

Lecture 5

ROB-GY 7863 / CSCI-GA 3033 7863:  
Planning, Learning, and Control for Space  
Robotics

Benjamin Riviere

October 6, 2025

# Logistics

- ▶ Project 1 Report/Presentation are due next week **Tuesday**

## Correction from Last time

- Derivatives of  $\phi_\mu$ :

$$\phi_\mu = -\mu \sum_i \log(-g_i(x))$$

$$\nabla \phi_\mu = -\mu \sum_i \frac{1}{g_i(x)} \nabla g_i(x)$$

$$\nabla^2 \phi_\mu = -\mu \sum_i \left( \frac{1}{g_i(x)} \nabla^2 g_i(x) - \frac{1}{g_i(x)^2} \nabla g_i(x) \nabla g_i(x)^T \right)$$

- Inside feasible region  $g_i(x) < 0$ , so  $\nabla^2 \phi_\mu(x) \succ 0$  bc  $g_i$  are convex and outer products are always positive semi-definite.
- Updated Lecture 4 are posted

# Space Culture

- ▶ Benefits of Microgravity: [2022](#) and [2023](#)
- ▶ [In Space Production Applications](#)
- ▶ [Merck Case Study](#)
- ▶ Job search idea: make a list of the companies that have recently gotten grants and write to them directly with a well-researched cover letter. [Previous Nasa awards](#).

# Recap Last Week I

- ▶ Mathematical Optimization:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \leq 0, \quad i \in \{1, \dots, m\} \\ & h_i(x) = 0, \quad i \in \{1, \dots, p\} \end{aligned}$$

- ▶ Convex Functions:

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y)$$

- ▶ Unconstrained minimization:

$$x_{k+1} = x_k - \alpha \nabla f(x_k) \quad \|x_k - x^*\| \leq \rho^k \|x_0 - x^*\|$$

- ▶ Newton's method Desoer et al., [1972](#):  $\forall x \in B$

$$x_{k+1} = x_k - \alpha (\nabla^2 f)^{-1} \nabla f(x_k) \quad \|x_k - x^*\| \leq \rho^k \|x_0 - x^*\|^2$$

## Recap Last Week II

► Equality Constraints:

$$x_{k+1} = x_k - \alpha \nabla_x L(x_k)$$

$$\lambda_{k+1} = \lambda_k + \beta \nabla_y L(x_k) \quad \|x_k - x^*\| \leq \rho^k \|x_0 - x^*\|$$

► Inequality constraints:

$$\phi(x) = -\mu \sum_i \log(-g_i(x))$$

$$f_\mu(x) = f(x) + \phi(x)$$

$$x_{k+1} = x_k - \alpha \nabla_x L_\mu(x_k)$$

$$\lambda_{k+1} = \lambda_k + \beta \nabla_y L_\mu(x_k) \quad \|x_k - x^*\| \leq \rho^k \|x_0 - x^*\|$$

# Agenda

- ▶ Coding Exercises
- ▶ Break
- ▶ Rocket Landings
- ▶ Swarm Guidance

# Unconstrained Minimization

- ▶ Define problem (recall we require  $\nabla^2 f$  is  $m$ -strongly convex and  $M$ -smooth):

$$\min_x f(x) = \min_x \frac{1}{2} x^T A x + b^T x + c$$

- ▶ Define contraction-based learning rate:  $\alpha = \frac{2}{M+m}$ ,  $\rho = \frac{\kappa-1}{\kappa+1}$
- ▶ Compare to standard learning rate:  $\alpha = \frac{1}{M}$ ,  $\rho = \frac{\kappa-1}{\kappa}$
- ▶ Compare to analytical solution:

$$x^* = -A^{-1}b, \quad f(x^*) = -\frac{1}{2}b^T A^{-1}b + c$$

- ▶ Implement gradient descent :

$$x_{k+1} = x_k - \alpha \nabla f(x_k)$$

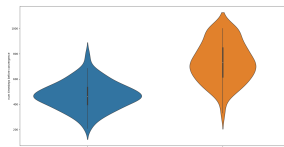
# My solution

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn

def algo_unconstrained_gradient_descent(x0, f, nabla_f, M, lr="contraction"):
    if lr == "contraction":
        alpha = 2/(M-m)
        rho = (M - m)/(M + m)
    elif lr == "classical":
        alpha = 1/M
        rho = 1 - m/M
    xs = [x0]
    for k in range(1000):
        grad = nabla_f(xs[-1])
        xs.append(xs[-1] - alpha * grad)
        if np.linalg.norm(grad) < 1e-6:
            break
    fxs = [f(x) for x in xs]
    return xs, fxs, rho

def plot_one_experiment(xs, fxs, rho, title):
    fig, ax = plt.subplots()
    ax.plot(fxs - fxs[-1])
    # ax.plot(fxs)
    ax.plot([fxs[0] + rho**k for k in range(len(fxs))], '--')
    ax.set_title(title)
    ax.set_xlabel("iteration")
    ax.set_ylabel("f(x_k) - f**")
    ax.grid()

def plot_stats(k0):
    fig, ax = plt.subplots()
    xs_0p = np.array(k0)
    seaborn.violinplot(data=xs_0p, alpha=0.5)
    ax.set_xticks([0, 1], ["contraction", "classical"])
    ax.set_ylabel("num timesteps before convergence")
```



- Why is analysis important? Without it, we would have to tune  $\alpha$  for every problem, which will only get harder once we add constraints.

# Equality Constrained Minimization I

- Define problem (recall we require  $\nabla^2 f$  is  $m$ -strongly convex and  $M$ -smooth):

$$\begin{aligned} \min_x \quad & \frac{1}{2}x^T Ax + b^T x + c \\ \text{s.t.} \quad & Dx = g \end{aligned}$$

- Define Lagrangian:

$$\begin{aligned} L(x, \lambda) &= x^T Ax + b^T x + \lambda^T (Dx - g) \\ L_\gamma(x, \lambda) &= L(x, \lambda) + \frac{\gamma}{2} \|Dx - g\|^2 \\ L_{\gamma, \eta}(x, \lambda) &= L_\gamma(x, \lambda) - \frac{\eta}{2} \|\lambda\|_2^2 \end{aligned}$$

# Equality Constrained Minimization II

- Define contraction-based learning rate:

$$\alpha = \frac{2}{M_\gamma + m_\gamma}, \quad \rho_x = \frac{\kappa_x - 1}{\kappa_x + 1}$$
$$\beta = \frac{2}{\sigma_{\max}^2(D) + \sigma_{\min}^2(D)}, \quad \rho_y = \frac{\kappa_y - 1}{\kappa_y + 1}$$

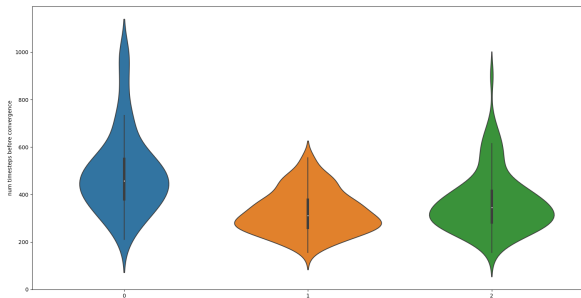
where  $M_\gamma = \lambda_{\max}(\nabla^2 f) + \gamma \lambda_{\max}(D^T D)$  and  $m_\gamma = \lambda_{\min}(\nabla^2 f) + \gamma \lambda_{\min}(D^T D)$ .

- Implement primal dual

$$x_{k+1} = x_k - \alpha \nabla_x L(x_k)$$

$$\lambda_{k+1} = \lambda_k + \beta \nabla_y L(x_k)$$

# My solution



# Inequality Constrained Minimization I

- Define problem (recall we require  $\nabla^2 f$  is  $m$ -strongly convex and  $M$ -smooth):

$$\begin{aligned} \min_x \quad & \frac{1}{2}x^T A x + b^T x + c \\ \text{s.t.} \quad & D x = g, \quad E x + h \leq 0 \end{aligned}$$

- Define Lagrangian with  $\mu$ -barrier:

$$L_{\gamma, \eta, \mu}(x, \lambda) = L_{\gamma, \eta}(x, \lambda) - \mu \sum_i \log((-E x - h)_i)$$

- Define contraction-based learning rate:

$$\begin{aligned} \alpha &= \frac{2}{M_{\gamma} + m_{\gamma}}, \quad \rho_x = \frac{\kappa_x - 1}{\kappa_x + 1} \\ \beta &= \frac{2}{\sigma_{\max}^2(D) + \sigma_{\min}^2(D)}, \quad \rho_y = \frac{\kappa_y - 1}{\kappa_y + 1} \end{aligned}$$

# Inequality Constrained Minimization II

- Implement barrier method

$$x_{k+1} = x_k - \alpha \nabla_x L_\mu(x_k)$$

$$\lambda_{k+1} = \lambda_k + \beta \nabla_y L_\mu(x_k)$$

$$\mu_{k+1} = \begin{cases} t\mu_k & \text{if inner loop converged} \\ \mu_k & \text{else} \end{cases}$$

where  $t \in (0, 1)$ .

# My solution I

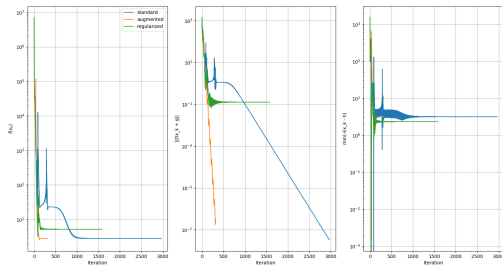
- ▶ Strictly feasible initialization via Analytic Center method, i.e.  $x_0$  is the solution of:

$$\begin{aligned} \min_x \quad & - \sum_i \log(-(Ex + h)_i) \\ \text{s.t.} \quad & Dx = g \end{aligned}$$

- ▶ Monotonic objective transformation:

$$f'_\mu = \frac{1}{\mu}(f(x) + \phi(x)) \tag{1}$$

# My solution II



- Works for about 50% of cases

# Notes

- ▶ We have derived and implemented the basic mechanism of first-order solvers for a wide class of problems. Nice job!
- ▶ Please do not use these solvers in practice!
- ▶ If MUJOCO:physics-based simulation, then e.g.  
GUROBI:mathematical optimization
- ▶ Solvers are products written by experts that use a lot of extra tricks (sparsity, acceleration, second-order information, etc.). Use a toolbox (CVX), do research based on your problem setting, and pick one that matches your problem type.

## Extension 1: Sequential Convex Programming

- Sometimes problems are not convex. For example, fixed-speed car dynamics:

$$\frac{d}{dt} \begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} + \Delta t \begin{bmatrix} v \cos \theta_k \\ v \sin \theta_k \\ \omega_k \end{bmatrix}$$

- If we linearize about an initial guess, we can make a new optimization problem that is a local approximation to the original problem and that is convex. We denote the initial guess solution  $\bar{x}$ . In the case of dynamics, we approximate  $h$  with a Taylor expansion:

$$\hat{h}(x) = h(\bar{x}) + \nabla h_{x=\bar{x}}^T (x - \bar{x})$$

Note that  $\hat{h}(x)$  is linear in  $x$  (but nonlinear in  $\bar{x}$ ).

- Sequential convex programming is when we set solution of optimization problem to initial guess of next iteration.

## Extension 2: Model Predictive Control

- Optimal Control Problem: Given  $x_0$ , solve:

$$\begin{aligned} x_{1:K}^*, u_{1:K}^* = \min_{x_{1:K}, u_{1:K}} \sum_k x_k^T Q x_k + u_k^T R u_k + \phi(x_K) \\ \text{s.t. } x_{k+1} = A x_k + B u_{k+1}, \forall k \in \{1, \dots, K\} \end{aligned}$$

- Model Predictive Control: Rollout only first state/action pair and re-solve optimal control problem from  $x_1^*$ :

$$\begin{aligned} x_{2:K+1}^*, u_{2:K+1}^* = \min_{x_{2:K+1}, u_{2:K+1}} \sum_k x_k^T Q x_k + u_k^T R u_k + \phi(x_K) \\ \text{s.t. } x_{k+1} = A x_k + B u_{k+1}, \forall k \in \{2, \dots, K+1\} \end{aligned}$$

- General form from  $x_t$ :

$$\begin{aligned} x_{t+1:K+t}^*, u_{t+1:K+t}^* = \min_{x_{t+1:K+t}, u_{t+1:K+t}} \sum_k x_k^T Q x_k + u_k^T R u_k + \phi(x_K) \\ \text{s.t. } x_{k+1} = A x_k + B u_{k+1}, \forall k \in \{t+1, \dots, K+t\} \end{aligned}$$

# Break

# Powered Guided Descent Problem I

- We are using this reference: Açıkmüş et al., 2007.

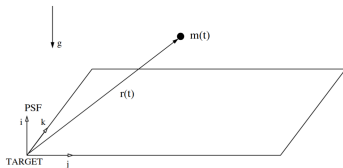
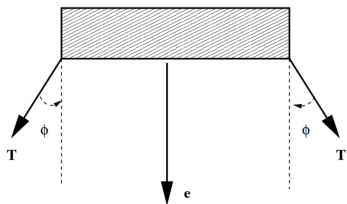


Fig. 1 Surface fixed coordinate frame.



Net thrust direction  
Fig. 2 Spacecraft geometry.

# Powered Guided Descent Problem II

- ▶ Decision variables: thrust profile  $T_c(t)$ , terminal time  $t_f$
- ▶ Objective: minimum fuel  $\int_t \|T_c(t)\|^2$  where  $T_c$  is net thrust vector
- ▶ Dynamics:

$$\ddot{r}(t) = g + \frac{T_c(t)}{m(t)} \quad (2)$$

$$\dot{m}(t) = -\alpha \|T_c(t)\| \quad (3)$$

where  $r$  is position of spacecraft,  $g$  is acceleration vector of planet,  $m$  is mass,  $\alpha$  is a constant of fuel consumption.

# Powered Guided Descent Problem III

- Thrust limits:

$$nT_1 \cos \phi = \rho_1 \leq \|T_c\| \leq \rho_2 = nT_2 \cos \phi \quad (4)$$

"The thrusters cannot be turned off once they are initiated during the maneuver. Further, there are minimum thrust levels below which the thrusters can not operate reliably. Because the spacecraft is modeled as a point mass with a single thruster, there is a lower nonzero bound on the thrust magnitude that defines a nonconvex feasible region in the control space."

- Initial and final conditions:

$$m(0) = m_{\text{wet}}, \quad r(0) = r_0, \quad \dot{r}(0) = \dot{r}_0, \quad r(t_f) = 0, \quad \dot{r}(t_f) = 0 \quad (5)$$

# Powered Guided Descent Problem IV

- ▶ General SOC constraints:

$$\|Sx(t) - v_j\| + c_j^T x(t) + a_j \leq 0 \quad \text{soc constraint} \quad (6)$$

- ▶ Particular rocket constraints:

$$r_1(t) > 0 \quad \text{no surface penetration} \quad (7)$$

$$\|\dot{r}(t)\| \leq V \quad \text{max speed} \quad (8)$$

$$\theta \leq \bar{\theta} \quad \text{glide slope} \quad (9)$$

where  $x = [r, \dot{r}]$ .

# Powered Guided Descent Problem V

*Problem 1:*

$$\begin{aligned} \max_{t_f, T_c(\cdot)} m(t_f) &= \min_{t_f, T_c(\cdot)} \int_0^{t_f} \|T_c(t)\| dt \\ \text{subject to } \ddot{r}(t) &= g + T_c(t)/m(t), \quad \dot{m}(t) = -\alpha \|T_c(t)\| \\ 0 < \rho_1 &\leq \|T_c(t)\| \leq \rho_2, \quad r_1(t) \geq 0 \\ \|S_j x(t) - v_j\| + c_j^T x(t) + a_j &\leq 0, \quad j = 1, \dots, n_s \\ m(0) &= m_{\text{wet}}, \quad r(0) = r_0, \quad \dot{r}(0) = \dot{r}_0 \\ r(t_f) &= \dot{r}(t_f) = 0 \end{aligned}$$

# Powered Guided Descent Problem VI

- ▶ Nonconvex problem because  $S$  is nonconvex.

$$P1: \min_x f(x) \text{ s.t. } x \in S \quad (10)$$

- ▶ Convex relaxation because  $C$  is convex.

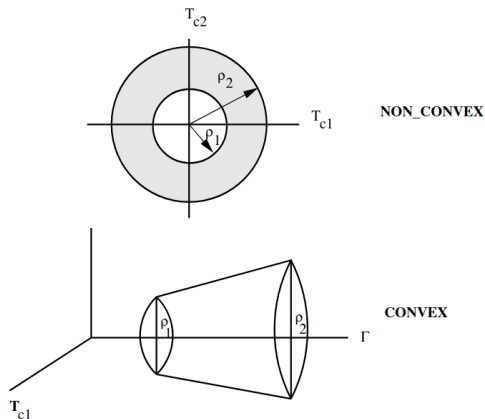
$$P2: \min_x f(x) \text{ s.t. } x \in C \quad (11)$$

- ▶ If we can show that the optimal and feasible solution of P2 is also the optimal and feasible solution of P1, we can solve P2 instead of P1.
- ▶ Convex Relaxation Example: Assignment problem

$$\min_x c^T x \text{ s.t. } Ax + b \leq 0, x_i \in \{0, 1\} \quad (12)$$

$$\min_x c^T x \text{ s.t. } Ax + b \leq 0, x_i \in [0, 1] \quad (13)$$

# Powered Guided Descent Problem VII



**Fig. 3 Convexification of thrust magnitude constraint.**

# Powered Guided Descent Problem VIII

*Problem 2:*

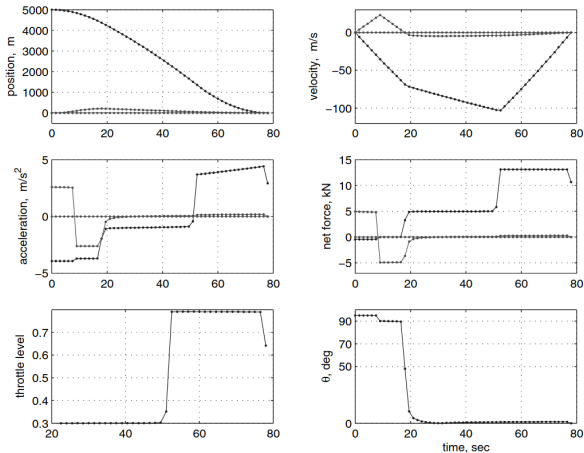
$$\min_{t_f, T_c(\cdot), \Gamma(\cdot)} \int_0^{t_f} \Gamma(t) dt \quad \text{subject to } \dot{m}(t) = -\alpha \Gamma(t) \quad (13)$$

$$\|T_c(t)\| \leq \Gamma(t) \quad (14)$$

$$0 < \rho_1 \leq \Gamma(t) \leq \rho_2 \quad (15)$$

$$\begin{aligned} \ddot{r}(t) &= g + T_c(t)/m(t), & r_1(t) &\geq 0 \\ \|S_j x(t) - v_j\| + c_j^T x(t) + a_j &\leq 0, & j &= 1, \dots, n_s \\ m(0) &= m_{\text{wet}}, & r(0) &= r_0, & \dot{r}(0) &= \dot{r}_0 \\ r(t_f) &= \dot{r}(t_f) = 0 \end{aligned}$$

# Powered Guided Descent Problem IX



# Swarm Guidance Problem I

- ▶ We use references Morgan et al., [2014](#) and Morgan et al., [2016](#).

# Swarm Guidance Problem II

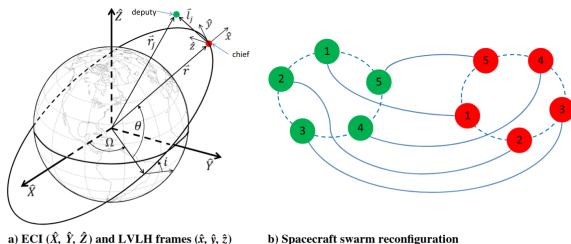


Fig. 1 Visualization of the relative coordinate system and a spacecraft swarm reconfiguration [3].

# Swarm Guidance Problem III

*Problem 1:* The nonlinear optimal control problem is

$$\min_{u_j, j=1, \dots, N} \sum_{j=1}^N \int_0^{t_f} \|u_j(t)\|_p dt \text{ subject to Eq.(1)} \quad (3)$$

and

$$\|u_j(t)\|_q \leq U_{\max} \quad \forall t \in [0, t_f], \quad j = 1, \dots, N \quad (4)$$

$$\begin{aligned} \|C[x_j(t) - x_i(t)]\|_2 &\geq \bar{R}_{\text{col}} \quad \forall t \in [0, t_f], \\ i &> j, \quad j = 1, \dots, N-1 \end{aligned} \quad (5)$$

$$x_j(0) = x_{j,0}, \quad x_j(t_f) = x_{j,f} \quad j = 1, \dots, N \quad (6)$$

$$\ddot{\ell}_j = -2S(\omega)\dot{\ell}_j - g(\ell_j, \omega) + u_j \quad (1)$$

# Swarm Guidance Problem IV

$$\dot{\mathbf{x}}_j = \mathbf{f}(\mathbf{x}_j, \boldsymbol{\alpha}) + \mathbf{B}\mathbf{u}_j \quad (7)$$

where  $\mathbf{B} = [\mathbf{0}_{3 \times 3} \quad \mathbf{I}_{3 \times 3}]^T$ . Linearizing Eq. (7) yields

$$\dot{\mathbf{x}}_j = \mathbf{A}(\bar{\mathbf{x}}_j, \boldsymbol{\alpha})\mathbf{x}_j + \mathbf{B}\mathbf{u}_j + \mathbf{c}(\bar{\mathbf{x}}_j, \boldsymbol{\alpha}) \quad (8)$$

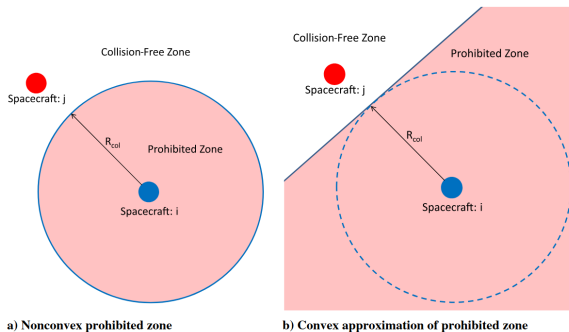
$$\begin{aligned} \mathbf{x}_j[k+1] &= \mathbf{A}_j[k]\mathbf{x}_j[k] + \mathbf{B}_j[k]\mathbf{u}_j[k] + \mathbf{c}_j[k], \\ k &= 0, \dots, T-1, \quad j = 1, \dots, N \end{aligned} \quad (11)$$

where  $\mathbf{x}_j[k] = \mathbf{x}_j(t_k)$ ,  $\mathbf{u}_j[k] = \mathbf{u}_j(t_k)$ ,  $\boldsymbol{\alpha}[k] = \boldsymbol{\alpha}(t_k)$ , and

$$\begin{aligned} \mathbf{A}_j[k] &= e^{\mathbf{A}(\bar{\mathbf{x}}_j(t_k), \boldsymbol{\alpha}(t_k))\Delta t}, \quad \mathbf{B}_j[k] = \int_0^{\Delta t} e^{\mathbf{A}(\bar{\mathbf{x}}_j(t_k), \boldsymbol{\alpha}(t_k))\tau} \mathbf{B} \, d\tau, \\ \mathbf{c}_j[k] &= \int_0^{\Delta t} e^{\mathbf{A}(\bar{\mathbf{x}}_j(t_k), \boldsymbol{\alpha}(t_k))\tau} \mathbf{c}(\bar{\mathbf{x}}_j(t_k), \boldsymbol{\alpha}(t_k)) \, d\tau \end{aligned} \quad (12)$$

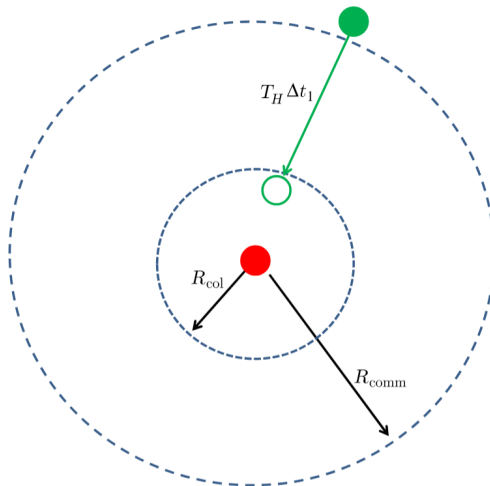
$$\begin{aligned} \mathbf{A}(\bar{\mathbf{x}}_j, \boldsymbol{\alpha}) &= \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_3 \\ -\left. \frac{\partial \mathbf{g}}{\partial \mathbf{x}_j} \right|_{\bar{\mathbf{x}}_j} & -2\mathbf{S}(\boldsymbol{\omega}) \end{bmatrix}, \\ \mathbf{c}(\bar{\mathbf{x}}_j, \boldsymbol{\alpha}) &= \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ -\mathbf{g}(\bar{\boldsymbol{\ell}}_j, \boldsymbol{\alpha}) + \left. \frac{\partial \mathbf{g}}{\partial \mathbf{x}_j} \right|_{\bar{\mathbf{x}}_j} \bar{\mathbf{x}}_j \end{bmatrix} \end{aligned}$$

# Swarm Guidance Problem V



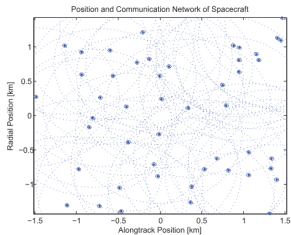
**Fig. 2** Convexification of the two-dimensional (2-D) collision-avoidance constraint.

# Swarm Guidance Problem VI

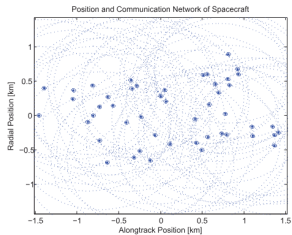


**Fig. 8** Illustration of a pair of spacecraft that do not have sufficient communication radii to guarantee detectable collisions.

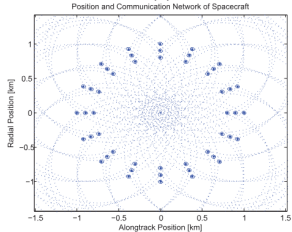
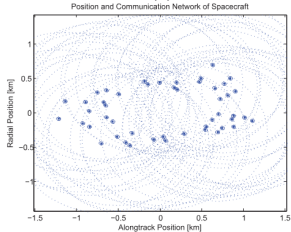
# Swarm Guidance Problem VII



(a)



(b)



## Autonomous In-Orbit Satellite Assembly from a Modular Heterogeneous Swarm using Sequential Convex Programming

Rebecca C. Foust\*

*University of Illinois at Urbana-Champaign, Urbana, IL, 61801, USA*

Soon-Jo Chung<sup>†</sup>

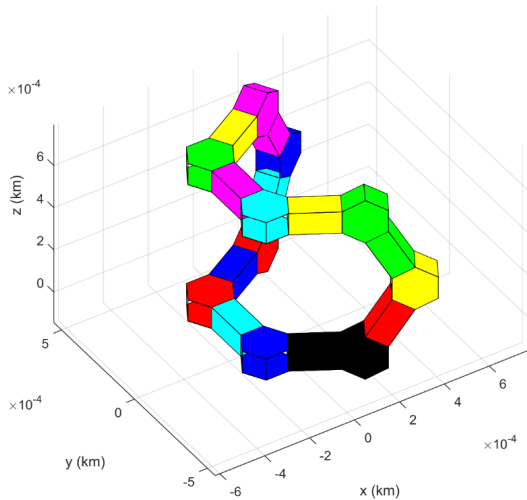
*California Institute of Technology, Pasadena, CA, 91125, USA*

Fred Y. Hadaegh<sup>‡</sup>




*Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 91109, USA*

Reference Foust et al., [2016](#)



# In orbit assembly II



# References I

-  Açıkmeşe, Behçet and Scott R. Ploen (2007). “Convex Programming Approach to Powered Descent Guidance for Mars Landing”. In: *Journal of Guidance, Control, and Dynamics* 30.5, pp. 1353–1366. DOI: [10.2514/1.27553](https://doi.org/10.2514/1.27553) (cit. on p. 21).
-  Desoer, C. and H. Haneda (1972). “The measure of a matrix as a tool to analyze computer algorithms for circuit analysis”. In: *IEEE Transactions on Circuit Theory* 19.5, pp. 480–486. DOI: [10.1109/TCT.1972.1083507](https://doi.org/10.1109/TCT.1972.1083507) (cit. on p. 5).
-  Foust, Rebecca, Soon-Jo Chung, and Fred Hadaegh (2016). “Autonomous in-orbit satellite assembly from a modular heterogeneous swarm using sequential convex programming”. In: *AIAA/AAS Astrodynamics Specialist Conference*, p. 5271 (cit. on p. 37).

## References II

-  Morgan, Daniel, Soon-Jo Chung, and Fred Y Hadaegh (2014). “Model predictive control of swarms of spacecraft using sequential convex programming”. In: *Journal of Guidance, Control, and Dynamics* 37.6, pp. 1725–1740 (cit. on p. 30).
-  Morgan, Daniel, Giri P Subramanian, Soon-Jo Chung, and Fred Y Hadaegh (2016). “Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and sequential convex programming”. In: *The International Journal of Robotics Research* 35.10, pp. 1261–1285 (cit. on p. 30).