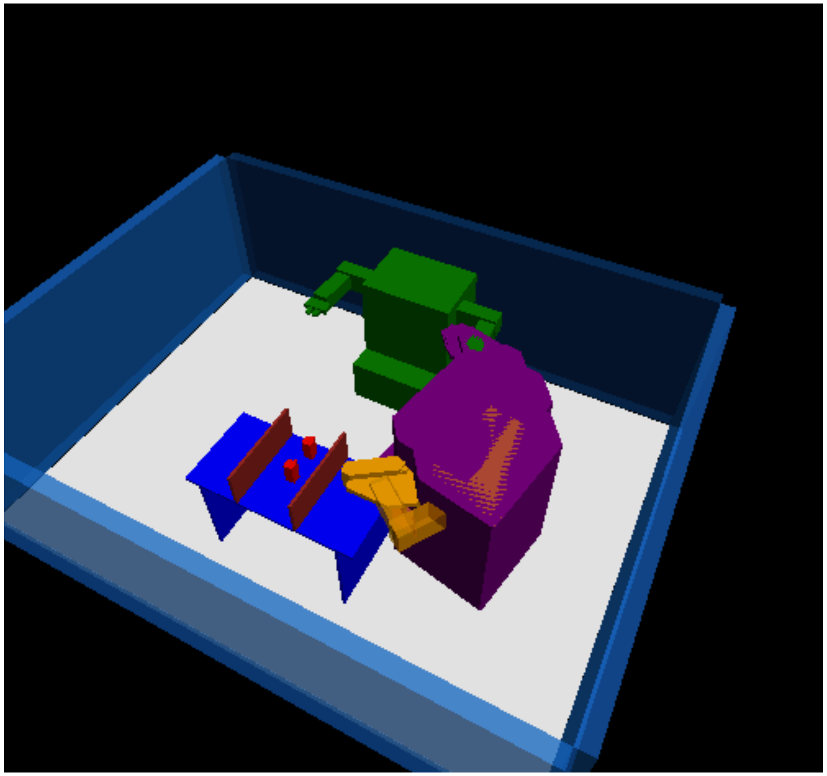


Regression and generation

Goal: In(a, leftR)

O: Place(a, leftR)

G: **Holding(a)**, ClearX(Swept(PlacePath(a, leftR)))



Regression and generation

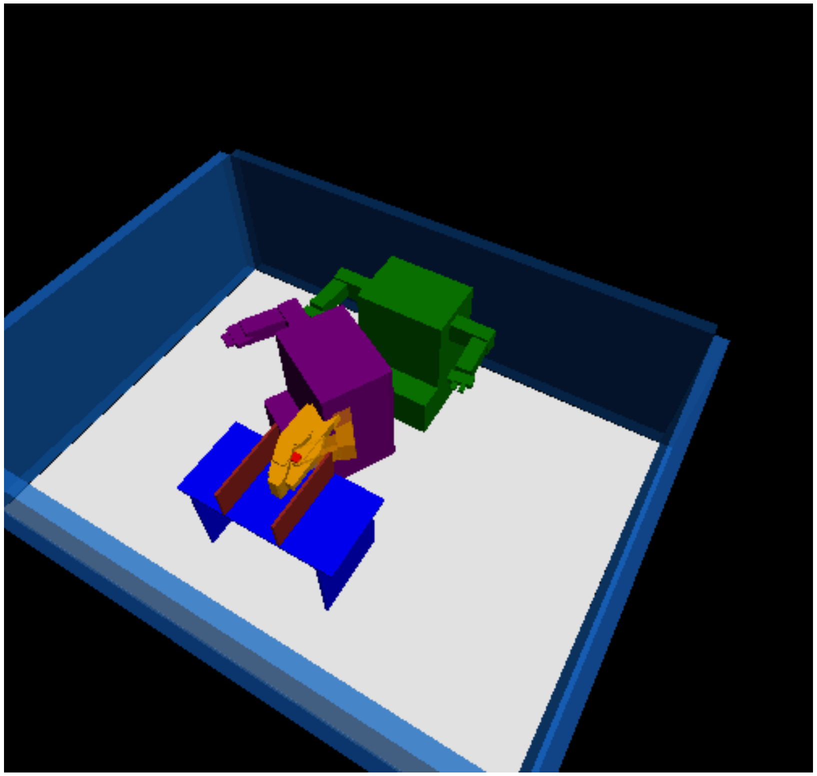
Goal: In(a, leftR)

O: Place(a, leftR)

G: **Holding(a)**, ClearX(Swept(PlacePath(a, leftR)))

O: Pick(a)

G: **ClearX(Swept(PickPath(a)),**
ClearX(Swept(PlacePath(a, leftR))



Regression and generation

Goal: In(a, leftR)

O: Place(a, leftR)

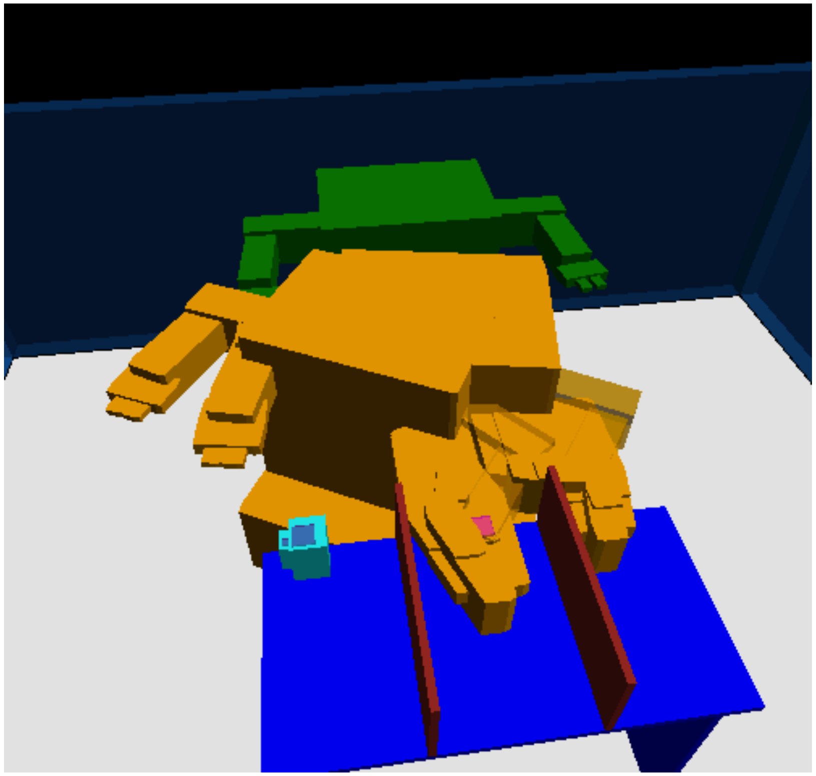
G: **Holding(a)**, ClearX(Swept(PlacePath(a, leftR)))

O: Pick(a)

G: **ClearX(Swept(PickPath(a))**,
ClearX(Swept(PlacePath(a, leftR))

O: Remove(b, Swept(PickPath(a))

G: **In(b, Parking(b))**,
ClearX(Swept(PlacePath(a, leftR))



Regression and generation

Goal: In(a, leftR)

O: **Place(a, leftR)**

G: Holding(a), ClearX(Swept(PlacePath(a, leftR)))

O: **Pick(a)**

G: ClearX(Swept(PickPath(a)), ClearX(Swept(PlacePath(a, leftR))

O: Remove(b, Swept(PickPath(a))

G: In(b, Parking(b)), ClearX(Swept(PlacePath(a, leftR))

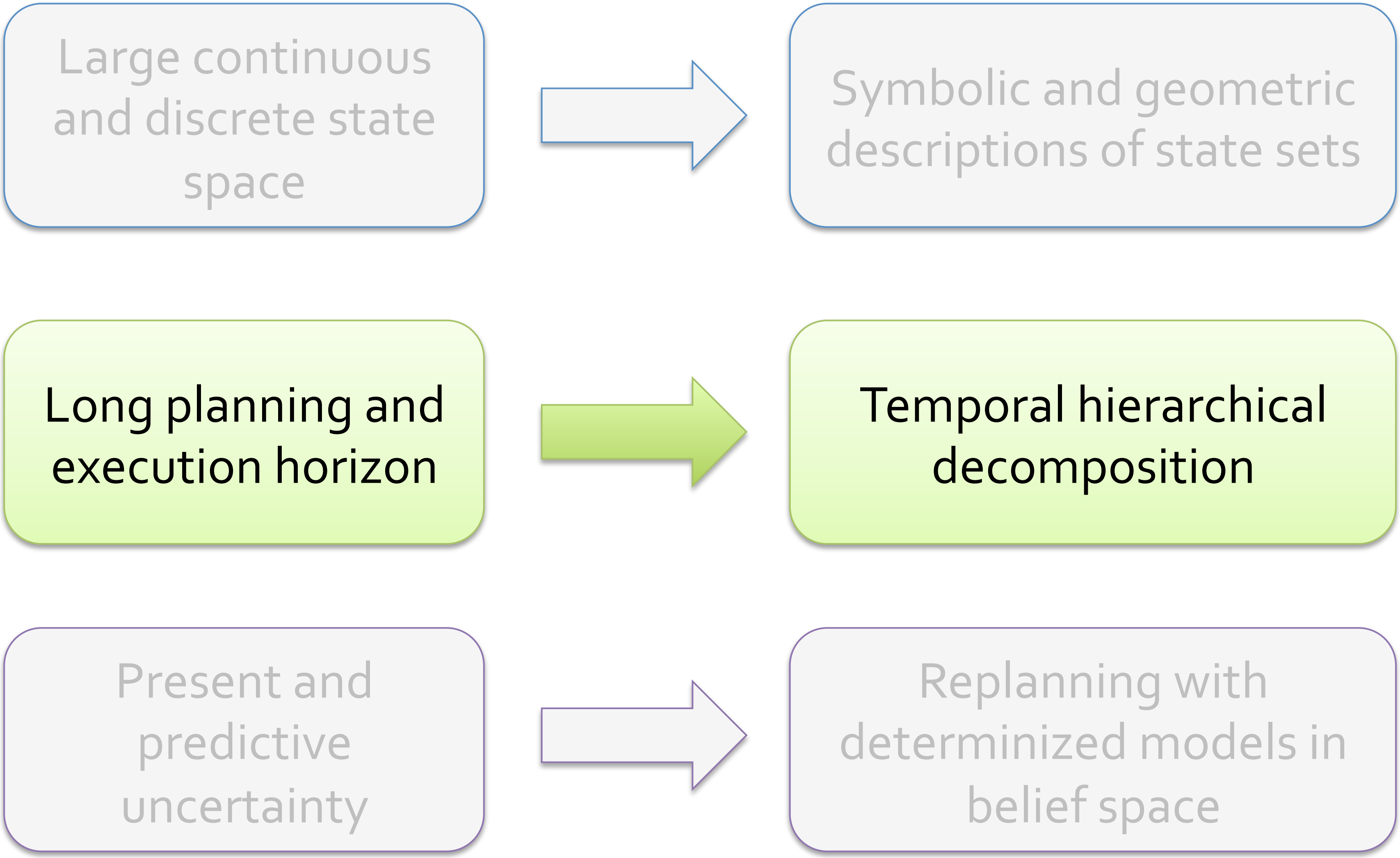
O: **Place(b, Parking(b))**

G: Holding(b), ClearX(Sw(PP(b, Park(b)))), ClearX(Sw(PlaceP(a, leftR))

O: **Pick(b)**

G: ClearX(Sw(PickP(b))), ClearX(Sw(PP(b, Park(b)))),
ClearX(Sw(PlaceP(a, leftR))

Addressing real world challenges

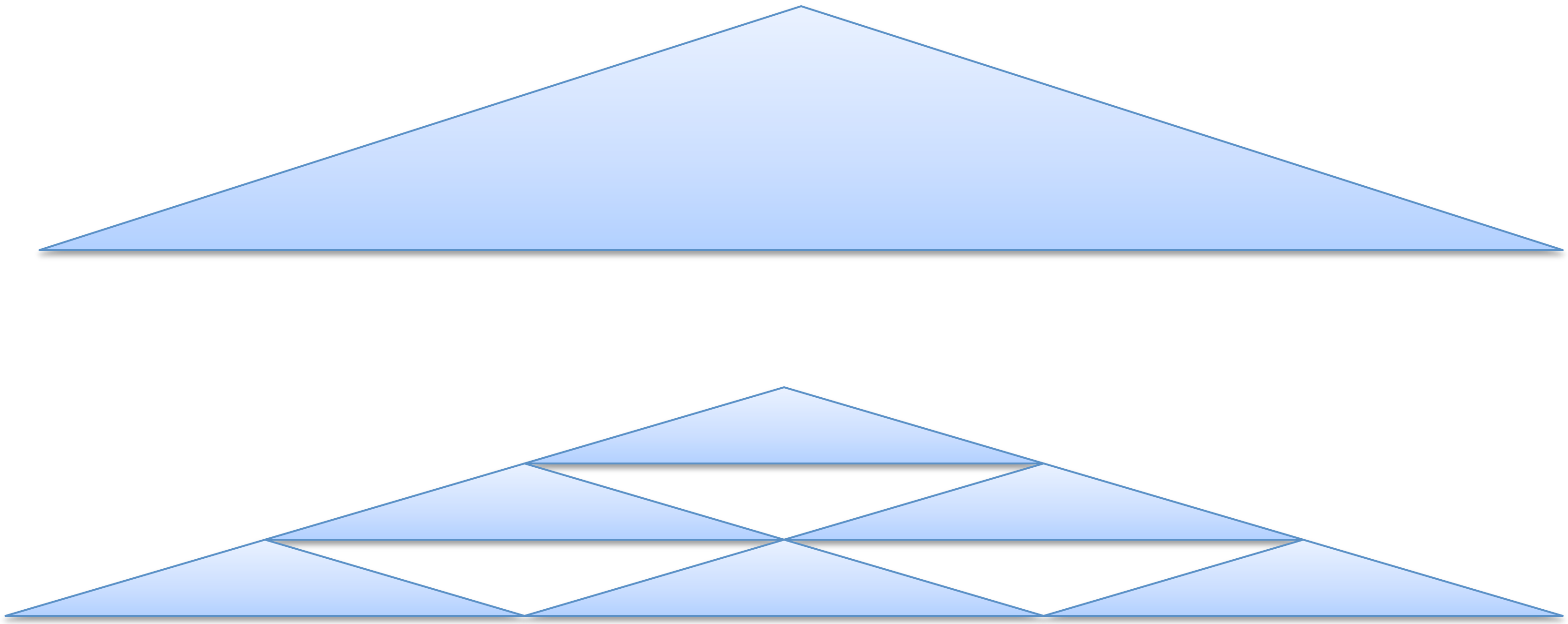


Large continuous and discrete state space → Symbolic and geometric descriptions of state sets

Long planning and execution horizon → Temporal hierarchical decomposition

Present and predictive uncertainty → Replanning with determinized models in belief space

Hierarchy crucial for large problems

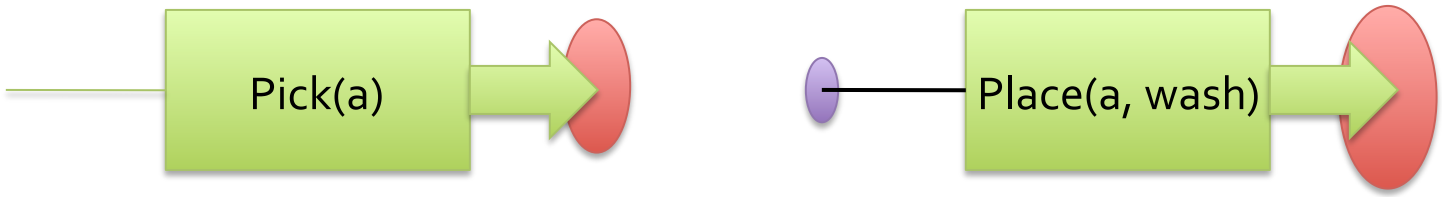


Reduce effective time horizon for planning

Postponing preconditions creates hierarchy

```
Pick(O):
pre: …
res: Holding(O)
```

```
Place(O, R):
pre: 0.Holding(O)
     1.InRobot(room(R))
res: In(O, R)
```

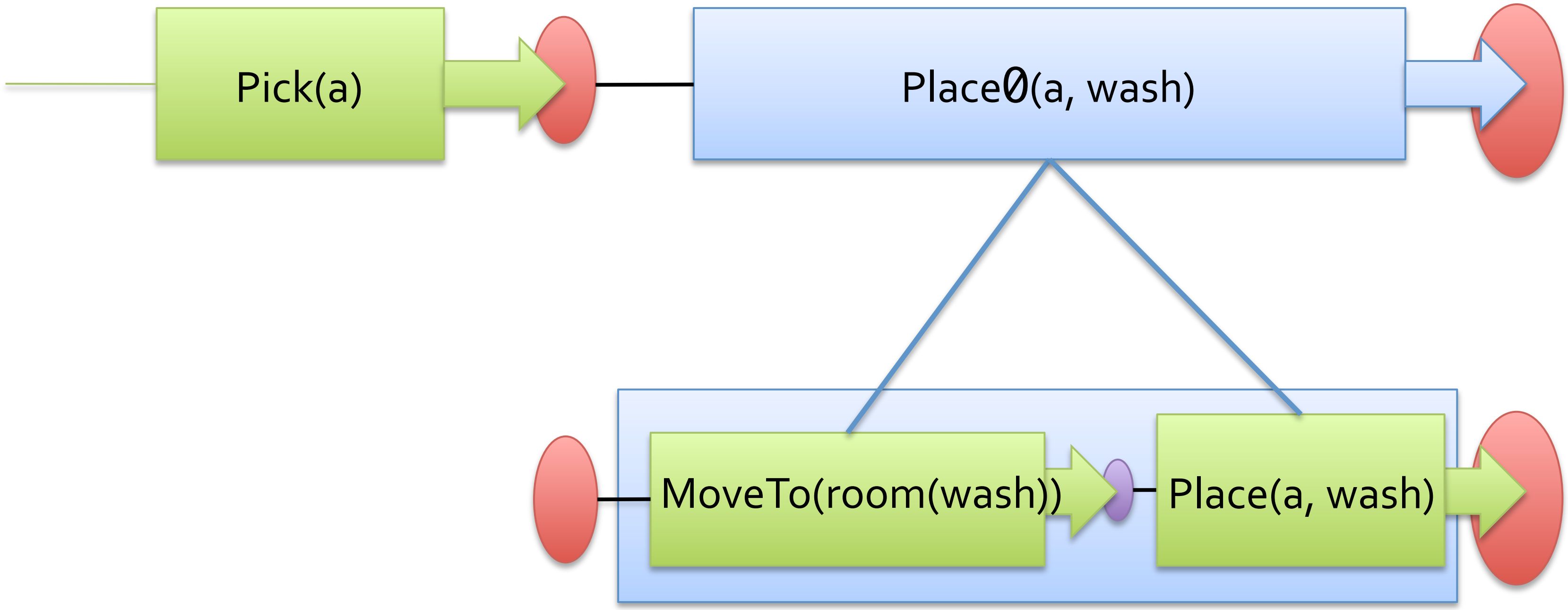


Sacerdoti

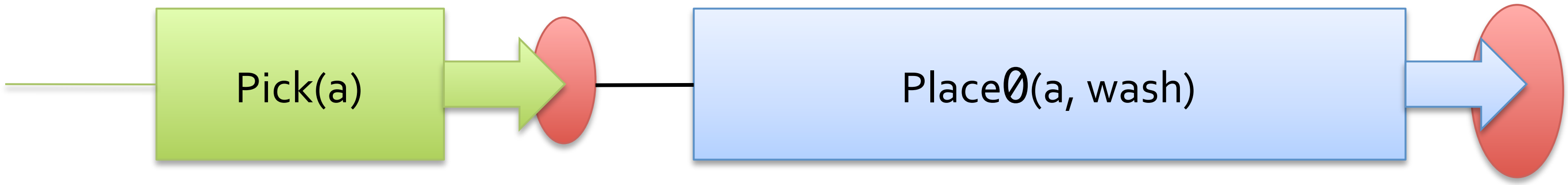
Hierarchical plan

```
Pick(O):
pre: …
res: Holding(O)
```

```
Place0(O, R):
pre: Holding(O)
res: In(O, R)
```



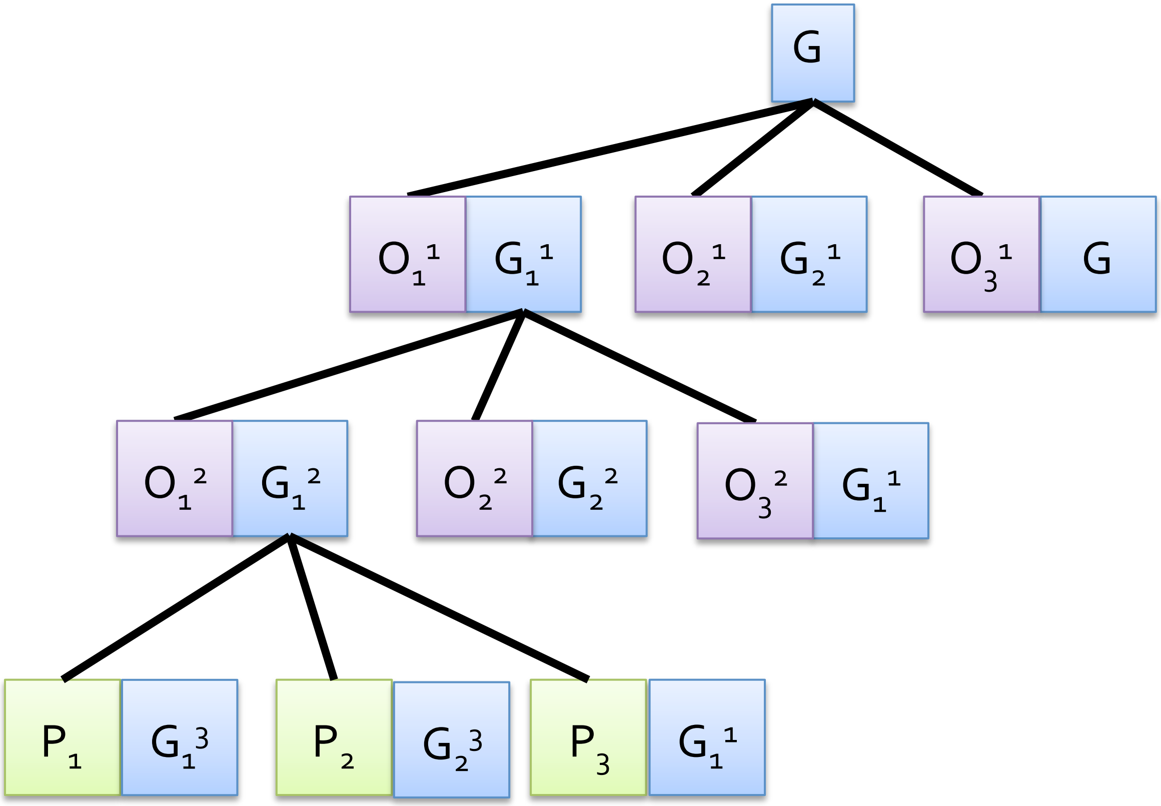
In the now



- Don’t consider all the ways pick(a) could terminate: grasp of object, location of robot, ….
- When it is time to plan for Place0(a, wash), actual world state is start state

Planning in the now

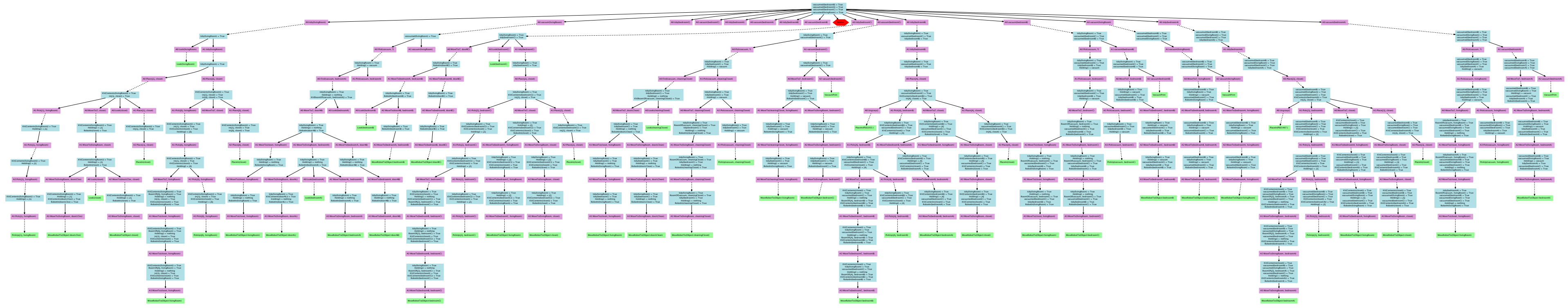
- maintain left expansion of plan tree
- each level uses a higher-fidelity model
- keep track of pre-image for each operation
- recursively plan to achieve those preconditions
- execute primitives



Cleaning house

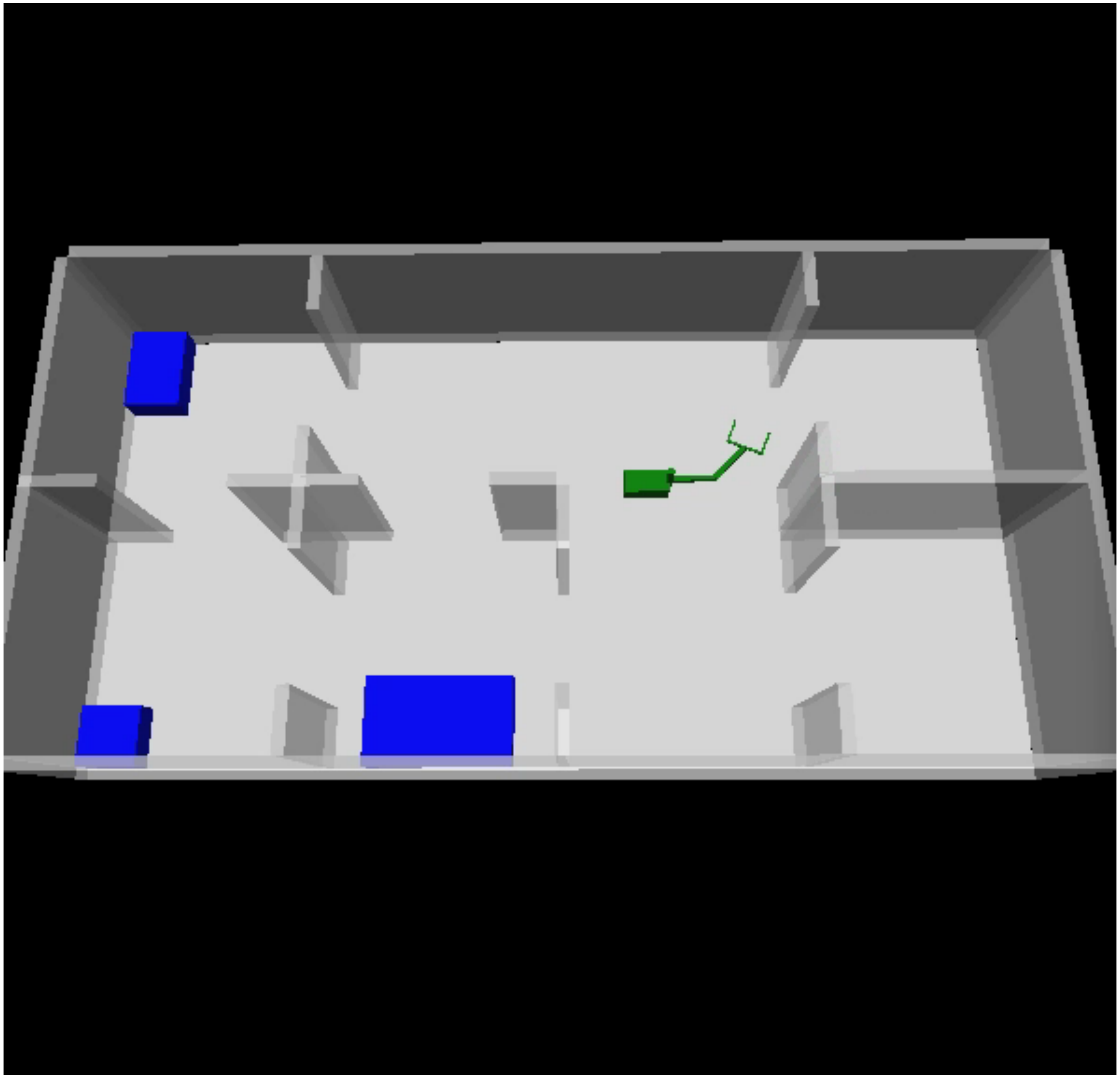
Goal: mop four of the rooms in the house

- have to vacuum before mopping
- have to put away junk items before vacuuming

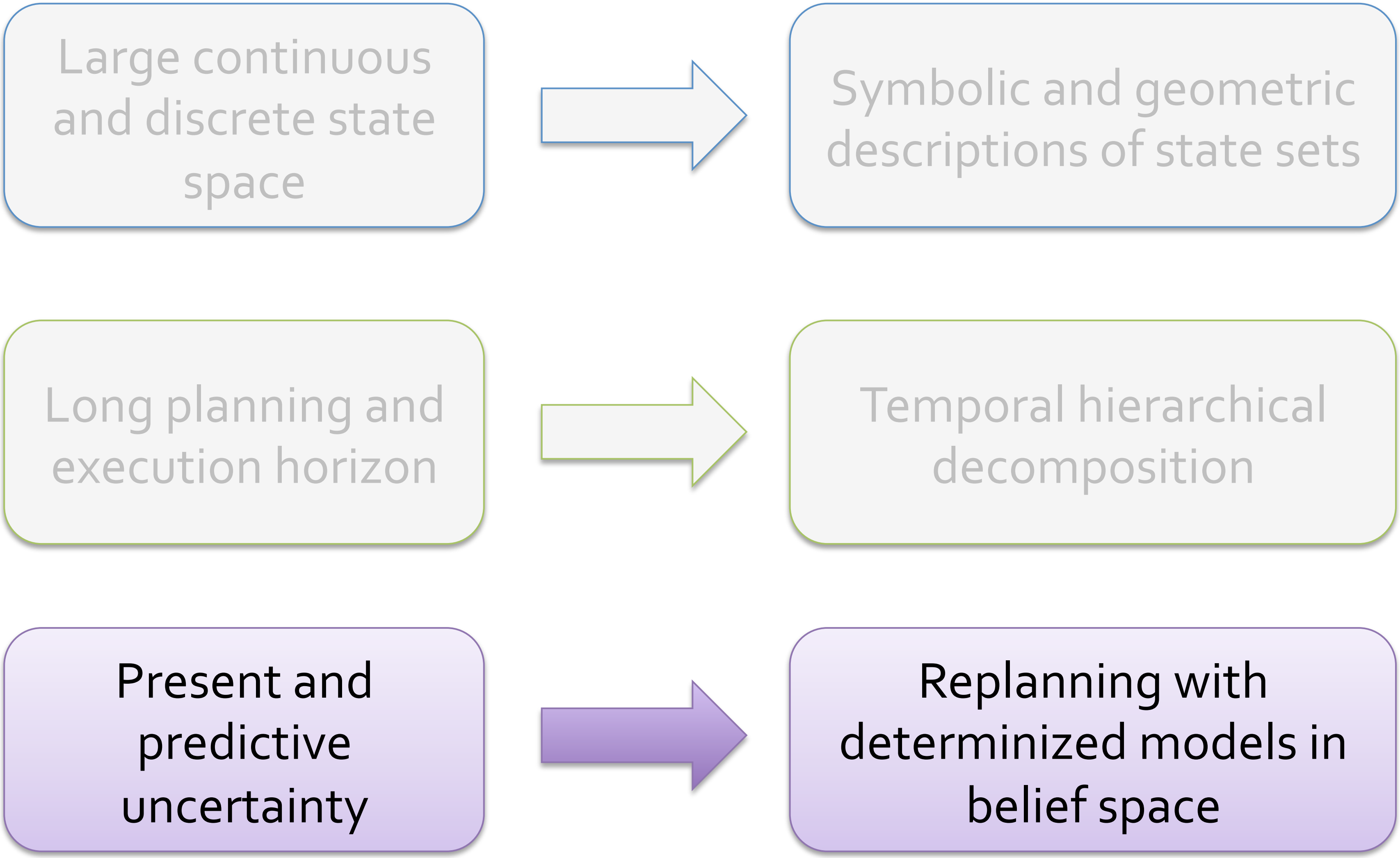


Trace of interleaved planning and execution

Cleaning house



Addressing real world challenges



Large continuous and discrete state space

Symbolic and geometric descriptions of state sets

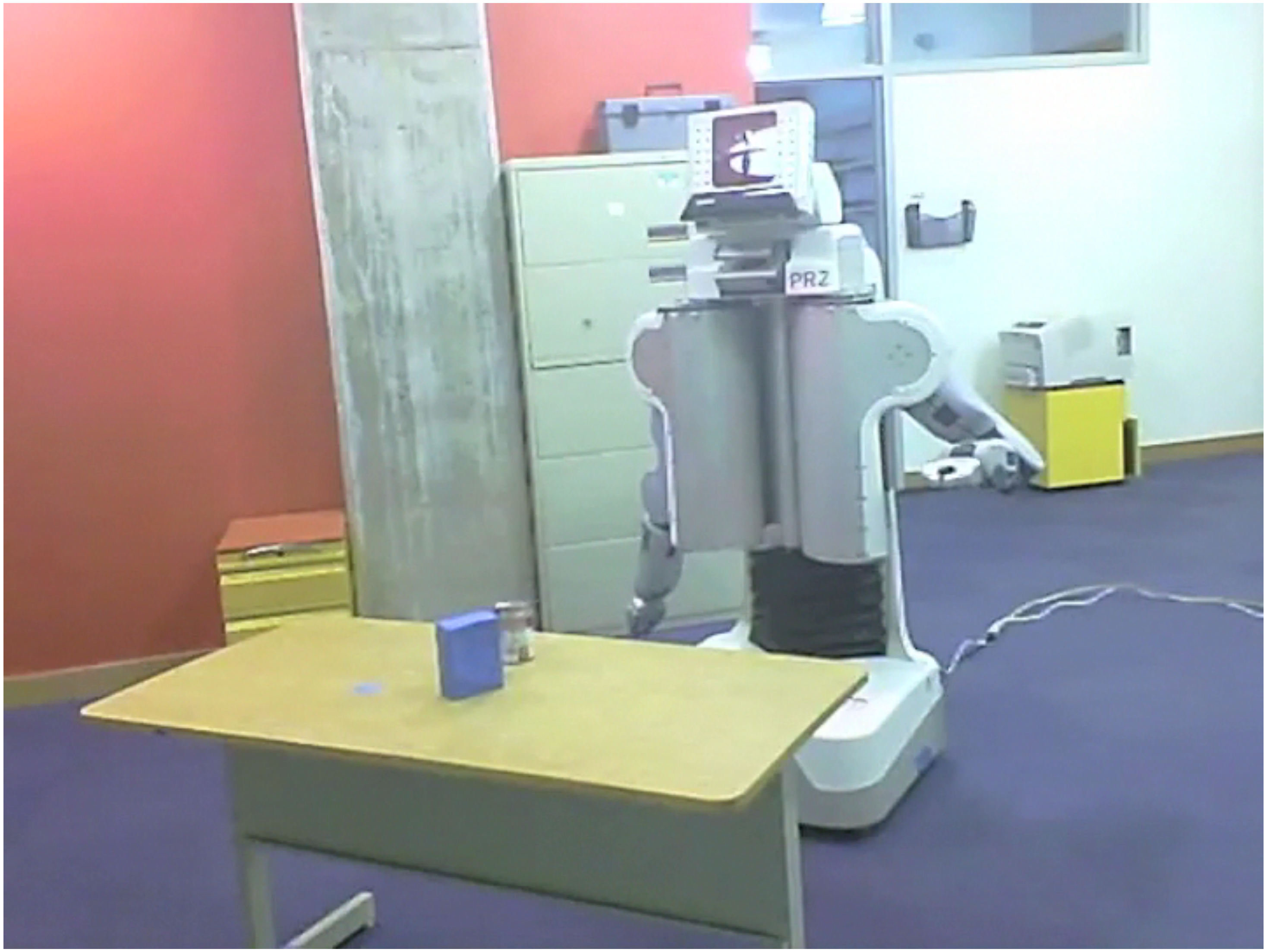
Long planning and execution horizon

Temporal hierarchical decomposition

Present and predictive uncertainty

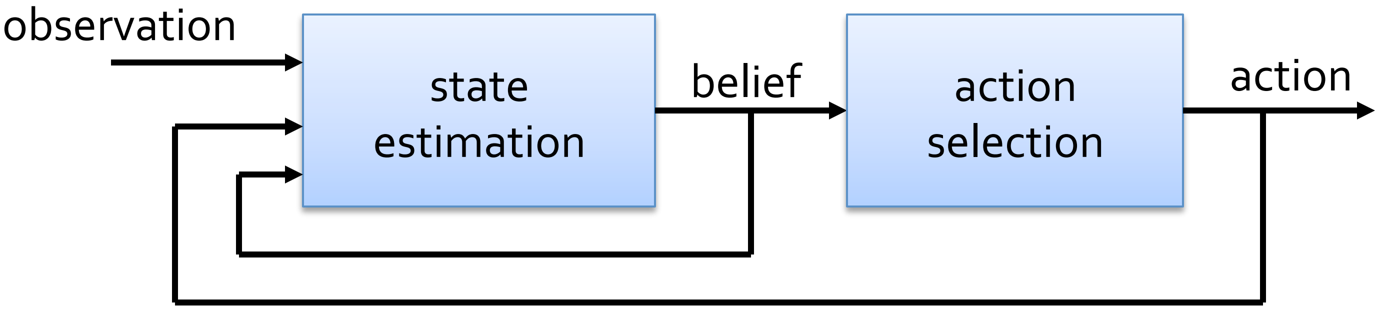
Replanning with determinized models in belief space

Pay attention to M&M paying attention



Pay attention to M&M paying attention

Action selection with partial observability



Plan in belief space:

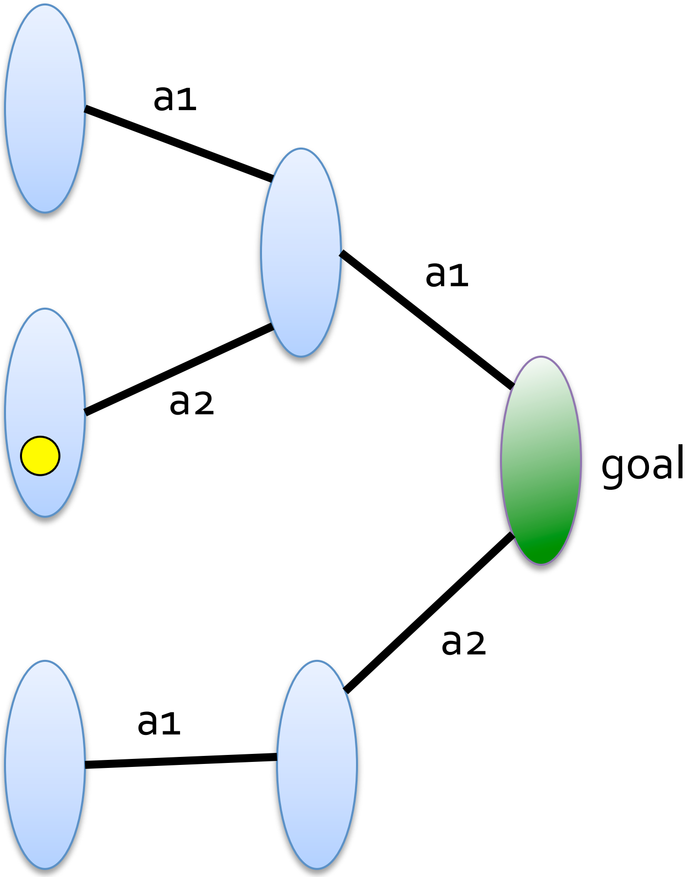
- every action gains information and changes the world
- changes are reflected in new belief via estimation
- goal is to believe that the environment is in a desired state

Pre-image back-chaining in belief space

Weakest precondition of goal set under each action sequence

Test whether start belief is in a pre-image

Represent goal and pre-images as conjunctions of logical expressions



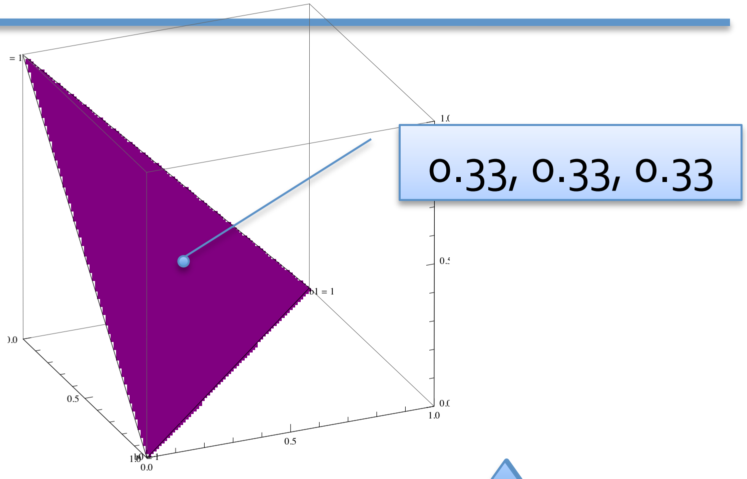
Belief-space pre-image of goal set G under action a:

$$\mathrm{R}(\mathrm{G}, \mathrm{a}) = \{\mathrm{b} \mid \mathrm{SE}(\mathrm{b}, \mathrm{a}, \mathrm{o}^*(\mathrm{b}, \mathrm{a})) \in \mathrm{G}\}$$

Fluents describe **sets** of belief states

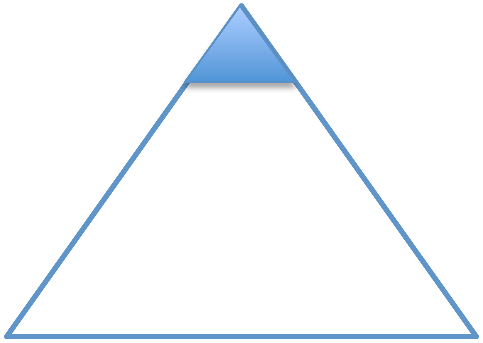
Simple example:

- three possible states
- belief space is 3-simplex

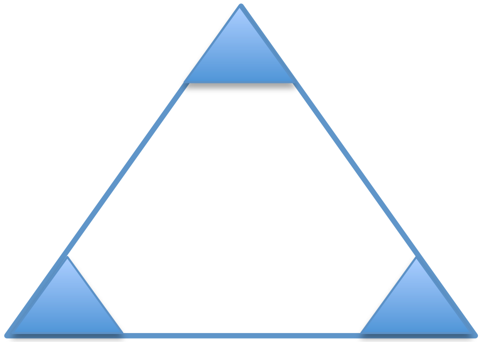


0.33, 0.33, 0.33

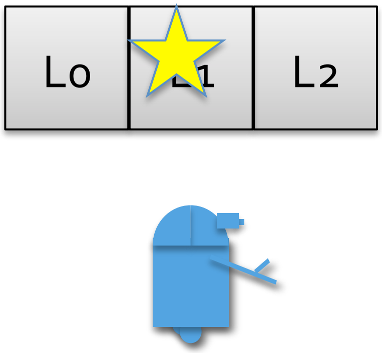
$$\mathrm{KLoc}(\mathrm{o}, \mathrm{l}, \mathrm{p}) \equiv \mathrm{Pr}(\mathrm{Loc}(\mathrm{o}) = \mathrm{l}) > \mathrm{p}$$



$$\mathrm{KVLoc}(\mathrm{o}, \mathrm{p}) \equiv \exists \mathrm{l}.\, \mathrm{Pr}(\mathrm{Loc}(\mathrm{o}) = \mathrm{l}) > \mathrm{p}$$



Tiny domain



State: object is in L0, L1, or L2

Actions:

- **Look** in L0, L1, or L2
  - FalsePos = 0.1, FalseNeg = 0.2
- **Move obj** from Start to Goal
  - Fails w.p. 1.0 if obj not in Start
  - Fails w.p. 0.2 otherwise

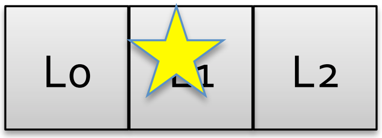
Observations:

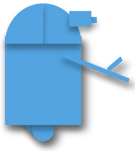
- After look action, **see** the object, or **not see** it

Goal:

- **Believe**, with probability > 0.95, that object is in L0

Planning operators in belief space





```
Move(0, TargetLoc):
   result: KLoc(0, TargetLoc, Prob)
   exists: StartLoc
   pre: KLoc(0, StartLoc, Prob / (1-MoveFailProb))
   cost: -log(Prob / (1-MoveFailProb))

Look(Loc):
   result: KLoc(0, Loc, Prob)
   pre: Kloc(0, Loc, lookRegres
   cost: -log(lookRegressPro

lookRegressProb(p) = falsePos
                (falsePos * p + (1  alseNeg) * (1-p))
```

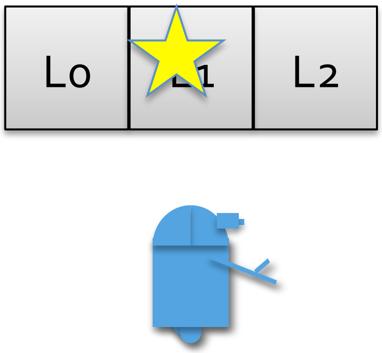
Same operators for any number of objects and locations!

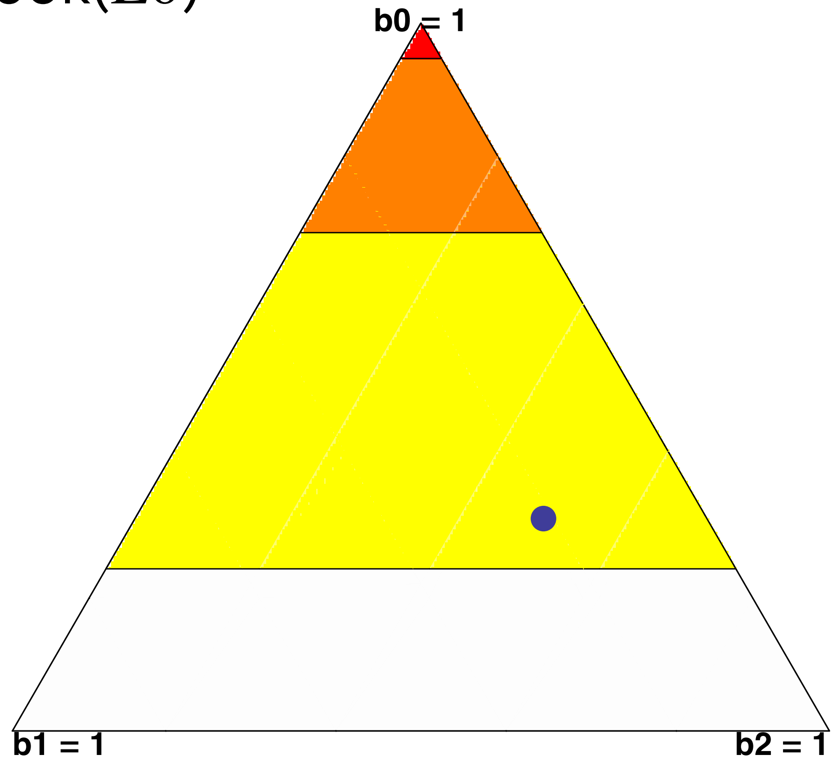
Planning, replanning, and execution

Goal: $\text{KLoc}(o, \text{L}0, 0.95)$

Belief: (0.2, 0.3, 0.5)

Plan: Look(L0), Look(L0)





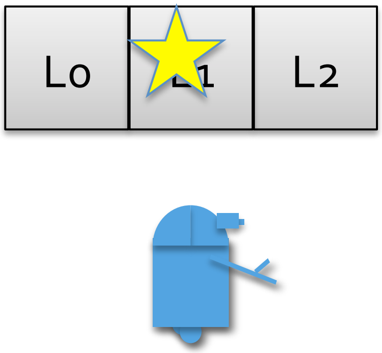
Planning, replanning, and execution

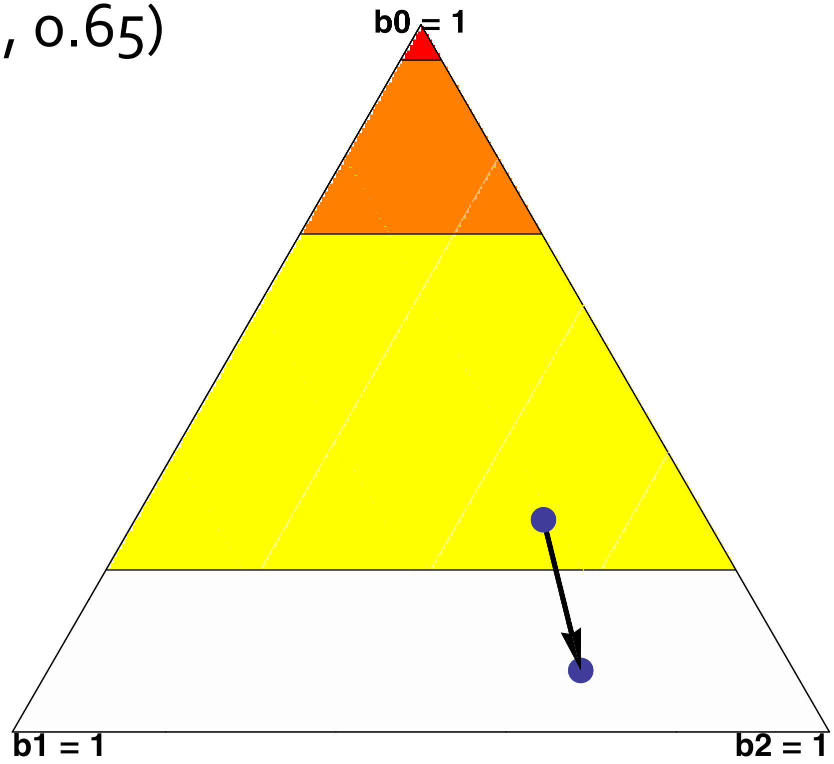
Goal: $\mathrm{KLoc}(\mathrm{o}, \mathrm{L0}, 0.95)$

Action: Look($\mathrm{L0}$)

Observation: No see

Belief: (0.09, 0.26, 0.65)



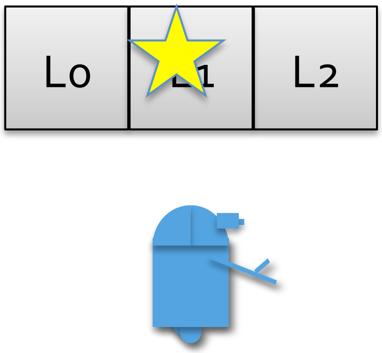


Planning, replanning, and execution

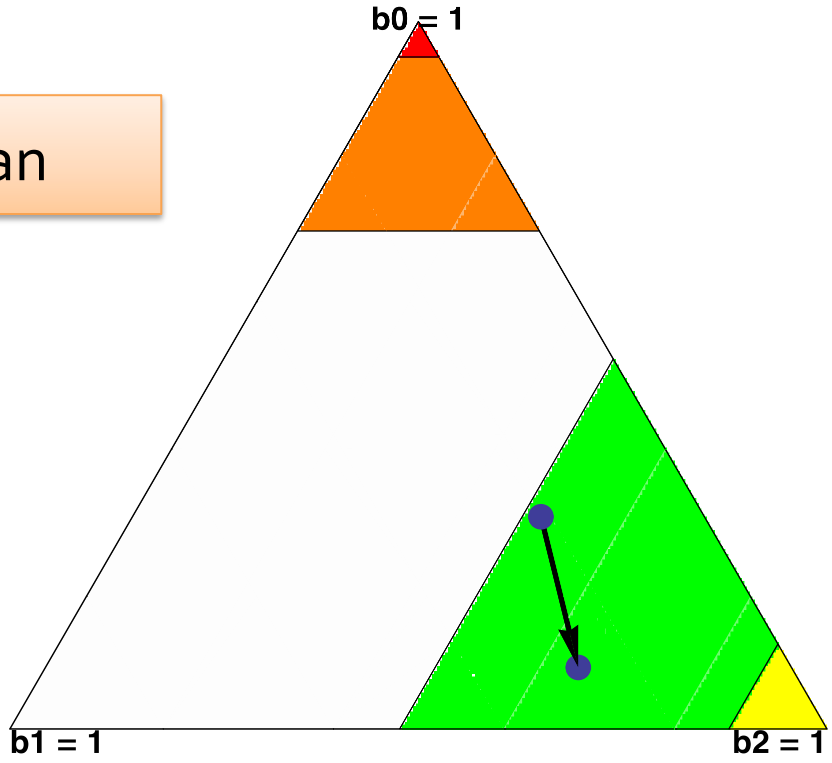
Goal: $\mathrm{KLoc}(o, L0, 0.95)$

Plan: Look($L2$), Move($L2$, $L0$), Look($L0$)

Belief: (0.09, 0.26, 0.65)



Replan



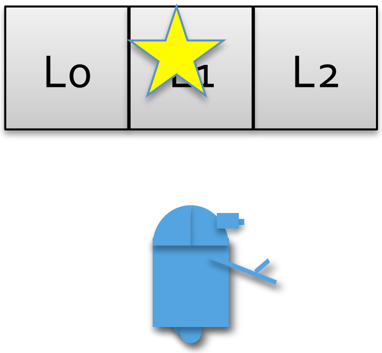
Planning, replanning, and execution

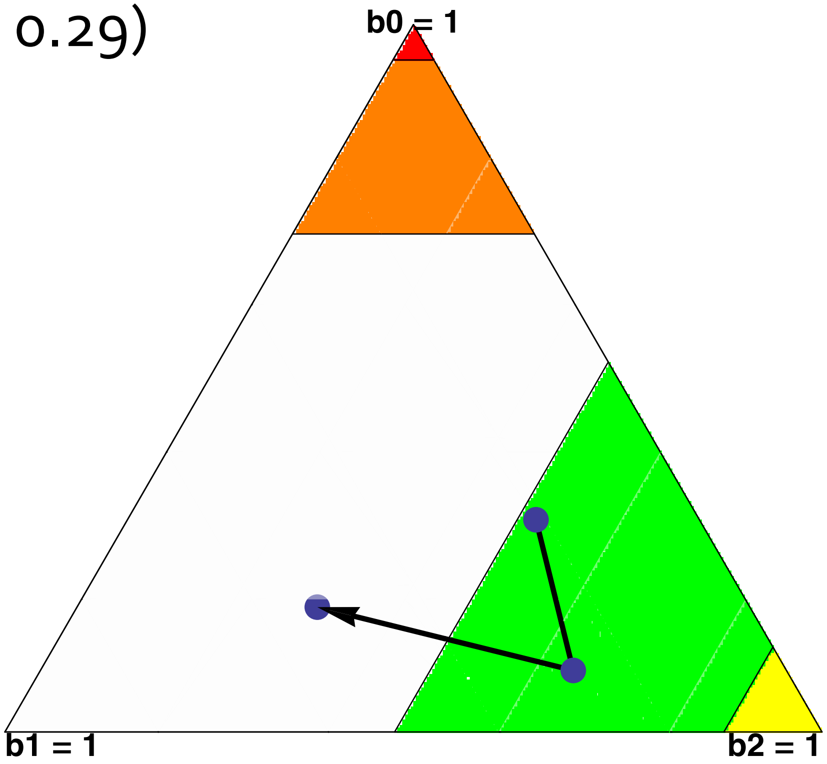
Goal: $\mathrm{KLoc}(o, \mathrm{L}0, 0.95)$

Action: Look($\mathrm{L}2$)

Observation: No see

Belief: (0.18, 0.53, 0.29)



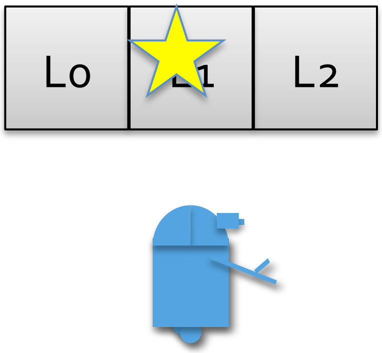


Planning, replanning, and execution

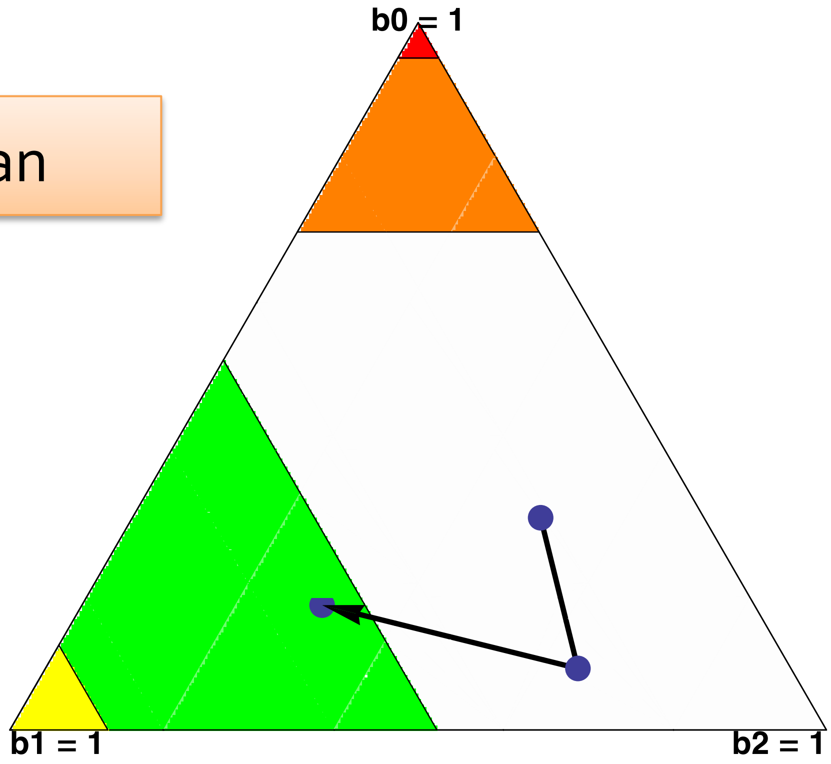
Goal: $\mathrm{KLoc}(o, L0, 0.95)$

Plan: Look($L1$), Move($L1$, $L0$), Look($L0$)

Belief: (0.18, 0.53, 0.29)



Replan



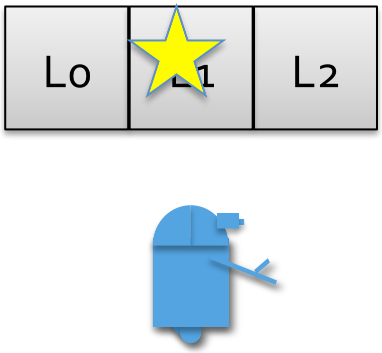
Planning, replanning, and execution

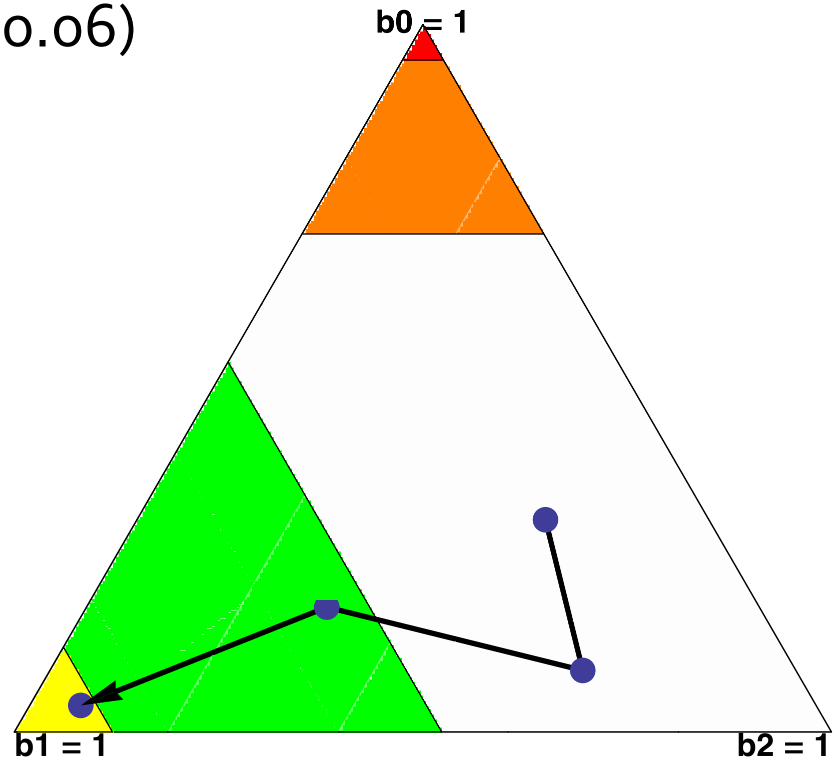
Goal: $\mathrm{KLoc}(o, L0, 0.95)$

Action: Look($L1$)

Observation: See object

Belief: (0.04, 0.9, 0.06)





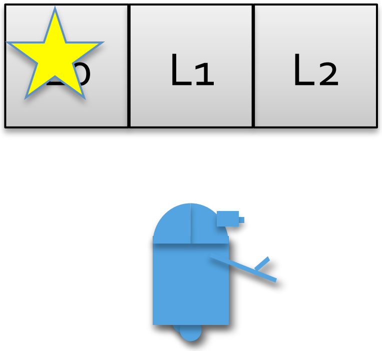
Planning, replanning, and execution

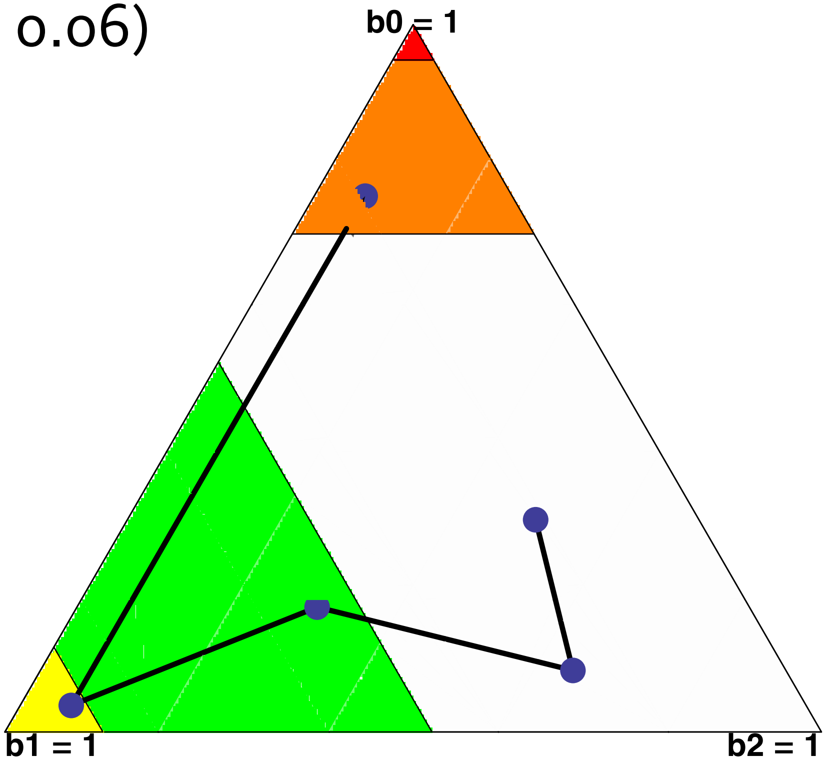
Goal: $\mathrm{KLoc}(o, L0, 0.95)$

Action: Move$(L1, L0)$

Observation: --

Belief: (0.76, 0.18, 0.06)





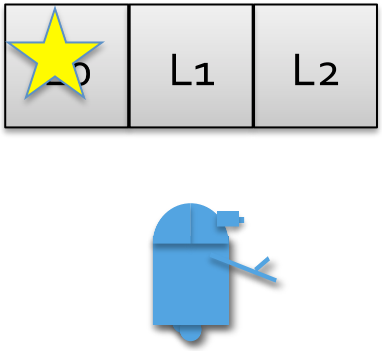
Planning, replanning, and execution

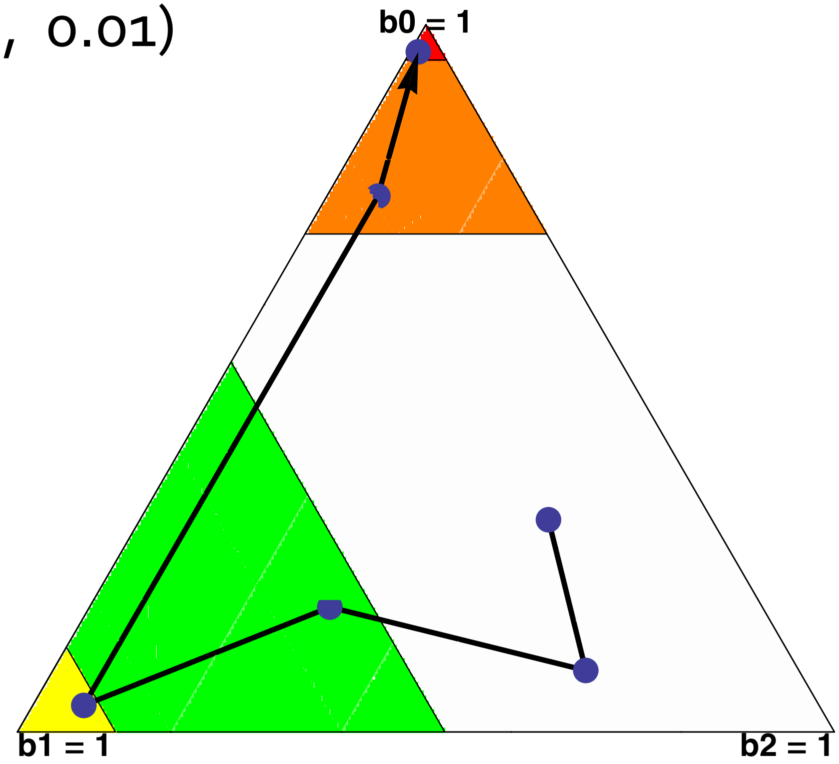
Goal: $\mathrm{KLoc}(o, \mathrm{L0}, 0.95)$

Action: Look($\mathrm{L0}$)

Observation: see object

Belief: (0.96, 0.03, 0.01)





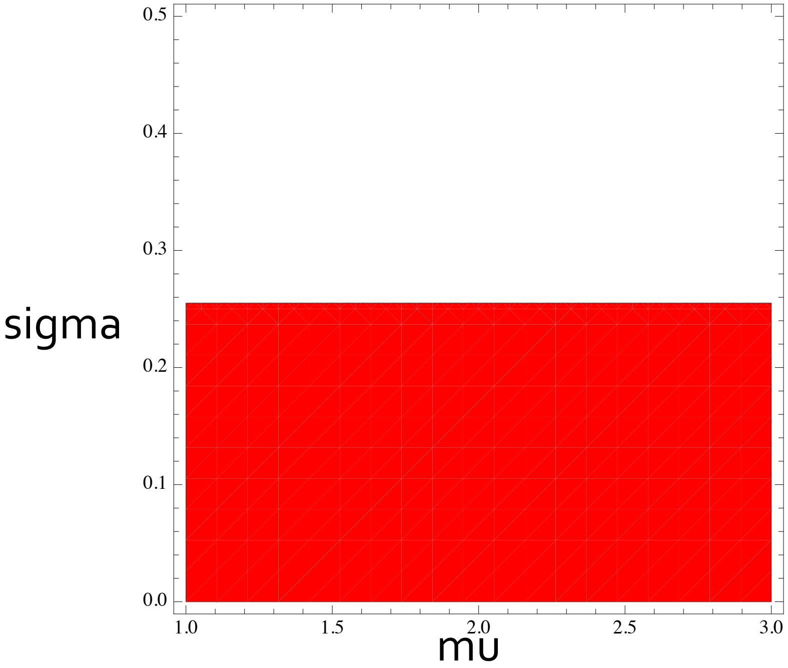
Fluents for beliefs on continuous spaces

$$\mathrm{BVLoc}(O, \epsilon, \delta) \equiv \Pr(|\mathrm{Loc}(O) - \mu(\mathrm{Loc}(O))| > \delta) \leq \epsilon$$

For a one-D Gaussian, probability near mean:

$$\mathrm{PNM}(X, \delta) = \Phi\left(\frac{\delta}{\sigma}\right) - \Phi\left(-\frac{\delta}{\sigma}\right) = \mathrm{erf}\left(\frac{\delta}{\sqrt{2}\sigma}\right)$$

Set of distributions:
PNM(X, 0.05) > 0.95

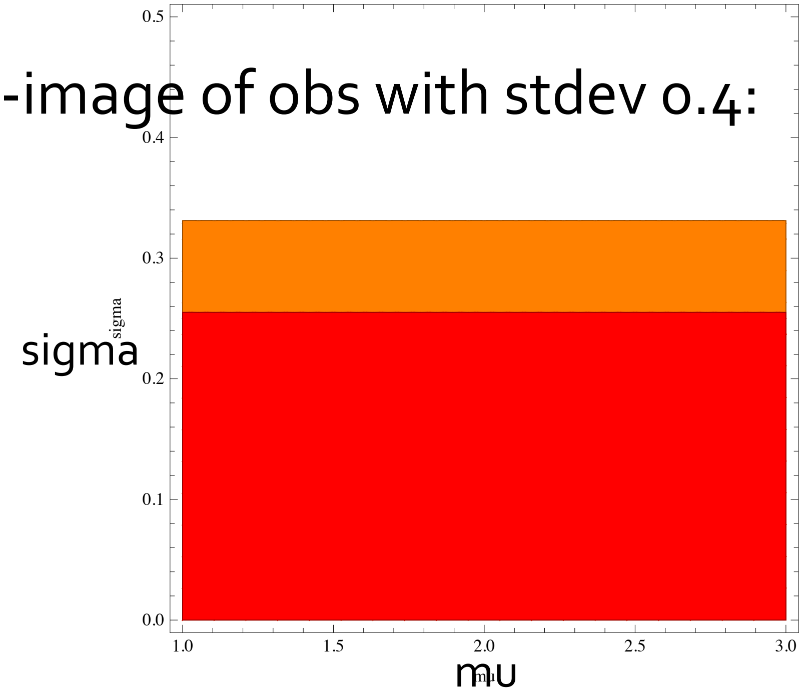


PNM Regression after observation

To guarantee $\mathrm{PNM}(X, \delta) > 1 - \epsilon_{t+1}$ after **observation** action, require $\mathrm{PNM}(X, \delta) > 1 - \epsilon_t$ before observation

$$\epsilon_t = 1 - \mathrm{erf}\left(\sqrt{\mathrm{erf}^{-1}(1 - \epsilon_{t+1})^2 - \frac{\delta^2}{2\sigma_o^2}}\right)$$

Set of distributions in pre-image of obs with stdev 0.4:

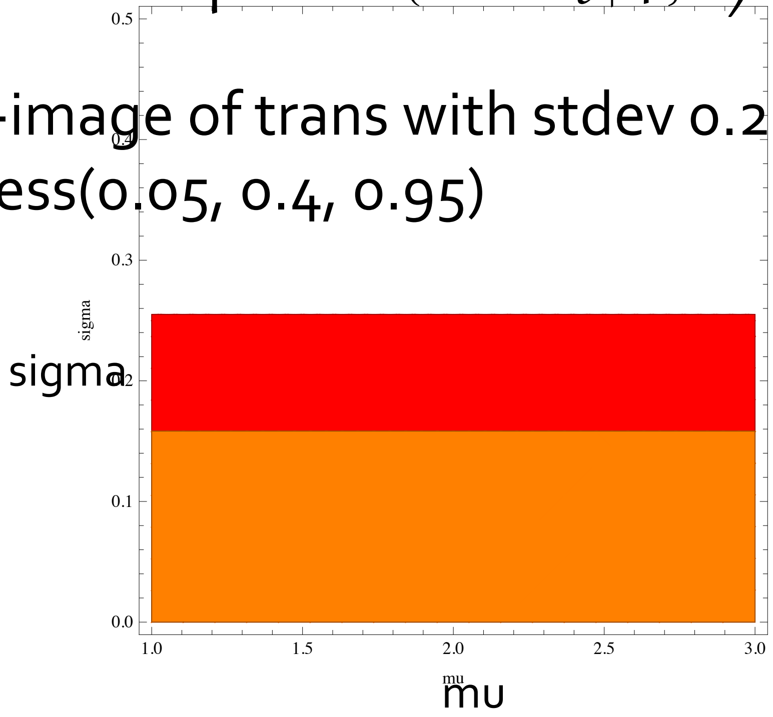


PNM Regression after transition

To guarantee $\mathrm{PNM}(X, \delta) > 1 - \epsilon_{t+1}$ after **transition** action, require $\mathrm{PNM}(X, \delta) > 1 - \epsilon_t$ before observation

$$\epsilon_t = 1 - \mathrm{erf}\left(\frac{\delta\ \mathrm{erf}^{-1}(1 - \epsilon_{t+1})}{\sqrt{\delta^2 - 2\sigma_Y^2\ \mathrm{erf}^{-1}(1 - \epsilon_{t+1})^2}}\right)$$

Set of distributions in pre-image of trans with stdev 0.2:
PNM(X, 0.05) > PNMRegress(0.05, 0.4, 0.95)

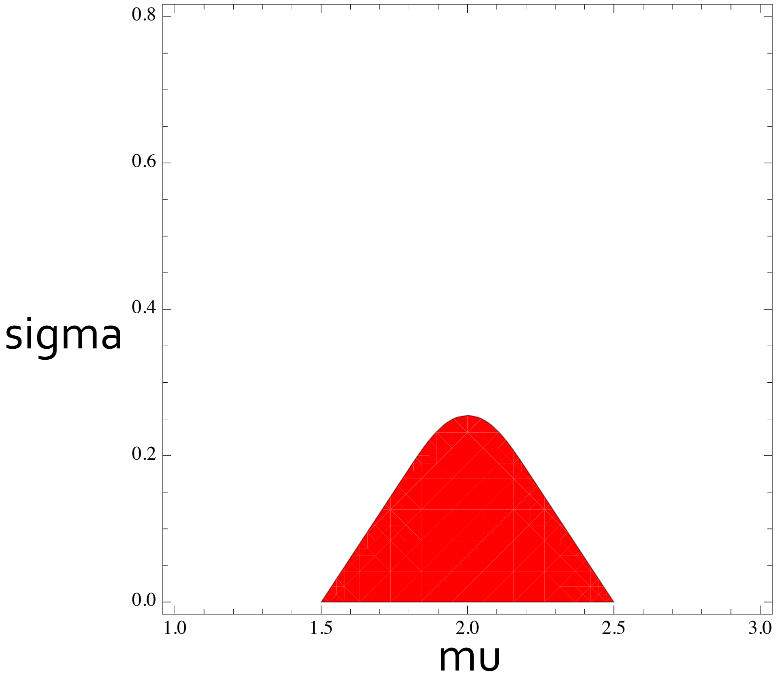


Probability near desired value

$$\mathrm{KLoc}(\mathrm{O}, \mathrm{L}, \epsilon, \delta) \equiv \mathrm{Pr}(|\mathrm{Loc}(\mathrm{O}) - \mathrm{L}|) > \delta) < \epsilon$$

For a one-D Gaussian, probability near value 2:

Set of distributions:
PNV(X, 2, 0.5) > 0.95

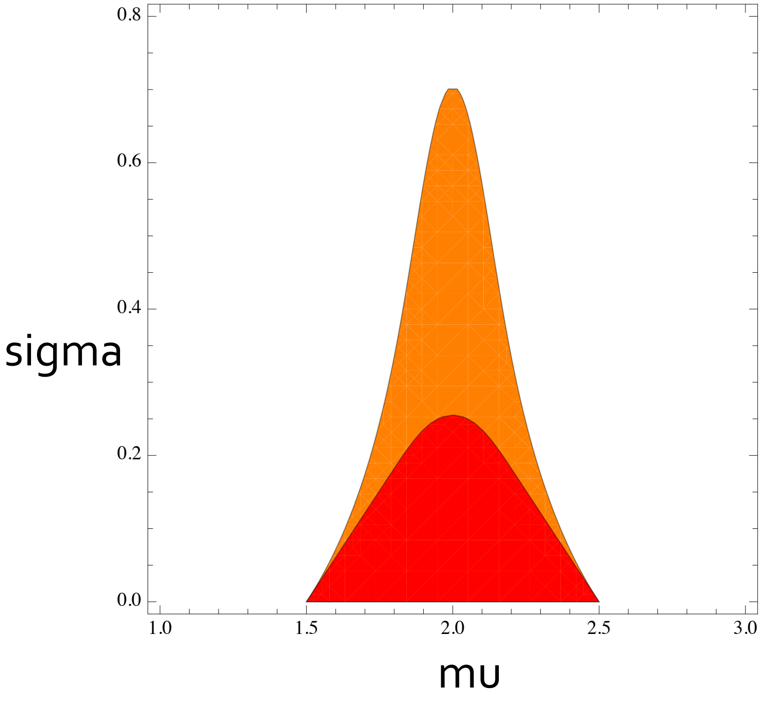


PNV Regression: Observation

$$\mathrm{KLoc}(\mathrm{O}, \mathrm{L}, \epsilon, \delta) \equiv \Pr(|\mathrm{Loc}(\mathrm{O}) - \mathrm{L}|) > \delta) < \epsilon$$

Set of distributions such that:

- after observation with sigma 0.4
- PNV(2, 0.5) > 0.95

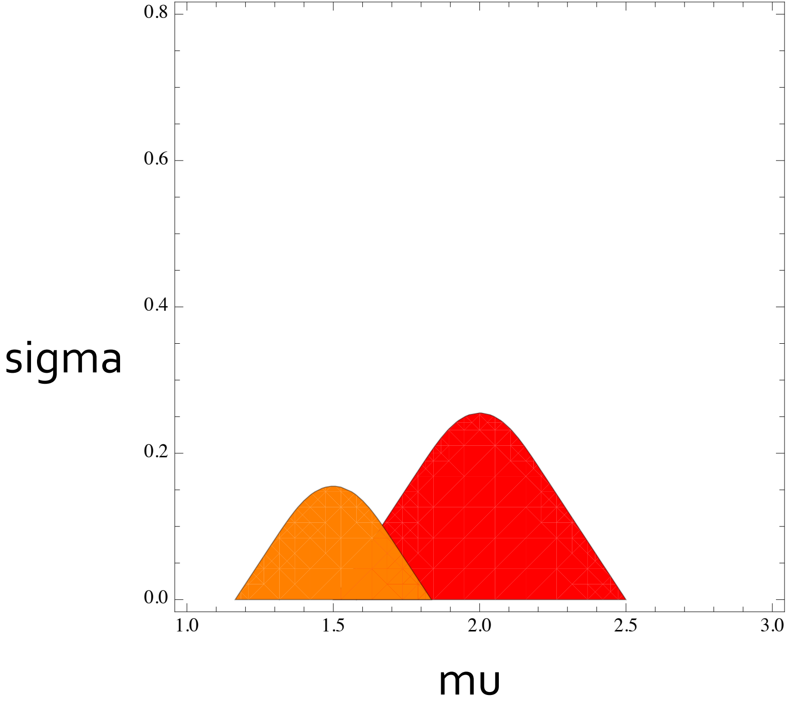


PNV Regression: Transition

$$\mathrm{KLoc}(\mathrm{O}, \mathrm{L}, \epsilon, \delta) \equiv \Pr(|\mathrm{Loc}(\mathrm{O}) - \mathrm{L}|) > \delta) < \epsilon$$

Set of distributions such that:

- after transition with mean 0.5, sigma 0.1
- PNV(2, 0.5) > 0.95

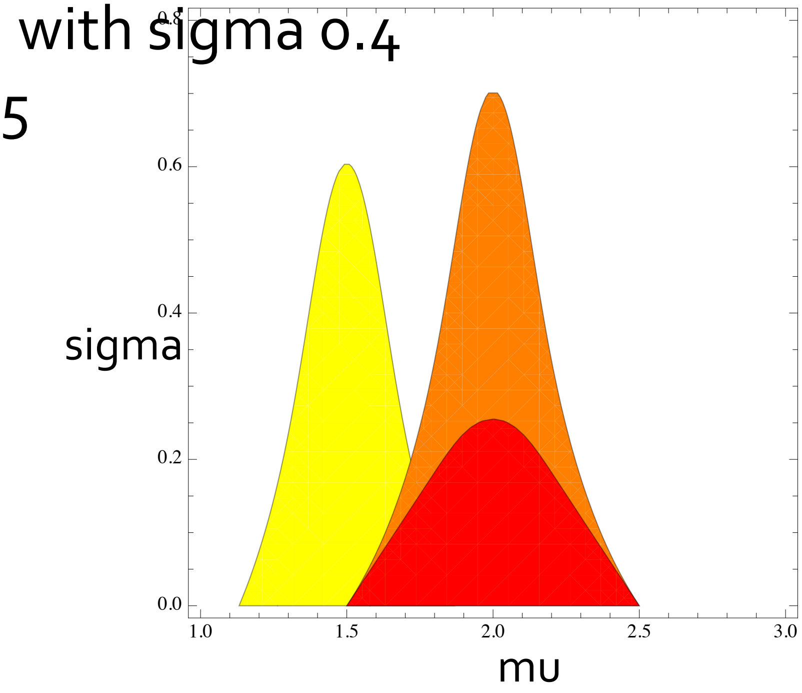


PNV Regression: Transition

$$\mathrm{KLoc}(\mathrm{O}, \mathrm{L}, \epsilon, \delta) \equiv \Pr(|\mathrm{Loc}(\mathrm{O}) - \mathrm{L}|) > \delta) < \epsilon$$

Set of distributions such that:

- after transition with mean 0.5, sigma 0.1
- then observation with sigma 0.4
- PNV(2, 0.5) > 0.95



Planning in belief space for 1D continuous space

Goal:

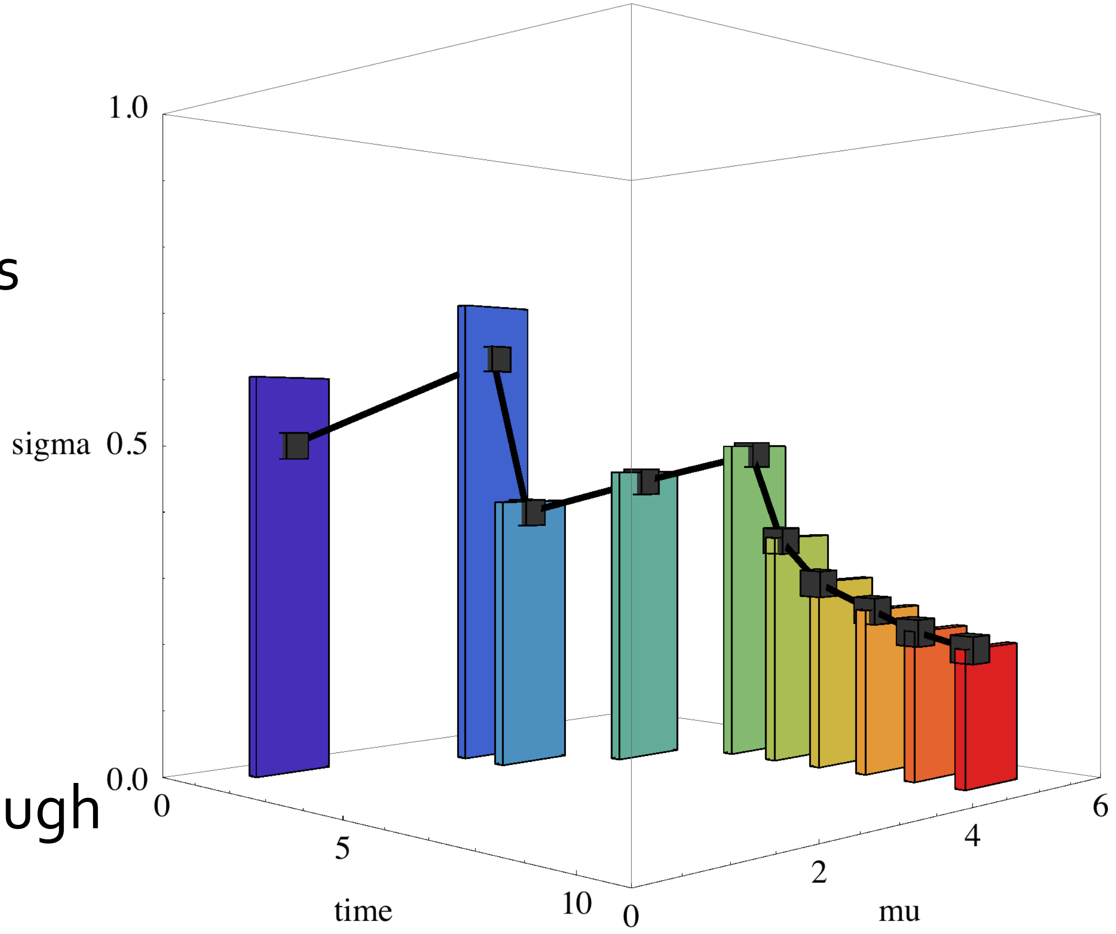
- ModeNear(5, 0.2)
- BV(0.05, 0.4)

Observation fails if stdev is too high

Low transition error

High observation error

Initial plan is followed through belief space



Planning in belief space for 1D continuous space

Goal:

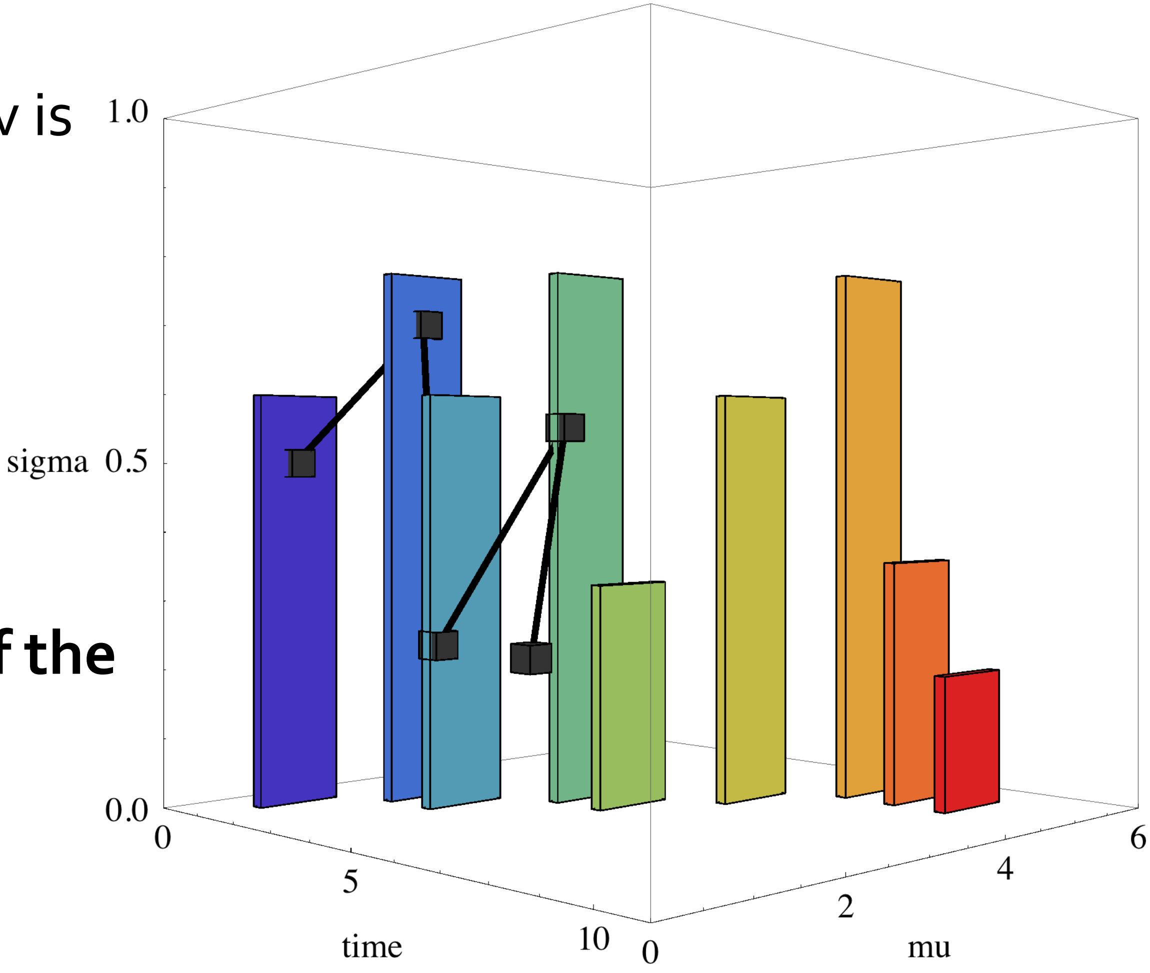
- ModeNear(5, 0.05)
- BV(0.05, 0.4)

Observation fails if stdev is too high

High transition error

Low observation error

After 4 steps, we fall off the plan



Planning in belief space for 1D continuous space

Goal:

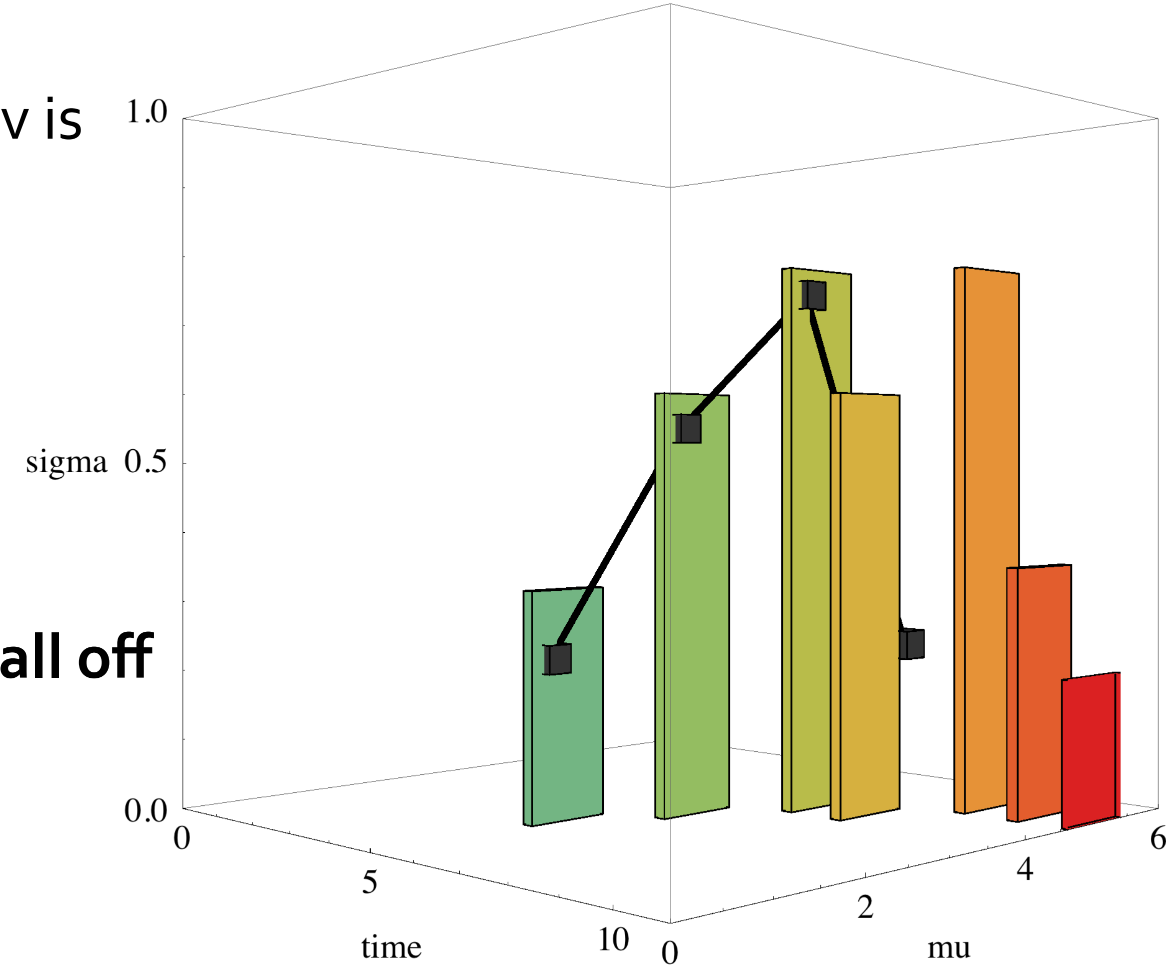
- ModeNear(5, 0.05)
- BV(0.05, 0.4)

Observation fails if stdev is too high

High transition error

Low observation error

Replan: after 3 steps, fall off again



Planning in belief space for 1D continuous space

Goal:

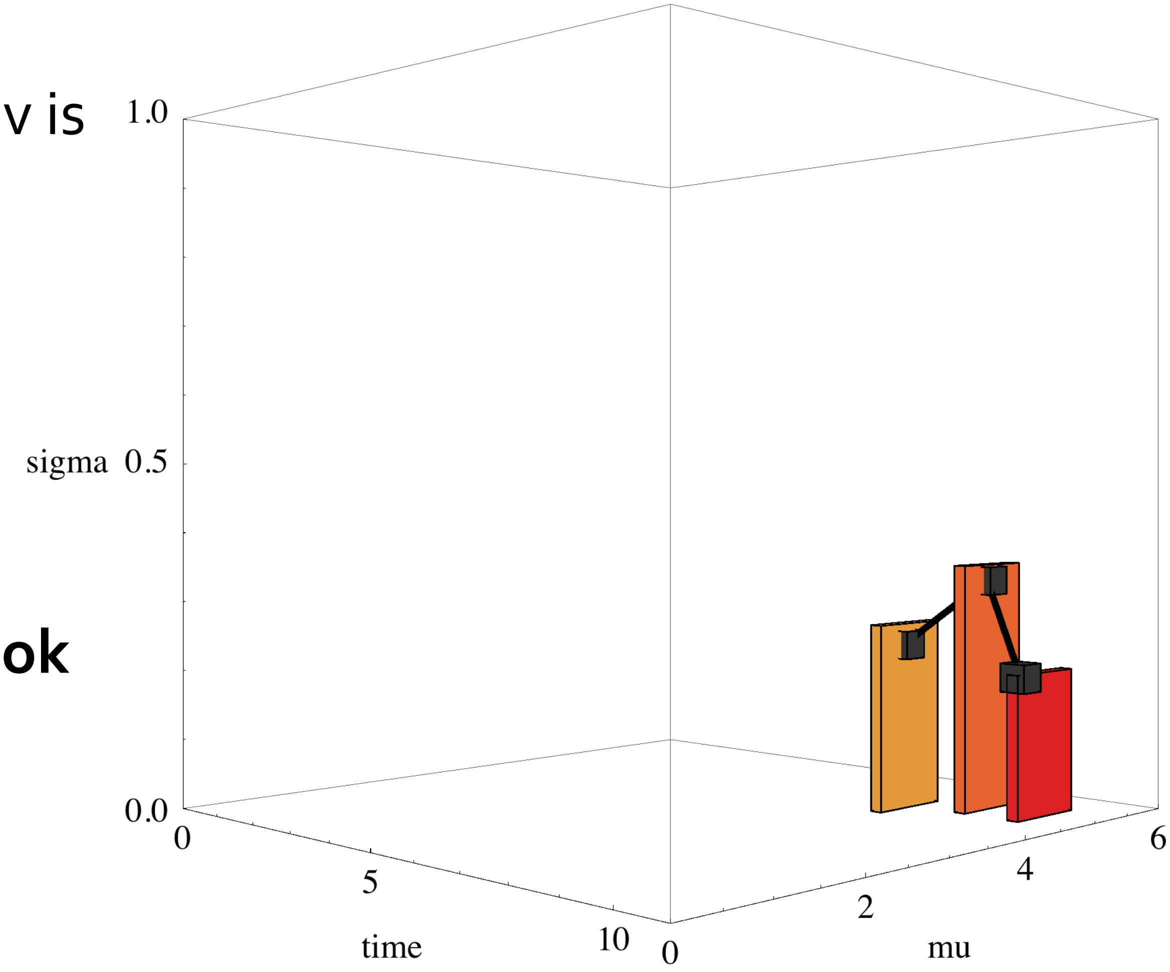
- ModeNear(5, 0.05)
- BV(0.05, 0.4)

Observation fails if stdev is too high

High transition error

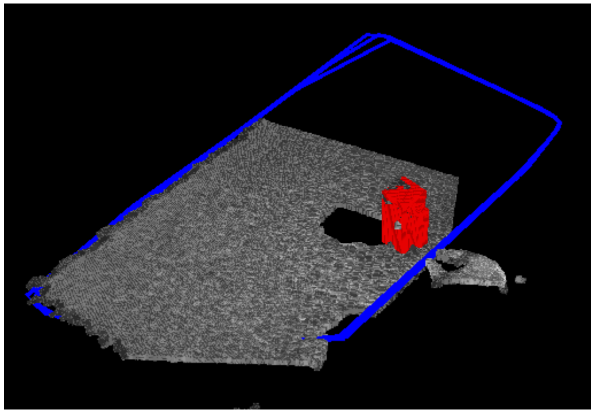
Low observation error

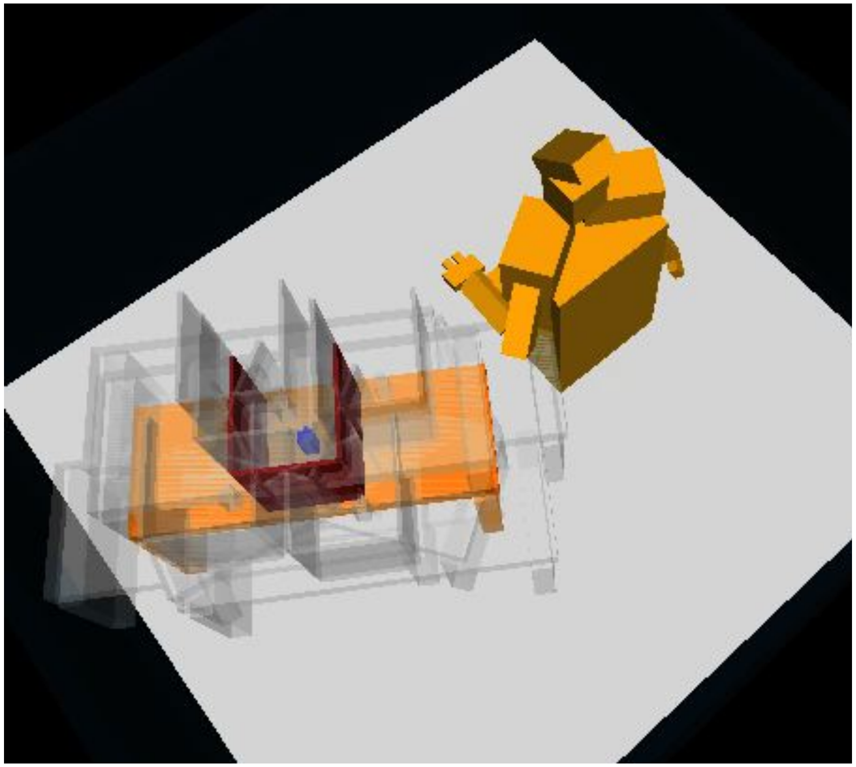
Replan: move, look, look succeeds



State estimation in the world

- Joint tangent-space Gaussian distribution on (X,Y,Z,Theta) poses of robot base and all objects
- Updated based on transitions and observations using EKF
- Assumes objects are recognizable in point cloud data





Fluents for mobile manipulation

Conditional distribution: e.g., dist on object given robot pose

- KVCondPose(Obj, Ref, Eps, Delta)
- KCondPose(Obj, Pose, Ref, Eps, Delta)
- PoseMeanNear(Obj, Pose, Delta)
- KCondRobotConf(Conf, Ref, Eps, Delta)
- RobotConfNear(Conf, Delta)
- KContents(Region)
- NotKNotClear(Region)
- KClearX(Region, Exceptions)

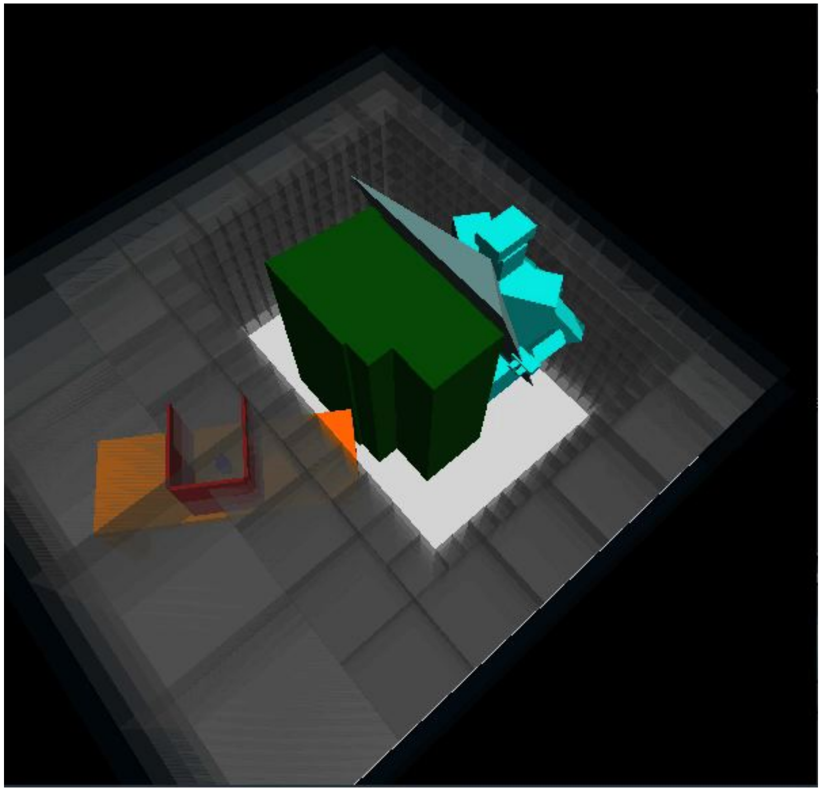
not KContents(Region) or KClear(Region)

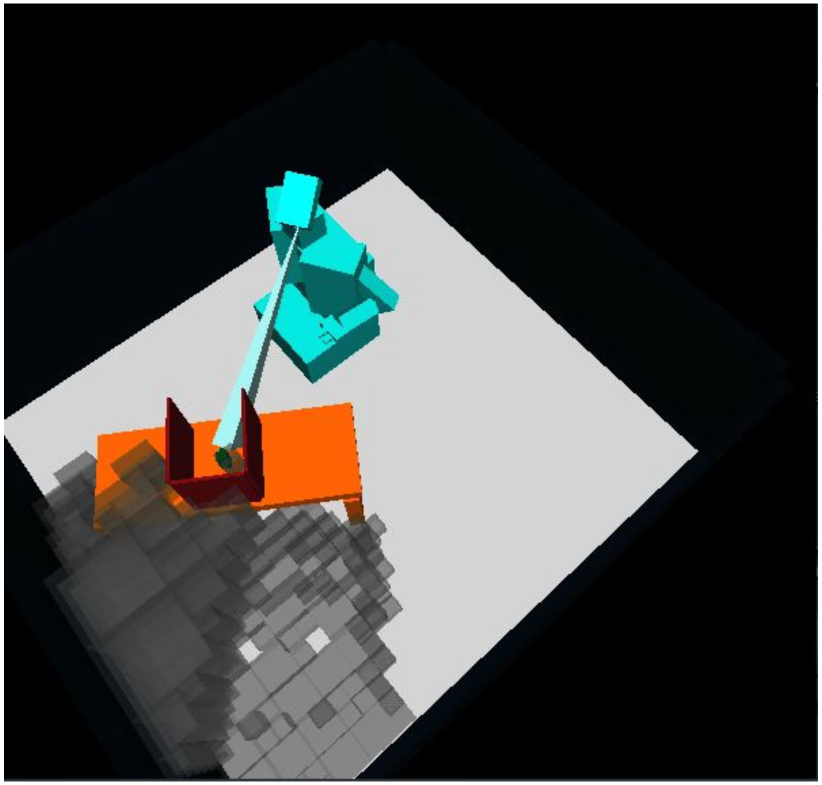
Kcontents(Region) and nothing overlaps

Logical fluents only need to be:

- **tested** in a detailed continuous representation of the belief
- **regressed** backward through actions

Representing known space





Geometric reasoning about visibility

Planning operator in belief space: Pick

```
Pick(O, ObjPose):
  pre: KVCondPose(O, ‘table’, bigEps, planDelta)

  exists: ObjPose ∈ {modeObjPose} U generateParking(O)
          P ∈ generatePickPaths(ObjPose)

  pre: KClearX(sweptVol(P), O)
       KHolding(None)
       KCondPose(O, ObjLoc, ‘table’, eps, graspDelta)
       KCondRobotConf(O, baseConf(P), smallEps,grspDelta)

  result: KHolding(O)
```

Need to know O pose roughly before further planning

Suggest approximate robot path quickly

Object is close to where we first saw it

Robot base is in position and well localized wrt O

Detailed prim execution depends on detailed beliefs: pose, shape, mass, etc

Planning operator in belief space: Look

```
Look(O):
  pre: PoseMeanNear(O, OPose, Delta)

  exists: (Rconf, ViewCone) ∈ generateViewPose(O, OPose)

  pre: NotKNotClear(ViewCone)
       KCondPose(O, OPose, RefObj,
                 PNRegress(eps, obsVar, Delta), Delta)
       KCondRobotConf(O, RobotConf, smallEps, lookDelta)

  result: KCondPose(O, OPose, RefObj, Eps, Delta)
```

Object mean is close to desired pose

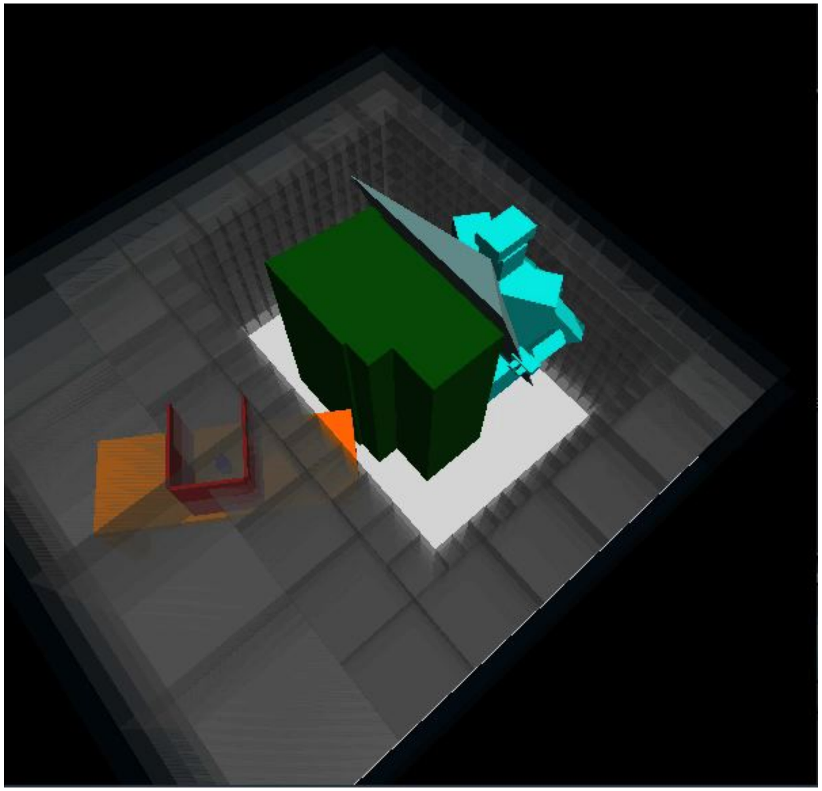
We think the view is clear

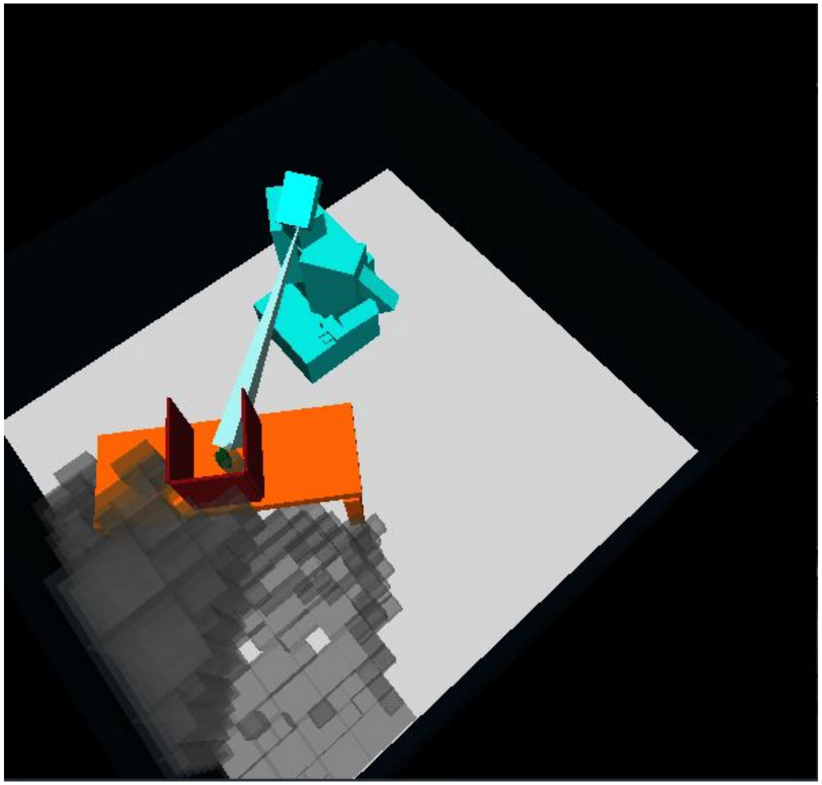
We are somewhat sure of object's pose

Robot is in a good configuration for viewing

We are more sure of object's pose

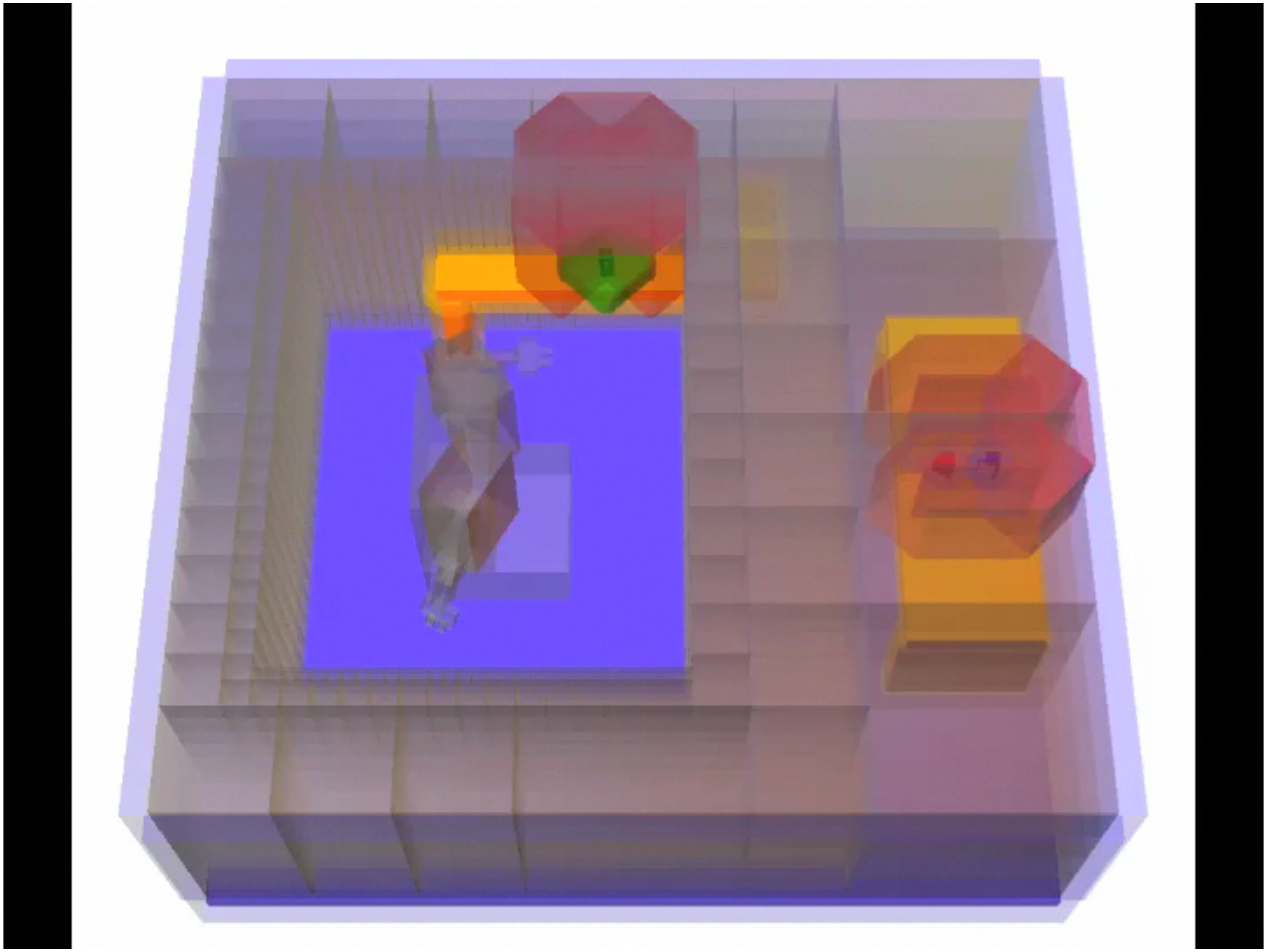
Representing known space



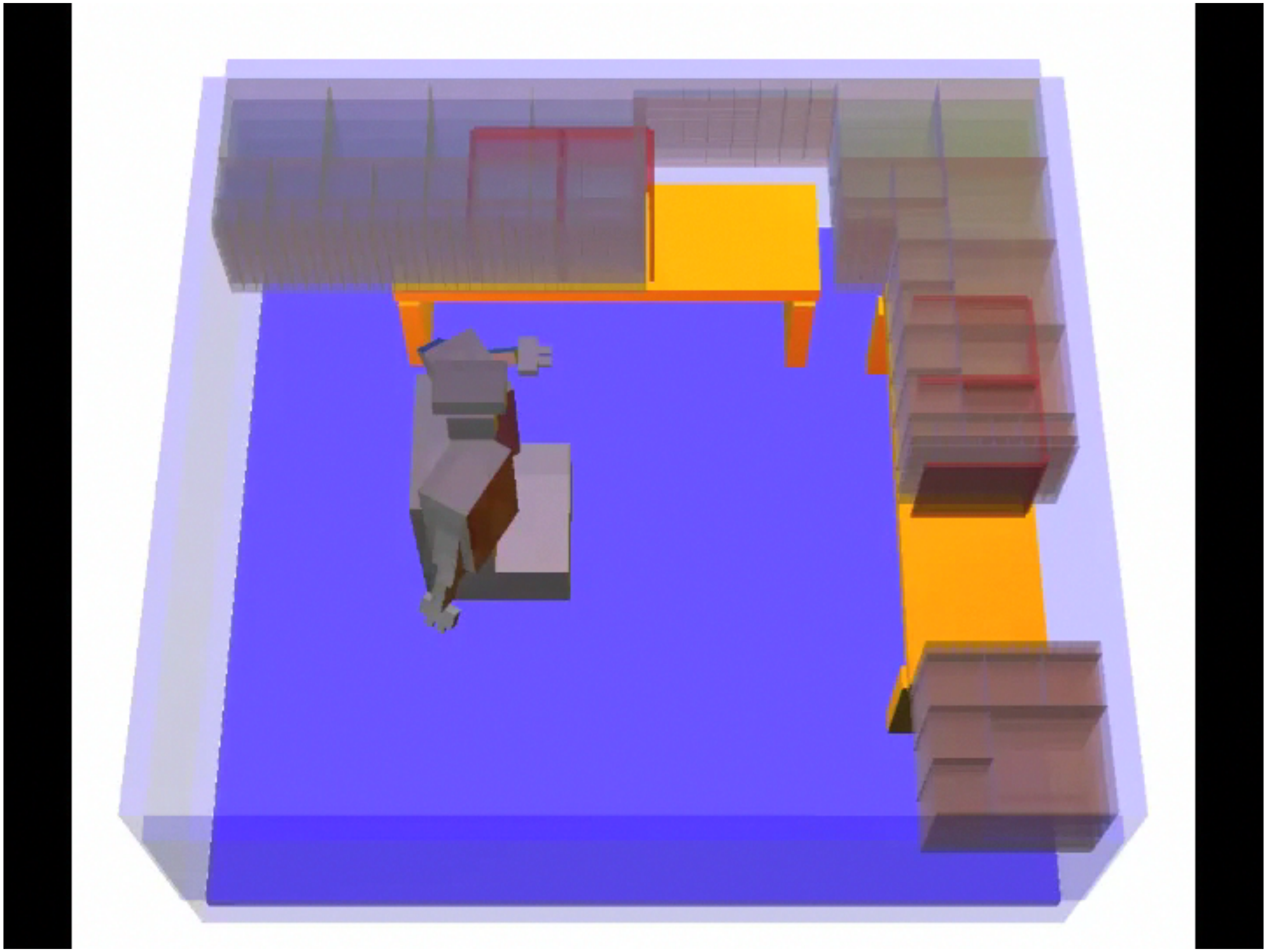


Geometric reasoning about visibility

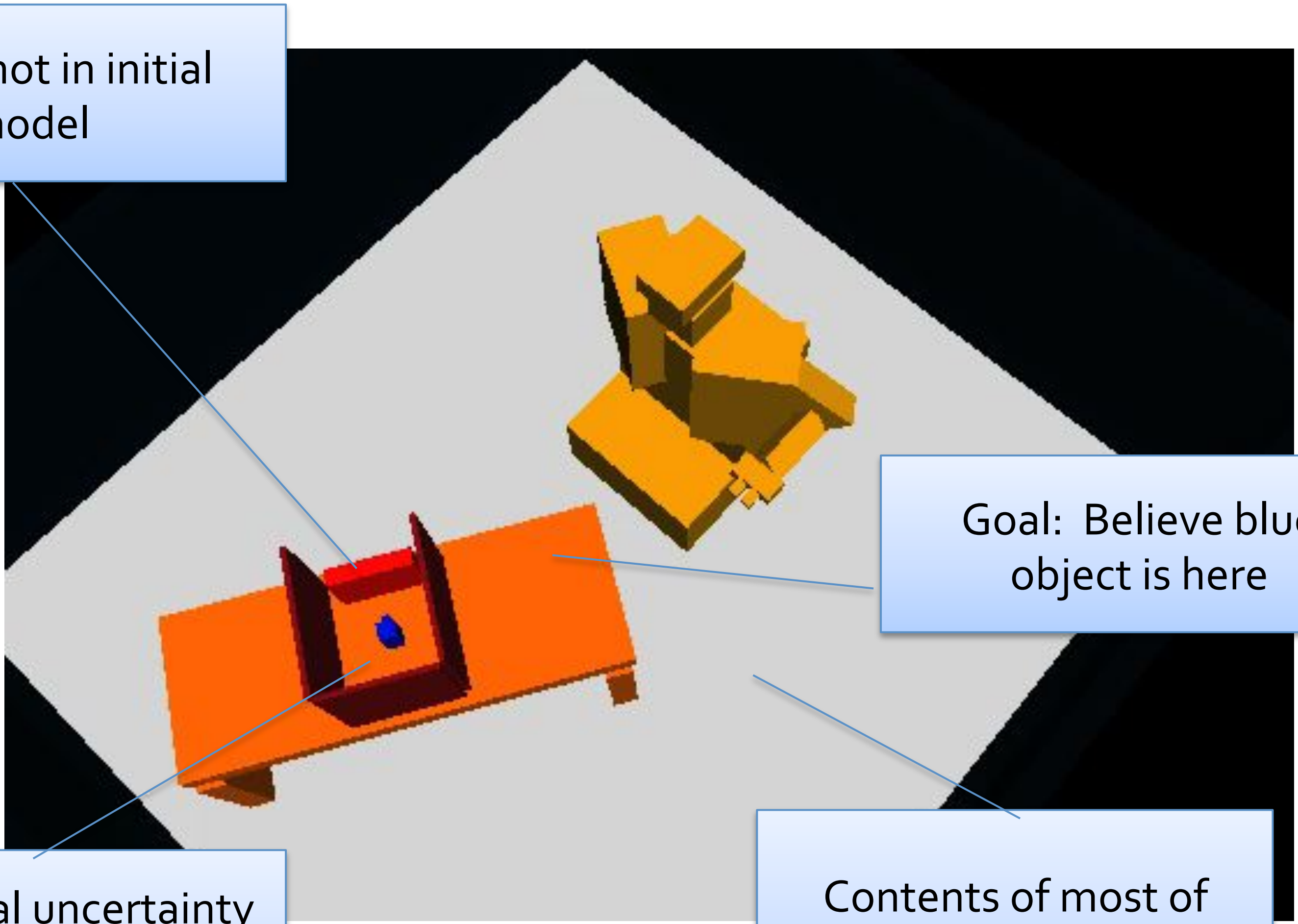
Belief movie



Looking for objects



Example problem



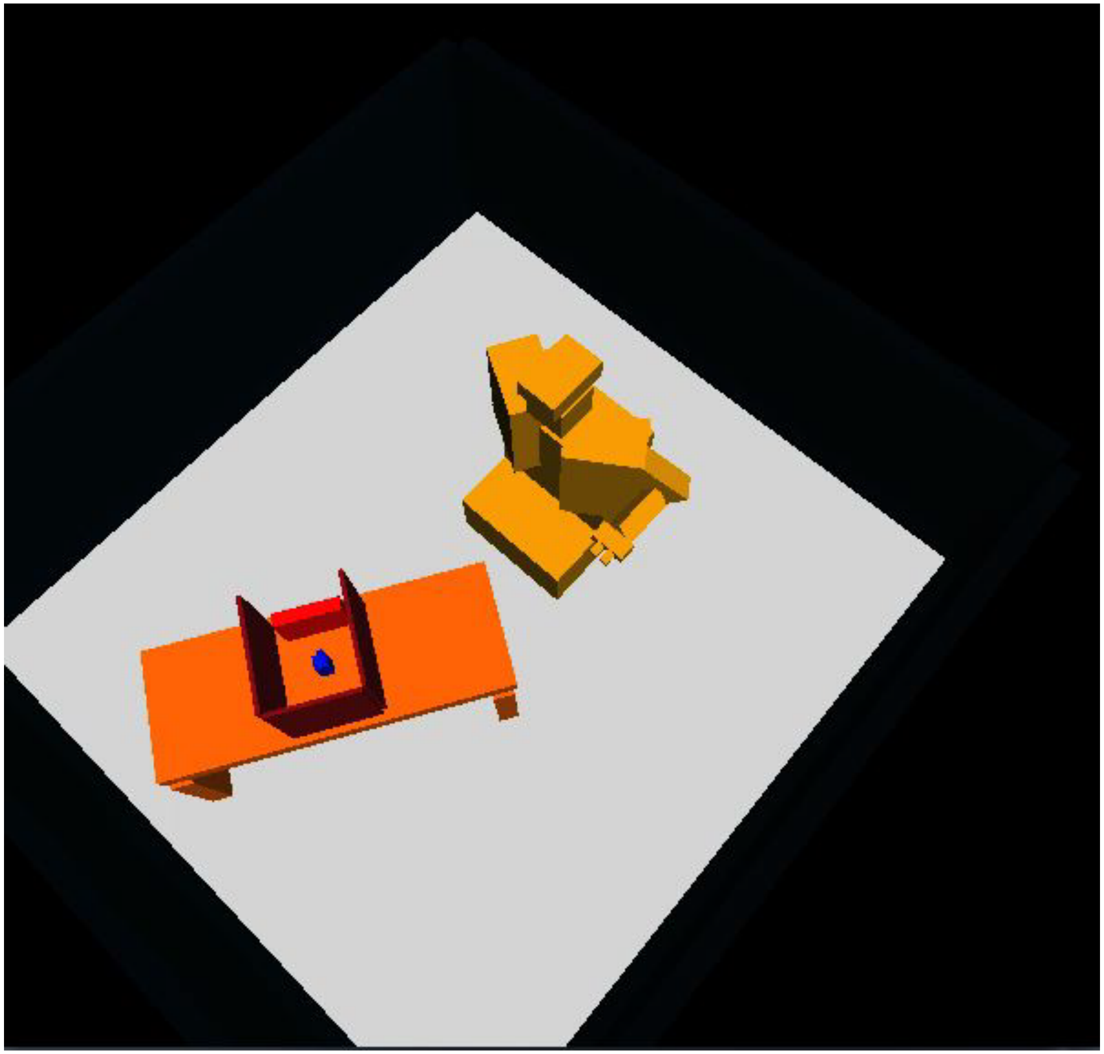
Object not in initial model

Goal: Believe blue object is here

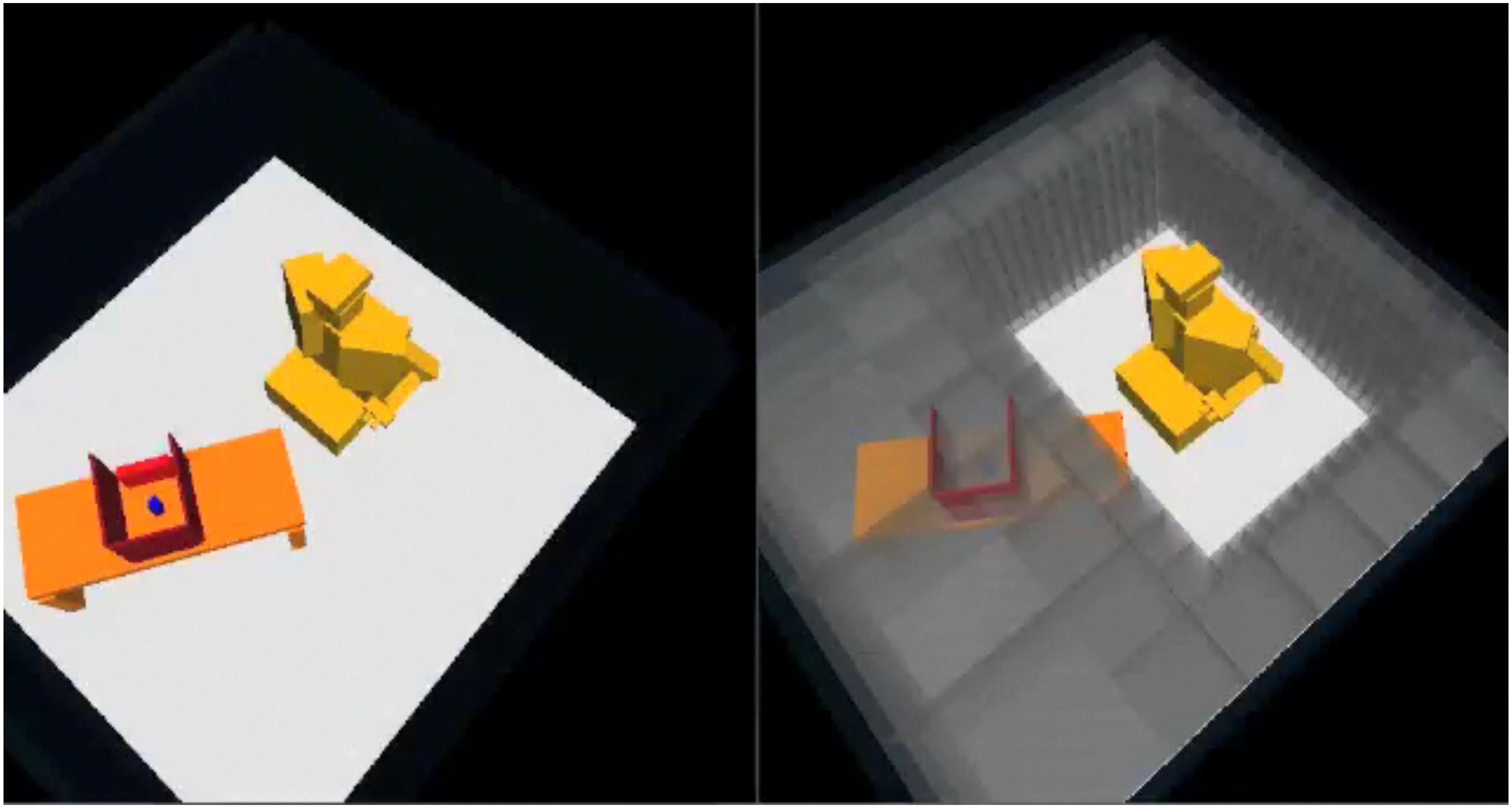
Substantial uncertainty about poses of known objects

Contents of most of space not known

Execution: observation and motion



Execution: belief and planning



PR2