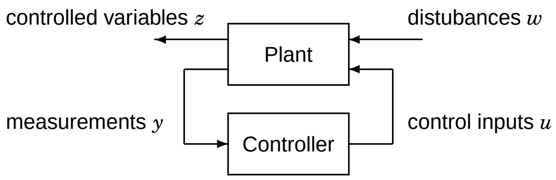
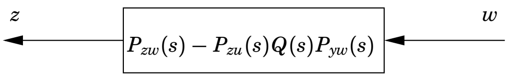
The Q-parametrization (Youla)



Idea for lecture 12-14:
The choice of controller generally corresponds to finding $Q(s)$, to get desirable properties of the map from w to z:



Once $Q(s)$ is determined, a corresponding controller is derived.

Lecture 13: Synthesis by Convex Optimization

- **Introduction to convex optimization**
- Controller optimization using Youla parametrization
- Example — DCservo revisited

Most of this lecture is based on source material from Boyd, Vandenberghe and coauthors. See
http://www.control.lth.se/Education/EngineeringProgram/FRTN10/multivariable-control.html

Least-squares

$$\text{minimize} \quad \|Ax - b\|_2^2$$

solving least-squares problems

- analytical solution: $x^\star = (A^TA)^{-1}A^Tb$
- reliable and efficient algorithms and software
- computation time proportional to n^2k ($A \in \mathbf{R}^{k\times n}$); less if structured
- a mature technology

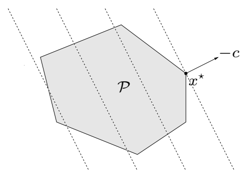
using least-squares

- least-squares problems are easy to recognize
- a few standard techniques increase flexibility ($e.g.$, including weights, adding regularization terms)

Linear program (LP)

$$\begin{array}{ll} \text{minimize} & c^Tx + d \\ \text{subject to} & Gx \preceq h \\ & Ax = b \end{array}$$

- convex problem with affine objective and constraint functions
- feasible set is a polyhedron



Linear programming

$$\begin{array}{ll} \text{minimize} & c^Tx \\ \text{subject to} & a_i^Tx \le b_i, \quad i = 1, \ldots, m \end{array}$$

solving linear programs

- no analytical formula for solution
- reliable and efficient algorithms and software
- computation time proportional to n^2m if $m \ge n$; less with structure
- a mature technology

using linear programming

- not as easy to recognize as least-squares problems
- a few standard tricks used to convert problems into linear programs ($e.g.$, problems involving ℓ_1- or ℓ_∞-norms, piecewise-linear functions)

Convex optimization problem

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

- objective and constraint functions are convex:

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

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

- includes least-squares problems and linear programs as special cases

solving convex optimization problems

- no analytical solution
- reliable and efficient algorithms
- computation time (roughly) proportional to $\max\{n^3, n^2m, F\}$, where F is cost of evaluating f_i's and their first and second derivatives
- almost a technology

using convex optimization

- often difficult to recognize
- many tricks for transforming problems into convex form
- surprisingly many problems can be solved via convex optimization

Brief history of convex optimization

theory (convex analysis): ca1900–1970

algorithms
- 1947: simplex algorithm for linear programming (Dantzig)
- 1960s: early interior-point methods (Fiacco & McCormick, Dikin, . . .)
- 1970s: ellipsoid method and other subgradient methods
- 1980s: polynomial-time interior-point methods for linear programming (Karmarkar 1984)
- late 1980s–now: polynomial-time interior-point methods for nonlinear convex optimization (Nesterov & Nemirovski 1994)

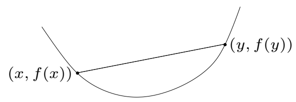
applications
- before 1990: mostly in operations research; few in engineering
- since 1990: many new applications in engineering (control, signal processing, communications, circuit design, . . .); new problem classes (semidefinite and second-order cone programming, robust optimization)

Definition

$f : \mathbf{R}^n \to \mathbf{R}$ is convex if $\mathbf{dom}\, f$ is a convex set and

$$f(\theta x + (1-\theta)y) \le \theta f(x) + (1-\theta) f(y)$$

for all $x, y \in \mathbf{dom}\, f$, $0 \le \theta \le 1$



- f is concave if $-f$ is convex
- f is strictly convex if $\mathbf{dom}\, f$ is convex and

$$f(\theta x + (1-\theta)y) < \theta f(x) + (1-\theta) f(y)$$

for $x, y \in \mathbf{dom}\, f$, $x \ne y$, $0 < \theta < 1$

Examples on R

convex:

- affine: $ax + b$ on $\mathbf{R}$, for any $a, b \in \mathbf{R}$
- exponential: e^{ax}, for any $a \in \mathbf{R}$
- powers: x^α on $\mathbf{R}_{++}$, for $\alpha \ge 1$ or $\alpha \le 0$
- powers of absolute value: $|x|^p$ on $\mathbf{R}$, for $p \ge 1$
- negative entropy: $x \log x$ on $\mathbf{R}_{++}$

concave:

- affine: $ax + b$ on $\mathbf{R}$, for any $a, b \in \mathbf{R}$
- powers: x^α on $\mathbf{R}_{++}$, for $0 \le \alpha \le 1$
- logarithm: $\log x$ on $\mathbf{R}_{++}$

Examples on $\mathbf{R}^n$ and $\mathbf{R}^{m\times n}$

affine functions are convex and concave; all norms are convex

examples on $\mathbf{R}^n$

- affine function $f(x) = a^T x + b$
- norms: $\|x\|_p = (\sum_{i=1}^n |x_i|^p)^{1/p}$ for $p \ge 1$; $\|x\|_\infty = \max_k |x_k|$

examples on $\mathbf{R}^{m\times n}$ ($m \times n$ matrices)

- affine function

$$f(X) = \mathbf{tr}(A^T X) + b = \sum_{i=1}^m \sum_{j=1}^n A_{ij} X_{ij} + b$$

- spectral (maximum singular value) norm

$$f(X) = \|X\|_2 = \sigma_{\max}(X) = (\lambda_{\max}(X^T X))^{1/2}$$

Convex optimization problem

standard form convex optimization problem

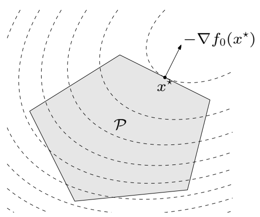
$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \le 0, \quad i = 1, \ldots, m \\ & a_i^T x = b_i, \quad i = 1, \ldots, p \end{array}$$

- $f_0, f_1, \ldots, f_m$ are convex; equality constraints are affine
- problem is *quasiconvex* if f_0 is quasiconvex (and $f_1, \ldots, f_m$ convex)

Quadratic program (QP)

$$\begin{array}{ll} \text{minimize} & (1/2)x^T P x + q^T x + r \\ \text{subject to} & Gx \preceq h \\ & Ax = b \end{array}$$

- $P \in \mathbf{S}^n_+$, so objective is convex quadratic
- minimize a convex quadratic function over a polyhedron



Second-order cone programming

$$\begin{array}{ll} \text{minimize} & f^T x \\ \text{subject to} & \|A_i x + b_i\|_2 \le c_i^T x + d_i, \quad i = 1, \ldots, m \\ & Fx = g \end{array}$$

$(A_i \in \mathbf{R}^{n_i \times n}, F \in \mathbf{R}^{p \times n})$

Semidefinite program (SDP)

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & x_1 F_1 + x_2 F_2 + \cdots + x_n F_n + G \preceq 0 \\ & Ax = b \end{array}$$

with $F_i, G \in \mathbf{S}^k$

- inequality constraint is called linear matrix inequality (LMI)

Matrix norm minimization

$$\text{minimize} \quad \|A(x)\|_2 = \left(\lambda_{\max}(A(x)^T A(x))\right)^{1/2}$$

where $A(x) = A_0 + x_1 A_1 + \cdots + x_n A_n$ (with given $A_i \in \mathbf{S}^{p\times q}$)

equivalent SDP

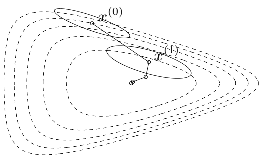
$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \begin{bmatrix} tI & A(x) \\ A(x)^T & tI \end{bmatrix} \succeq 0 \end{array}$$

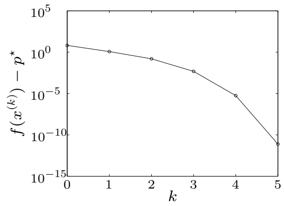
- variables $x \in \mathbf{R}^n$, $t \in \mathbf{R}$
- constraint follows from

$$\begin{aligned} \|A\|_2 \le t \quad &\Longleftrightarrow \quad A^T A \preceq t^2 I, \quad t \ge 0 \\ &\Longleftrightarrow \quad \begin{bmatrix} tI & A \\ A^T & tI \end{bmatrix} \succeq 0 \end{aligned}$$

Newton's method

given a starting point $x \in \mathbf{dom}\, f$, tolerance $\epsilon > 0$.
repeat
1. *Compute the Newton step and decrement.*
   $\Delta x_{\mathrm{nt}} := -\nabla^2 f(x)^{-1}\nabla f(x)$; $\lambda^2 := \nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x)$.
2. *Stopping criterion.* **quit** if $\lambda^2/2 \le \epsilon$.
3. *Line search.* Choose step size t by backtracking line search.
4. *Update.* $x := x + t\Delta x_{\mathrm{nt}}$.





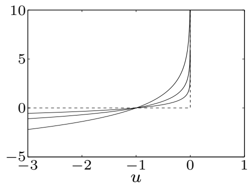
Barrier method for constrained minimization

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \le 0 \quad 1 = 1, \dots, m \\ & Ax = b \end{array}$$

approximation via logarithmic barrier

$$\begin{array}{ll} \text{minimize} & f_0(x) - (1/t)\sum_{i=1}^m \log(-f_i(x)) \\ \text{subject to} & Ax = b \end{array}$$

- an equality constrained problem
- for $t > 0$, $-(1/t)\log(-u)$ is a smooth approximation of I_-
- approximation improves as $t \to \infty$



Outline

- Introduction to convex optimization
- **Controller optimization using Youla parametrization**
- Example — DCservo revisited

Scheme for numerical optimization of Q

Given some fixed set of basis function $\phi_0(s), \dots, \phi_N(s)$, we will search numerically for matrices $Q_0, \dots, Q_N$ such that the closed loop transfer matrix $G_{zw}(s)$ satisfies given specifications when

$$G_{zw}(s) = P_{zw}(s) - P_{zu}(s)Q(s)P_{yw}(s) \qquad \text{and } Q(s) = \sum_{k=0}^{N} Q_k\phi_k(s)$$

Once $Q(s)$ has been determined, we will recover the desired controller from the formula

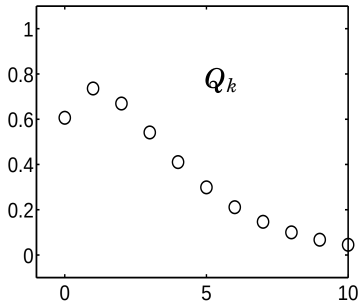
$$C(s) = \left[I - Q(s)P_{yu}(s)\right]^{-1}Q(s)$$

It is possible to choose the sequence $\phi_0(s), \phi_1(s), \phi_2(s), \dots$ such that every stable Q can be approximated arbitrarily well. Hence, in principle, every convex control design problem can be solved this way.

But, what specifications give a convex design problem?

Pulse response parameterization

We will use an intuitively simple parametrization of $Q(s)$ where each parameter Q_k represents a point on the corresponding impulse response in time domain.

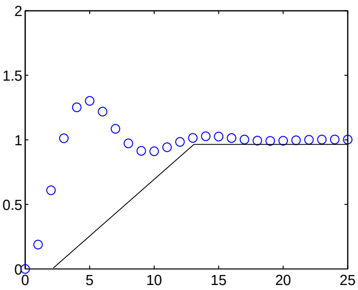


Mini-problem

Which specifications are convex constraints on Q_k?

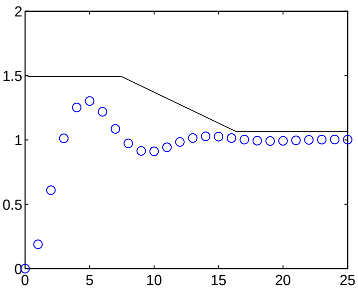
1. Stability of the closed loop system
2. Lower bound on step response from w_i to z_j at time t_i
3. Upper bound on step response from w_i to z_j at time t_i
4. Lower bound on Bode amplitude from w_i to z_j at frequency ω_i
5. Upper bound on Bode amplitude from w_i to z_j at frequency ω_i
6. Interval bound on Bode phase from w_i to z_j at frequency ω_i

Lower bound on step response



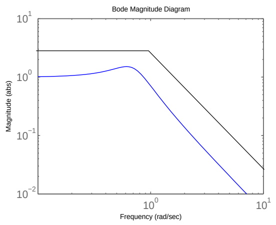
The step response depends linearly on Q_k, so every time t_k with a lower bound gives a linear constraint.

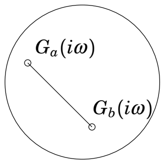
Upper bound on step response



Every time t_k with an upper bound also gives a linear constraint.

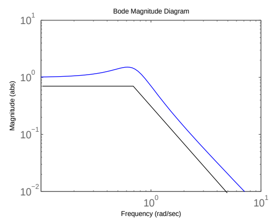
Upper bound on Bode amplitude

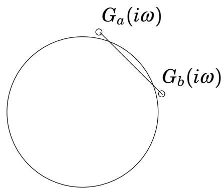




An amplitude bound $|G(i\omega_i)| < c$ is a quadratic constraint.

Lower bound on Bode amplitude





An lower bound $|G(i\omega_i)|$ is a *non-convex* quadratic constraint. This should be avoided in optimization.

Synthesis by convex optimization

A general control synthesis problem can be stated as a convex optimization problem in the variables $Q_0, \dots, Q_m$. The problem has a quadratic objective, with linear and quadratic constraints:

Minimize $\int_{-\infty}^{\infty} |P_{zw}(i\omega) + P_{zu}(i\omega) \overbrace{\sum_k Q_k \phi_k(i\omega)}^{Q(i\omega)} P_{yw}(i\omega)|^2 d\omega$ } quadratc objective

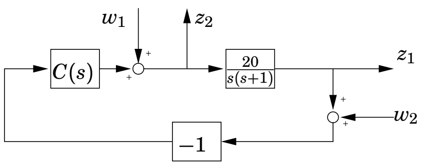
subject to
- step response $w_i \to z_j$ is smaller than f_{ijk} at time t_k
- step response $w_i \to z_j$ is bigger than g_{ijk} at time t_k } linear constraints
- Bode magnitude $w_i \to z_j$ is smaller than h_{ijk} at ω_k } quadratic constraints

Once the variables $Q_0, \dots, Q_m$ have been optimized, the controller is obtained as $C(s) = \left[I - Q(s)P_{yu}(s)\right]^{-1} Q(s)$

Outline

- ∘ Introduction to convex optimization
- ∘ Controller optimization using Youla parametrization
- • **Example — DCservo revisited**

Example — DC-motor



The transfer matrix from (w_1, w_2) to (z_1, z_2) is

$$G_{zw}(s) = \begin{bmatrix} \frac{P}{1+PC} & \frac{-PC}{1+PC} \\ \frac{1}{1+PC} & \frac{-C}{1+PC} \end{bmatrix}$$

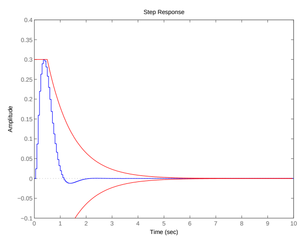
with $P(s) = \frac{20}{s(s+1)}$. We will choose $C(s)$ to minimize

$$\text{trace} \int_{-\infty}^{\infty} G_{zw}(i\omega) G_{zw}(i\omega)^* d\omega$$

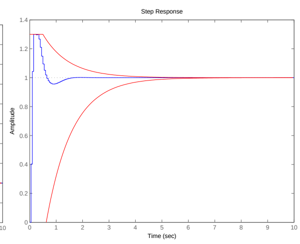
subject time-domain bounds.

DC-servo with time domain bounds

Input step disturbance



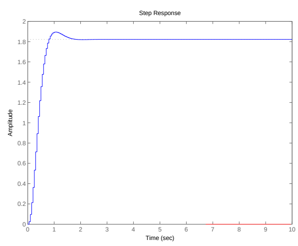
Reference step



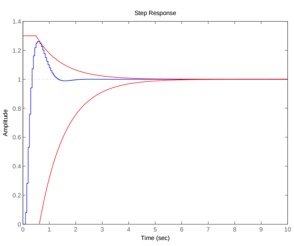
What if we remove the upper bound on the response to input disturbances ?

DC-servo with time domain bounds

Input step disturbance



Reference step



The integral action in the controller is lost, just as in lecture 11!

Summary

- There are efficient algorithms for convex optimization, e.g.
  - Linear programming (LP)
  - Quadratic programming (QP)
  - Second order cone programming (SOCP)
  - Semi-definite programming (SDP)
- The Youla parametrization allows us to use these algorithms for control synthesis
- Resulting controllers have high order. Order reduction will be studies in the last lecture before course summary.