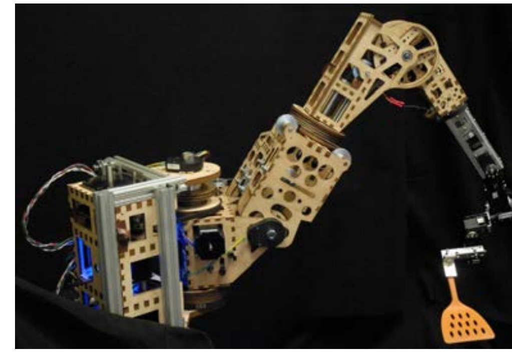


Sigma Hulls for Gaussian Belief Space Planning for Imprecise Articulated Robots amid Obstacles

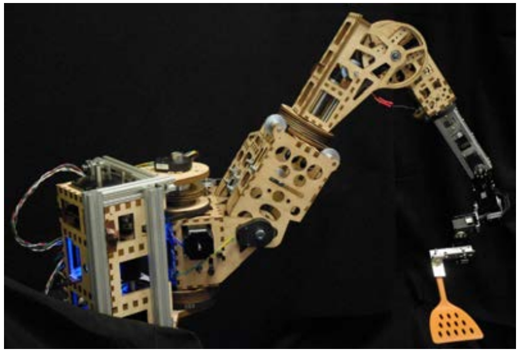
Alex Lee, Yan Duan, Sachin Patil, John Schulman, Zoe McCarthy, Jur van den Berg*, Ken Goldberg and Pieter Abbeel

UC Berkeley, *University of Utah

Motivation

Facilitate reliable operation of cost-effective robots that use:

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

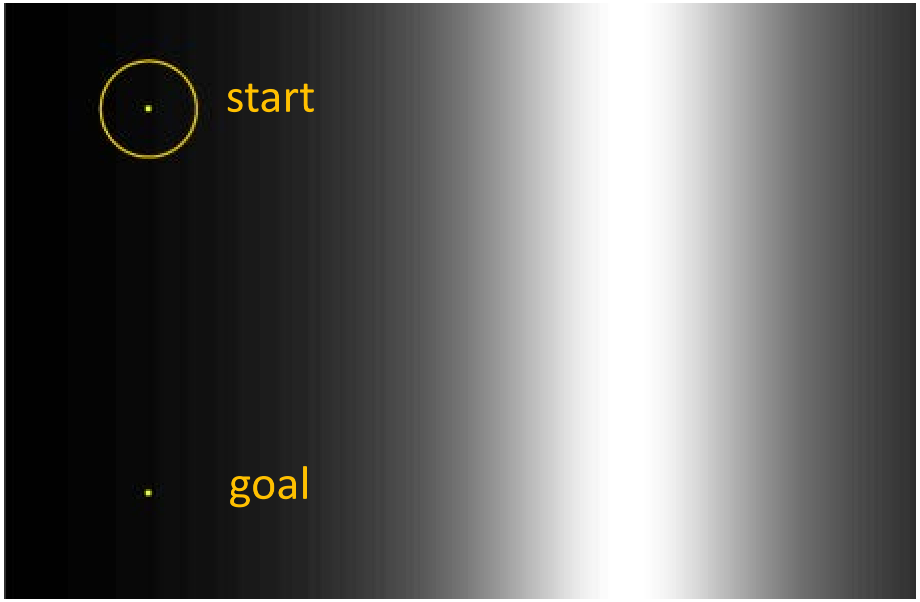


Presenter: Alex Lee (UC Berkeley)

Prior Work on Gaussian Belief Space Planning

- Planning under motion and sensing uncertainty is a POMDP in general
  - Intractable in general
  - Compute locally optimal solutions
- Bry et al (ICRA 2011), Li et al (IJC 2007), van den Berg et al (IJRR 2011), van den Berg et al (IJRR 2012), Platt et al (RSS 2010)

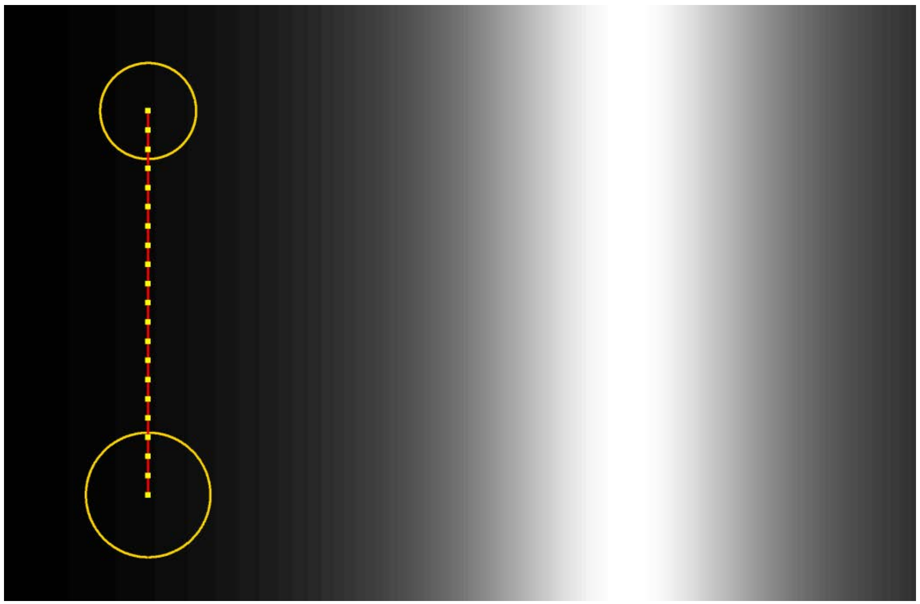
Gaussian Belief Space Planning



Problem Setup

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

Gaussian Belief Space Planning

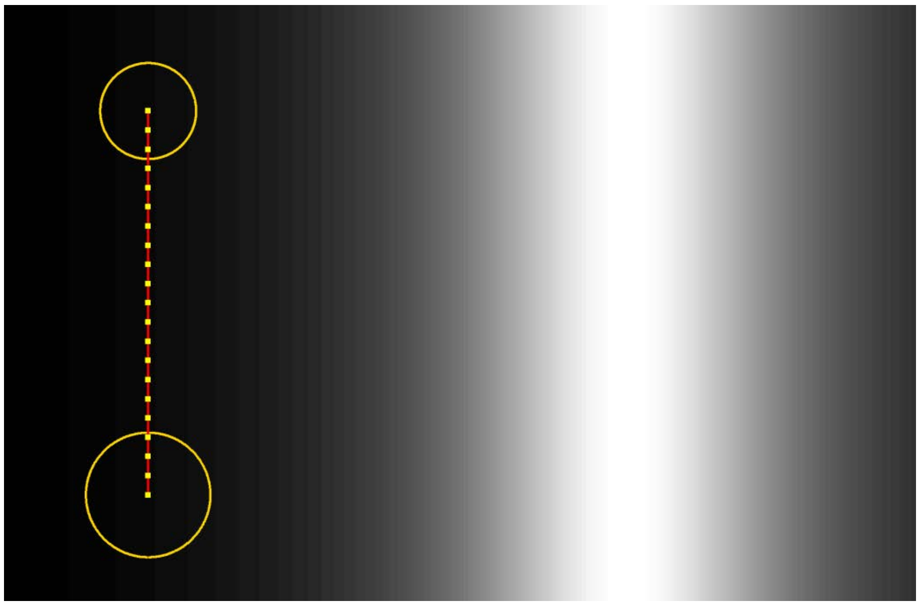


State space plan

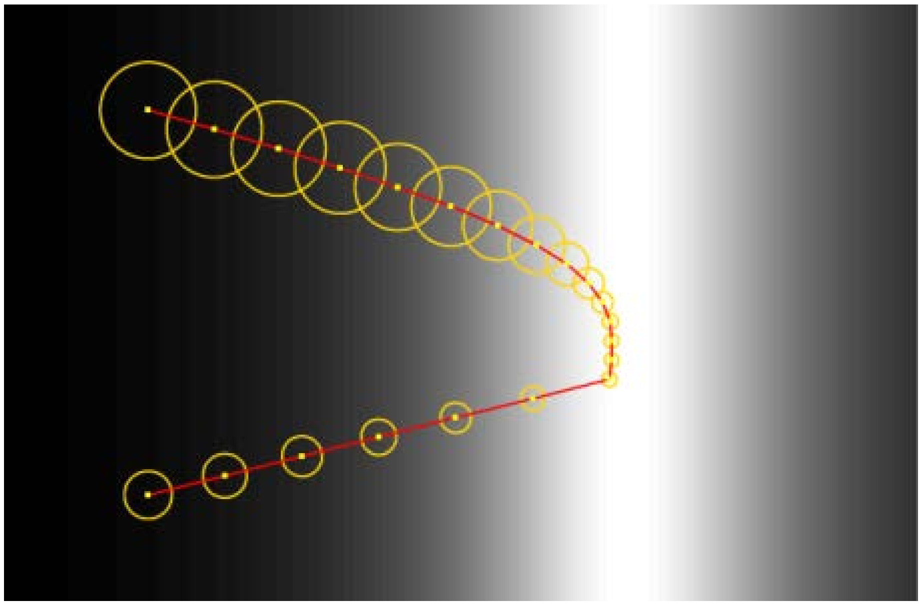
[Example from Platt, Tedrake, Kaelbling, Lozano-Perez, 2010]

Presenter: Alex Lee (UC Berkeley)

Gaussian Belief Space Planning



State space plan



Belief space plan

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

Presenter: Alex Lee (UC Berkeley)

Gaussian Belief Space Planning using Trajectory Optimization

- Gaussian belief state in joint space: $b_t = \begin{bmatrix} \mu_t \\ \sqrt{\Sigma_t} \end{bmatrix}$

  mean → μ_t

  square root of covariance → $\sqrt{\Sigma_t}$

Presenter: Alex Lee (UC Berkeley)

Gaussian Belief Space Planning using Trajectory Optimization

- Gaussian belief state in joint space: $b_t = \begin{bmatrix} \mu_t \\ \sqrt{\Sigma_t} \end{bmatrix}$ ← mean; ← square root of covariance

- Optimization problem:

$$\min C(b_0, \dots, b_T, u_0, \dots, u_{T-1})$$
$$\text{s.t. } \forall\, t = 1, \dots, T$$

$b_{t+1} = \text{belief\_dynamics}(b_t, u_t)$	Unscented Kalman Filter dynamics
$\mu_T = \text{goal}$	Reach desired end-effector pose
$u_t \in U$	Control inputs are feasible

Presenter: Alex Lee (UC Berkeley)

Gaussian Belief Space Planning using Trajectory Optimization

- Gaussian belief state in joint space: $b_t = \begin{bmatrix} \mu_t \\ \sqrt{\Sigma_t} \end{bmatrix}$ ← mean; ← square root of covariance

- Optimization problem:

$$\min C(b_0, \ldots, b_T, u_0, \ldots, u_{T-1})$$
$$\text{s.t. } \forall\ t = 1, \ldots, T$$

$$b_{t+1} = \text{belief\_dynamics}(b_t, u_t)$$ Unscented Kalman Filter dynamics

$$\mu_T = \text{goal}$$ Reach desired end-effector pose

$$u_t \in U$$ Control inputs are feasible

- Non-convex optimization – Can be solved using sequential quadratic programming (SQP)

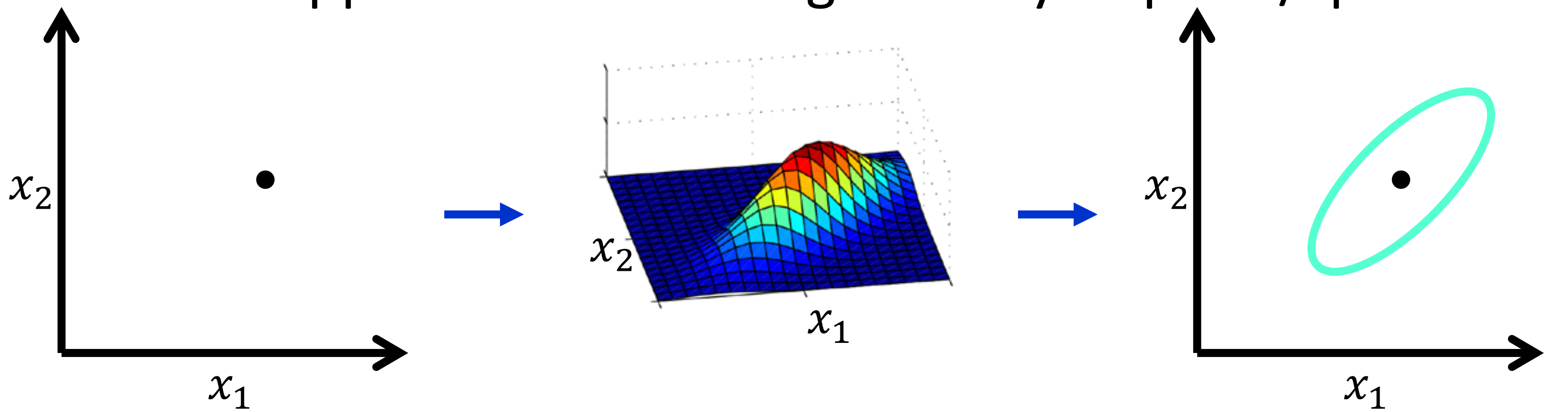
Presenter: Alex Lee (UC Berkeley)

Prior Work on Gaussian Belief Space Planning

- Want to include probabilistic collision avoidance constraints

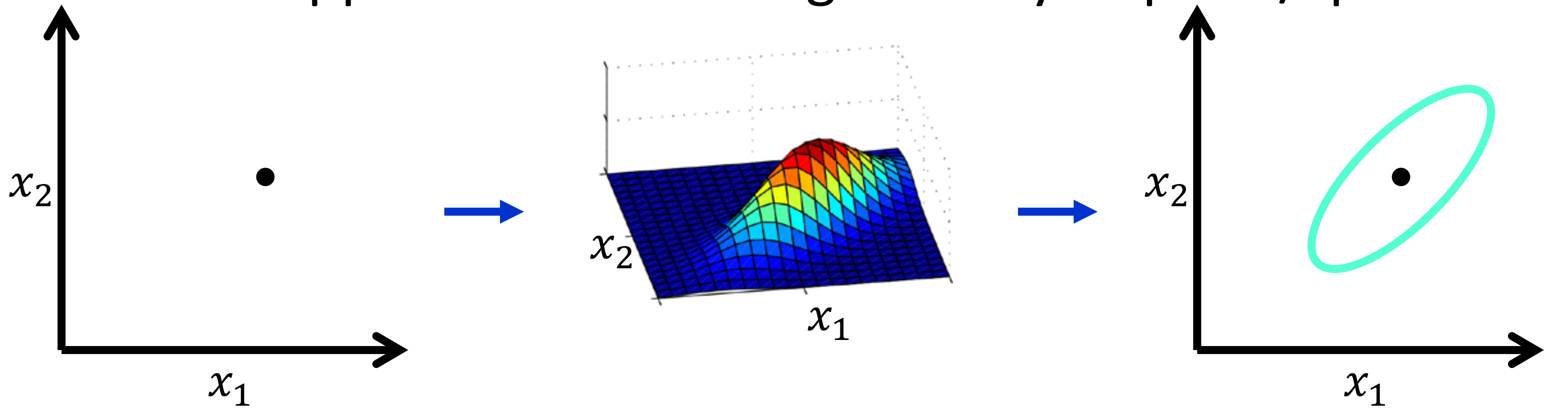
Prior Work on Gaussian Belief Space Planning

- Want to include probabilistic collision avoidance constraints
- Prior work approximates robot geometry as point/spheres

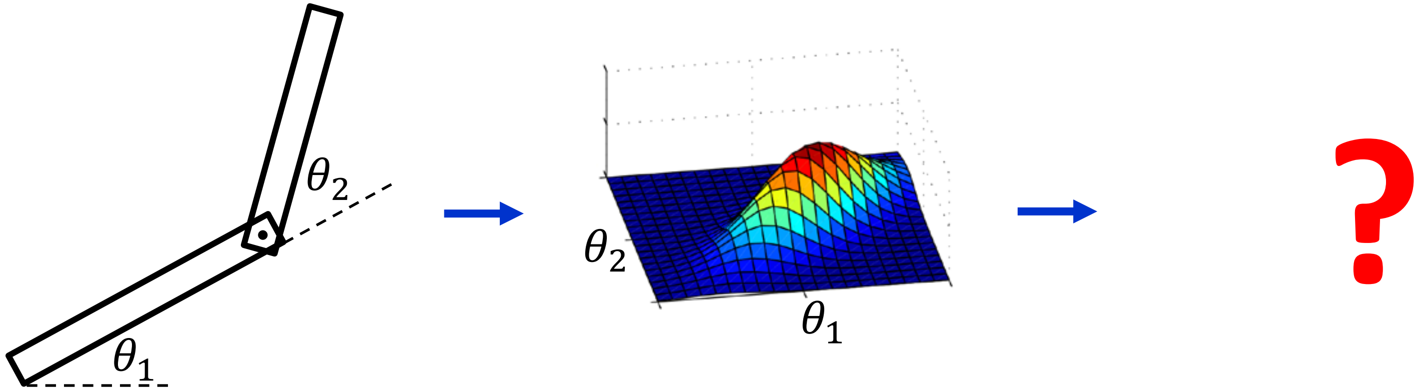


Prior Work on Gaussian Belief Space Planning

- Want to include probabilistic collision avoidance constraints
- Prior work approximates robot geometry as point/spheres



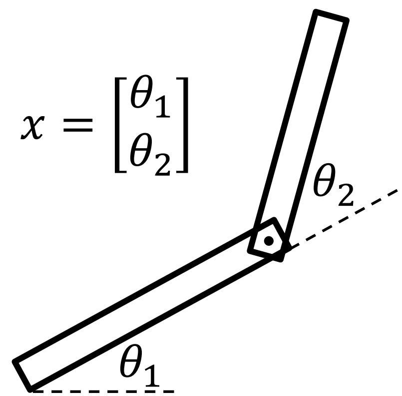
- How do you formulate the constraint for a robot link?

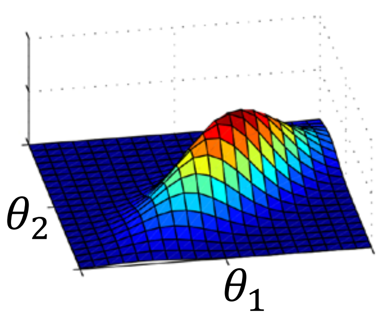


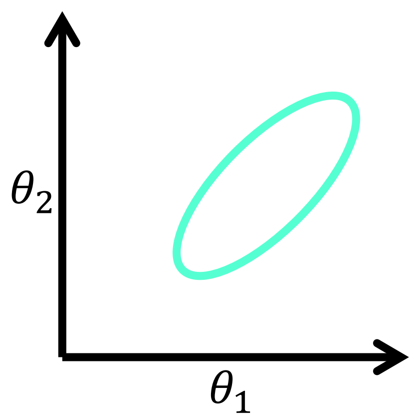
Presenter: Alex Lee (UC Berkeley)

Main Contribution: Incorporation of Collision Avoidance Constraints under Uncertainty through Sigma Hulls

$$x = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$$



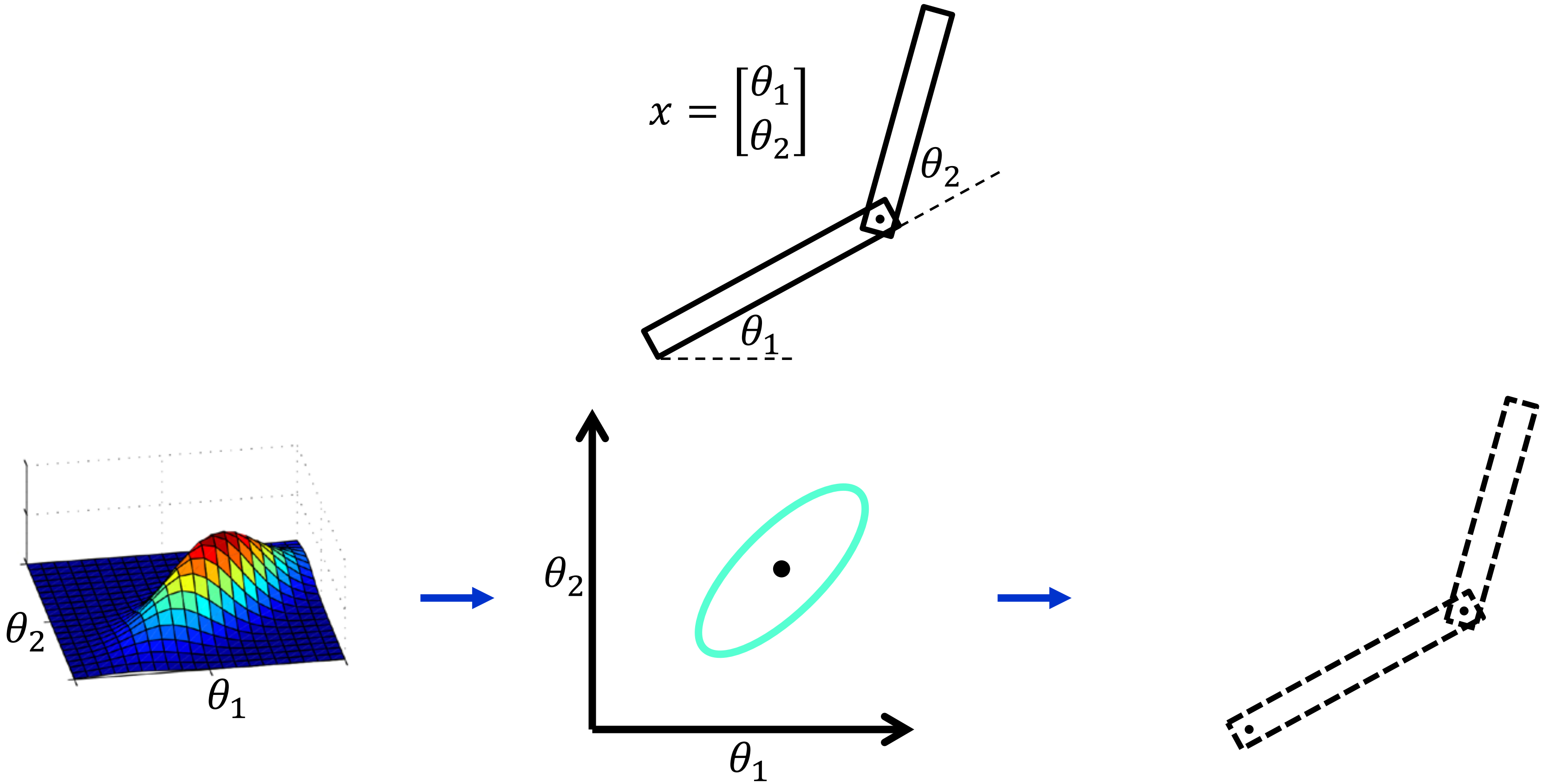




$$\mathcal{X} = [x \quad x \quad x \quad x \quad x] + \lambda[0 \quad \sqrt{\Sigma} \quad -\sqrt{\Sigma}]$$

Presenter: Alex Lee (UC Berkeley)

Main Contribution: Incorporation of Collision Avoidance Constraints under Uncertainty through Sigma Hulls

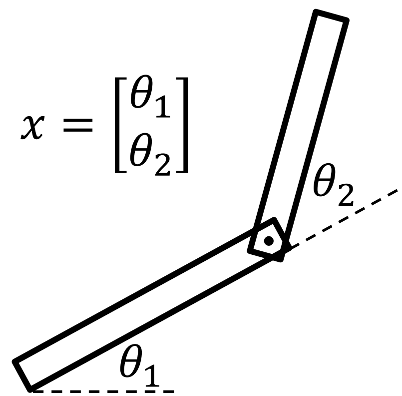


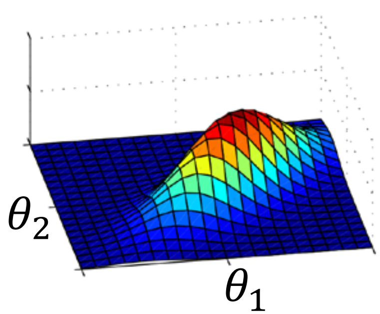
$$\mathcal{X} = [x \quad x \quad x \quad x \quad x] + \lambda[0 \quad \sqrt{\Sigma} \quad -\sqrt{\Sigma}]$$

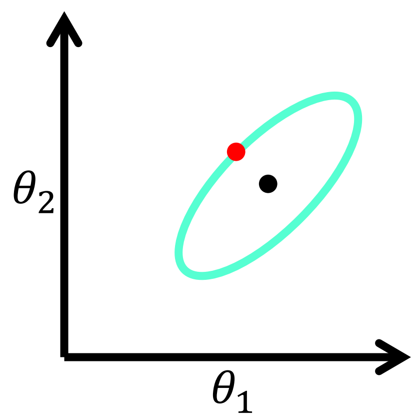
Presenter: Alex Lee (UC Berkeley)

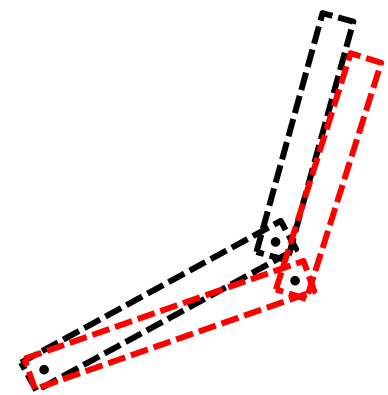
Main Contribution: Incorporation of Collision Avoidance Constraints under Uncertainty through Sigma Hulls

$$x = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$$





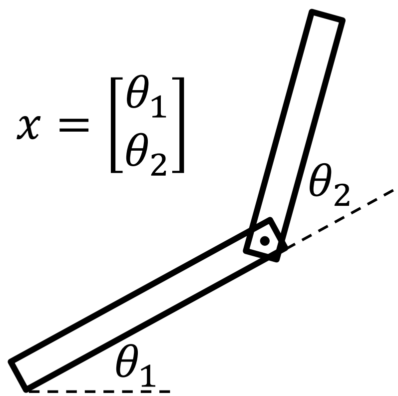


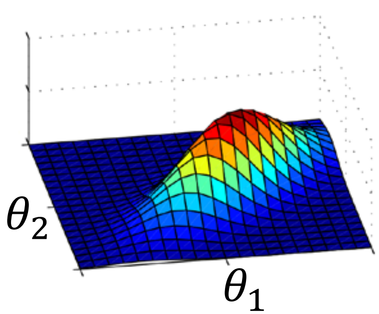


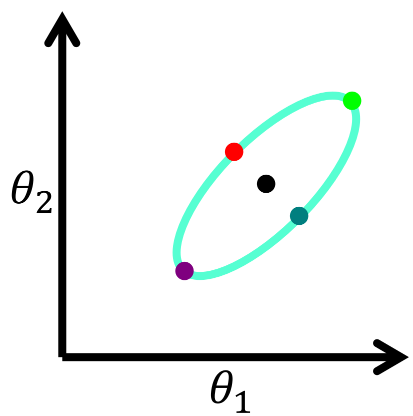
$$\mathcal{X} = [x \quad x \quad x \quad x \quad x] + \lambda[0 \quad \sqrt{\Sigma} \quad -\sqrt{\Sigma}]$$

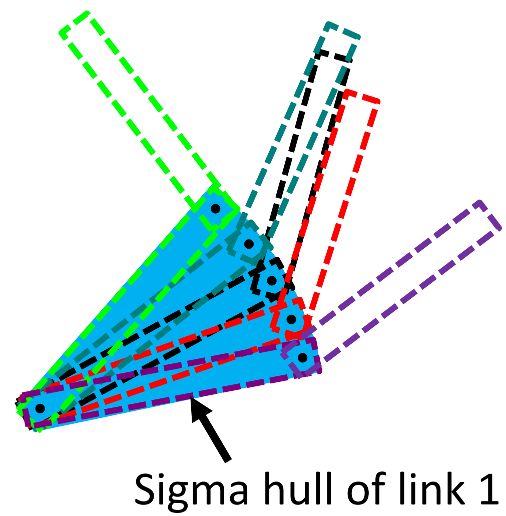
Presenter: Alex Lee (UC Berkeley)

Main Contribution: Incorporation of Collision Avoidance Constraints under Uncertainty through Sigma Hulls







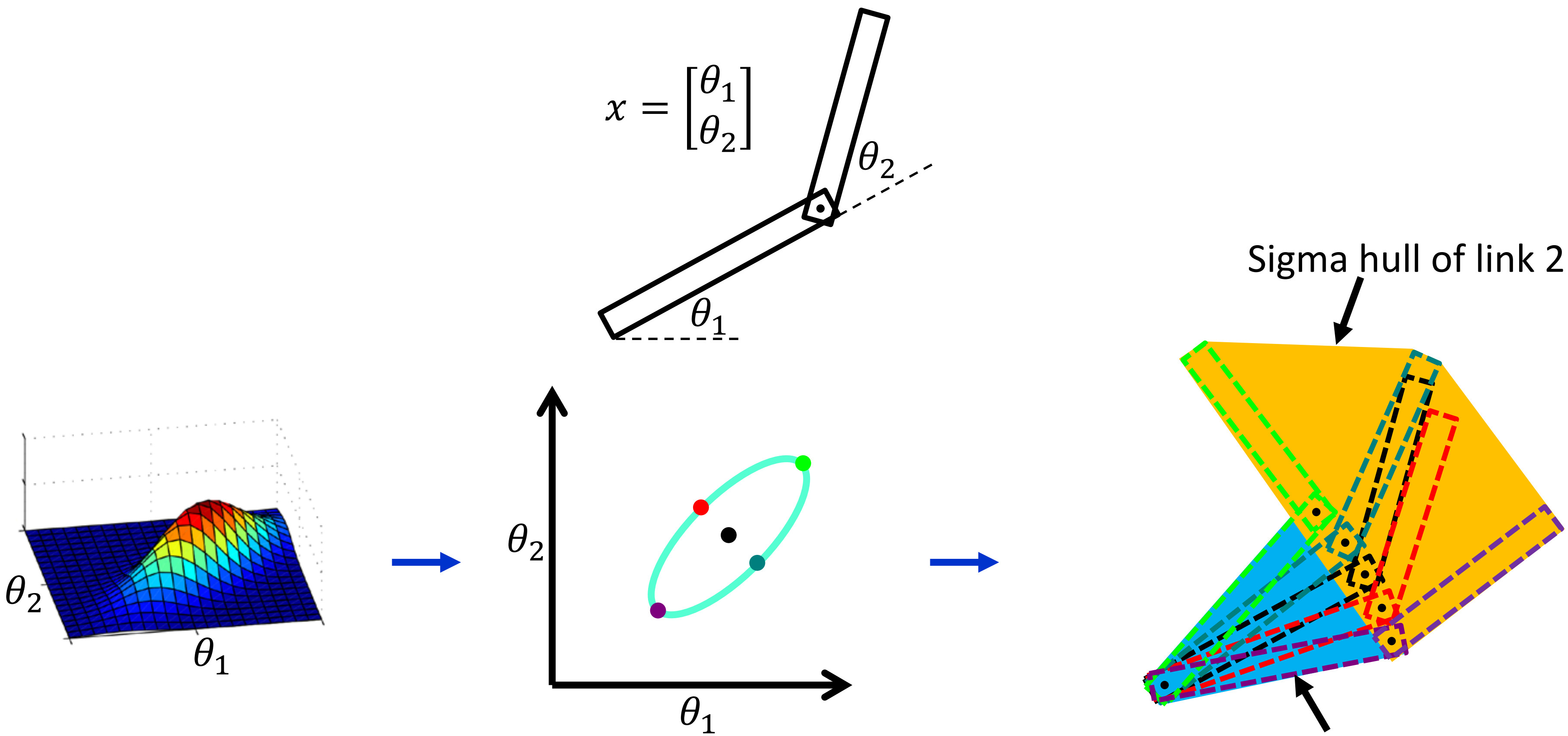


$$\mathcal{X} = [x \quad x \quad x \quad x \quad x] + \lambda[0 \quad \sqrt{\Sigma} \quad -\sqrt{\Sigma}]$$

Presenter: Alex Lee (UC Berkeley)

Main Contribution: Incorporation of Collision Avoidance Constraints under Uncertainty through Sigma Hulls

Sigma hull: Convex hull of a robot link transformed (in joint space) according to sigma points



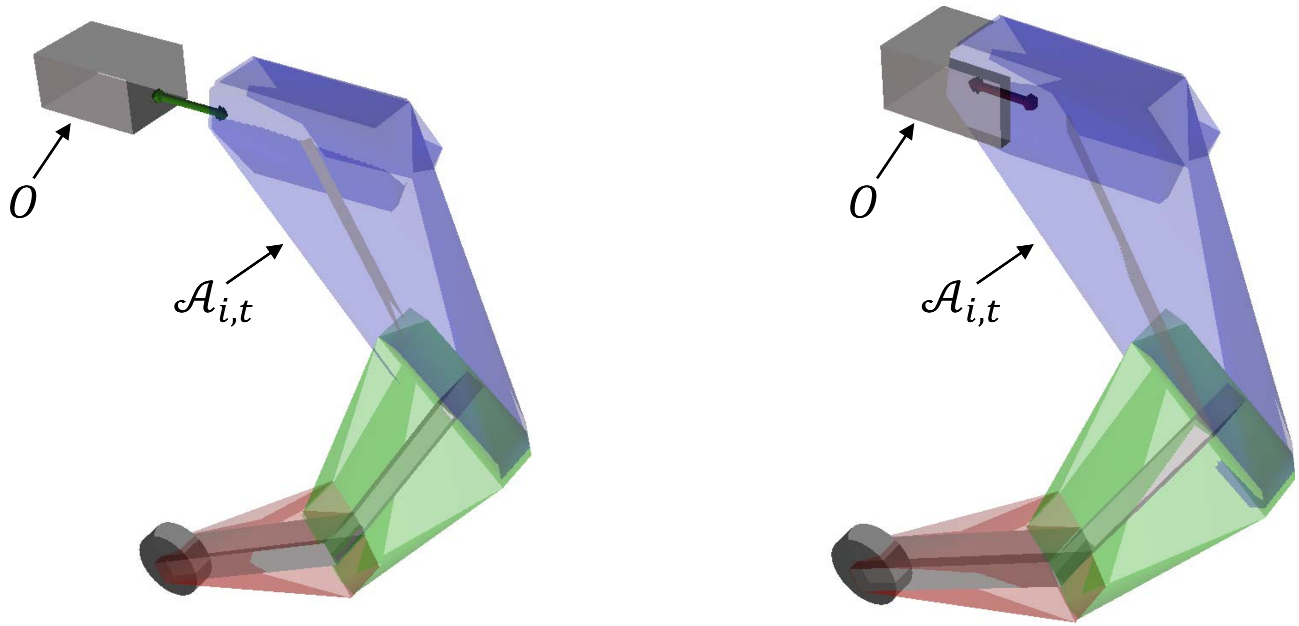
$$\mathcal{X} = [x \quad x \quad x \quad x \quad x] + \lambda[0 \quad \sqrt{\Sigma} \quad -\sqrt{\Sigma}]$$

Presenter: Alex Lee (UC Berkeley)

Signed Distance

Consider signed distance between obstacle O and sigma hull $\mathcal{A}_{i,t}$ of the i-th link at time t

$$\mathcal{A}_{i,t} = \text{sigmahull}(\text{link}_{i,t})$$



Collision Avoidance Constraint: Signed Distance

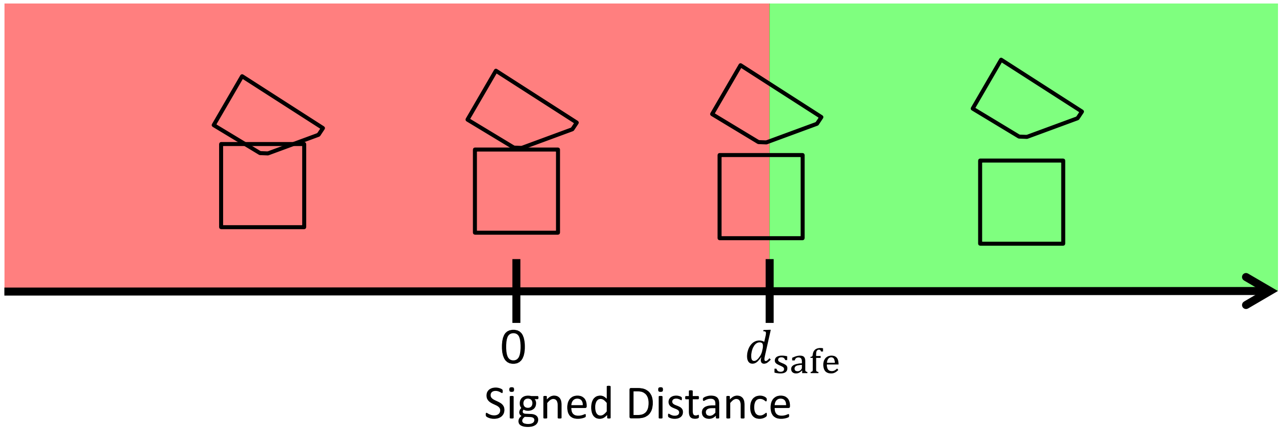
- Use convex-convex collision detection (GJK and EPA algorithm)
  - Computes signed distance of convex hull efficiently

Collision Avoidance Constraint: Signed Distance

- Use convex-convex collision detection (GJK and EPA algorithm)
  - Computes signed distance of convex hull efficiently
- Sigma hulls should stay at least distance d_{safe} from other objects

  $\forall$ times t, $\forall$ links i, $\forall$ obstacles O

$$\text{sd}\left(\mathcal{A}_{i,t}, O\right) \geq d_{\text{safe}}$$



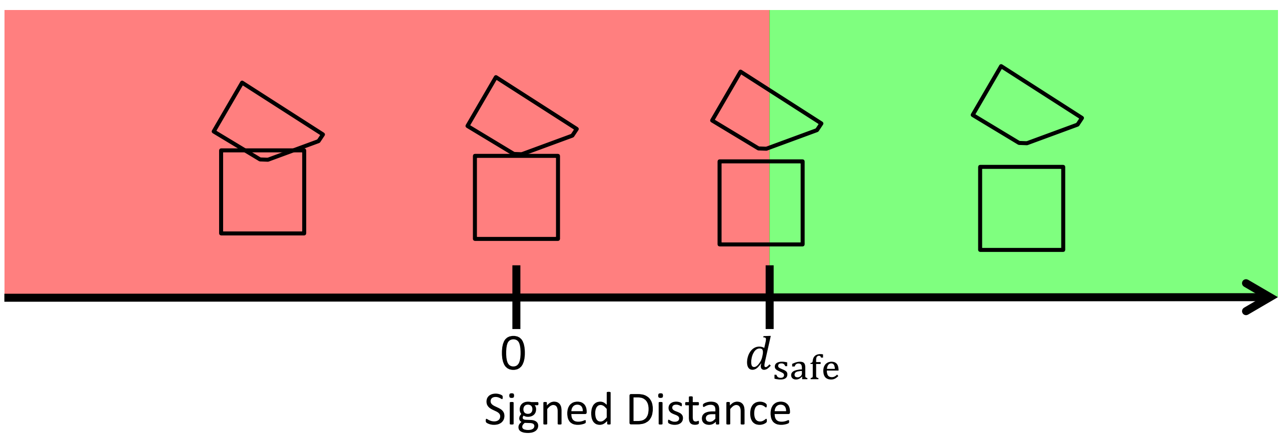
Collision Avoidance Constraint: Signed Distance

- Use convex-convex collision detection (GJK and EPA algorithm)
  - Computes signed distance of convex hull efficiently
- Sigma hulls should stay at least distance d_{safe} from other objects

$$\forall \text{ times } t, \forall \text{ links } i, \forall \text{ obstacles } O$$

$$\text{sd}(\mathcal{A}_{i,t}, O) \geq d_{\text{safe}}$$

Non-convex!

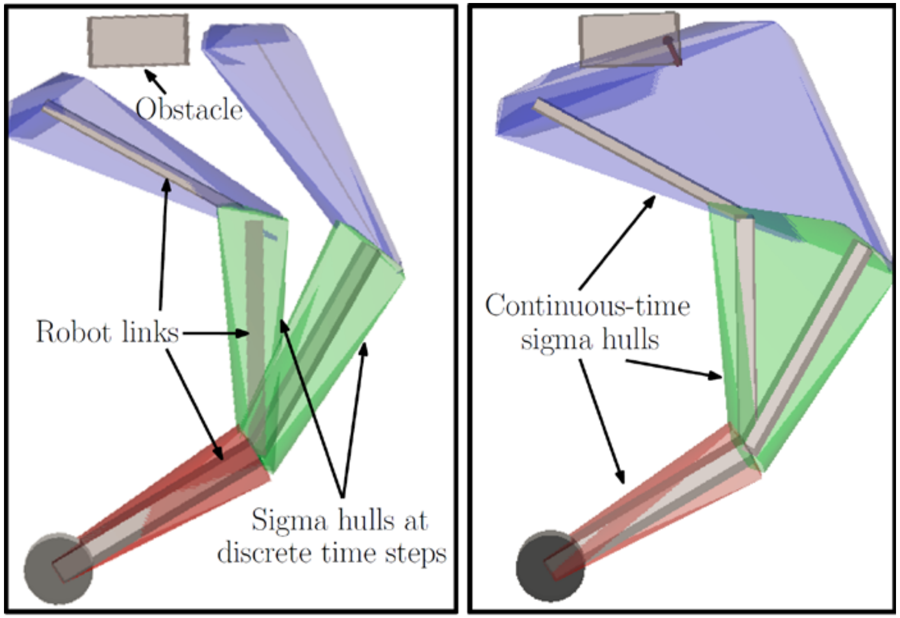


- Use analytical gradients for the signed distance

Presenter: Alex Lee (UC Berkeley)

Continuous Collision Avoidance Constraint

- Discrete collision avoidance can lead to trajectories that collide with obstacles in between time steps



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

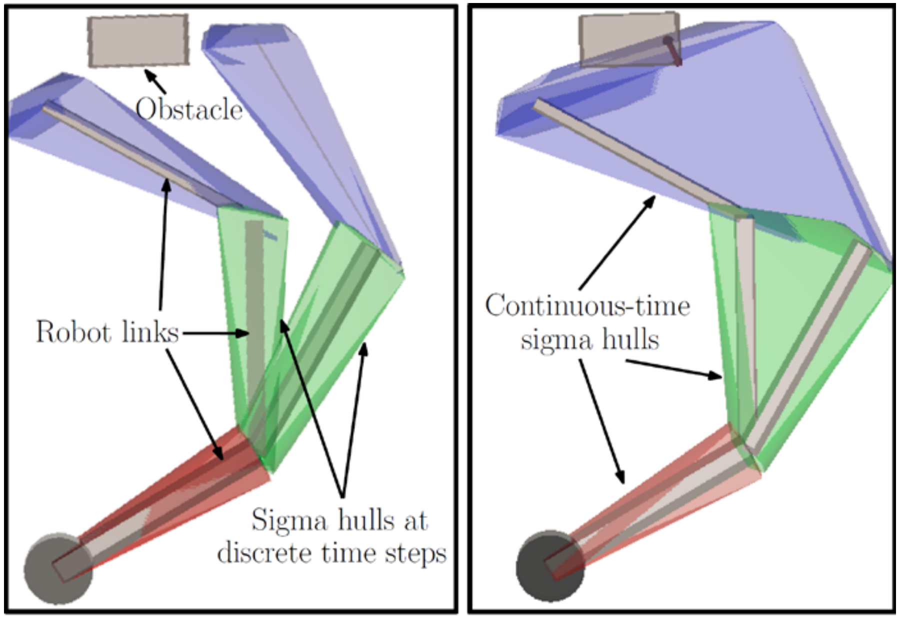
(b) Obstacle overlaps with continuous-time sigma hulls

Presenter: Alex Lee (UC Berkeley)

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

$$\mathrm{sd}\big(\mathrm{convhull}(\mathcal{A}_{i,t}, \mathcal{A}_{i,t+1}), O\big) \geq d_{\mathrm{safe}} \qquad \forall\, t, i, O$$



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

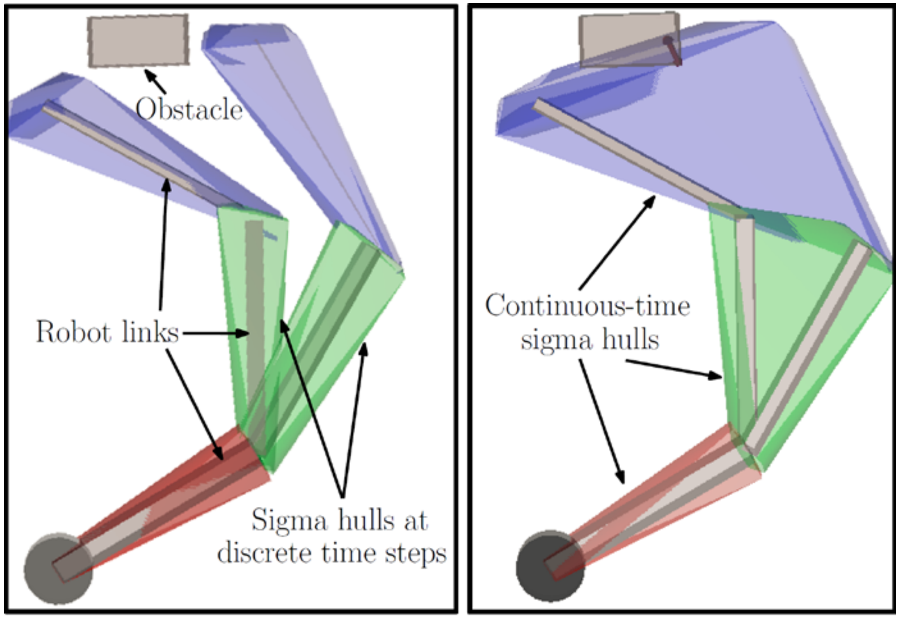
Presenter: Alex Lee (UC Berkeley)

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

$$\mathrm{sd}\big(\mathrm{convhull}(\mathcal{A}_{i,t}, \mathcal{A}_{i,t+1}), O\big) \geq d_{\text{safe}} \qquad \forall\, t, i, O$$

- 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

Presenter: Alex Lee (UC Berkeley)

Gaussian Belief Space Planning using Trajectory Optimization

- Gaussian belief state in joint space: $b_t = \begin{bmatrix} x_t \\ \sqrt{\Sigma_t} \end{bmatrix}$ ← mean; ← square root of covariance

- Optimization problem:

$$\min C(b_0, \ldots, b_T, u_0, \ldots, u_{T-1})$$
$$\text{s.t. } \forall t = 1, \ldots, T$$

$b_{t+1} = \text{belief\_dynamics}(b_t, u_t)$	Unscented Kalman Filter dynamics
$\text{pose}(x_T) = \text{target\_pose}$	Reach desired end-effector pose
$u_t \in U$	Control inputs are feasible
?	Probabilistic collision avoidance

- Non-convex optimization – Can be solved using sequential quadratic programming (SQP)

Gaussian Belief Space Planning using Trajectory Optimization

- Gaussian belief state in joint space: $b_t = \begin{bmatrix} x_t \\ \sqrt{\Sigma_t} \end{bmatrix}$ ← mean; ← square root of covariance
- Optimization problem:

$$\min C(b_0, \dots, b_T, u_0, \dots, u_{T-1})$$

$$\text{s.t. } \forall t = 1, \dots, T$$

$b_{t+1} = \text{belief\_dynamics}(b_t, u_t)$	Unscented Kalman Filter dynamics
$\text{pose}(x_T) = \text{target\_pose}$	Reach desired end-effector pose
$u_t \in U$	Control inputs are feasible
$\text{sd}(\text{sigma\_hull}_i(b_t), O) \geq d_{\text{safe}} \ \forall\, i, O$	Probabilistic collision avoidance

- Non-convex optimization – Can be solved using sequential quadratic programming (SQP)

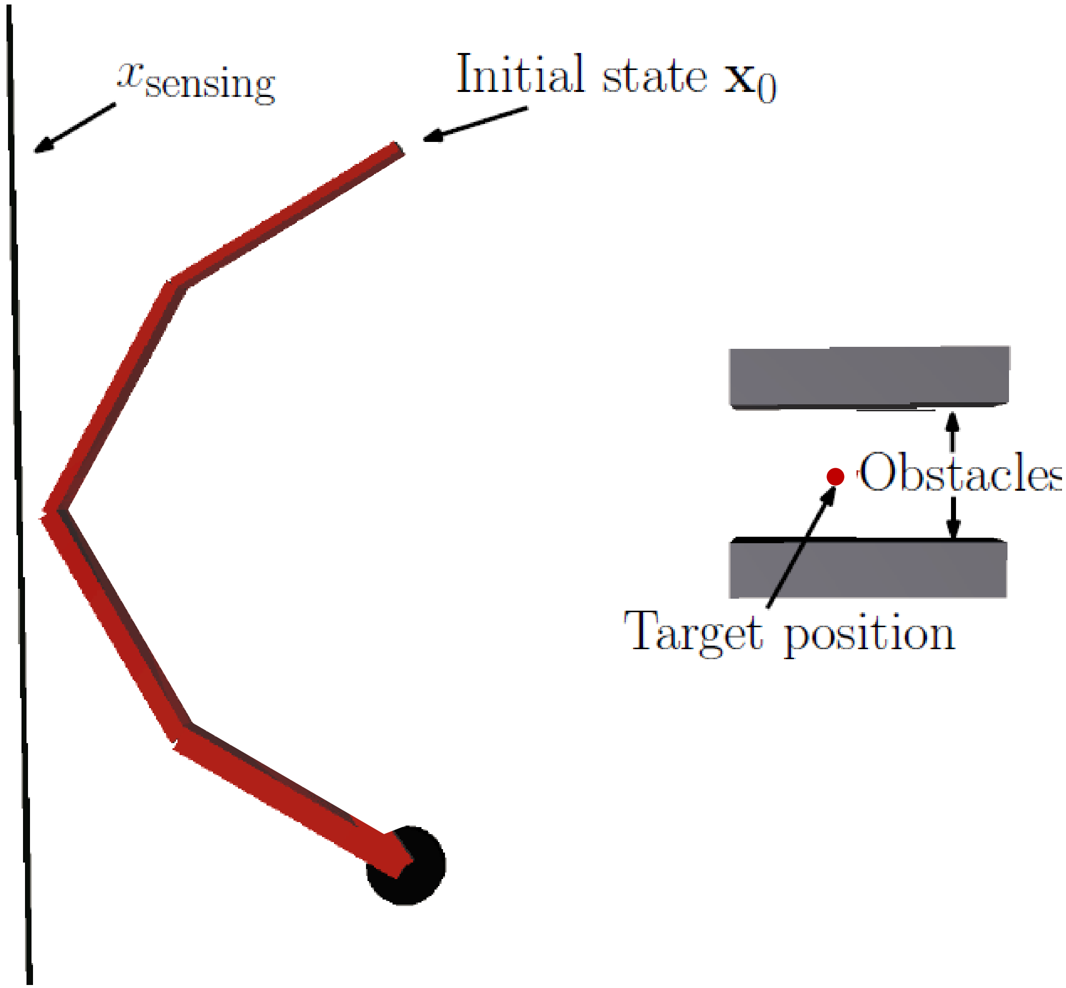
Presenter: Alex Lee (UC Berkeley)

Model Predictive Control (MPC)

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

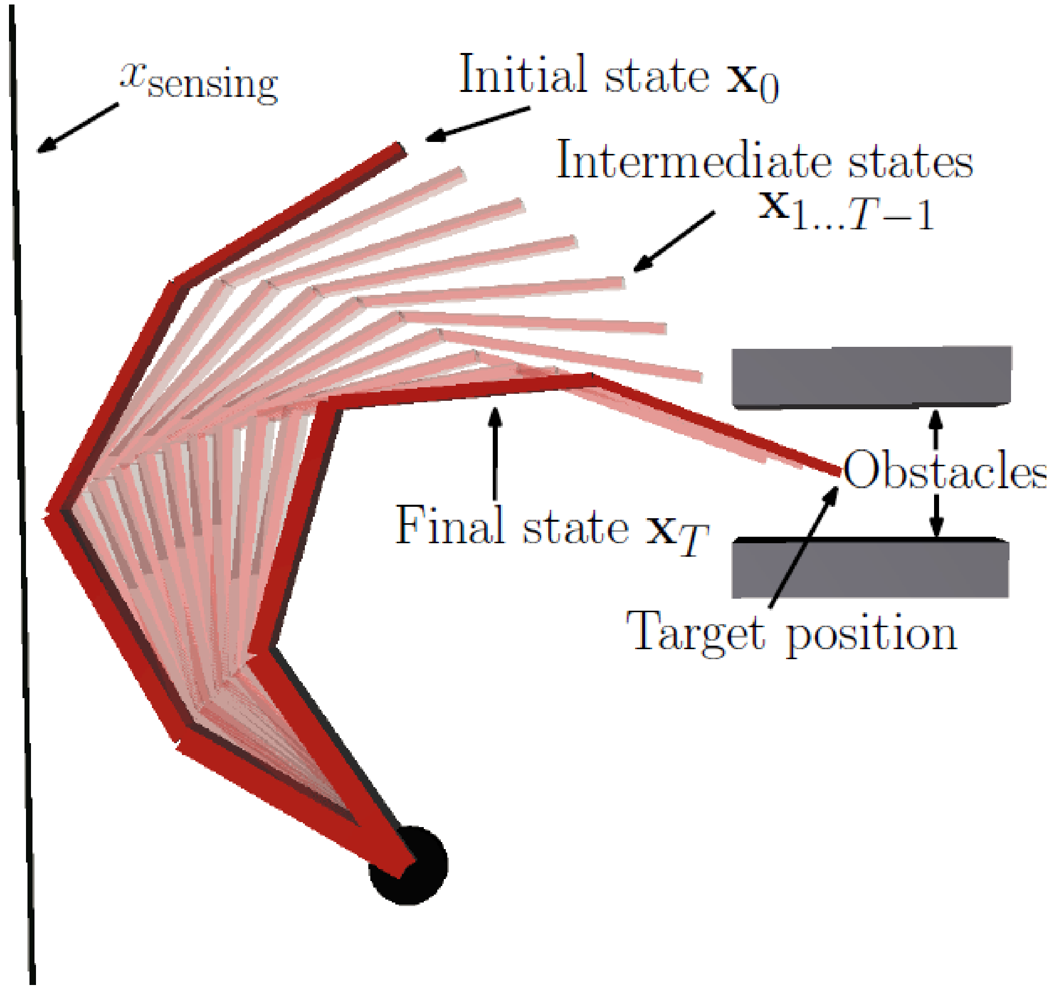
Example: 4-DOF planar robot

- Problem setup



Example: 4-DOF planar robot

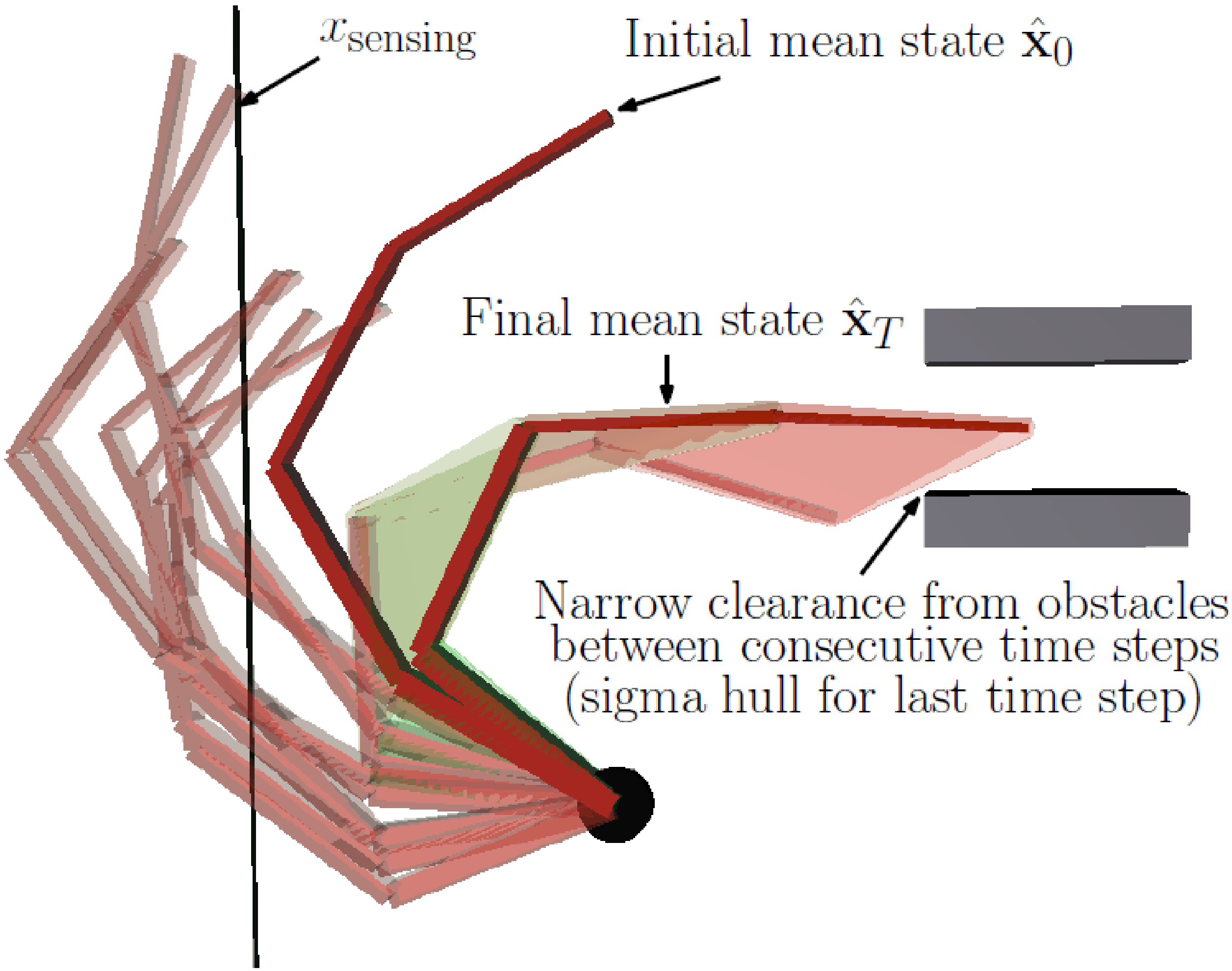
State-space trajectory



Presenter: Alex Lee (UC Berkeley)

Example: 4-DOF planar robot

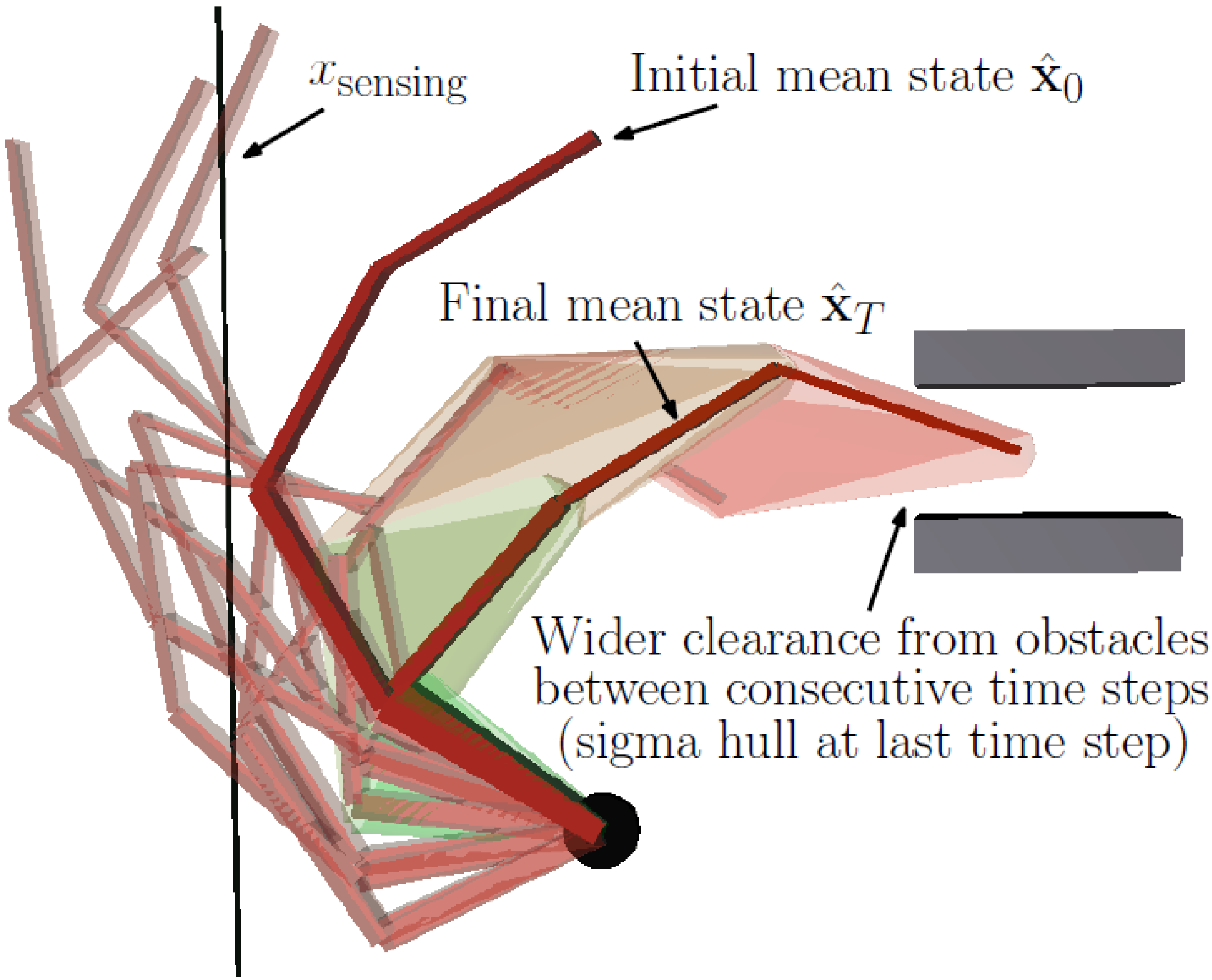
1-standard deviation belief space trajectory



Presenter: Alex Lee (UC Berkeley)

Example: 4-DOF planar robot

4-standard deviation belief space trajectory



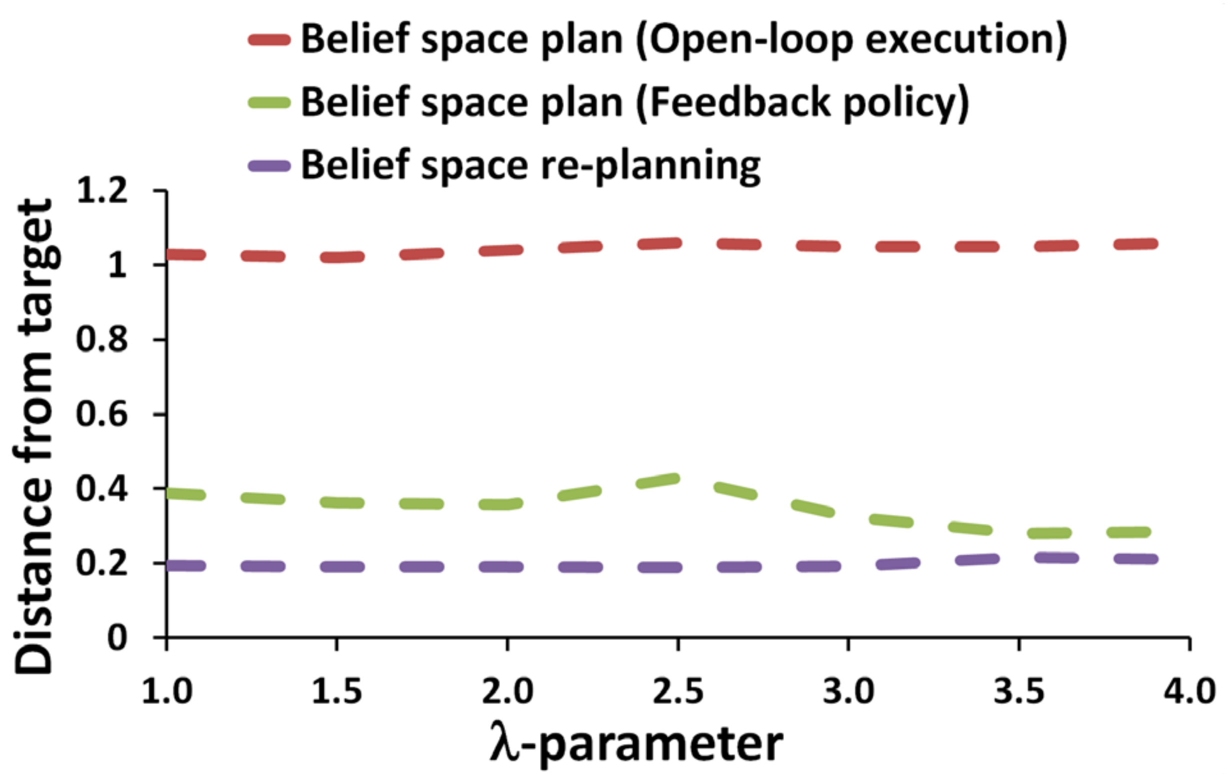
Presenter: Alex Lee (UC Berkeley)

Experiments: 4-DOF planar robot

- Open-loop execution
- Feedback linear policy
- Re-planning (MPC)

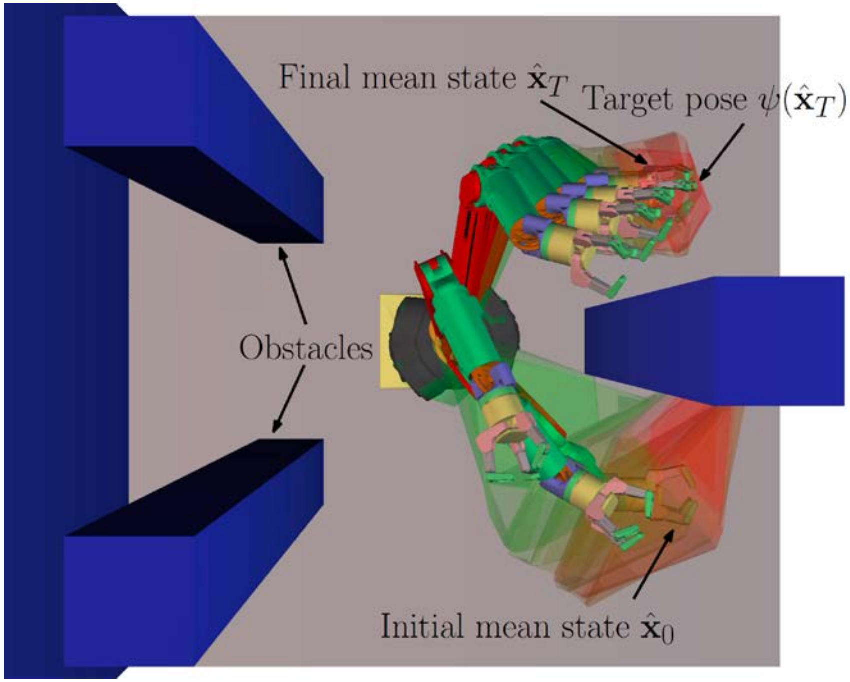
Experiments: 4-DOF planar robot

Mean distance from target



Presenter: Alex Lee (UC Berkeley)

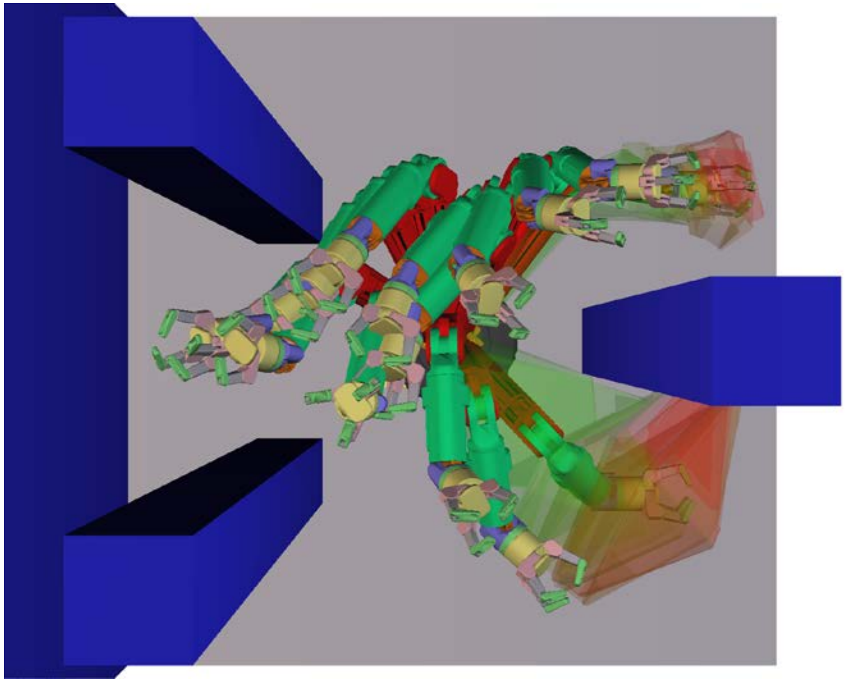
Example: 7-DOF articulated robot



State space trajectory

7 dimensions

1.9 secs



Belief space trajectory

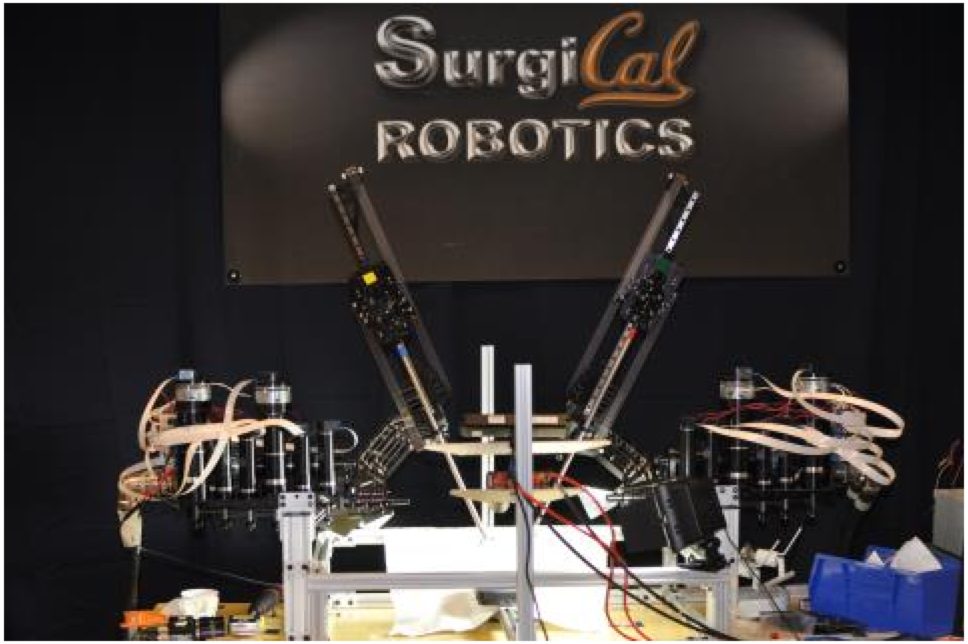
35 dimensions

8.2 secs

Presenter: Alex Lee (UC Berkeley)

Extensions

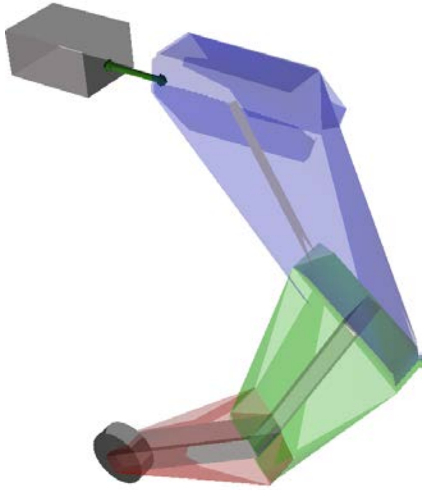
- Planning in uncertain environments
- Multi-modal belief spaces
- Physical experiments with the Raven surgical robot

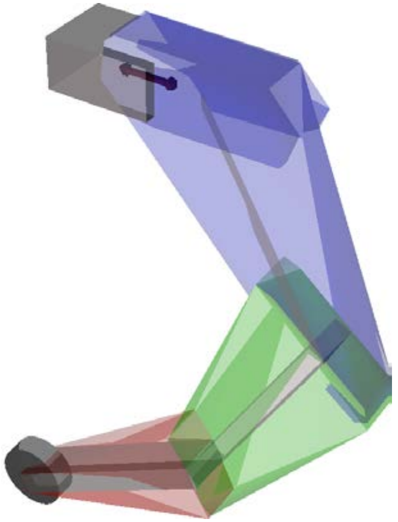


Presenter: Alex Lee (UC Berkeley)

Conclusions

- 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 – use fast convex collision detection and analytical gradients
- Iterative re-planning in belief space (MPC)

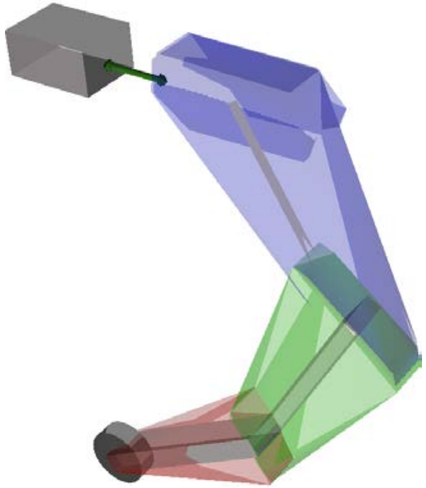


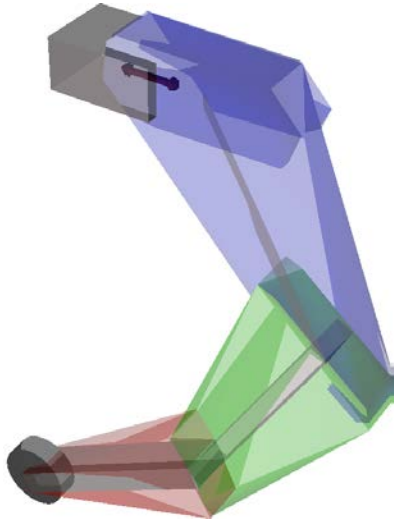


Presenter: Alex Lee (UC Berkeley)

Thank You

- Code available upon request
- Contact: alexlee_gk@berkeley.edu





Presenter: Alex Lee (UC Berkeley)