# Convex Optimization in Engineering Analysis and Design

Stephen Boyd

Electrical Engineering Department, Stanford

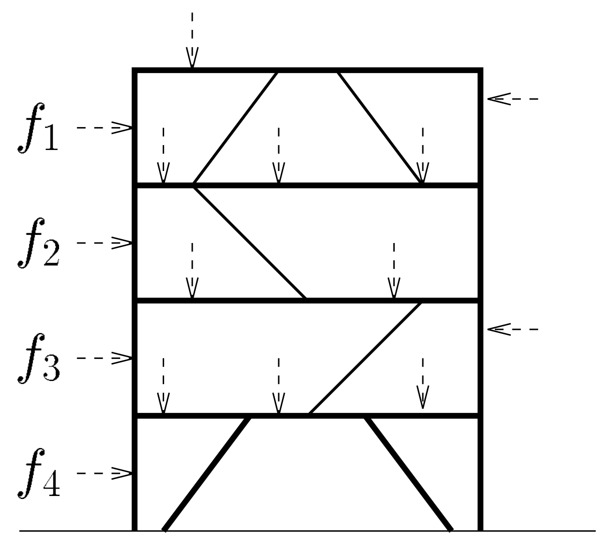
MIT EE Colloquium Lecture 3/14/94

# Basic idea

- Many problems arising in engineering analysis and design can be cast as **convex optimization problems**
- Hence, are **fundamentally tractable**
- Recent interior-point methods can exploit problem structure to solve such problems **very efficiently**

# Example 1

- linear elastic structure; forces $f_1, \ldots, f_{100}$ induce deflections $d_1, \ldots, d_{300}$
- $0 \leq f_i \leq F_i^{\max}$, several hundred other constraints: max load per floor, max wind load per side, etc.




**Problem 1a:** find worst-case deflection, *i.e.*, $\max_i |d_i|$

**Problem 1b:** find worst-case deflection, with each force "on" or "off", *i.e.*, $f_i = 0$ or $F_i^{\max}$

# Example 1

Problem 1a is **very easy**

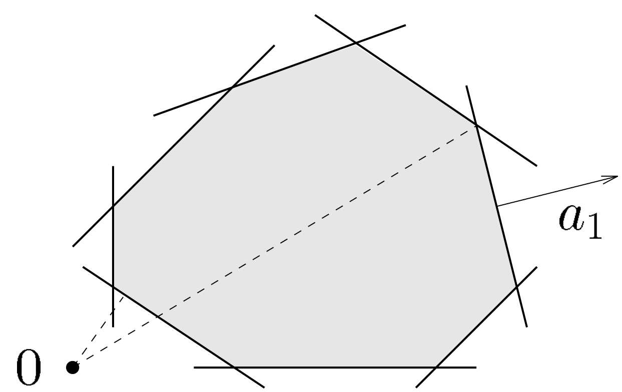
- readily solved in a few minutes on small workstation
- general problem has polynomial complexity

Problem 1b is **very difficult**

- could take weeks to solve ...
- general problem NP-complete

# Example 2

Polytope described by linear inequalities, $a_i^T x \leq b_i$, $i = 1, \ldots, L$




**Problem 2a:** find point **closest** to origin, *i.e.*, $\min \|x\|$

**Problem 2b:** find point **farthest** from origin, *i.e.*, $\max \|x\|$

# Example 2

Problem 2a is **very easy**

- readily solved on small workstation
- polynomial complexity

Problem 2b is **very difficult**

- difficult even with supercomputer
- NP-complete

moral:

**very difficult** and **very easy** problems can look quite similar

# Outline

- **Convex optimization**
- Some examples
- Interior-point methods

# Convex optimization

minimize $f_0(x)$

subject to $f_1(x) \le 0, \ldots f_L(x) \le 0$

$f_i : \mathbf{R}^n \to \mathbf{R}$ are convex, *i.e.*, for all $x$, $y$, $0 \le \lambda \le 1$,

$$f_i(\lambda x + (1-\lambda)y) \le \lambda f_i(x) + (1-\lambda) f_i(y)$$

- can have linear equality constraints
- differentiability not needed
- examples: Linear Programs (LPs), problems 1a, 2a
- other formulations possible (feasibility, multicriterion)

(Roughly speaking,)

# Convex optimization problems are fundamentally tractable

- computation time is small, grows gracefully with problem size and required accuracy
- large problems solved quickly in practice
- supported by strong theoretical results (Nemirovsky and Yudin)
- not widely enough appreciated

What "solve" means:

- find **global** optimum within a given tolerance, or,
- find **proof** (certificate) of infeasibility

Algorithms:

- Classical optimization algorithms **do not** work
- Ellipsoid algorithm
  - **very simple, universally applicable**
  - efficient in terms of worst-case complexity theory
  - slow but robust in practice
- (General) interior-point methods (more later ...)
  - efficient in theory and practice

# Outline

- Convex optimization
- **Some examples**
- Interior-point methods

# Well-known example: FIR filter design

transfer function: $H(z) \triangleq \sum_{i=0}^{n} h_i z^{-i}$

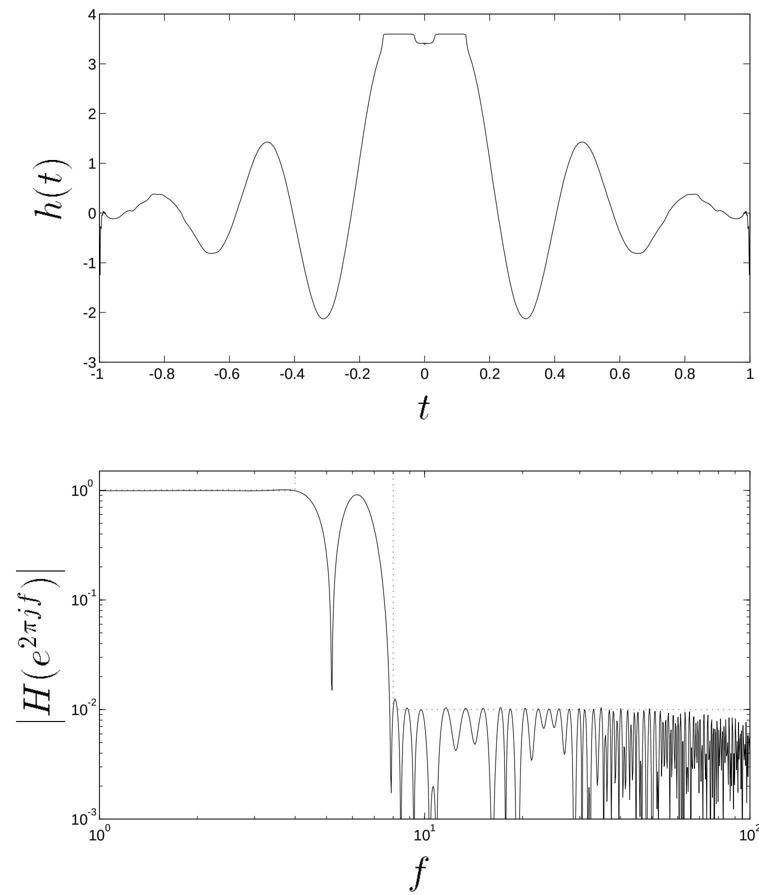
design variables: $x \triangleq [h_0\ h_1\ \ldots\ h_n]^T$

sample convex constraints:

- $H(e^{j0}) = 1$ (unity DC gain)
- $H(e^{j\omega_0}) = 0$ (notch at $\omega_0$)
- $|H(e^{j\omega})| \leq 0.01$ for $\omega_s \leq \omega \leq \pi$
  (min. 40dB atten. in stop band)
- $|H(e^{j\omega})| \leq 1.12$ for $0 \leq \omega \leq \omega_b$
  (max. 1dB upper ripple in pass band)
- $h_i = h_{n-i}$ (linear phase constraint)
- $s(t) \triangleq \sum_{i=0}^{t} h_i \leq 1.1 H(e^{j0})$ (max. 10% step response overshoot)

# FIR filter design example (M. Grant)

- sample rate $2/n$ sec$^{-1}$
- linear phase
- max $\pm$1dB ripple
  up to 0.4Hz
- min 40dB atten
  above 0.8Hz
- minimize $\max_i |h_i|$
- some solution times:
  $n = 255$: 5 sec
  $n = 2047$: 4 min



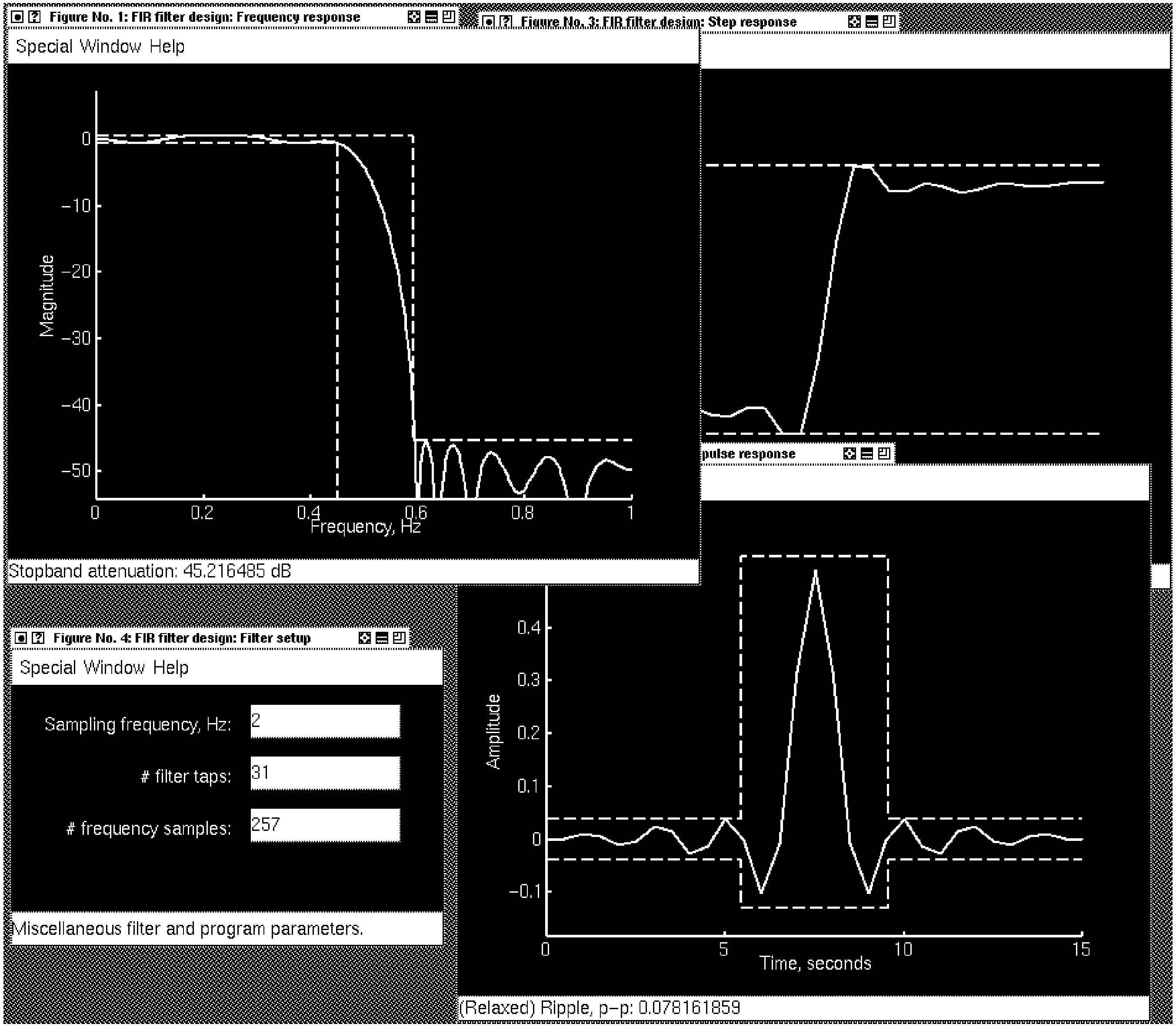
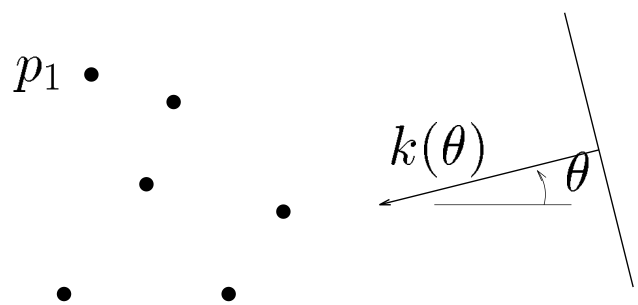


Figure No. 1: FIR filter design: Frequency response
Special Window Help
Magnitude
0
−10
−20
−30
−40
−50
0
0.2
0.4
0.6
0.8
1
Frequency, Hz
Stopband attenuation: 45.216485 dB
Figure No. 3: FIR filter design: Step response
pulse response
Amplitude
0.4
0.3
0.2
0.1
0
−0.1
0
5
10
15
Time, seconds
(Relaxed) Ripple, p−p: 0.078161859
Figure No. 4: FIR filter design: Filter setup
Special Window Help
Sampling frequency, Hz: 2
# filter taps: 31
# frequency samples: 257
Miscellaneous filter and program parameters.

# Beamforming

omnidirectional antenna elements at positions $p_1, \ldots, p_n \in \mathbf{R}^2$
plane wave incident from angle $\theta$:

$$\exp j(k(\theta)^T p - \omega t), \qquad k(\theta) = -[\cos\theta \ \ \sin\theta]^T$$




demodulate to get $y_i = \exp(jk(\theta)^T p_i)$
form weighted sum $y(\theta) = \sum_{i=1}^{n} w_i y_i$

design variables: $x = [\mathbf{Re}\, w^T \ \mathbf{Im}\, w^T]^T$
(antenna array weights or shading coefficients)

$G(\theta) \triangleq |y(\theta)|$ antenna gain pattern

Sample convex constraints:

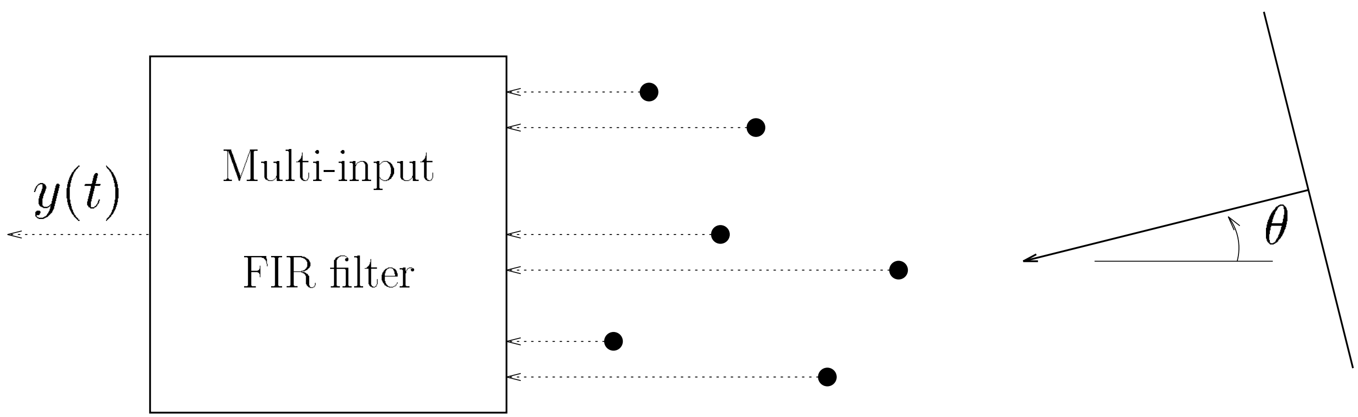
- $y(\theta_t) = 1$ (target direction normalization)
- $G(\theta_0) = 0$ (null in direction $\theta_0$)
- $w$ is real (amplitude only shading)
- $|w_i| \leq 1$ (attenuation only shading)

Sample convex objectives:

- $\max \{G(\theta) \mid |\theta - \theta_t| \geq 5^\circ\}$
  (sidelobe level with $10^\circ$ beamwidth)
- $\sigma^2 \sum_i |w_i|^2$ (noise power in $y$)

# Acoustic array filtering

acoustic field: sum of plane waves (2D for simplicity)
omnidirectional microphones, sampled output $u_i(t)$




FIR filter and add: $y(t) \triangleq \sum_{i=1}^{n_{\text{mic}}} \sum_{\tau=1}^{n_{\text{tap}}} h_{i\tau} u_i(t-\tau)$

design variables: $x \triangleq [h_{11} \ldots h_{n_{\text{mic}} n_{\text{tap}}}]^T$

incidence-angle dependent transfer function: $H(\theta, \omega)$

Sample (convex) specification:

- $|H(\theta_t,\omega) - H_{\text{des}}(\omega)| / |H_{\text{des}}(\omega)| \leq 0.1$
  (max. 10% deviation from desired TF in target direction)

Sample (convex) objective:

- $\max\{|H(\theta,\omega)| \omega_L \leq \omega \leq \omega_U,\ |\theta - \theta_t| \geq 10^\circ\}$
  (broadband attenuation outside 20° beamwidth)

# Open-loop trajectory planning

Discrete-time linear system, input $u(t) \in \mathbf{R}^p$, output $y(t) \in \mathbf{R}^q$

Sample convex constraints:

- $|u_i(t)| \le U$ (limit on input amplitude)
- $|u_i(t+1) - u_i(t)| \le S$ (limit on input slew rate)
- $l_i(t) \le y_i(t) \le u_i(t)$ (envelope bounds for output)

Sample convex objective:

- $\max_{t,i} |y_i(t) - y_i^{\mathrm{des}}(t)|$ (peak tracking error)

# Robust open-loop trajectory planning

input must work well with **multiple plants**

one input $u$ applied to $L$ plants; outputs are $y^{(1)}, \ldots, y^{(L)}$

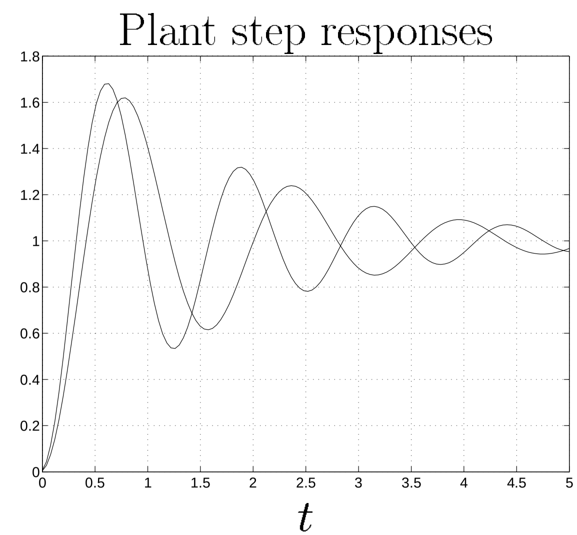
constraints are to hold for **all** $y^{(i)}$

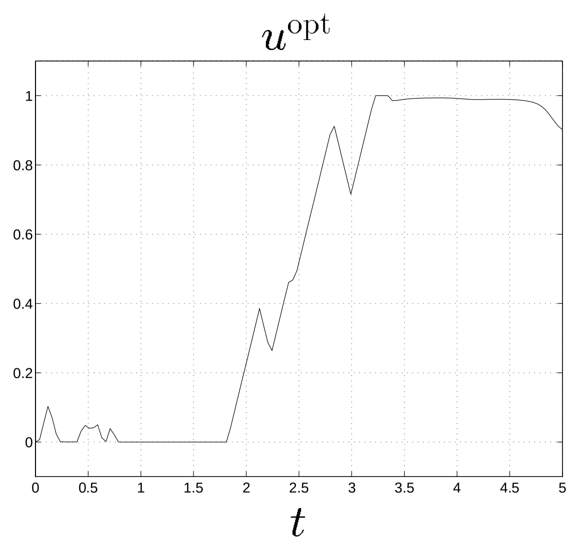
sample objectives:

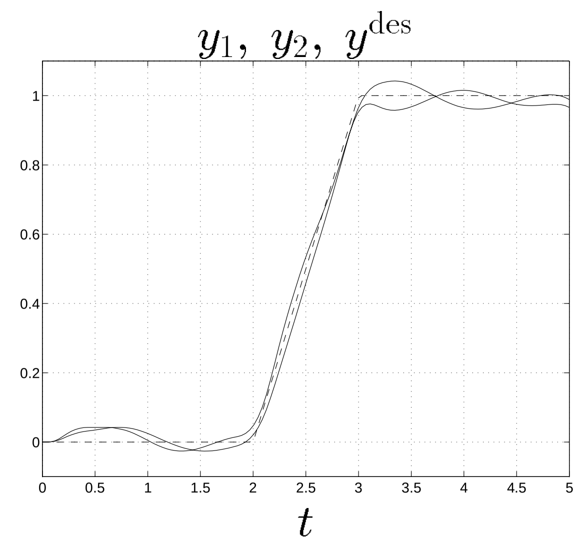
- weighted sum of objectives for each $i$ (average performance)
- max over objectives for each $i$ (worst-case performance)

# Example

- two plants
- $0 \le u(t) \le 1$
- $|\Delta u(t)| \le 1.25/\text{sec}$
- minimize worst-case peak tracking error
- 128 time samples (variables)

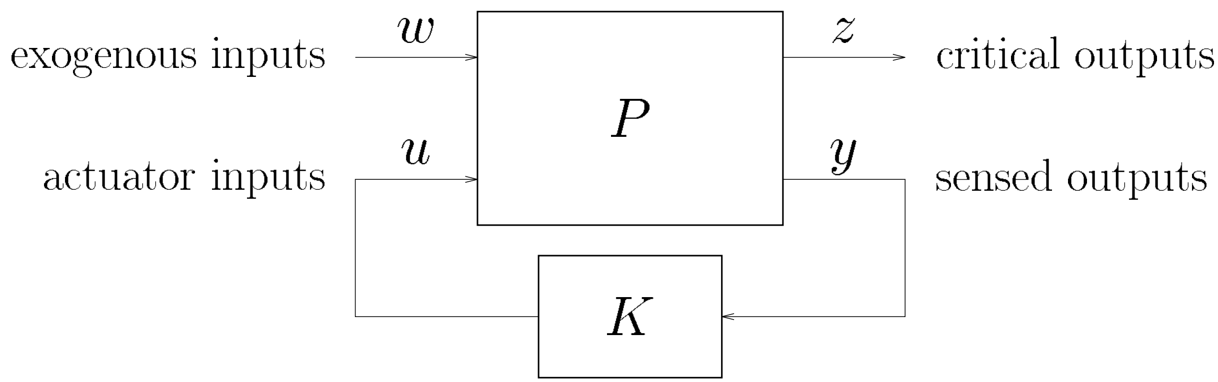


# Linear controller design

(static case for simplicity)




linear plant $P$ given; design linear feedback controller $K$

closed-loop I/O relation: $z = Hw$,

$$H = P_{zw} + P_{zu}K(I - P_{yu}K)^{-1}P_{yw}$$

most specifications, objectives convex in $H$, **not** $K$

# Transform to convex problem

Linear-fractional (projective) transformation:

$$Q \triangleq K(I - P_{yu}K)^{-1}$$

$H = P_{zw} + P_{zu}QP_{yw}$: constraints, objectives are convex in $Q$!

- design $Q$ via convex programming
- set $K = Q(I + P_{yu}Q)^{-1}$

Extends to dynamic case ...

- time and frequency domain limits on actuator effort, regulation, tracking error
- some robustness specifications

# Other examples

- synthesis of Lyapunov functions, state feedback
- filter/controller realization
- system identification problems
- truss design
- VLSI transistor sizing
- design centering
- nonparametric statistics
- computational geometry

# Outline

- Convex optimization
- Some examples
- **Interior-point methods**

# Interior-point convex programming methods

History:

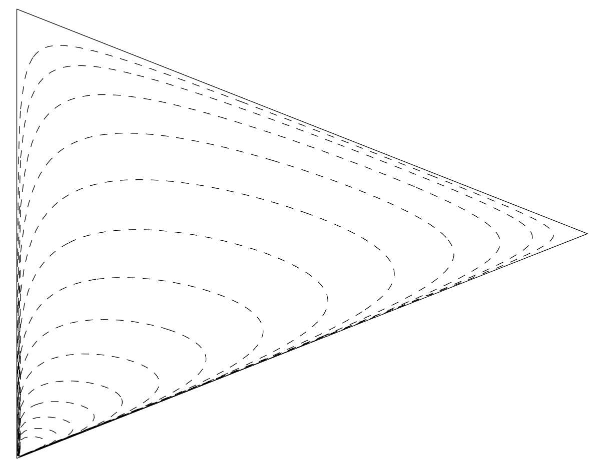
- Dikin; Fiacco & McCormick's SUMT (1960s)
- Karmarkar's LP algorithm (1984); many more since then
- Nesterov & Nemirovsky's general formulation (1989)

General:

- # iterations small, grows slowly with problem size (typical number: 10s)
- each iteration is basically least-squares problem

# Basic idea

Choose (potential fct.) $\varphi$ s.t.



- $\varphi$ smooth
- $\varphi(x) \to +\infty$
  as $x \to$ feasible set boundary
- $\varphi(x) \to -\infty$ as $x \to$ optimal

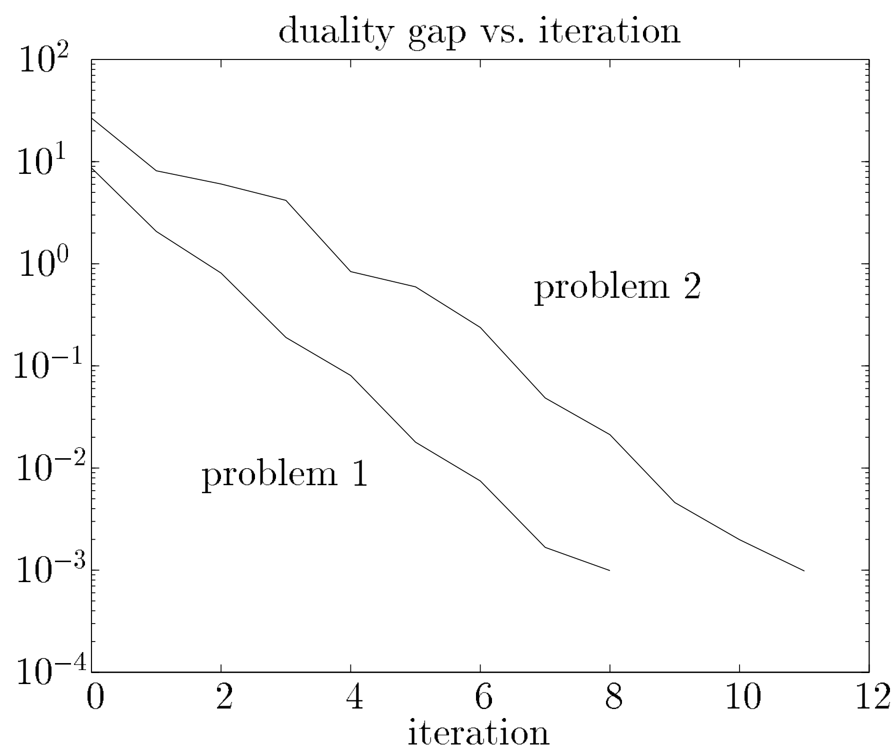
Minimize $\varphi$ by (modified) Newton method

**If $\varphi$ is properly chosen**:
algorithm is polynomial, efficient in practice

# Typical example: matrix norm minimization

$$\text{minimize} \;\; \|A_0 + x_1 A_1 + \cdots + x_k A_k\|$$

two specific problems: 5 matrices, $5 \times 5$; 50 matrices, $50 \times 50$



**Cost per iteration:**
computing Newton direction, a **least squares problem** with same structure as original problem (Toeplitz, etc.)

**Hence:**
cost of solving **convex problem**
$\approx 10 \times$ cost of solving similar **least-squares** problem

**Hence:**
can solve **least-squares problem** efficiently
$\Longrightarrow$ can solve **convex problem** efficiently

# Exploiting problem structure via CG

Conjugate Gradients: solve $\min_x \|Ax - b\|$, $x \in \mathbf{R}^m$ via $m$ evaluations of $x \to Ax$ and $y \to A^T y$

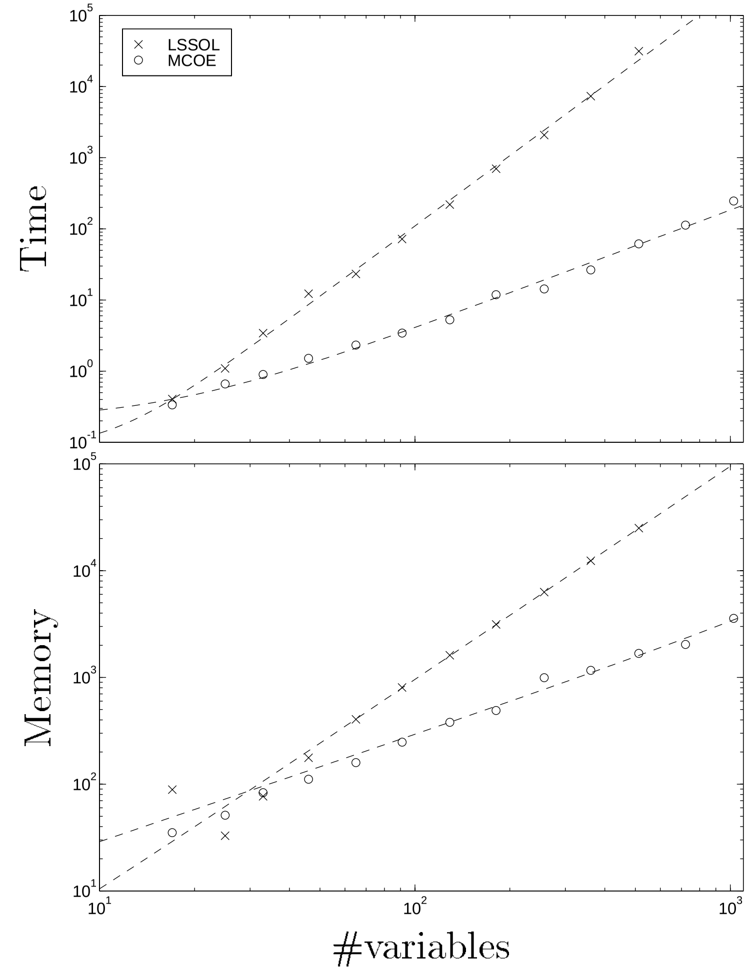
- roughly: can evaluate response and adjoint fast
  $\Longrightarrow$ can solve least-squares problem fast
  ($\Longrightarrow$ can solve convex problem fast)
- don't need exact solution for interior-point methods (allows early termination)
- preconditioning (problem specific)

Examples:

- FIR filter: fast ($N \log N$) convolution
- Input design: system state, co-state simulation

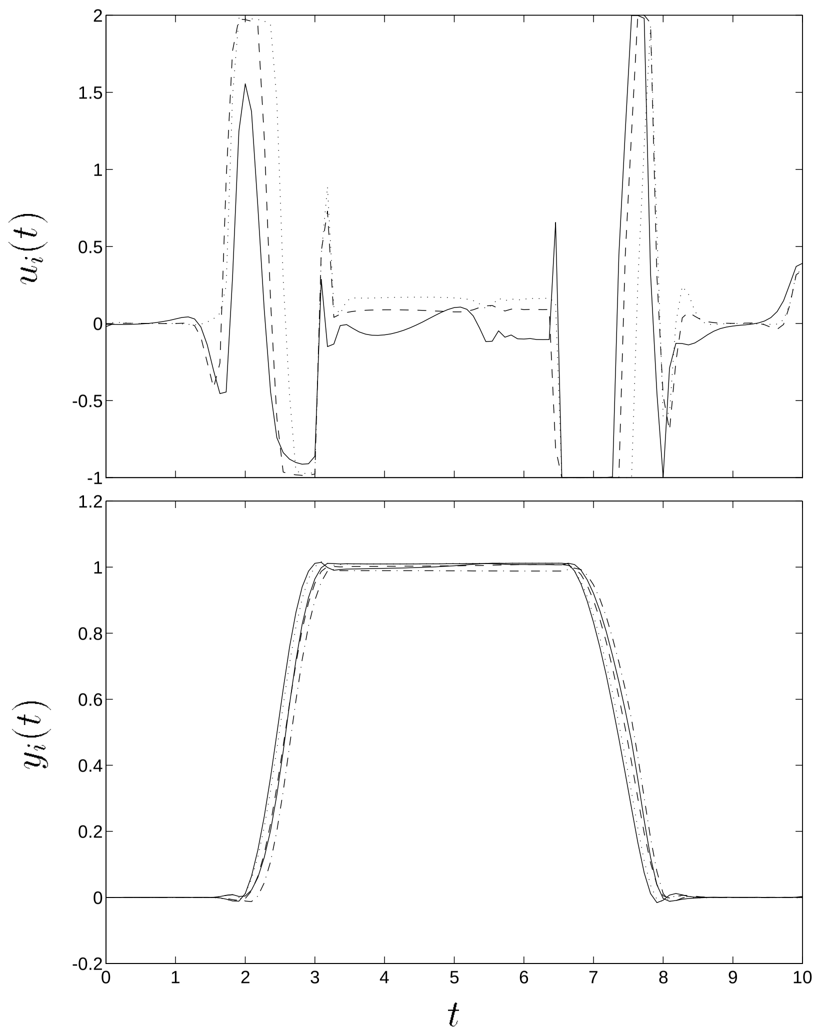
# FIR filter design example (M. Grant)

- forward, adjoint operator: FFT
- #taps $\approx 2 \cdot$ #variables
- #constraints $\approx 10 \cdot$ #variables
- $>1000$ variables, $>10000$ constraints solved in 4 min, 4Mb



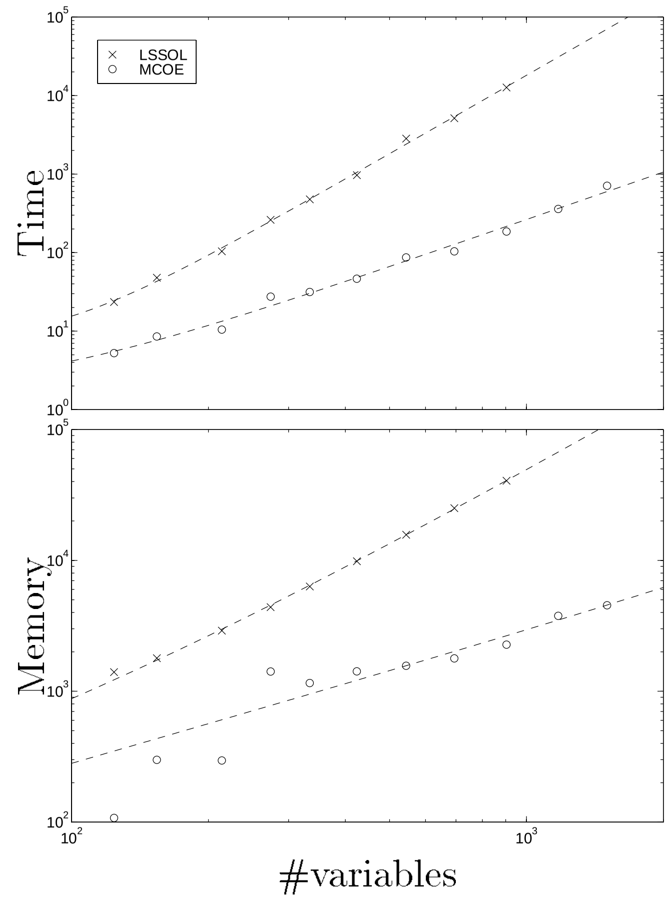
# MIMO input design example (M. Grant)

- 3 inputs, 8 outputs, 8 states
- amplitude limits on inputs
- slew limits on 3 outputs
- minimize peak tracking error on 5 outputs



# MIMO input design example (M. Grant)

- forward, adjoint operator: state, co-state simulation
- #vbles $= 3 \cdot$ #time steps
- #constr $\approx 7 \cdot$ #vbles
- $>$ 1500 variables,
  $>$ 10000 constraints
  solved in 12 min, 5Mb



# Example: multiple Lyapunov inequalities

(L. Vandenberghe)

**Problem:** minimize linear fct of matrix $P \in \mathbf{R}^{n\times n}$ subject to:

$$A_i^T P + P A_i + Q_i \leq 0, \qquad i = 1, \ldots, L$$

$A_i$, $Q_i$ given; $P$ is the variable

- number of variables: $m \triangleq n(n+1)/2$
- cost of solving similar least-squares problem, not exploiting problem structure: $O(Lm^3)$
- cost of solving problem with primal-dual method, exploiting problem structure: $O(L^{1.2}m^2)$
- problems with $> 1000$ variables, $> 10000$ constraints solved on small workstation in few minutes

# Exploiting structure in convex problems

can evaluate response, adjoint fast *(exploiting structure)*

$\Downarrow$

can solve least-squares problem fast *(using conj grad)*

$\Downarrow$

can solve convex problem fast *(using int-pt methods)*

# Main point

- Many problems arising in engineering analysis and design can be cast as **convex optimization problems**
- Hence, can be efficiently solved by **interior-point methods** that **exploit problem structure**

- As available computing power increases, this observation becomes more relevant
- convex problems not widely enough recognized

# (A few) references


- Nesterov and Nemirovsky, *Interior-point polynomial algorithms in convex programming*, SIAM, 1994.
- Vandenberghe and Boyd, *Positive Definite Programming,* submitted, SIAM Review, avail. anon. ftp.
- Boyd and Barratt, *Linear controller design: Limits of performance,* Prentice-Hall, 1991.
- Boyd, El Ghaoui, Feron and Balakrishnan, *Linear matrix inequalities in system and control theory,* SIAM, 1994.
- Ben Tal and Nemirovskii, *Interior point polynomial-time method for truss topology design*, Technion report, 1992.
- Vandenberghe and Boyd, *Primal-dual potential reduction method for problems involving matrix inequalities,* Math. Programming 1994.
- Sapatnekar, *A Convex Programming Approach to Problems in VLSI Design*, PhD thesis, Univ. of Illinois, 1992.

**. . . the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity.**

— R. Rockafellar, SIAM Review 1993