Chapter 8
Least-Squares Estimation

8.3 The Least-Squares (LS) Approach

All the previous methods we've studied… required a probabilistic model for the data: Needed the PDF $p(\mathbf{x};\theta)$

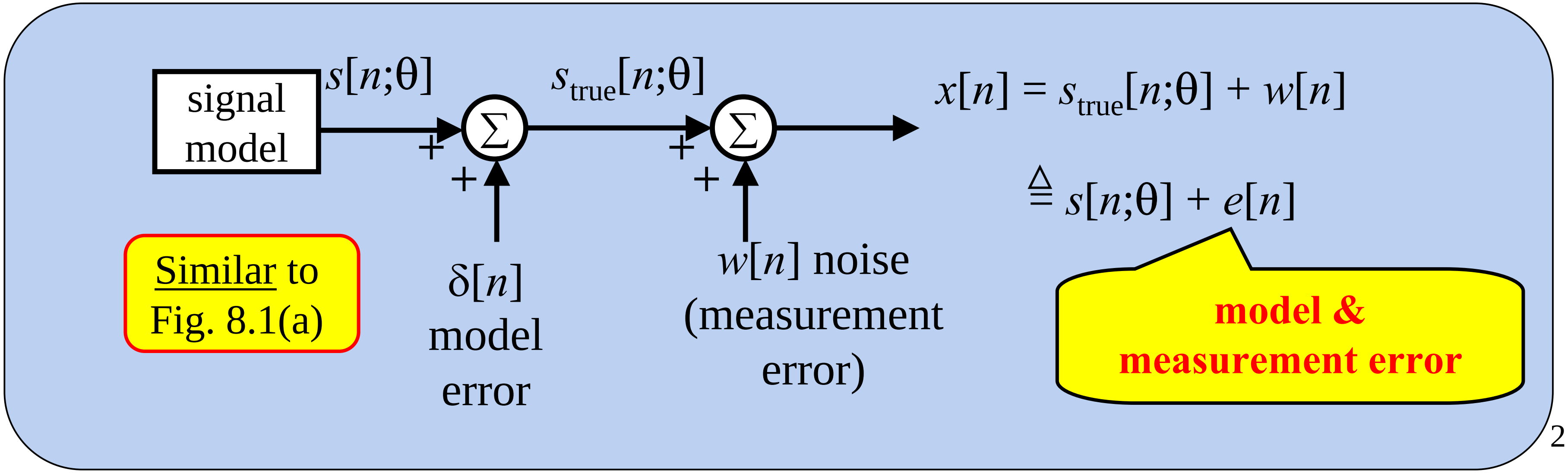
For a Signal + Noise problem we needed:

Signal Model & Noise Model

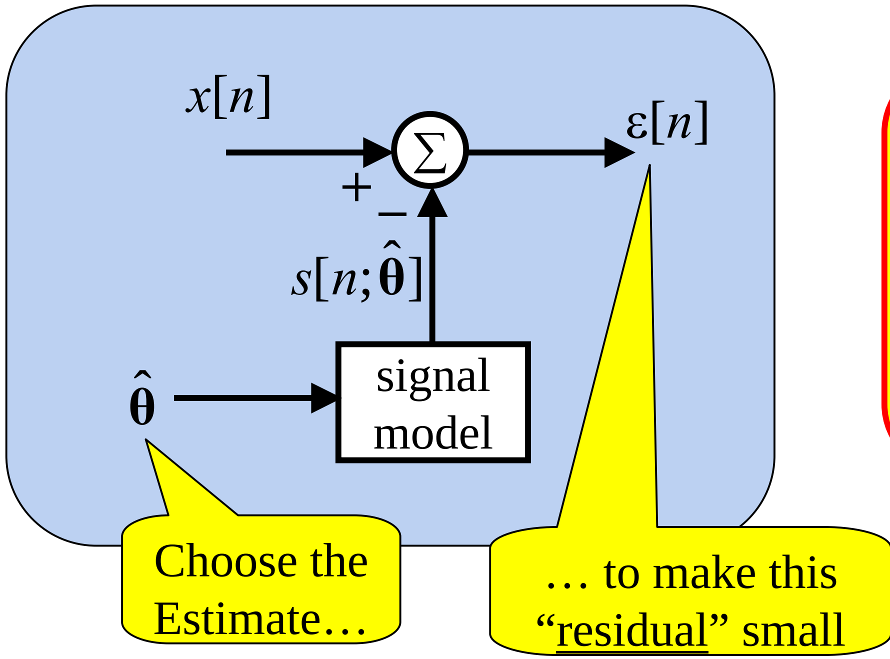
Least-Squares is not statistically based!!!

⇒ Do NOT need a PDF Model

⇒ Do NEED a Deterministic Signal Model



Least-Squares Criterion



Minimize the LS Cost

$$J(\boldsymbol{\theta}) = \sum_{n=0}^{N-1} \varepsilon^2[n] = \sum_{n=0}^{N-1} \left(x[n] - s[n;\boldsymbol{\theta}]\right)^2$$

Ex. 8.1: Estimate DC Level

$x[n] = A + e[n] = s[n;\theta] + e[n]$

$$J(A) = \sum_{n=0}^{N-1} (x[n] - A)^2$$

To Minimize…

$$Set\ \frac{\partial J(A)}{\partial A} = 0 \quad \Rightarrow \quad \hat{A} = \frac{1}{N}\sum_{n=0}^{N-1} x[n] = \bar{x}$$

Same thing we've gotten before!

Note:
If $e[n]$ is WGN, then LS = MVU

Weighted LS Criterion

Sometimes not all data samples are equally good:

$x[0], x[1], \ldots , x[N\text{-}1]$

Say you know $x[10]$ was poor in quality compared to other data…

You'd want to de-emphasize its importance in the sum of squares:

$$J(\theta) = \sum_{n=0}^{N-1} w_n (x[n] - s[n;\theta])^2$$

set this small to de-emphasize a sample

8.4 Linear Least-Squares

A linear least-squares problem is one where the parameter observation model is linear: $\mathbf{s} = \mathbf{H}\boldsymbol{\theta}$ $\qquad$ $\mathbf{x} = \mathbf{H}\boldsymbol{\theta} + \mathbf{e}$

$N\times 1$

$N\times p$ Known Matrix

$p\times 1$

p = **Order of the model**

We must assume that **H** is full rank… otherwise there are multiple parameter vectors that will map to the same **s!!!**

Note: Linear LS does NOT mean "fitting a line to data"… although that is a special case:

$$s[n] = A + Bn \quad \Rightarrow \quad \mathbf{s} = \underbrace{\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ \vdots & \vdots \\ 1 & N-1 \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} A \\ B \end{bmatrix}}_{\boldsymbol{\theta}}$$

Finding the LSE for the Linear Model

For the linear model the LS cost is: $J(\boldsymbol{\theta}) = \sum_{n=0}^{N-1}\left(x[n] - s[n;\boldsymbol{\theta}]\right)^2$

$$= \left(\mathbf{x} - \mathbf{H}\boldsymbol{\theta}\right)^T\left(\mathbf{x} - \mathbf{H}\boldsymbol{\theta}\right)$$

Now, to minimize, first expand:

$$J(\boldsymbol{\theta}) = \mathbf{x}^T\mathbf{x} - \mathbf{x}^T\mathbf{H}\boldsymbol{\theta} - \boldsymbol{\theta}^T\mathbf{H}^T\mathbf{x} + \boldsymbol{\theta}^T\mathbf{H}^T\mathbf{H}\boldsymbol{\theta}$$

$$= \mathbf{x}^T\mathbf{x} - 2\mathbf{x}^T\mathbf{H}\boldsymbol{\theta} + \boldsymbol{\theta}^T\mathbf{H}^T\mathbf{H}\boldsymbol{\theta}$$

Scalar = scalar^T So…
$\boldsymbol{\theta}^T\mathbf{H}^T\mathbf{x} = (\boldsymbol{\theta}^T\mathbf{H}^T\mathbf{x})^T = \mathbf{x}^T\mathbf{H}\boldsymbol{\theta}$

Now setting $\dfrac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathbf{0}$ gives $-2\mathbf{H}^T\mathbf{x} + 2\mathbf{H}^T\mathbf{H}\hat{\boldsymbol{\theta}} = \mathbf{0}$

$$\Rightarrow \quad \mathbf{H}^T\mathbf{H}\hat{\boldsymbol{\theta}} = \mathbf{H}^T\mathbf{x}$$

Called the "LS Normal Equations"

Because **H** is full rank we know that $\mathbf{H}^T\mathbf{H}$ is invertible:

$$\Rightarrow \quad \hat{\boldsymbol{\theta}}_{LS} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{x} \quad \Rightarrow \quad \hat{\mathbf{s}}_{LS} = \mathbf{H}\hat{\boldsymbol{\theta}}_{LS} = \mathbf{H}\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{x}$$

Comparing the Linear LSE to Other Estimates

Model	Estimate
$\mathbf{x} = \mathbf{H}\boldsymbol{\theta} + \mathbf{e}$ No Probability Model Needed	$\hat{\boldsymbol{\theta}}_{LS} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{x}$
$\mathbf{x} = \mathbf{H}\boldsymbol{\theta} + \mathbf{w}$ PDF Unknown, White	$\hat{\boldsymbol{\theta}}_{BLUE} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{x}$
$\mathbf{x} = \mathbf{H}\boldsymbol{\theta} + \mathbf{w}$ PDF Gaussian, White	$\hat{\theta}_{ML} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{x}$
$\mathbf{x} = \mathbf{H}\boldsymbol{\theta} + \mathbf{w}$ PDF Gaussian, White	$\hat{\boldsymbol{\theta}}_{MVU} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{x}$

If you assume Gaussian & apply these… BUT you are WRONG… you at least get the LSE!

The LS Cost for Linear LS

For the linear LS problem…

what is the resulting LS cost for using $\hat{\boldsymbol{\theta}}_{LS} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{x}$?

$$J_{\min} = \left(\mathbf{x} - \mathbf{H}\hat{\boldsymbol{\theta}}_{LS}\right)^T\left(\mathbf{x} - \mathbf{H}\hat{\boldsymbol{\theta}}_{LS}\right) = \left(\mathbf{x} - \mathbf{H}\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{x}\right)^T\left(\mathbf{x} - \mathbf{H}\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{x}\right)$$

Properties of Transpose

$$= \left(\mathbf{x}^T - \mathbf{x}^T\mathbf{H}\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\right)\left(\mathbf{x} - \mathbf{H}\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{x}\right)$$

Factor out ***x****'s*

$$= \mathbf{x}^T\underbrace{\left(\mathbf{I} - \mathbf{x}^T\mathbf{H}\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\right)\left(\mathbf{I} - \mathbf{H}\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\right)}_{=\left(\mathbf{I}-\mathbf{H}\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\right)}\mathbf{x}$$

Easily Verified!

Note: if **AA** = **A** then **A** is called idempotent

$$J_{\min} = \mathbf{x}^T\left(\mathbf{I} - \mathbf{H}\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\right)\mathbf{x}$$

⇒

$$J_{\min} = \mathbf{x}^T\mathbf{x} - \mathbf{x}^T\mathbf{H}\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{x}$$

⇒

$$0 \le J_{\min} \le \|\mathbf{x}\|^2$$

Weighted LS for Linear LS

Recall: de-emphasize bad samples' importance in the sum of squares:

$$J(\boldsymbol{\theta}) = \sum_{n=0}^{N-1} w_n \left(x[n] - s[n;\boldsymbol{\theta}]\right)^2$$

For the linear LS case we get: $J(\theta) = (\mathbf{x} - \mathbf{H}\boldsymbol{\theta})^T \mathbf{W} (\mathbf{x} - \mathbf{H}\boldsymbol{\theta})$

Diagonal Matrix

Minimizing the weighted LS cost gives:

$$\hat{\boldsymbol{\theta}}_{WLS} = \left(\mathbf{H}^T \mathbf{W} \mathbf{H}\right)^{-1} \mathbf{H}^T \mathbf{W} \mathbf{x}$$

$$J_{\min} = \mathbf{x}^T \left(\mathbf{W} - \mathbf{W}\mathbf{H}\left(\mathbf{H}^T \mathbf{W} \mathbf{H}\right)^{-1} \mathbf{H}^T \mathbf{W} \right) \mathbf{x}$$

Note: Even though there is no true LS-based reason… many people use an inverse cov matrix as the weight: $\mathbf{W} = \mathbf{C}_{\mathbf{x}}^{-1}$

This makes WLS look like BLUE!!!!

8.5 Geometry of Linear LS

- Provides different derivation
- Enables new versions of LS

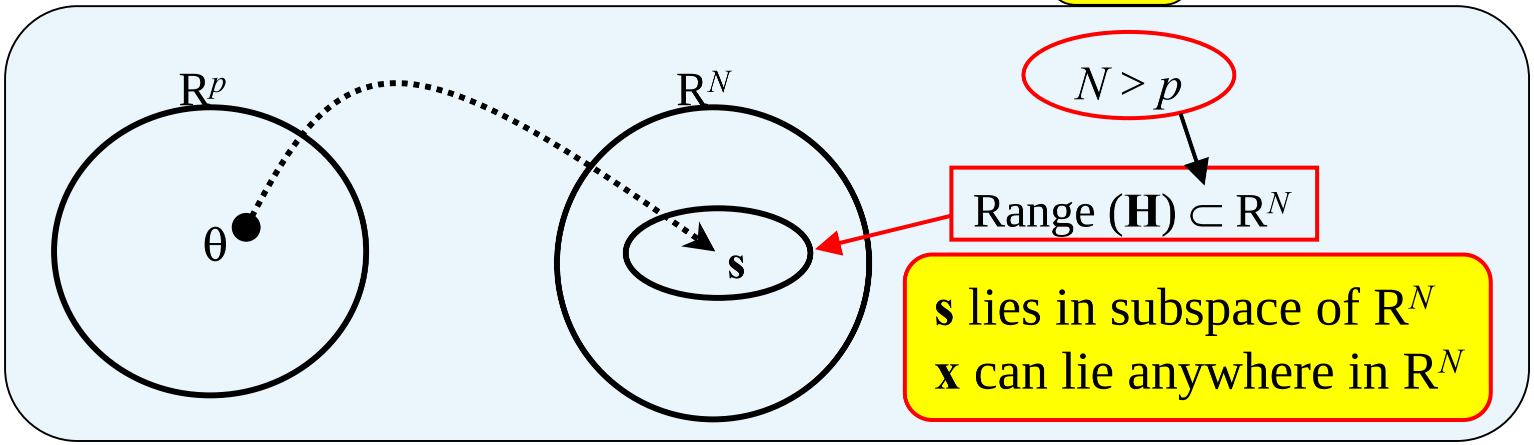
– Order Recursive
– Sequential

Recall the LS Cost to be minimized: $J(\boldsymbol{\theta}) = (\mathbf{x} - \mathbf{H}\boldsymbol{\theta})^T(\mathbf{x} - \mathbf{H}\boldsymbol{\theta}) = \|\mathbf{x} - \underbrace{\mathbf{H}\boldsymbol{\theta}}_{\hat{\mathbf{s}}}\|^2$

Thus, LS minimizes the length of the error vector between the data and the signal estimate: $\boldsymbol{\varepsilon} = \mathbf{x} - \hat{\mathbf{s}}$

But… For Linear LS we have $\mathbf{s} = \mathbf{H}\boldsymbol{\theta} = \sum_{i=1}^{p} \theta_i \mathbf{h}_i \qquad \mathbf{H} = [\mathbf{h}_1 \quad \mathbf{h}_2 \quad \cdots \quad \mathbf{h}_p]$

$N \times p$

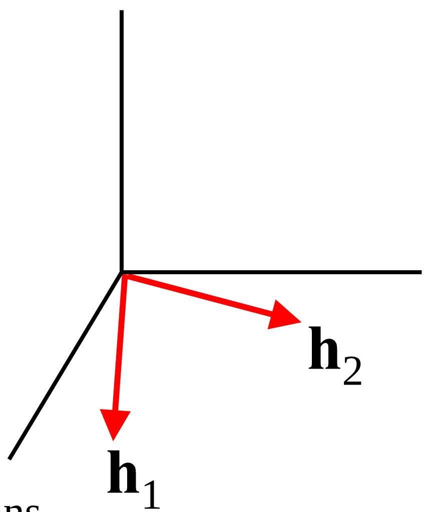


LS Geometry Example $N = 3$ $p = 2$

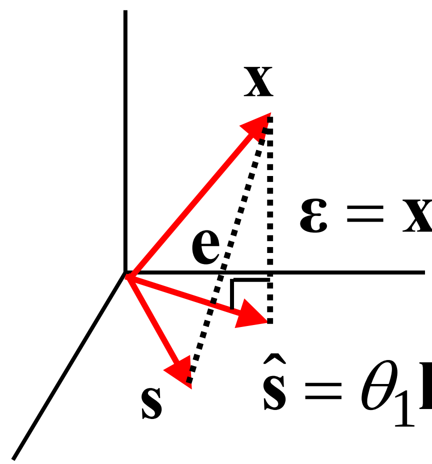
Notation a bit different from the book

$$\mathbf{x} = \mathbf{s} + \mathbf{e}$$

"noise" takes **s** out of Range(**H**) and into R^N



H columns lie in this plane = "subspace" spanned by the columns of **H** = S^2
(S^p in general)



$$\boldsymbol{\varepsilon} = \mathbf{x} - \hat{\mathbf{s}}$$

$$\boldsymbol{\varepsilon} \perp \mathbf{h}_i$$

$$\hat{\mathbf{s}} = \theta_1 \mathbf{h}_1 + \theta_2 \mathbf{h}_2$$

LS Orthogonality Principle ★★

The LS error vector must be ⊥ to all columns of H

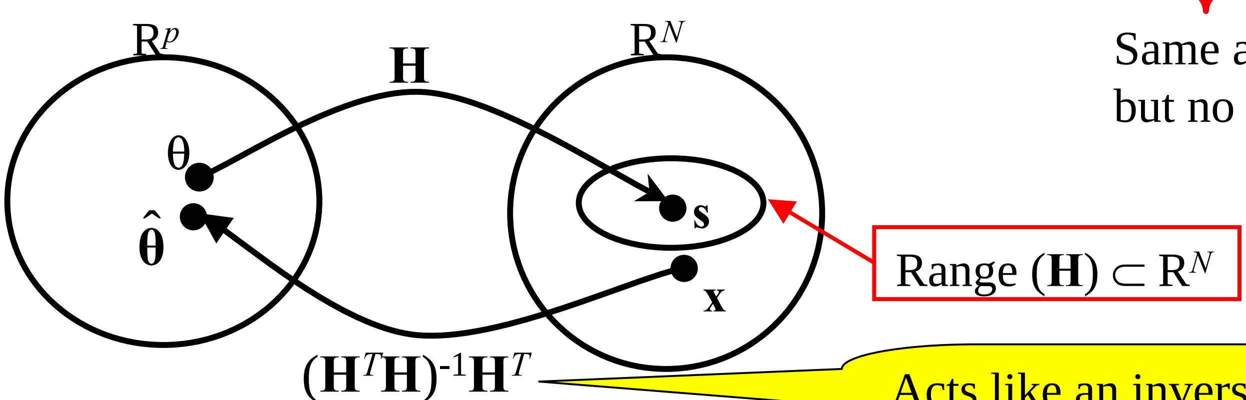
$$\Rightarrow \quad \boldsymbol{\varepsilon}^T \mathbf{H} = \mathbf{0}^T \quad \text{or} \quad \mathbf{H}^T \boldsymbol{\varepsilon} = \mathbf{0}$$

Can use this property to derive the LS estimate:

$$\mathbf{H}^T \boldsymbol{\varepsilon} = \mathbf{0} \quad \Rightarrow \quad \mathbf{H}^T (\mathbf{x} - \mathbf{H}\boldsymbol{\theta}) = \mathbf{0}$$

$$\Rightarrow \quad \mathbf{H}^T \mathbf{H}\boldsymbol{\theta} = \mathbf{H}^T \mathbf{x} \quad \Rightarrow \quad \hat{\boldsymbol{\theta}}_{LS} = \left(\mathbf{H}^T \mathbf{H}\right)^{-1} \mathbf{H}^T \mathbf{x}$$

Same answer as before…
but no derivatives to worry about!



Acts like an inverse from R^N back to R^p… called pseudo-inverse of **H**

LS Projection Viewpoint

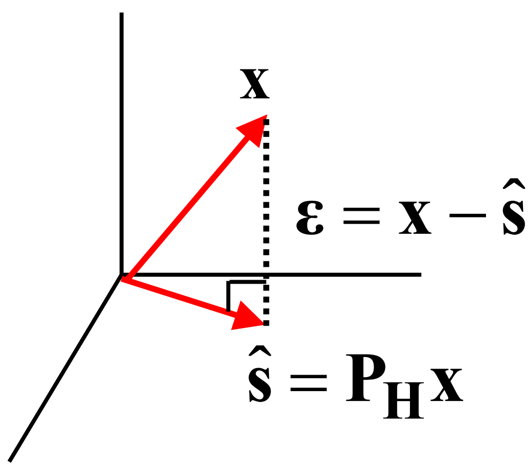
From the R^3 example earlier… we see that $\hat{\mathbf{s}}$ must lie "right below" $\mathbf{x}$

$\hat{\mathbf{s}}$ = "Projection" of $\mathbf{x}$ onto Range($\mathbf{H}$)

(Recall: Range($\mathbf{H}$) = subspace spanned by columns of $\mathbf{H}$)

From our earlier results we have: $$\hat{\mathbf{s}} = \mathbf{H}\hat{\boldsymbol{\theta}}_{LS} = \underbrace{\left[\mathbf{H}\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\right]}_{\triangleq \mathbf{P}_\mathbf{H}}\mathbf{x}$$

"Projection Matrix onto Range($\mathbf{H}$)"



Aside on Projections

If something is "on the floor"… its projection onto the floor = itself!

$$\text{if } \mathbf{z} \in \text{Range}(\mathbf{H}), \text{ then } \mathbf{P_H z} = \mathbf{z}$$

Now… for a given $\mathbf{x}$ in the full space… $\mathbf{P_H x}$ is already in Range($\mathbf{H}$)

… so $\mathbf{P_H}(\mathbf{P_H x}) = \mathbf{P_H x}$

Thus… for any projection matrix $\mathbf{P_H}$ we have: $\mathbf{P_H P_H} = \mathbf{P_H}$

$$\mathbf{P_H^2} = \mathbf{P_H}$$

Projection Matrices are Idempotent

Note also that the projection onto Range($\mathbf{H}$) is symmetric:

$$\mathbf{P_H} = \mathbf{H}\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T$$

Easily Verified

What Happens w/ Orthonormal Columns of H

Recall the general Linear LS solution: $\hat{\boldsymbol{\theta}}_{LS} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{x}$

where

$$\mathbf{H}^T\mathbf{H} = \begin{bmatrix} \langle \mathbf{h}_1, \mathbf{h}_1 \rangle & \langle \mathbf{h}_1, \mathbf{h}_2 \rangle & \cdots & \langle \mathbf{h}_1, \mathbf{h}_p \rangle \\ \langle \mathbf{h}_2, \mathbf{h}_1 \rangle & \langle \mathbf{h}_2, \mathbf{h}_2 \rangle & \cdots & \langle \mathbf{h}_2, \mathbf{h}_p \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \mathbf{h}_p, \mathbf{h}_1 \rangle & \langle \mathbf{h}_p, \mathbf{h}_2 \rangle & \cdots & \langle \mathbf{h}_p, \mathbf{h}_p \rangle \end{bmatrix}$$

If the columns of **H** are orthonormal then $\langle \mathbf{h}_i, \mathbf{h}_j \rangle = \delta_{ij} \Rightarrow \mathbf{H}^T\mathbf{H} = \mathbf{I}$

$$\boxed{\hat{\boldsymbol{\theta}}_{LS} = \mathbf{H}^T\mathbf{x}}$$

Easy!! No Inversion Needed!!

Recall Vector Space Ideas with ON Basis!!

Geometry with Orthonormal Columns of H

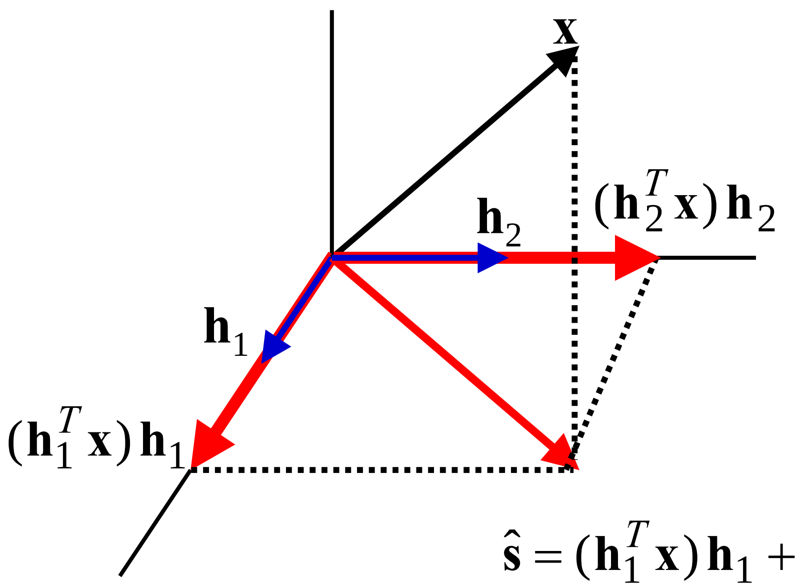
Re-write this LS solution as: $\hat{\theta}_i = \mathbf{h}_i^T \mathbf{x}$

Inner Product Between i^{th} Column and Data Vector

Then we have:

$$\hat{\mathbf{s}} = \mathbf{H}\hat{\boldsymbol{\theta}} = \sum_{i=1}^{p} \hat{\theta}_i \mathbf{h}_i = \sum_{i=1}^{p} \underbrace{(\mathbf{h}_i^T \mathbf{x})\,\mathbf{h}_i}$$

Projection of x onto $\mathbf{h}_i$ axis



$$\hat{\mathbf{s}} = (\mathbf{h}_1^T \mathbf{x})\,\mathbf{h}_1 + (\mathbf{h}_2^T \mathbf{x})\,\mathbf{h}_2$$

When the columns of H are ⊥ we can first find the projection onto each 1-D subspace independently, then add these independently derived results. <u>Nice</u>!