

8. Least Squares

- The pseudo-inverse
- Example: pseudo-inverse
- Estimation and least-squares
- Effects of noise on estimation
- Example: navigation
- Regression or curve-fitting
- Example: fitting polynomials
- Example: rocket
- Control and minimum-norm problems
- Example: force on mass
- Matlab and the pseudo-inverse
- History of least-squares

The Key Points of This Section

estimation problems: given y_{meas} , find the *least-squares* solution x , that minimizes

$$\|y_{\text{meas}} - Ax\|$$

control problems: given y_{des} , find the *minimum-norm* x that satisfies

$$y_{\text{des}} = Ax$$

- the SVD gives a computational approach
- it also gives useful information even when important assumptions don't hold
 - estimation: usually need A skinny and full rank
 - control: usually need A fat and full rank
- it gives us quantitative information about the usefulness of the solutions

important facts

$$\text{null}(A^T) = \text{range}(A)^\perp$$

easy via the SVD:

because if the SVD of A is

$$A = U\Sigma V^T$$

then $\text{range}(A) = \text{span}\{u_1, \dots, u_r\}$

also the SVD of A^T is

$$A^T = V\Sigma^T U^T$$

so $\text{null}(A^T) = \text{span}\{u_{r+1}, \dots, u_n\}$

one more

$$\text{null}(A^T A) = \text{null}(A)$$

also easy via the SVD:

$$\begin{aligned} A^T A &= V \Sigma^T U^T U \Sigma V^T \\ &= V \Sigma^T \Sigma V^T \end{aligned}$$

which gives an SVD of $A^T A$.

$\Sigma^T \Sigma$ has the same number of non-zero elements as Σ , so both A and $A^T A$ have null space

$$\text{span}\{v_{r+1}, \dots, v_n\}$$

The Pseudo-Inverse

the thin SVD is $A = \hat{U}\hat{\Sigma}V^T$

$$A = \hat{U} \hat{\Sigma} V^T$$

here

- $\hat{\Sigma}$ is square, diagonal, positive definite
- \hat{U} and \hat{V} are skinny, orthonormal columns

the *pseudo-inverse* of A is

$$A^\dagger = \hat{V} \hat{\Sigma}^{-1} \hat{U}^T$$

it is computed using the SVD

example

rank 2 matrix:

$$A = \begin{bmatrix} -5 & -5 & -14 & -8 & 1 \\ -1 & 0 & 4 & 5 & 4 \\ 11 & 10 & 24 & 11 & -6 \end{bmatrix}$$

the full svd:

$$= \begin{bmatrix} -0.49 & 0.30 & 0.82 \\ 0.12 & -0.91 & 0.41 \\ 0.86 & 0.30 & 0.41 \end{bmatrix} \begin{bmatrix} 35.69 & 0 & 0 & 0 & 0 \\ 0 & 7.02 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.33 & 0.31 & 0.79 & 0.39 & -0.15 \\ 0.38 & 0.20 & -0.11 & -0.53 & -0.73 \\ 0.25 & -0.86 & 0.36 & -0.25 & 0.01 \\ 0.45 & -0.26 & -0.47 & 0.67 & -0.25 \\ 0.69 & 0.22 & -0.14 & -0.25 & 0.62 \end{bmatrix}$$

the thin svd:

$$= \begin{bmatrix} -0.49 & 0.30 \\ 0.12 & -0.91 \\ 0.86 & 0.30 \end{bmatrix} \begin{bmatrix} 35.69 & 0 \\ 0 & 7.02 \end{bmatrix} \begin{bmatrix} 0.33 & 0.31 & 0.79 & 0.39 & -0.15 \\ 0.38 & 0.20 & -0.11 & -0.53 & -0.73 \end{bmatrix}$$

pseudo-inverse:

$$A^\dagger = \begin{bmatrix} 0.33 & 0.38 \\ 0.31 & 0.20 \\ 0.79 & -0.11 \\ 0.39 & -0.53 \\ -0.15 & -0.73 \end{bmatrix} \begin{bmatrix} \frac{1}{35.69} & 0 \\ 0 & \frac{1}{7.02} \end{bmatrix} \begin{bmatrix} -0.49 & 0.12 & 0.86 \\ 0.30 & -0.91 & 0.30 \end{bmatrix}$$

key point: pseudo-inverse solves

- least-squares estimation problems
- minimum-norm control problems

properties of the pseudo-inverse

- if A is invertible, then $A^\dagger = A^{-1}$
- A is $m \times n$ \implies A^\dagger is $n \times m$
- $(A^\dagger)^\dagger = A$
- $(A^T)^\dagger = (A^\dagger)^T$
- $(\lambda A)^\dagger = \lambda^{-1} A^\dagger$ for $\lambda \neq 0$
- caution: in general, $(AB)^\dagger \neq B^\dagger A^\dagger$

Estimation and Least-Squares

- assume A is skinny and full rank, so
 - $m > n$ so we have more measurements than unknowns; equations are *overdetermined*
 - $\text{null}(A) = \{0\}$, so there is at most one solution x to $Ax = y_{\text{meas}}$
- usually $y_{\text{meas}} = Ax + w$ with w some error or noise; there are usually no solutions to $Ax = y_{\text{meas}}$
- instead find the *least-squares solution*, the x that minimizes

$$\|y_{\text{meas}} - Ax\|$$

using differentiation

the *residual* is

$$r = Ax - y_{\text{meas}}$$

which we would like to minimize

so

$$\|r\|^2 = x^T A^T Ax - 2y_{\text{meas}}^T Ax + \|y_{\text{meas}}\|^2$$

differentiate with respect to x and set to zero

$$2x^T A^T A - 2y_{\text{meas}}^T A = 0$$

so the optimum x is

$$x_{\text{opt}} = (A^T A)^{-1} A^T y_{\text{meas}}$$

geometric approach

pick as estimate x_{opt} ; by orthogonality

$$Ax_{\text{opt}} - y_{\text{meas}} \perp \text{range}(A)$$

which holds if and only if

$$Ax_{\text{opt}} - y_{\text{meas}} \in \text{null}(A^T)$$

which holds if and only if

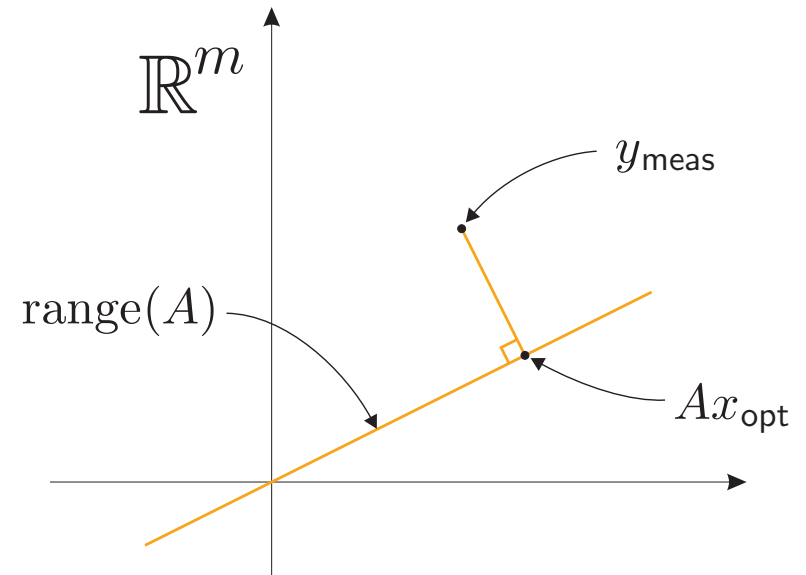
$$A^T(Ax_{\text{opt}} - y_{\text{meas}}) = 0$$

so x_{opt} is optimal if and only if

$$A^T Ax_{\text{opt}} = A^T y_{\text{meas}} \quad \text{called } \textit{the normal equations}$$

A is skinny and full rank, so $A^T A$ is invertible, so as before

$$x_{\text{opt}} = (A^T A)^{-1} A^T y_{\text{meas}}$$



pseudo-inverse approach

if A is skinny and full rank then $A^T A$ is invertible and

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

to see this, notice that the thin SVD of A is $A = \hat{U}\hat{\Sigma}V^T$, where V is square and orthogonal, so

$$A^T A = V\hat{\Sigma}^2V^T$$

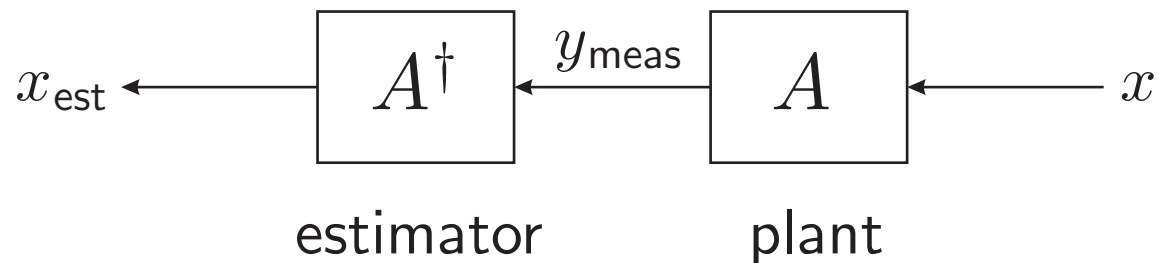
and

$$(A^T A)^{-1} A^T = V\hat{\Sigma}^{-2}V^T V\hat{\Sigma}\hat{U}^T = V\hat{\Sigma}^{-1}\hat{U}^T = A^\dagger$$

left-inverse property

when A is skinny and full rank, A^\dagger is a *left-inverse* for A

$$A^\dagger A = I$$



because $A^\dagger = (A^T A)^{-1} A^T$

this is exactly what we need for estimation, because if $y_{\text{meas}} = Ax$, choosing estimate $x_{\text{est}} = A^\dagger y$ give

$$x_{\text{est}} = A^\dagger Ax = x$$

effects of noise on estimation

Suppose we measure $y_{\text{meas}} = Ax + w$ and we use estimator B with $BA = I$.

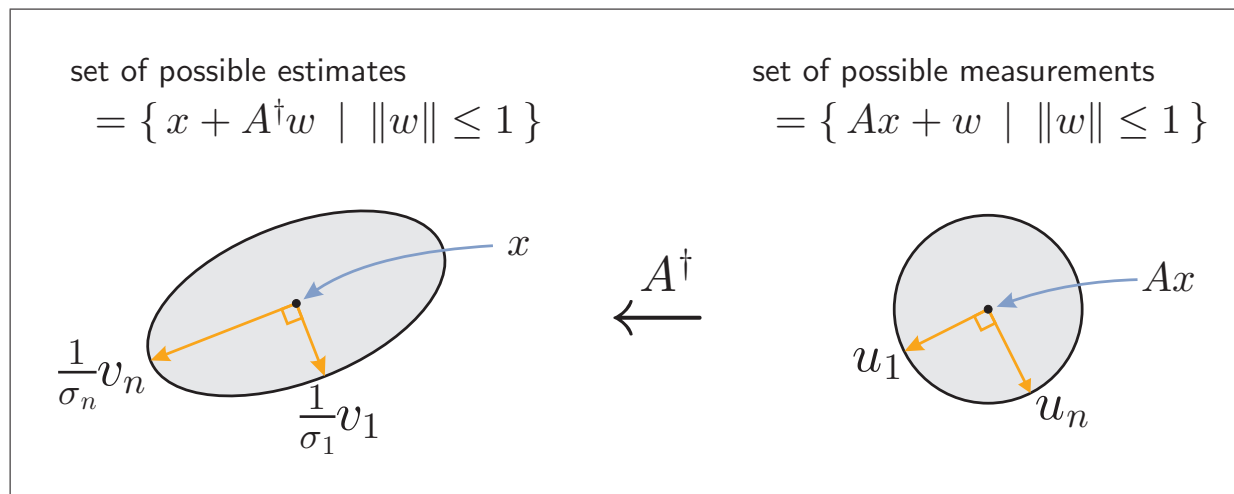
$$x_{\text{est}} = By_{\text{meas}}$$

If $\|w\| \leq 1$, then the estimate lies in the ellipsoid

$$x_{\text{est}} \in \{x + Bw \mid \|w\| \leq 1\}$$

because $x_{\text{est}} = B(Ax + w) = x + Bw$.

Picking $B = A^\dagger$ gives semiaxis directions v_i and semiaxis lengths σ_i^{-1} , worst error $\|e\| = 1/\sigma_{\min}(A)$



the best estimator

if A is skinny and full rank, then there are many left-inverses. e.g.

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad C = \begin{bmatrix} 0.5 & 0 & 0.5 \\ 0 & 1 & 0 \end{bmatrix}$$

both B and C are left-inverses of A

C averages out measurements 1 and 3, B just discards measurement 3

if $BA = I$ then B will work as an estimator, but

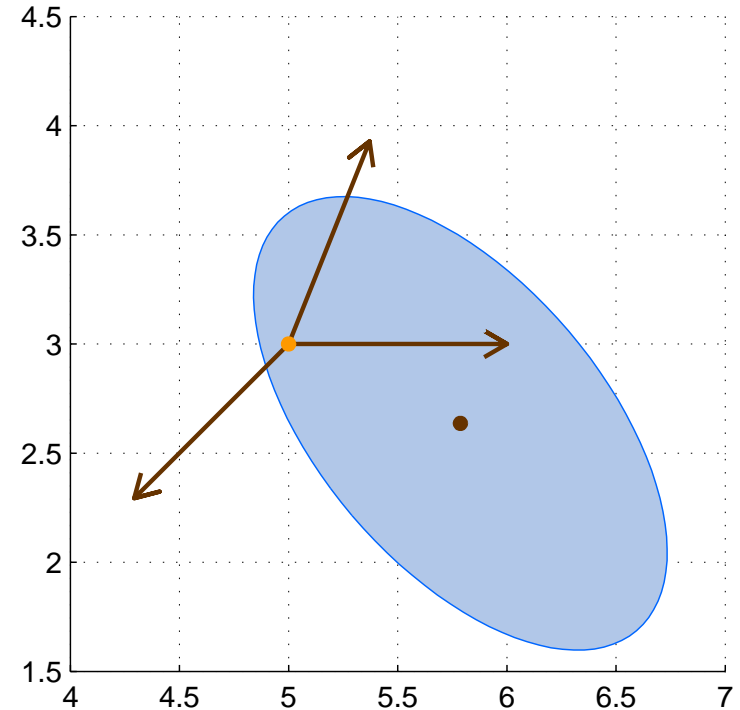
- $A^\dagger A^{\dagger T} \leq BB^T$
- $\|A^\dagger\| \leq \|B\|$; the pseudo-inverse is 'smaller'
- in fact, $\sigma_i(A^\dagger) \leq \sigma_i(B)$ for all i
i.e., the pseudo-inverse generates a smaller error-ellipsoid

example: navigation

here $A \in \mathbb{R}^{3 \times 2}$ with

$$A = \begin{bmatrix} b_1^T \\ b_2^T \\ b_3^T \end{bmatrix}$$

and $y = Ax$. Each b_i is a unit vector.



- x is unknown.
- y is measured; y_i is range measurement in the direction b_i with noise w added
- beacons at $\begin{bmatrix} 50 \\ 0 \end{bmatrix}$, $\begin{bmatrix} 20 \\ 50 \end{bmatrix}$, $\begin{bmatrix} -50 \\ -50 \end{bmatrix}$
- figure shows least-squares estimate plus set of locations consistent with $\|w\| \leq 1$

Proof of optimality property

suppose $A \in \mathbb{R}^{m \times n}$, $m > n$, is full rank

SVD: $A = U\Sigma V^T$, with V orthogonal

$B_{\text{ls}} = A^\dagger = V\Sigma^{-1}U^T$, and B satisfies $BA = I$

define $Z = B - B_{\text{ls}}$, so $B = B_{\text{ls}} + Z$

then $ZA = ZU\Sigma V^T = 0$, so $ZU = 0$ (multiply by $V\Sigma^{-1}$ on right)

therefore

$$\begin{aligned} BB^T &= (B_{\text{ls}} + Z)(B_{\text{ls}} + Z)^T \\ &= B_{\text{ls}}B_{\text{ls}}^T + B_{\text{ls}}Z^T + ZB_{\text{ls}}^T + ZZ^T \\ &= B_{\text{ls}}B_{\text{ls}}^T + ZZ^T \\ &\geq B_{\text{ls}}B_{\text{ls}}^T \end{aligned}$$

using $ZB_{\text{ls}}^T = (ZU)\Sigma^{-1}V^T = 0$

code for navigation example

```
beacons=[ 50, 0 ;    20, 50 ;    -50, -50    ];
m=size(beacons,1);

% create measurement matrix
for i=1:m
    A(i,:)=beacons(i,:)/norm(beacons(i,:));
end

ship_location=[5; 3];           % the unknown

for i=1:m
    range_measurements(i,1)=norm(ship_location-beacons(i,:));
    y(i,1) = norm(beacons(i,:)) - range_measurements(i,1) ;
end

y = y + randn(m,1);           % add noise

[U,S,V]=svd(A,0);             % thin svd

x_est=V*(inv(S)*(U'*y));       % estimate
```

regression or curve fitting

- model using a linear combination of functions

$$f(t) = x_1 f_1(t) + x_2 f_2(t) + \cdots + x_n f_n(t)$$

- collect m data samples

$$y_i = f(t_i) \quad i = 1, \dots, m$$

- write in matrix form

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} f_1(t_1) & \cdots & f_n(t_1) \\ f_1(t_2) & & f_n(t_2) \\ \vdots & & \vdots \\ f_1(t_m) & \cdots & f_n(t_m) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

- find least-squares estimate for x by $x_{\text{est}} = A^\dagger y$
- called *curve fitting* or *linear regression*; functions f_i are called *regressors*

example: polynomial curve fitting

- model

$$f(t) = x_1 + x_2t + x_3t^2 + \cdots + x_nt^{n-1}$$

- data samples

$$y_i = f(t_i) = \sum_{j=1}^n x_j t_i^{j-1}$$

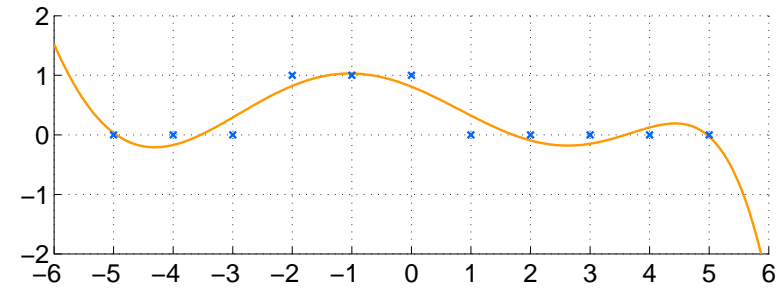
- write this as $y = Ax$ with

$$A = \begin{bmatrix} 1 & t_1 & t_1^2 & \cdots & t_1^{n-1} \\ 1 & t_2 & t_2^2 & \cdots & t_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_m & t_m^2 & \cdots & t_m^{n-1} \end{bmatrix}$$

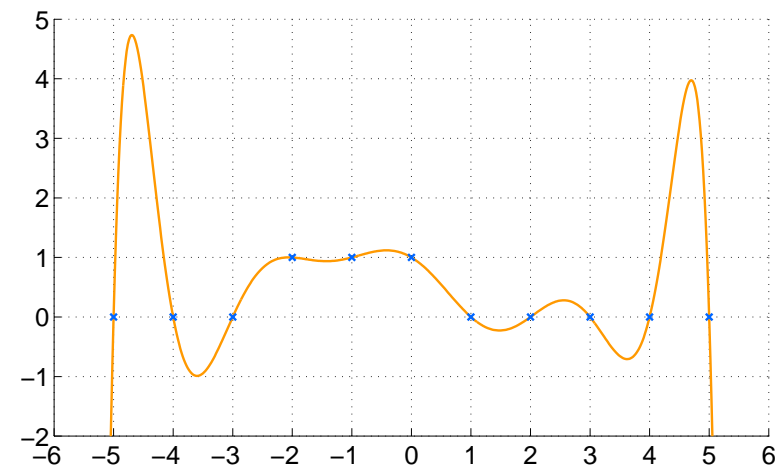
called *Vandermonde matrix*

- A has full rank if $m \geq n$ and t_i are distinct

order 7 fit



order 11 fit (exact)



example: rocket

- model

$$\text{height } h(t) = x_1 + x_2 t + x_3 t^2$$

where

x_1 = initial height

x_2 = initial vertical velocity

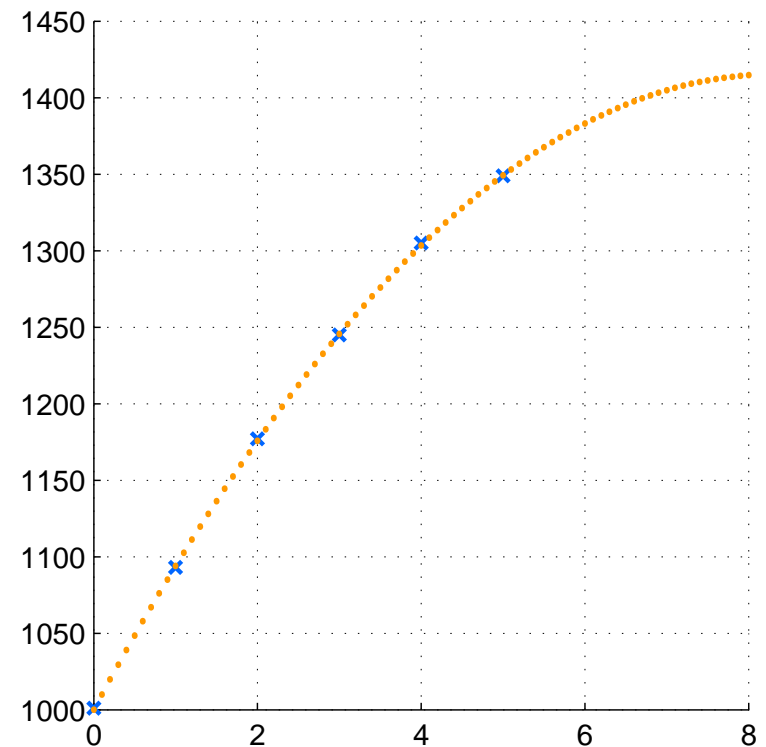
$x_3 = \frac{1}{2}(g - d)$ gravity minus drag

assume drag is constant over short burst
of data

- data from radar

t	0	1	2	3	4	5
$h(t)$	1001	1093	1177	1245	1305	1349

- estimated initial height $x_1 \approx 1000.43$
initial velocity $x_2 \approx 99.83$



code for rocket example

```
y=[1001; 1093; 1177; 1245; 1305; 1349]; % height measurement data
t=[ 0; 1; 2; 3; 4; 5];

m=size(y,1);
for i=1:m
    A(i,:)= [1, t(i), 0.5*t(i)^2]; % create measurement matrix
end

[U,S,V]=svd(A,0); % thin svd
x_est=V*(inv(S)*(U'*y));

t_other=0:0.1:8; % other times to estimate height
for i=1:size(t_other,2)
    A_other(i,:)= [1, t_other(i), 0.5*t_other(i)^2];
end

y_other=A_other*x_est; % estimated heights

h_initial=[1 0 0]*x_est; % estimates of initial height
v_initial=[0 1 0]*x_est; % and velocity
```

Control and Minimum-Norm Solutions

- assume A is fat and full rank
 - $m < n$, so more control inputs than outputs; equations are *underdetermined*
 - $\text{range}(A) = \mathbb{R}^m$, so there is always at least one x which achieves $y_{\text{des}} = Ax$
- usually there is more than one solution x
- among all x that satisfy $y_{\text{des}} = Ax$ we find the one with minimum norm
- called *minimum-norm solution*

geometric approach

orthogonality gives

$$x_{\text{opt}} \perp \text{null}(A)$$

which holds if and only if

$$x_{\text{opt}} \in \text{range}(A^T)$$

which holds if and only if

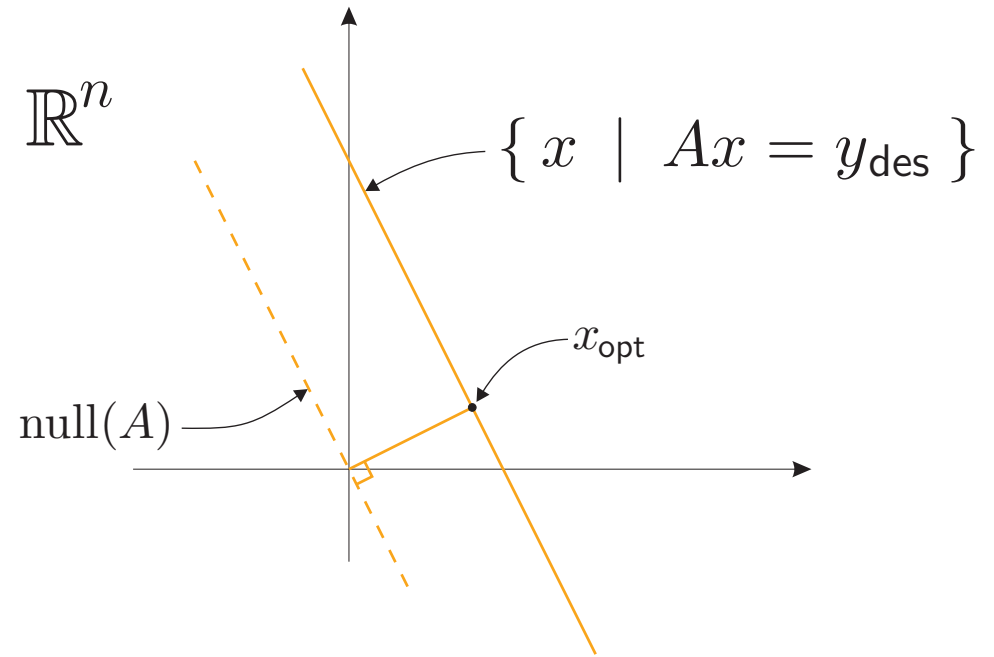
$$x_{\text{opt}} = A^T z \text{ for some } z$$

since $Ax_{\text{opt}} = y_{\text{des}}$ this holds if and only if

$$AA^T z = y_{\text{des}} \quad \text{the } \textit{normal equations}$$

A is fat and full rank, so AA^T is invertible, so $z = (AA^T)^{-1}y_{\text{des}}$ so

$$x_{\text{opt}} = A^T (AA^T)^{-1} y_{\text{des}}$$



solution via Lagrange multipliers

$$\begin{array}{ll} \text{minimize} & \|x\|^2 \\ \text{subject to} & Ax = y_{\text{des}} \end{array}$$

introduce Lagrange multipliers

$$L(x, \lambda) = x^T x + \lambda^T (Ax - y_{\text{des}})$$

optimality conditions are

$$\frac{\partial L}{\partial x} = 2x_{\text{opt}}^T + \lambda^T A = 0$$

$$\frac{\partial L}{\partial \lambda} = (Ax_{\text{opt}} - y_{\text{des}})^T = 0$$

from the first condition $x_{\text{opt}} = -\frac{1}{2}A^T\lambda$, and

$$AA^T\lambda = -2y_{\text{des}} \quad \implies \quad \lambda = -2(AA^T)^{-1}y_{\text{des}}$$

so as before

$$x_{\text{opt}} = A^T(AA^T)^{-1}y_{\text{des}}$$

pseudo-inverse approach

if A is fat and full rank then

$$A^\dagger = A^T(AA^T)^{-1}$$

because the thin SVD of A is $A = U\hat{\Sigma}\hat{V}^T$ where U is square and orthogonal, so

$$AA^T = U\hat{\Sigma}^2U^T$$

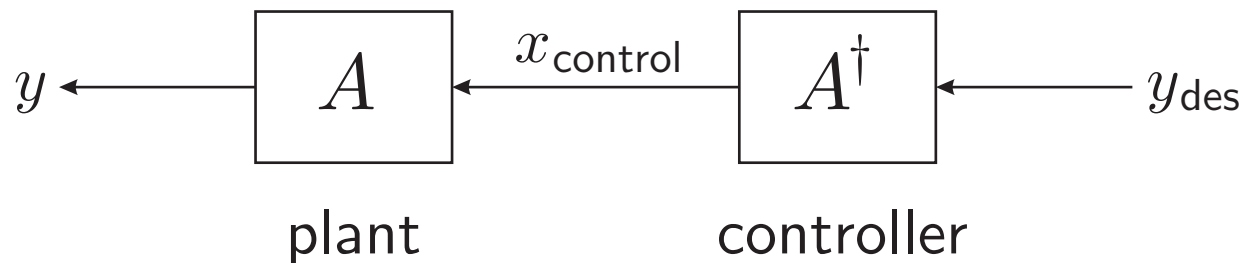
and

$$A^T(AA^T)^{-1} = \hat{V}\hat{\Sigma}U^T U\hat{\Sigma}^{-2}U^T = \hat{V}\hat{\Sigma}^{-1}U^T = A^\dagger$$

right-inverse property

when A is fat and full rank, A^\dagger is a *right-inverse* for A

$$AA^\dagger = I$$



because $A^\dagger = A^T(AA^T)^{-1}$

- this is exactly what we need for control, because choosing $x_{\text{control}} = A^\dagger y_{\text{des}}$ gives

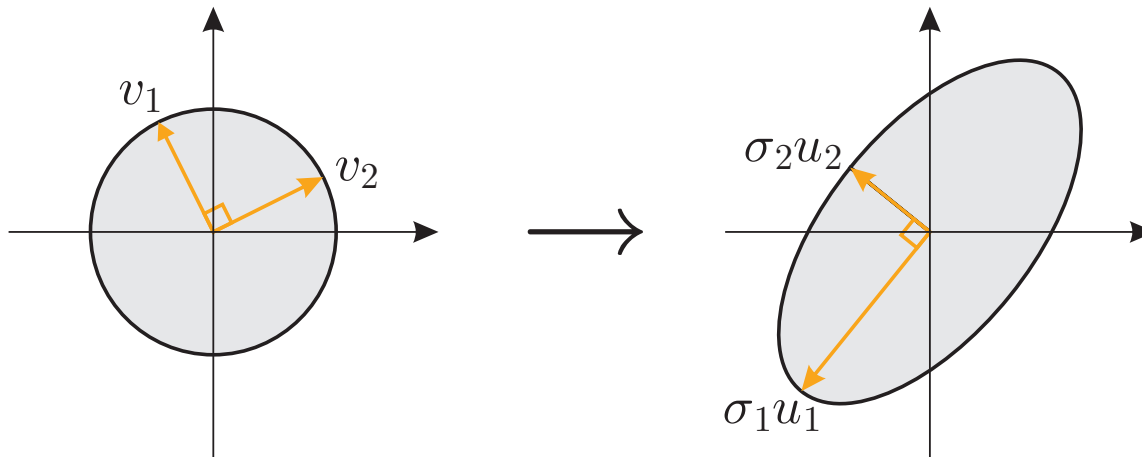
$$y = Ax_{\text{control}} = AA^\dagger y_{\text{des}} = y_{\text{des}}$$

- if B is a left-inverse for A^T then B^T is a right-inverse for A ; control and estimation problems are called *dual*

size of optimal input

$$\begin{aligned}\|x_{\text{opt}}\|^2 &= \|A^T(AA^T)^{-1}y_{\text{des}}\|^2 \\ &= y_{\text{des}}^T(AA^T)^{-1}y_{\text{des}}\end{aligned}$$

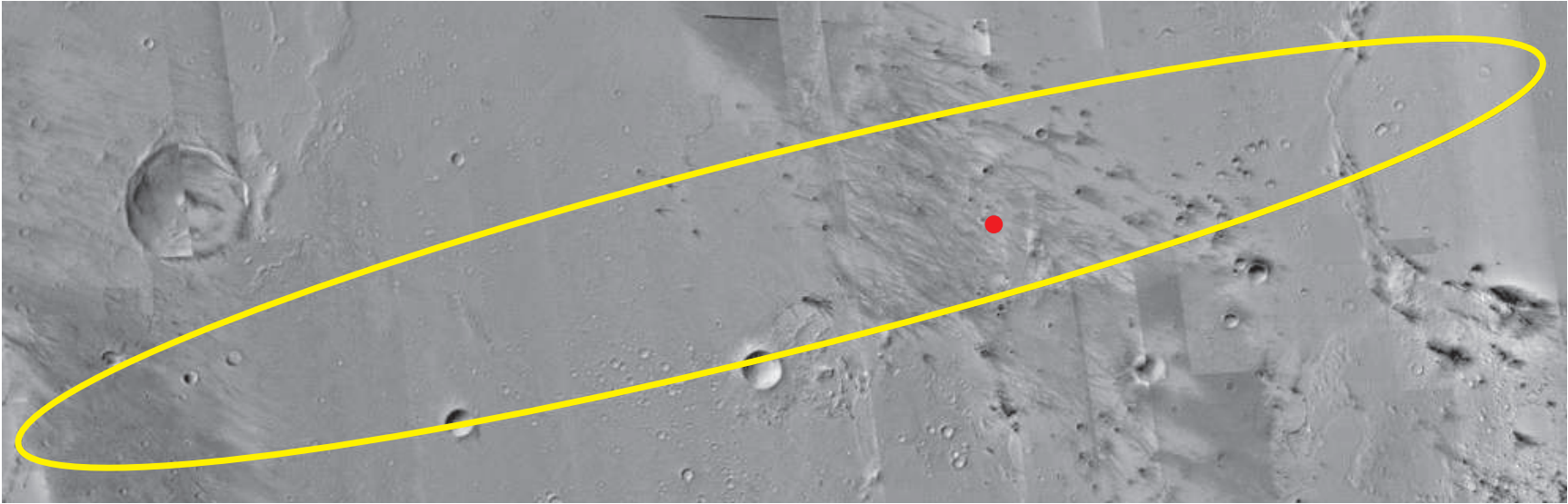
as expected from SVD



if A is fat and full rank, then the ellipsoid is

$$E = \left\{ y \in \mathbb{R}^m \mid y^T(AA^T)^{-1}y \leq 1 \right\}$$

example: ellipsoids

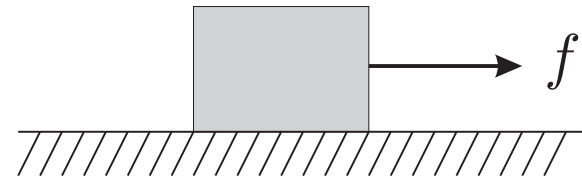
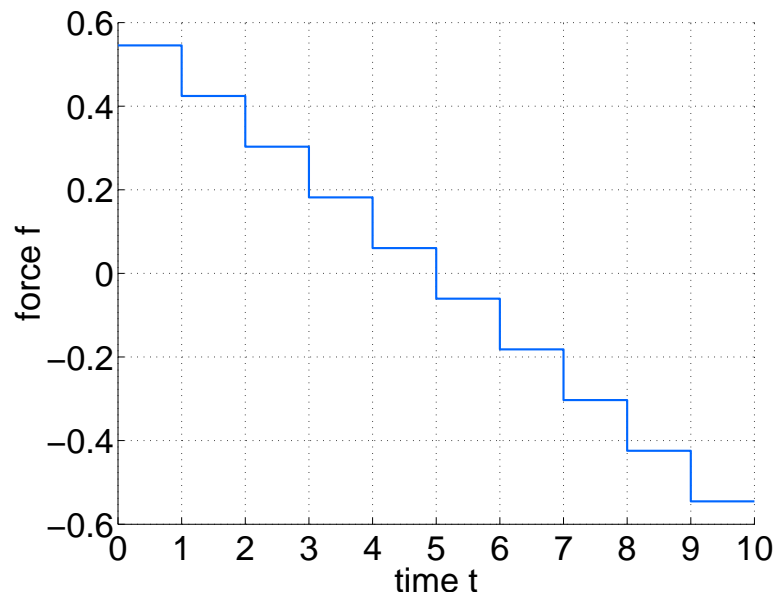


- Mars rover *Spirit*, landed 3 Jan 2004
- Predicted to land within ellipsoid 83km by 10km
- View looking east is below



example: force on mass

- x is the sequence of applied forces, so $f(t) = x_j$ for t in the interval $[j - 1, j]$.
- y_1, y_2 are final position and velocity
- $y = Ax$ where $A = \begin{bmatrix} 9.5 & 8.5 & 7.5 & 6.5 & 5.5 & 4.5 & 3.5 & 2.5 & 1.5 & 0.5 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$
- find minimum norm force input x so that final position = 10, final velocity = 0.
- optimal sequence of forces is



Matrices Without Full Rank

- estimation: A is skinny – typically more than one least squares solution
- control: A is fat – typically no x satisfying $Ax = y_{\text{des}}$

$A^\dagger y$ gives

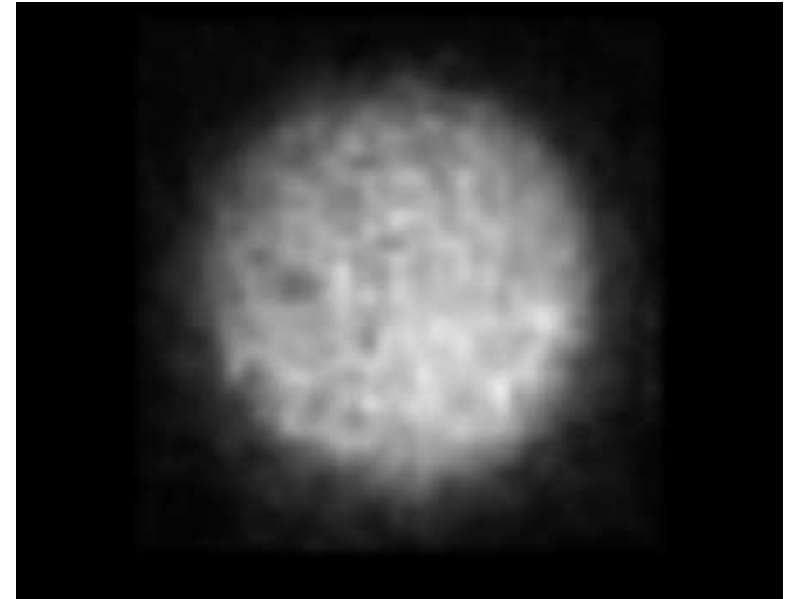
from all x that minimize $\|Ax - y\|$, $A^\dagger y$ is the one with minimum norm

Matlab and the Pseudo-Inverse

- A^\dagger is given by `pinv(A)`
- `svd(A,0)` returns the thin svd for *skinny matrices*, but not for fat ones
- don't compute `pinv(A)*y`, instead
 1. compute thin SVD (look at the singular values, and truncate U , Σ and V)
 2. compute $z = \hat{U}^T y$, then $w = \hat{\Sigma}^{-1} z$, then $x = \hat{V} w$.
- if A is skinny and full rank, then `x=A\y` will give the least squares solution
- if A is *not* skinny and full rank, then `A\y` will give you something else, which you probably don't want

History of Least Squares

- January 1801: Giuseppe Piazzi, director of Palermo observatory, observed a new 'star'
- really asteroid *Ceres*, 900km diameter
- by autumn it had disappeared behind the sun; nobody could find it again
- September 1801, Carl F. Gauss (1777–1855) developed *least squares*; purpose was to fit observed data to an elliptical trajectory
- Gauss predicted its trajectory
- scientific community was *amazed*.
- Gauss became director of Göttingen observatory
- he published the method in 1809



Hubble telescope, ultraviolet, 2001



bounds on relative error

suppose we are computing $y = Ax$, and A is square and invertible

perturbing x to $x + \delta x$ results in y changing to $y + \delta y$

relative error is

$$\begin{aligned} \frac{\|\delta y\|}{\|y\|} / \frac{\|\delta x\|}{\|x\|} &= \frac{\|A(x + \delta x) - Ax\|}{\|Ax\|} / \frac{\|\delta x\|}{\|x\|} \\ &= \frac{\|A\delta x\|}{\|\delta x\|} / \frac{\|Ax\|}{\|x\|} \\ &\leq \|A\| \|A^{-1}\| \end{aligned}$$

because

$$\frac{\|x\|}{\|Ax\|} = \frac{\|A^{-1}Ax\|}{\|Ax\|} \leq \|A^{-1}\|$$

The quantity $\kappa(A) = \|A\| \|A^{-1}\|$ is called the *condition number* of A .

properties of the condition number

- $\kappa(A) = \|A\| \|A^{-1}\|$ if A is square and invertible
- otherwise $\kappa(A) = \|A\| \|A^\dagger\| = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}$
- if $\kappa(A)$ is small we call A *well-conditioned*, otherwise we say A is *ill-conditioned*.
- $\kappa(A)$ is the *eccentricity* of the ellipse that is the image of the unit ball under A .

uses of the condition number

- relative error computing y using $y = Ax$ is $\frac{\|\delta y\|}{\|y\|} / \frac{\|\delta x\|}{\|x\|} \leq \kappa(A)$
- relative error computing x from $y = Ax$ is $\frac{\|\delta x\|}{\|x\|} / \frac{\|\delta y\|}{\|y\|} \leq \kappa(A)$
- relative error computing x from $y = Ax$ with *errors in A* is

$$\frac{\|\delta x\|}{\|x\|} / \frac{\|\delta A\|}{\|A\|} \leq \kappa(A)$$