Applied Spacecraft Trajectory Optimization: Lecture 1

Erica Jenson

ASEN 6020 Guest Lectures

Spring 2022

My Background

Celestial and Spaceflight Mechanics Laboratory

- 2017, B.S. Aerospace & Mechanical Engineering, University of Florida
- 2019, M.S. Aerospace Engineering Sciences, CU Boulder
- Current PhD candidate

Research Experience:

- Stochastic optimal control, spacecraft guidance
- Analytical measures of nonlinearity and their application to GN&C
- Collaboration with NASA JPL and Goddard Space Flight Center

Overview

Celestial and Spaceflight Mechanics Laboratory

- Lecture 1: Homotopy Methods and Multi-Objective Optimization
    - Detailed indirect optimization + homotopy example
    - Tips for class project
    - Scalarized methods vs. evolutionary algorithms for MOO
- Lecture 2:
    - Optimization Under Uncertainty
        - Types of uncertainty/stochastics
        - Notation & terminology
        - Techniques
    - Dynamic programming
        - LQR from dynamic programming
        - DDP, HDDP, SDDP

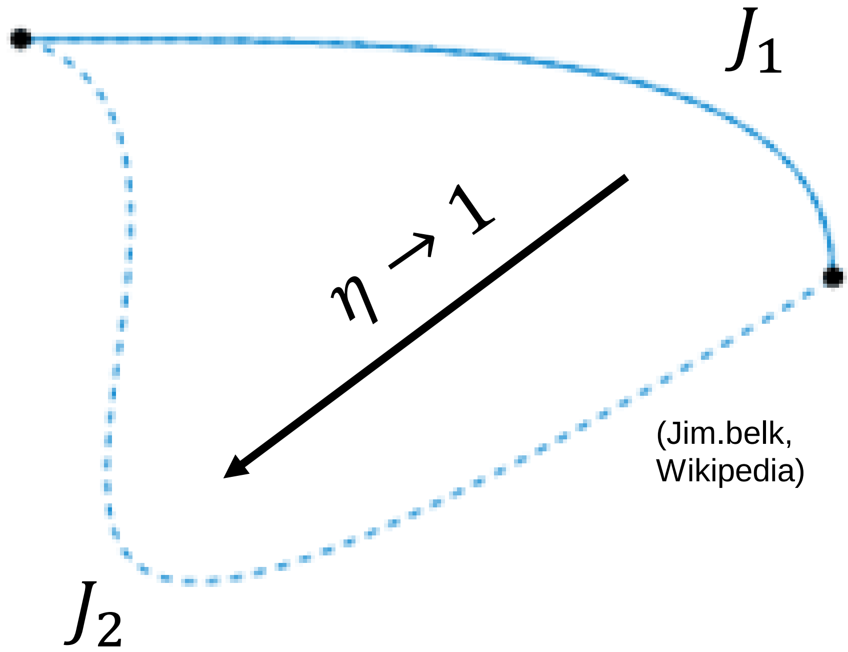
Homotopy/Continuation Methods

(and some indirect optimization examples)

Homotopy/Continuation

- A concept in topology
- Continuous deformation from one function to another
- Used *informally* in the optimization literature
- Helps generate initial guess
- For example:
  - we want to optimize J_2
  - difficult to find good initial guess
  - but we *can* optimize J_1
  - transform the solution from J_1 to J_2

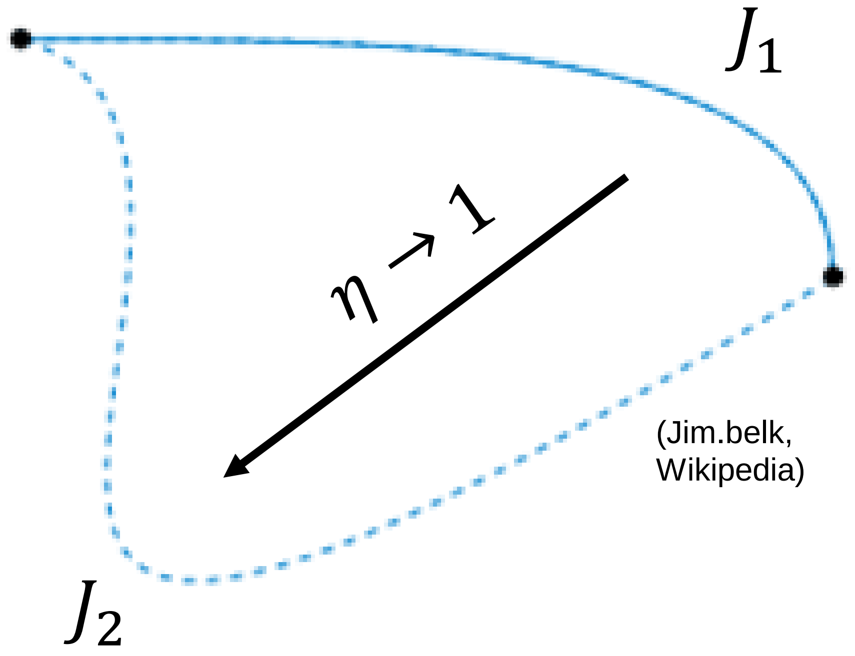
$$J = (1-\eta)J_1 + \eta J_2$$



Homotopy Use Cases

- Indirect methods, direct methods, & other
- Different cost functions
- Varying constraints
- Different dynamics
  - e.g., solve with 2BP first, then “turn on” a dynamical perturbation
  - improve accuracy of integrator, ephemeris model, etc.

$$J = (1-\eta)J_1 + \eta J_2$$



(Jim.belk, Wikipedia)

Different Cost Functions: Energy vs. Mass

- Minimize energy
  - squared L^2 norm of control
  - Continuous control profile
  - Less sensitive, easier to find solutions

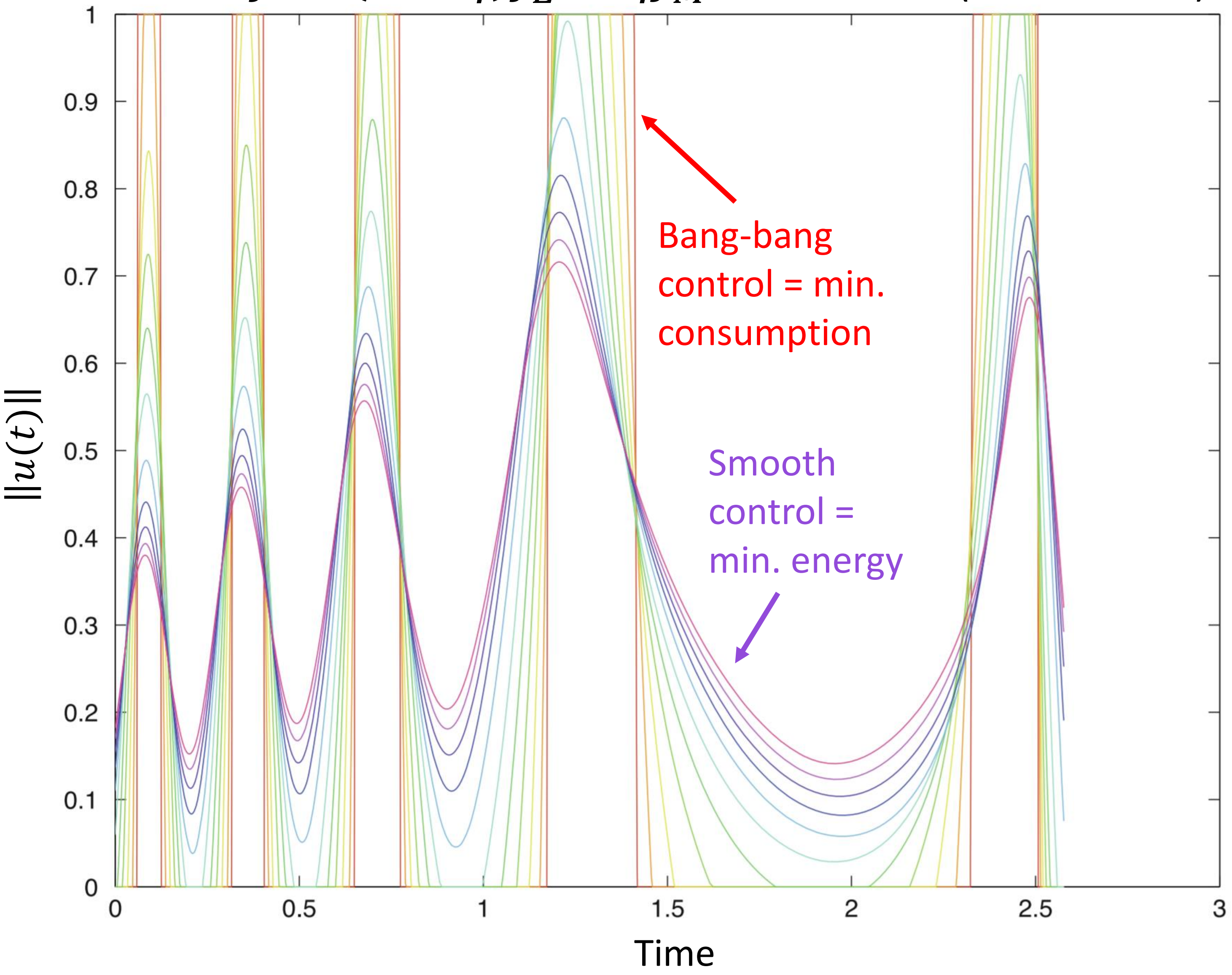
$$J_E = \int_{t_0}^{t_f} \|u(t)\|^2 dt = \int_{t_0}^{t_f} u(t)^T u(t) dt$$

- Minimize consumption
  - aka "minimize fuel," "minimize propellant," "maximize final mass"
  - L^2 norm of control
  - Bang-bang structure

$$J_M = \int_{t_0}^{t_f} \|u(t)\| dt \qquad \text{or} \qquad J_M = -m_f$$

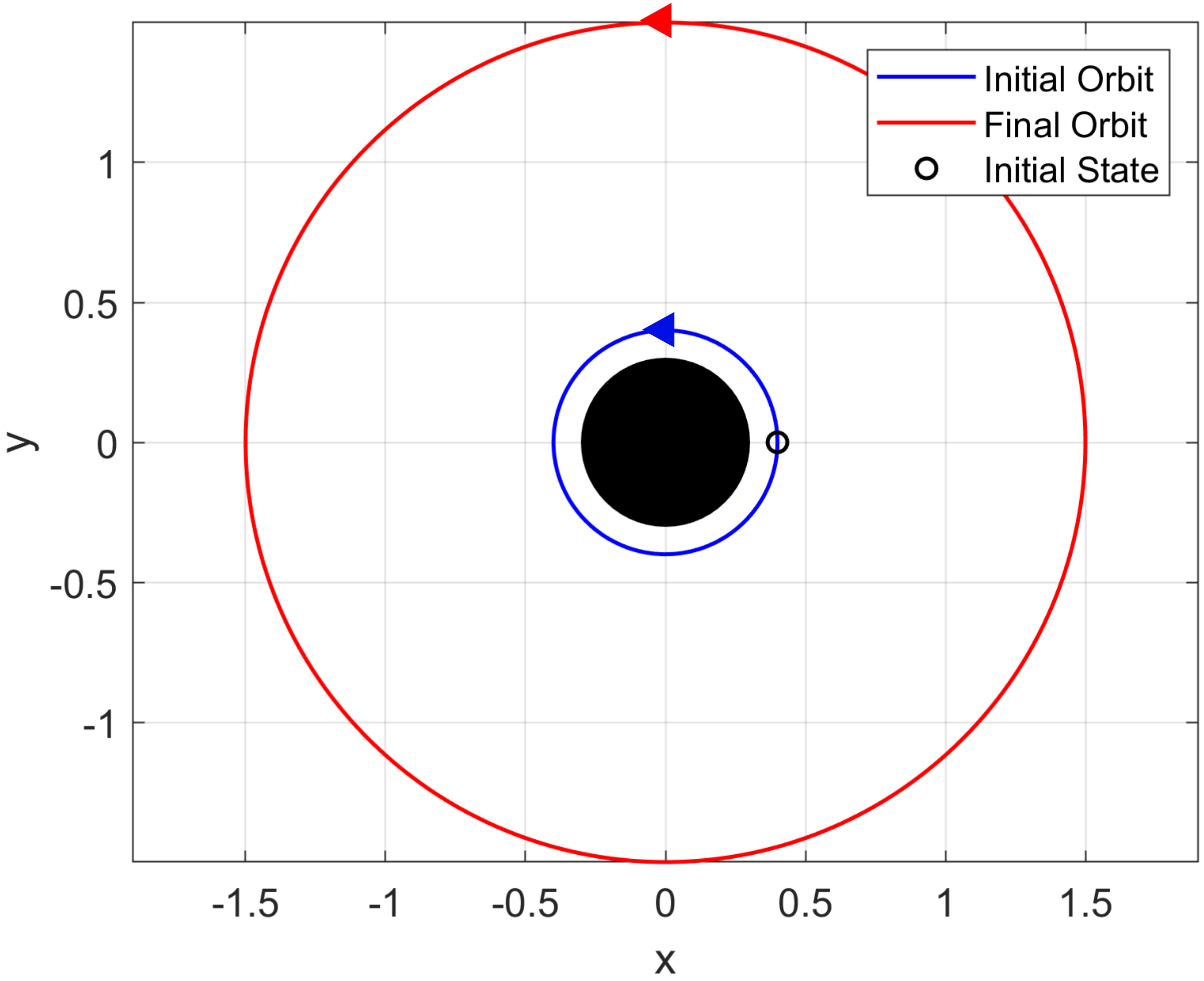
$$J = (1-\eta)J_E + \eta J_M$$

(Caillau 2012)



Example: Vary Constraints

- Fixed-time transfer between coplanar, circular asteroid orbits
- Minimize energy: $J_E = \frac{1}{2}\int_{t_0}^{t_f} \|u(t)\|^2 dt$
- Constrain $\|u(t)\|$
- Final true anomaly is unconstrained
- Indirect methods & single shooting
- Use homotopy to vary final time

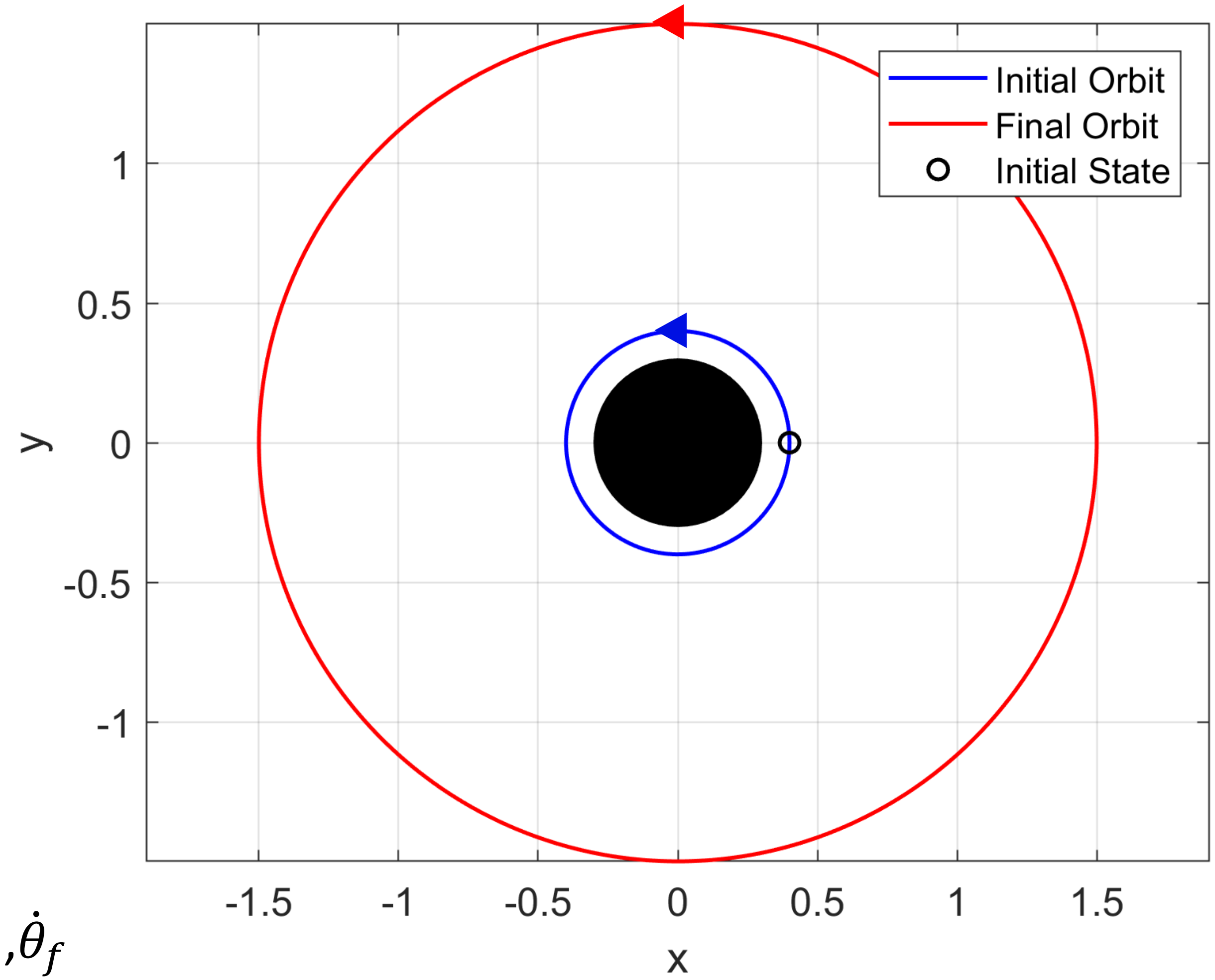


Ex. Detailed Problem Setup

Celestial and Spaceflight Mechanics Laboratory

- Minimize energy: $J_E = \frac{1}{2}\int_{t_0}^{t_f} \|u(t)\|^2 dt$
- Transfer between coplanar, circular asteroid orbits:
  - Planar two-body dynamics
  - Cartesian coordinates
  - $\mu = 5$ m³/s²
  - $r_0 = 400$ m
  - $r_f = 1500$ m
  - RK4, $\Delta t = 0.01$ (nondim.)
  - Nondimensionalize by

    $$l_{nd} = 1000 \text{ m}, \; t_{nd} = \sqrt{l_{nd}^3/\mu} \text{ sec}$$
- Constrain:
  - Initial and final time (vary t_f during homotopy)
  - Initial state
  - Final orbit geometry (true anomaly is free): r_f, $\dot{r}_f$, $\dot{\theta}_f$
  - $\|u(t)\| \leq$ 5e-6 N



Ex. Necessary Conditions

Celestial and Spaceflight Mechanics Laboratory

– Hamiltonian: $H = \boldsymbol{p}_r^T \boldsymbol{v} - \boldsymbol{p}_v^T \frac{\mu}{r^3} \boldsymbol{r} + \boldsymbol{p}_v^T \boldsymbol{u} + \frac{1}{2} \boldsymbol{u}^T \boldsymbol{u}$

– Optimal control: $\frac{\partial H}{\partial \boldsymbol{u}} = \boldsymbol{p}_v^T + \boldsymbol{u}^T = 0 \longrightarrow \boldsymbol{u} = -\boldsymbol{p}_v$

or

$\boldsymbol{u} = -u_{max} \boldsymbol{p}_v / \|\boldsymbol{p}_v\|$ when $\|\boldsymbol{p}_v\| \geq u_{max}$

– Adjoint dynamics: $\dot{\boldsymbol{p}} = -\frac{\partial H^*}{\partial \boldsymbol{X}}$

$$\dot{\boldsymbol{p}}_r = -\left(\frac{\partial \left(-\frac{\mu}{r^3} \boldsymbol{r} \right)}{\partial \boldsymbol{r}} \right)^T \boldsymbol{p}_v \qquad \dot{\boldsymbol{p}}_v = -\boldsymbol{p}_r$$

Ex. Transversality Conditions

– Constraints;

we want $\boldsymbol{g} = \boldsymbol{0}$:

$$\boldsymbol{g} = \Big[t_0, \quad t_f - t'_f, \quad \boldsymbol{X}_0^T - \boldsymbol{X'}_0^T, \quad \underbrace{r_f - r'_f, \quad \dot{r}_f, \quad \dot{\theta}_f - \dot{\theta}'_f}_{\text{only 3 constraints...}}\Big]^T$$

– Useful transversality conditions:

$$\boldsymbol{p}_f = \left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{X}_f}\right)^T \boldsymbol{\lambda} \longrightarrow \begin{bmatrix} p_x(t_f) \\ p_y(t_f) \\ p_{\dot{x}}(t_f) \\ p_{\dot{y}}(t_f) \end{bmatrix} = \begin{bmatrix} \frac{\partial r_f}{\partial x} & \frac{\partial \dot{r}_f}{\partial x} & \frac{\partial \dot{\theta}_f}{\partial x} \\ \frac{\partial r_f}{\partial y} & \frac{\partial \dot{r}_f}{\partial y} & \frac{\partial \dot{\theta}_f}{\partial y} \\ \frac{\partial r_f}{\partial \dot{x}} & \frac{\partial \dot{r}_f}{\partial \dot{x}} & \frac{\partial \dot{\theta}_f}{\partial \dot{x}} \\ \frac{\partial r_f}{\partial \dot{y}} & \frac{\partial \dot{r}_f}{\partial \dot{y}} & \frac{\partial \dot{\theta}_f}{\partial \dot{y}} \end{bmatrix} \begin{bmatrix} \lambda_7 \\ \lambda_8 \\ \lambda_9 \end{bmatrix}$$

– We only need 3 equations to solve for $\lambda_7, \lambda_8, \lambda_9$ in terms of $\boldsymbol{p}_f$

– Fourth equation provides an additional constraint: $\alpha(\boldsymbol{X}_f, \boldsymbol{p}_f)$

Ex. Single Shooting

$$\boldsymbol{G} = \begin{bmatrix} r_f - r'_f \\ \dot{r}_f \\ \dot{\theta}_f - \dot{\theta}'_f \\ \alpha(\boldsymbol{X}_f, \boldsymbol{p}_f) \end{bmatrix}$$

- Single shooting solution process:
  - $t_0, t_f, \boldsymbol{X}_0$ are fixed
  - Guess $\boldsymbol{p}_0$
  - Integrate state and adjoints from t_0 to t_f
  - Evaluate $\boldsymbol{G}$ at t_f
  - Update $\boldsymbol{p}_0$ until $||\boldsymbol{G}|| <$ tolerance
- How to update $\boldsymbol{p}_0$?
  - Differential corrector
  - Nonlinear equation solvers (e.g., fsolve.mat, nlsolve.jl)
  - Boundary value problem solvers
  - Gradient-based solvers in "feasibility mode" (e.g., SNOPT or fmincon.mat)

Ex. Differential Corrector

– Linear update:

$$\delta \boldsymbol{G} = \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{p}_0} \delta \boldsymbol{p}_0 \qquad \delta \boldsymbol{G} = \boldsymbol{0} - \boldsymbol{G}$$

$$\delta \boldsymbol{p}_0 = \left(\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{p}_0}\right)^{-1} \delta \boldsymbol{G} \qquad \delta \boldsymbol{p}_0 = \boldsymbol{p}_0^+ - \boldsymbol{p}_0^-$$

Scaling $\zeta < 1$ promotes numerical stability

$$\boldsymbol{p}_0^+ = \boldsymbol{p}_0^- - \zeta \left(\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{p}_0}\right)^{-1} \boldsymbol{G}$$

$$\boldsymbol{G} = \begin{bmatrix} r_f - r'_f \\ \dot{r}_f \\ \dot{\theta}_f - \dot{\theta}'_f \\ \alpha(\boldsymbol{X}_f, \boldsymbol{p}_f) \end{bmatrix}$$

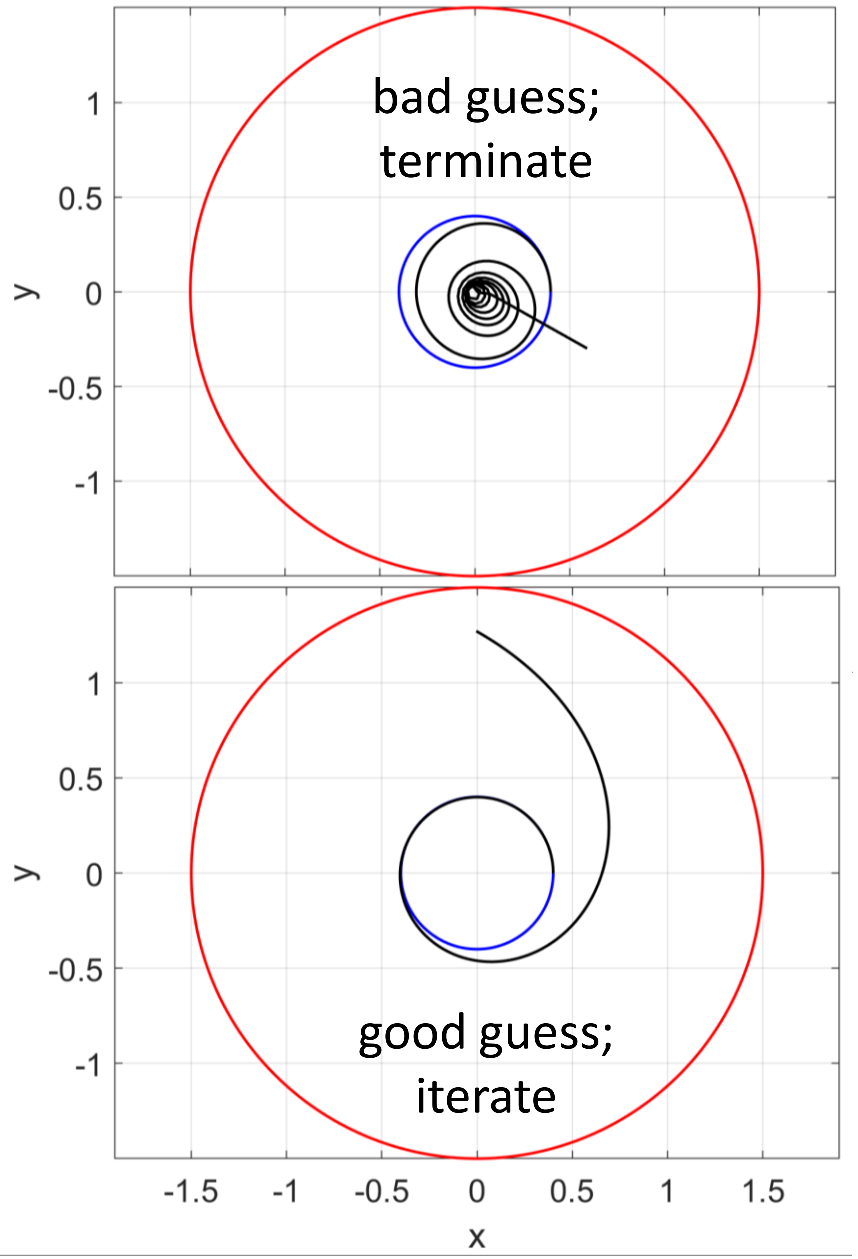
– Compute $\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{p}_0}$ analytically, by automatic differentiation, or by finite difference

– Forward finite difference: $\frac{\partial G_i}{\partial p_j} \approx \frac{G_i(p_j + \delta p_j) - G_i(p_j)}{\delta p_j}$ $\qquad \delta p_j = 1\text{e-}8$

– Fixed-step, fixed-order integrators are best when using finite differencing

Ex. Differential Corrector Tuning

- Knobs to turn (values I used in red):
  - # of initial guesses (30)
  - max. iterations per guess (30)
  - tolerance for convergence (1e-10)
  - max. error to continue iterating (15)
  - min. radius to continue iterating
  - step scale (ζ) [1e-1,…,1]
  - initial guess standard deviation (1e-2)
  - finite difference step size (if applicable) (1e-8)
  - # of repeated solutions
  - Problem scaling
- Make these values adaptive

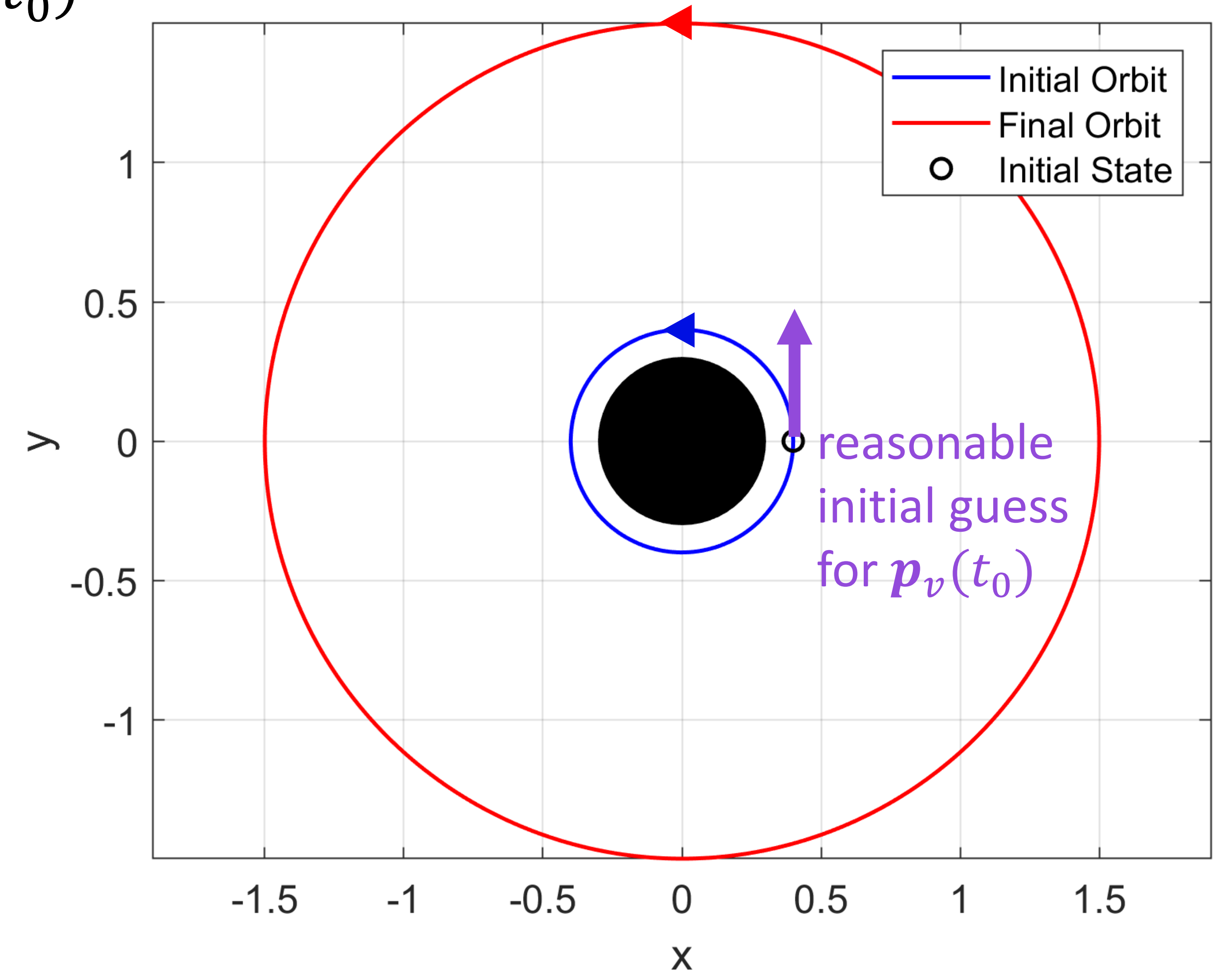


Adjoint-Control Transformation

- Primer vector: $\hat{\boldsymbol{u}} = -\boldsymbol{p}_v/||\boldsymbol{p}_v||$
- Can guess $\boldsymbol{p}_v(t_0)$ intuitively by guessing $\boldsymbol{u}(t_0)$
- Guess $\boldsymbol{p}_r(t_0)$ by guessing $\dot{\boldsymbol{u}}(t_0)$:

$$\boldsymbol{p}_r = -\dot{\boldsymbol{p}}_v$$

- See adjoint-control transformation (Ranieri 2005) for more info



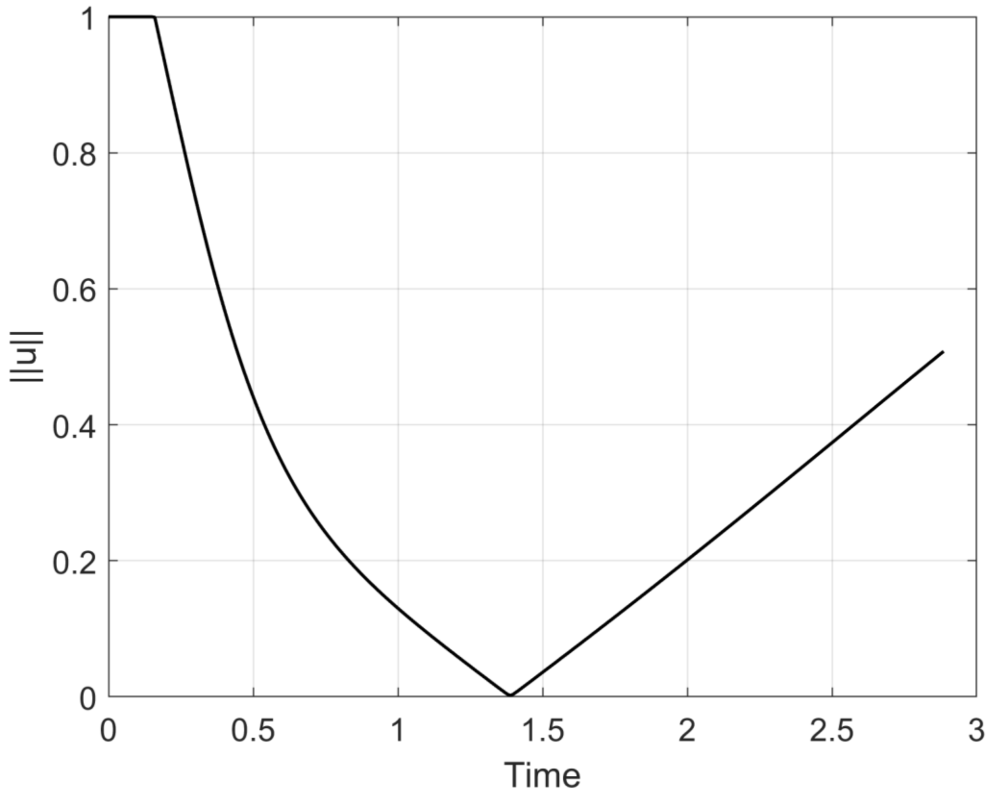
Ex. A solution!

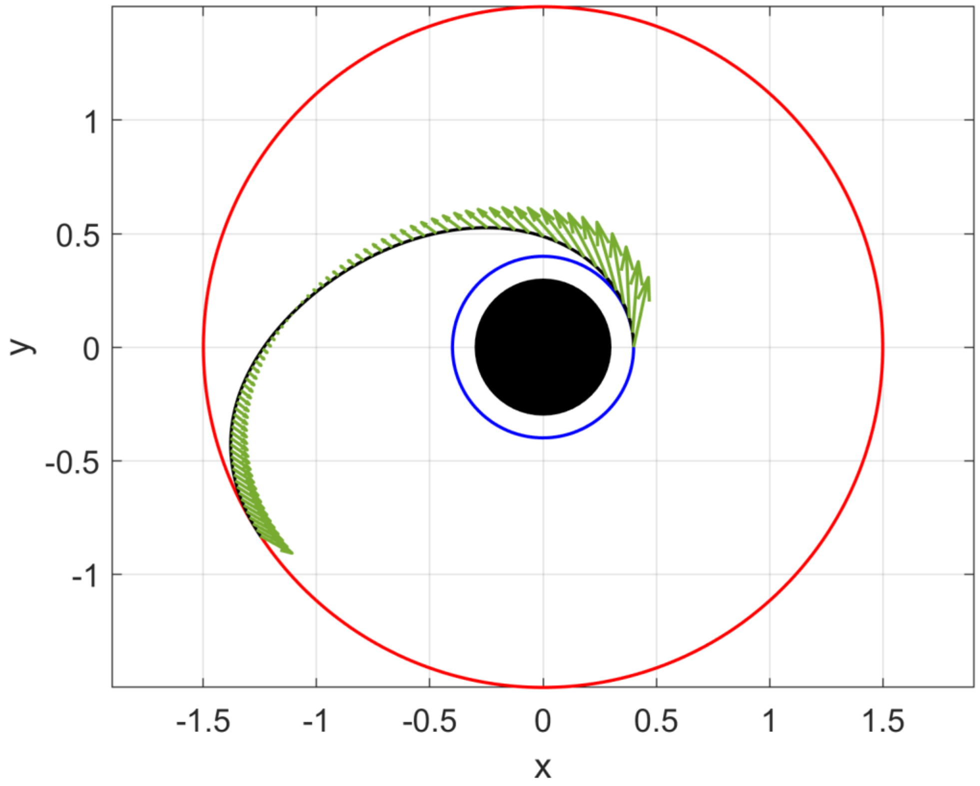
Converges < 10 iterations

$t_f = \frac{1}{2}$ red orbit period
$= 2.885737117864195$

Adjoint initial guess:
$\boldsymbol{p}_0$ =[-0.014264361594797
-0.010144507677053
-0.002132671883074
-0.003253477803605]

Adjoint solution:
$\boldsymbol{p}_0$ = [-4.369086817002391
-0.575469975692559
-0.265681228707072
-1.248322281877049]

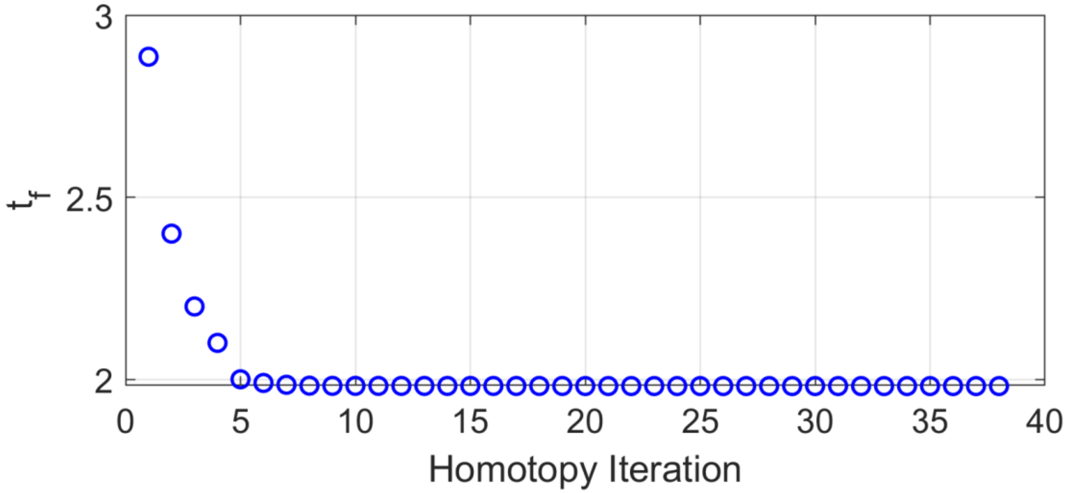


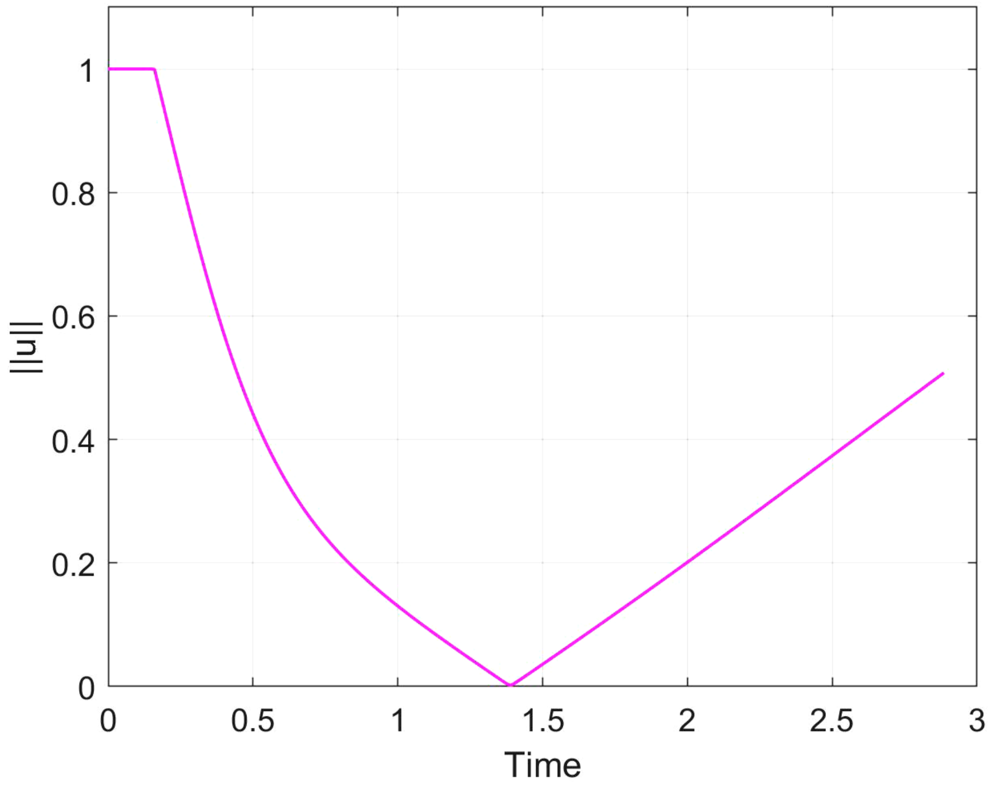


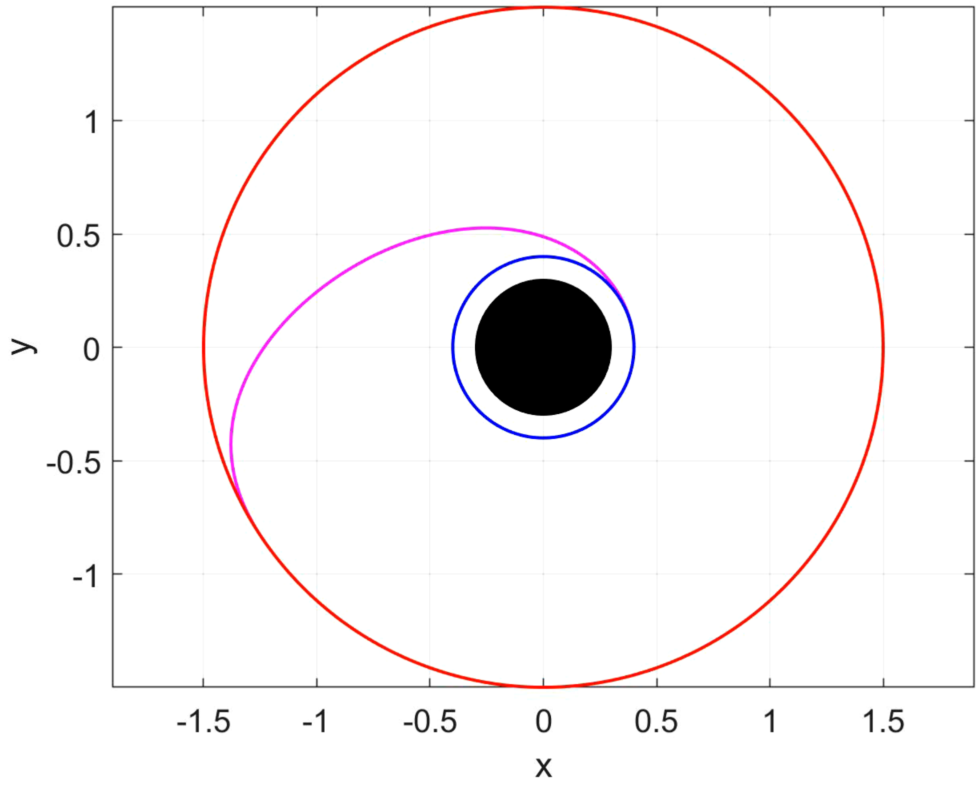
Ex. Homotopy

– Take progressively smaller steps δt_f

– Approaching maximum thrust as t_f is reduced

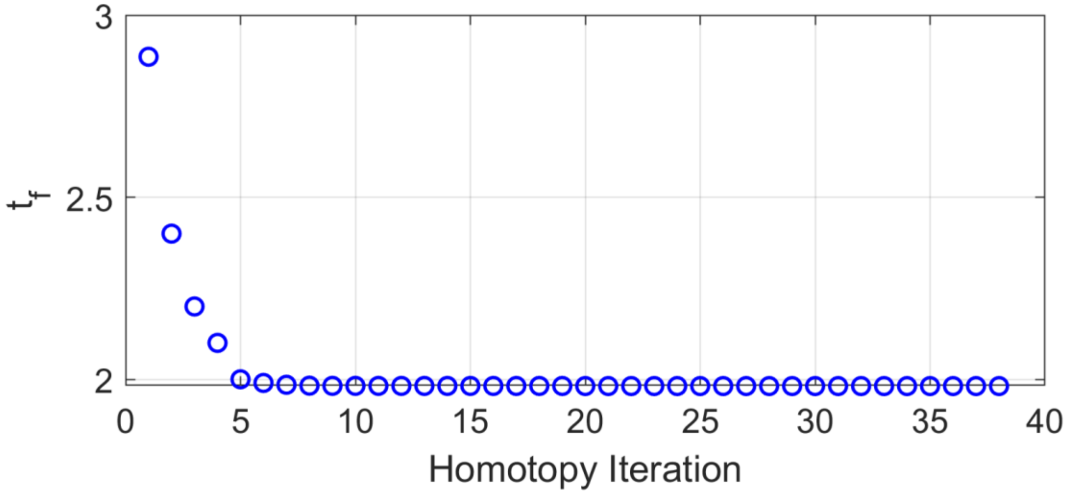


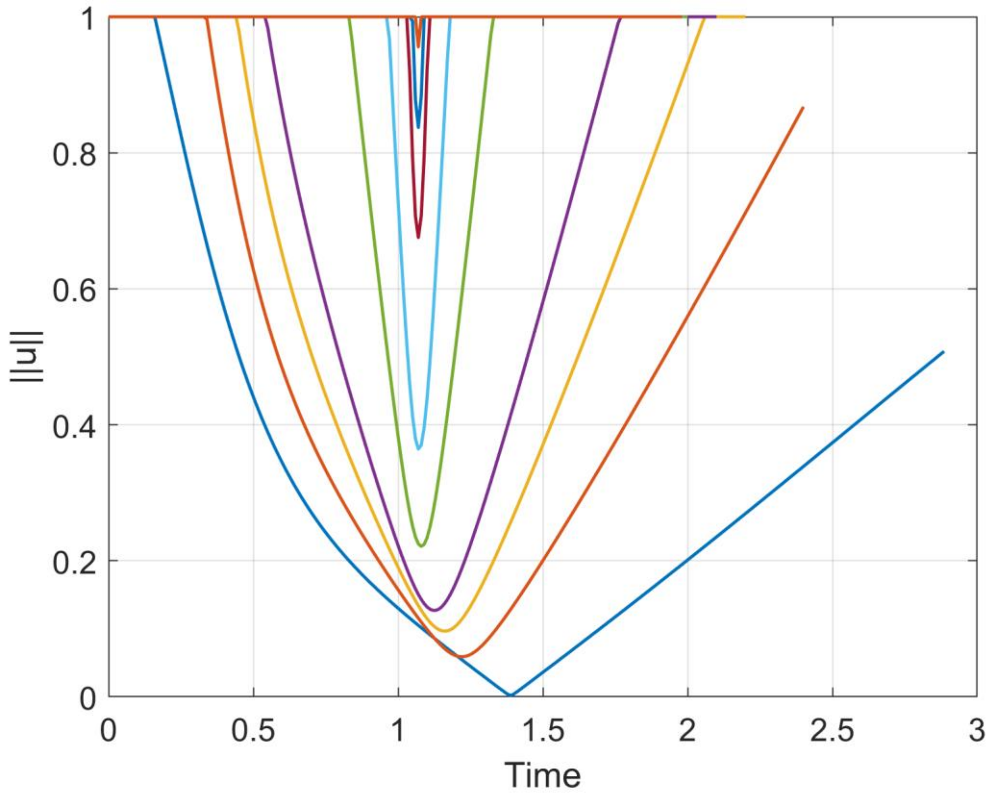


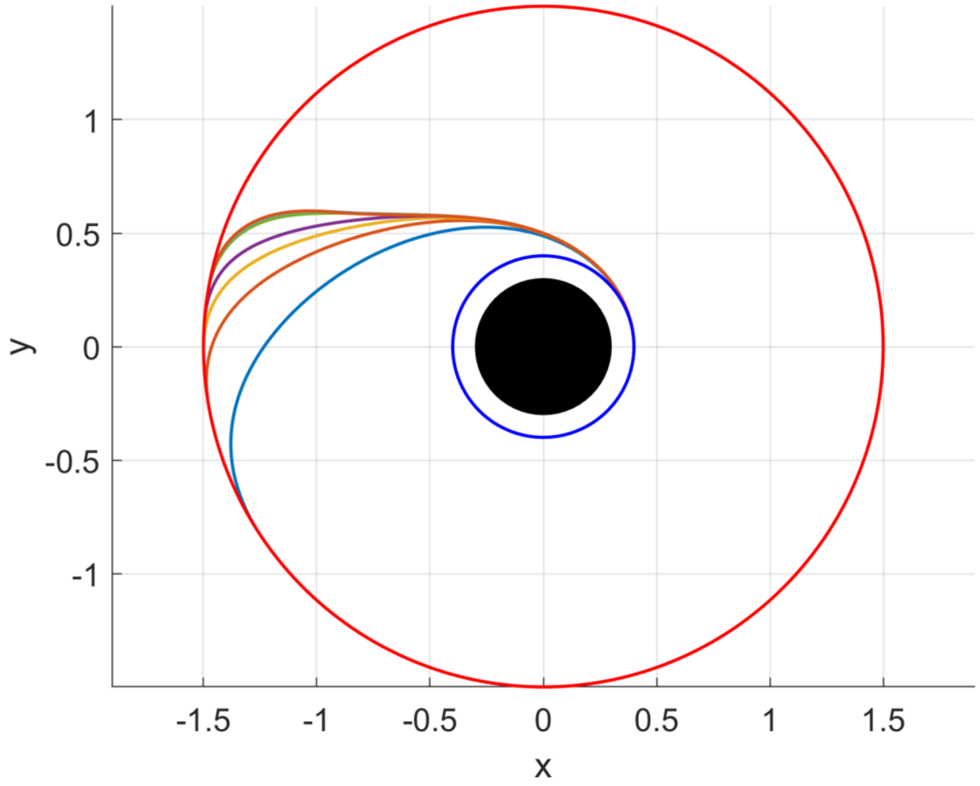


Ex. Homotopy

- Take progressively smaller steps δt_f
- Approaching maximum thrust as t_f is reduced

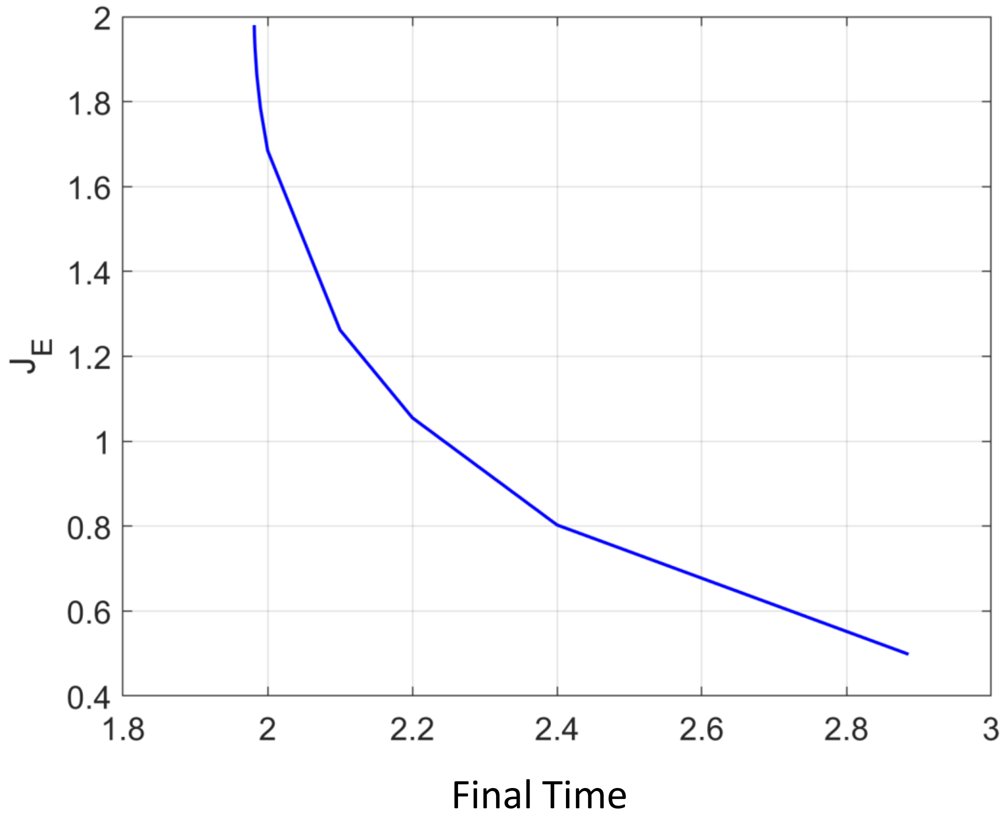


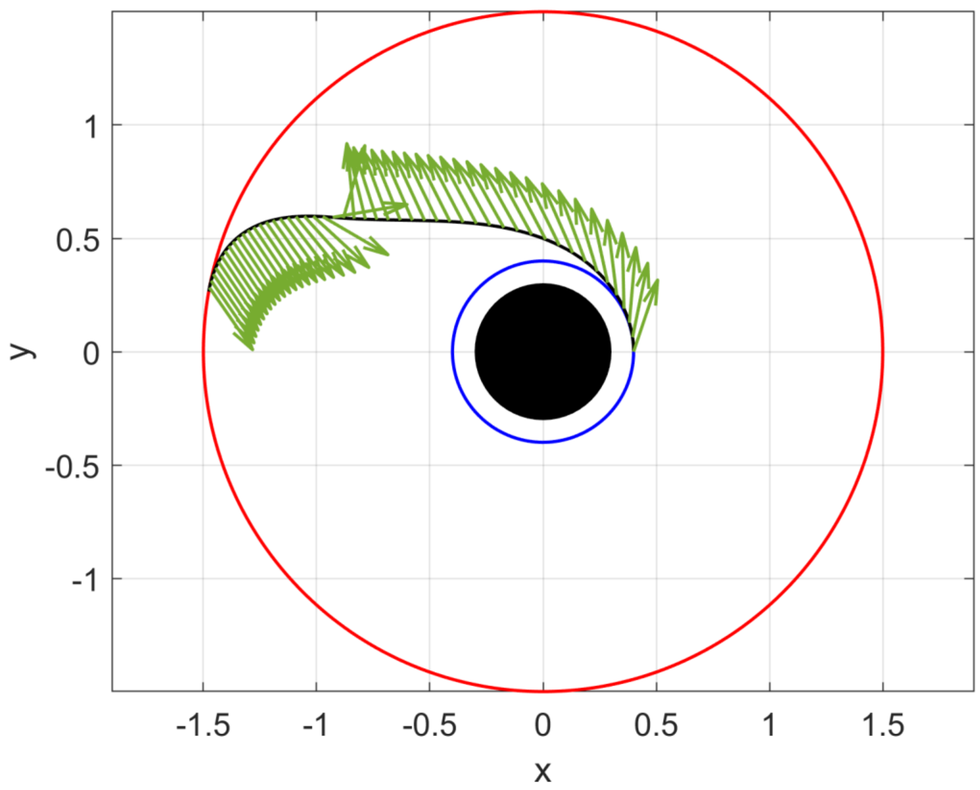




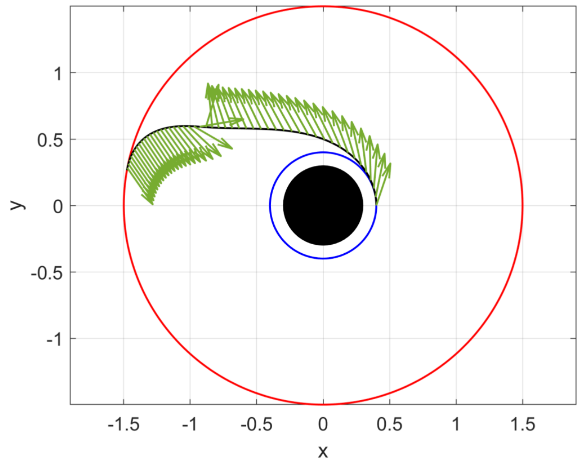
Ex. Result

Looks like a trade-off between energy and time of flight, but this is not a true multi-objective optimization

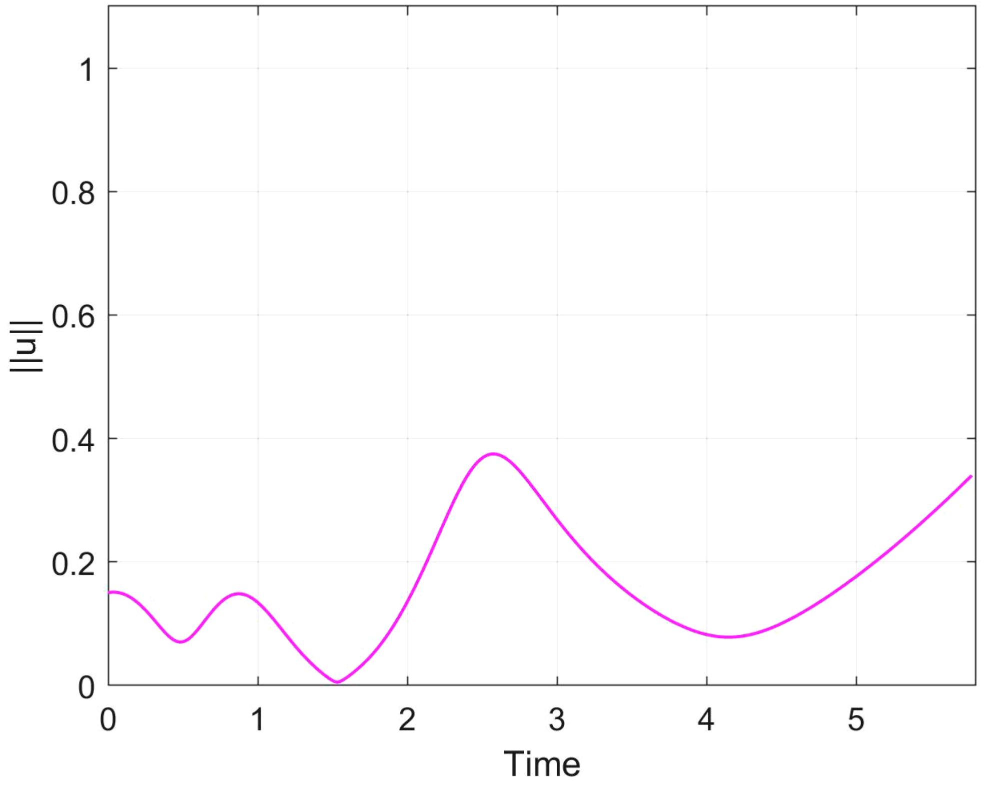


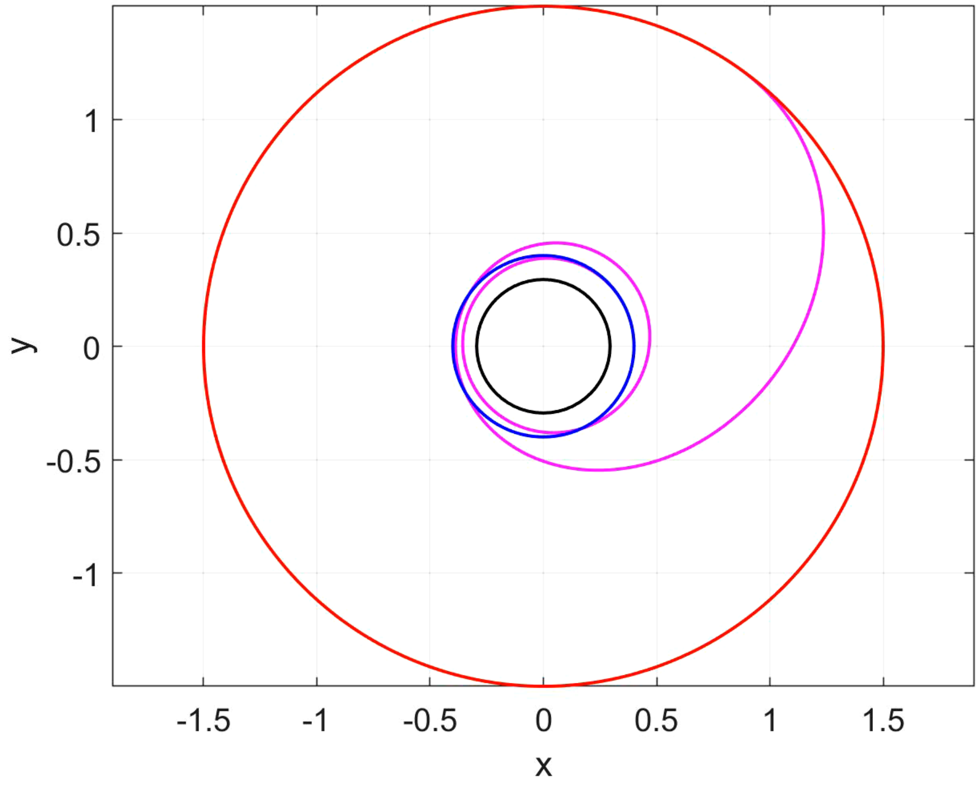


Homotopy Disadvantages



- Homotopy may not exist
- Not appropriate for multi-objective optimization (in general)
- Restricted solution space
  - E.g., homotopy from a “bad” solution impacts the asteroid surface:





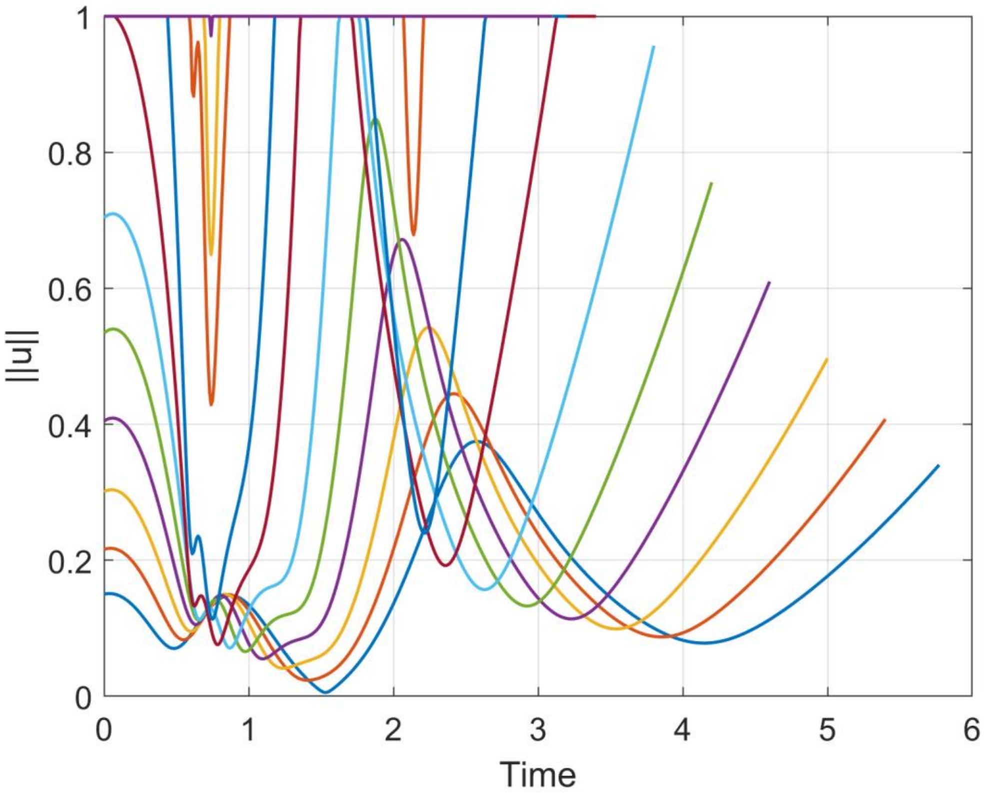
Homotopy Disadvantages

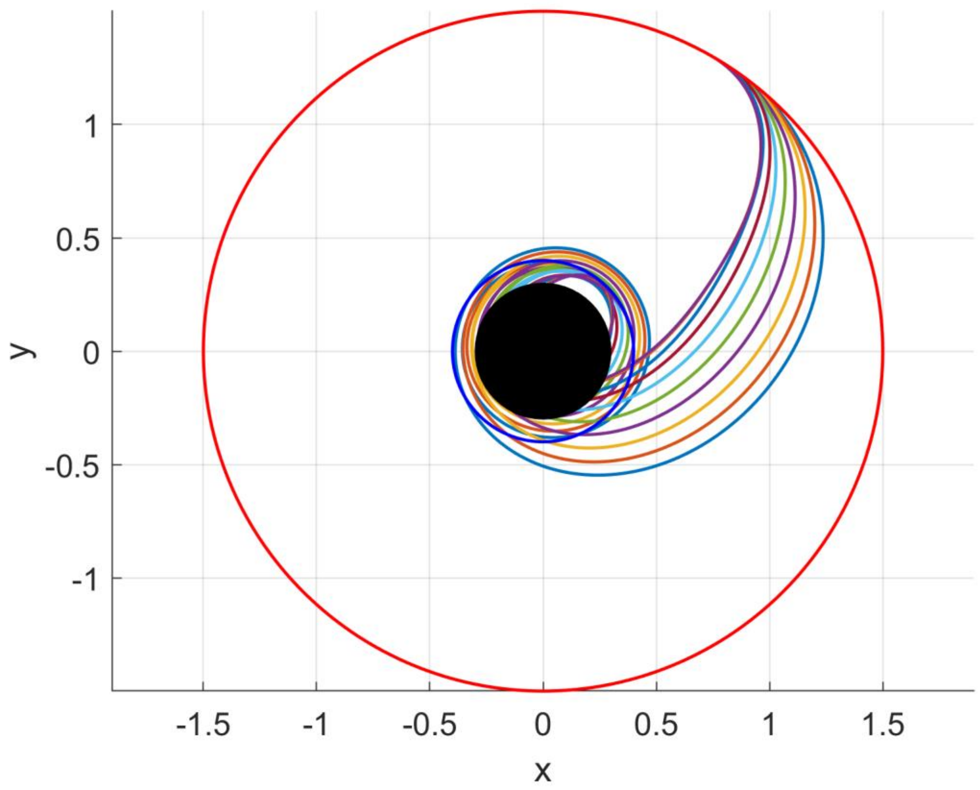
– Homotopy may not exist

– Not appropriate for multi-objective optimization (in general)

– Restricted solution space

  – E.g., homotopy from a “bad” solution impacts the asteroid surface:

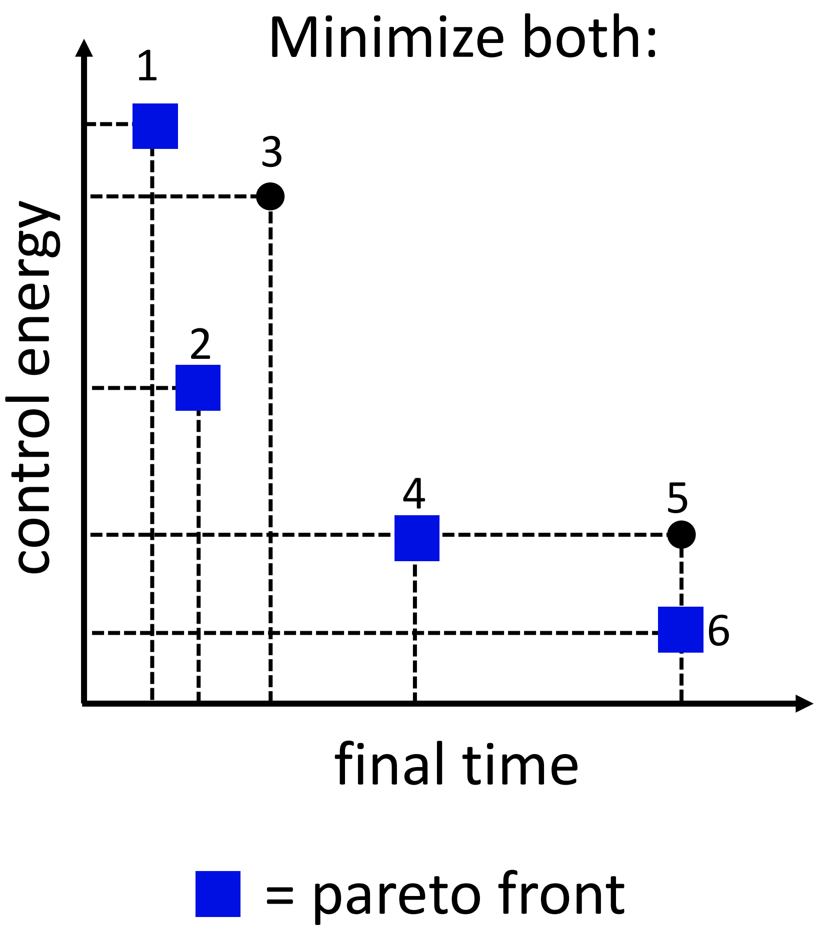




Multi-Objective Optimization

Multi-Objective Optimization

- Multi-objective solution is a trade-off between objectives
- Solution a **dominates** solution b if
  - a is better than b in at least one objective
  - a is no worse than b in other objectives
- Example:
  - 2 dominates 3
  - 4 and 6 dominate 5
- **Pareto optimal** solutions are non-dominated; cannot be improved in one objective without decreasing performance in another objective



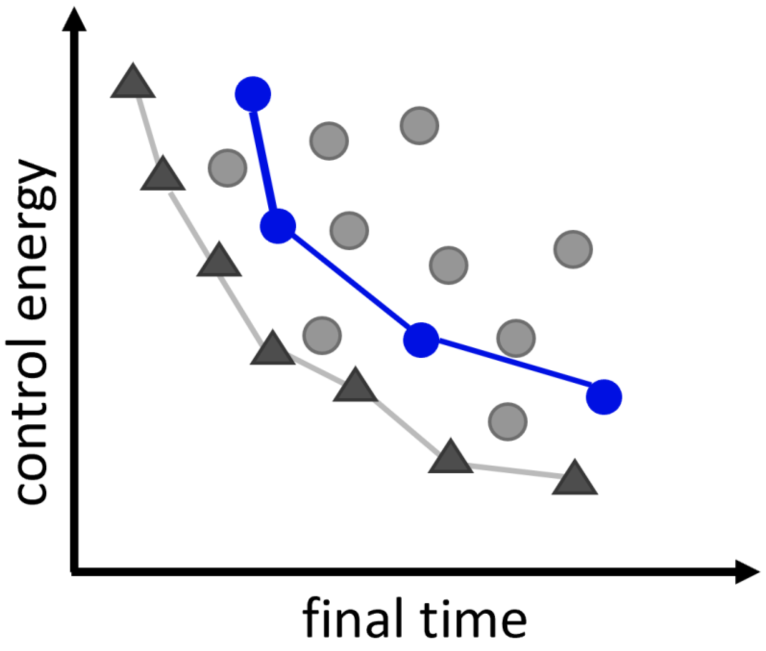
Scalarized Methods

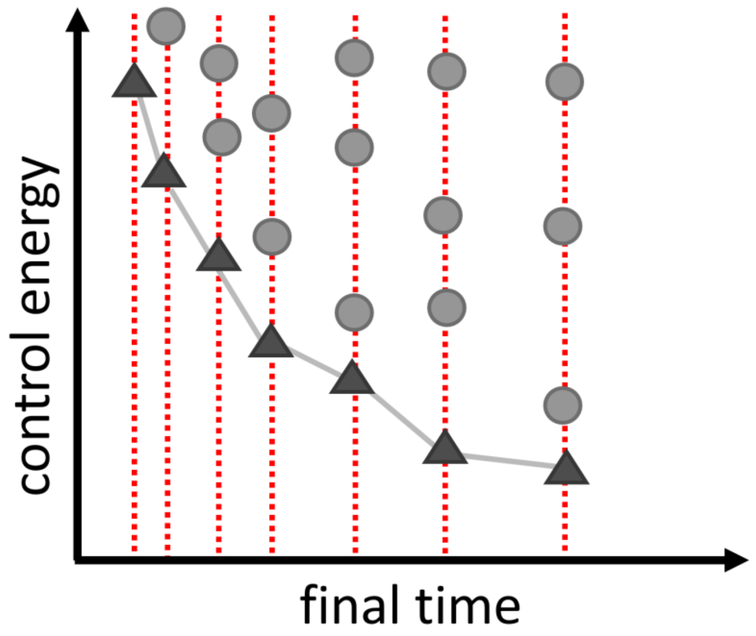
- Weighted-sum
  - Not guaranteed to be pareto-optimal

$$J = (1 - \eta)J_1 + \eta J_2$$

- Epsilon-constraint (Haimes 1987, Mavrotas 2009)
  - Optimize one cost function, constrain others
  - Can be equality or inequality constraints
  - Globally-optimal solutions lie on Pareto front

- If both methods have the same necessary/transversality conditions, weighted-sum can be Pareto optimal (Jenson 2021)





Evolutionary Algorithms

- State of the art in multi-objective optimization
- Metaheuristic, **population-based** algorithms
- E.g., nondominated sorting genetic algorithm (NDSGA-II) (Deb 2002)
  - Genetic algorithms are inspired by natural selection
  - A population of candidate solutions are mated, mutated, etc. from one generation to the next
  - “Elitism” improves convergence: best solutions are carried into next generation
  - NDSGA-III (Deb 2014)

References

- J.-B. Caillau, B. Daoud, J. Gergaud, “Minimum fuel control of the planar circular restricted three-body problem,” Celest Mech Dyn Astr, 2012, DOI: 10.1007/s10569-012-9443-x
- C. L. Ranieri, C. A. Ocampo, “Optimization of Roundtrip, Time-Constrained, Finite Burn Trajectories via Indirect Method,” Journal of Guidance, Control, and Dynamics, 2005, DOI: 10.2514/1.5540
- Y. Y. Haimes, D. Li, “Hierarchical Multiobjective Analysis for Large-Scale Systems: Current Status,” IFAC Proceedings Volumes, 1987, DOI: 10.1016/S1474-6670(17)55679-7
- G. Mavrotas, “Effective implementation of the ε-constraint method in Multi-Objective Mathematical Programming problems,” Applied Mathematics and Computation, 2009, DOI: 10.1016/j.amc.2009.03.037
- E. L. Jenson, D. J. Scheeres, “Multi-Objective Optimization of Covariance and Energy for Asteroid Transfers,” ,” Journal of Guidance, Control, and Dynamics, 2021, DOI: 10.2514/1.G005609
- K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” IEEE Transactions on Evolutionary computation, 2002, DOI: 10.1109/4235.996017

Questions?

Celestial and Spaceflight Mechanics Laboratory

– Erica.Jenson@colorado.edu