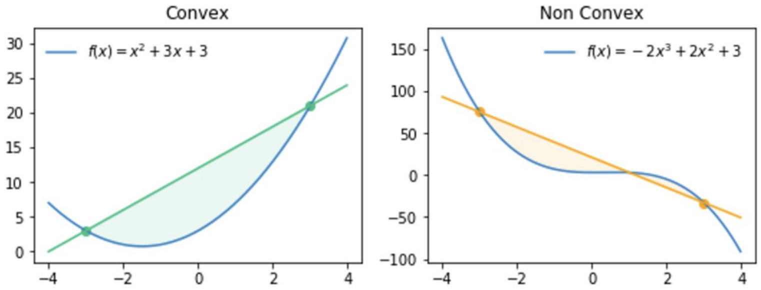
Convex Optimization and Fuel Optimal Rendezvous

Govind Chari and Samuel Noles

What is Convex Optimization?

Optimizing a convex function over a convex set



Convex functions:

- Secant line lies above the function
- Key: **A local minima is a global minima**

[1]

[1] L. Blackmore, Autonomous precision landing of space rockets, The Bridge on Frontiers of Engineering, 4 (2016), pp. 15–20.

Quadratic Programs

- Program → Optimization Problem
- Convex quadratic function subject to equality and inequality constraints
- Most common class of convex optimization problems for trajectory generation

$$
\begin{aligned}
\min_{x} \quad & \frac{1}{2}x^T Q x + q^T x \\
\text{s.t.} \quad & Ax = b \\
& Gx \preceq h
\end{aligned}
$$

KKT Conditions
(with inequality constraints)

- Unconstrained optimality condition: $\nabla f = 0$
  - Does not work for the constrained problem

Steps

1. Inequality to equality constraint
2. Form Lagrangian
3. Take gradient wrt x
4. Form rest of KKT conditions

$$\begin{array}{ll} \min_x & \frac{1}{2}x^TQx + q^Tx \\ \text{s.t.} & Ax = b \\ & Gx \preceq h \end{array}$$

$$\begin{array}{ll} \min_x & \frac{1}{2}x^TQx + q^Tx \\ \text{s.t.} & Ax = b \\ & Gx + s = h \\ & s \geq 0 \end{array}$$

$$\mathcal{L}(x, s, \lambda, \mu) = \frac{1}{2}x^TQx + q^Tx + y^T(Ax - b) + z^T(Gx + s - h)$$

$$\begin{gathered} \nabla_x \mathcal{L} = 0 \\ Ax - b = 0 \\ Gx + s - h = 0 \\ s \geq 0 \\ z \geq 0 \\ z_i s_i = 0 \end{gathered}$$

Primal Dual Interior Point Method [2]

- Assemble the KKT conditions into a function
  - The roots of the function are the solution to the optimization problem
- Solve with Newton iteration

$$\nabla_x \mathcal{L} = 0$$
$$Ax - b = 0$$
$$Gx + s - h = 0$$
$$s \geq 0$$
$$z \geq 0$$
$$z_i s_i = 0$$

$$F(x, s, y, z) = \begin{bmatrix} Qx + q + A^T + G^T x \\ ZS\mathbf{1}^T \\ Gx + s - h \\ Ax - b \end{bmatrix} = 0$$

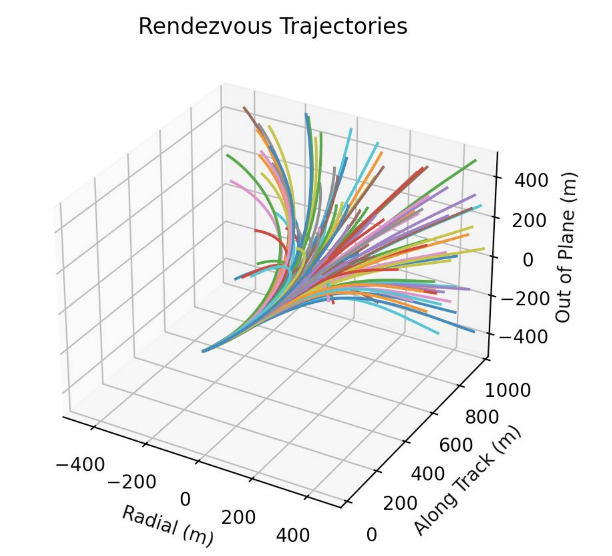
Newton Step:

$$\begin{bmatrix} Q & 0 & G^T & A^T \\ 0 & Z & S & 0 \\ G & I & 0 & 0 \\ A & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta s \\ \Delta z \\ \Delta y \end{bmatrix} = \begin{bmatrix} -(Qx + q + A^T + G^T x) \\ -(ZS\mathbf{1}^T) \\ -(Gx + s - h) \\ -(Ax - b) \end{bmatrix}$$

[2] Vandenberghe L (2010) The cvxopt linear and quadratic cone program solvers. http://abel.ee.ucla.edu/cvxopt/documentation/coneprog.pdf, March 2010

Application to Rendezvous

- Close approach trajectory of spacecraft to ISS
- Constraints
    - Trajectory constrained to an approach cone
    - Upper thrust bound



Dynamics

- Clohessy Wiltshire
- Dynamics must be discretized

State vector

$$\dot{\boldsymbol{x}} = A_c\boldsymbol{x} + B_c\boldsymbol{u}$$

Control Inputs

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 0 & 2n & 0 \\ 0 & 0 & 0 & -2n & 0 & 0 \\ 0 & 0 & -n^2 & 0 & 0 & -n^2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k$$

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 4-3cnt & 0 & 0 & (1/n)snt & (2/n)(1-cnt) & 0 \\ 6(snt-nt) & 1 & 0 & (2/n)(cnt-1) & (1/n)(4snt-3nt) & 0 \\ 0 & 0 & cnt & 0 & 0 & (1/n)snt \\ 3nsnt & 0 & 0 & cnt & 2snt & 0 \\ 6n(cnt-1) & 0 & 0 & -2snt & 4cnt-3 & 0 \\ 0 & 0 & -nsnt & 0 & 0 & cnt \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} + \begin{bmatrix} (1/n)snt & (2/n)(1-cnt) & 0 \\ (2/n)(cnt-1) & (1/n)(4snt-3nt) & 0 \\ 0 & 0 & (1/n)snt \\ cnt & 2snt & 0 \\ -2snt & 4cnt-3 & 0 \\ 0 & 0 & cnt \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}$$

where $nt = n\Delta t$, $snt = sin(n\Delta t)$, $cnt = cos(n\Delta t)$, and $u_i = f_i\Delta t$

Unconstrained Rendezvous Problem

- Time horizon → T
    - Number of steps to break the trajectory into
- Seek to minimize control effort
- Only constraints are dynamics and initial and final decision

$$\begin{aligned} \min_{\boldsymbol{x}_0,\ldots,\boldsymbol{x}_T,\boldsymbol{u}_0,\ldots,\boldsymbol{u}_{T-1}} \quad & \textstyle\sum_{k=0}^{T-1} \|\boldsymbol{u}_k\|_2^2 \\ \text{s.t. } \boldsymbol{x}_{k+1} &= A\boldsymbol{x}_k + B\boldsymbol{u}_k \\ \boldsymbol{x}(0) &= \boldsymbol{x}_0 \\ \boldsymbol{x}(\mathrm{T}) &= 0 \end{aligned}$$

Canonicalization

Problem 1

$$\min_{\boldsymbol{x}_0,\ldots,\boldsymbol{x}_T,\boldsymbol{u}_0,\ldots,\boldsymbol{u}_{T-1}} \quad \sum_{k=0}^{T-1} \|\boldsymbol{u}_k\|_2^2$$
$$\text{s.t.} \quad \boldsymbol{x}_{k+1} = A\boldsymbol{x}_k + B\boldsymbol{u}_k$$
$$\boldsymbol{x}(0) = \boldsymbol{x}_0$$
$$\boldsymbol{x}(\mathrm{T}) = 0$$

→

Problem 2

$$\min_x \quad \tfrac{1}{2}x^TQx + q^Tx$$
$$\text{s.t.} \quad Ax = b$$
$$Gx \preceq h$$

- We know how to solve Problem 2 but we have Problem 1
- The following definitions allow us to cast Problem 1 to Problem 2
- A lot of packages do this for you
  - CVX (MATLAB), CVXPY (Python), Convex.jl (Julia), Epigraph (C++)

$$x = \begin{bmatrix} \boldsymbol{u}_0 & \boldsymbol{x}_1 & \boldsymbol{u}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{u}_{T-1} \end{bmatrix}^T$$

$$Q = \begin{bmatrix} I & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & I & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & I \end{bmatrix}$$

$$A = \begin{bmatrix} B & -I & 0 & 0 & \cdots & 0 \\ 0 & A & B & -I & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & A & B \end{bmatrix}$$

$$b = \begin{bmatrix} -A\boldsymbol{x}_0 & 0 & 0 & 0 & \cdots & \boldsymbol{x}_T \end{bmatrix}^T$$

Adding State and Input Constraints

Upper Thrust Bound → $\boldsymbol{u}_k^T \boldsymbol{u}_k \leq u_{max}^2$

Approach Cone → $\|S\boldsymbol{x}_k\| + \boldsymbol{c}^T \boldsymbol{x}_k \leq 0$

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad c = \begin{bmatrix} 0 & -tan(\gamma) & 0 & 0 & 0 & 0 \end{bmatrix}$$

Problem 3

$$\min_{\boldsymbol{x}_0,\ldots,\boldsymbol{x}_T,\boldsymbol{u}_0,\ldots,\boldsymbol{u}_{T-1}} \quad \sum_{k=0}^{T-1} \|\boldsymbol{u}_k\|_2^2$$

$$\text{s.t. } \boldsymbol{x}_{k+1} = A\boldsymbol{x}_k + B\boldsymbol{u}_k$$

$$\boldsymbol{u}_k^T \boldsymbol{u}_k \leq u_{max}^2$$

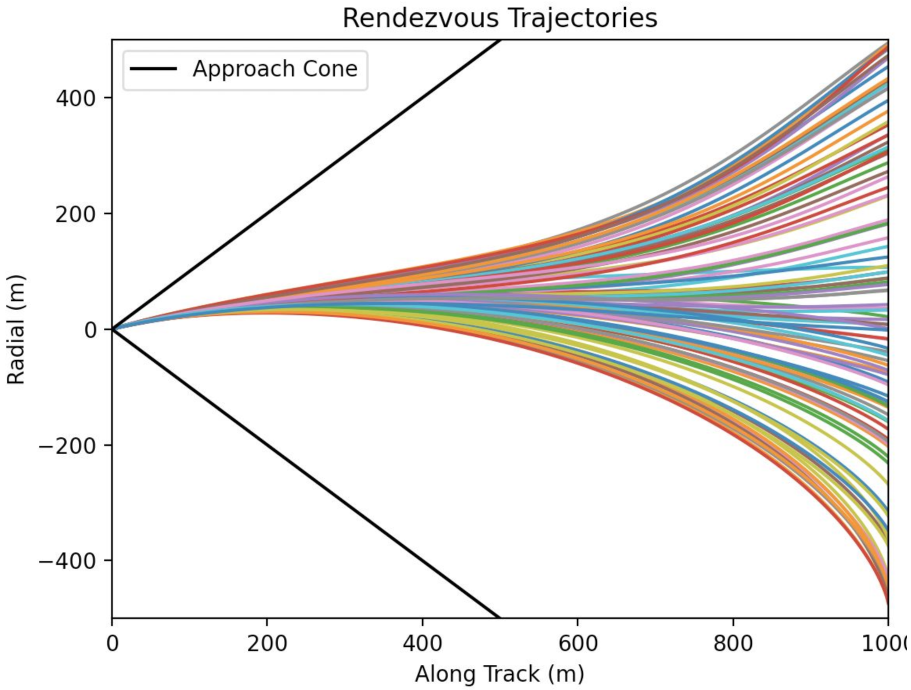
$$\|S\boldsymbol{x}_k\| + \boldsymbol{c}^T \boldsymbol{x}_k \leq 0$$

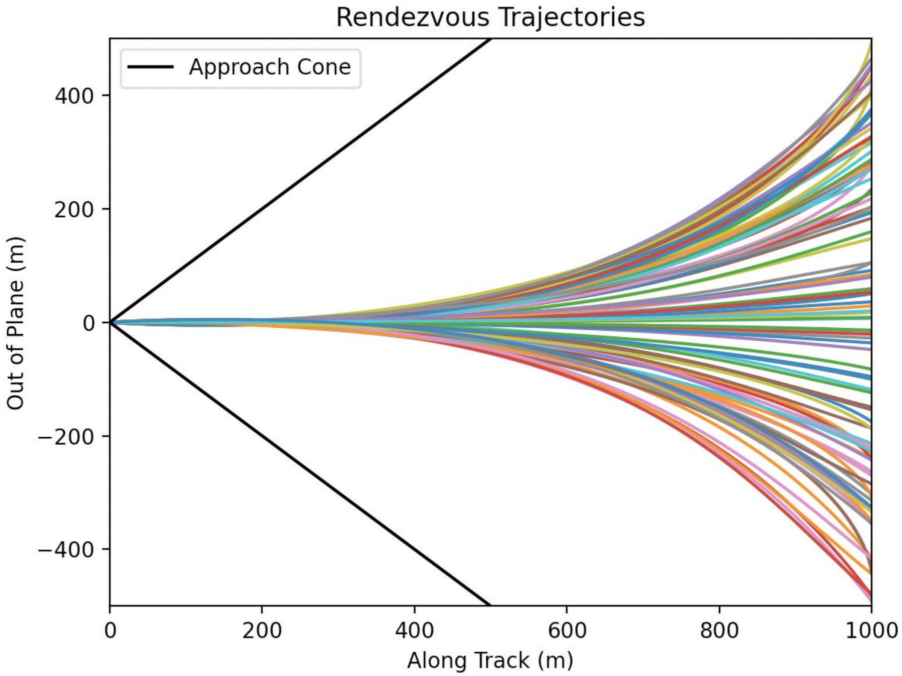
$$\boldsymbol{x}(0) = \boldsymbol{x}_0$$

$$\boldsymbol{x}(\mathrm{T}) = 0$$

This is not a QP but a Second Order Cone Program...

Monte-Carlo Trajectories





L1 vs L2 Penalties

L2 Norm Penalty

Problem 3

$$\min_{\boldsymbol{x}_0,\ldots,\boldsymbol{x}_T,\boldsymbol{u}_0,\ldots,\boldsymbol{u}_{T-1}} \quad \sum_{k=0}^{T-1} \|\boldsymbol{u}_k\|_2^2$$

$$\text{s.t. } \boldsymbol{x}_{k+1} = A\boldsymbol{x}_k + B\boldsymbol{u}_k$$

$$\boldsymbol{u}_k^T \boldsymbol{u}_k \leq u_{max}^2$$

$$\|S\boldsymbol{x}_k\| + \boldsymbol{c}^T \boldsymbol{x}_k \leq 0$$

$$\boldsymbol{x}(0) = \boldsymbol{x}_0$$

$$\boldsymbol{x}(\mathrm{T}) = 0$$

L1 Norm Penalty

Problem 4

$$\min_{\boldsymbol{x}_0,\ldots,\boldsymbol{x}_T,\boldsymbol{u}_0,\ldots,\boldsymbol{u}_{T-1}} \quad \sum_{k=0}^{T-1} \|\boldsymbol{u}_k\|_1$$

$$\text{s.t. } \boldsymbol{x}_{k+1} = A\boldsymbol{x}_k + B\boldsymbol{u}_k$$

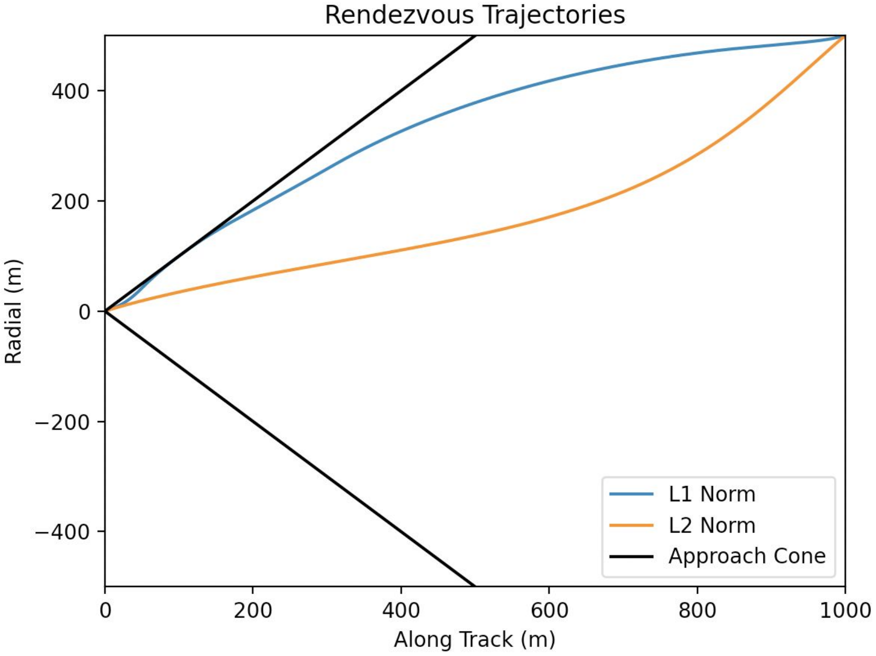
$$\boldsymbol{u}_k^T \boldsymbol{u}_k \leq u_{max}^2$$

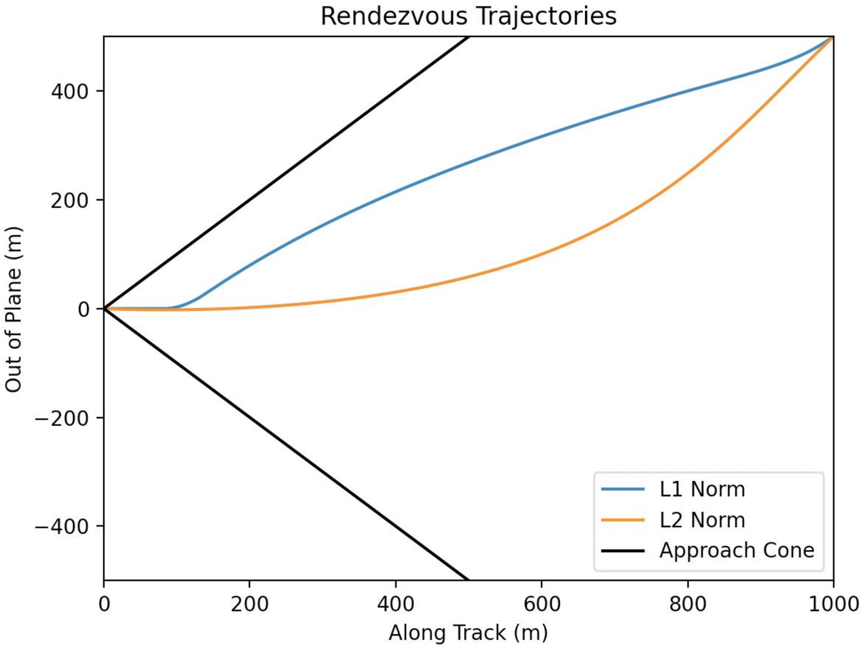
$$\|S\boldsymbol{x}_k\| + \boldsymbol{c}^T \boldsymbol{x}_k \leq 0$$

$$\boldsymbol{x}(0) = \boldsymbol{x}_0$$

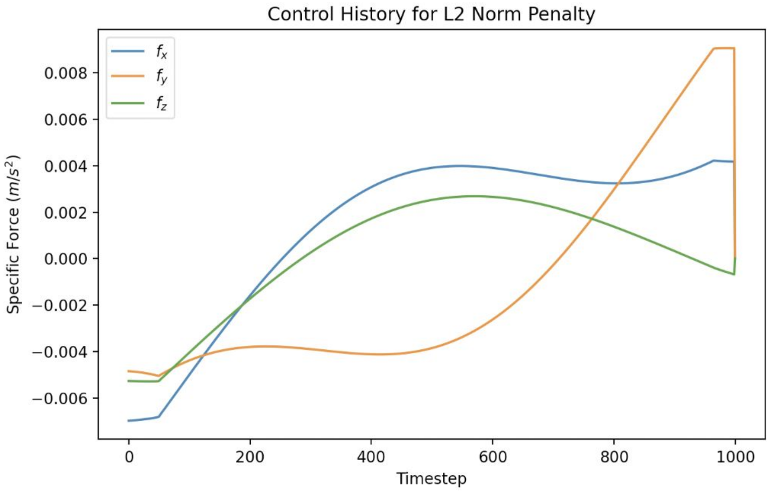
$$\boldsymbol{x}(\mathrm{T}) = 0$$

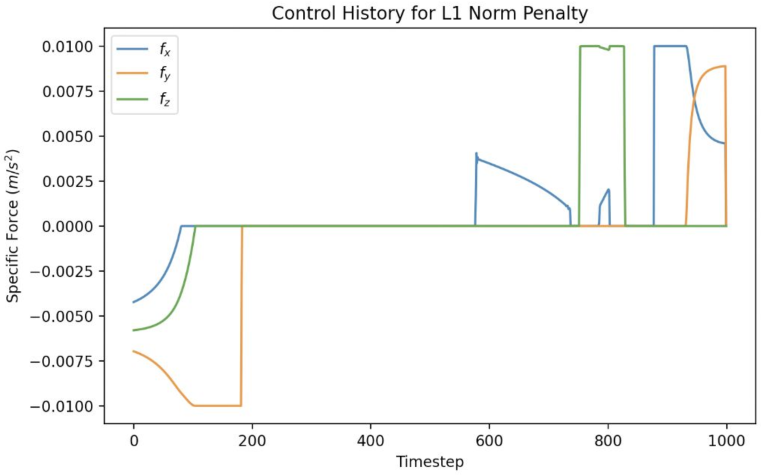
L1 vs L2 Norm Trajectories





L1 vs L2 Thrust History





Code to generate all the figures and an installation guide can be found here.