11. Constrained least squares

- least norm problem
- least squares with equality constraints
- linear quadratic control
Least norm problem

\[
\begin{align*}
\text{minimize} & \quad \| x \|^2 \\
\text{subject to} & \quad Cx = d
\end{align*}
\]

• \( C \) is a \( p \times n \) matrix, \( d \) is a \( p \)-vector
• in most applications \( p < n \) and the equation \( Cx = d \) is underdetermined
• the goal is to find the solution of the equation \( Cx = d \) with the smallest norm

we will assume that \( C \) has linearly independent rows

• the equation \( Cx = d \) has at least one solution for every \( d \)
• \( C \) is wide or square \(( p \leq n \))
• if \( p < n \) there are infinitely many solutions
• unit mass, with zero initial position and velocity
• piecewise-constant force $F(t) = x_j$ for $t \in [j - 1, j)$ for $j = 1, \ldots, 10$
• position and velocity at $t = 10$ are given by $y = Cx$ where

$$C = \begin{bmatrix}
\frac{19}{2} & \frac{17}{2} & \frac{15}{2} & \cdots & \frac{1}{2} \\
1 & 1 & 1 & \cdots & 1
\end{bmatrix}$$
Example

forces that move mass over a unit distance with zero final velocity satisfy

\[
\begin{bmatrix}
19/2 & 17/2 & 15/2 & \cdots & 1/2 \\
1 & 1 & 1 & \cdots & 1
\end{bmatrix}
x = \begin{bmatrix}
1 \\
0
\end{bmatrix}
\]

some interesting solutions:

- solutions with only two nonzero elements:

  \[x = (1, -1, 0, \ldots, 0), \quad x = (0, 1, -1, \ldots, 0), \quad \ldots\]

- least norm solution: minimizes

  \[
  \int_0^{10} F(t)^2 dt = x_1^2 + x_2^2 + \cdots + x_{10}^2
  \]
Example

\[ x = (1, -1, 0, \ldots, 0) \]
Least distance solution

as a variation, we can minimize the distance to a given point \( a \neq 0 \):

\[
\begin{align*}
\text{minimize} & \quad \|x - a\|^2 \\
\text{subject to} & \quad Cx = d
\end{align*}
\]

• reduces to least norm problem by a change of variables \( y = x - a \)

\[
\begin{align*}
\text{minimize} & \quad \|y\|^2 \\
\text{subject to} & \quad Cy = d - Ca
\end{align*}
\]

• from least norm solution \( y \), we obtain solution \( x = y + a \) of first problem
if $C$ has linearly independent rows (is right-invertible), then

$$\hat{x} = C^T (CC^T)^{-1} d$$

$$= C^\dagger d$$

is the unique solution of the least norm problem

$$\text{minimize} \quad \|x\|^2$$
$$\text{subject to} \quad Cx = d$$

• in other words if $Cx = d$ and $x \neq \hat{x}$, then $\|x\| > \|\hat{x}\|$  
• recall from page 4.25 that

$$C^T (CC^T)^{-1} = C^\dagger$$

is the pseudo-inverse of a right-invertible matrix $C$
Proof

1. we first verify that \( \hat{x} \) satisfies the equation:

\[
C\hat{x} = CC^T (CC^T)^{-1} d = d
\]

2. next we show that \( ||x|| > ||\hat{x}|| \) if \( Cx = d \) and \( x \neq \hat{x} \)

\[
||x||^2 = ||\hat{x} + x - \hat{x}||^2
= ||\hat{x}||^2 + 2\hat{x}^T (x - \hat{x}) + ||x - \hat{x}||^2
= ||\hat{x}||^2 + ||x - \hat{x}||^2
\geq ||\hat{x}||^2
\]

with equality only if \( x = \hat{x} \)

on line 3 we use \( Cx = C\hat{x} = d \) in

\[
\hat{x}^T (x - \hat{x}) = d^T (CC^T)^{-1} C(x - \hat{x}) = 0
\]
QR factorization method

use the QR factorization $C^T = QR$ of the matrix $C^T$:

\[
\hat{x} = C^T(C^T)^{-1}d \\
  = QR(R^TQ^TQR)^{-1}d \\
  = QR(R^TR)^{-1}d \\
  = QR^{-T}d
\]

Algorithm

1. compute QR factorization $C^T = QR$ (2$p^2n$ flops)
2. solve $R^Tz = d$ by forward substitution ($p^2$ flops)
3. matrix-vector product $\hat{x} = Qz$ (2$pn$ flops)

complexity: 2$p^2n$ flops
Example

\[ C = \begin{bmatrix} 1 & -1 & 1 & 1 \\ 1 & 0 & 1/2 & 1/2 \end{bmatrix}, \quad d = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \]

- QR factorization \( C^T = QR \)

\[
\begin{bmatrix} 1 & 1 \\ -1 & 0 \\ 1 & 1/2 \\ 1 & 1/2 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/\sqrt{2} \\ -1/2 & 1/\sqrt{2} \\ 1/2 & 0 \\ 1/2 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} \]

- solve \( R^Tz = b \)

\[
\begin{bmatrix} 2 & 0 \\ 1 & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \]

\( z_1 = 0, \, z_2 = \sqrt{2} \)

- evaluate \( \hat{x} = Qz = (1, 1, 0, 0) \)
Outline

- least norm problem
- least squares with equality constraints
- linear quadratic control
Constrained least squares

minimize \( \|Ax - b\|^2 \)
subject to \( Cx = d \)

- \( A \) is an \( m \times n \) matrix, \( C \) is a \( p \times n \) matrix, \( b \) is an \( m \)-vector, \( d \) is a \( p \)-vector
- in most applications \( p < n \), so equations are underdetermined
- the goal is to find the solution of \( Cx = d \) with smallest value of \( \|Ax - b\|^2 \)
- we make no assumptions about the shape of \( A \)

Special cases

- least squares problem is a special case with \( p = 0 \) (no constraints)
- least norm problem is a special case with \( A = I \) and \( b = 0 \)
Piecewise-polynomial fitting

- fit two polynomials $f(x)$, $g(x)$ to points $(x_1, y_1), \ldots, (x_N, y_N)$

\[
f(x_i) \approx y_i \quad \text{for points } x_i \leq a, \quad g(x_i) \approx y_i \quad \text{for points } x_i > a
\]

- make values and derivatives continuous at point $a$: $f(a) = g(a), f'(a) = g'(a)$
Constrained least squares formulation

- assume points are numbered so that \( x_1, \ldots, x_M \leq a \) and \( x_{M+1}, \ldots, x_N > a \):

\[
\text{minimize } \sum_{i=1}^{M} (f(x_i) - y_i)^2 + \sum_{i=M+1}^{N} (g(x_i) - y_i)^2
\]

subject to \( f(a) = g(a), \ f'(a) = g'(a) \)

- for polynomials \( f(x) = \theta_1 + \cdots + \theta_d x^{d-1} \) and \( g(x) = \theta_{d+1} + \cdots + \theta_{2d} x^{d-1} \)

\[
A = \begin{bmatrix}
1 & x_1 & \cdots & x_1^{d-1} & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots \\
1 & x_M & \cdots & x_M^{d-1} & 0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & 1 & x_{M+1} & \cdots & x_{M+1}^{d-1} \\
\vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots \\
0 & 0 & \cdots & 0 & 1 & x_N & \cdots & x_N^{d-1}
\end{bmatrix}, \quad b = \begin{bmatrix} y_1 \\
\vdots \\
y_M \\
y_{M+1} \\
\vdots \\
y_N \end{bmatrix}
\]

\[
C = \begin{bmatrix}
1 & a & \cdots & a^{d-1} & -1 & -a & \cdots & -a^{d-1} \\
0 & 1 & \cdots & (d-1)a^{d-2} & 0 & -1 & \cdots & -(d-1)a^{d-2}
\end{bmatrix}, \quad d = \begin{bmatrix} 0 \\
0 \end{bmatrix}
\]
Assumptions

\[
\begin{align*}
\text{minimize} & \quad \|Ax - b\|^2 \\
\text{subject to} & \quad Cx = d
\end{align*}
\]

we will make two assumptions:

1. the stacked \((m + p) \times n\) matrix
   \[
   \begin{bmatrix}
   A \\
   C
   \end{bmatrix}
   \]
   has linearly independent columns (is left-invertible)

2. \(C\) has linearly independent rows (is right-invertible)

- note that assumption 1 is a weaker condition than left invertibility of \(A\)
- assumptions imply that \(p \leq n \leq m + p\)
Optimality conditions

\( \hat{x} \) solves the constrained LS problem if and only if there exists a \( z \) such that

\[
\begin{bmatrix}
A^T A & C^T \\
C & 0
\end{bmatrix}
\begin{bmatrix}
\hat{x} \\
z
\end{bmatrix}
=
\begin{bmatrix}
A^T b \\
d
\end{bmatrix}
\]

(proof on next page)

- this is a set of \( n + p \) linear equations in \( n + p \) variables
- we’ll see that the matrix on the left-hand side is nonsingular
- equations are also known as Karush–Kuhn–Tucker (KKT) equations

Special cases

- least squares: when \( p = 0 \), reduces to normal equations \( A^T A \hat{x} = A^T b \)
- least norm: when \( A = I, b = 0 \), reduces to \( C \hat{x} = d \) and \( \hat{x} + C^T z = 0 \)
Proof

Suppose \( x \) satisfies \( Cx = d \), and \((\hat{x}, z)\) satisfies the equation on page 11.15

\[
\|Ax - b\|^2 = \|A(x - \hat{x}) + A\hat{x} - b\|^2
\]

\[
= \|A(x - \hat{x})\|^2 + \|A\hat{x} - b\|^2 + 2(x - \hat{x})^T A^T (A\hat{x} - b)
\]

\[
= \|A(x - \hat{x})\|^2 + \|A\hat{x} - b\|^2 - 2(x - \hat{x})^T C^T z
\]

\[
= \|A(x - \hat{x})\|^2 + \|A\hat{x} - b\|^2
\]

\[
\geq \|A\hat{x} - b\|^2
\]

- On line 3 we use \( A^T A\hat{x} + C^T z = A^T b \); on line 4, \( Cx = C\hat{x} = d \)
- Inequality shows that \( \hat{x} \) is optimal
- \( \hat{x} \) is the unique optimum because equality holds only if

\[
A(x - \hat{x}) = 0, \quad C(x - \hat{x}) = 0 \quad \implies \quad x = \hat{x}
\]

By the first assumption on page 11.14
Nonsingularity

if the two assumptions hold, then the matrix

\[
\begin{bmatrix}
A^T A & C^T \\
C & 0
\end{bmatrix}
\]

is nonsingular

Proof.

\[
\begin{bmatrix}
A^T A & C^T \\
C & 0
\end{bmatrix} \begin{bmatrix}
x \\
z
\end{bmatrix} = 0 \implies x^T (A^T Ax + C^T z) = 0, \quad Cx = 0
\]

\[
\implies \|Ax\|^2 = 0, \quad Cx = 0
\]

\[
\implies Ax = 0, \quad Cx = 0
\]

\[
\implies x = 0 \quad \text{by assumption 1}
\]

if \(x = 0\), we have \(C^T z = -A^T Ax = 0\); hence also \(z = 0\) by assumption 2
Nonsingularity

if the assumptions do not hold, then the matrix

\[
\begin{bmatrix}
A^T A & C^T \\
C & 0
\end{bmatrix}
\]

is singular

• if assumption 1 does not hold, there exists \( x \neq 0 \) with \( Ax = 0, Cx = 0 \); then

\[
\begin{bmatrix}
A^T A & C^T \\
C & 0
\end{bmatrix}
\begin{bmatrix}
x \\
0
\end{bmatrix} = 0
\]

• if assumption 2 does not hold there exists a \( z \neq 0 \) with \( C^T z = 0 \); then

\[
\begin{bmatrix}
A^T A & C^T \\
C & 0
\end{bmatrix}
\begin{bmatrix}
0 \\
z
\end{bmatrix} = 0
\]

in both cases, this shows that the matrix is singular
Solution by LU factorization

\[
\begin{bmatrix}
A^T A & C^T \\
C & 0
\end{bmatrix}
\begin{bmatrix}
x \\
z
\end{bmatrix} =
\begin{bmatrix}
A^T b \\
d
\end{bmatrix}
\]

Algorithm

1. compute \( H = A^T A \) \((mn^2\) flops)
2. compute \( c = A^T b \) \((2mn\) flops)
3. solve the linear equation

\[
\begin{bmatrix}
H & C^T \\
C & 0
\end{bmatrix}
\begin{bmatrix}
x \\
z
\end{bmatrix} =
\begin{bmatrix}
c \\
d
\end{bmatrix}
\]

by the LU factorization \(((2/3)(p+n)^3\) flops)

complexity: \( mn^2 + (2/3)(p+n)^3 \) flops
Solution by QR factorization

we derive one of several possible methods based on the QR factorization

\[
\begin{bmatrix}
A^T A & C^T \\
C & 0
\end{bmatrix}
\begin{bmatrix}
\hat{x} \\
z
\end{bmatrix}
= 
\begin{bmatrix}
A^T b \\
d
\end{bmatrix}
\]

- if we make a change of variables \( w = z - d \), the equation becomes

\[
\begin{bmatrix}
A^T A + C^T C & C^T \\
C & 0
\end{bmatrix}
\begin{bmatrix}
\hat{x} \\
w
\end{bmatrix}
= 
\begin{bmatrix}
A^T b \\
d
\end{bmatrix}
\]

- assumption 1 guarantees that \( A^T A + C^T C \) is nonsingular (see page 4.21)

- assumption 1 guarantees that the following QR factorization exists:

\[
\begin{bmatrix}
A \\
C
\end{bmatrix}
= QR = 
\begin{bmatrix}
Q_1 \\
Q_2
\end{bmatrix}
R
\]
Solution by QR factorization

substituting the QR factorization gives the equation

\[
\begin{bmatrix}
R^T R & R^T Q_2^T \\
Q_2 R & 0
\end{bmatrix}
\begin{bmatrix}
\hat{x} \\
w
\end{bmatrix}
= 
\begin{bmatrix}
R^T Q_1^T b \\
d
\end{bmatrix}
\]

- multiply first equation with \(R^{-T}\) and make change of variables \(y = R\hat{x}\):

\[
\begin{bmatrix}
I & Q_2^T \\
Q_2 & 0
\end{bmatrix}
\begin{bmatrix}
y \\
w
\end{bmatrix}
= 
\begin{bmatrix}
Q_1^T b \\
d
\end{bmatrix}
\]

- next we note that the matrix \(Q_2 = CR^{-1}\) has linearly independent rows:

\[
Q_2^T u = R^{-T} C^T u = 0 \implies C^T u = 0 \implies u = 0
\]

because \(C\) has linearly independent rows (assumption 2)
Solution by QR factorization

we use the QR factorization of $Q^T_2$ to solve

$$\begin{bmatrix}
    I & Q^T_2 \\
    Q_2 & 0
\end{bmatrix}
\begin{bmatrix}
    y \\
    w
\end{bmatrix} =
\begin{bmatrix}
    Q^T_1 b \\
    d
\end{bmatrix}$$

• from the 1st block row, $y = Q^T_1 b - Q^T_2 w$; substitute this in the 2nd row:

$$Q_2 Q^T_2 w = Q_2 Q^T_1 b - d$$

• we solve this equation for $w$ using the QR factorization $Q^T_2 = \tilde{Q} \tilde{R}$:

$$\tilde{R}^T \tilde{R} w = \tilde{R}^T \tilde{Q}^T Q^T_1 b - d$$

which can be simplified to

$$\tilde{R} w = \tilde{Q}^T Q^T_1 b - \tilde{R}^{-T} d$$
Summary of QR factorization method

\[
\begin{bmatrix}
A^T A + C^T C & C^T \\
C & 0
\end{bmatrix}
\begin{bmatrix}
\hat{x} \\
w
\end{bmatrix}
= 
\begin{bmatrix}
A^T b \\
d
\end{bmatrix}
\]

Algorithm

1. compute the two QR factorizations

\[
\begin{bmatrix}
A \\
C
\end{bmatrix} = \begin{bmatrix}
Q_1 \\
Q_2
\end{bmatrix} R, \quad Q_2^T = \tilde{Q} \tilde{R}
\]

2. solve \( \tilde{R}^T u = d \) by forward substitution and compute \( c = \tilde{Q}^T Q_1^T b - u \)

3. solve \( \tilde{R} w = c \) by back substitution and compute \( y = Q_1^T b - Q_2^T w \)

4. compute \( R \hat{x} = y \) by back substitution

complexity: \( 2(p + m)n^2 + 2np^2 \) flops for the QR factorizations
Comparison of the two methods

**Complexity:** roughly the same

- **LU factorization**
  \[ mn^2 + \frac{2}{3}(p + n)^3 \leq mn^2 + \frac{16}{3}n^3 \text{ flops} \]

- **QR factorization**
  \[ 2(p + m)n^2 + 2np^2 \leq 2mn^2 + 4n^3 \text{ flops} \]

Upper bounds follow from \( p \leq n \) (assumption 2)

**Stability:** 2nd method avoids calculation of Gram matrix \( A^T A \)
Outline

- least norm problem
- least squares with equality constraints
- linear quadratic control
Linear quadratic control

Linear dynamical system

\[ x_{t+1} = A_t x_t + B_t u_t, \quad y_t = C_t x_t, \quad t = 1, 2, \ldots \]

- \( n \)-vector \( x_t \) is system state at time \( t \)
- \( m \)-vector \( u_t \) is system input
- \( p \)-vector \( y_t \) is system output
- \( x_t, u_t, y_t \) often represent deviations from a standard operating condition

Objective: choose inputs \( u_1, \ldots, u_{T-1} \) that minimizes \( J_{\text{output}} + \rho J_{\text{input}} \) with

\[ J_{\text{output}} = \|y_1\|^2 + \cdots + \|y_T\|^2, \quad J_{\text{input}} = \|u_1\|^2 + \cdots + \|u_{T-1}\|^2 \]

State constraints: initial state and (possibly) the final state are specified

\[ x_1 = x^{\text{init}}, \quad x_T = x^{\text{des}} \]
Linear quadratic control problem

minimize  \[ \|C_1 x_1\|^2 + \cdots + \|C_T x_T\|^2 + \rho (\|u_1\|^2 + \cdots + \|u_{T-1}\|^2) \]
subject to  \[ x_{t+1} = A_t x_t + B_t u_t, \quad t = 1, \ldots, T - 1 \]
\[ x_1 = x^{\text{init}}, \quad x_T = x^{\text{des}} \]

variables: \( x_1, \ldots, x_T, u_1, \ldots, u_{T-1} \)

Constrained least squares formulation

minimize  \[ \|\tilde{A} z - \tilde{b}\|^2 \]
subject to  \[ \tilde{C} z = \tilde{d} \]

variables: the \( (nT + m(T - 1))\)-vector

\[ z = (x_1, \ldots, x_T, u_1, \ldots, u_{T-1}) \]
Linear quadratic control problem

Objective function: \( \|\tilde{A}z - \tilde{b}\|^2 \) with

\[
\tilde{A} = \begin{bmatrix} C_1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & C_T & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \sqrt{\rho}I & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & \sqrt{\rho}I \end{bmatrix}, \quad \tilde{b} = 0
\]

Constraints: \( \tilde{C}z = \tilde{d} \) with

\[
\tilde{C} = \begin{bmatrix} A_1 & -I & 0 & \cdots & 0 & 0 & B_1 & 0 & \cdots & 0 \\ 0 & A_2 & -I & \cdots & 0 & 0 & 0 & B_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & A_{T-1} & -I & 0 & 0 & \cdots & B_{T-1} \\ I & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & I & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad \tilde{d} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ x^{\text{init}} \\ x^{\text{des}} \end{bmatrix}
\]
Example

- a system with three states, one input, one output
- system is time-invariant (matrices $A_t = A$, $B_t = B$, and $C_t = C$ are constant)
- figure shows “open-loop” output $CA^{t-1}x^{\text{init}}$

\[ J_{\text{output}} + \rho J_{\text{input}} \] with final state constraint $x^{\text{des}} = 0$ at $T = 100$
Optimal trade-off curve

Constrained least squares
Three solutions on the trade-off curve

\[ u_t = \begin{cases} 0 & \text{if } \rho = 0.05 \\ 0 & \text{if } \rho = 0.2 \\ 0 & \text{if } \rho = 1 \end{cases} \]
Linear state feedback control

Linear state feedback

- linear state feedback control uses the input

\[ u_t = Kx_t, \quad t = 1, 2, \ldots \]

- \( K \) is the state feedback gain matrix

- widely used, especially when \( x_t \) should converge to zero, \( T \) is not specified

One possible choice for \( K \)

- solve the linear quadratic control problem with \( x^{\text{des}} = 0 \)
- solution \( u_t \) is a linear function of \( x^{\text{init}} \), hence \( u_1 \) can be written as \( u_1 = Kx^{\text{init}} \)
- columns of \( K \) can be found by computing \( u_1 \) for \( x^{\text{init}} = e_1, \ldots, e_n \)
- use this \( K \) as state feedback gain matrix
- system matrices of previous example
- blue curve uses optimal linear quadratic control for $T = 100$
- red curve uses simple linear state feedback $u_t = Kx_t$