Controlling Agile Robots with Formal Safety Guarantees

Anirudha Majumdar

Stanford → Princeton

December 11, 2016

Introduction

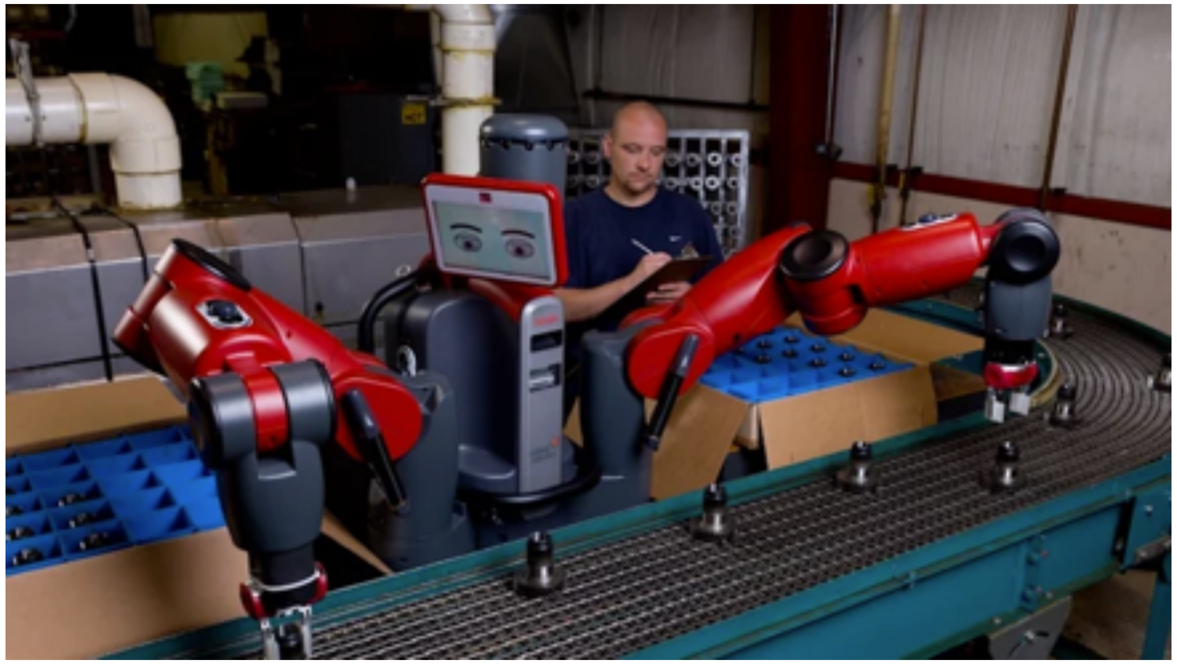
- The Federal Aviation Administration (FAA) is very concerned
- **Safety** with autonomy in the loop
  - Regulation, certification

Ideal goal: ***Formally guarantee*** that the robot will operate safely

- Dynamics are complicated and uncertain
- Cluttered (e.g., urban) environments

Introduction

- Challenges not restricted to unmanned aerial vehicles (UAVs)



Roadmap

1. Focus on high-speed unmanned aerial vehicle (UAV) flight
2. Sum-of-squares (SOS) programming-based algorithms for designing feedback controllers with formal guarantees on safety
3. Real-time planning in previously unseen environments
4. Hardware experiments on a small fixed-wing airplane flying through cluttered environments

General problem

Guarantee that UAV will fly through an environment without collisions given no prior map

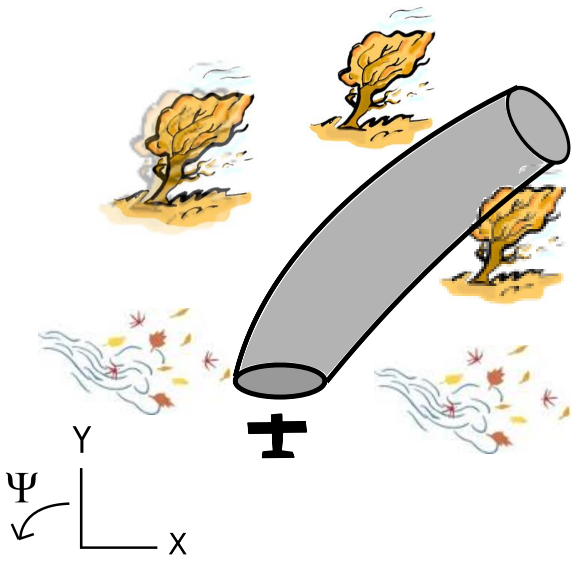
- Assumptions:
  - Model of dynamics
  - Model of bounded uncertainty in dynamics
  - Obstacles reported in a finite sensor horizon

Important sub-problem

- How to plan a *single maneuver* such that the UAV is guaranteed to be collision-free?

Related work

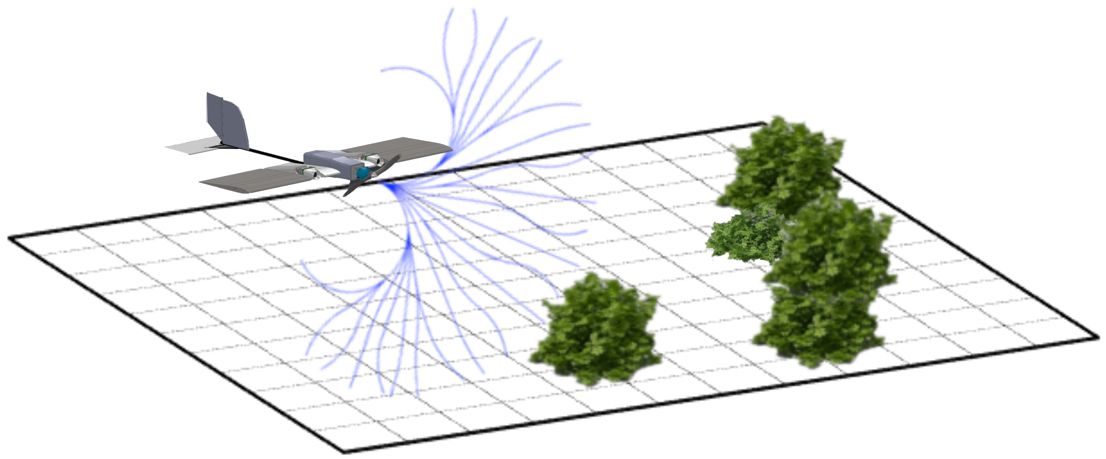
- Robust kinematic motion planning
  - [Brooks ’82, Lozano-Perez ’84, Jacobs ’90, Latombe ’90, Missiuro ’06, Guibas ’08, Malone ’13, ...]
- Planning under uncertainty
  - [Feldman ’77, Littman ’95, Kaelbling ’98, LaValle ’98, Prentice ’09, Candido’11, Patil ’15 ...]
- Computing reachable sets using Hamilton Jacobi Bellman (HJB) equation
  - Linear systems + bounded uncertainty [Kurzhanski ’01, Girard ’05, ...]
  - Nonlinear systems + bounded uncertainty [Tomlin ’03, Mitchell ’05, Gillula ’10, ...]

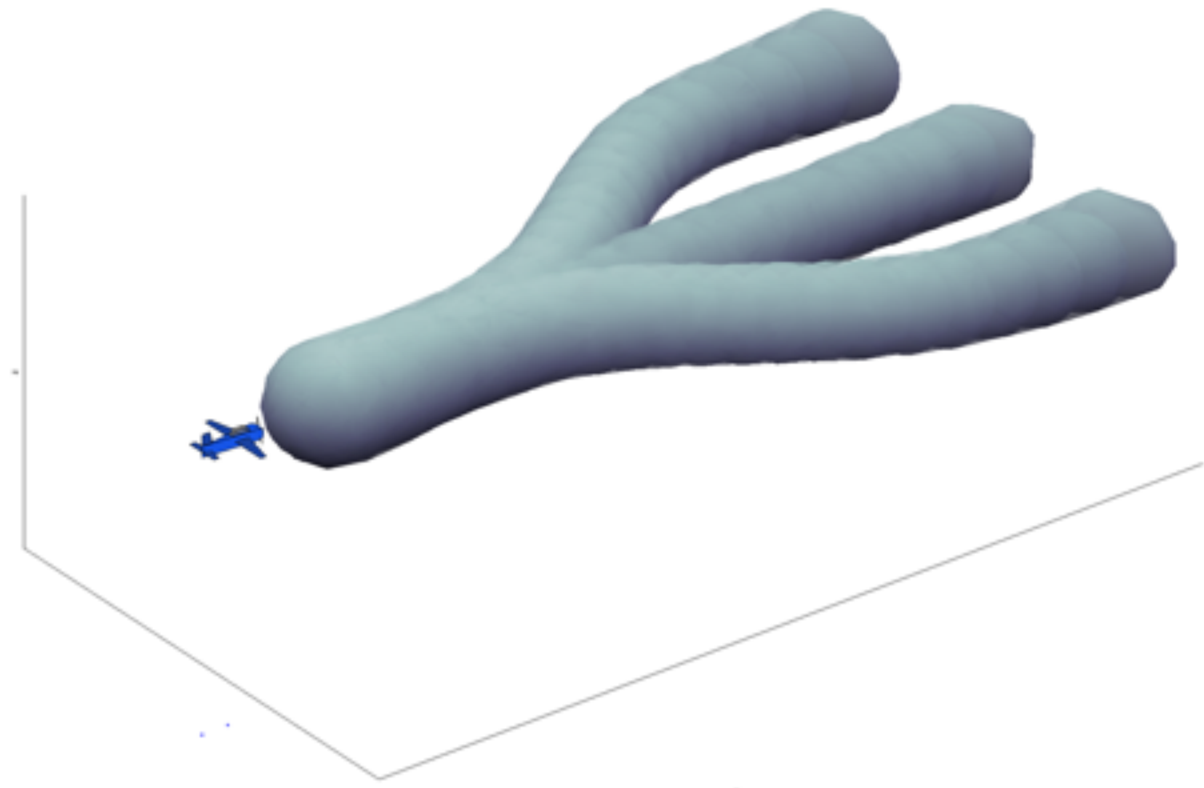


Approach

Offline computation:

- Compute trajectory library [Stolle ’06, Frazzoli ’01]
  - Widely used: [Stentz ’07, Liu ’13, Barry ’16...]
- Compute feedback controllers
- Compute “funnels” around each maneuver that the airplane is *guaranteed* to remain in

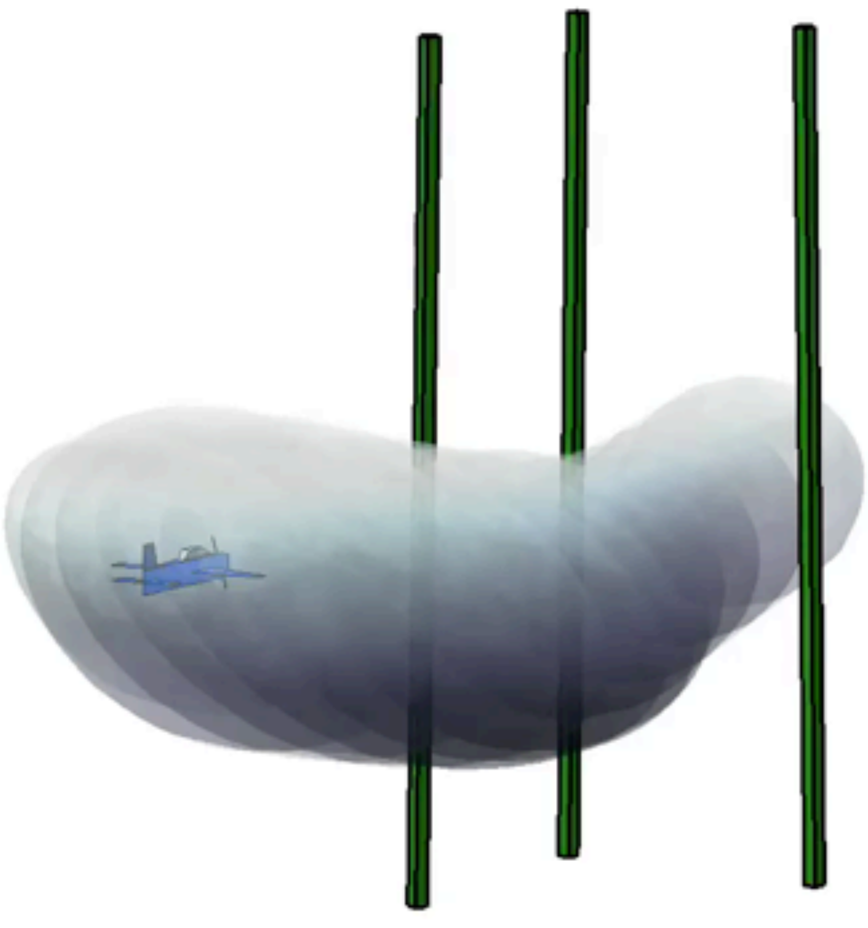




Approach

Online computation:

- Search through library to find a collision free funnel



This *guarantees* that the robot will remain collision free

Offline Computation

Funnels

- Control system:

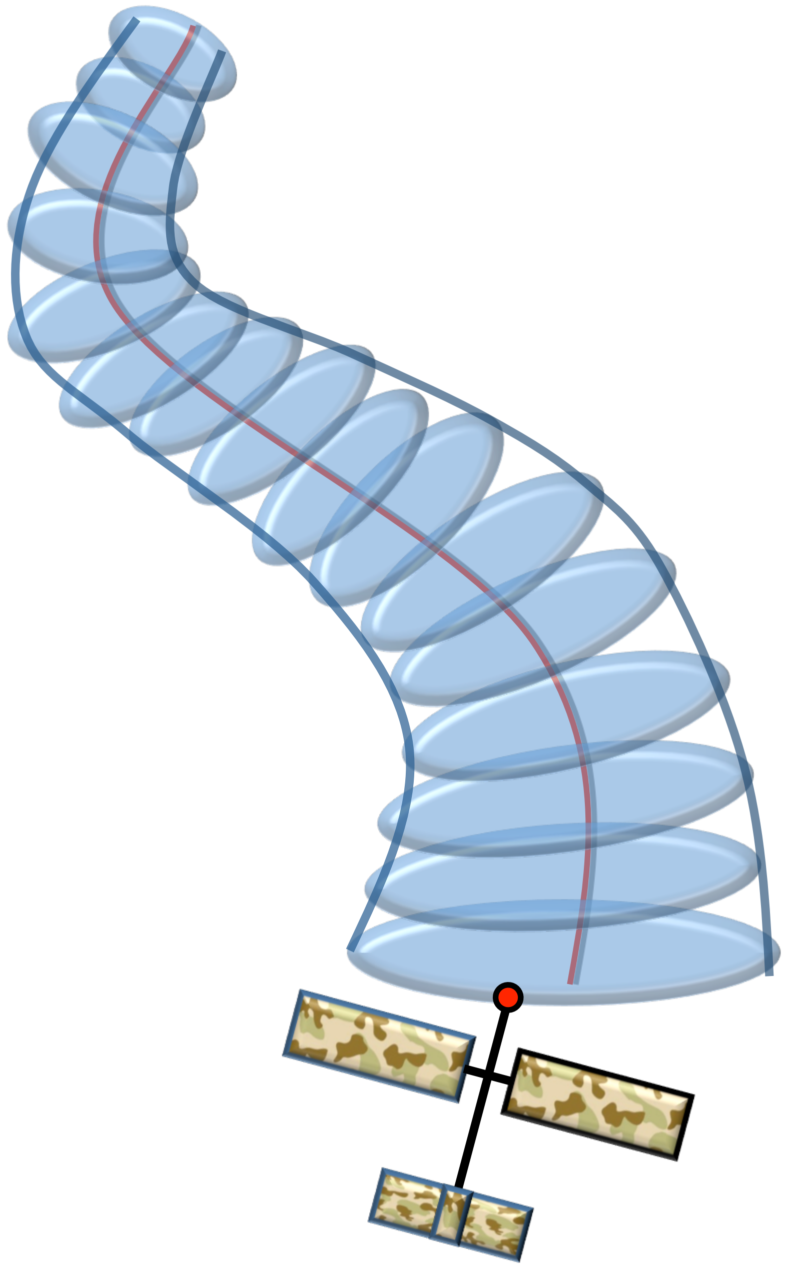
$$\dot{\mathbf{x}} = f(\mathbf{x}(t), \mathbf{w}(t), \mathbf{u}(t)), \;\; \mathbf{u}(t) \in \mathbb{R}^m$$

$\mathbf{x}(t) \in \mathbb{R}^n$: state

$\mathbf{w}(t) \in \mathbb{R}^d$: disturbance

$\mathbf{u}(t) \in \mathbb{R}^m$: control input

- Design feedback controller that minimizes size of funnel



Funnel

- Funnel is represented as a time-varying sub-level set:

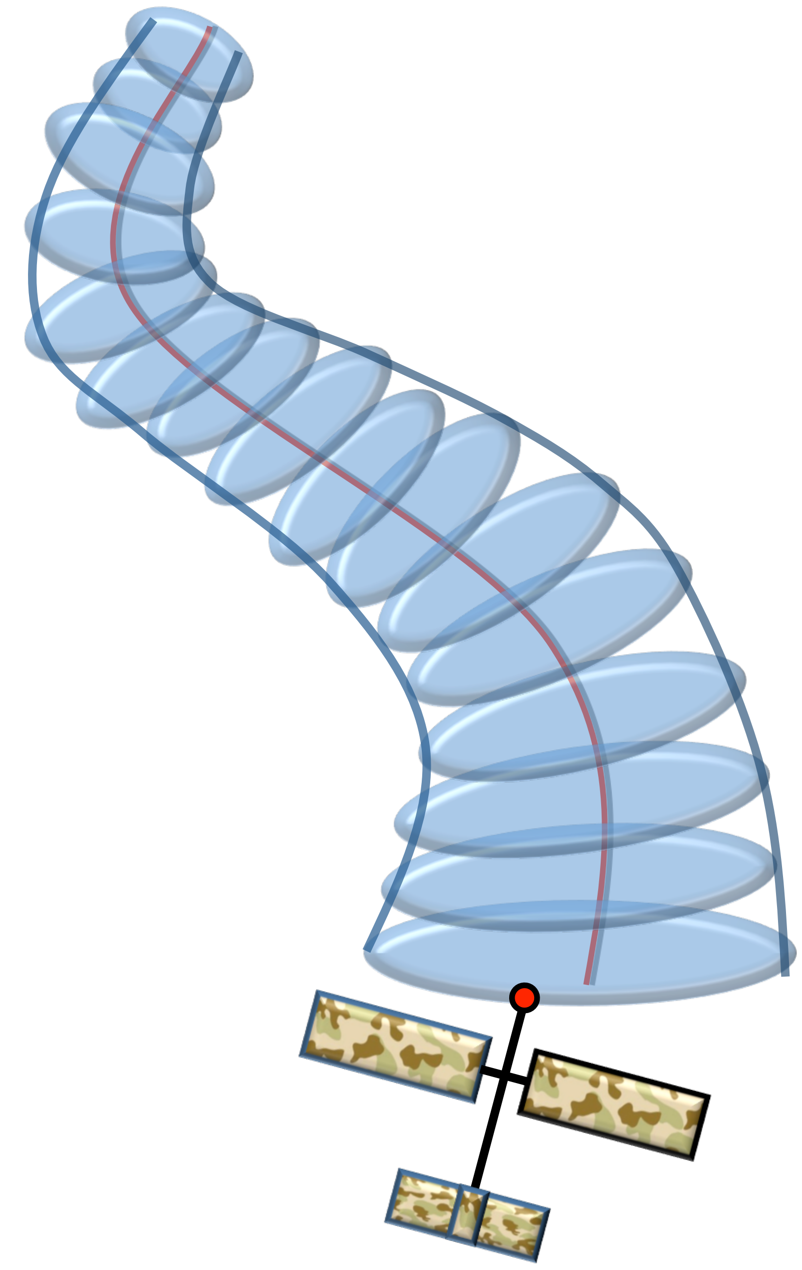
$$\mathcal{F}(t) = \{\mathbf{x}(t) \in \mathbb{R}^n | V(t, \mathbf{x}(t)) \leq \rho(t)\}$$

- Guarantees *invariance*:

$$\mathbf{x}(0) \in \mathcal{F}(0) \implies \mathbf{x}(t) \in \mathcal{F}(t), \forall t \in [0, T]$$

- Lyapunov condition [Tedrake '09]:

$$V(t, \mathbf{x}) = \rho(t) \implies \dot{V}(t, \mathbf{x}) < \dot{\rho}(t), \ \forall t \in [0, T]$$



Robust Funnel

- Dynamics subject to bounded uncertainty:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{w}(t)), \;\; \mathbf{w}(t) \in \mathcal{W}$$

- Can guarantee invariance despite bounded uncertainty:

$$\mathbf{x}(0) \in \mathcal{F}(0) \implies \mathbf{x}(t) \in \mathcal{F}(t), \forall \mathbf{w} : [0, T] \to \mathcal{W}$$

- Lyapunov condition:

$$V(t, \mathbf{x}) = \rho(t) \implies \dot{V}(t, \mathbf{x}, \mathbf{w}) < \dot{\rho}(t), \forall \mathbf{w} \in \mathcal{W}$$

[Majumdar & Tedrake, WAFR'12]

Implementation using SOS Programming

$$V(t,\mathbf{x}) = \rho(t) \implies \dot{V}(t,\mathbf{x},\mathbf{w}) < \dot{\rho}(t), \forall \mathbf{w} \in \mathcal{W}$$

- Approximate dynamics with polynomials
- Impose Lyapunov conditions for funnels using using Sum-of-Squares (SOS) programming
- SOS programs for UAV: 12 variables (typically degree 4 SOS constraints)

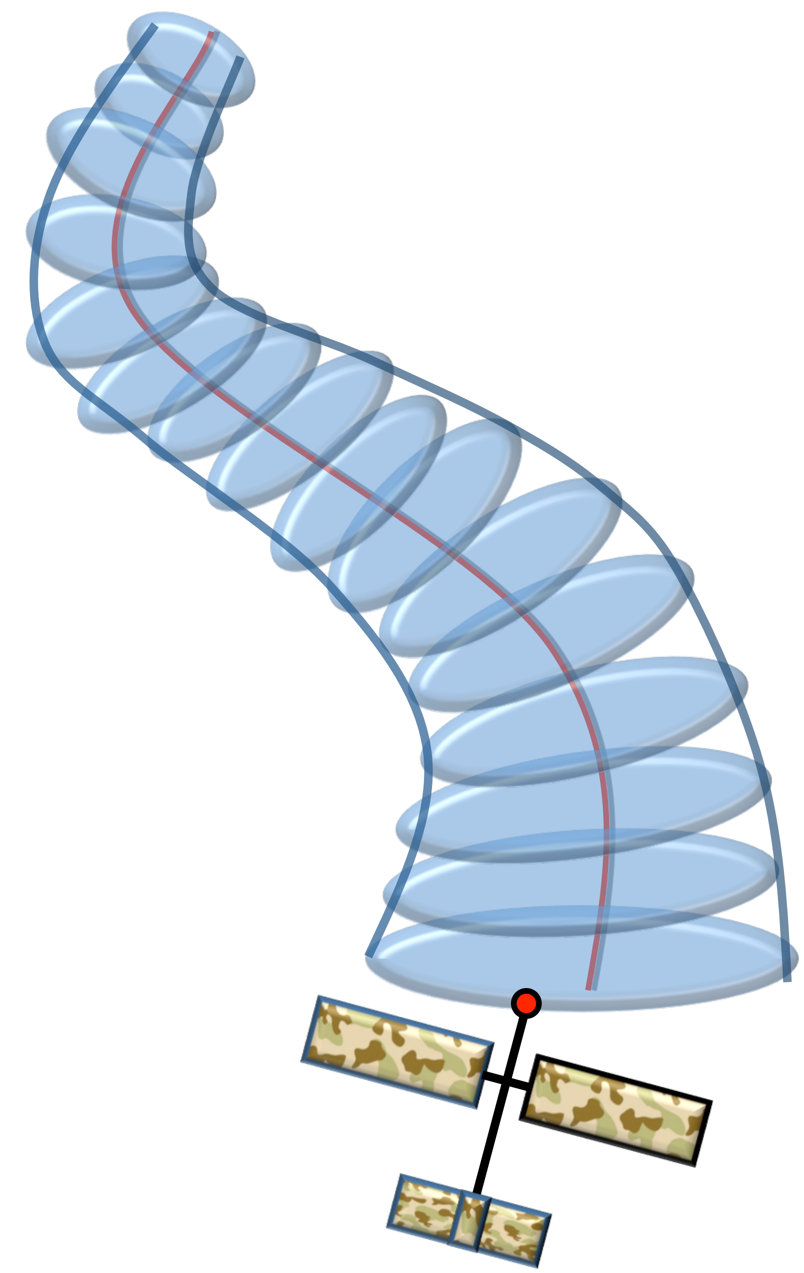
Feedback control synthesis

- Control system:

$$\dot{\mathbf{x}} = f(\mathbf{x}(t), \mathbf{w}(t), \mathbf{u}(t)), \ \ \mathbf{u}(t) \in \mathbb{R}^m$$

- Can design controller that explicitly minimizes size of funnel

- Alternate between search for Lyapunov function and controller



[Majumdar, Ahmadi & Tedrake, ICRA'13]. Best Paper Award.

Hardware validation

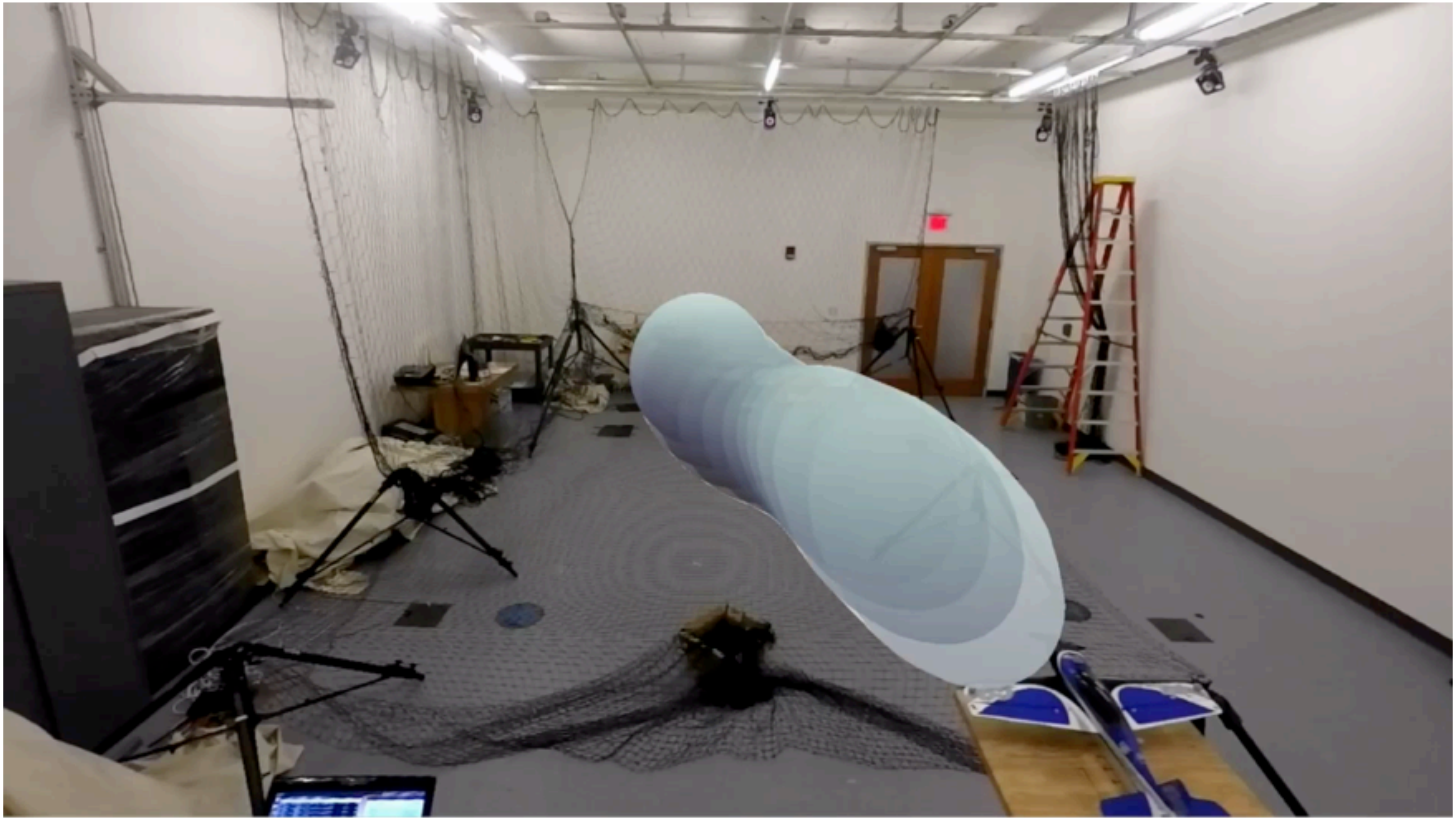
Goal: Demonstrate that guarantees from funnel are valid on real hardware system

Hardware validation on fixed-wing airplane

- SBach:
  - Small acrobatic RC airplane
- System identification: accurate 12 state dynamic model
  - Rigid-body subject to aerodynamic forces
  - Lift/drag coefficients: flat-plate model + correction
  - Model refined using data from flights
  - Parametric uncertainty: decreased as model improved
- Experiments are in motion capture arena
- Computation is off-board

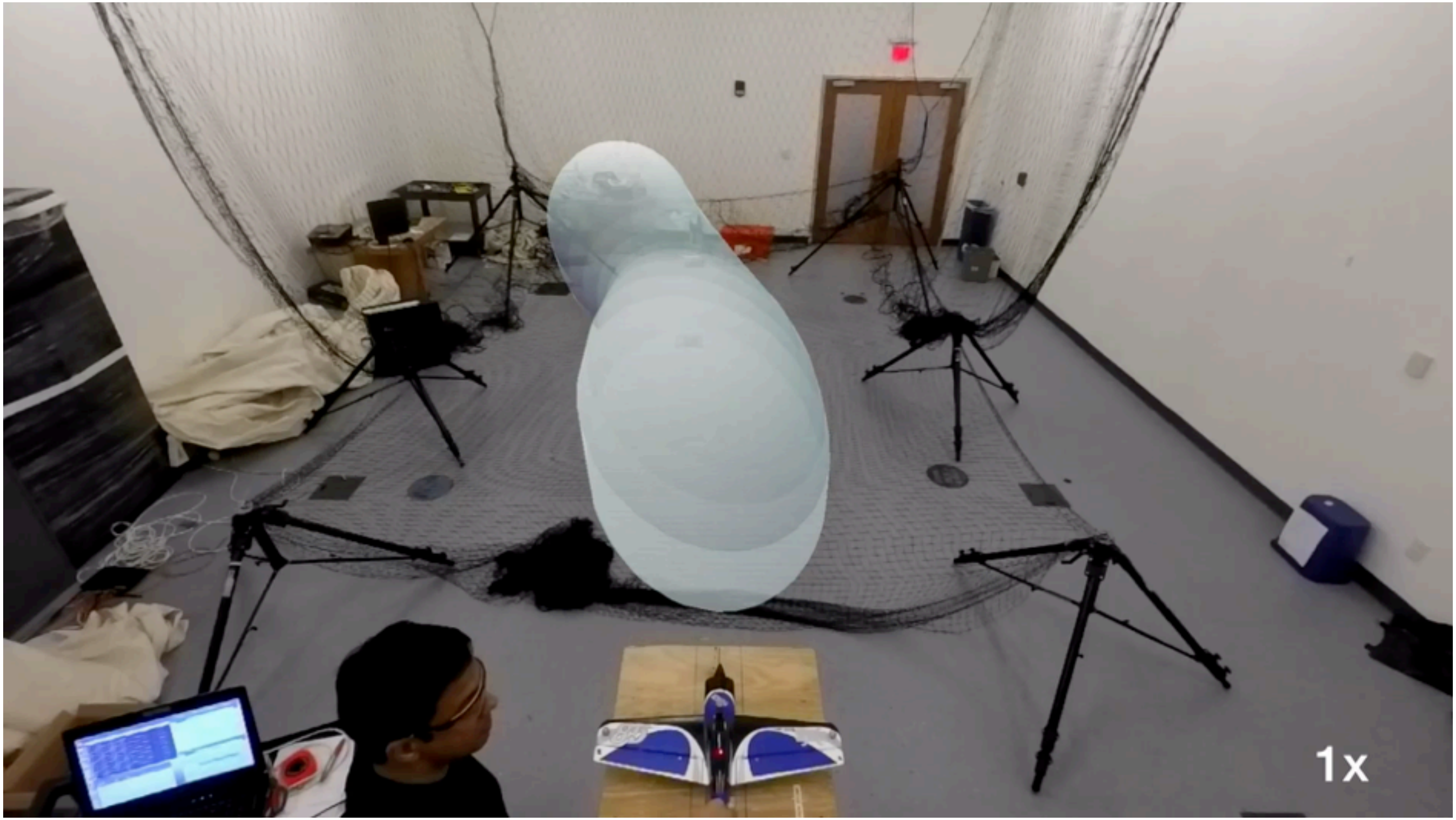
[Majumdar & Tedrake '16, arXiv:1601.04037]

Funnel



Computation time: ~30 mins

Flying through funnel



Experimental results

- **Funnel is valid for hardware experiments**

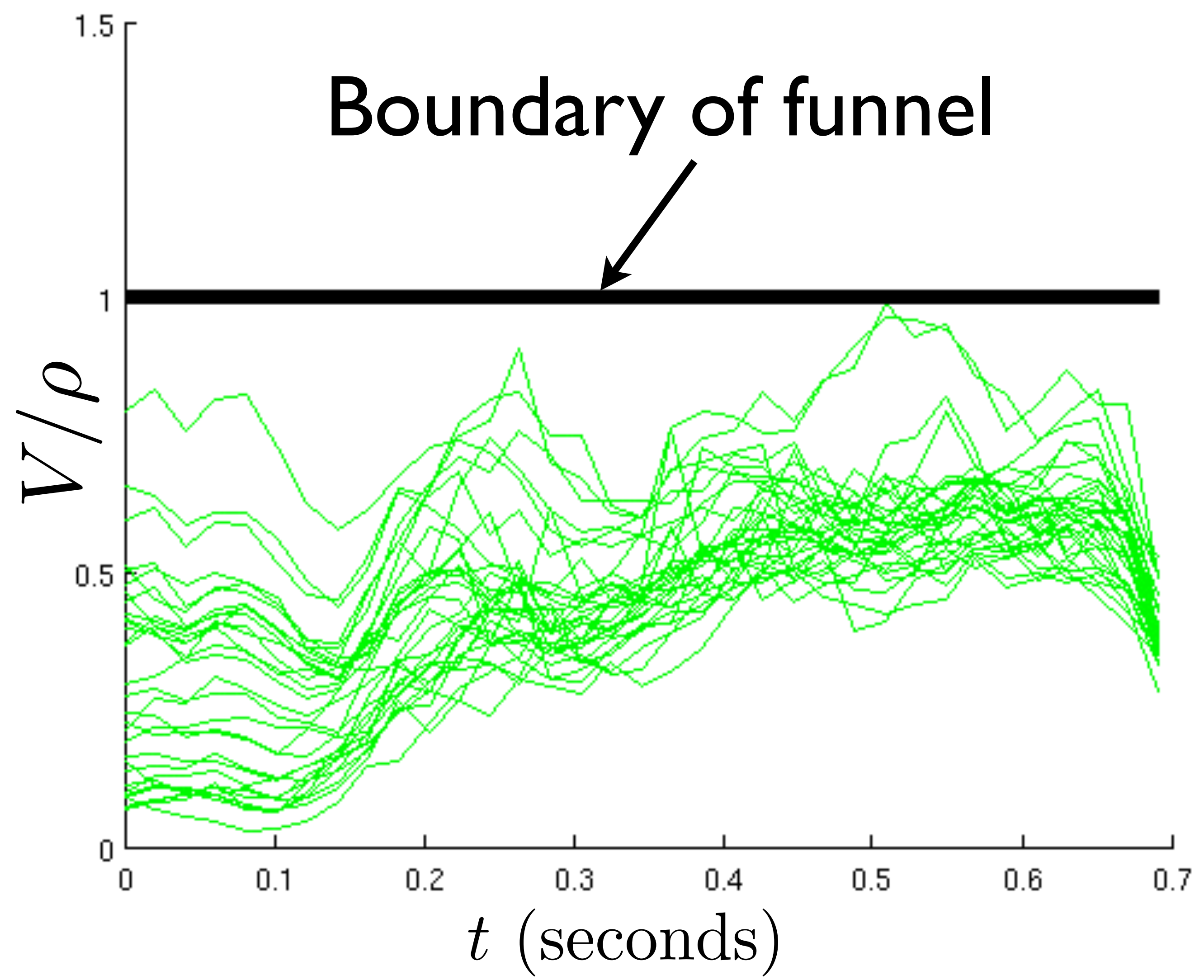
- **30 flights**

    - Different initial conditions in inlet of funnel

- ***All trajectories*** remain within the 12 dimensional funnel

Experimental results

- **Funnel is valid for hardware experiments**
- **30 flights**
    - Different initial conditions in inlet of funnel
- ***All trajectories*** remain within the 12 dimensional funnel



Online Computation

Online computation

- How to deal with obstacles reported by sensors at runtime?

Real-time planning with funnels

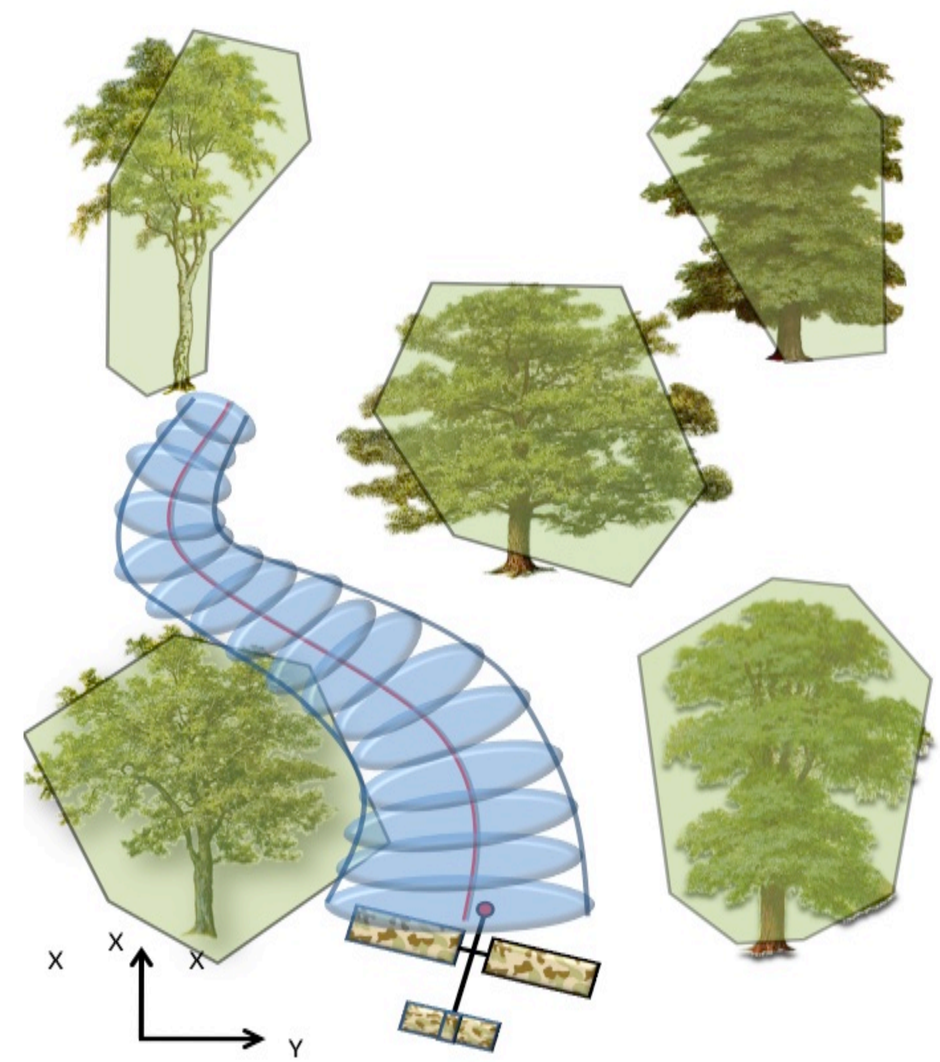
- Search through pre-computed library of funnels

- Find one whose *projection* is collision free
  - This is a purely geometric problem
  - Leverage mature collision libraries (e.g., Bullet)

[Majumdar & Tedrake, WAFR'12]

Exploiting invariances in dynamics

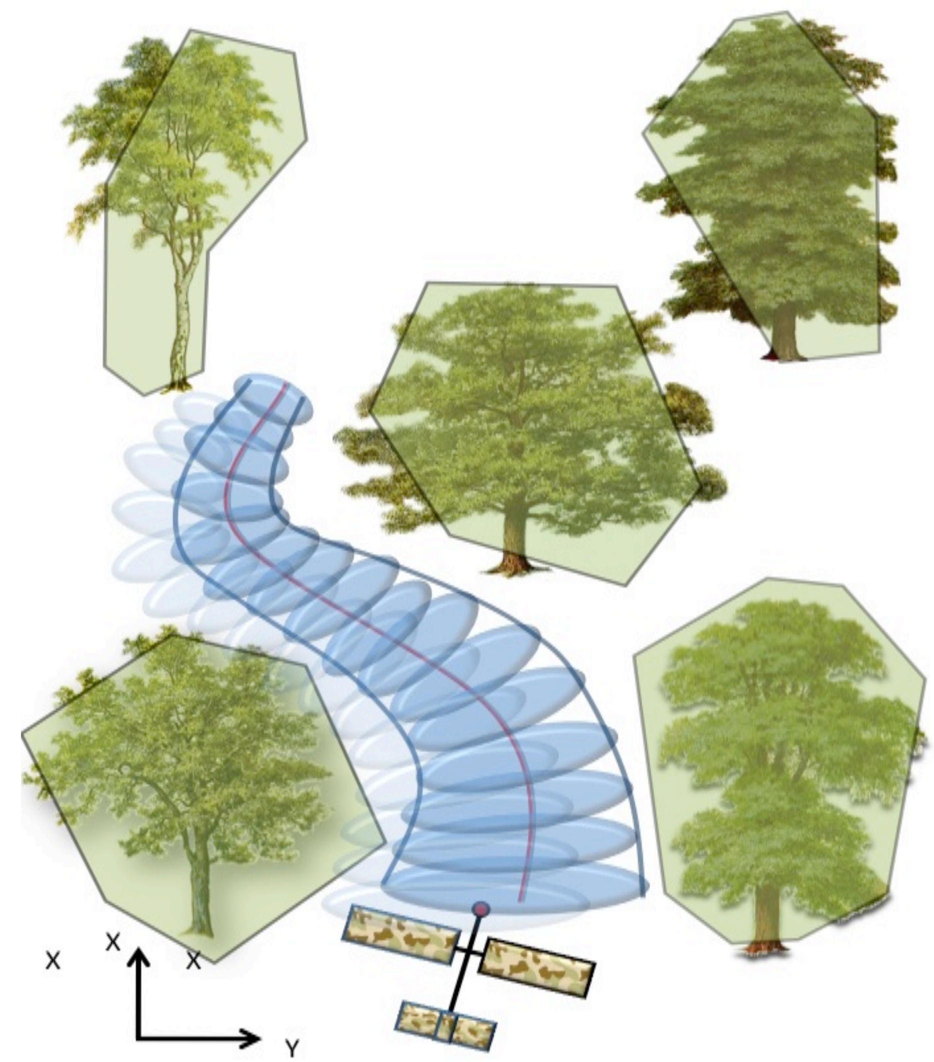
- What if we cannot find a collision-free funnel?
- **Idea:** Exploit invariances in dynamics
  - e.g., shift invariance
- Shifting a funnel results in a valid funnel
- Shift funnel while maintaining current state in "inlet"



[Majumdar & Tedrake '16, arXiv:1601. 04037]

Exploiting invariances in dynamics

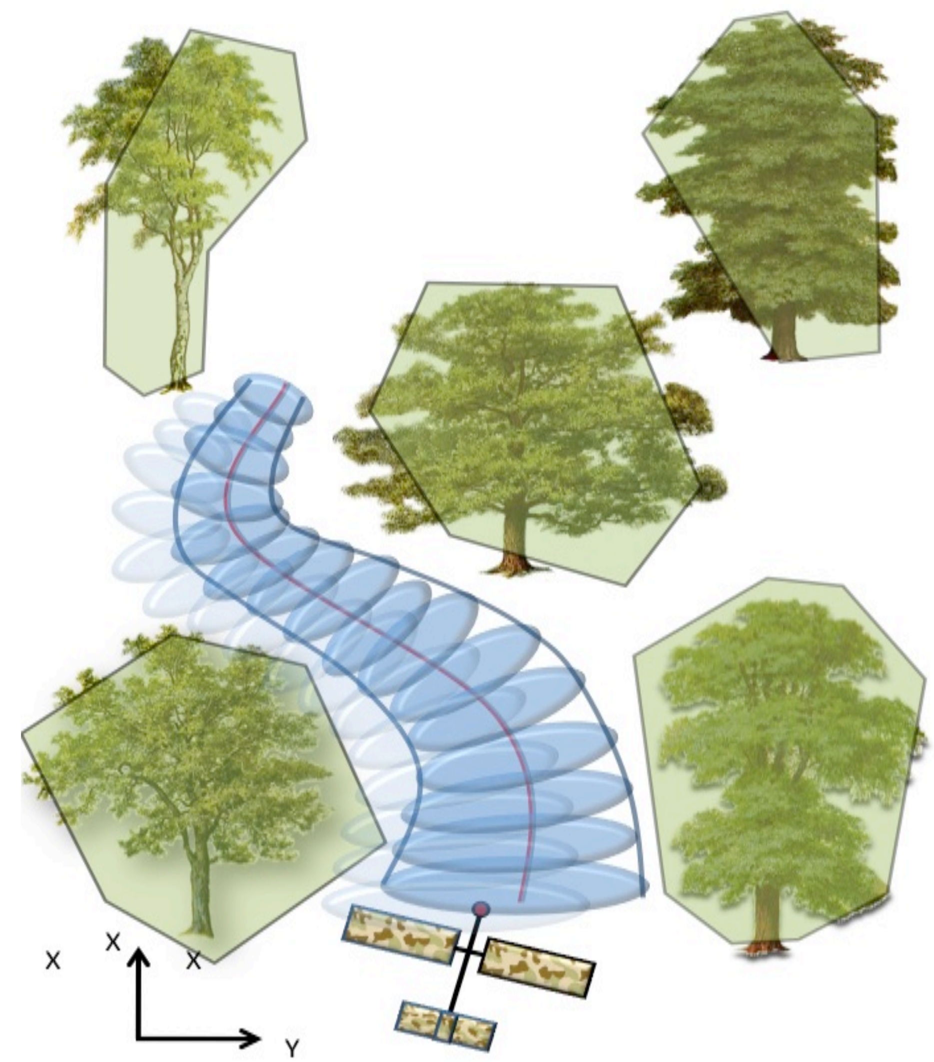
- What if we cannot find a collision-free funnel?
- **Idea:** Exploit invariances in dynamics
  - e.g., shift invariance
- Shifting a funnel results in a valid funnel
- Shift funnel while maintaining current state in "inlet"



[Majumdar & Tedrake '16, arXiv:1601. 04037]

Exploiting invariances in dynamics

- What if we cannot find a collision-free funnel?
- **Idea:** Exploit invariances in dynamics
  - e.g., shift invariance
- Shifting a funnel results in a valid funnel
- Shift funnel while maintaining current state in “inlet”

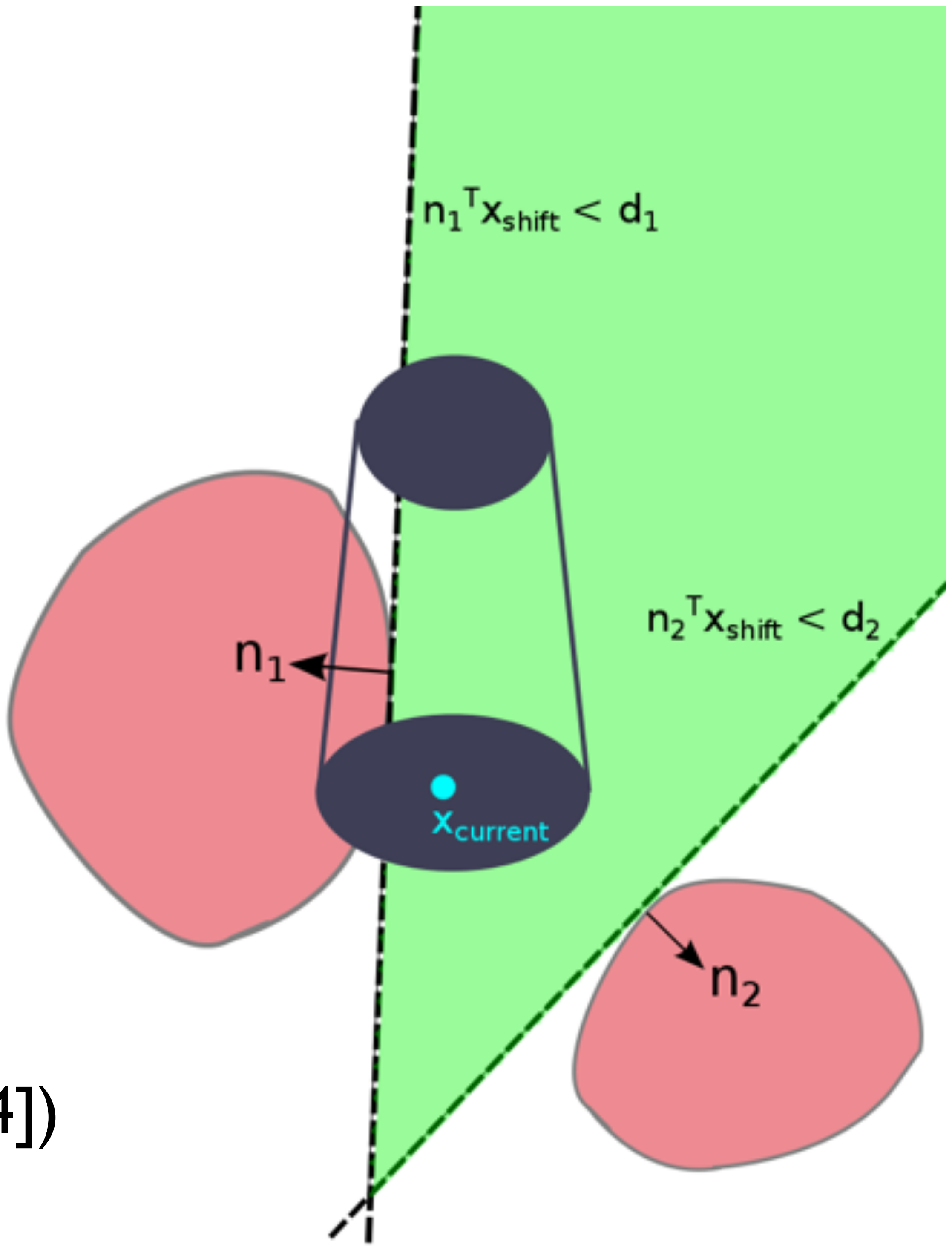


Can handle using a convex Quadratically Constrained Quadratic Program

[Majumdar & Tedrake ’16, arXiv:1601. 04037]

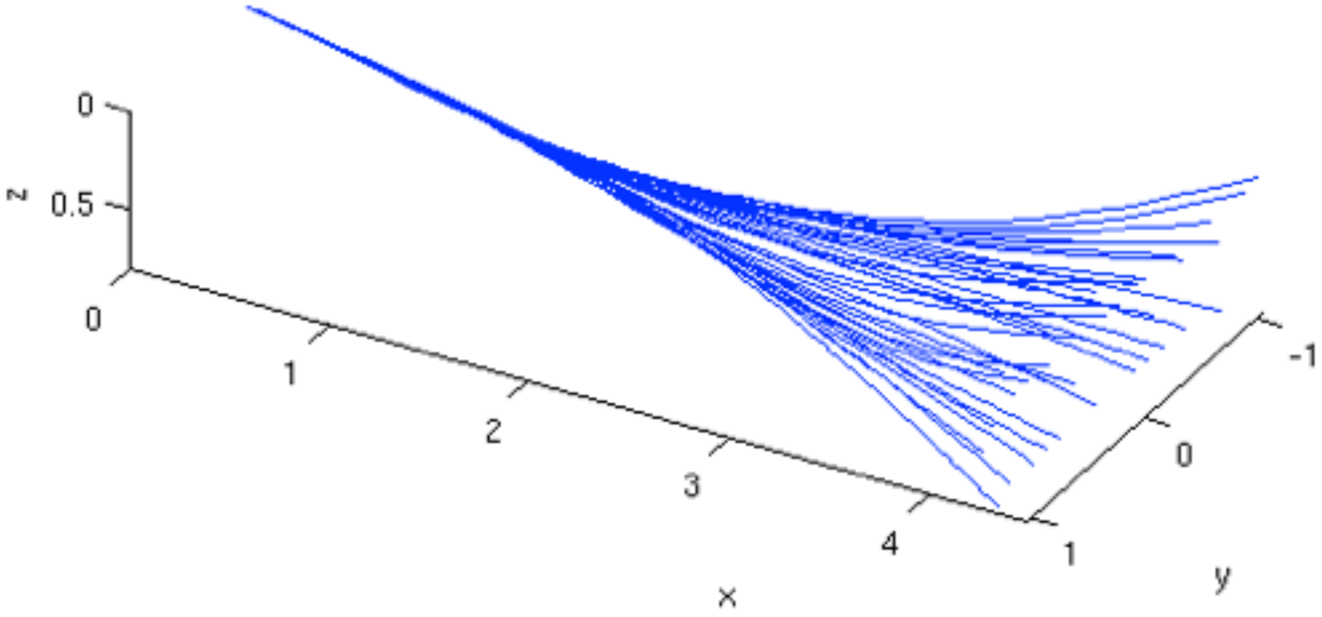
Exploiting invariances in dynamics

- For each segment of funnel, find collision normal and distance to each obstacle
- This defines a separating hyperplane to each obstacle
- These are linear constraints
- Containment of current state in inlet of funnel (convex quadratic constraint)
- Convex QCQP!
- Extremely fast software based on code generation (e.g., ForcesPro [Domahidi '14])



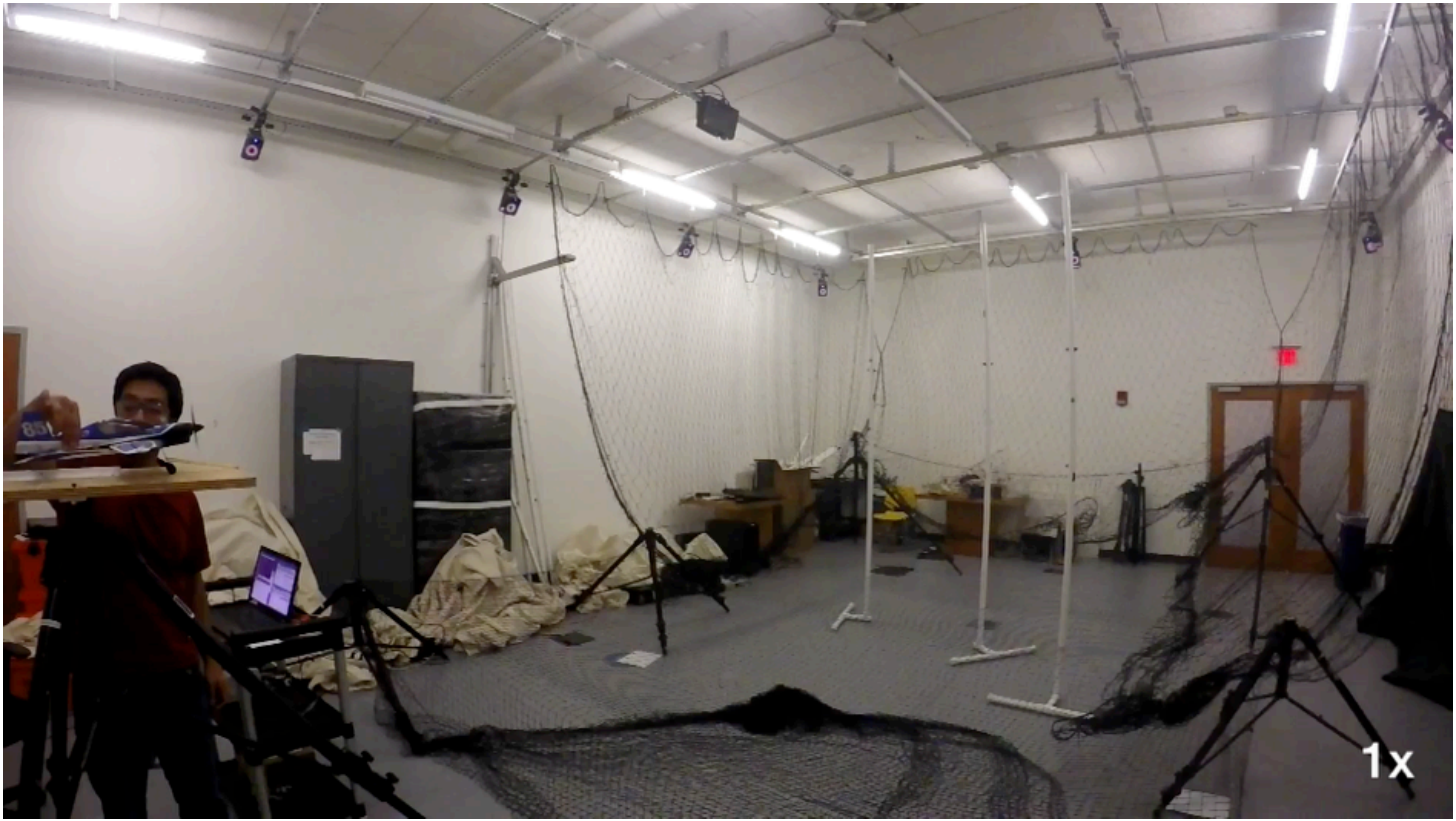
Hardware experiments

- 40 pre-computed funnels
- Planner is informed obstacle locations when airplane clears launcher

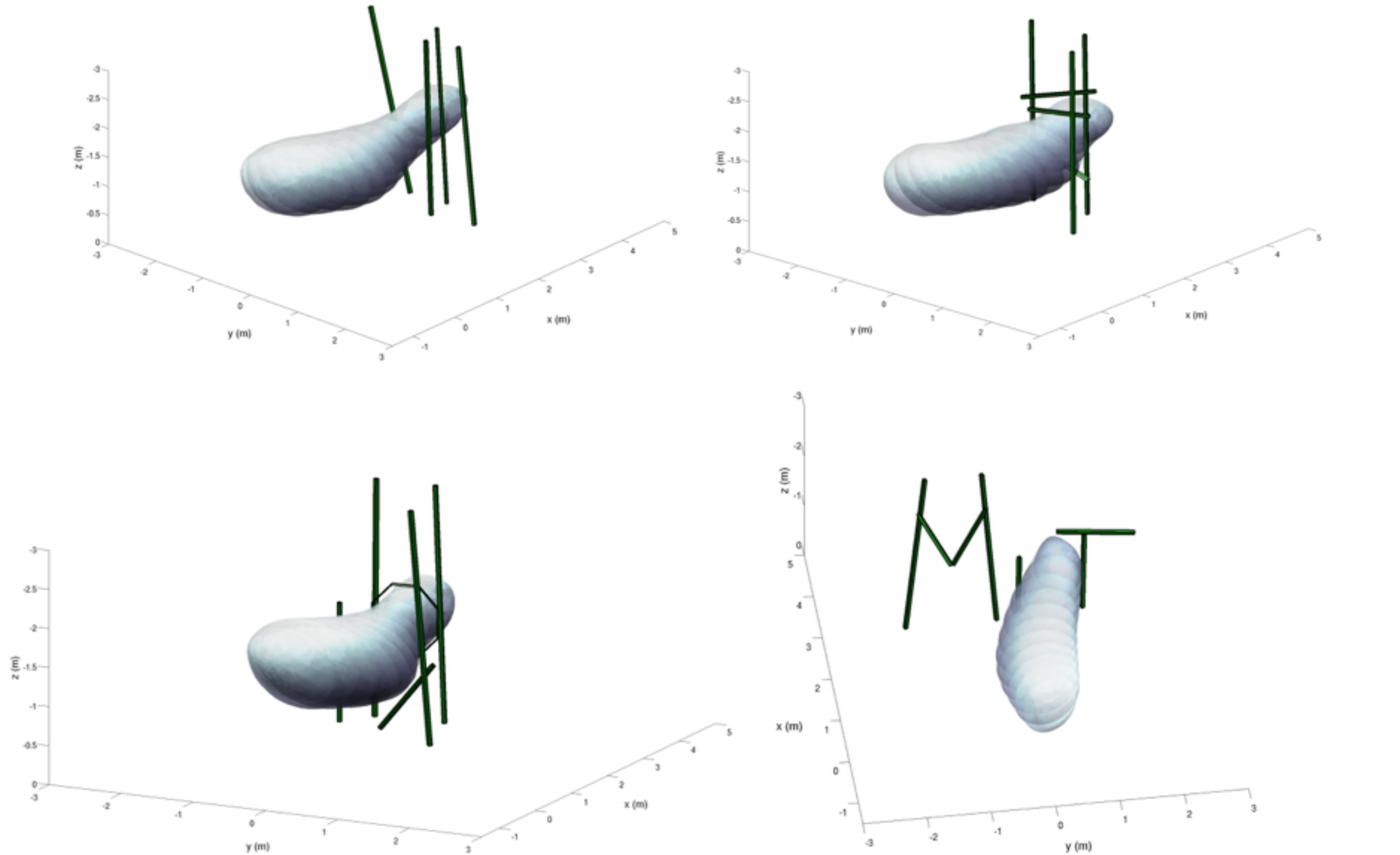


[Majumdar & Tedrake '16, arXiv:1601. 04037]

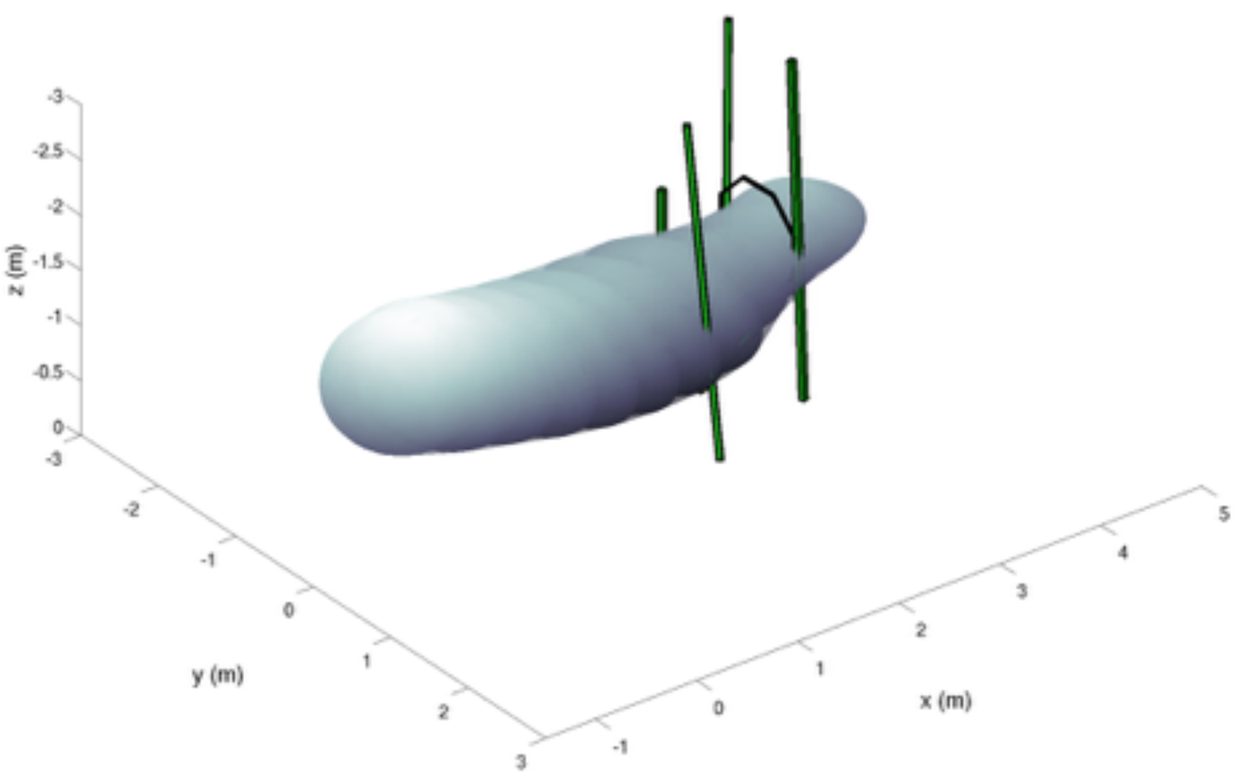
Results



Examples of planned funnels

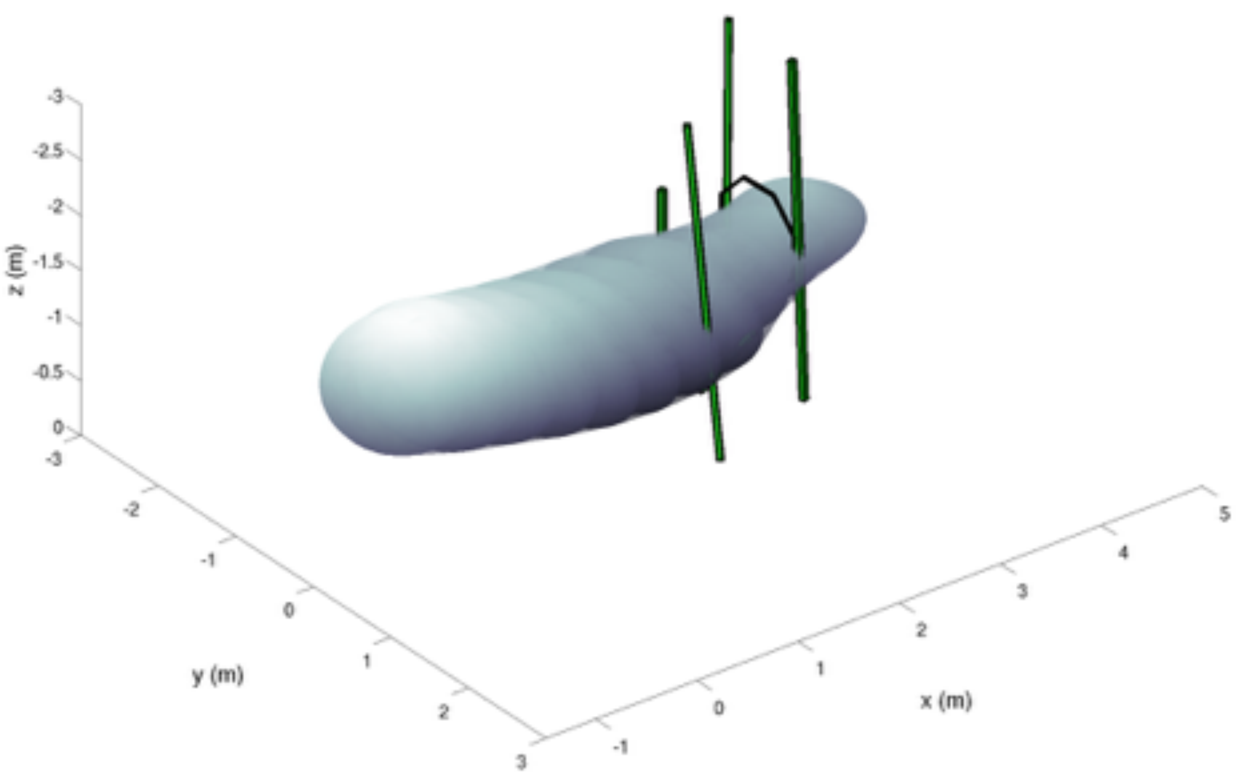


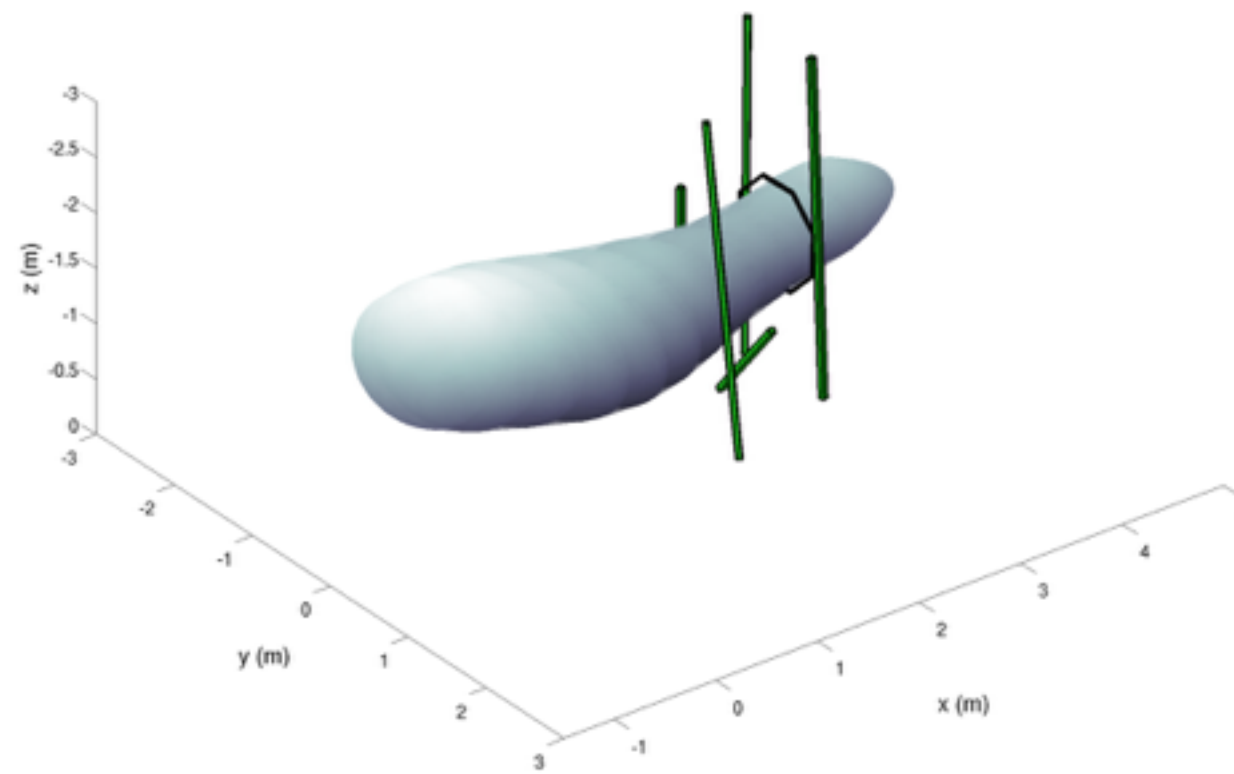
Importance of exploiting invariances



Best funnel with no shifting

Importance of exploiting invariances



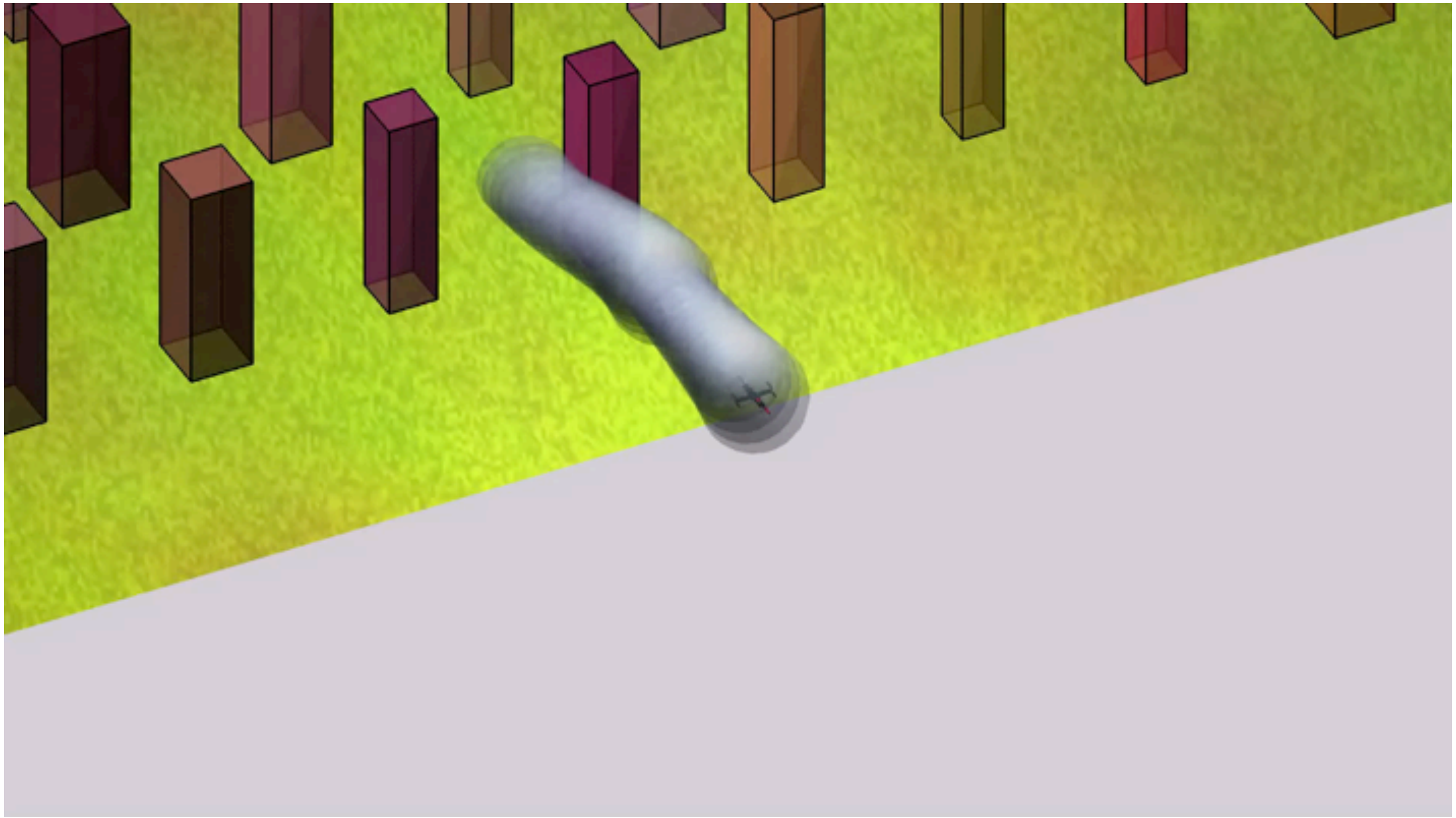


Best funnel with no shifting

Best funnel with shifting

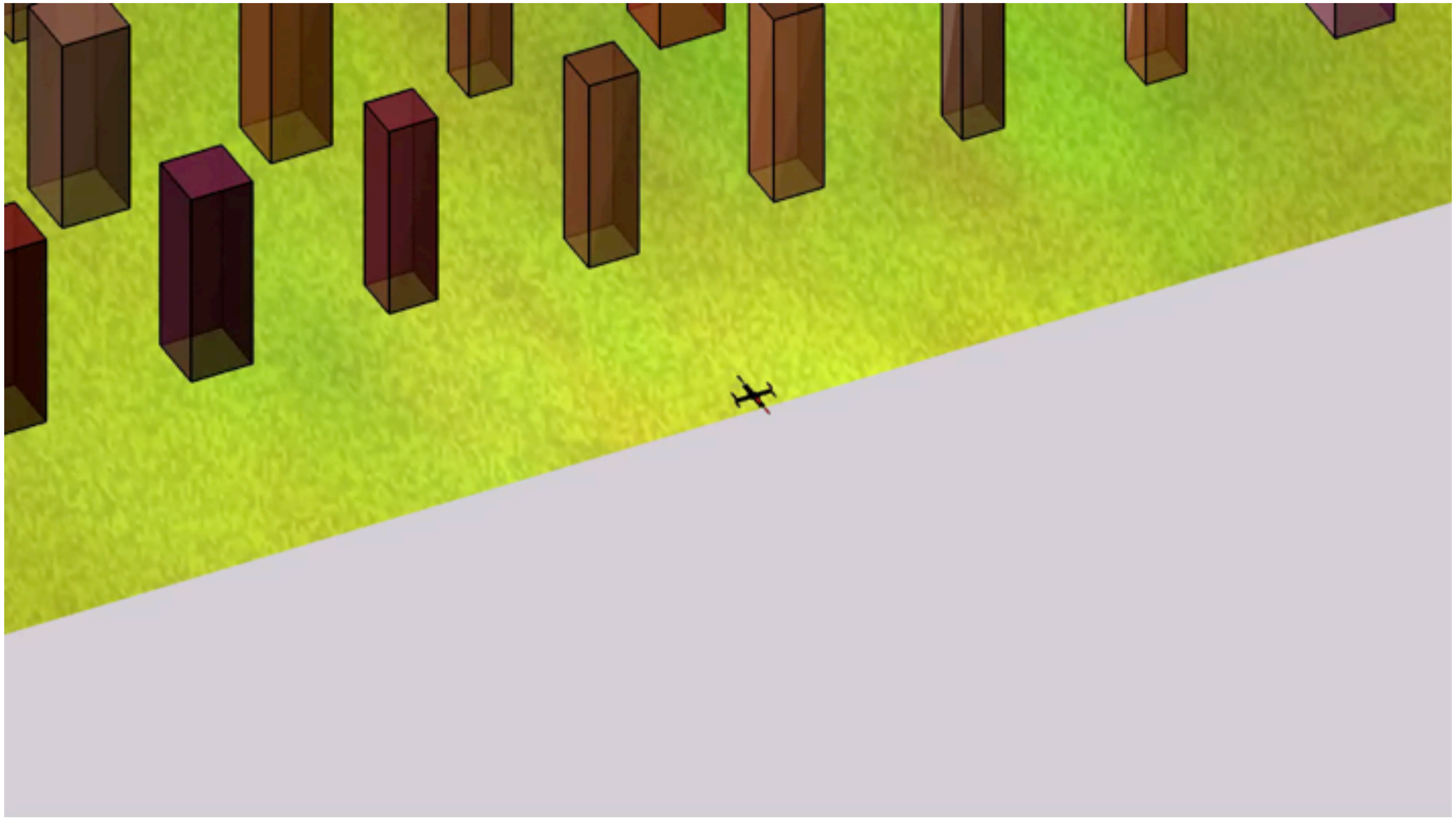
Flying continuously

Receding horizon planning



- Quadrotor: 12 states, 4 control inputs
- Uncertain "cross-wind"

Receding horizon planning



Constraints on environment guarantee that collision-free funnels will always be found

Robust real-time planning using contraction theory

Joint work with Sumeet Singh (Stanford), Marco Pavone (Stanford), Jean-Jacques Slotine (MIT)

Do we need a fixed library of funnels?

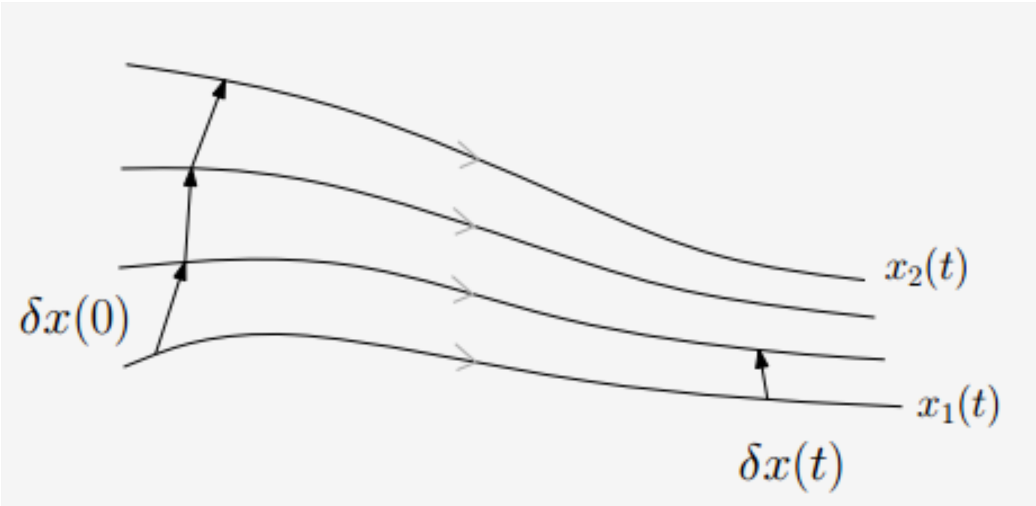
- Ideal goal: Generate a funnel around any nominal trajectory
- Need a notion of invariance that is independent of a specific trajectory
- Incremental exponential stability (IES):

$$\|x^\star(t) - x(t)\| \leq Ce^{-\lambda t}\|x^\star(0) - x(0)\|$$

- Goal: design a tracking feedback controller that can be applied to *any* feasible trajectory and make it IES

Contraction Theory

- Contraction theory [Lohmiller and Slotine '98]:
  - Convergence between trajectories
  - Dynamics of (infinitesimal) distances δ between trajectories is linear



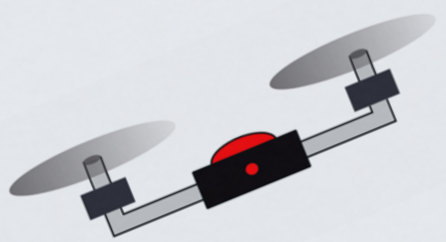
- Control contraction metrics (CCMs) [Manchester and Slotine '15, '16]:
  - Design a differential controller using a differential Control Lyapunov Function
  - Conditions based on SOS programming

Robust real-time motion planning

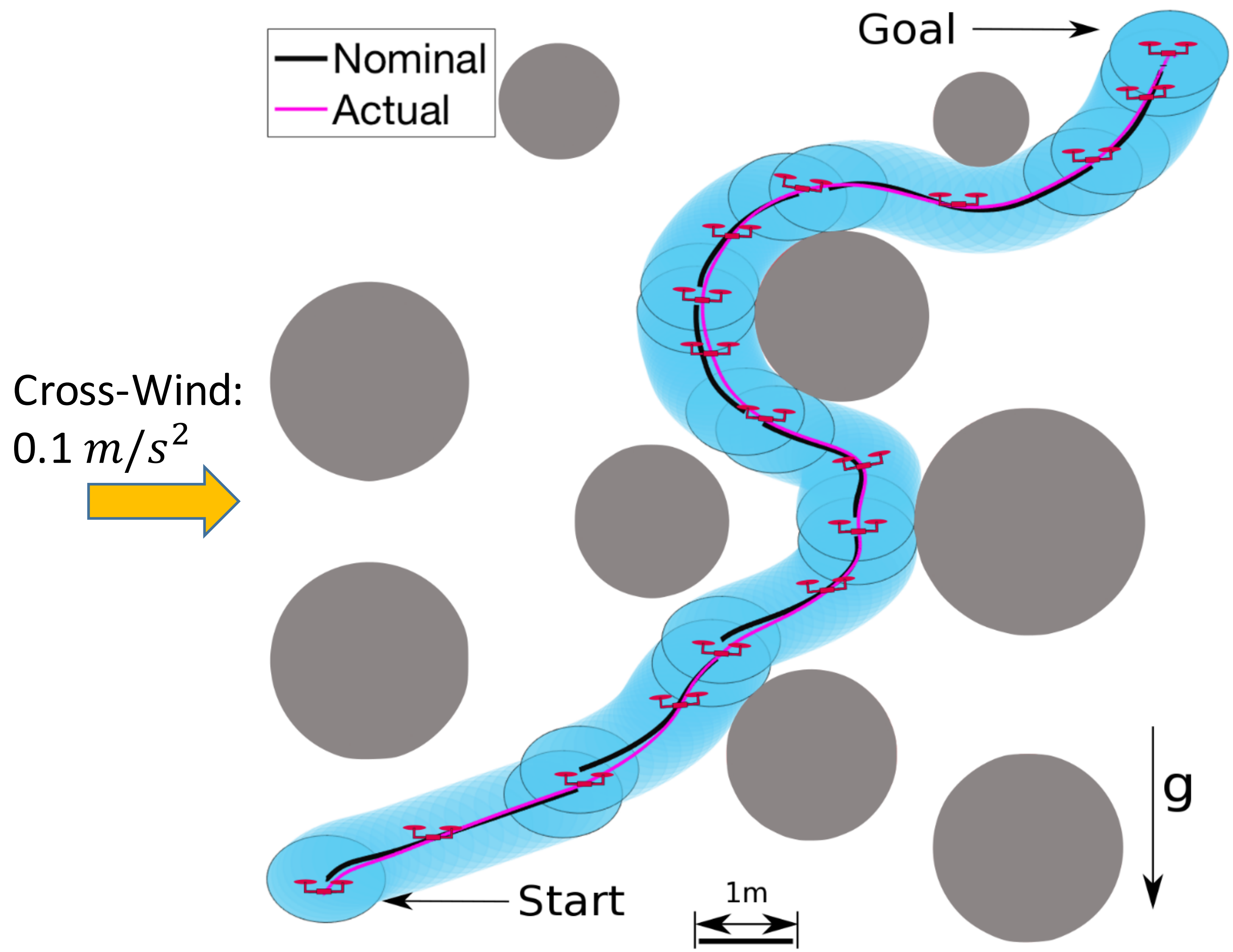
- Offline:
- CCM tracking controller can be used to make *any* nominal trajectory IES
- This gives us a (fixed-size) funnel around any nominal trajectory
    - Analysis extends to bounded disturbances
- Online:
    - Compute nominal trajectory such that funnel around it is collision-free
    - Receding horizon planning

[Singh, Majumdar, Slotine, Pavone '17 (Under Review)]

Example: Planar Quadrotor



6 states, 2 inputs



[Singh, Majumdar, Slotine, Pavone '17 (Under Review)]

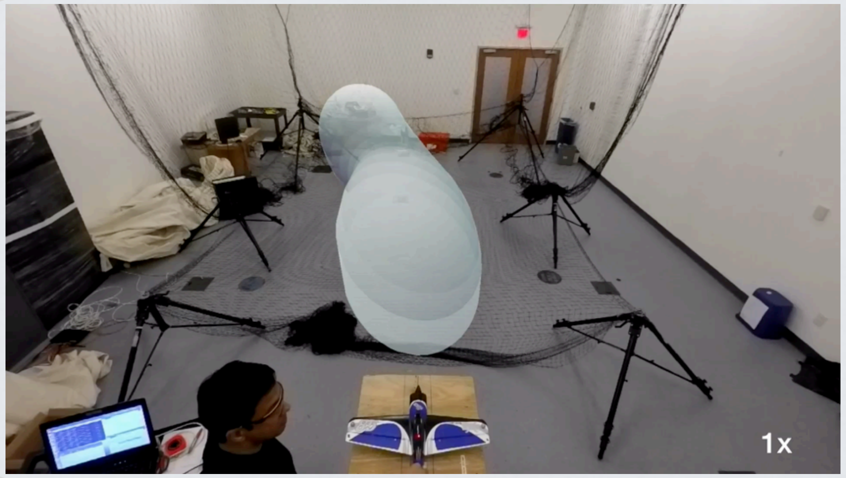
Challenges and Future Directions

- Sensing and estimation
    - Exciting new sensors (e.g., Intel's RealSense, FPGA stereo, sparse stereo, …)
    - How can we make guarantees with sensing?
- Real-time planning with probabilistic guarantees
    - e.g., won't collide with 0.95 probability (with stochastic wind gusts)
    - Using such certificates for real-time planning
- Formal/model-based tools and data-driven learning
    - How can we combine model-based tools with data-driven approaches?

Acknowledgements

- Collaborators:
  - Russ Tedrake, Amir Ali Ahmadi
  - Sumeet Singh, Marco Pavone, Jean-Jacques Slotine
- Funding sources:
  - ONR MURI grant N00014-09-1-1051
  - ONR Science of Autonomy, N00014-15-1-2673

Contributions and Future Work



Funnels and controllers

- Guarantees that system will remain within funnel
- Computed using powerful tools from SOS programming

Real-time planning using funnels

- Collision-free funnel guarantees safety
- Can handle complicated geometric constraints at runtime

Future work

- Guarantees with sensing/estimation
- Real-time planning with probabilistic guarantees
- Formal guarantees and learning

Tremendous potential to make robots operate safely in real environments