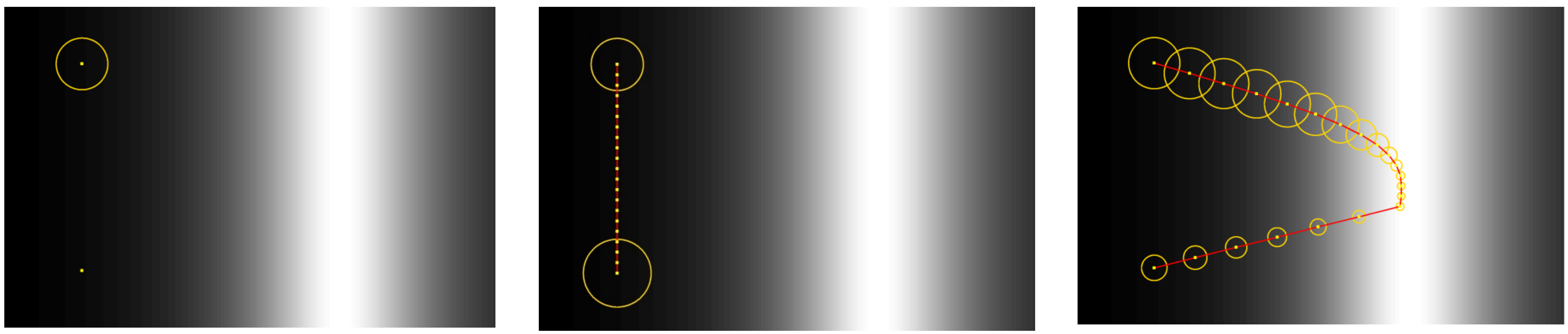


Partially Observable Markov Decision Processes (POMDPs)

Sachin Patil

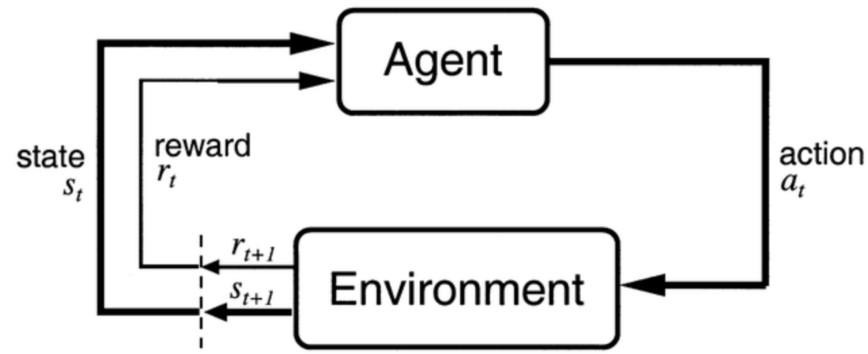
Guest Lecture: CS287 Advanced Robotics

Slides adapted from Pieter Abbeel, Alex Lee

Outline

- Introduction to POMDPs
- Locally Optimal Solutions for POMDPs
  - Trajectory Optimization in (Gaussian) Belief Space
  - Accounting for Discontinuities in Sensing Domains
- Separation Principle

Markov Decision Process (S, A, H, T, R)



Given

- S: set of states
- A: set of actions
- H: horizon over which the agent will act
- T: S x A x S x {0,1,…,H} → [0,1] , $T_t(s,a,s') = P(s_{t+1} = s' \mid s_t = s, a_t = a)$
- R: S x A x S x {0, 1, …, H} → $\mathbb{R}$, $R_t(s,a,s')$ = reward for ($s_{t+1} = s'$, $s_t = s$, a_t =a)

Goal:

- Find π : S x {0, 1, …, H} → A that maximizes expected sum of rewards, i.e.,

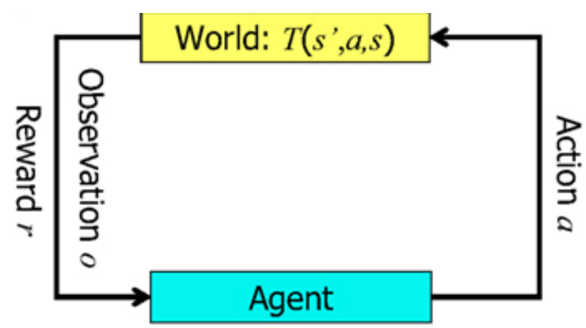
$$\pi^* = \arg\max_{\pi} \mathrm{E}[\sum_{t=0}^{H} R_t(S_t, A_t, S_{t+1}) | \pi]$$

POMDP – Partially Observable MDP

= MDP

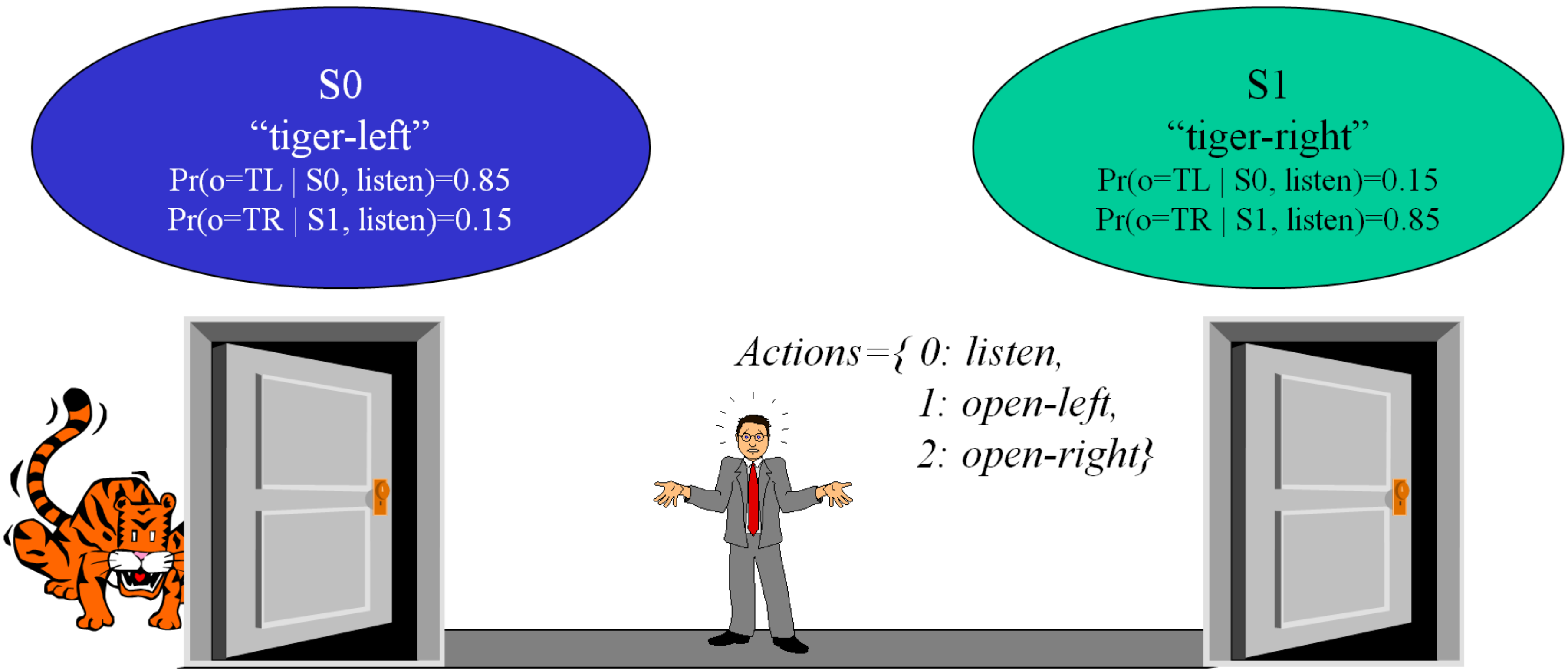
BUT

don't get to observe the state itself, instead get sensory measurements



Now: what action to take given current probability distribution rather than given current state.

POMDPs: Tiger Example



Reward Function

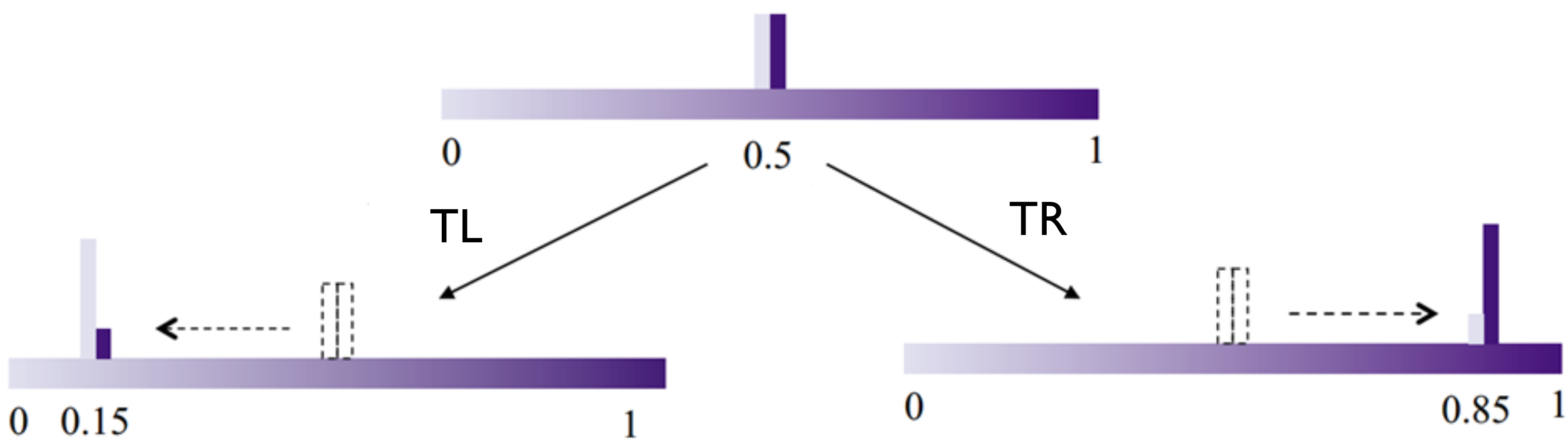
- *Penalty for wrong opening: -100*
- *Reward for correct opening: +10*
- *Cost for listening action: -1*

Observations

- *to hear the tiger on the left (TL)*
- *to hear the tiger on the right(TR)*

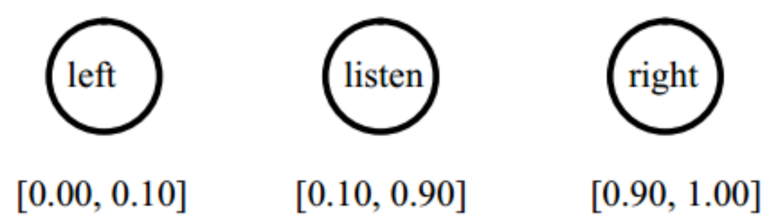
Belief State

- Probability of S0 vs S1 being true underlying state
- Initial belief state: p(S0)=p(S1)=0.5
- Upon listening, the belief state should change according to the Bayesian update (filtering)

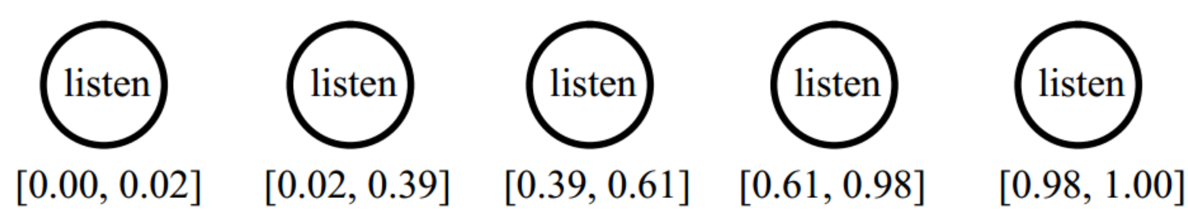


Policy – Tiger Example

- Policy π is a map from [0,1] → {listen, open-left, open-right}
- What should the policy be?
  - Roughly: listen until sure, then open
- But where are the cutoffs?



Tiger example optimal policy for $t = 1$



Tiger example optimal policy for $t = 2$

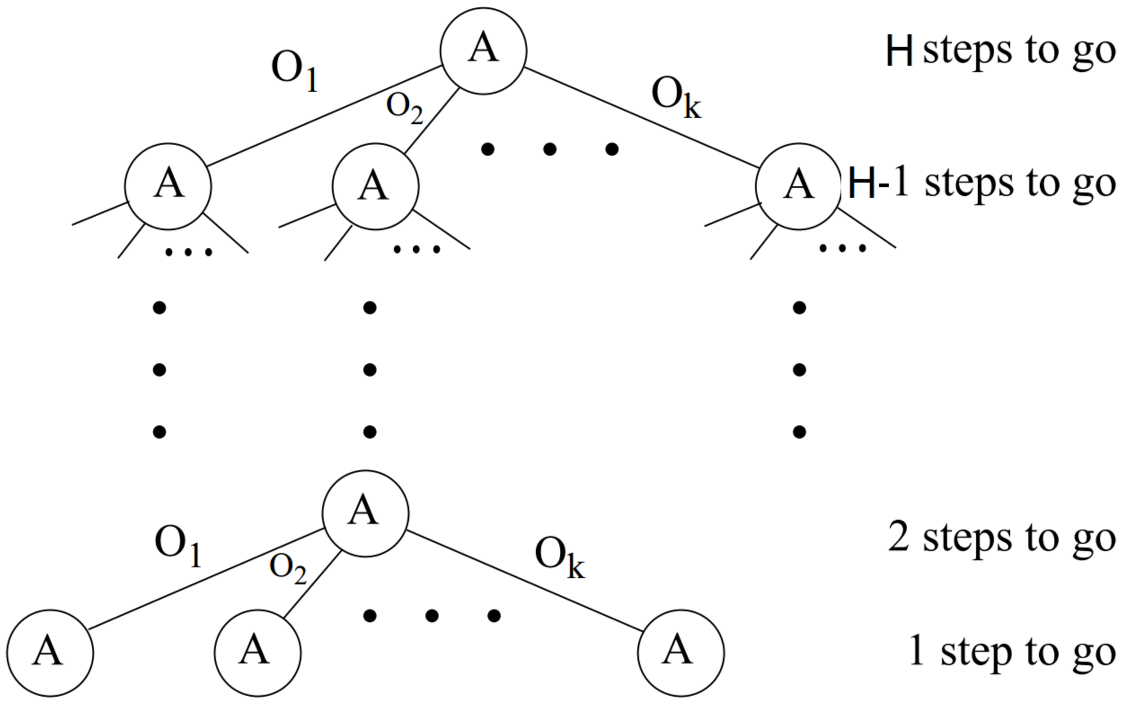
Solving POMDPs

- Canonical solution method 1: Continuous state "belief MDP"
  - Run value iteration, but now the state space is the space of probability distributions
  - → value and optimal action for every possible probability distribution
  - → will automatically trade off information gathering actions versus actions that affect the underlying state

  - Value iteration updates cannot be carried out because uncountable number of belief states – approximation

Solving POMDPs

- Canonical solution method 2:
  - Search over sequences of actions with limited look-ahead
  - Branching over actions and observations



Finite horizon:

$|\mathcal{A}|^{\frac{|\mathcal{O}|^H - 1}{|\mathcal{O}| - 1}}$ nodes

Solving POMDPs

- Approximate solution: becoming tractable for |S| in millions
  - α-vector point-based techniques
  - Monte Carlo Tree Search
  - …Beyond scope of course…

Solving POMDPs

- Canonical solution method 3:
  - Plan in the MDP
  - Probabilistic inference (filtering) to track probability distribution
  - Choose optimal action for MDP for currently most likely state

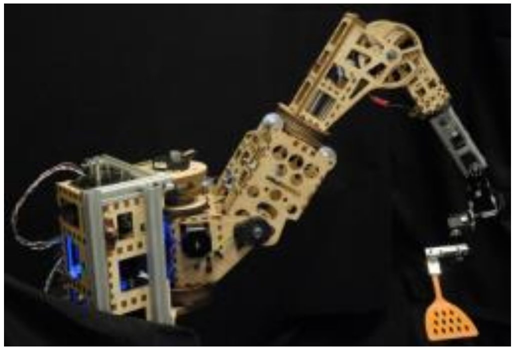
Outline

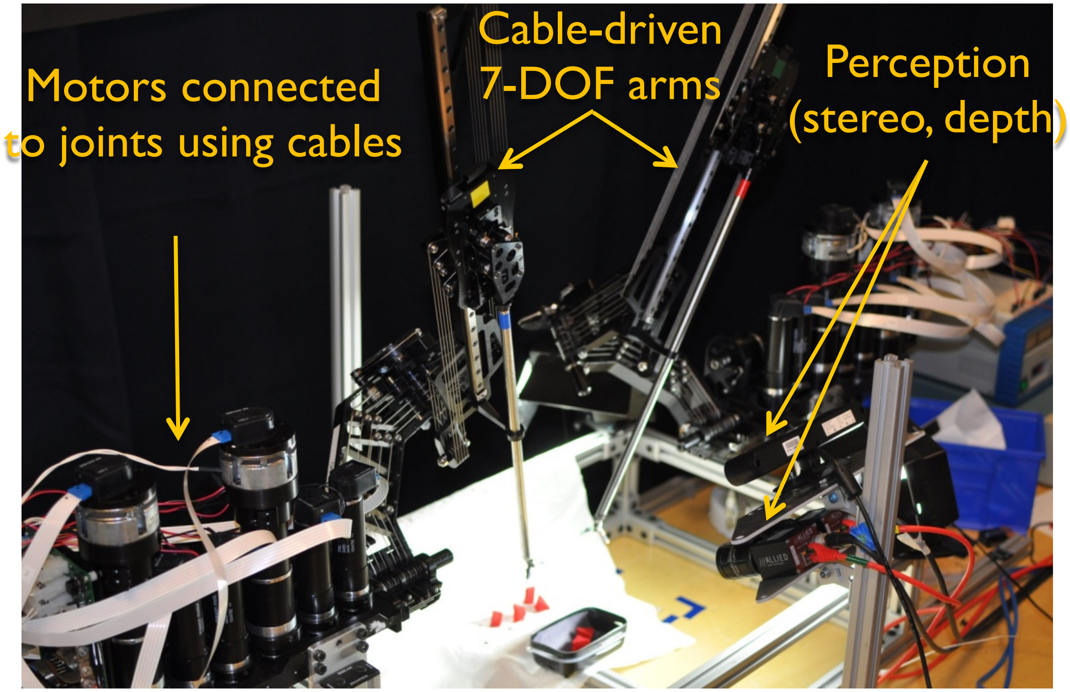
- Introduction to POMDPs
- Locally Optimal Solutions for POMDPs
  - Trajectory Optimization in (Gaussian) Belief Space
  - Accounting for Discontinuities in Sensing Domains
- Separation Principle

Motivation

Facilitate reliable operation of cost-effective robots that use:

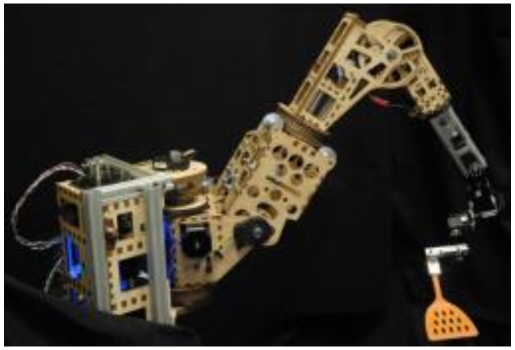
- Imprecise actuation mechanisms – serial elastic actuators, cables
- Inaccurate encoders and sensors – gyros, accelerometers

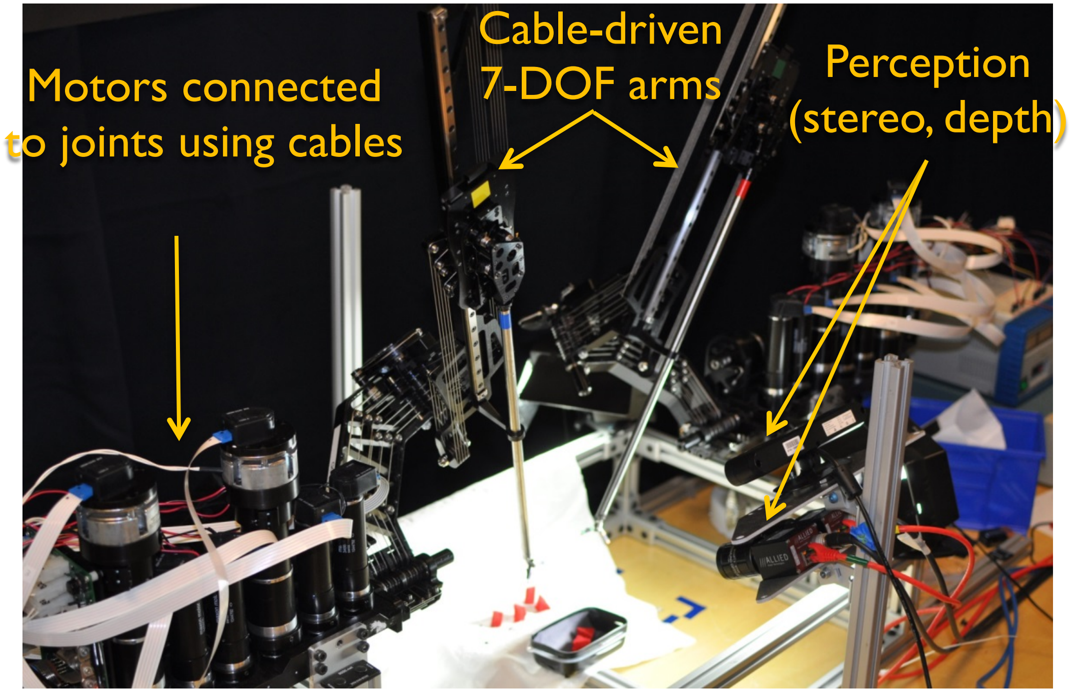




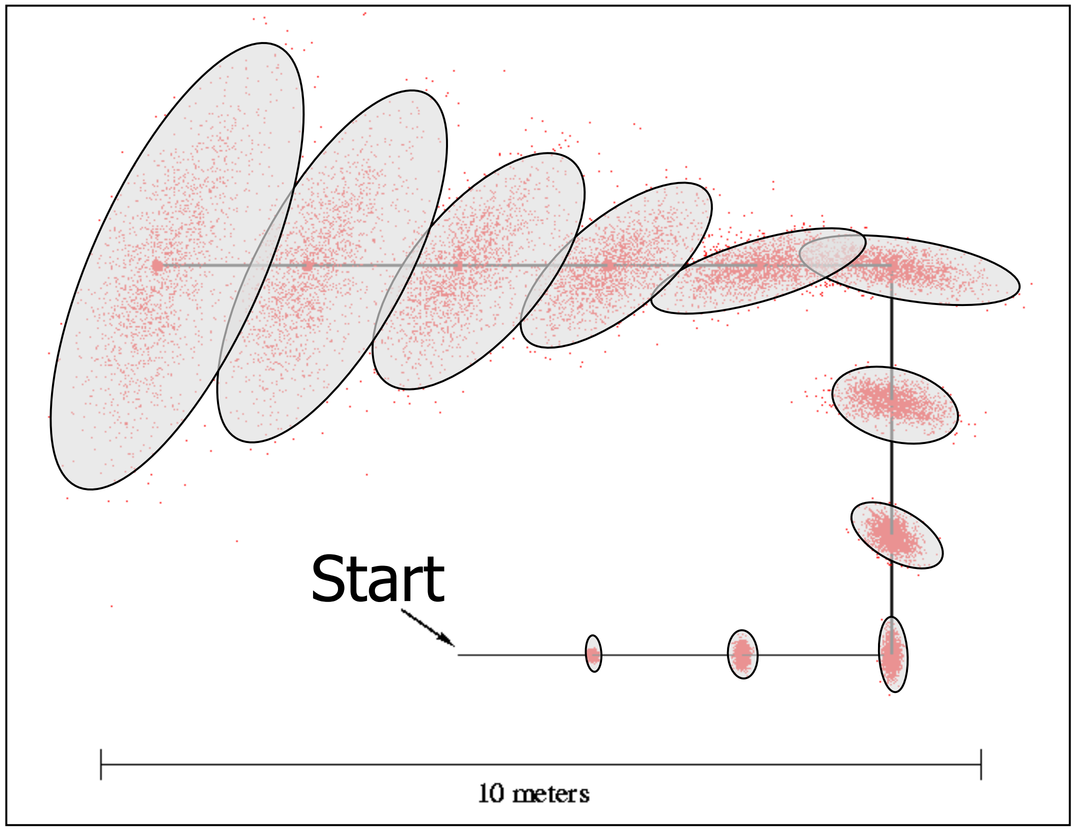
Motivation

Continuous state/action/observation spaces



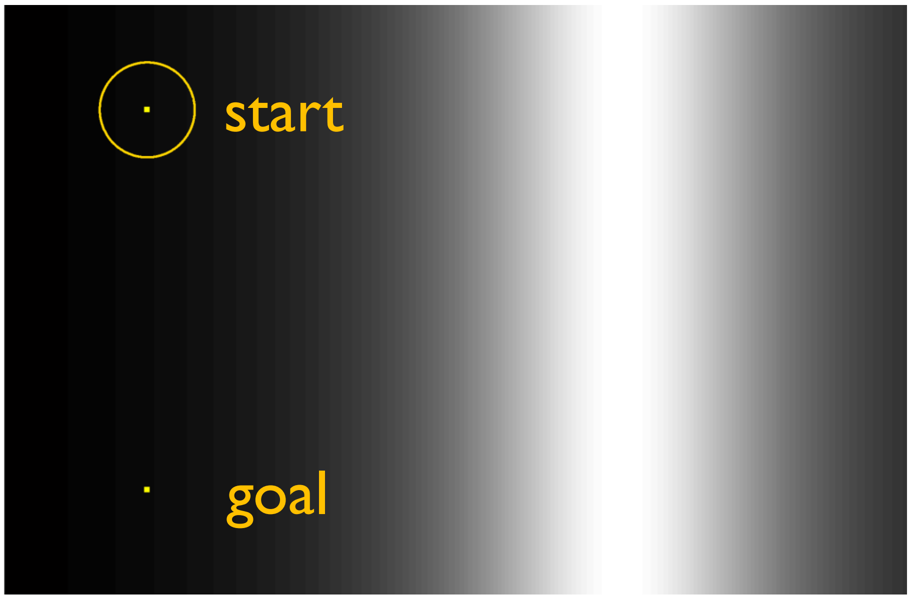


Model Uncertainty As Gaussians

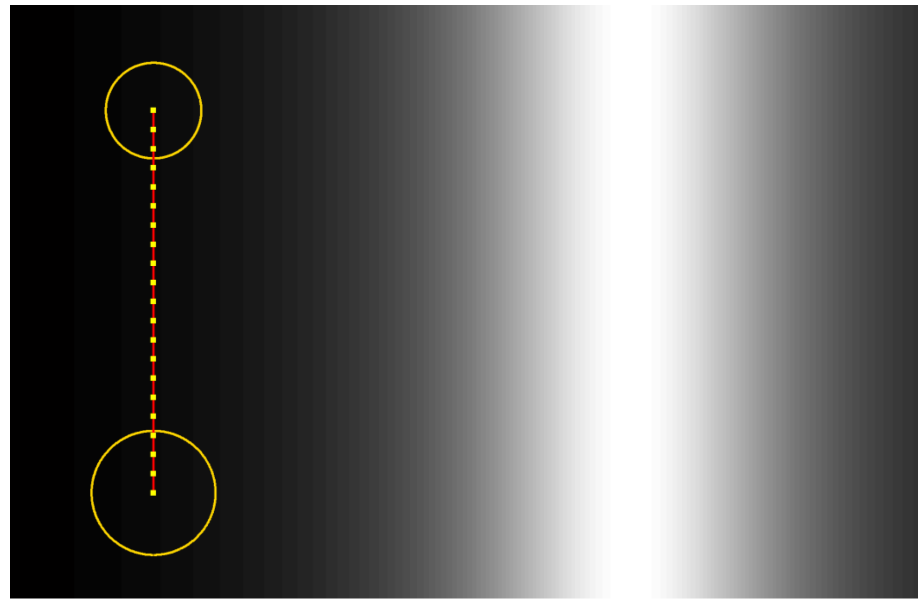


Uncertainty parameterized by
mean and covariance

Dark-Light Domain



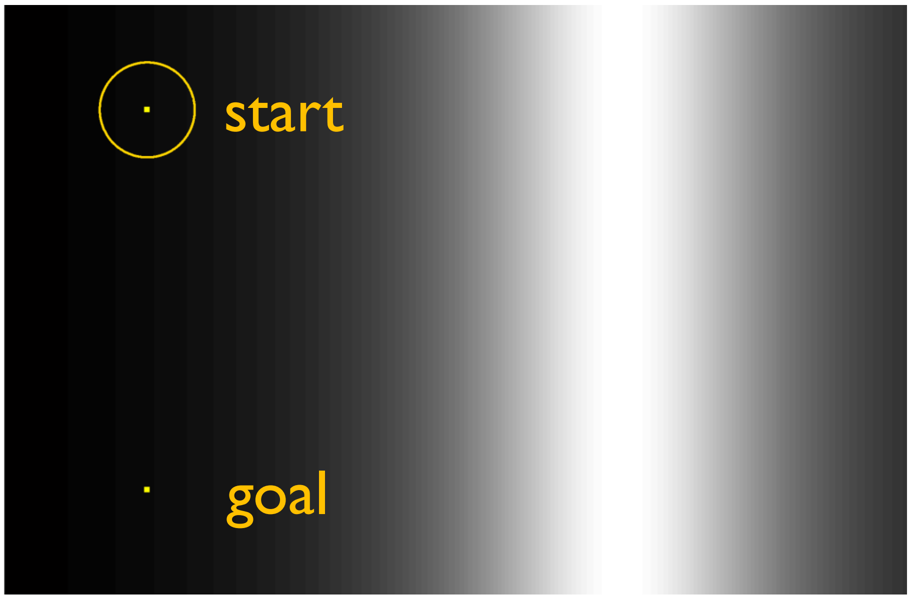
Problem Setup



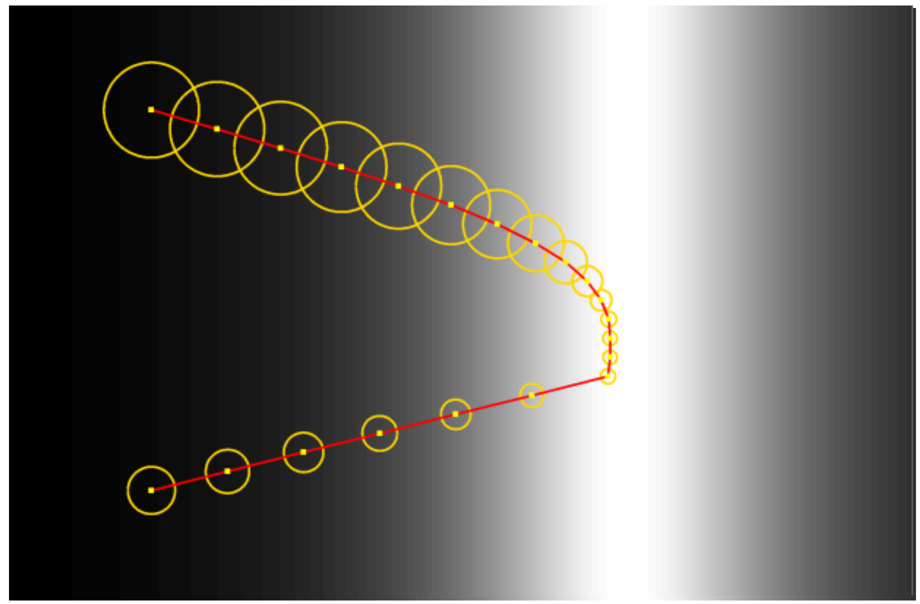
State space plan

[Example from Platt, Tedrake, Kaelbling, Lozano-Perez, 2010]

Dark-Light Domain



Problem Setup



Belief space plan

Tradeoff information gathering vs. actions

Problem Setup

- Stochastic motion and observation Model

$$\mathbf{x}_{t+1} = \mathbf{f}[\mathbf{x}_t, \mathbf{u}_t, \mathbf{m}_t], \qquad \mathbf{m}_t \sim \mathcal{N}[\mathbf{0}, I],$$
$$\mathbf{z}_t = \mathbf{h}[\mathbf{x}_t, \mathbf{n}_t], \qquad \mathbf{n}_t \sim \mathcal{N}[\mathbf{0}, I],$$

  - Non-linear

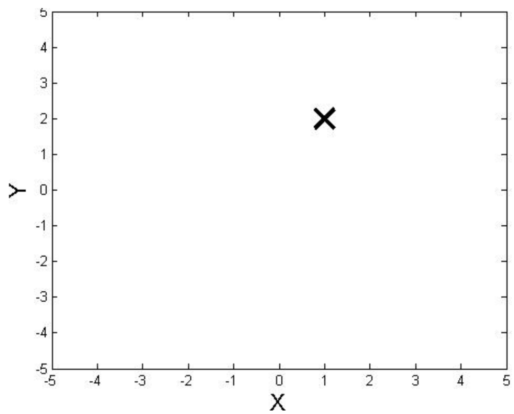
- User-defined objective / cost function

- Plan trajectory that minimizes expected cost

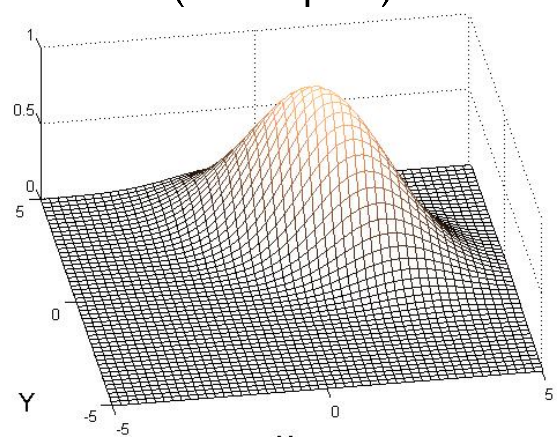
Locally Optimal Solutions

- Belief is Gaussian
  - $\mathbf{b}_t = (\hat{\mathbf{x}}_t, \Sigma_t),$
- Belief dynamics – Bayesian filter
  - [X] Kalman Filter

(underlying state space)



(belief space)



State Space – Trajectory Optimization

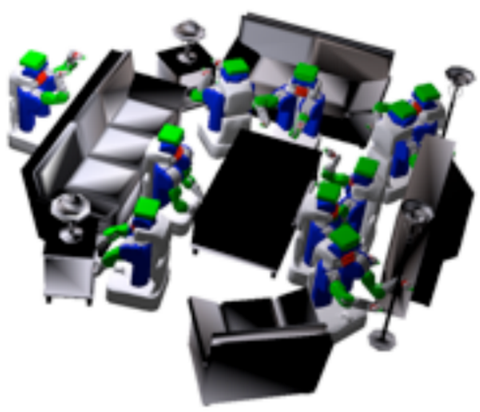
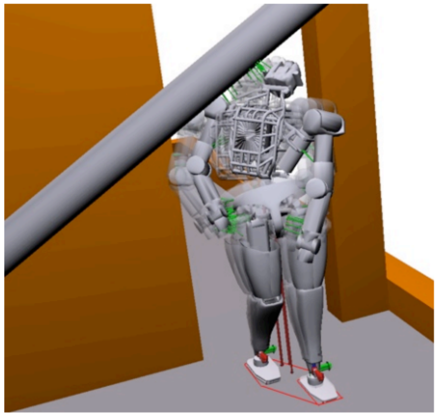
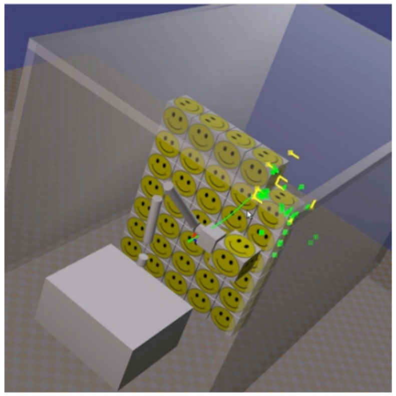
$$\min_{\boldsymbol{\theta}_{1:T}} \sum_t \|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|^2 + \text{ other costs}$$

subject to

no collisions

joint limits

other constraints

(Gaussian) Belief Space Planning

$$
\begin{aligned}
min_{\mu,\Sigma,u} \quad & \sum_{t=0}^{H} c(\mu_t, \Sigma_t, u_t) \\
\text{s.t.} \quad & (\mu_{t+1}, \Sigma_{t+1}) = xKF(\mu_t, \Sigma_t, u_t, w_t, v_t) \\
& \mu_H = \text{goal} \\
& u \in \mathcal{U}
\end{aligned}
$$

(Gaussian) Belief Space Planning

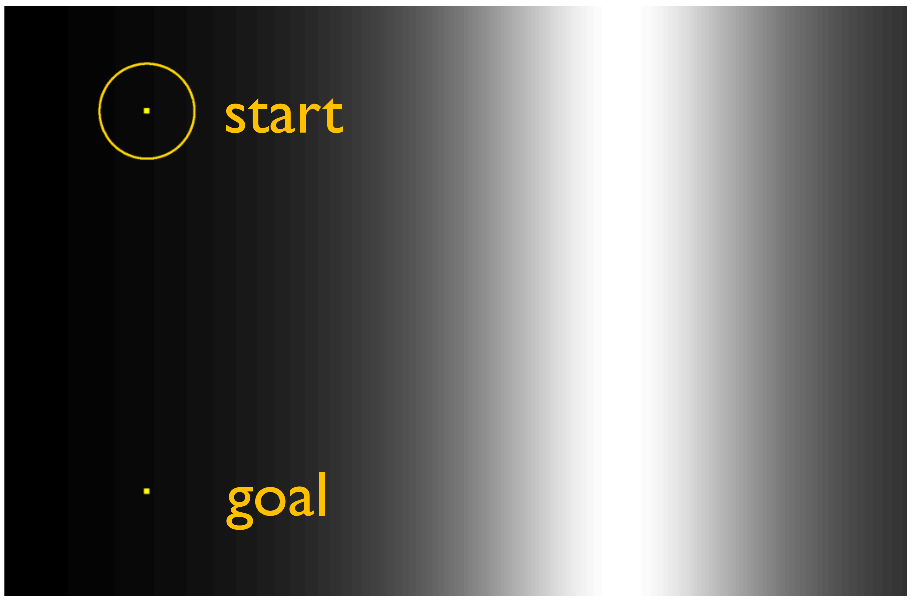
$$
\begin{aligned}
min_{\mu,\Sigma,u} \quad & \sum_{t=0}^{H} c(\mu_t, \Sigma_t, u_t) \\
\text{s.t.} \quad & (\mu_{t+1}, \Sigma_{t+1}) = xKF(\mu_t, \Sigma_t, u_t, 0, 0) \\
& \mu_H = \text{goal} \\
& u \in \mathcal{U}
\end{aligned}
$$

Obstacles?

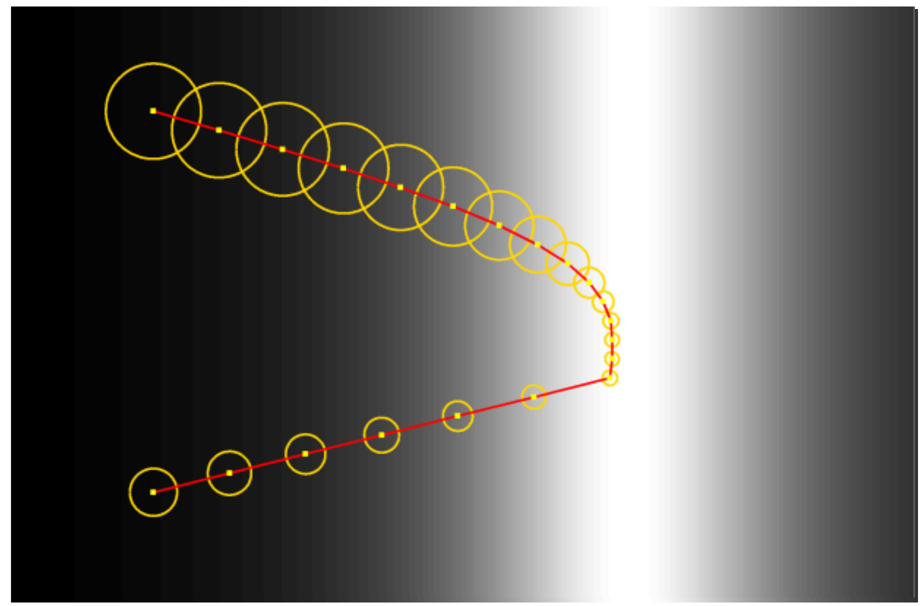
= maximum likelihood assumption for observations
Can now be solved by Sequential Convex Programming

[Platt et al., 2010; also Roy et al ; van den Berg et al. 2011, 2012]

Dark-Light Domain



Problem Setup

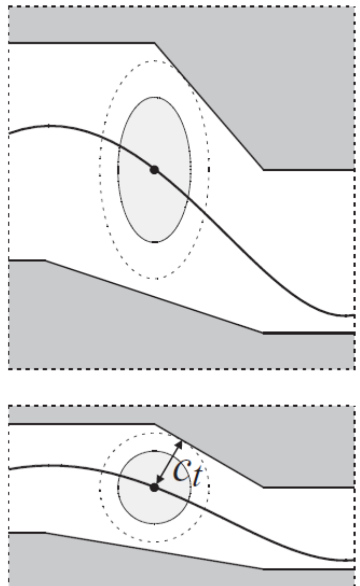


Belief space plan

Tradeoff information gathering vs. actions

Collision Avoidance

- Prior work approximates robot geometry as points or spheres

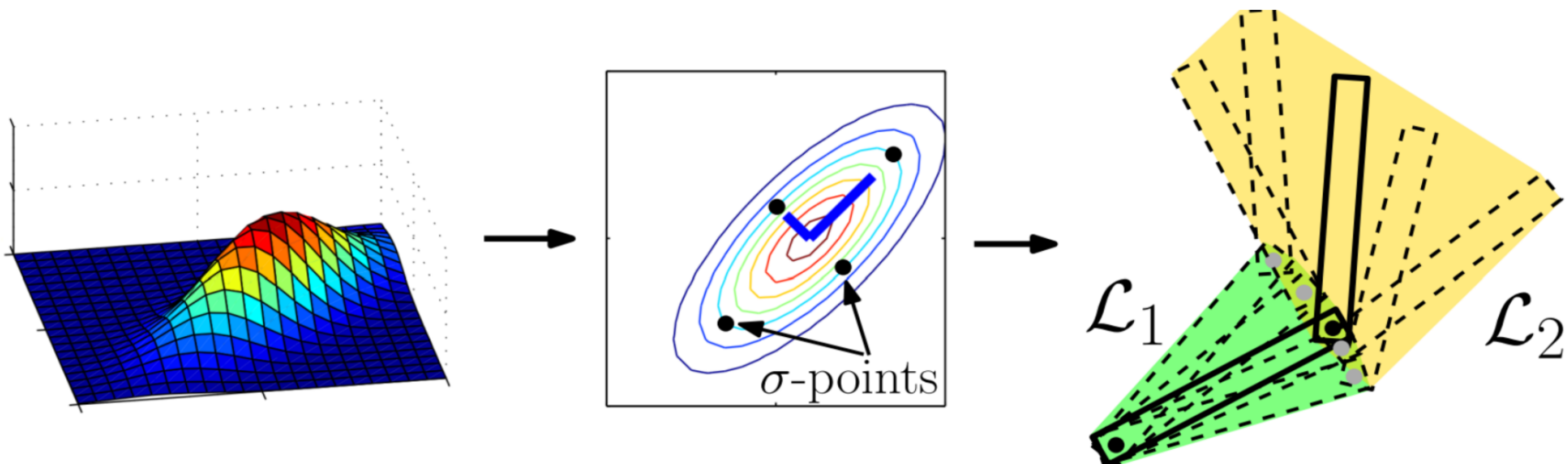


Van den Berg et al.

- Articulated robots cannot be approximated as points/spheres
  - Gaussian noise in joint space
  - Need probabilistic collision avoidance w.r.t robot links

Sigma Hulls

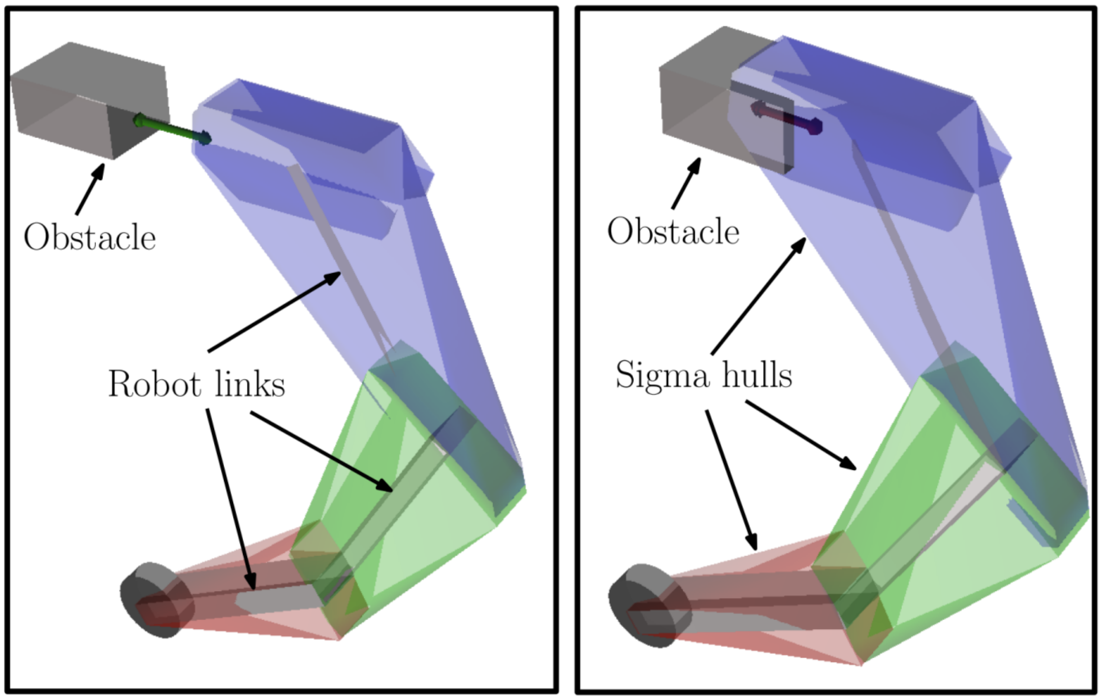
- Definition: Convex hull of a robot link transformed (in joint space) according to sigma points
- Consider sigma points lying on the λ-standard deviation contour of uncertainty covariance (UKF)



Gaussian belief $\quad \mathcal{X} = [\hat{\mathbf{x}} \ldots \hat{\mathbf{x}}] + \lambda[\mathbf{0}\ \sqrt{\Sigma}\ \ -\sqrt{\Sigma}]$ $\quad$ Sigma Hulls (per link)

Collision Avoidance Constraint

Consider signed distance between obstacle and sigma hulls



(a) Obstacle outside sigma hulls (b) Obstacle overlaps sigma hulls

Belief space planning using trajectory optimization

- Gaussian belief state in joint space: $b_t = [\mu_t \; \Sigma_t]$ ← mean
  ← covariance
- Optimization problem:

Variables:

$$\hat{\mathcal{B}} = [\hat{\mathbf{b}}_0 \dots \hat{\mathbf{b}}_T]^T \qquad \hat{\mathcal{U}} = [\hat{\mathbf{u}}_0 \dots \hat{\mathbf{u}}_{T-1}]^T$$

$$\min_{\hat{\mathcal{B}},\hat{\mathcal{U}}} \quad \mathbf{C}(\hat{\mathcal{B}},\hat{\mathcal{U}})$$

$$\text{s.t.}_{\forall t \in \mathcal{T}} \quad \hat{\mathbf{b}}_{t+1} = \mathbf{g}(\hat{\mathbf{b}}_t, \hat{\mathbf{u}}_t), \quad \text{Belief dynamics (UKF)}$$

$$\boxed{\mathbf{\Phi}(\hat{\mathcal{B}},\hat{\mathcal{U}},\lambda) \geq \mathbf{0}, \quad \text{Probabilistic collision avoidance}}$$

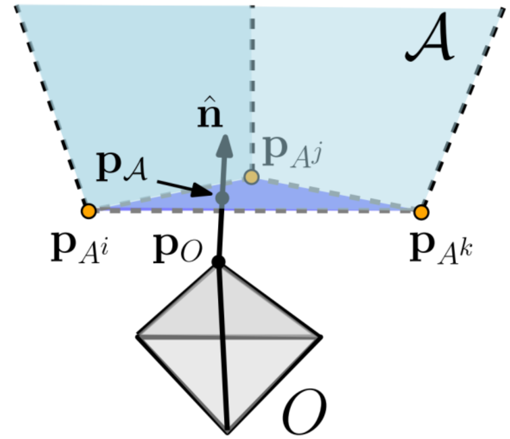
$$\psi(\hat{\mathbf{x}}_T) = \psi_{\text{target}}, \quad \text{Reach desired end-effector pose}$$

$$\hat{\mathbf{u}}_t \in F_{\mathcal{U}}, \quad \text{Control inputs are feasible}$$

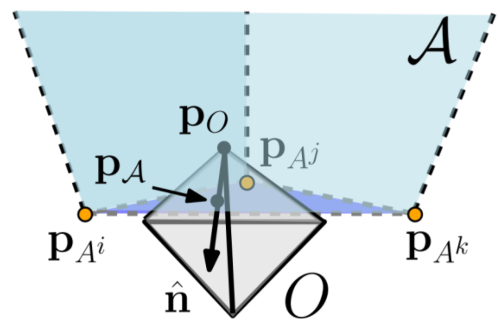
Collision avoidance constraint

- Robot trajectory should stay at least d_{safe} distance from other objects

$$\texttt{sd}(\mathcal{A}, O) \geq d_{\text{safe}} \quad \forall\ O \in \mathcal{O}$$



(a) $\texttt{sd}(\mathcal{A}, O) > 0$



(b) $\texttt{sd}(\mathcal{A}, O) < 0$

Collision avoidance constraint

- Robot trajectory should stay at least d_{safe} distance from other objects

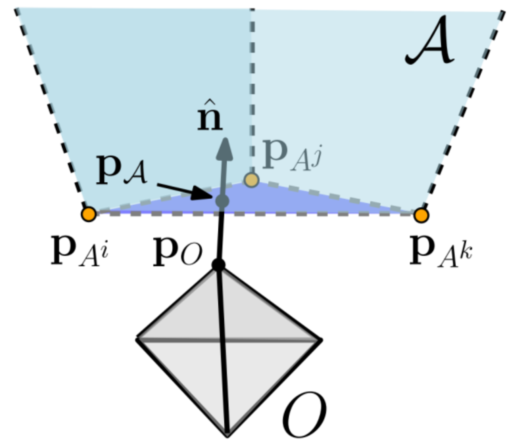
$$\mathtt{sd}(\mathcal{A}, O) \geq d_{\text{safe}} \quad \forall\, O \in \mathcal{O}$$

- Linearize signed distance at current belief

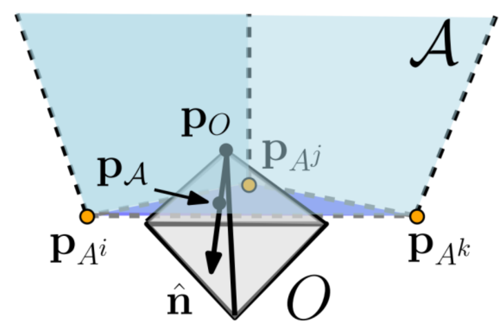
$$\mathtt{sd}_{\mathcal{A}O}(\hat{\mathbf{b}}_t) \approx \hat{\mathbf{n}}(\bar{\mathbf{b}}_t) \cdot (\mathbf{p}_O - \mathbf{p}_{\mathcal{A}}(\hat{\mathbf{b}}_t))$$

$$\mathtt{sd}_{\mathcal{A}O}(\hat{\mathbf{b}}_t) \approx \mathtt{sd}_{\mathcal{A}O}(\bar{\mathbf{b}}_t) + S_t(\hat{\mathbf{b}}_t - \bar{\mathbf{b}}_t),$$

$$S_t = \frac{\partial \mathtt{sd}_{\mathcal{A}O}}{\partial \hat{\mathbf{b}}}(\bar{\mathbf{b}}_t) \approx -\hat{\mathbf{n}}(\bar{\mathbf{b}}_t)^T \frac{\partial \mathbf{p}_{\mathcal{A}}}{\partial \hat{\mathbf{b}}}(\bar{\mathbf{b}}_t).$$



(a) $\mathtt{sd}(\mathcal{A}, O) > 0$



(b) $\mathtt{sd}(\mathcal{A}, O) < 0$

Collision avoidance constraint

- Robot trajectory should stay at least d_{safe} distance from other objects

$$\mathtt{sd}(\mathcal{A}, O) \geq d_{\text{safe}} \quad \forall\, O \in \mathcal{O}$$

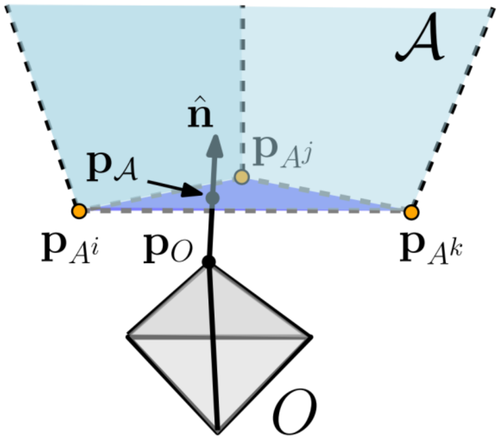
- Linearize signed distance at current belief

$$\mathtt{sd}_{\mathcal{A}O}(\hat{\mathbf{b}}_t) \approx \hat{\mathbf{n}}(\bar{\mathbf{b}}_t) \cdot (\mathbf{p}_O - \mathbf{p}_{\mathcal{A}}(\hat{\mathbf{b}}_t))$$

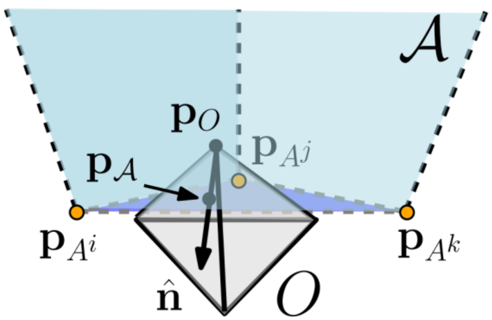
$$\mathtt{sd}_{\mathcal{A}O}(\hat{\mathbf{b}}_t) \approx \mathtt{sd}_{\mathcal{A}O}(\bar{\mathbf{b}}_t) + S_t(\hat{\mathbf{b}}_t - \bar{\mathbf{b}}_t),$$
$$S_t = \frac{\partial \mathtt{sd}_{\mathcal{A}O}}{\partial \hat{\mathbf{b}}}(\bar{\mathbf{b}}_t) \approx -\hat{\mathbf{n}}(\bar{\mathbf{b}}_t)^T \frac{\partial \mathbf{p}_{\mathcal{A}}}{\partial \hat{\mathbf{b}}}(\bar{\mathbf{b}}_t).$$

- Consider the closest point $\mathbf{p}_{\mathcal{A}}(\hat{\mathbf{b}}_t)$ lies on a face spanned by vertices $\mathbf{p}_{\mathcal{A}^i}, \mathbf{p}_{\mathcal{A}^j}, \mathbf{p}_{\mathcal{A}^k}$

$$\frac{\partial \mathbf{p}_{\mathcal{A}}}{\partial \hat{\mathbf{b}}}(\bar{\mathbf{b}}_t) = \sum_{l \in \{i,j,k\}} \alpha_l \frac{\partial \mathbf{p}_{\mathcal{A}^l}}{\partial \hat{\mathbf{b}}}(\bar{\mathbf{b}}_t)$$



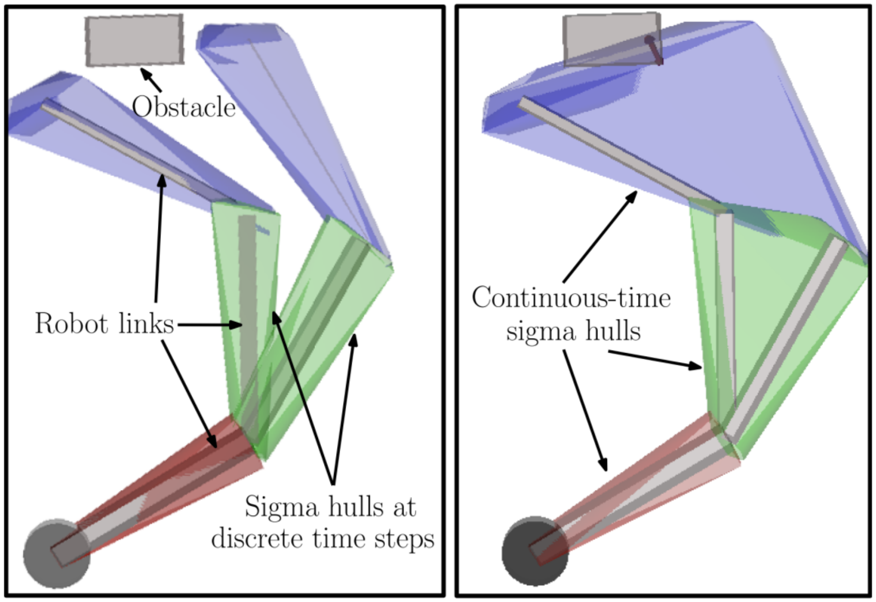
(a) $\mathtt{sd}(\mathcal{A}, O) > 0$



(b) $\mathtt{sd}(\mathcal{A}, O) < 0$

Continuous Collision Avoidance Constraint

- Discrete collision avoidance can lead to trajectories that collide with obstacles in between time steps
- Use convex hull of sigma hulls between consecutive time steps $\mathtt{sd}(\mathtt{convhull}(\mathcal{A}_t, \mathcal{A}_{t+1}), O) \geq d_{\text{safe}} \quad \forall\, O \in \mathcal{O}$
- Advantages:
  - Solutions are collision-free in between time-steps
  - Discretized trajectory can have less time-steps



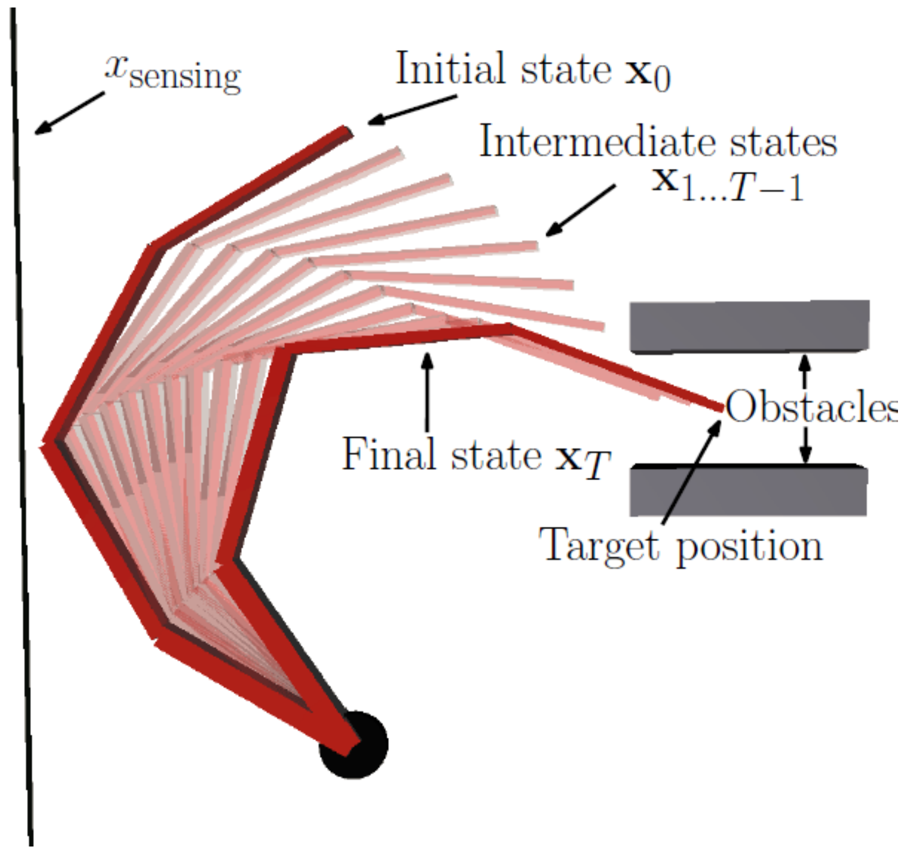
(a) Obstacle does not collide with discrete-time sigma hulls
(b) Obstacle overlaps with continuous-time sigma hulls

Model Predictive Control (MPC)

- During execution, update the belief state based on the actual observation
- Re-plan after every belief state update
- Effective feedback control, provided one can re-plan sufficiently fast

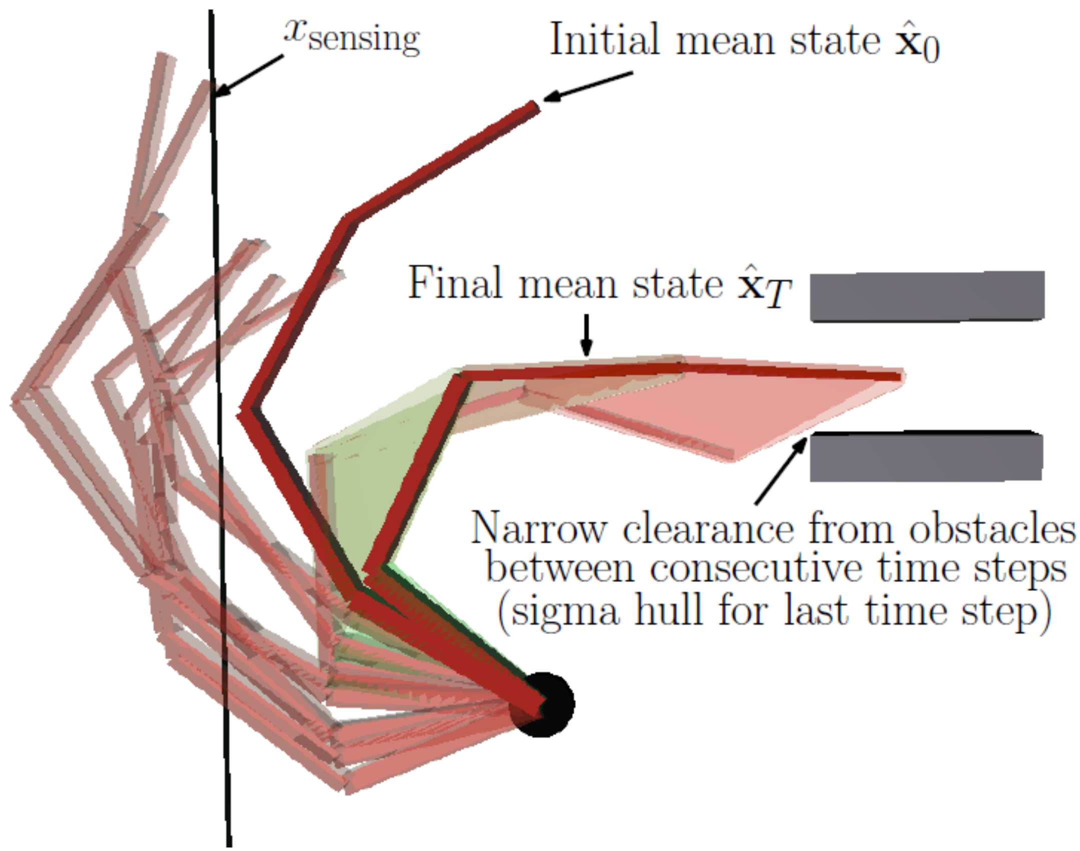
Example: 4-DOF planar robot

State space trajectory



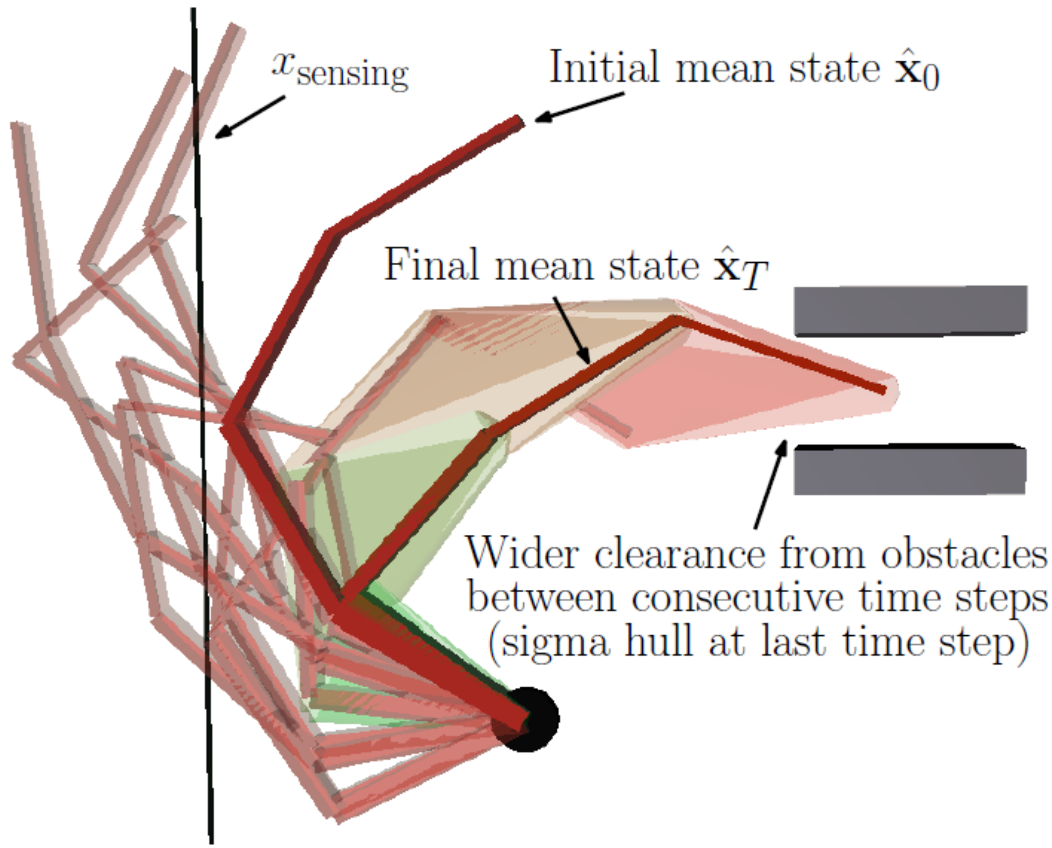
Example: 4-DOF planar robot

1-standard deviation belief space trajectory

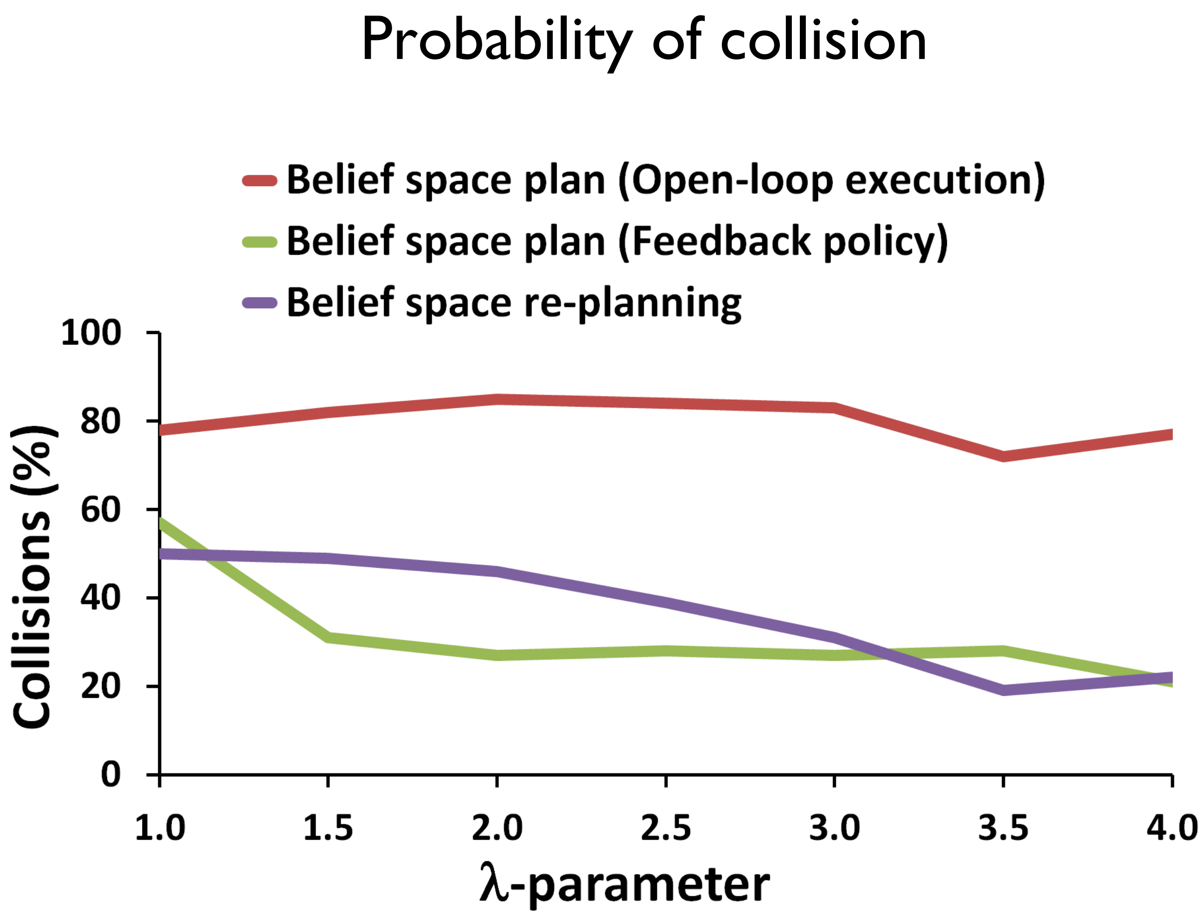


Example: 4-DOF planar robot

4-standard deviation belief space trajectory

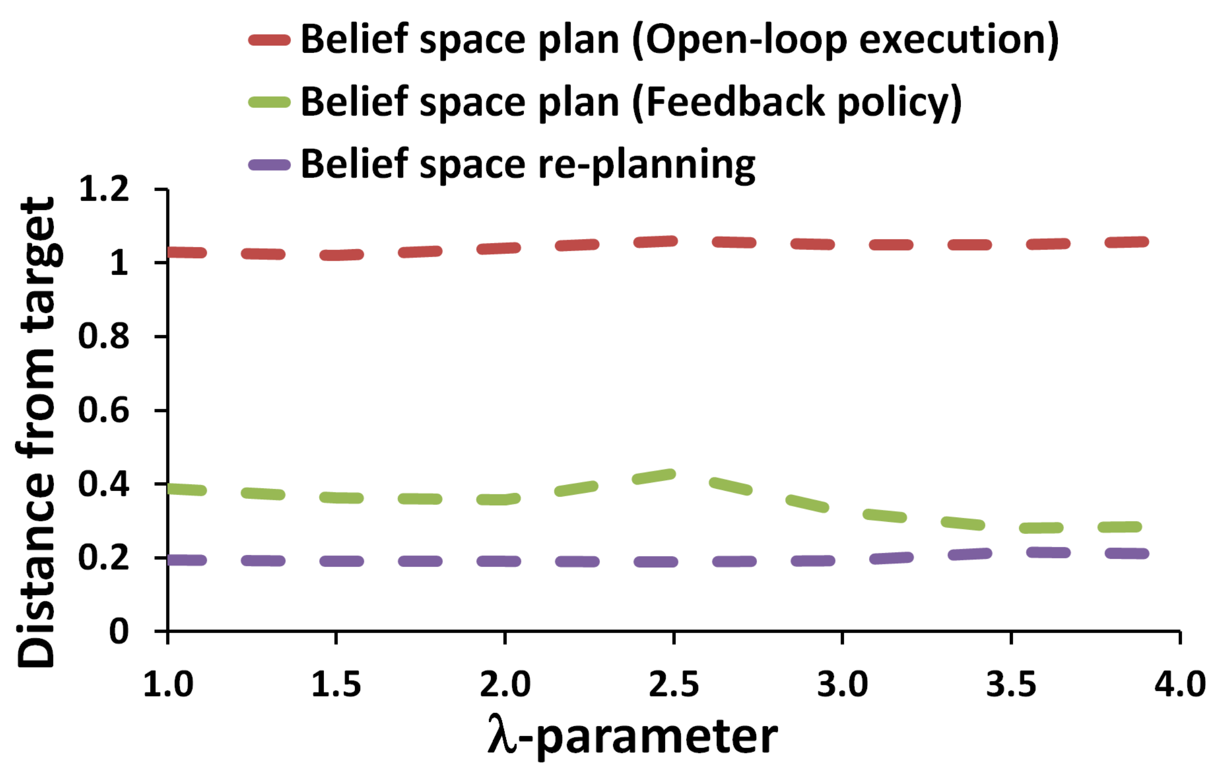


Experiments: 4-DOF planar robot



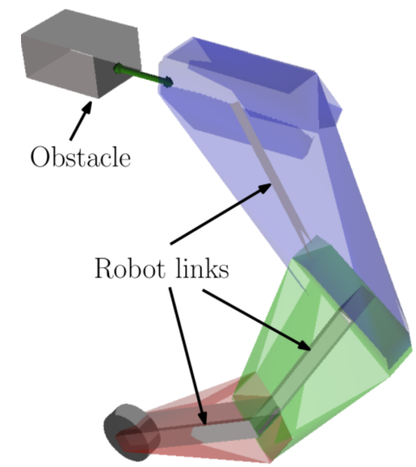
Experiments: 4-DOF planar robot

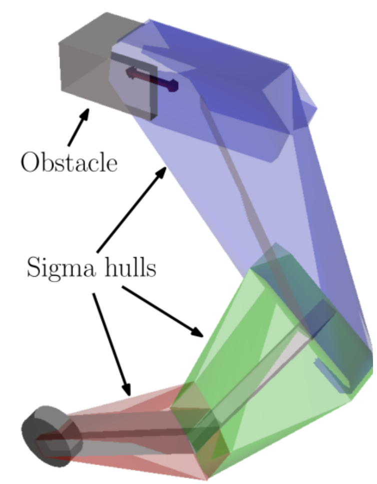
Mean distance from target



Take-Away

- Efficient trajectory optimization in Gaussian belief spaces to reduce task uncertainty
- Prior work approximates robot geometry as a point or a single sphere
- Pose collision constraints using signed distance between sigma hulls of robot links and obstacles
- Sigma hulls never explicitly computed – fast convex collision checking and analytical gradients
- Iterative re-planning in belief space (MPC)

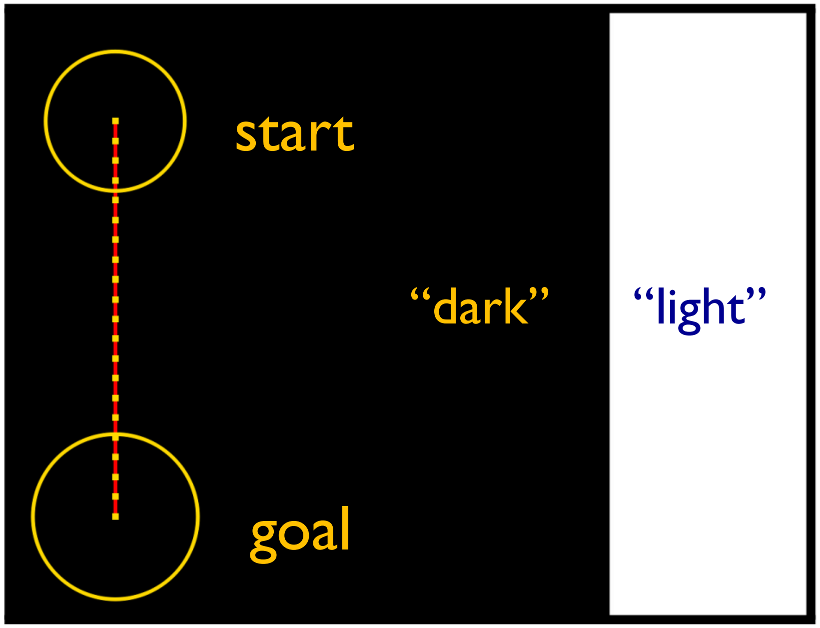




Outline

- Introduction to POMDPs
- Locally Optimal Solutions for POMDPs
  - Trajectory Optimization in (Gaussian) Belief Space
  - Accounting for Discontinuities in Sensing Domains
- Separation Principle

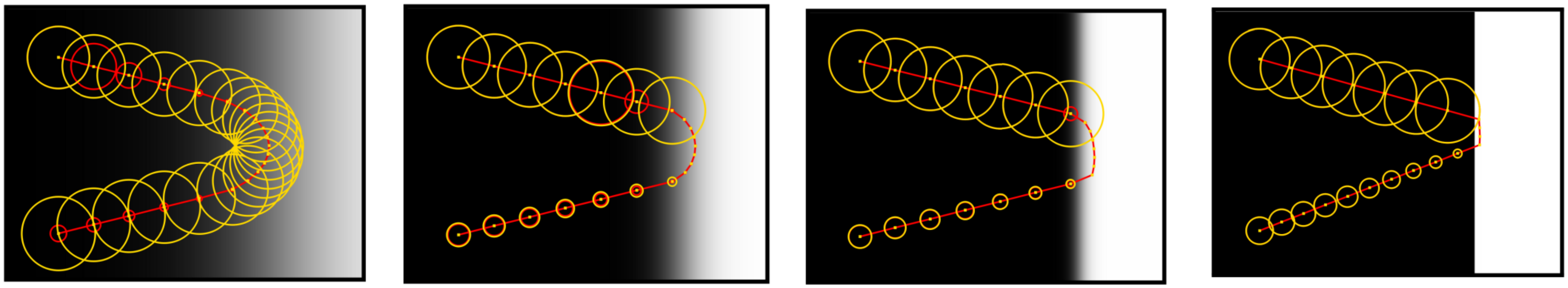
Discontinuities in Sensing Domains



Zero gradient, hence local optimum

Patil et al., under review

Discontinuities in Sensing Domains

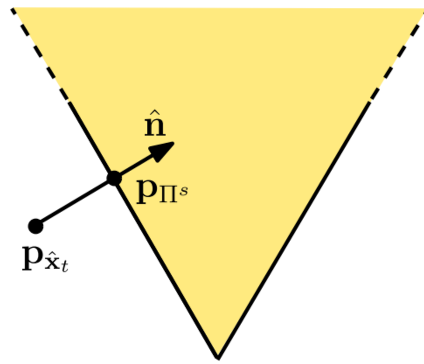


Increasing difficulty

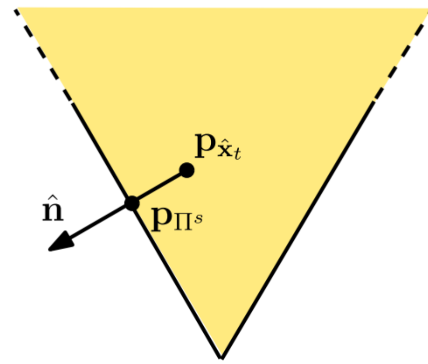
Noise level determined by signed distance to sensing region * homotopy iteration

Signed Distance to Sensing Discontinuity

Field of view (FOV) discontinuity

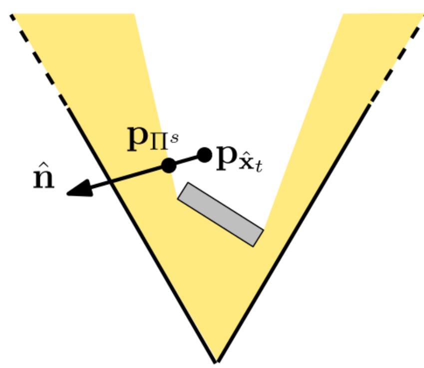


(a) $\mathtt{sd}(\hat{\mathbf{b}}_t, \Pi^s) > 0$
Outside field of view

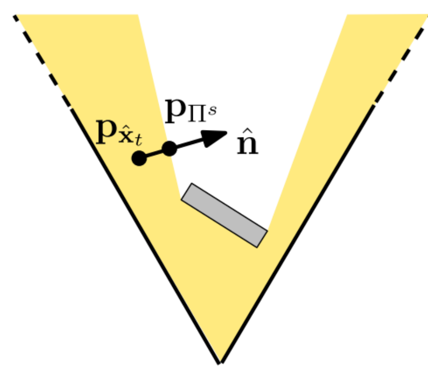


(b) $\mathtt{sd}(\hat{\mathbf{b}}_t, \Pi^s) < 0$
Inside field of view

Occlusion discontinuity

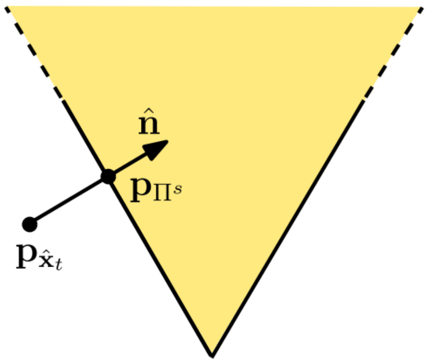


(c) $\mathtt{sd}(\hat{\mathbf{b}}_t, \Pi^s) > 0$
Occluded view

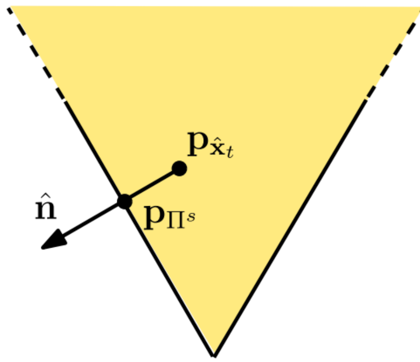


(d) $\mathtt{sd}(\hat{\mathbf{b}}_t, \Pi^s) < 0$
Unoccluded view

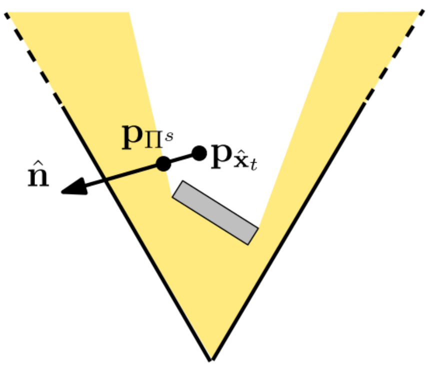
δ_t^s vs. Signed distance



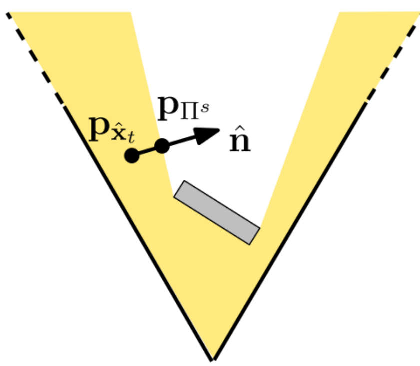
(a) $\mathtt{sd}(\hat{\mathbf{b}}_t, \Pi^s) > 0$
Outside field of view



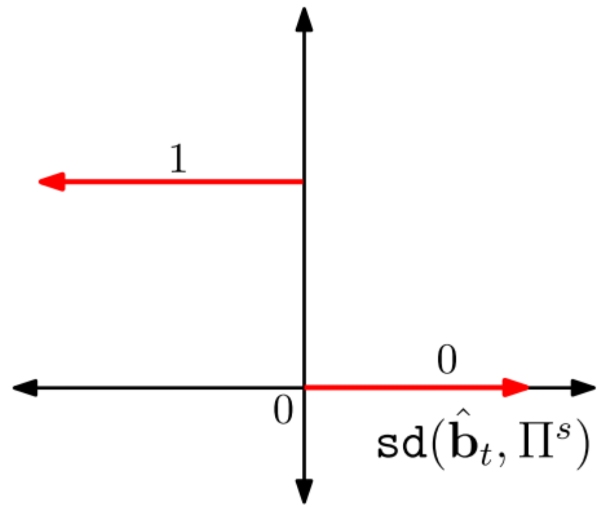
(b) $\mathtt{sd}(\hat{\mathbf{b}}_t, \Pi^s) < 0$
Inside field of view



(c) $\mathtt{sd}(\hat{\mathbf{b}}_t, \Pi^s) > 0$
Occluded view



(d) $\mathtt{sd}(\hat{\mathbf{b}}_t, \Pi^s) < 0$
Unoccluded view



$$\delta_t^s = \chi(\mathtt{sd}(\hat{\mathbf{b}}_t, \Pi^s))$$

Modified Belief Dynamics

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, \mathbf{q}_t), \quad \mathbf{q}_t \sim \mathcal{N}(\mathbf{0}, I),$$
$$\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t, \mathbf{r}_t), \qquad \mathbf{r}_t \sim \mathcal{N}(\mathbf{0}, I),$$

$$\hat{\mathbf{b}}_{t+1} = \mathbf{g}(\hat{\mathbf{b}}_t, \hat{\mathbf{u}}_t) = \begin{bmatrix} \hat{\mathbf{x}}_{t+1} \\ \mathrm{vec}[\sqrt{\Sigma_{t+1}^{-} - K_t H_t \Sigma_{t+1}^{-}}] \end{bmatrix}$$

$$\hat{\mathbf{x}}_{t+1} = \mathbf{f}(\hat{\mathbf{x}}_t, \hat{\mathbf{u}}_t, \mathbf{0}), \qquad \Sigma_{t+1}^{-} = A_t \sqrt{\Sigma_t}(A_t \sqrt{\Sigma_t})^T + Q_t Q_t^T,$$
$$A_t = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\hat{\mathbf{x}}_t, \hat{\mathbf{u}}_t, \mathbf{0}), \qquad Q_t = \frac{\partial \mathbf{f}}{\partial \mathbf{q}}(\hat{\mathbf{x}}_t, \hat{\mathbf{u}}_t, \mathbf{0}),$$
$$H_t = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}(\hat{\mathbf{x}}_{t+1}, \mathbf{0}), \qquad R_t = \frac{\partial \mathbf{h}}{\partial \mathbf{r}}(\hat{\mathbf{x}}_{t+1}, \mathbf{0}),$$
$$\boxed{K_t = \Sigma_{t+1}^{-} H_t^T \Delta_{t+1}(\Delta_{t+1} H_t \Sigma_{t+1}^{-} H_t^T \Delta_{t+1} + R_t R_t^T)^{-1} \Delta_{t+1}.}$$

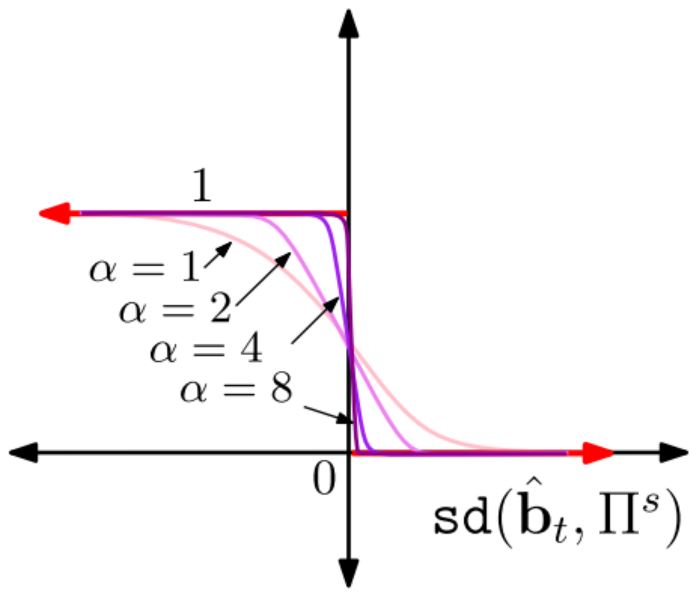
δ_t^s : Binary variable {0,1}

0 -> No measurement

1 -> Measurement

Incorporating δ_t^s in SQP

- Binary non-convex program – difficult to solve
- Solve successively smooth approximations

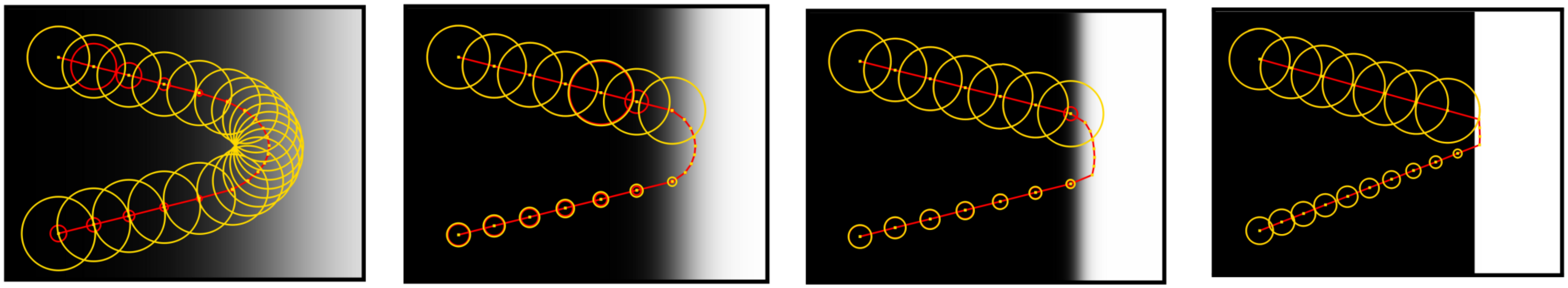


$$\delta_t^s(\alpha) = \tilde{\chi}(\mathrm{sd}(\hat{\mathbf{b}}_t, \Pi^s), \alpha)$$
$$= 1 - \frac{1}{1+\exp(-\alpha \cdot \mathrm{sd}(\hat{\mathbf{b}}_t, \Pi^s))}$$

Algorithm Overview

- While δ not within desired tolerance
  - Solve optimization problem with current value of α
  - Increase α
  - Re-integrate belief trajectory
  - Update δ

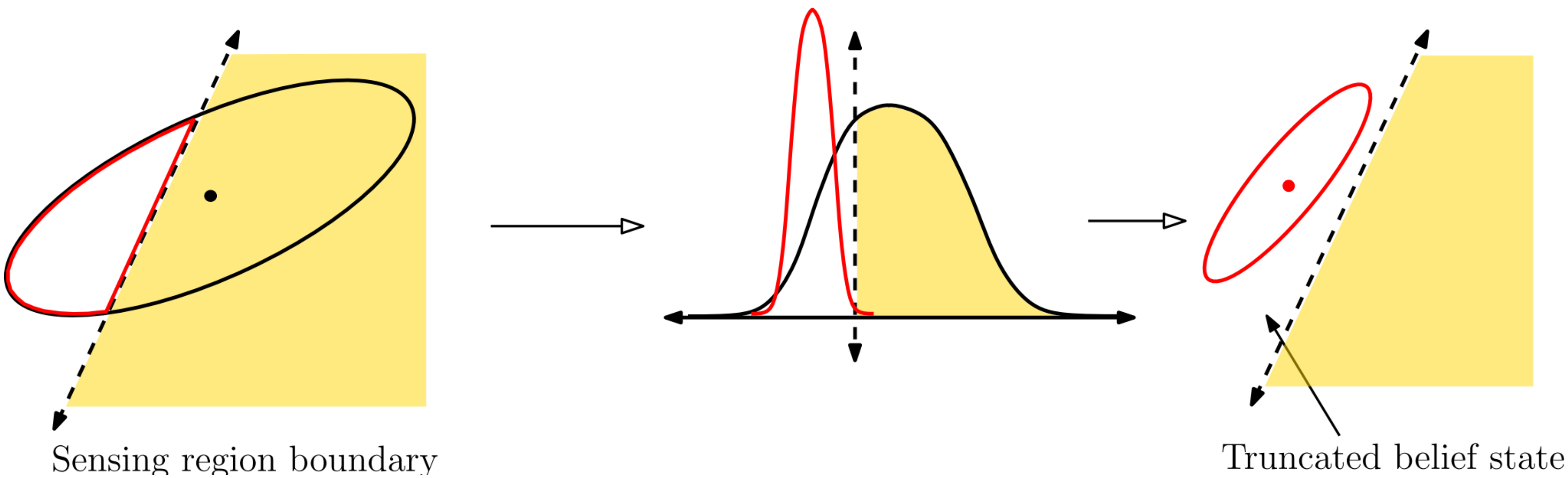
Discontinuities in Sensing Domains



Increasing difficulty →

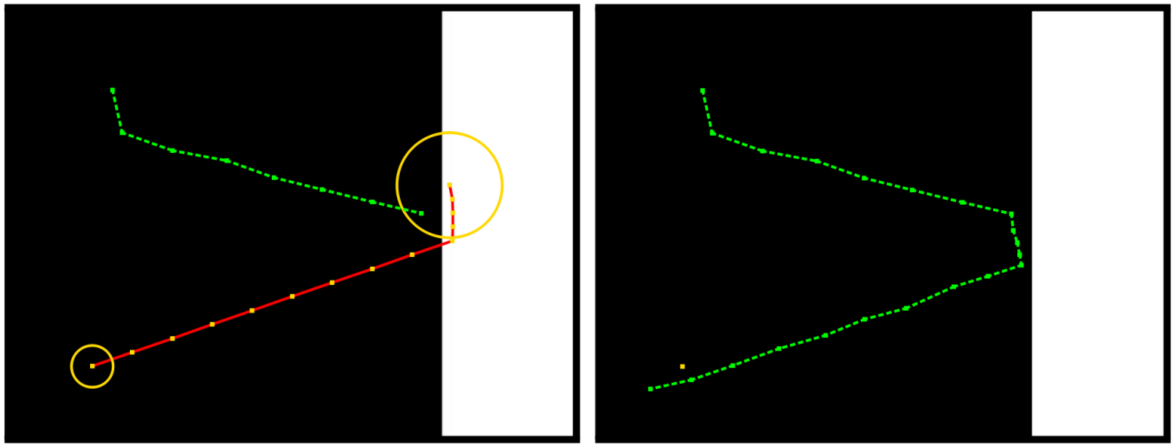
Noise level determined by signed distance to sensing region * homotopy iteration

“No measurement” Belief Update

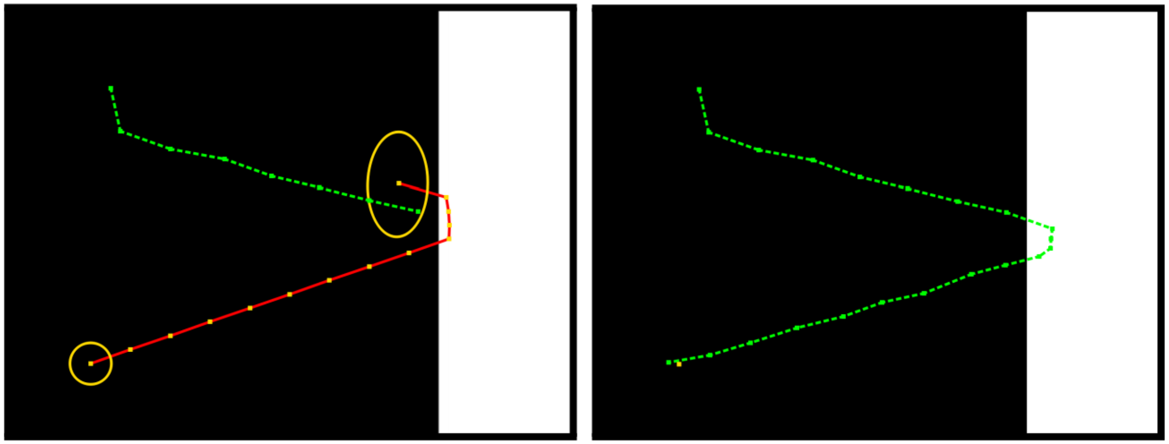


Truncate Gaussian Belief if no measurement obtained

Effect of Truncation

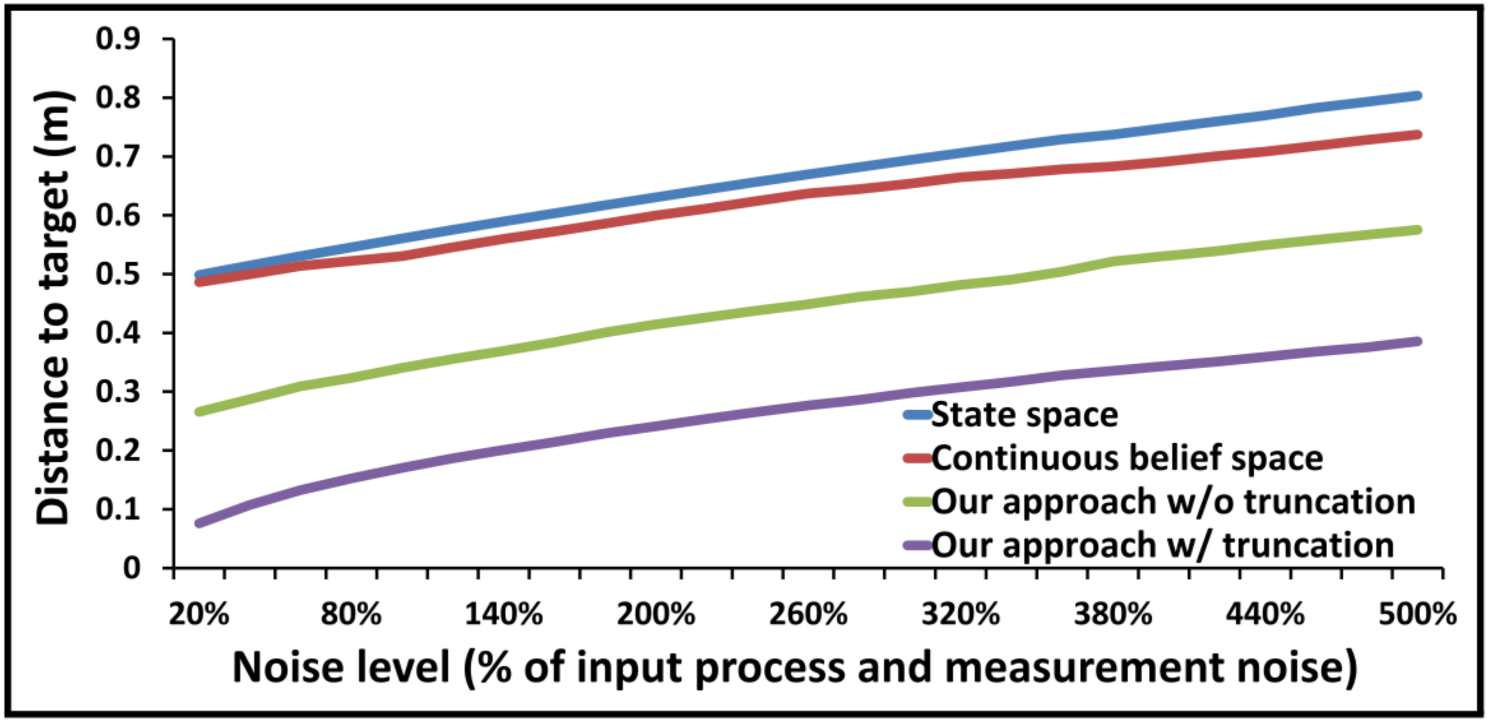


Without "No measurement" Belief Update

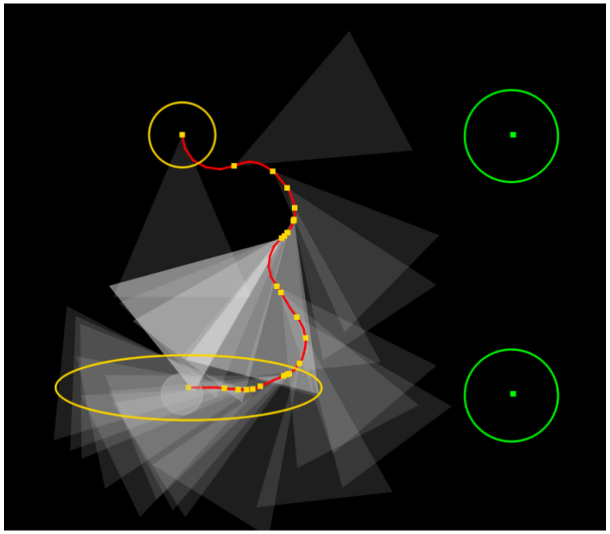


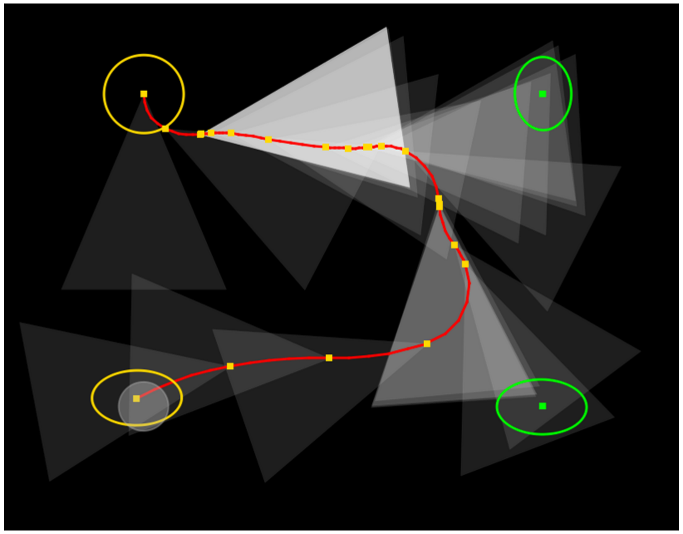
With "No measurement" Belief Update

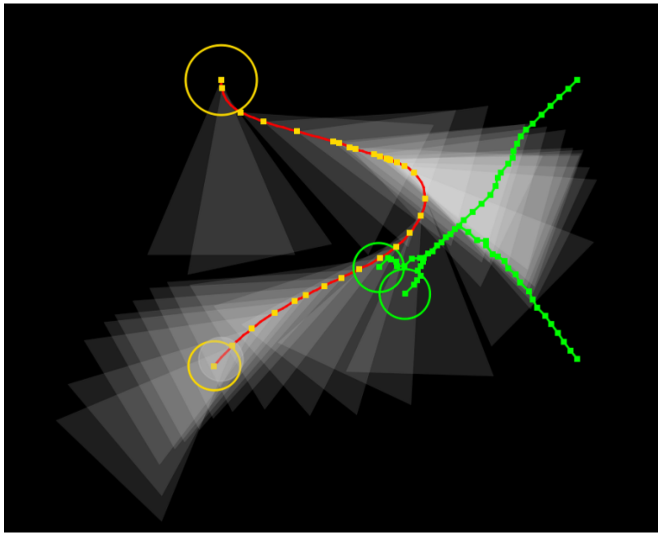
Experiments



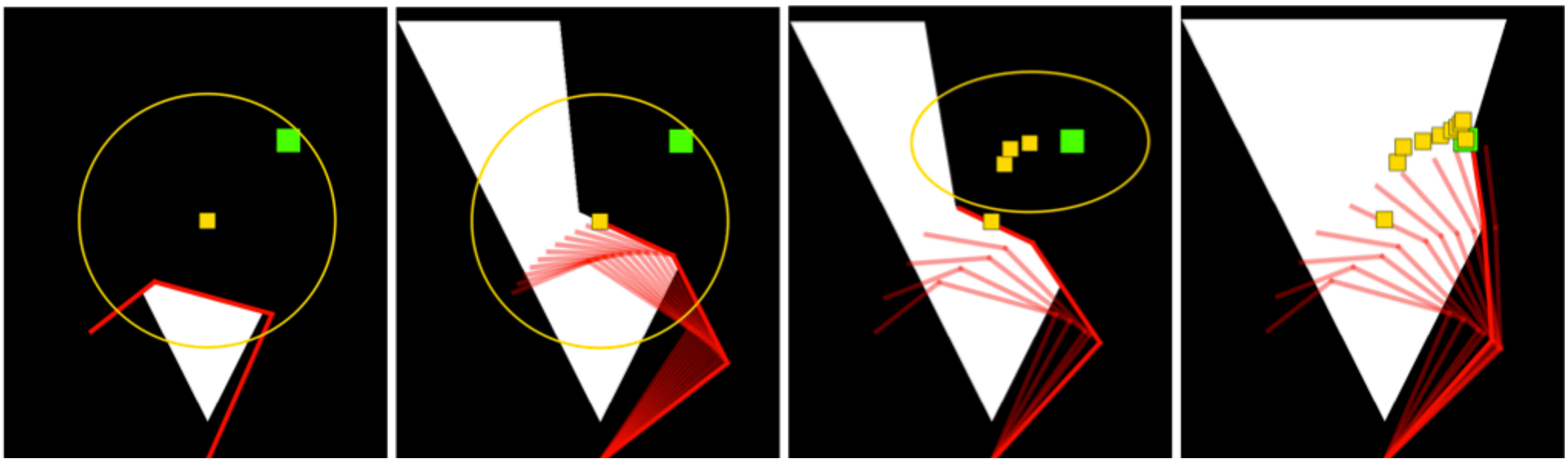
Car and Landmarks (Active Exploration)





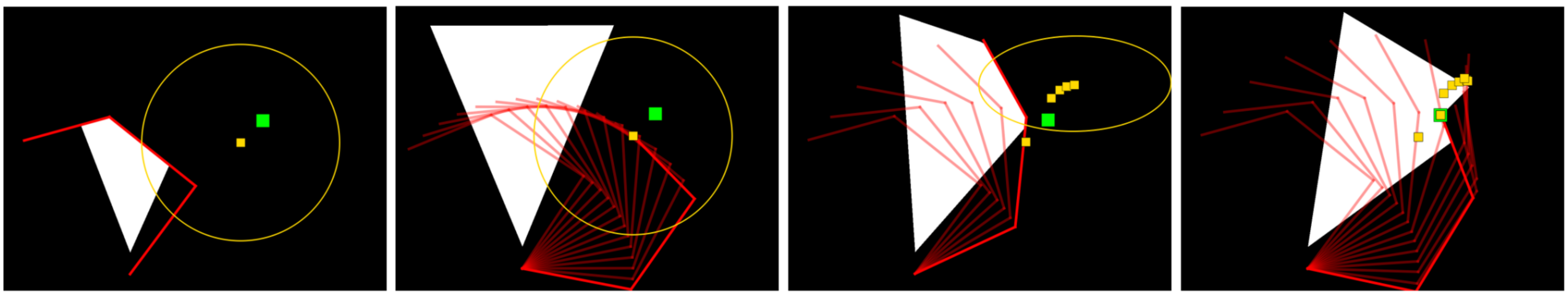


Arm Occluding (Static) Camera



Initial belief | State space plan execution | (way-point) (end) Belief space plan execution

Arm Occluding (Moving) Camera



Initial belief

State space plan execution

(way-point) (end)
Belief space plan execution

Outline

- Introduction to POMDPs
- Locally Optimal Solutions for POMDPs
  - Trajectory Optimization in (Gaussian) Belief Space
  - Accounting for Discontinuities in Sensing Domains
- Separation Principle

Separation Principle

- Assume:
$$x_{t+1} = Ax_t + Bu_t + w_t \qquad w_t \sim \mathcal{N}(0, Q_t)$$
$$z_t = Cx_t + v_t \qquad v_t \sim \mathcal{N}(0, R_t)$$

- Goal:
$$\text{minimize E}\left[\sum_{t=0}^{H} u_t^\top U_t u_t + x_t^\top X_t x_t\right]$$

- Then, optimal control policy consists of:

1. Offline/Ahead of time: Run LQR to find optimal control policy for fully observed case, which gives sequence of feedback matrices $K_1, K_2, \ldots$

2. Online: Run Kalman filter to estimate state, and apply control
$$u_t = K_t \mu_{t|0:t}$$

Extensions

- Current research directions
  - Fast! belief space planning
  - Multi-modal belief spaces
  - Physical experiments with the Raven surgical robot

Recap

- POMDP = MDP but sensory measurements instead of exact state knowledge
- Locally optimal solutions in Gaussian belief spaces
  - Augmented state vector (mean, covariance)
  - Trajectory optimization
- Sigma Hulls for probabilistic collision avoidance
- Homotopy methods for dealing with discontinuities in sensing domains