Optimal Control

Introduction to Convex Optimization

Instructor: Yanjie Li (李衍杰)
Office: G1011
Email: autolyj@hit.edu.cn

- Teacher:

  Yanjie Li 李衍杰

  Office: G1011

  Email: autolyj@hit.edu.cn

- Teach Assistant

  Xiongtao Shi 石雄涛

  Office: A415

群名称：2022年秋季最优控制

群 号：522737987

About This Course

- Grading System:

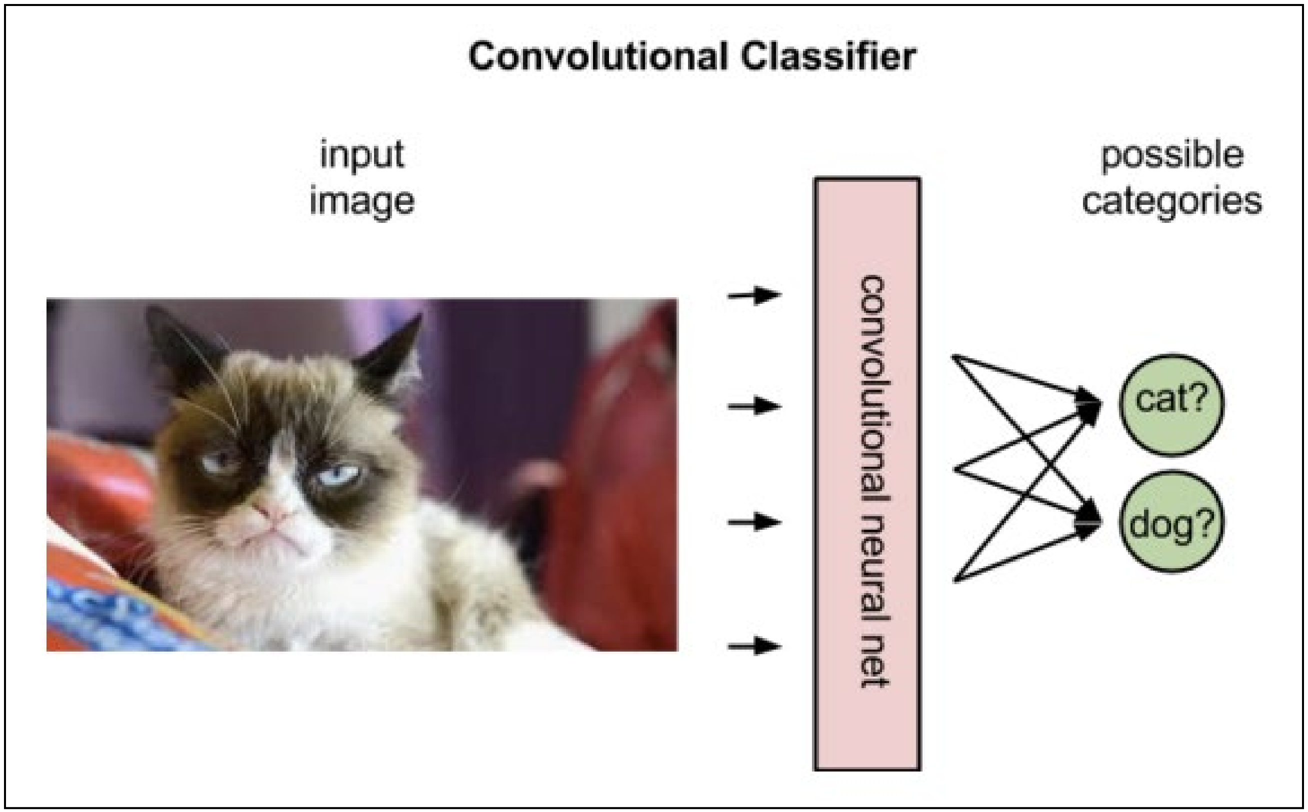
  In-class Quizzes: 10%

  Homework: 20%

  Final Exam: 70%

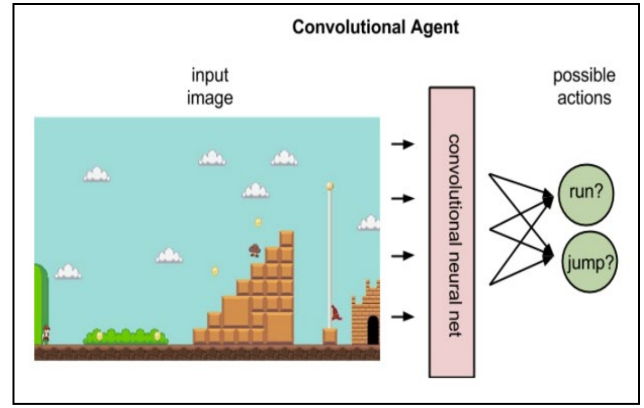
Daily performance

Artificial intelligence (Deep learning)



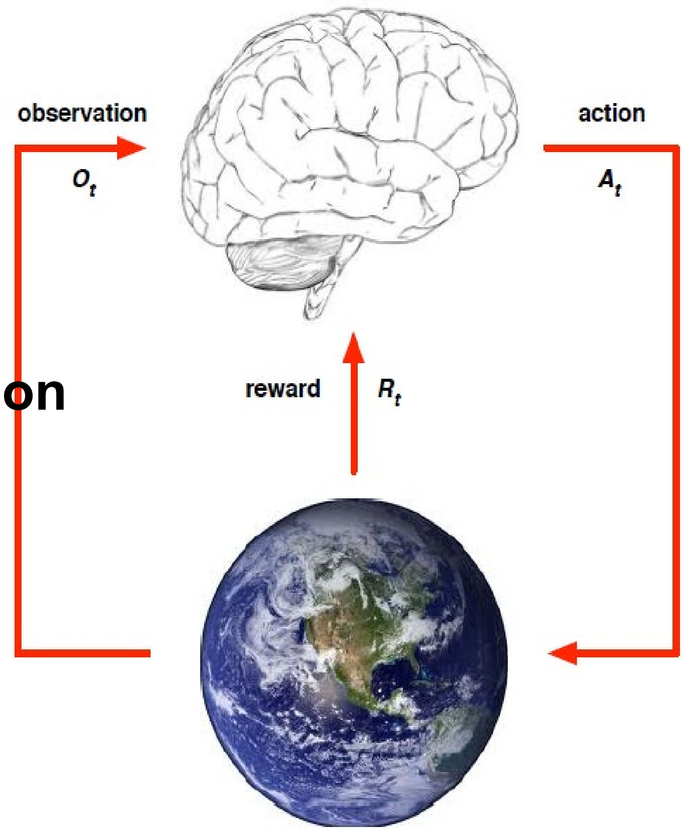
Min $\sum_{i} \left\| f(x_i, \omega) - y_i \right\|$

Artificial intelligence (Reinforcement learning)



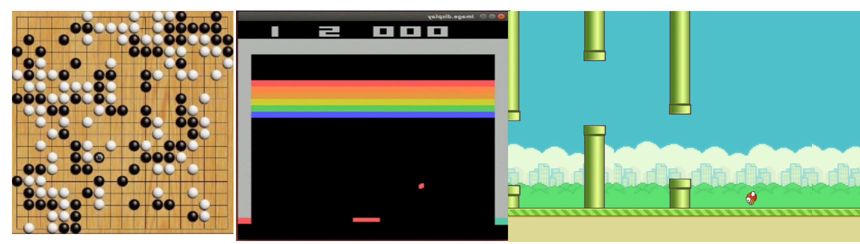
observation

s



action

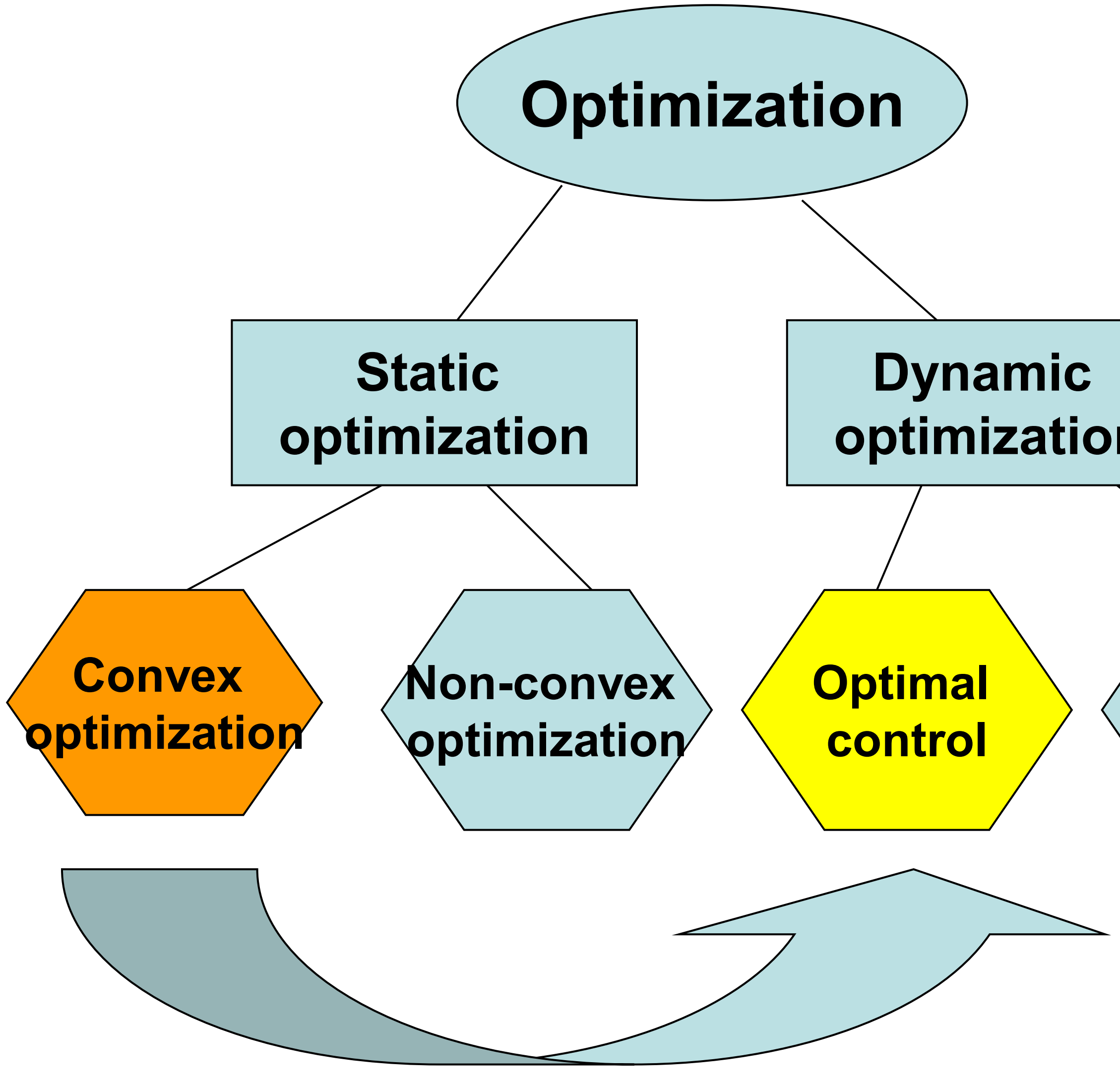
$\pi(s)$



Total reward: $V^{\pi}(s) = \mathrm{E}\left[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots \,\middle|\, s_0 = s, \pi\right].$

Maximization or minimization

Background


Optimization
Static optimization
Dynamic optimization
Convex optimization
Non-convex optimization
Optimal control
Other optimization problems

Content of this Course

- Convex optimization (Static optimization) (70%)
  - Covex set/ Convex function
  - Duality/Algorithms
- Calculus of variations
- Pontryagin maximum principle
- Bellman dynamic programming

(Dynamic optimization) (30%)

Content

- Static optimization
  - A simple example
  - General mathematical formulation
- Least squares problem
  - An example
- Linear programming
  - An example
- Convex optimization

Portfolio optimization

- Portfolio optimization
  - X : Capital
  - x_i : Investment in the i assets
  - Constraints:

$$x_1 + x_2 + \cdots + x_n \le X$$

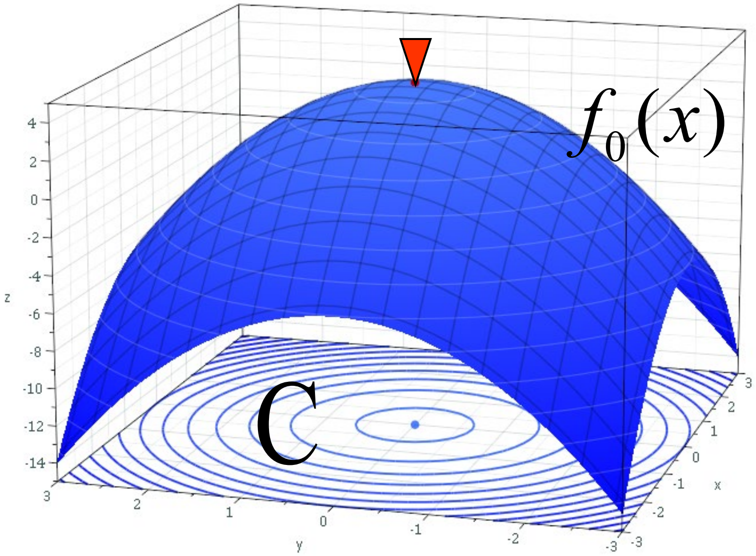
$$x_i \ge 0, i = 1, 2, 3, \ldots, n$$

  - Objective performance:
    - Max gain

$$\sum_{i=1}^{n} f_i(x_i)$$

Static optimization

- Choosing the best element from some set of available alternatives



$$\max_{x} \quad f_0(x)$$

$$s.t. \quad x \in C \subset \mathrm{R}^n$$

General mathematical formulation

Optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq b_i, \quad i = 1, \ldots, m \end{array}$$

- $x = (x_1, \ldots, x_n)$: optimization variables
- $f_0 : \mathbf{R}^n \to \mathbf{R}$: objective function
- $f_i : \mathbf{R}^n \to \mathbf{R}$, $i = 1, \ldots, m$: constraint functions

Optimal Solution: $x^\star$ has smallest value of f_0 among all vectors that satisfy the constraints

Solving optimization problem

General optimization problem

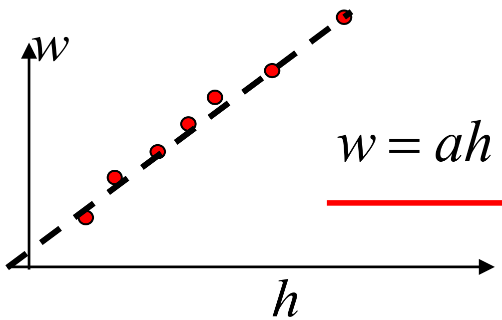
- **very difficult to solve**
- methods involve some **compromise**, *e.g*
  - very long computation time
  - not always finding the solution

What problems are easy to solve?

- least-squares problems
- linear programming problems

Least-squares problems (LS)

Example: weight & height (Assumption: Linear)



$$w = ah + b$$

$$\boxed{\begin{aligned} ah_1 + b &= w_1 \\ ah_2 + b &= w_2 \\ &\vdots \\ ah_k + b &= w_k \end{aligned}}$$

Matrix form

$$\begin{bmatrix} h_1 & 1 \\ h_2 & 1 \\ \vdots & \vdots \\ h_k & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_k \end{bmatrix}$$

$$Ax = b$$

LS: minimize $\|Ax - b\|_2^2$

Solving least-squares problems

- analytical solution: $x^\star = (A^T A)^{-1} A^T b$
- reliable and efficient algorithms and software

Linear programming problems (LP)

Example

- **diet problem:** choose quantities $x_1, \ldots, x_n$ of n foods
  - one unit of food j costs c_j, contains amount a_{ij} of nutrient i
  - healthy diet requires nutrient i in quantity at least b_i

to find cheapest healthy diet,

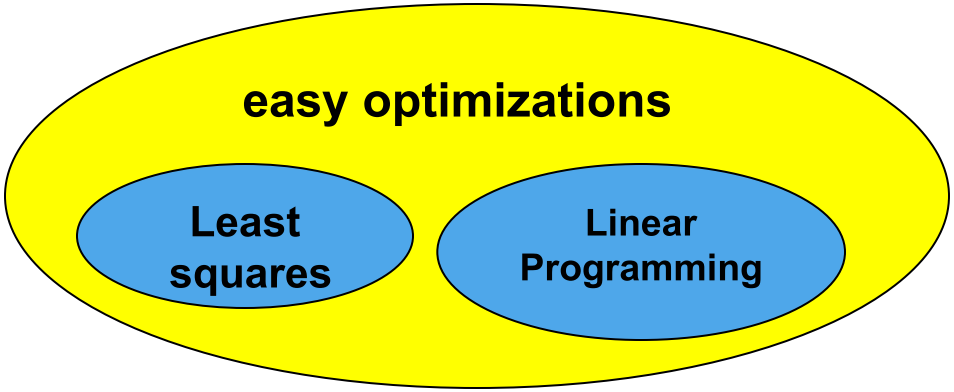
LP:

$$\boxed{\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \succeq b, \quad x \succeq 0 \end{array}}$$

Solving linear programming problem

- no analytical formula for solution
- reliable and efficient algorithms (simplex) and software

Extend LS and LP



Linear function:

$$f(\alpha x+\beta y)=\alpha f(x)+\beta f(y)$$

2-norm function:

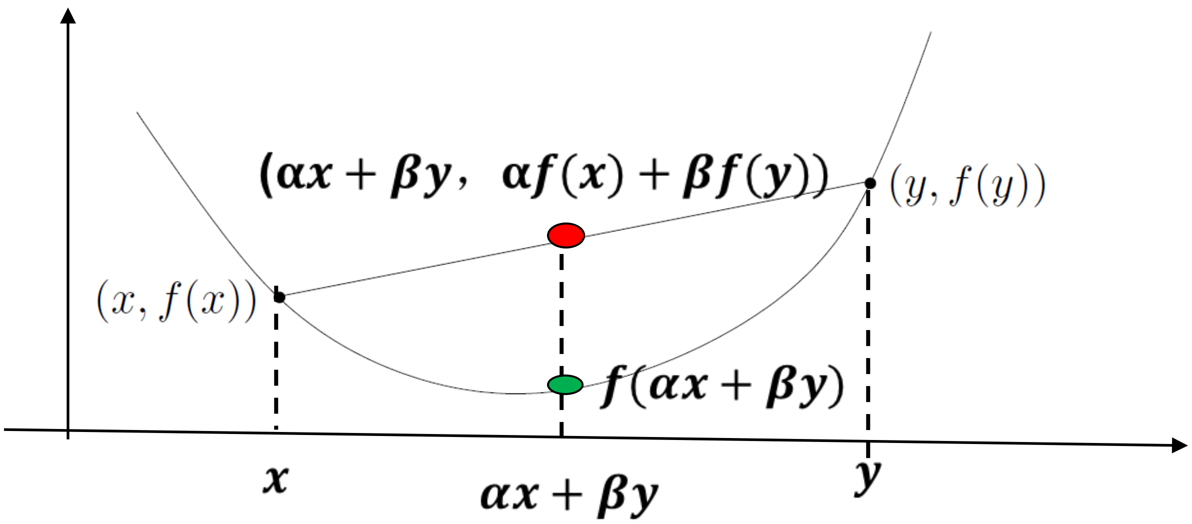
$$\|\alpha x+\beta y\|_2 \le \alpha\|x\|_2+\beta\|y\|_2$$

$$f(\alpha x+\beta y)\le \alpha f(x)+\beta f(y)$$

Convex function

Convex function $f(\alpha x+\beta y)\le \alpha f(x)+\beta f(y)$

if $\alpha+\beta=1,\ \alpha\ge 0,\ \beta\ge 0$



Convex optimization

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq b_i, \quad i = 1, \dots, m \end{array}$$

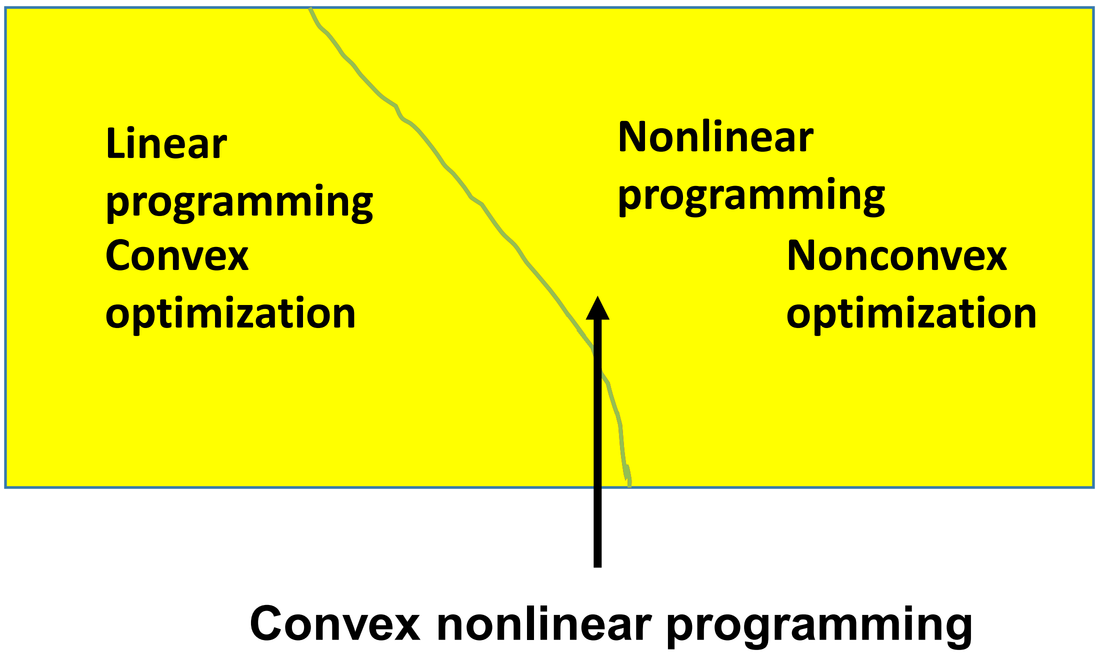
- objective and constraint functions are convex

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y)$$

if $\alpha + \beta = 1$, $\alpha \geq 0$, $\beta \geq 0$

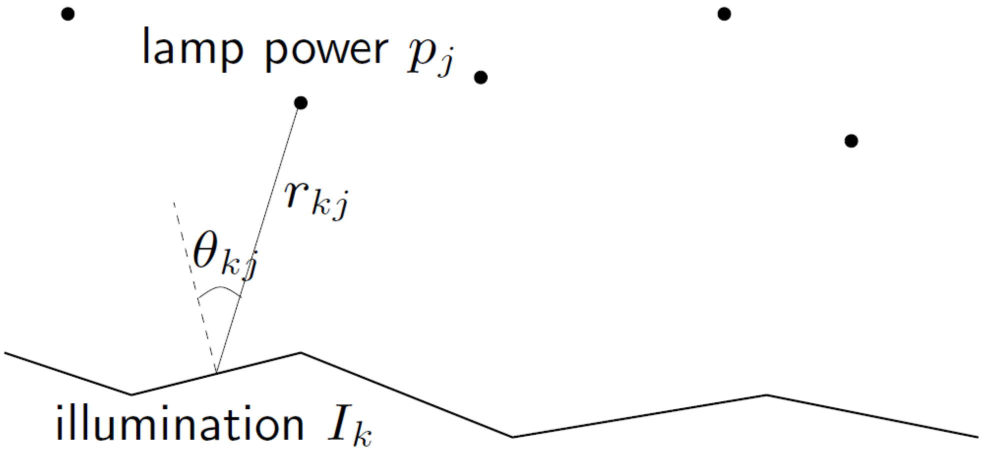
Summary

Static optimization

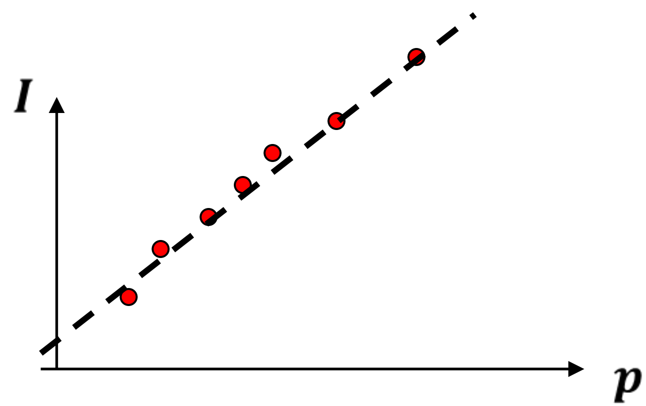


An engineering example

m lamps illuminating n (small, flat) patches



Model



intensity I_k at patch k depends linearly on lamp powers p_j:

$$I_k = \sum_{j=1}^{m} a_{kj} p_j, \qquad a_{kj} = r_{kj}^{-2} \max\{\cos\theta_{kj}, 0\}$$

problem: achieve desired illumination I_{des} with bounded lamp powers

$$\mathbf{0 \leq p_j \leq p_{max}}$$

Optimization

Researching

1. **Use uniform power:** $p_j = p$, vary p

2. **Use least-squares:**

$$\text{minimize} \quad \sum_{k=1}^{n} (I_k - I_{\text{des}})^2$$

round p_j if $p_j > p_{\max}$ or $p_j < 0$

3. **Use weighted least-squares:**

$$\text{minimize} \quad \sum_{k=1}^{n} (I_k - I_{\text{des}})^2 + \sum_{j=1}^{m} w_j (p_j - p_{\max}/2)^2$$

iteratively adjust weights w_j until $0 \le p_j \le p_{\max}$

- easy to recognize

Linear programming

$$\begin{array}{ll} \text{minimize} & \max_{k=1,\ldots,n} |I_k - I_{\text{des}}| \\ \text{subject to} & 0 \le p_j \le p_{\max}, \quad j = 1, \ldots, m \end{array}$$

Trick

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & |I_k - I_{des}| \le t \\ & 0 \le p_j \le p_{\max}, \quad j = 1, \ldots, m \end{array}$$

$$\begin{cases} \min \| x \|_1 \\ s.t.\ Ax = b \end{cases}$$

$$\begin{cases} \min e^T x^+ + e^T x^- \\ s.t.\ Ax^+ - Ax^- = b \\ x^+ \geqslant 0,\ x^- \geqslant 0 \end{cases}$$

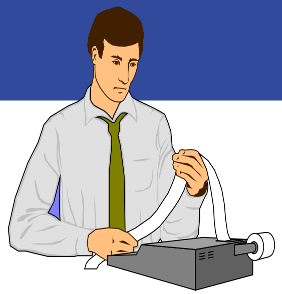
$$x_j^+ = \frac{1}{2}(|x_j| + x_j),\quad x_j^- = \frac{1}{2}(|x_j| - x_j)$$

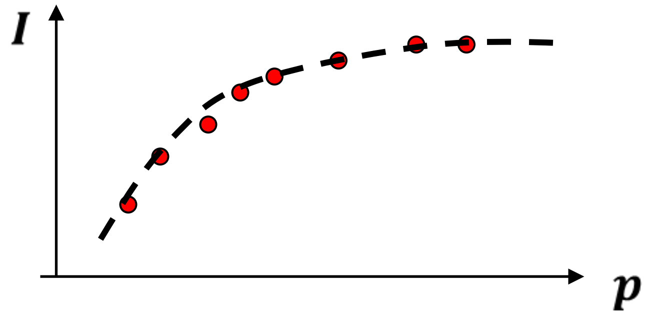
Problem

$$\text{minimize} \quad \sum_{k=1}^{n} |I_k - I_{des}|$$

$$s.t. \quad 0 \le p_j \le p_{\max},\ j = 1, 2, \ldots, m$$

Convex optimization

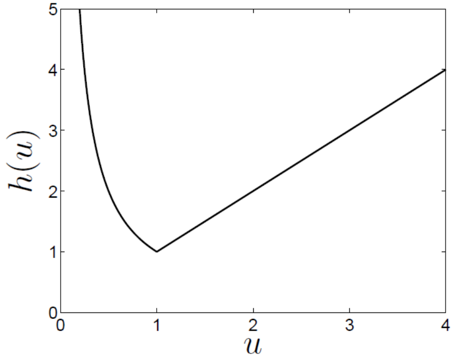




$$\begin{array}{ll} \text{minimize} & \max_{k=1,\ldots,n} |\log I_k - \log I_{\mathrm{des}}| \\ \text{subject to} & 0 \le p_j \le p_{\max}, \quad j = 1, \ldots, m \end{array}$$

$$\begin{array}{ll} \text{minimize} & f_0(p) = \max_{k=1,\ldots,n} h(I_k/I_{\mathrm{des}}) \\ \text{subject to} & 0 \le p_j \le p_{\max}, \quad j = 1, \ldots, m \end{array}$$

with $h(u) = \max\{u, 1/u\}$



f_0 is convex because maximum of convex functions is convex

Problems

additional constraints: does adding 1 or 2 below complicate the problem?

1. no more than half of total power is in any 10 lamps
2. no more than half of the lamps are on $(p_j > 0)$

- answer: with (1), still easy to solve; with (2), extremely difficult
- moral: (untrained) intuition doesn't always work; without the proper background very easy problems can appear quite similar to very difficult problems

$$I(p_1 > 0) + I(p_2 > 0) + \cdots + I(p_m > 0) \leq m/2$$

Research is going on!

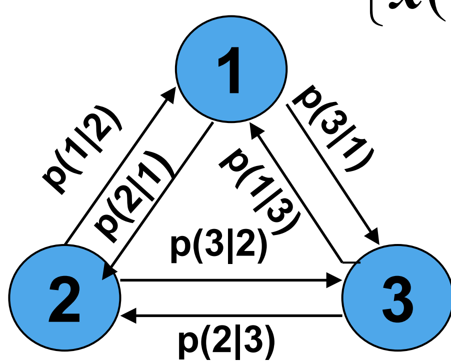
Dynamic optimization

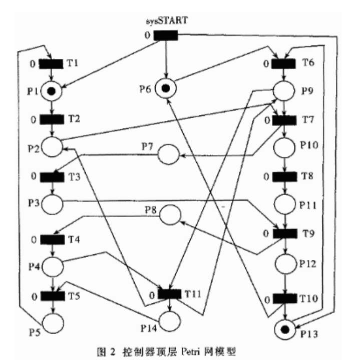
❖ Dynamic process $x(t)$

Max **performance** $$g_0(\boldsymbol{u}) = \Phi_0(\boldsymbol{x}(T)) + \int_0^T L_0(t, \boldsymbol{x}(t), \boldsymbol{u}(t))dt$$

s.t.

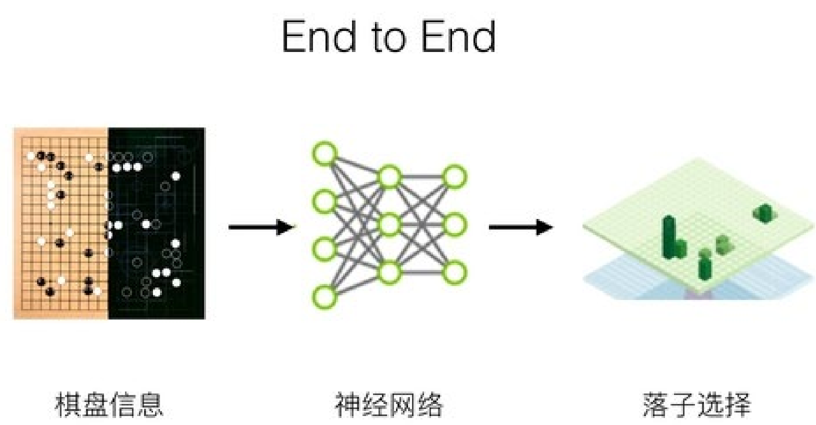
State evolution rule $$\begin{cases} \dfrac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(t,\ \boldsymbol{x}(t),\ \boldsymbol{u}(t)), \quad t \in (0,T] \\ \boldsymbol{x}(0) = \boldsymbol{x}^0 \end{cases}$$

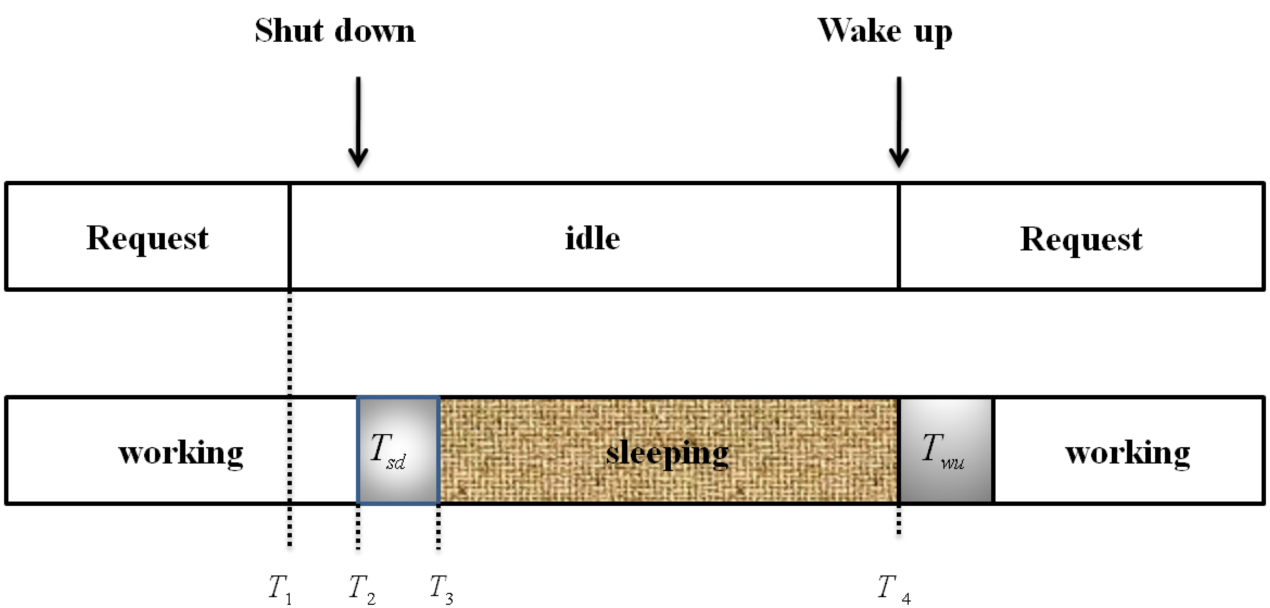




Find a sequence or functional to optimize the performance

Two Examples





Standard Optimal Control Problem

Minimize $g_0(\boldsymbol{u}) = \Phi_0(\boldsymbol{x}(T)) + \int_0^T L_0(t, \boldsymbol{x}(t), \boldsymbol{u}(t))dt$

subject to

State equation

$$\begin{cases} \dfrac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(t,\ \boldsymbol{x}(t),\ \boldsymbol{u}(t)), \quad t \in (0, T] \\ \boldsymbol{x}(0) = \boldsymbol{x}^0 \end{cases}$$

Terminal constraint

$$\psi_i(\boldsymbol{x}(T)) \ \le\ 0, \ \ \text{and / or} \ = 0, \ \ i = 1, \ldots, N_T$$

Continuous constraint

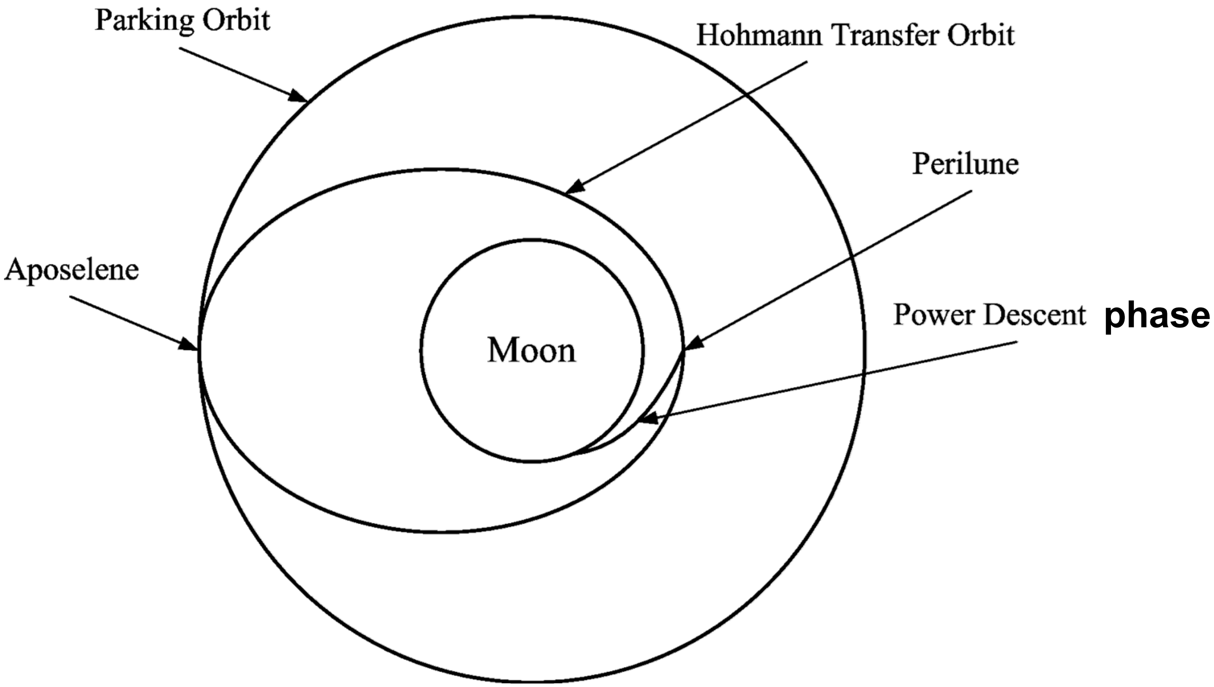
$$h_i(t,\ \boldsymbol{x}(t)) \ \le\ 0, \ \ t \in [0,\ T], \ \ i = 1, \ldots, N$$

Control constraint

$$a_i \le u_i(t) \le b_i, \ \ \forall t \in [0, T], \ \ i = 1, \ldots, r.$$

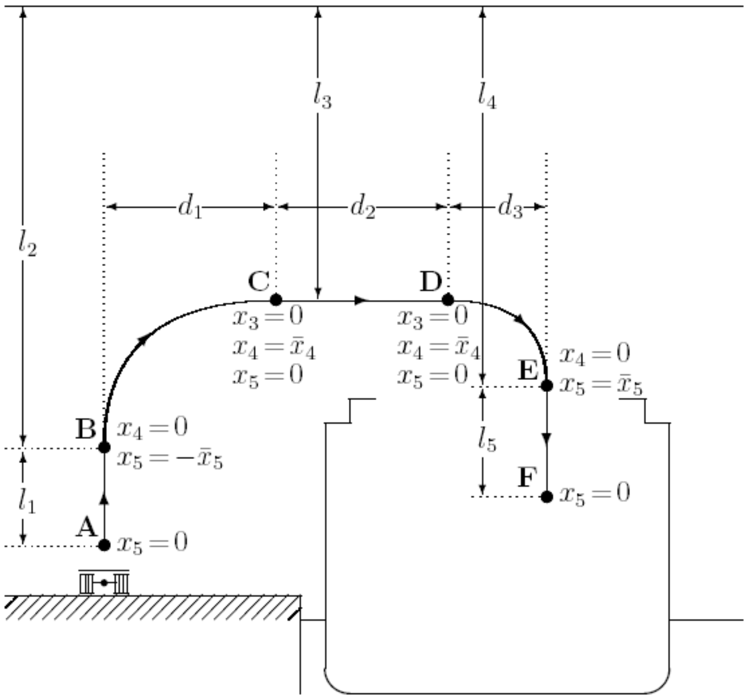
$$\boldsymbol{x} = [x_1, \ldots, x_n]^T, \boldsymbol{u} = [u_1, \ldots, u_r]^T$$

Lunar Module Soft Landing

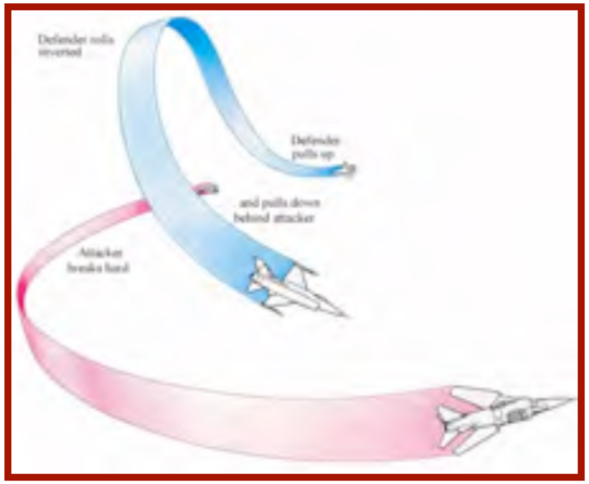


Aposelene: 100km above the moon surface
Perilune: 15km above the moon surface

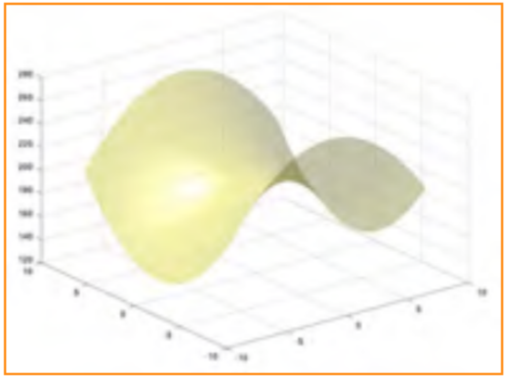
Optimal Control of Container Cranes



Example: Pursuit-Evasion: A Competitive Optimization Problem



- Pursuer's goal: minimize final miss distance
- Evader's goal: maximize final miss distance



- "Minimax" (saddle-point) cost function
- Optimal control laws for pursuer and evader

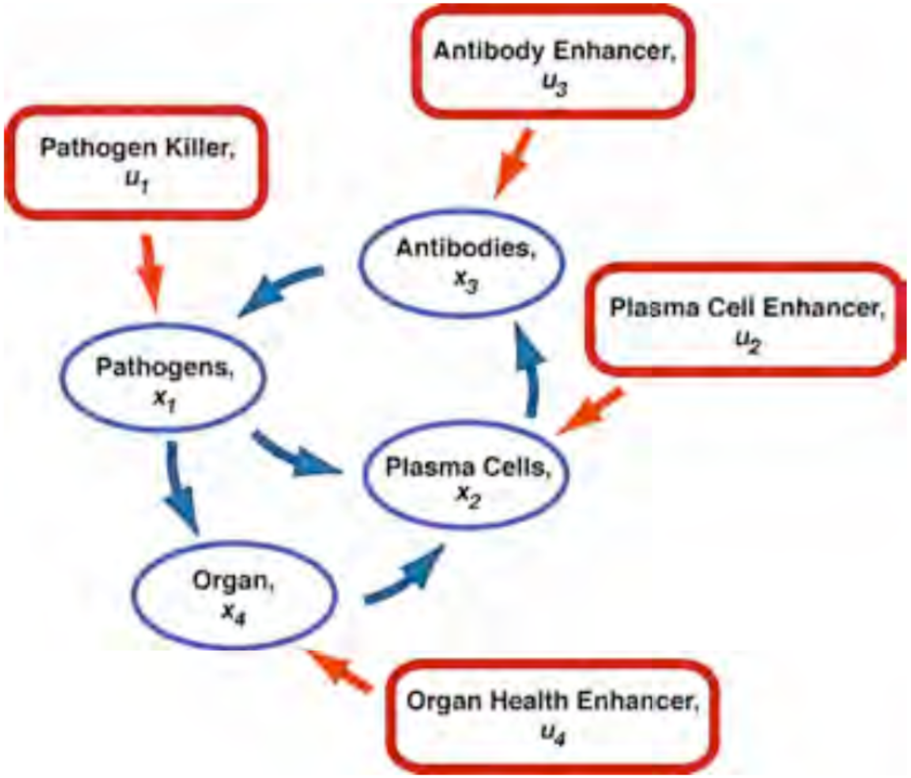
$$\mathbf{u}(t) = \begin{bmatrix} \mathbf{u}_P(t) \\ \mathbf{u}_E(t) \end{bmatrix} = -\begin{bmatrix} \mathbf{C}_P(t) & \mathbf{C}_{PE}(t) \\ \mathbf{C}_{EP}(t) & \mathbf{C}_E(t) \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}_P(t) \\ \hat{\mathbf{x}}_E(t) \end{bmatrix}$$

- Example of a *differential game*, Isaacs (1965), Bryson & Ho (1969)

From Lecture note of Princeton University

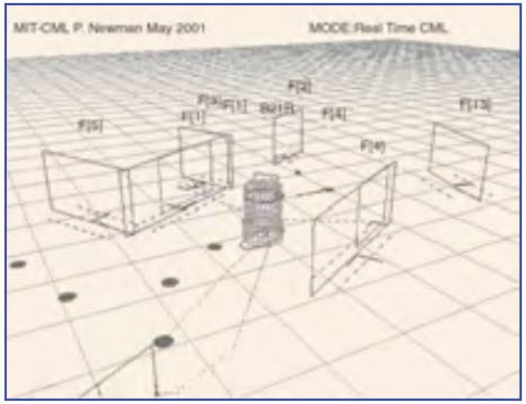
Example: Minimize Concentrations of Virus, Infected Cells, and Drug Usage

- x_1 = Concentration of a pathogen, which displays antigen
- x_2 = Concentration of plasma cells, which are carriers and producers of antibodies
- x_3 = Concentration of antibodies, which recognize antigen and kill pathogen
- x_4 = Relative characteristic of a damaged organ [0 = healthy, 1 = dead]

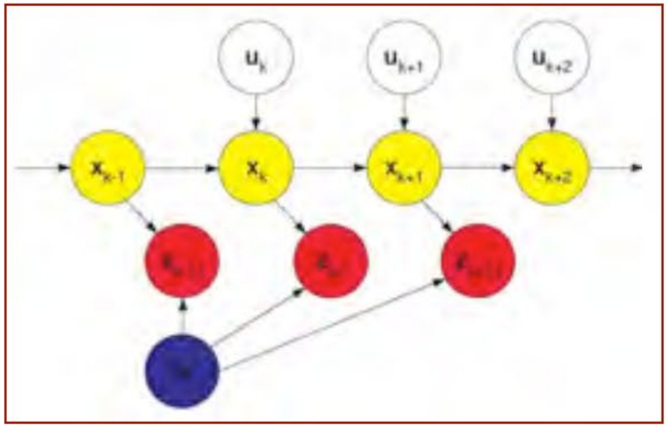


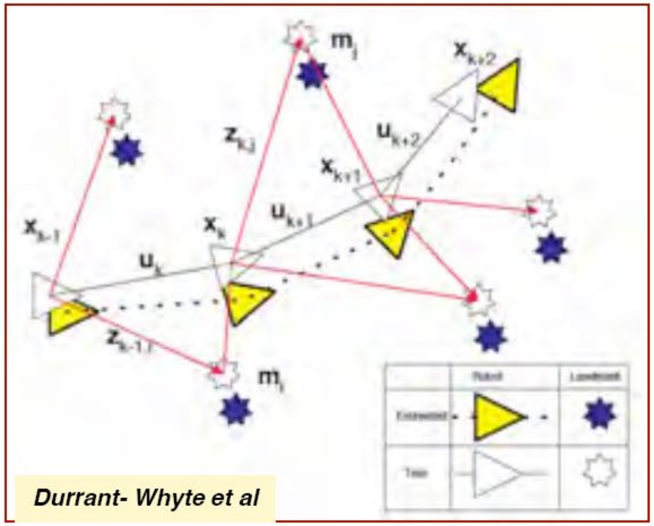
From Lecture note of Princeton University

<u>Example</u>: Simultaneous Location and Mapping (SLAM)



- Build or update a local map within an unknown environment
  - Stochastic map, defined by mean and covariance
  - SLAM Algorithm = State estimation with extended Kalman filter
  - Landmark and terrain tracking

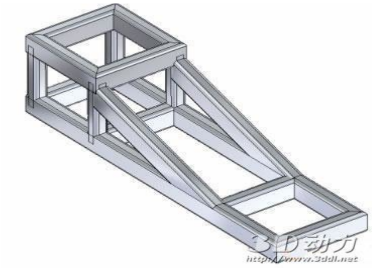


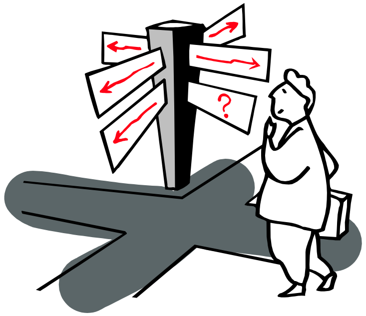


From Lecture note of Princeton University

Conclusion

- **Static**
  - Optimal state, x*, and control, u*, are fixed, i.e., they do not change over time
    - $J^* = J(x^*, u^*)$
    - Functional minimization (or maximization)
    - Parameter optimization
- **Dynamic**
  - Optimal state and control vary over time
    - $J^* = J[x^*(t), u^*(t)]$
    - Optimal trajectory
    - Optimal feedback strategy
- **Optimized cost function, J^*, is a scalar, real number in both cases**

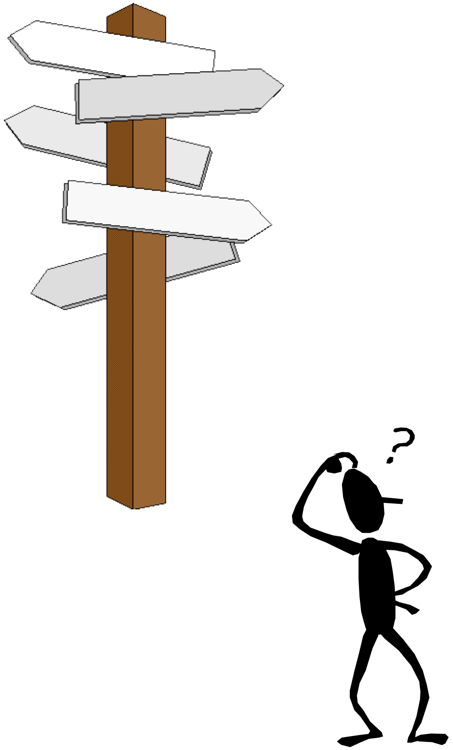




What is optimization

- Choice of best parameter
- Choice of best strategy
- Choice of best control
- Choice of best estimate

Choosing the best element from some set of available alternatives



Thank You