5: Convex Optimization

- optimization
- convexity
- linear programming example

Optimization

standard form optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & g_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_j(x) = 0, \quad j = 1, \ldots, p \end{array}$$

- $x \in \mathbb{R}^n$: **decision variables** – the things you choose
- $f_0 : \mathbb{R}^n \to \mathbb{R}$: **objective function** – the cost you pay for choosing x
- $g_i : \mathbb{R}^n \to \mathbb{R}$: **inequality constraint functions** – criteria your choice must satisfy
- $h_j : \mathbb{R}^n \to \mathbb{R}$: **equality constraint functions**

a **solution** or **optimal point** $x^\star$ returns the smallest value of f_0 from all choices of x that satisfy all the constraints

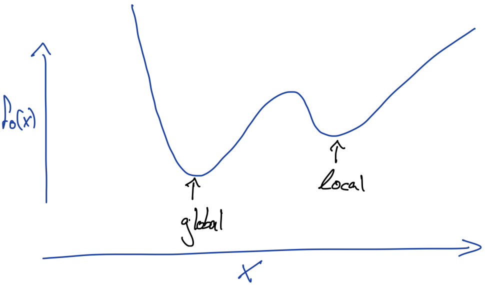
a **feasible point** is any x that satisfies all the constraints

Minima

we say that $\bar{x}$ is a **local minimum** if $\bar{x}$ is feasible and there exists an $\epsilon > 0$ such that

$$f_0(\bar{x}) \leq f_0(x) \quad \text{for all } x \in \mathcal{B}(\bar{x}, \epsilon)$$

when the inequality holds for all $x \in \mathbb{R}^n$, then $\bar{x}$ is **global minimum**



Solving optimization problems

- most problems are very difficult to solve
- large n is a computational not a theoretical problem
- in many cases, sub-optimal solutions are fine
  - provided we can quantify the sub-optimality level
- how the problem is modeled will determine how/if it can be solved

Which problems are solvable?

- convex optimization problems
  - least-squares
  - linear programs
  - conic programs
- a few special cases
  - problems involving exactly two quadratics, a few others if n is small

Affine sets

let V be a vector space, then $S \subseteq V$ is an **affine set** if

$$x, y \in S \quad \lambda, \gamma \in \mathbb{R} \quad \lambda + \gamma = 1 \quad \Rightarrow \quad \lambda x + \gamma y \in S$$

Geometrically: the line v through x, y, with $x, y \in S$

$$\{v \in V \mid v = \theta x + (1 - \theta) y, \quad \theta \in \mathbb{R}\}$$

is contained in S

Representations:

- $\mathrm{range}(Au + b)$, with $u \in U$ where $A : U \to V$
- the solution to a set of linear equations: let $B : V \to U$

$$S = \{x \mid Bx = c\}$$

Convex sets

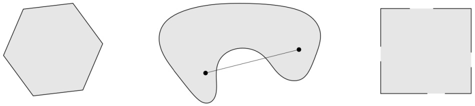
the set $S \subseteq V$ is **convex** if

$$x, y \in S \quad \lambda, \gamma \geq 0 \quad \lambda + \gamma = 1 \quad \Rightarrow \quad \lambda x + \gamma y \in S$$

Geometrically: the line segment between x, y, with $x, y \in S$

$$\mathcal{L}(x, y) = \{v \in V \mid v = \theta x + (1 - \theta) y, \quad \theta \in [0, 1]\}$$

is contained in S



[Figure from Boyd & Vandenberghe]

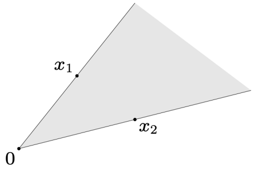
Convex cones

the set $S \subseteq V$ is a **cone** if

$$x \in S, \quad \theta \geq 0 \quad \Rightarrow \quad \theta x \in S$$

S is a **convex cone** if

$$x_1, x_2 \in S \quad \lambda, \gamma \geq 0 \quad \Rightarrow \quad \lambda x_1 + \gamma x_2 \in S$$



Geometrically: a set that contains all conic combinations of x_1 and x_2

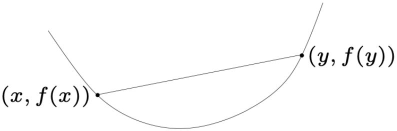
[Figure from Boyd & Vandenberghe]

Convex functions

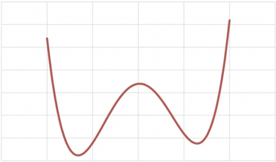
let $X \subseteq V$ be a convex subset of V, then $f : X \to \mathbb{R}$ is **convex** if

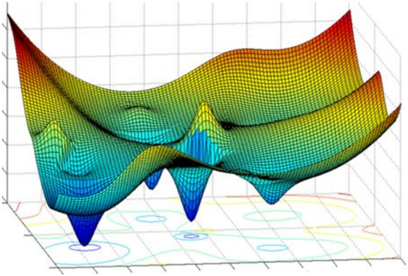
$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y) \quad \text{for all } x, y \in X \text{ and } \theta \in [0,1]$$

Geometrically: line segment between $(x, f(x))$ and $(y, f(y))$ lies above the graph of f



- f is strictly convex if inequality is strict
- if f is convex, $-f$ is **concave**





Verifying convexity

- in 1 or 2 dimensions, plot the function
- use the inequality definition definition
- if f is twice differentiable everywhere, then if for all $x \in X$

$$\nabla^2 f(x) \succeq 0$$

  then f is convex

Examples of convex functions

Scalar functions

- $ax + b$ on $\mathbb{R}$ for any $a, b \in \mathbb{R}$
- $e^{\alpha x}$, any $\alpha \in \mathbb{R}$
- x^α on $\mathbb{R}_+ +$ for $\alpha \geq 1$ or $\alpha \leq 0$
- $|x|^p$ on $\mathbb{R}$ for $p \geq 1$
- relu function: $\max\{0, x\}$

Vector functions

- $a^T x + b$
- $\|x\|$ (any norm)
- softmax function: $\log(e^{x_1} + e^{x_2} + \cdots + e^{x_n})$

Constructive convex analysis

verify convexity by showing that the function is built as follows:

- **non-negative scaling:** f convex, $\alpha \geq 0 \implies \alpha f$ convex
- **summation:** f, g convex $\implies f + g$ convex
- **affine composition:** f convex $\implies f(Ax + b)$ convex
- **pointwise maximum:** $f_1, \ldots f_l$ convex $\implies \max_i \; f_i(x)$ convex
- **convex:** h convex increasing, f convex $\implies h(f(x))$ is convex

Convex optimization

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & g_i(x) \leq 0, \quad i = 1, \ldots, m \\ & a_j^T x = b, \quad j = 1, \ldots, p \end{array}$$

- $x \in \mathbb{R}^n$: **decision variables**
- $f_0 : \mathbb{R}^n \to \mathbb{R}$: **convex objective function**
- $g_i : \mathbb{R}^n \to \mathbb{R}$: **convex inequality constraint functions**
- $h_j : \mathbb{R}^n \to \mathbb{R}$: **inequality constraint functions** – must be affine

some important facts

- the feasible set is a convex set
- all solutions $x^\star$ are **globally optimal**
- the set of solutions forms a convex set

Assumption for this course: if a problem is convex, we can solve it

Theorem

For a convex problem, all local minima are also global minima.

Theorem

For a convex problem, all local minima are also global minima.

Proof.

let $\bar{x}$ be a local minimum: $f_0(\bar{x}) \leq f_0(x)$ for all $x \in \mathcal{B}(\bar{x}, \epsilon)$

assume a contradiction, feasible z such that $f_0(z) < f_0(\bar{x})$

the feasible set is convex, so

$$\theta\bar{x} + (1-\theta)z \quad \text{is feasible for } \theta \in [0,1]$$

as f_0 is convex

$$\begin{aligned} f_0(\theta\bar{x} + (1-\theta)z) &\leq \theta f_0(\bar{x}) + (1-\theta) f_0(z) \\ &< \theta f_0(\bar{x}) + (1-\theta) f_0(\bar{x}) = f_0(\bar{x}) \end{aligned}$$

as $\theta \to 1$, $(\theta\bar{x} + (1-\theta)z) \to \bar{x}$ and we have a contradiction

□

Epigraph formulation

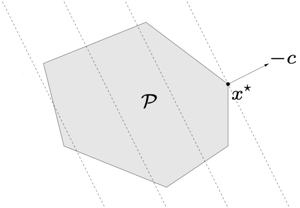
it is often convenient to specify an optimization problem in an equivalent form

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & g_i(x) \leq 0, \quad i = 1, \ldots, m \\ & f_0(x) \leq t \\ & Ax = b \end{array}$$

- t is a new scalar variable
- has the same feasible set as the original problem

Linear programming

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{s.t.} \quad & Gx \preceq h \\ & Ax = b \end{aligned}$$



- feasible set is a **polytope**
- the optimal point, if it exists, is, w.l.o.g at a vertex

[Figure from Boyd & Vandenberghe]

Open-loop output tracking

consider the linear, scalar input/output system

$$y_t = h_0 u_t + h_1 u_{t-1} + h_2 u_{t-3} + \dots \quad u_t = 0 \text{ if } t < 0$$

for $t \in [0, M]$, write in matrix form, write at $y = Hu$, where

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} h_0 & 0 & 0 & \dots & 0 \\ h_1 & h_0 & 0 & \dots & 0 \\ h_2 & h_1 & h_0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ h_M & h_{M-1} & h_{M-2} & \dots & h_0 \\ \vdots & \vdots & \vdots & & 0 \\ h_N & H_{N-1} & h_{N-2} & \dots & H_{N-M} \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_M \end{bmatrix}$$

- design $u_0, u_1, \dots u_M$ to achieve desired output

[Example from Lieven Vandenberghe's EE236A class]

- **Objective:** minimize the maximum deviation from a desired trajectory y_{des}

$$\max_{t \in [0,N]} \quad |y_t - y_{\text{des},t}|$$

- **Constraint 1:** input amplitude bounds

$$|u_i| \le U \quad \text{for } t = 1, \dots, M \qquad \Longleftrightarrow \qquad \|u\|_\infty \le U$$

- **Constraint 2:** input slew-rate bound

$$|u_{t+1} - u_t| \le S \quad \text{for } t = 1, \dots, M-1$$

implement slew rate via the linear inequality $Du \preceq S\mathbf{1}$ where

$$D := \begin{bmatrix} -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 \\ \vdots & & & & & \\ 0 & 0 & \dots & 0 & -1 & 1 \end{bmatrix}$$

LP formulation

$\|z\|_\infty$-norm constraint can be expressed as a **set** of linear inequalities

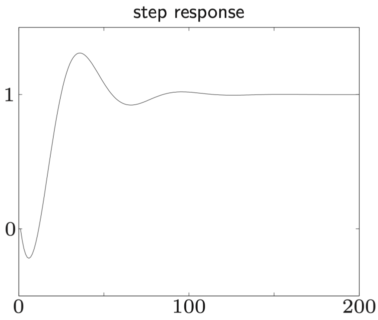
$$z \in \mathbb{R}^2 \quad \|z\|_\infty \le 5 \quad \iff \quad -5 \le z_1 \le 5 \text{ and } -5 \le z_2 \le 5$$

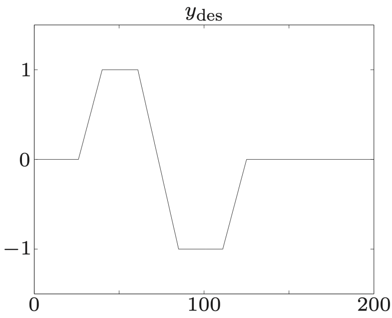
as a result, the optimization problem

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & \|Hu - y_{\text{des}}\|_\infty \\ \text{s.t.} \quad & \|u\|_\infty \le U \\ & \|Du\|_\infty \le S \end{aligned}$$

can be written in "standard form"

$$\begin{aligned} \underset{\gamma, u}{\text{minimize}} \quad & \gamma \\ \text{s.t.} \quad & -\gamma \mathbf{1} \le Hu - y_{\text{des}} \le \gamma \mathbf{1} \\ & -U\mathbf{1} \le u \le U\mathbf{1} \\ & -S\mathbf{1} \le Du \le S\mathbf{1} \end{aligned}$$

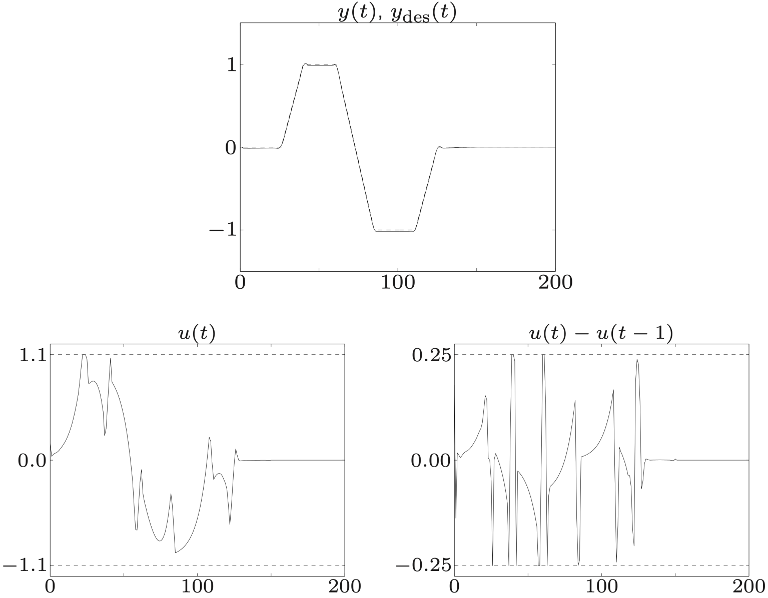




Design specifications:

- y_{des} as shown
- $\|u\|_\infty \leq 1.1$
- $\|u_{t+1} - u_t\|_\infty \leq \frac{1}{4}$
- input horizon: $M = 150$
- output horizon $N = 200$

LP Solution



$y(t)$, $y_{\mathrm{des}}(t)$
1
0
−1
0
100
200
$u(t)$
1.1
0.0
−1.1
0
100
200
$u(t) - u(t-1)$
0.25
0.00
−0.25
0
100
200

Linear quadratic regulator

the output tracking problem is open-loop, i.e.:

- the input at time k is oblivious to the state at time $k-1$
- if the model is wrong (it always is), or there is any disturbance (there always is), the control policy cannot correct for it

The LQR problem

consider the discrete time system

$$x_{t+1} = Ax_t + Bu_t, \quad t = 0, \ldots, N$$

with initial condition $x_0 = x^{\text{init}}$ over **time horizon** N

control objective: pick inputs $u_0, u_1, \ldots, u_{N-1}$ in order to make

- $x_0, x_1, \ldots$ small (i.e., good regulation **regulation** or **control**)
- $u_0, u_1, \ldots$ small (i.e., input efficiency)

these objectives are in competition with each other

Cost function

define the **quadratic** cost function

$$J(u) := \underset{u_0, u_1, \ldots u_{N-1}}{\text{minimize}} \quad \sum_{i=1}^{N-1} (x_t^T Q x_t + u_t^T R u_t) + x_N^T Q_f x_N$$

where $Q \succeq 0$, $Q_f \succeq 0$, and $R \succ 0$ are given

positive (semi)definiteness ensure the minimum possible cost is non-negative

- Q determines the state cost and Q_f the terminal state cost
- R determines the input cost, $R \succ 0$ means any (non-zero) input adds to J

the LQR problem

$$\begin{array}{ll} \underset{u_0, u_1, \ldots u_{N-1}}{\text{minimize}} & \sum_{i=1}^{N-1} (x_t^T Q x_t + u_t^T R u_t) + x_N^T Q_f x_N \\ \text{subject to} & x_{t+1} = A x_t + B u_t, \quad t = 0, \ldots, N \end{array}$$

just a **least squares** problem in disguise–a convex quadratic program

stack the state and control input vectors into augmented vectors

$$X = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_N \end{bmatrix} \quad \text{and} \quad U = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

X is a **linear function** of U:

$$\underbrace{\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix}}_{X} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ B & 0 & 0 & \dots & 0 \\ AB & B & 0 & \dots & 0 \\ A^2B & AB & B & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & A^{N-3}B & \dots & B \end{bmatrix}}_{G} \underbrace{\begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{bmatrix}}_{U} + \underbrace{\begin{bmatrix} I \\ A \\ A^2 \\ A^3 \vdots \\ A^N \end{bmatrix}}_{H} x_0$$

rewrite the dynamics and cost as

$$X = GU + Hx_0$$

$$J(U) = X^T \mathcal{Q} X + U^T \mathcal{R} U$$

with

$$\mathcal{Q} = \mathrm{diag}(Q, \dots, Q, Q_f) \quad \mathcal{R} = \mathrm{diag}(R, \dots, R)$$

substituting X into J gives gives

$$J(U) = (GU + Hx_0)^T \mathcal{Q}(GU + Hx_0) + U^T \mathcal{R} U$$

- problem data: $G \in \mathbb{R}^{Nn \times Nm}$, $H \in \mathbb{R}^{Nn \times n}$, Q, Q_f, R, and x_0
- minimizing $J(U)$ is an **unconstrained least squares problem**
- could solve via QR factorization with cost $O(N^3 nm^2)$
- optimal solution is

$$U^\star = -(G^T \mathcal{Q} G + \mathcal{R})^{-1} G^T \mathcal{Q} G x_0$$

LQR least-squares solution notes

- the solution method described is conceptually very simple
- easily handles time-varying systems

$$x_{t+1} = A_t x_t + B_t u_t$$

- even the least-squares approach to time invariant systems is **not practical**
  - as N increases, so do G and H
- nor is it **robust**, the optimal policy is again**open-loop**

$$-(G^T \mathcal{Q} G + \mathcal{R})^{-1} G^T \mathcal{Q} G x_0$$

- we will show that a dynamic programming formulation provides a closed-loop solution next