$y = Ax \longrightarrow x = A \backslash y$ ← introduced by MATLAB mdivide ...

Least-Squares

Gabriel Hugh Elkaim

Least-Squares

- Least-squares (approximate) solution of overdetermined equations
- Projection and orthagonality principle
- Least-squares estimation
- BLUE (Best Linear Unbiased Estimator) property

Overdeterimined Linear Equations

$y = Ax \quad A \in \mathbb{R}^{m \times n} \quad m > n$ strictly "skinny" A

over determined – more equations than unknowns

for most y, cannot solve for x

$R(A)$ is at most n.

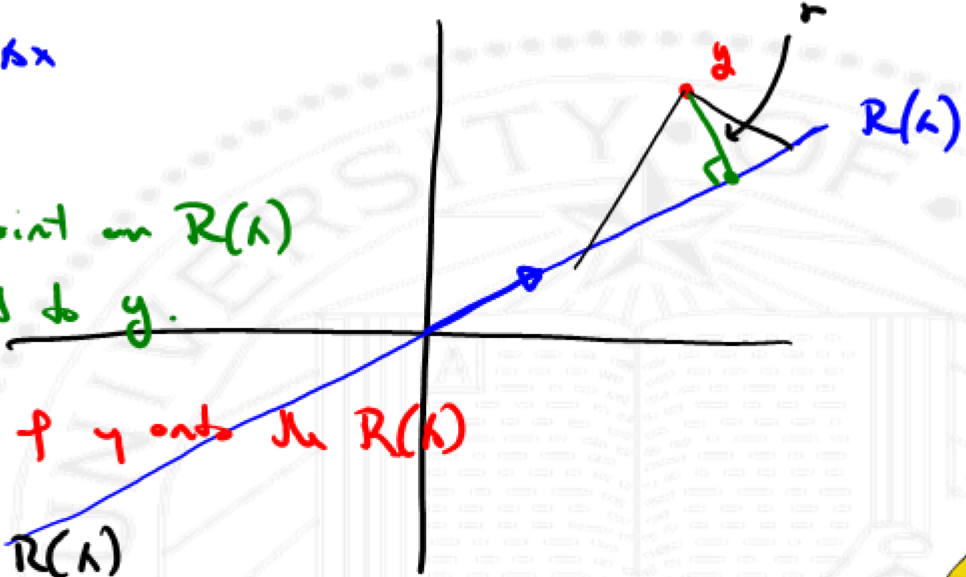
inconsistent answers.

Approximately solve $y = Ax$, error $r \triangleq Ax - y$

find x_{ls} $\min \|r\|_2$ → LEAST SQUARES

Geometric Interpretation

$A = \begin{bmatrix} \vdots \end{bmatrix}$ $\qquad y = Ax$



Ax_{ls} is the point on $R(A)$ that is closest to y.

Ax_{ls} projection of y onto the $R(A)$

$r \perp R(A)$

Least-Squares (approximate) solution

Assume that A is skinny and full rank $\begin{bmatrix} \\ \\ \end{bmatrix}$

$$\|r\|_2^2 = (y - Ax)^T (y - Ax) = x^T A^T A x - 2y^T A x + y^T y$$

$$\frac{\partial \|r\|_2^2}{\partial x} = 0 = 2x^T A^T A - 2y^T A = 0$$

$$\left[x^T A^T A = y^T A\right]^T \longrightarrow A^T A x = A^T y$$

$$\underbrace{[A^T A]^{-1} A^T A}_{I} x = [A^T A]^{-1} A^T y$$

$$\boxed{x_{ls} = [A^T A]^{-1} A^T y}$$ ← Normal Equations

$[A^T A]^{-1} A^T$ — A^{+} pseudo-inverse

moore-penrose

$x = A \backslash y = \text{pinv}(A) \cdot y$

Least-Squares (approximate) solution

$$x_{ls} = \underline{(A^TA)^{-1}A^T}\, y$$

x_{ls} is a linear function of y. $\quad x_{ls} = By \quad B = (A^TA)^{-1}A^T$

$A^\dagger \triangleq (A^TA)^{-1}A^T$ pseudo-inverse $\quad A^TA$ is "small", square, invertible

if A is square and invertible, then $A^\dagger = A^{-1}$

$$(A^TA)^{-1}A^TA = A^{-1}\underbrace{A^{-T}\cdot A^T}_{I}A = A^{-1}A = I.$$

$A^{\dagger}$ is a LEFT inverse of A when A is skinny, full rank

$$A^{\dagger} A = I_{n \times n}$$

$$\begin{array}{cc} \overset{n}{\longleftrightarrow} & \\ \begin{bmatrix} \\ \\ \end{bmatrix} & m \end{array} \quad m > n$$

Projection on *R(A)*

Ax_{ls} by definition the point on $R(A)$ closest to y.

Ax_{ls} is the projection of y onto $R(A)$

$$Ax_{ls} = P_r(x)\,y = Ax_{ls} = A(A^TA)^{-1}A^Ty.$$

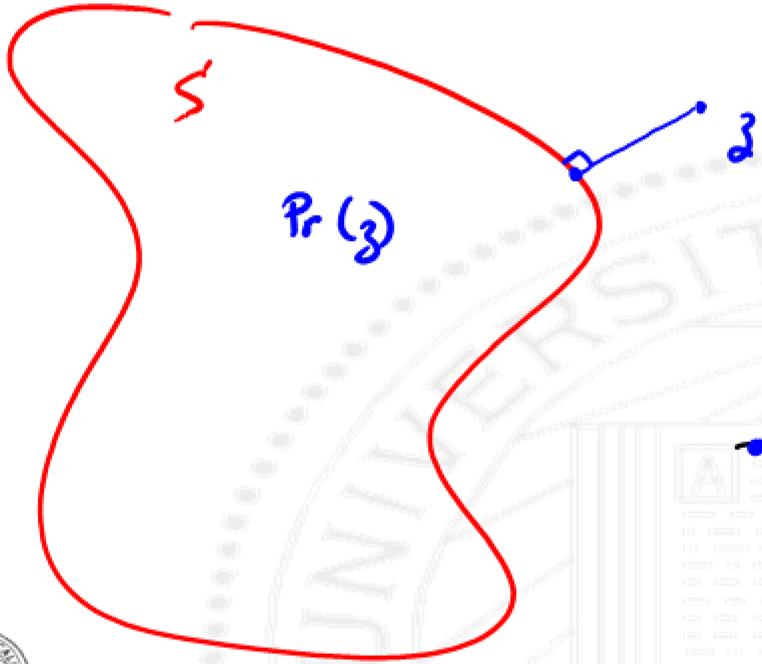
$$P_r(x) \triangleq A(A^TA)^{-1}A^T \leftarrow \text{Projection matrix.}$$

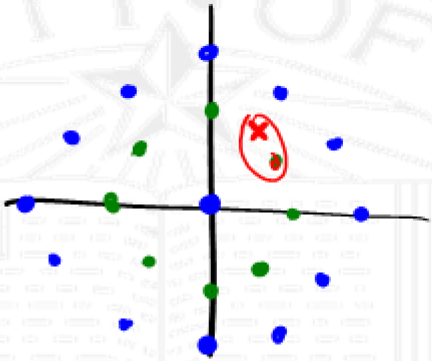
Orthogonality Principle

Optimal Residue

$$\|r\|_{min} = Ax_{ls} - y = \underbrace{\left[A(A^TA)^{-1}A^T - I\right]}_{\text{orthogonal to } R(A)} y.$$

$$\langle r, Az \rangle = y^T\left[A(A^TA)^{-1}A^T - I\right]^T Az = 0 \quad \forall\, z \in \mathbb{R}^n.$$


S
z
Pr (z)



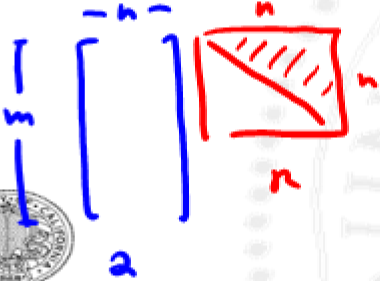
Least Squares via QR Factorization

$A \in \mathbb{R}^{m\times n}$ skinny, full rank $\begin{bmatrix} \; \end{bmatrix}$

$A = QR$ $\quad Q^TQ = I_{n\times n}$ $\quad R \in \mathbb{R}^{n\times n}$

upper triangular

$Q \in \mathbb{R}^{m\times n}$



$$A^\dagger = (A^TA)^{-1}A^T = [R^T\underbrace{Q^TQ}_{I}R]^{-1}R^TQ^T$$

$$= [R^TR]^{-1}R^TQ^T = R^{-1}\underbrace{R^{-T}R^T}_{I_{n\times n}}Q^T$$

$$\boxed{A^\dagger = R^{-1}Q^T}$$

$$P_r(A) = A(A^TA)^{-1}A^T = AR^{-1}Q^T = Q\underbrace{RR^{-1}}_{I}Q^T = QQ^T$$

$$A = QR$$

$$\boxed{P_r(A) = QQ^T}$$ ← NOT IDENTITY!!

$$P_r(A)\cdot y = \sum_{i=1}^{n} (q_i^T y)\cdot q_i$$

Least-Squares via "Full" QR Factorization (1.3)

$$A = [Q_1 \vdots Q_2]\begin{bmatrix} R_1 \\ \cdots \\ 0 \end{bmatrix} \qquad [Q_1 \vdots Q_2] \in \mathbb{R}^{m \times m} \text{ orthogonal}$$

$$R_1 \in \mathbb{R}^{n \times n} \text{, invertible}$$

$$\|Ax - y\|_2^2 = \left\| [Q_1\ Q_2]\begin{bmatrix} R_1 \\ 0 \end{bmatrix} x - y \right\|_2^2 \qquad [Q_1, Q_2]^T$$

$$\left\| [Q_1, Q_2]^T [Q_1, Q_2] \begin{bmatrix} R_1 \\ 0 \end{bmatrix} x - [Q_1, Q_2]^T y \right\|^2$$

$$\left\| \begin{matrix} R_1 x - Q_1^T y \\ 0 - Q_2^T y \end{matrix} \right\|^2 = \|R_1 x - Q_1^T y\|^2 + \|Q_2^T y\|^2$$

Least-Squares via "Full" QR Factorization (2.3)

$$\|Ax - y\|^2 = \|R_1 x - Q_1^T y\|^2 + \|Q_2^T y\|^2$$

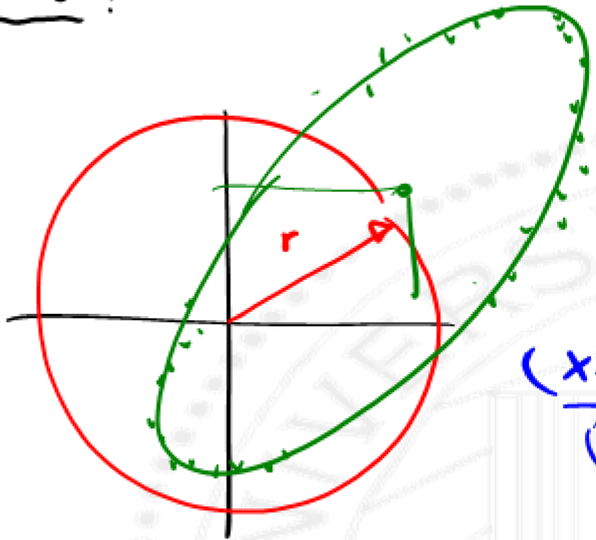
$$x_{ls} = R_1^{-1} Q_1^T y \rightarrow \underbrace{\|\underbrace{R_1 R_1^{-1}}_{I} Q_1^T y - Q_1^T y\|^2}_{0} + \|Q_2^T y\|^2$$

$$A x_{ls} - y = -Q_2 Q_2^T y.$$

$Q_1 Q_1^T$ projection of y on $R(A)$

$Q_2 Q_2^T$ projection of y on $R(A)^{\perp} = N(A^T)$.

Questions ?



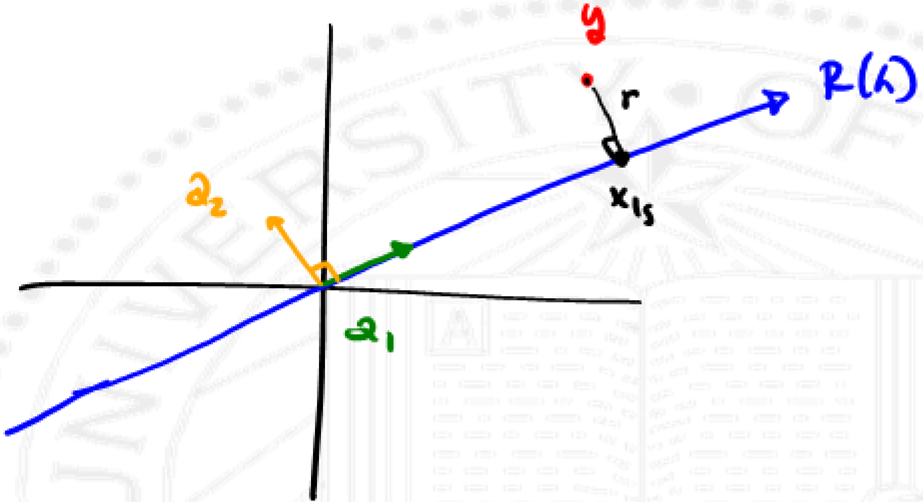
$$y = \left[\;\right] \underline{x} + [b]$$

$$\frac{(x-x_0)^2}{a} + \frac{(x-x_0)(y-y_0)}{ab} + \frac{(y-y_0)^2}{b^2}$$

r^2

Least-Squares via "Full" QR Factorization (3.3)

$y = Ax$



Least-Squares Estimation (1.2)

$$y = Ax + v$$

noise

Forward model

x is what we want to reconstruct or estimate ($\hat{x}$)

y our sensor measurements.

v is an unknown measurement noise or sensor error

assume that v is "small"

Least-Squares Estimation (2.2)

i^{th} row of A characterizes the i^{th} sensor.

$$\hat{x} \simeq x \qquad \hat{y} = A\hat{x} \longrightarrow \boxed{\hat{v} = y - A\hat{x}}$$

$\min_{\hat{x}} \|A\hat{x} - y\|$ minimize the deviation between

- what we observe (y)
- what we would observe if $x = \hat{x}$ and there were no noise.

$$\hat{x} = (A^T A)^{-1} A^T y$$

BLUE Property (1.2)

BEST LINEAR ESTIMATOR

UNBIASED

$y = Ax + v \quad A \in \mathbb{R}^{m \times n}$ skinny, full rank

linear estimator : $\hat{x} = By \qquad \hat{x} = B(Ax+v) = BAx + Bv$

B must be a LEFT inverse of A.

unbiased if $\hat{x} = x$ if $v = 0$. $\longrightarrow BA = I$. any left inverse of A.

$x - \hat{x} = x - BAx - Bv = -Bv$

WANT B to be the "SMALLEST" LEFT inverse of A

BLUE Property (2.2)

$A^{+} = [A^T A]^{-1} A^T$ is the smallest left inverse of A.

for any $B \mid BA = I$ $\quad \sum_{i,j} B_{ij}^2 \geq \sum_{i,j} A^{+2}_{i,j}$ } Frobenius Norm

$A^{+} A = I$

$(A^{+} + \alpha F) A = I$

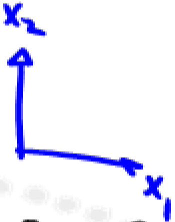
$\alpha F A = 0$.

F should be small

$F \in N(A^{+})$

rows of F are orthogonal to columns of A (2nd LLS & SVD)

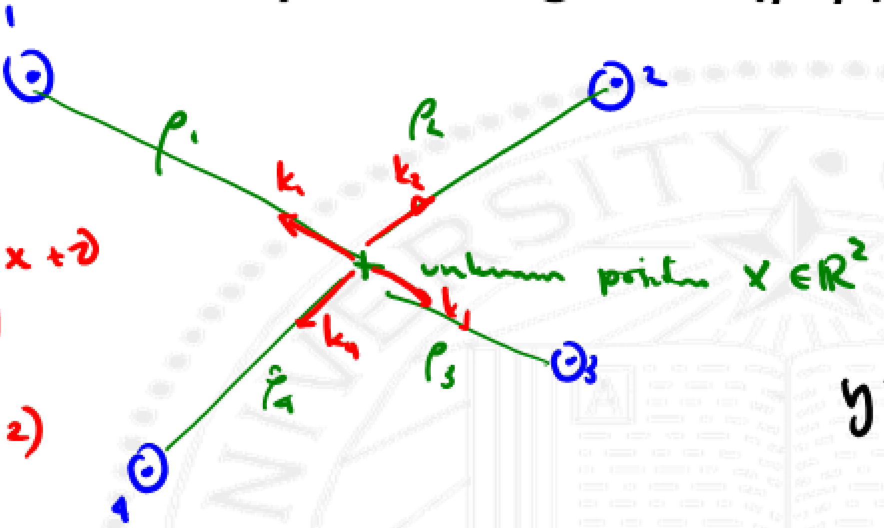
Example: Navigation (ρ-ρ)



initial $x_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$$y = -\begin{bmatrix} k_1^T \\ \vdots \\ k_4^T \end{bmatrix} x + v$$

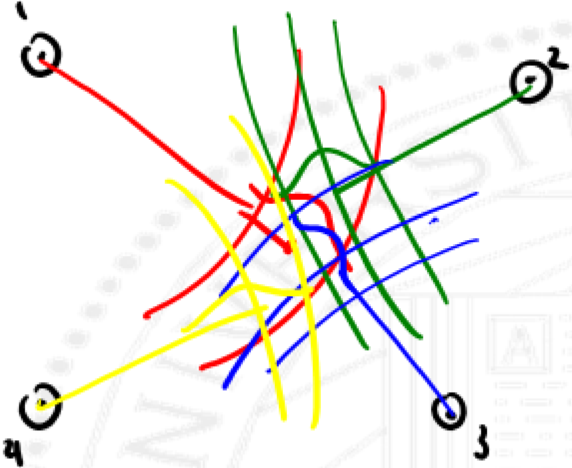
$v \sim (0, 2)$



$$x_{true} = \begin{bmatrix} 5.59 \\ 10.55 \end{bmatrix}$$

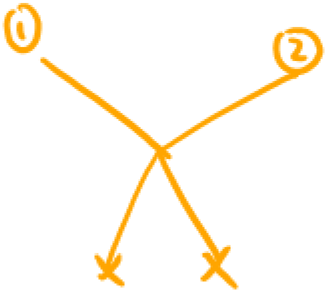
$$y = \begin{bmatrix} -11.95 \\ -2.84 \\ -9.81 \\ 2.81 \end{bmatrix}$$

Example: Navigation (ρ-ρ)



given y, estimate x.

Just Enough Measurements



$$\hat{x} = By = \begin{bmatrix} 0 & -1 & \boxed{\begin{matrix} 0 \\ 0 \end{matrix}} & \boxed{\begin{matrix} 0 \\ 0 \end{matrix}} \\ -1.12 & 0.5 & & \end{bmatrix} y.$$

$$\hat{x} = \begin{bmatrix} 2.79 \\ 11.9 \end{bmatrix} \quad \|r\| = 3.02$$

BEST SOLUTION IF I KNOW BEACONS 3 + 4 ARE BAD.

R.A.J.M.

Least-Squares Method

$$\hat{x} = (A^T A)^{-1} A^T y = A^{\dagger} y = \underbrace{\begin{bmatrix} -.21 & -0.48 & 0.09 & 0.79 \\ -.98 & -0.02 & -.51 & -.16 \end{bmatrix}}_{\text{BLENDING MATRIX}}$$

$$\hat{x} = \begin{bmatrix} 9.95 \\ 10.26 \end{bmatrix} \qquad \|r\| - 0.72$$

SENSOR FUSION

B and $A^{\dagger}$ are both left inverses of A

Larger entries of B_{ij} lead to larger errors.

Example from Overview: Comm

$$\xrightarrow{u} \boxed{H(s)} \xrightarrow{w} \boxed{A/D} \xrightarrow{y}$$

u is piecewise constant, period of 1 sec until $0 \le t \le 10$

$$w(t) = \int_0^t h(t-\tau)\, u(\tau)\, d\tau$$

sample y @ 10x $\quad \tilde{y}_i = w(0.1 i) \quad i = 1$ to 100.

3 bit A/D $\quad Q(a) = \frac{1}{4}\left[\mathrm{round}(4a + \frac{1}{2}) - \frac{1}{2}\right]$

Comm. Example (1.3)

given $y = \begin{bmatrix} \vdots \end{bmatrix}^{100\times 1}$

estimate $\hat{u}$ 10×1

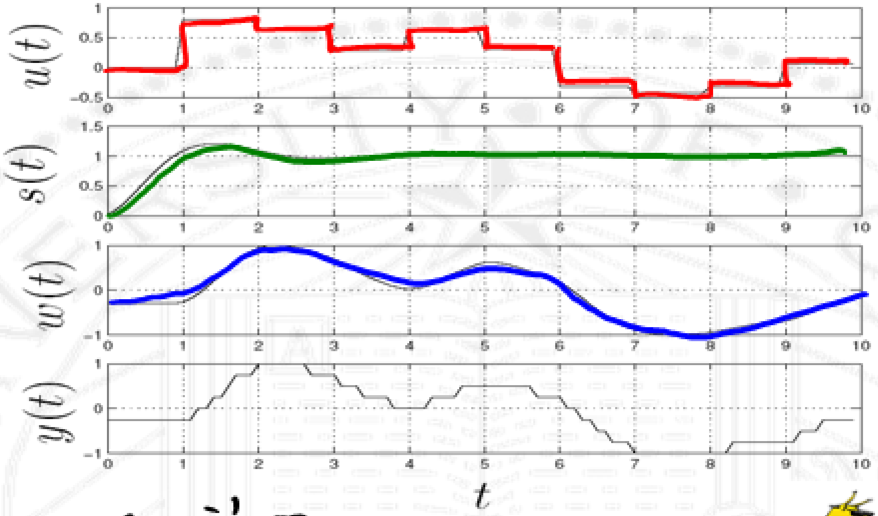
$y = Ax + v$

$A \in \mathbb{R}^{100\times 10}$

$A_{ij} = \int_{j-1}^{j} h(0.1i - \tau)\, d\tau$

$v \in \mathbb{R}^{100}$ $\quad v_i = Q(\tilde{y}_i)$

$|v_i| \le 0.125$



$x_{ls} = (A^T A)^{-1} A^T y.$

$x_{ls} = A^{\dagger} y$

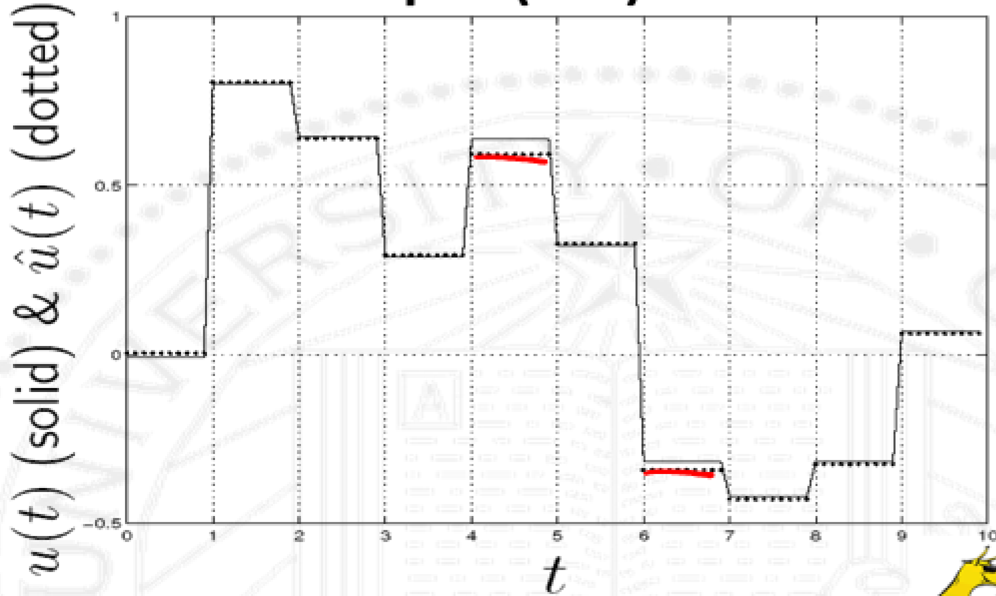
$\dfrac{\|x - x_{ls}\|}{\sqrt{10}} = \underline{0.02}$

$h(s) = 1.$

$\downarrow$

rms: $\underline{0.07}$

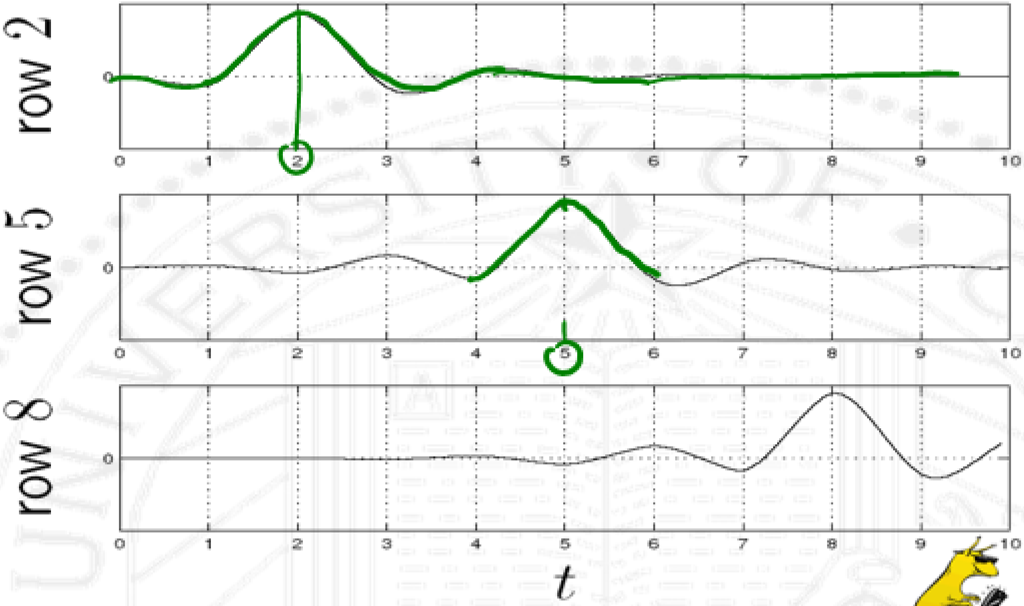
Comm. Example (2.3)



Row of $B_{ls} = A^t$ $\hat{x} = B_{ls} \cdot y$.

Comm. Example (3.3)

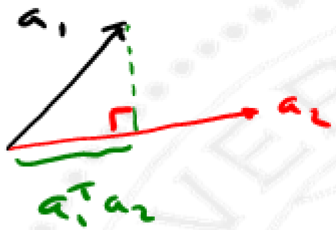
$x_5 \approx f(y_{40} - y_{60})$



Questions?

$\langle v_i, v_j \rangle = v_i^T v_j$ — inner product

orthogonal vectors — $\langle v_i, v_j \rangle = 0 \quad \forall\, i \neq j$



normal vector $\langle v_i, v_i \rangle = 1$