Gecko Phase 2 Experiments

Safe trajectory optimization with theoretical guarantees

Presenter: Somrita Banerjee
PI: Marco Pavone

Autonomous Systems Lab
Department of Aeronautics and Astronautics

Stanford University

Agenda

1. Quick recap of Phase 1 experiments
2. Why is trajectory optimization important for gecko grippers on Astrobee?
3. What are the ideal features of this trajectory optimization algorithm?
4. Presenting GuSTO
5. Overview of our plans for Phase 2 experiments

Agenda

1. **Quick recap of Phase 1 experiments**
2. Why is trajectory optimization important for gecko grippers on Astrobee?
3. What are the ideal features of this trajectory optimization algorithm?
4. Presenting GuSTO
5. Overview of our plans for Phase 2 experiments

Phase 1 – testing the gecko gripper

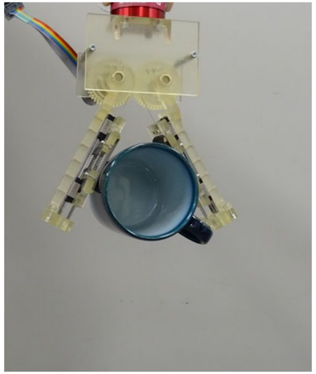
Agenda

1. Quick recap of Phase 1 experiments
2. **Why is trajectory optimization important for gecko grippers on Astrobee?**
3. What are the ideal features of this trajectory optimization algorithm?
4. Presenting GuSTO
5. Overview of our plans for Phase 2 experiments

Gecko grippers work best with stable positioning

Factors that improve performance of gecko grippers:

- **Large contact area** between gripper and target object
- **Uniform distribution of stress** across adhesives
- **Low relative velocity** between gripper and target object



The likelihood of these is increased when:

- Astrobee has **accurate and stable positioning** at its final goal state
- Astrobee **stays within known safety zones**, so gripper has no collisions or accidental grasps

Agenda

1. Quick recap of Phase 1 experiments
2. Why is trajectory optimization important for gecko grippers on Astrobee?
3. **What are the ideal features of this trajectory optimization algorithm?**
4. Presenting GuSTO
5. Overview of our plans for Phase 2 experiments

Features of an ideal trajectory optimization algorithm

Basic requirements:

- Operate over 6 DOF
- Run in real-time
- Incorporate obstacle avoidance
- Respect keep-in and keep-out zones

Provided by existing QP planner

Added requirements:

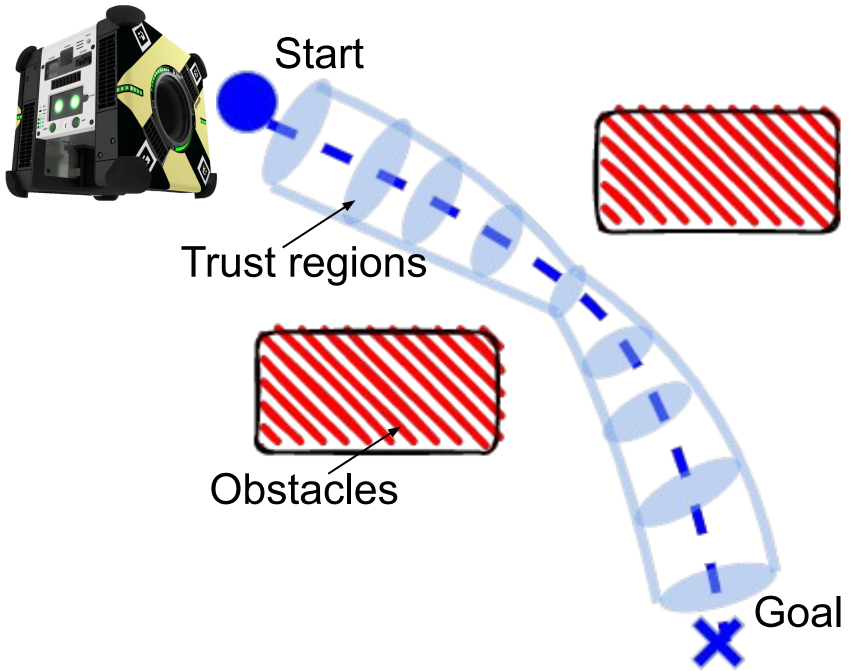
- Solve full non-convex optimization
- Have theoretical guarantees for optimality
- Establish “trust regions” for explainability and safety

Improvements due to GuSTO/SCP

Agenda

1. Quick recap of Phase 1 experiments
2. Why is trajectory optimization important for gecko grippers on Astrobee?
3. What are the ideal features of this trajectory optimization algorithm?
4. **Presenting GuSTO**
5. Overview of our plans for Phase 2 experiments

GuSTO: Trajectory optimization for Astrobee



GuSTO (Guaranteed Sequential Trajectory Optimization) via Sequential Convex Programming

Bonalli et al. ICRA, 2019

GuSTO was designed with Astrobee in mind

- **Real-time optimization** by convexifying problem and using convex solvers
- **Theoretically proven guarantees** on optimality of final trajectory
- **Enforce safety** using trust regions and keep-out zones

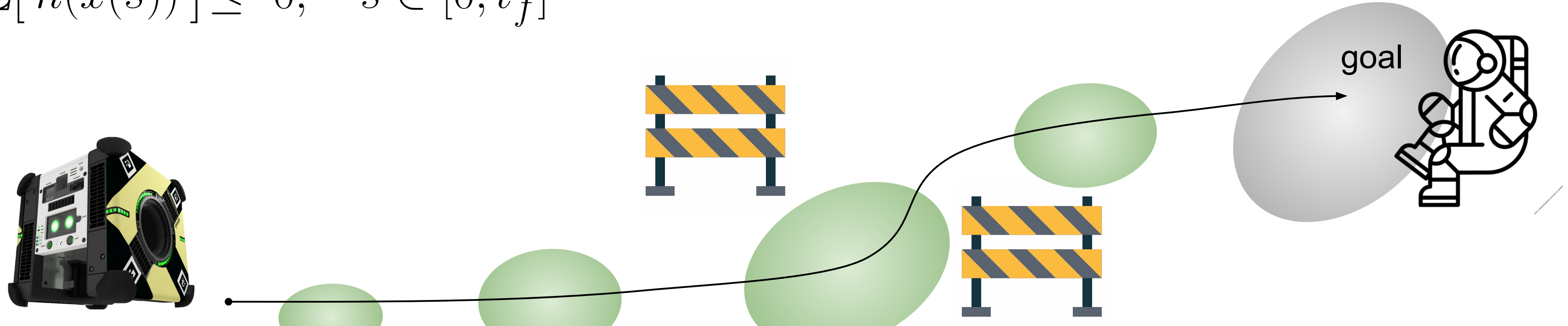
Mathematical formulation of the problem

$$\min_{u \in \mathcal{U}} \quad \mathbb{E}\left[\int_0^{t_f} f^0(s, u(s), x(s)) \, \mathrm{d}s \right]$$ ← Cost

$$\mathrm{d}x(s) = f(s, u(s), x(s)) \, \mathrm{d}s + \sigma(s, x(s)) \, \mathrm{d}B_s$$ ← Dynamics

$$x(0) = x^0, \quad \mathbb{E}[\, g(x(t_f)) \,] = 0$$ ← Initial / Final Conditions

$$\mathbb{E}[\, h(x(s)) \,] \leq 0, \quad s \in [0, t_f]$$ ← State Constraints



Bonalli, Riccardo, et al. "Sequential convex programming for non-linear stochastic optimal control." ESAIM: Control, Optimisation and Calculus of Variations 28 (2022): 64.

GuSTO is based on SCP

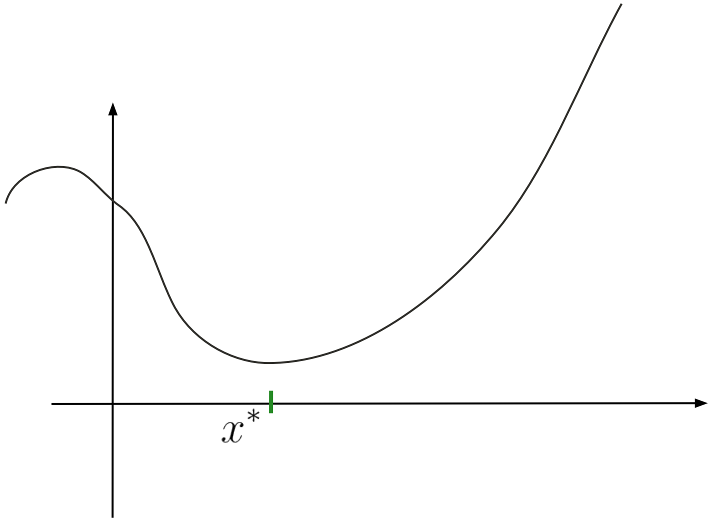
Sequential Convex Programming (SCP)

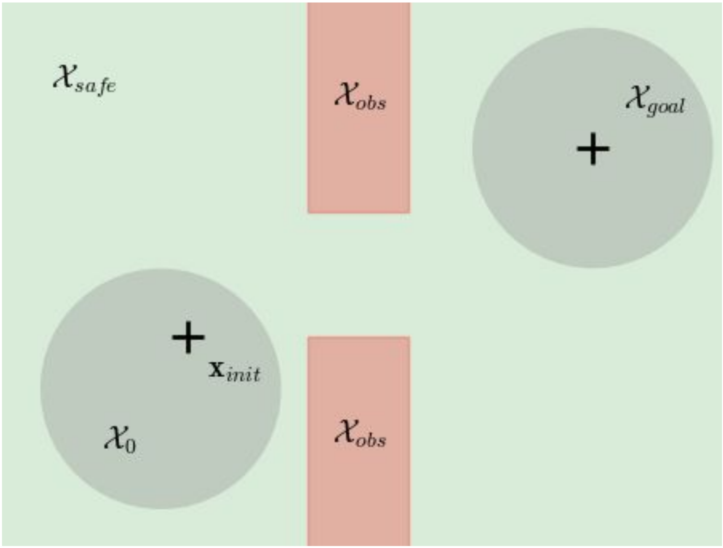
Solve a sequence of convex problems that stem from successive linearizations of a non-convex problem

Benefits:

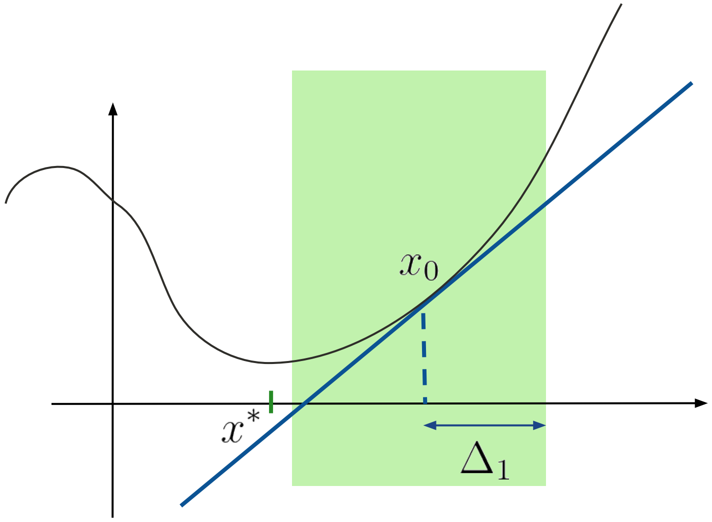
1) We can solve each convex subproblem efficiently
2) We can derive theoretical guarantees of convergence

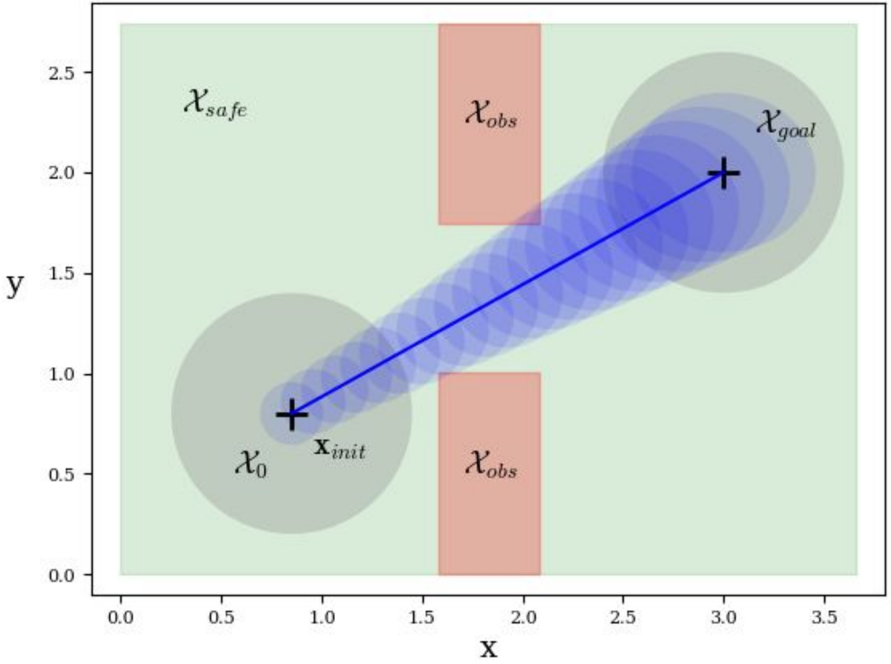
Intuitive introduction to SCP





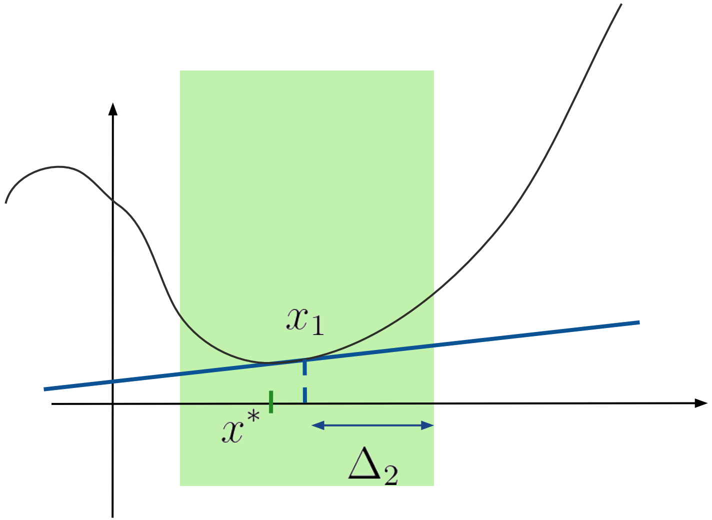
Intuitive introduction to SCP

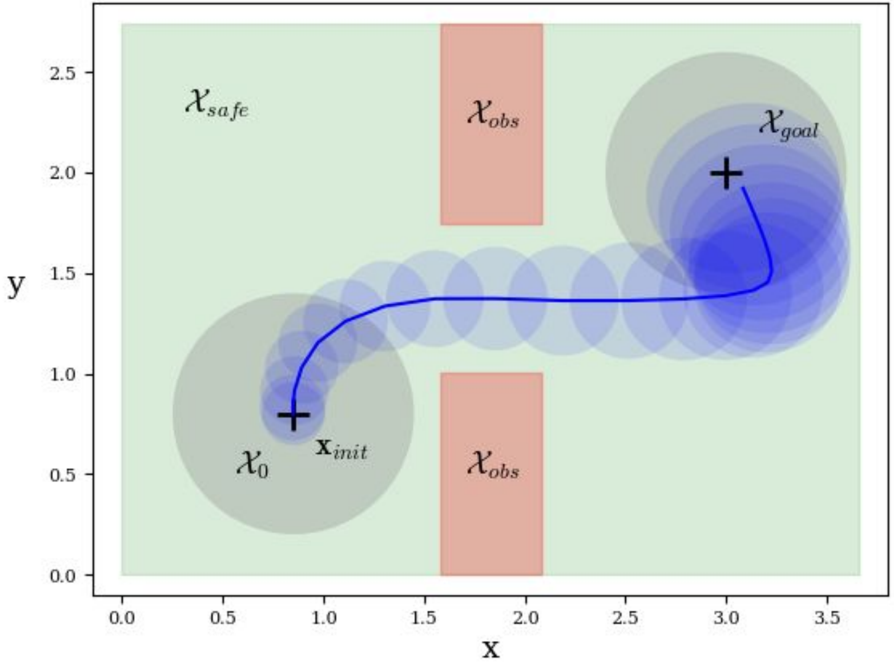




Intuitive introduction to SCP

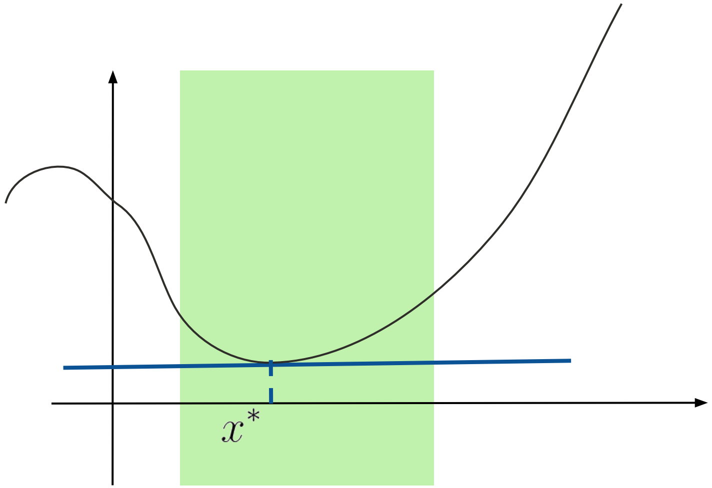
A few iterations later...



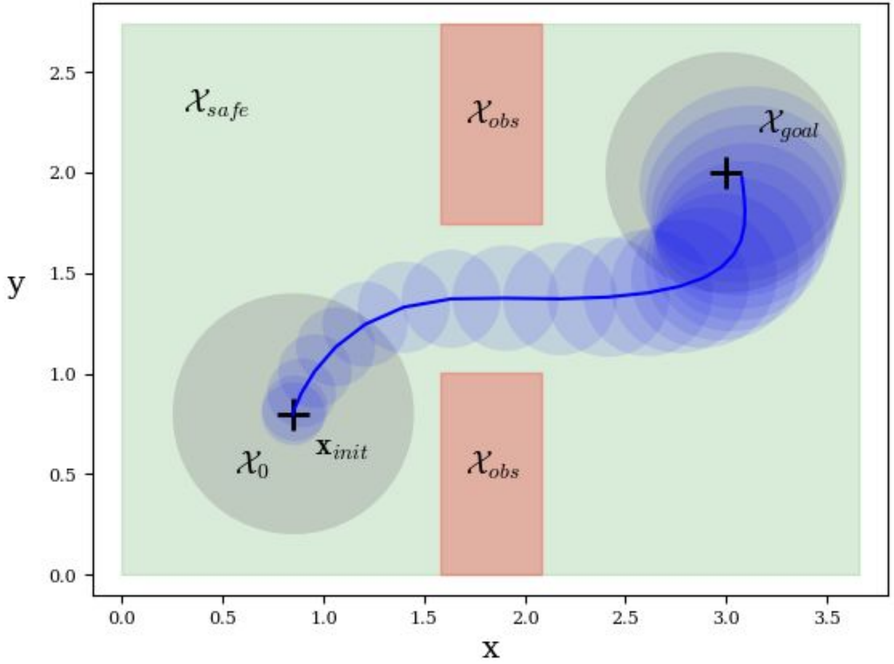


Intuitive introduction to SCP

A few iterations later...



Convergence!



3

GuSTO is SCP with additional features

Theoretical guarantee on convergence

- Under mild assumptions, SCP finds a local optimum for the original non-convex optimal control problem (OCP), in the sense of the Pontryagin Maximum Principle (PMP).

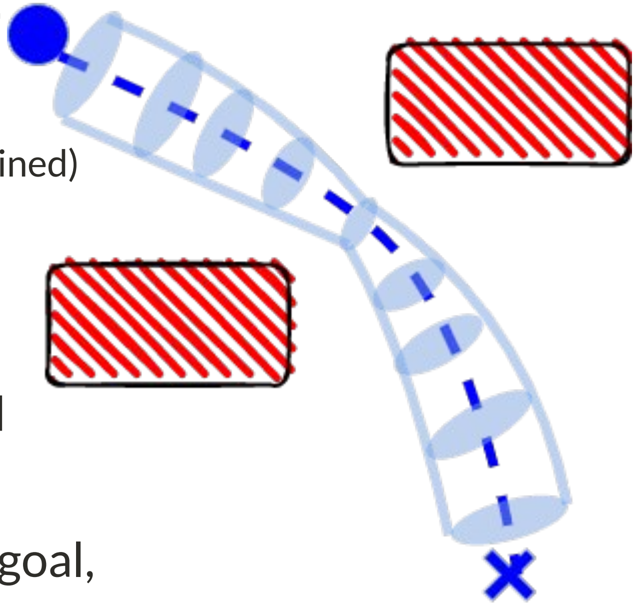
Updating rule for trust regions and constraint satisfaction

- A suitable choice of the maximal value of the trust region radius $\triangle_0$ may be crucial to allow the method to correctly explore the space if the provided initialization is far from any optimal strategy.
- Increasing the value of weights ω eases the search for solutions satisfying state constraints up to the ε tolerance.

Agenda

1. Quick recap of Phase 1 experiments
2. Why is trajectory optimization important for gecko grippers on Astrobee?
3. What are the ideal features of this trajectory optimization algorithm?
4. Presenting GuSTO
5. **Overview of our plans for Phase 2 experiments**

Envisioned experiment



- Software only (no gecko gripper)
- Events:
    - Within one section of the ISS
    - Astrobee moves from (pre-determined) start to (pre-determined) goal
    - *Virtual* (pre-determined) obstacles in the path
    - Reset and repeat ~3x
- Estimated 2 hours of crew time (including setup and teardown)
- Full success: Astrobee goes from each start to each goal, without entering virtual obstacle zones
    - Additional metrics: computation time, optimality of path

Fits seamlessly into mobility subsystem

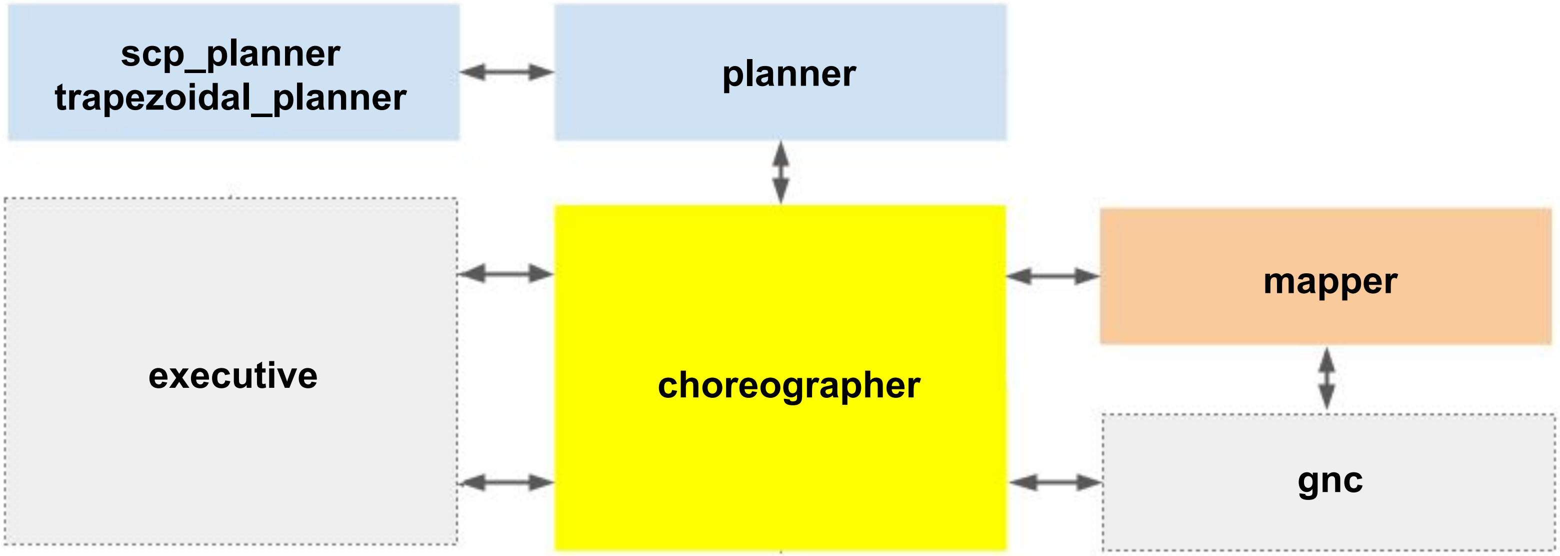
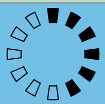
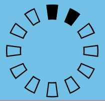
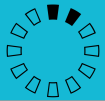
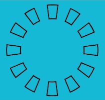
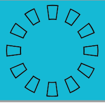


Image adapted from https://nasa.github.io/astrobee/html/mobility.html

Stages for Phase 2

		Status
Internal development	Development of GuSTO in C++	
	Addition of collision checking using Bullet	
	Development of test suite with Astrobee dynamics	
Integration	Integration of our code into NASA Astrobee repository	
	Packaging of our code	
Testing	Development of test procedures (ground and flight)	
	Granite table tests at Ames	
	Flight test on ISS	

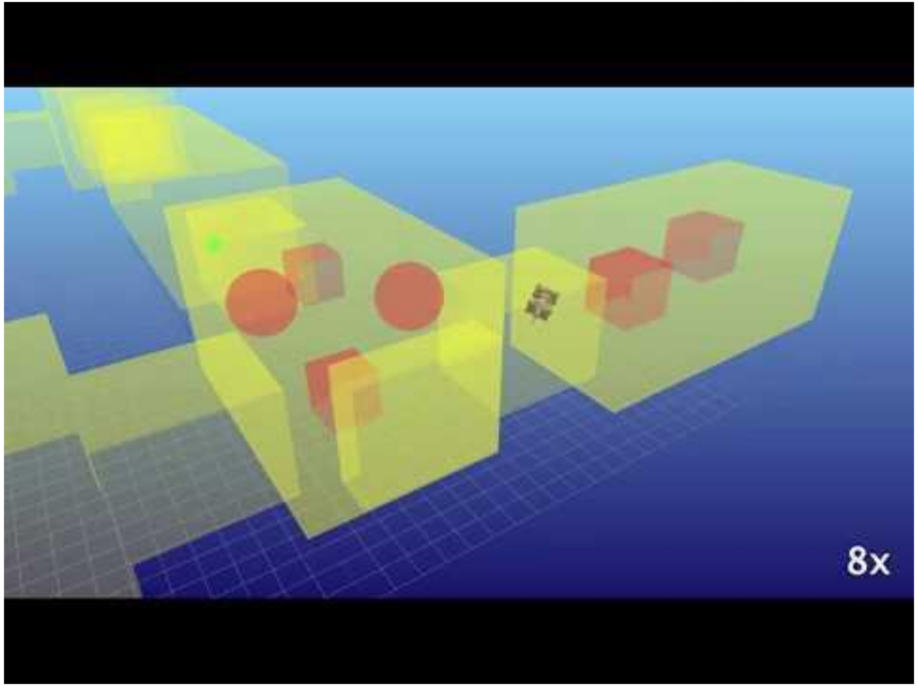
GuSTO: Trajectory optimization for Astrobee

Reference:
Bonalli, Riccardo, et al. "GuSTO: Guaranteed sequential trajectory optimization via sequential convex programming." ICRA, 2019.

Thanks to Jose Benavides, Cristian Garcia, Ruben Garcia Ruiz, Jose Cortez, Aric Katterhagen, Tyler Dorval, and the full Astrobee team.

Contact:
somrita.banerjee@nasa.gov or
somrita@stanford.edu

PI:
Marco Pavone, Stanford University



Astrobee aboard the International Space Station

Backup slides

Continuous-time SCP formulation

$$(\mathbf{OCP}) \quad \begin{aligned} &\min_{u \in \mathcal{U}} \quad \int_0^{t_f} u(s)^2 + h(x(s))\, \mathrm{d}s \\ &\dot{x}(s) = b(x(s), u(s)) \\ &\qquad \triangleq f_0(x(s)) + u(s) f_1(x(s)) \\ &x(0) = x^0, \quad g(x(t_f)) = 0 \end{aligned}$$

- The dynamics are control-affine (mandatory)
- Any state constraint is penalized within the cost (mandatory)
- The system is deterministic (not mandatory)
- The final time is fixed and $f^0(s, u, x) = u^2$ (not mandatory)

$$(\mathbf{COCP})_{k+1} \quad \begin{aligned} &\min_{u \in \mathcal{U}} \quad \int_0^{t_f} u(s)^2 + h(x_k(s)) + \frac{\partial h}{\partial x}(x_k(s))(x(s) - x_k(s))\, \mathrm{d}s \\ &\dot{x}(s) = b(x_k(s), u(s)) + \frac{\partial b}{\partial x}(x_k(s), u_k(s))(x(s) - x_k(s)) \\ &x(0) = x^0, \quad g(x_k(t_f)) + \frac{\partial g}{\partial x}(x_k(t_f))(x(t_f) - x_k(t_f)) = 0 \end{aligned}$$

Be careful: the linearization makes sense only locally. Add trust-region constraints:

$$\int_0^{t_f} \|x(s) - x_k(s)\|^2\, \mathrm{d}s \leq \Delta_{k+1}$$

$$\Delta_{k+1} \in \mathbb{R}_+, \quad \Delta_{k+1} \longrightarrow 0$$