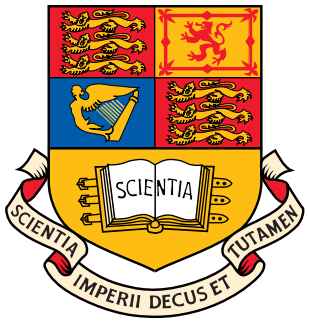
Statistical Signal Processing & Inference
The Method of Least Squares

Danilo Mandic

room 813, ext: 46271



Department of Electrical and Electronic Engineering
Imperial College London, UK

d.mandic@imperial.ac.uk, URL: www.commsp.ee.ic.ac.uk/∼mandic

Aims

- The concept of least squares estimation (LSE), a method which is indispensable in areas ranging from finance through to neuroscience
- Geometry of LS: The the orthogonality principle $\looparrowright$ signal and noise subspaces, ordinary least squares (OLS)
- Measurement space, basis functions, constrained least squares, order recursive least squares, nonlinear least squares, separable least squares
- Establish parallels with the ML estimation, BLUE, MVUE, and CRLB
- Move from block-based estimation to estimation based on streaming data: Sequential Least Squares (SLS), link with state space models
- Incorporating prior knowledge and domain knowledge: Weighted least squares, confidence levels in data samples
- Role of estimator memory $\looparrowright$ forgetting factor and online adaptation
- Practical applications (regression, Noise Canc. headphones, finance, ...)

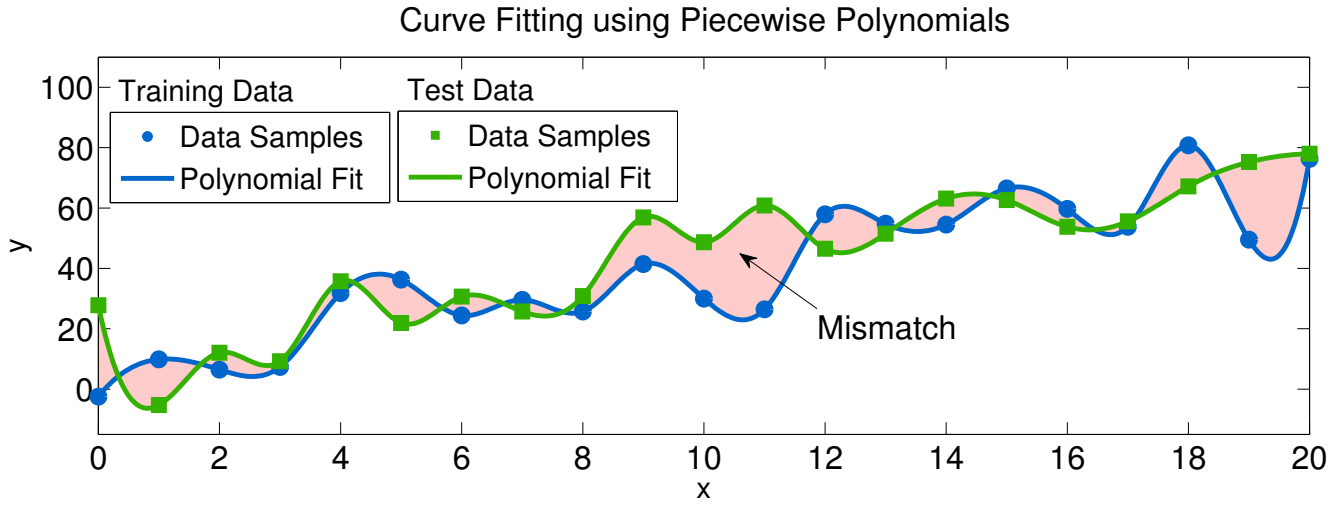
The method of Least Squares

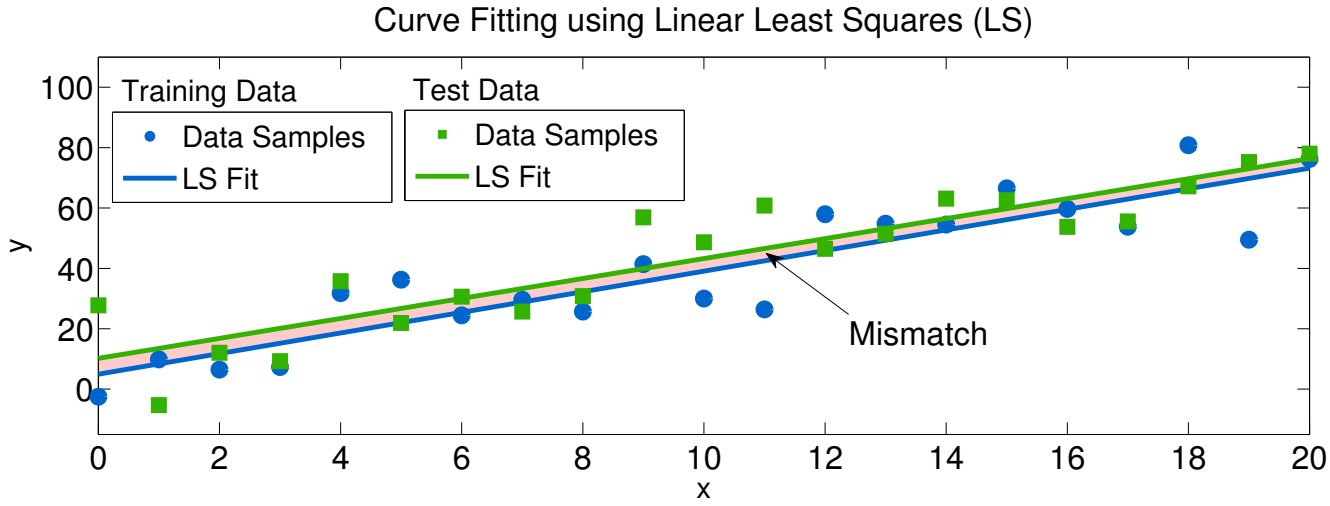
This class of estimators has, generally, no optimality properties

- But, do we necessarily require optimality $\looparrowright$ an optimal estimator may be mathematically intractable or computationally too complex
- Makes good sense for many practical problems $\looparrowright$ this dates back to Gauss who in 1795 introduced the method to study planetary motions
- **LS is not statistically based** $\looparrowright$ no probabilistic assumptions are made about the data, no need for knowledge of a pdf or second order stats
- We only need to assume a deterministic signal model
- Usually easy to implement, either in a block–based or sequential manner $\looparrowright$ this amounts to the minimisation of a quadratic cost function
- Within the (LS) approach we attempt to minimise the squared difference between the observed data and the assumed model of noiseless data
- Rigorous statistical performance cannot be assessed without some specific assumptions about probabilistic structure in the data

Motivation: A simpler model often generalises better

Consider two models for $x[n] = A + Bn + q[n]$ (q $\rightsquigarrow$ noise)





← consistent

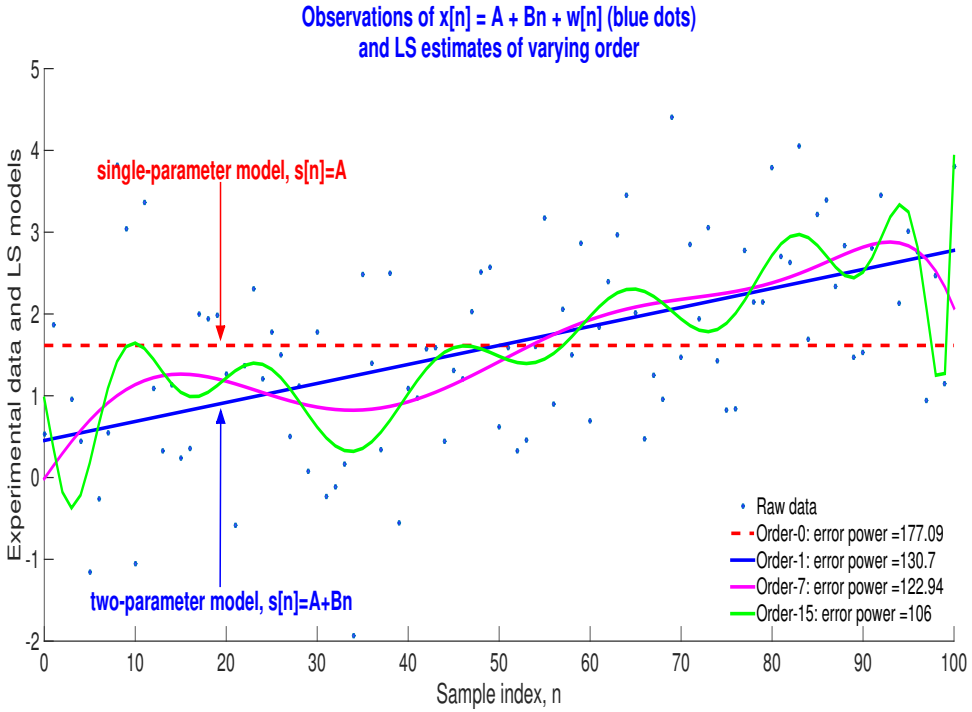
☞ **Model is more useful than an exact fit!**

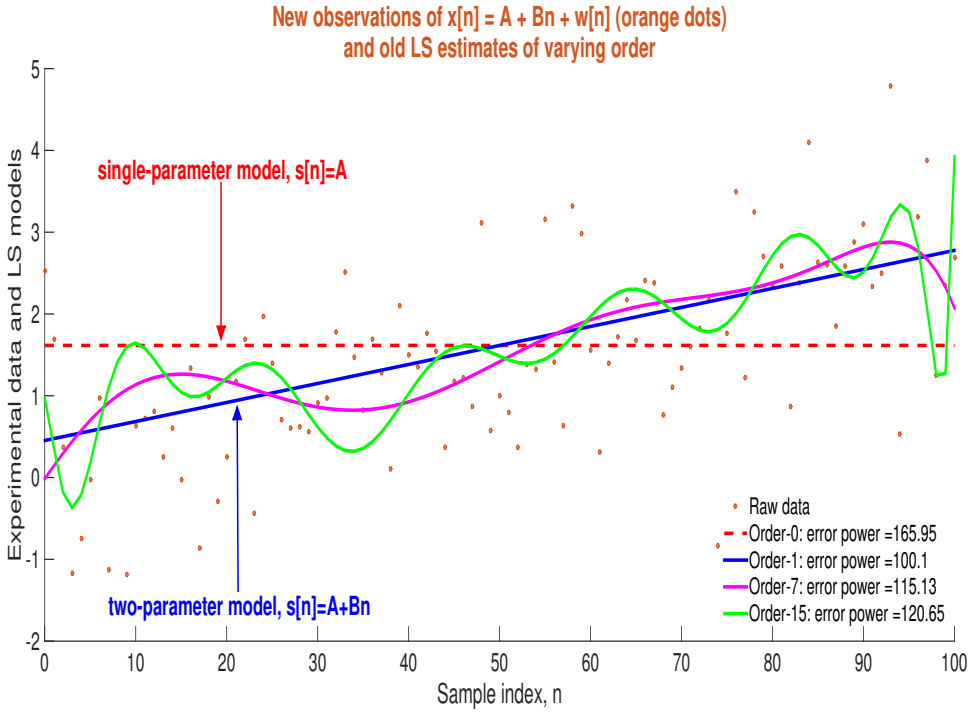
But, careful with over–fitting (see Appendix 1)

Least_squares_overfitting.m Least_Squares_Order_Selection_Interactive.m

Data considered was a noisy line: $x[n] = A + Bn + q[n], \quad q \sim \mathcal{N}(0, \sigma^2)$

☞ **So, the correct data model was LS of order–1** (blue line in the figures below)





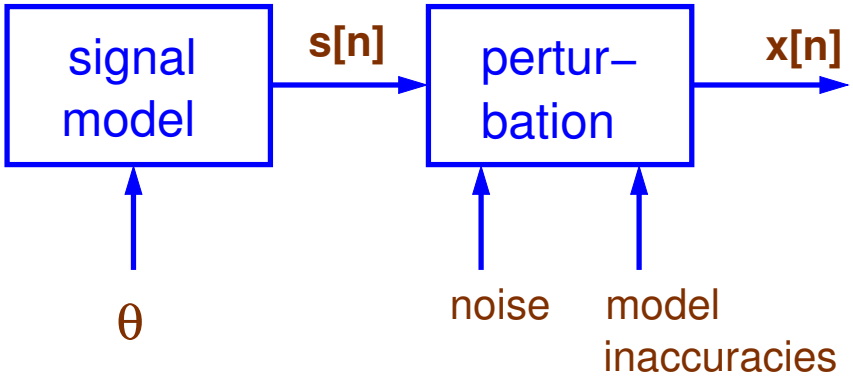
☞ Order–7 and Order–15 Least Squares (LS) fits to the data gave a lower "within–sample" **training error** power than the correct Order–1 fit (122 and 106 versus 130) (left panel)

☞ But this leads to over–fitting, i.e. worse extrapolation (inference) on "out-of-sample" **test data** from the same generative model (120 for Order–15 vs 100 for Order-1) (right panel)

Data model and the Least Squares Error (LSE) criterion

no probabilistic assumptions made about the data

The signal $s[n]$ is assumed to be generated by a deterministic model which depends upon an unknown parameter θ or a vector parameter $\boldsymbol{\theta}$.



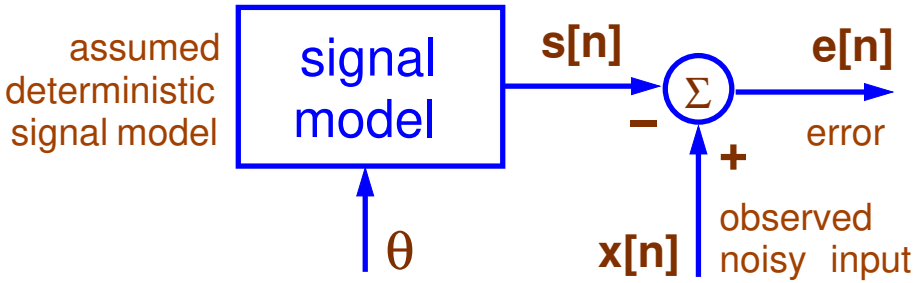
Least squares data model

The observed signal $x[n]$ is subject to:

- external noise $q[n]$
- model inaccuracies

No probabilistic assumptions!

Only signal model assumed $\leadsto$ wide range of applications.



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

LSE objective: $\min_{\theta} J(\theta) \equiv \arg\min_{\theta} J(\theta)$

- The LS estimator of θ finds that value of θ which makes the model output $s[n]$ closest to the observed data $x[n]$;
- The closeness is measured by **LS error criterion** (error power) $J(e^2) = J(\theta)$.

Example 1: DC Level in WGN

Our old example: DC level in WGN (*cf.* MLE needs an assumed pdf)

Data model: $s[n;\theta] = A$

Measurement model: $x[n] = s[n;\theta] + q[n] = A + q[n], \quad q[n] \not\hookrightarrow$ any noise

LSE formulation:

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

LSE solution:

set the derivative to zero $\quad \dfrac{dJ(A)}{dA} = -2 \sum_{n=0}^{N-1} (x[n] - A) = 0$

the LS estimator : $\quad \hat{A} = \dfrac{1}{N} \sum_{n=0}^{N-1} x[n]$ is a MVU estimator

We cannot claim optimality in the MVU sense, except for the Gaussian noise $q \sim \mathcal{N}(0, \sigma^2)$. All we can say is that the LSE estimator minimises the sum of squared errors (error power).

☞ Still, this leads to a very powerful and practically useful class of estimators.

The method of Least Squares is very convenient
how do we use it in practice?

1. **Problem with signal mean.** If the noise is not zero–mean, then the sample mean estimator actually models $\boldsymbol{x[n] = A + q[n] + q'[n]}$

$q[n] \sim$ nonzero mean noise $\quad q'[n] \sim$ zero mean noise $\quad \rightarrow \quad E\{x[n]\} = A + E\{q[n]\}$

☞ The presence of non-zero mean noise $q[n]$ **biases** the LSE estimator, as the LS approach assumes that the observed data are composed of a **signal** (described by a deterministic model) and **zero mean** noise.

2. **Nonlinear signal model.** For instance $s[n] = \cos 2\pi f_0 n$, where the frequency f_0 is to be estimated. The LSE criterion (TSE may help)

$$J(f_0) = \sum_{n=0}^{N-1} (x[n] - \cos 2\pi f_0 n)^2$$

is highly nonlinear in $f_0 \rightarrow$ closed form minimisation is impossible.

- However, for $s[n] = A \cos 2\pi f_0 n$, if f_0 is known and A is unknown, then we can still use the LS method, as A is "linear in the data"
- **Separable least squares.** When estimating both A and f_0, the error is **quadratic in A** and **non-quadratic in** $f_0 \rightsquigarrow$ minimize J wrt A for a given f_0, reducing to minimising J over f_0 only (see Slide 23).

Geometric interpretation & Example 2: Fourier analysis

Cost function (error power): $J(\theta) = \sum_{n=0}^{N-1} e^2[n] = \sum_{n=0}^{N-1} (\underbrace{x[n] - s[n]}_{e[n]})^2 = \mathbf{e}^T\mathbf{e}$

Consider a signal model $s[n] = a\cos 2\pi f_0 n + b \sin 2\pi f_0 n$, with f_0 known.

Task: Determine the unknown parameters, that is, the amplitudes a and b.

Solution: With f_0 known and $\boldsymbol{\theta} = [a, b]^T$, we have

$$\underbrace{\begin{bmatrix} s[0] \\ s[1] \\ \vdots \\ s[N-1] \end{bmatrix}}_{\mathbf{s}} = \begin{bmatrix} 1 & 0 \\ \cos 2\pi f_0 & \sin 2\pi f_0 \\ \vdots & \vdots \\ \underbrace{\cos 2\pi f_0[N-1]}_{\mathbf{h}_1} & \underbrace{\sin 2\pi f_0[N-1]}_{\mathbf{h}_2} \end{bmatrix} \underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_{\boldsymbol{\theta}} = \underbrace{[\mathbf{h}_1 \,|\, \mathbf{h}_2]}_{\mathbf{H}} \underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_{\boldsymbol{\theta}}$$

☞ We must assume a full rank of $\mathbf{H}$, otherwise multiple $\boldsymbol{\theta}$ map to the same $\mathbf{s}$

$\mathbf{s} = a\,\mathbf{h}_1 + b\,\mathbf{h}_2$ (linear combination of $\mathbf{h}_1$ & $\mathbf{h}_2$); error $\mathbf{e} = \mathbf{x} - \mathbf{s} = \mathbf{x} - \mathbf{H}\boldsymbol{\theta}$

In general, $\mathbf{s} = \mathbf{H}\,\boldsymbol{\theta} \;\Leftrightarrow\; \mathbf{s} = \underbrace{[\mathbf{h}_1 \,|\, \cdots \,|\, \mathbf{h}_p]}_{\text{columns of } \mathbf{H}} [\theta_1, \ldots, \theta_p]^T = \sum_{i=1}^{p} \theta_i \mathbf{h}_i$

☞ **Signal model is a linear combination of "signal space" basis vectors $\{\mathbf{h}_1, \ldots, \mathbf{h}_p\}$**

and the Least Squares (LS) cost: $J(\boldsymbol{\theta}) = J(\mathbf{e}^T\mathbf{e}) = \left(\mathbf{x} - \mathbf{H}\boldsymbol{\theta}\right)^T \left(\mathbf{x} - \mathbf{H}\boldsymbol{\theta}\right)$

Geometric interpretation ↬ continued

Recall that the signal vector $\mathbf{s}$ is a linear combination of the columns of $\mathbf{H}$

This can be rewritten in a more elegant form. Assume $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_p]^T$.

Recall that the Euclidean length $\| \cdot \|_2$ of a general $N \times 1$ vector $\mathbf{q} = [q_1, q_2, \ldots, q_N]^T \in \mathbb{R}^{N \times 1}$ is given by

$$\| \mathbf{q} \|_2 = \sqrt{\sum_{i=1}^{N} q_i^2} = \sqrt{\mathbf{q}^T \mathbf{q}} = \sqrt{< \mathbf{q}, \mathbf{q} >}$$

Then, for $\mathbf{q} = \mathbf{a} - \mathbf{b}$, the square distance between $\mathbf{a} = \mathbf{x}$ and $\mathbf{b} = \mathbf{H}\boldsymbol{\theta}$ is

$$J(\boldsymbol{\theta}) = (\mathbf{x} - \mathbf{H}\boldsymbol{\theta})^T (\mathbf{x} - \mathbf{H}\boldsymbol{\theta}) = \|\mathbf{x} - \mathbf{H}\boldsymbol{\theta}\|_2^2 = \|\mathbf{x} - \sum_{i=1}^{p} \theta_i \mathbf{h}_i\|_2^2$$

☞ The LSE attempts **to minimise the square of the distance** between the measured (noisy) data vector, $\mathbf{x}$, and the signal estimate, $\hat{\mathbf{s}}$, given by

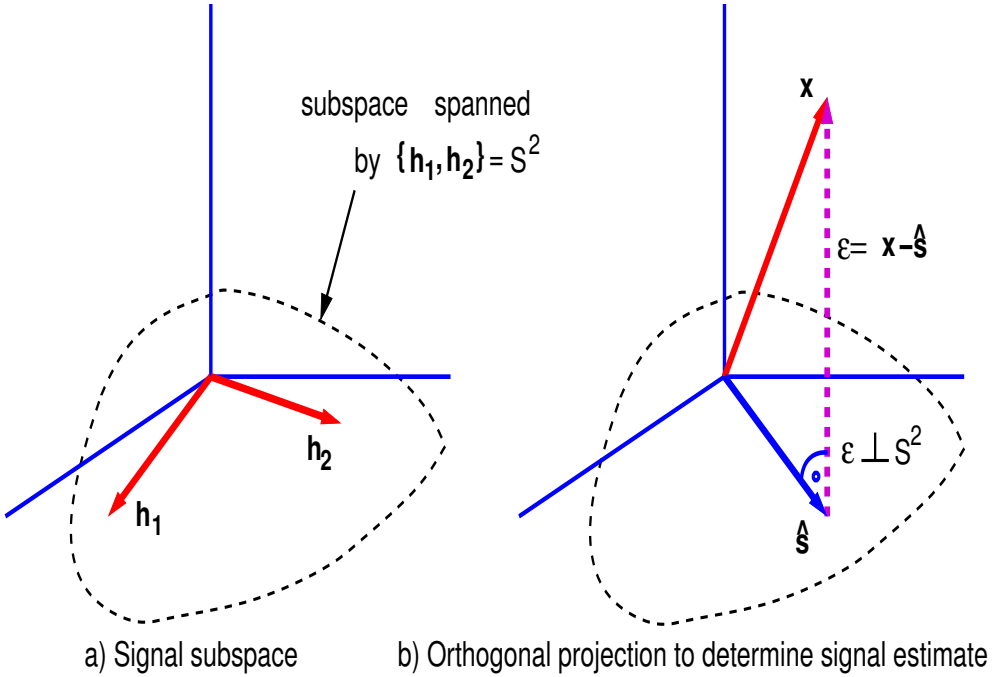
$$\hat{\mathbf{s}} = \sum_{i=1}^{p} \hat{\theta}_i \mathbf{h}_i$$

☞ The signal estimate, $\hat{\mathbf{s}}$, resides in a p–dimensional subspace, S^p, spanned by the columns $\mathbf{h}_1, \ldots, \mathbf{h}_p$ of $\mathbf{H}$ (range of $\mathbf{H}$). **For LS estimation, $N > p$.**

Geometry of LSE: Vector space projections

signal dimension is lower than measurement dimension (signal lives in a subspace)

☞ The observation, $\mathbf{x} \in \mathbb{R}^{N\times 1}$, resides in $\mathbb{R}^N$, while signal vector, $\mathbf{s}$, lies in a p-dimen. subspace $S^p \subset \mathbb{R}^N$. For example, for N=3 and $p = 2$, we have:



a) Signal subspace b) Orthogonal projection to determine signal estimate

⊛ The vector in S^2 which is closest to $\mathbf{x}$ in the Euclidean sense is the component $\hat{\mathbf{s}} \in S^2$, that is, the **orthogonal projection** of $\mathbf{x}$ onto S^2, $\hat{\mathbf{s}} = \mathbf{P}\mathbf{x}$, $\mathbf{P} \hookrightarrow$ projection matrix.

⊛ Two vectors are orthogonal if their scalar product $\mathbf{x}^T\mathbf{y} = 0$

⊛ Therefore, to determine $\hat{\mathbf{s}}$, we use so-called **orthogonality condition:**

$$\boldsymbol{\varepsilon} = (\mathbf{x}-\hat{\mathbf{s}}) \perp \mathbf{H} \Leftrightarrow (\mathbf{x}-\hat{\mathbf{s}}) \perp S^2$$

$$\boldsymbol{\varepsilon} \perp S^2 \Leftrightarrow \boldsymbol{\varepsilon} \perp \mathbf{h}_1 \;\&\; \boldsymbol{\epsilon} \perp \mathbf{h}_2 \qquad (\mathbf{a}): \quad \left(\mathbf{x}-\hat{\mathbf{s}}\right) \perp \mathbf{h}_1 \;\Rightarrow\; \left(\mathbf{x}-\hat{\mathbf{s}}\right)^T \mathbf{h}_1 = 0$$

$$(\mathbf{b}): \quad \left(\mathbf{x}-\hat{\mathbf{s}}\right) \perp \mathbf{h}_2 \;\Rightarrow\; \left(\mathbf{x}-\hat{\mathbf{s}}\right)^T \mathbf{h}_2 = 0$$

Finally: LS solution for our Fourier example

Observe: $\hat{\mathbf{s}}$ = projection of $\mathbf{x}$ onto Range($\mathbf{H}$)

Letting $\hat{\mathbf{s}} = \hat{\theta}_1\mathbf{h_1} + \hat{\theta}_2\mathbf{h_2} = \mathbf{H}\hat{\boldsymbol{\theta}}$ (here $\mathbf{H_{N\times 2}}, \hat{\boldsymbol{\theta}}_{\mathbf{2\times 1}}$)

and based on the orthogonality conditions (a) and (b) (previous slide)

$$\begin{aligned} \left(\mathbf{x} - \hat{\theta}_1\mathbf{h_1} - \hat{\theta}_2\mathbf{h_2}\right)^{\mathbf{T}}\mathbf{h_1} &= 0 \quad \equiv \quad \boldsymbol{\varepsilon}^T\mathbf{h}_1 = 0 \\ \left(\mathbf{x} - \hat{\theta}_1\mathbf{h_1} - \hat{\theta}_2\mathbf{h_2}\right)^{\mathbf{T}}\mathbf{h_2} &= 0 \quad \equiv \quad \boldsymbol{\varepsilon}^T\mathbf{h}_2 = 0 \end{aligned}$$

Since $\mathbf{H} = [\mathbf{h}_1 \,|\, \mathbf{h}_2]$, $\hat{\boldsymbol{\theta}} = [\hat{a},\ \hat{b}]^T$, and $\boldsymbol{\varepsilon} = \mathbf{x} - \mathbf{H}\hat{\boldsymbol{\theta}}$, the above conditions can be combined into a vector/matrix form as (use $\mathbf{u}^T\mathbf{v} = \mathbf{v}^T\mathbf{u}$)

$$\boldsymbol{\varepsilon}^T\mathbf{H} = \mathbf{0}^T \quad \text{so that} \quad \mathbf{H}^T\boldsymbol{\varepsilon} = \mathbf{0} \quad \text{and} \quad \mathbf{H}^T(\mathbf{x} - \mathbf{H}\hat{\boldsymbol{\theta}}) = \mathbf{0}$$

☞ The equivalent system $\mathbf{H}^T\mathbf{H}\,\hat{\boldsymbol{\theta}} = \mathbf{H}^T\mathbf{x}$ is called **the LS normal equations.**

Solve for the unknown vector parameter, $\hat{\boldsymbol{\theta}}$, to arrive at the general **Least Squares Estimate (LSE)** for any dimension, p, in the form

$$\hat{\boldsymbol{\theta}}_{ls} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{x} \qquad (\mathbf{H}_{N\times p}, \hat{\boldsymbol{\theta}}_{p\times 1})$$

In our Fourier example, $\hat{\boldsymbol{\theta}}_{ls} = [\hat{a}, \hat{b}]^T$ is 2×1-dimensional, and $\mathbf{H}$ is $(N \times 2)$-dim.

Example 2: Fourier analysis ↬ continued (we have $\mathbf{H} = [\mathbf{h}_1|\mathbf{h}_2]$)

For more detail see Example 9 in Lecture 4 **(NB: power of $A\cos\omega n$ is $A^2/2$)**

For $f_0 = k/N$, with $k = 1, 2, \ldots, N/2-1$, and large N, the scalar product of the columns of the observation matrix $\mathbf{H}$ becomes (orthogonality)

$$\mathbf{h}_1^T\mathbf{h}_2 \;=\; \sum_{n=0}^{N-1} \cos\left(2\pi\frac{k}{N}n\right)\sin\left(2\pi\frac{k}{N}n\right) = 0 \quad\Leftrightarrow\quad \mathbf{h}_1 \perp \mathbf{h}_2 \quad \text{(orthogonal)}$$

$$\text{while} \qquad \mathbf{h}_1^T\mathbf{h}_1 = \frac{N}{2} \qquad\qquad \mathbf{h}_2^T\mathbf{h}_2 = \frac{N}{2} \qquad \text{(from power of cos and sin)}$$

Combining the above results gives $\mathbf{H}^T\mathbf{H} = \frac{N}{2}\mathbf{I}$ and therefore

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{x} = \frac{2}{N}\mathbf{H}^T\mathbf{x} = \begin{bmatrix} \frac{2}{N}\sum_{n=0}^{N-1} x[n]\cos(2\pi\frac{k}{N}n) \\ \frac{2}{N}\sum_{n=0}^{N-1} x[n]\sin(2\pi\frac{k}{N}n) \end{bmatrix}$$

☞ For orthonormal columns, $\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{x} = \mathbf{H}^T\mathbf{x}$ and $\hat{\mathbf{s}} = \mathbf{H}\hat{\boldsymbol{\theta}} = \mathbf{H}\mathbf{H}^T\mathbf{x}$

In general, the columns of $\mathbf{H}$ may not be orthogonal, and the signal estimate is obtained as (in a more complicated way, via projections)

$$\hat{\mathbf{s}} = \mathbf{H}\hat{\boldsymbol{\theta}} = \underbrace{\mathbf{H}\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T}_{\text{projection matrix } \mathbf{P}}\mathbf{x} = \mathbf{P}\mathbf{x}$$

(see Slide 24 and Appendix 8)

Summary: Computational advantages of having orthogonal representation bases (columns of $\mathbf{H}$)

For a general case of $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_p]^T$, the LS estimator finds

$$\hat{\boldsymbol{\theta}}_{ls} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{x} \quad \textit{where} \quad \mathbf{H} = [\mathbf{h}_1 \,|\, \mathbf{h}_2 \,|\, \cdots \,|\, \mathbf{h}_p]_{N\times p} \quad \textit{so that}$$

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

for orthonormal columns $\quad \langle \mathbf{h}_i, \mathbf{h}_j \rangle = \delta_{ij} \quad \Rightarrow \quad \mathbf{H}^T\mathbf{H} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} = \mathbf{I}$

In that case $\quad \hat{\boldsymbol{\theta}} = \mathbf{H}^T\mathbf{x} \quad$ and $\quad \hat{\mathbf{s}} = \mathbf{H}\hat{\boldsymbol{\theta}} = \mathbf{H}\mathbf{H}^T\mathbf{x}$

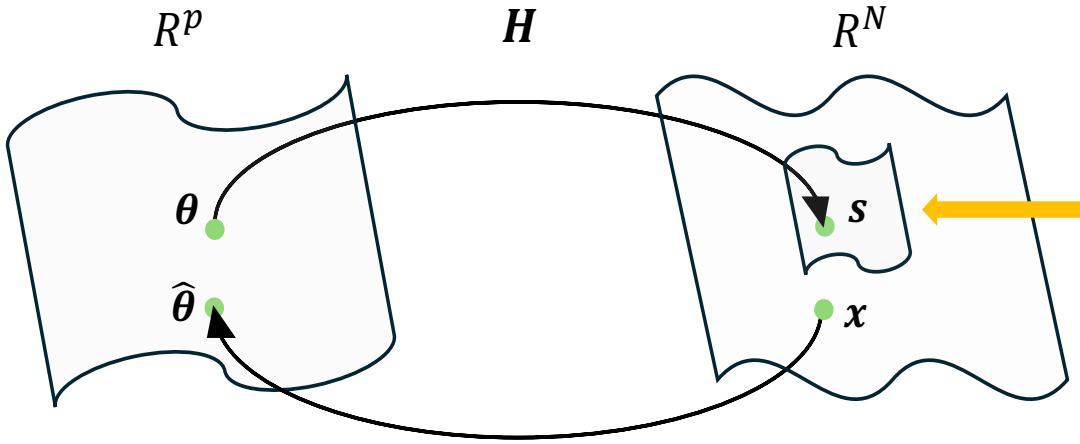
☞ **No inversion needed!** For every unknown parameter, θ_i, the inner product $\theta_i = \mathbf{h}_i^T\mathbf{x}$ represents a projection of observations $\mathbf{x}$ onto a column $\mathbf{h}_i$ of $\mathbf{H}$

Back to the geometry of Least Squares: General case

We know that in a general case, $\theta = [\theta_1, \ldots, \theta_p]^T$

In general $\quad \mathbf{s} = \theta_1\mathbf{h_1} + \theta_2\mathbf{h_2} + \cdots + \theta_p\mathbf{h_p} = \mathbf{H}\,\boldsymbol{\theta}$

Model: Matrix $\mathbf{H}$ multiplies **true** parameters, $\boldsymbol{\theta} \in \mathbb{R}^p$, to yield the signal $\mathbf{s}$.



$Range(\boldsymbol{H}) \subset R^N$, $N > p$

$\mathbf{s}$ lies in S^p, a subspace of $\mathbb{R}^N$

$\mathbf{x}$ can lie anywhere in $\mathbb{R}^N$

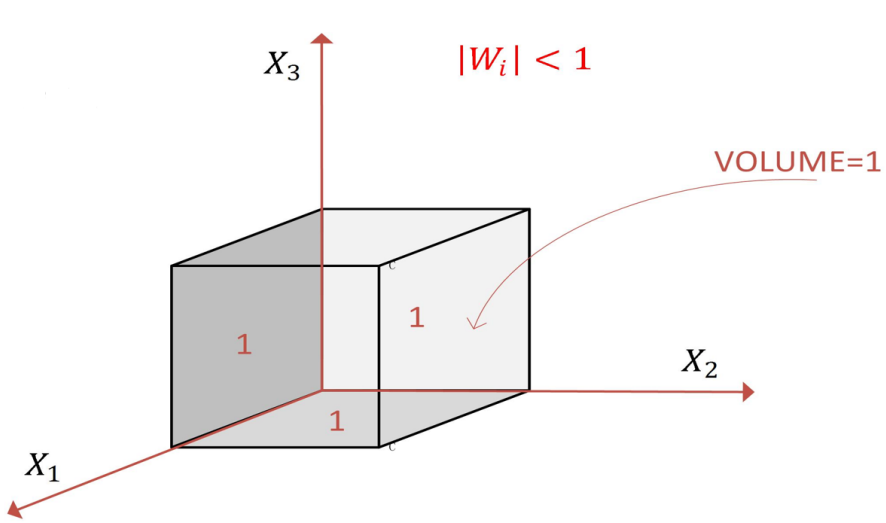
$(\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T$ ← Acts as an "inverse" from $\mathbb{R}^N$ back to $\mathbb{R}^p$

The **estimated** parameter vector, $\hat{\boldsymbol{\theta}} \in \mathbb{R}^p$ is obtained from the noisy observations, $\mathbf{x} = \mathbf{s} + \mathbf{q},\ \mathbf{x} \in \mathbb{R}^N$, in the form

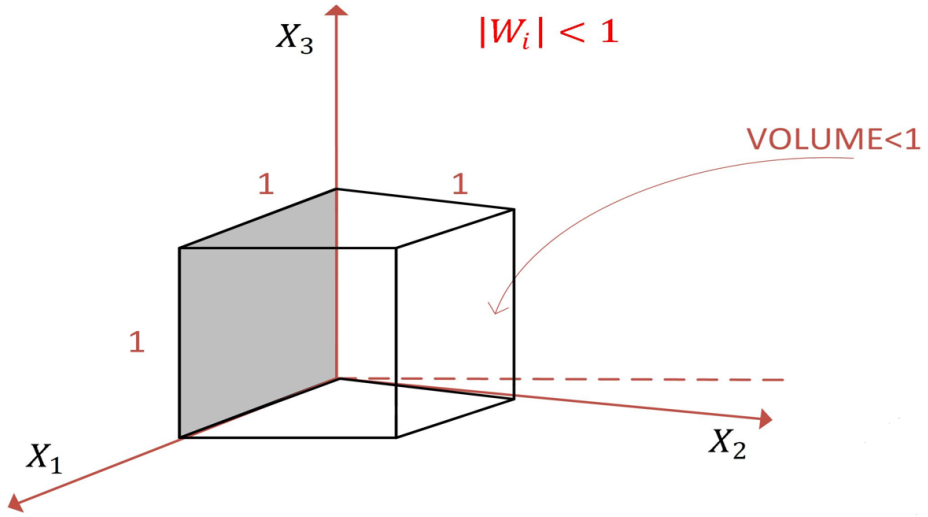
$$\hat{\boldsymbol{\theta}}_{ls} = \underbrace{\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T}_{\text{pseudoinverse of } \mathbf{H}}\mathbf{x} \qquad (\mathbf{H}_{N\times p}, \hat{\boldsymbol{\theta}}_{p\times 1})$$

Benefits of orthogonal basis functions (geometry)

Orthogonal basis



Non–orthogonal basis



- **Largest volume** is possible only for an orthogonal set of bases!
- This also greatly simplifies the maths related to the various estimation algorithms which rest upon vector space projections (see e.g. Slide 14).
- The best scenario is if we have orthonormal bases (orthogonal and unit length), as elaborated in the sequel.

This also justifies our preference for linear systems and orthogonal bases, wherever possible. (see also Appendix 5b)

Linear least squares in a nutshell

Assume a linear observation model $\mathbf{x} = \mathbf{H}\boldsymbol{\theta} + \mathbf{q}$. Then the **cost function**

$$J(\boldsymbol{\theta}) = \sum_{n=0}^{N-1} \big(x[n] - s[n,\theta]\big)^2 = \big(\mathbf{x} - \underbrace{\mathbf{H}\boldsymbol{\theta}}_{\mathbf{s}}\big)^T \big(\mathbf{x} - \underbrace{\mathbf{H}\boldsymbol{\theta}}_{\mathbf{s}}\big) \qquad (*)$$

$$= \mathbf{x}^T\mathbf{x} - 2\mathbf{x}^T\mathbf{H}\boldsymbol{\theta} + \boldsymbol{\theta}^T\mathbf{H}^T\mathbf{H}\boldsymbol{\theta} \qquad (\mathbf{H} \text{ is full rank})$$

The gradient of the cost function is then

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -2\mathbf{H}^T\mathbf{x} + 2\mathbf{H}^T\mathbf{H}\boldsymbol{\theta} = \mathbf{0}$$

1. The LSE estimator of $\boldsymbol{\theta}$ becomes $\hat{\boldsymbol{\theta}} = \big(\mathbf{H}^T\mathbf{H}\big)^{-1}\mathbf{H}^T\mathbf{x}$
2. The minimum LS cost is therefore (replace $\hat{\boldsymbol{\theta}}$ into $J(\boldsymbol{\theta})$ in (*) above)

$$J_{min} = J(\hat{\boldsymbol{\theta}}) = \mathbf{x}^T\big[\mathbf{I} - \mathbf{H}\underbrace{\big(\mathbf{H}^T\mathbf{H}\big)^{-1}\mathbf{H}^T}_{\hat{\boldsymbol{\theta}}}\big]\mathbf{x} = \mathbf{x}^T\big(\underbrace{\mathbf{x} - \underbrace{\mathbf{H}\hat{\boldsymbol{\theta}}}_{\hat{\mathbf{s}}}}_{\boldsymbol{\varepsilon}}\big) = \mathbf{x}^T\mathbf{x} - \mathbf{x}^T\mathbf{H}\hat{\boldsymbol{\theta}}$$

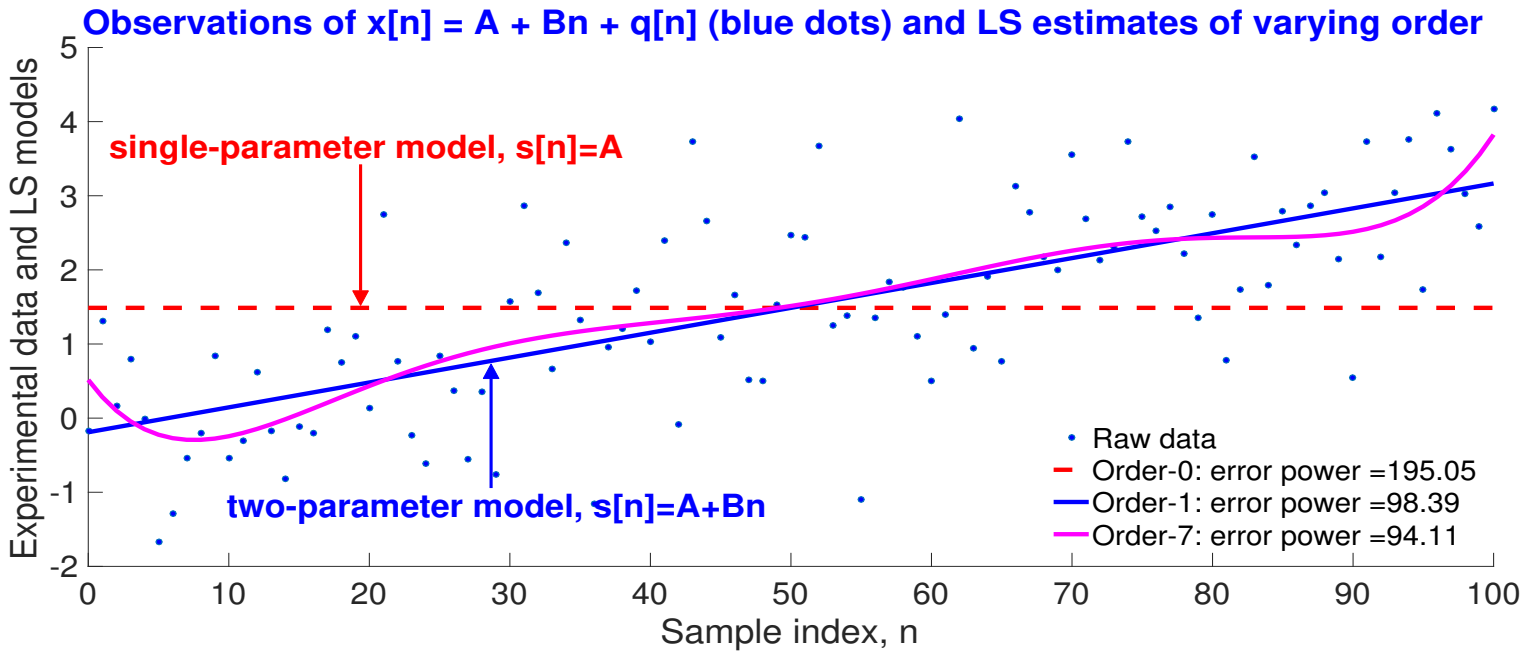
$\mathbf{x}^T\mathbf{x}$: power of $\mathbf{x}$; $\mathbf{x}^T\mathbf{H}\hat{\boldsymbol{\theta}}$: cross-corr$(\mathbf{x}, \hat{\mathbf{s}})$ $\approx$ power of $\hat{\mathbf{s}}$

☞ J_{min} represents the error power which is **not explained by** the model for $\hat{\mathbf{s}}$

Linear least squares in a nutshell, continued

Matlab: Least_Squares_Order_Selection_Interactive.m

- The LS approach can be interpreted as the problem of approximating a data vector $\mathbf{x} \in \mathbb{R}^N$ by another vector $\hat{\mathbf{s}}$ which is a linear combination of vectors $\{\mathbf{h}_1, \ldots, \mathbf{h}_p\}$ that lie in a p-dimensional subspace $S \in \mathbb{R}^p \subset \mathbb{R}^N$
- The problem is solved by choosing $\hat{\mathbf{s}}$ so as to be an orthogonal projection of $\mathbf{x}$ on the subspace spanned by $\mathbf{h}_i, i = 1, \ldots, p$ (S=range of $\mathbf{H}$)
- The LS estimator is very sensitive to the correct deterministic model of $\mathbf{s}$, as shown in the figure below for the LS fit of $x[n] = A + Bn + q[n]$.



Summary: The role of the model order p

Recall the AR process order $x(n) = a_1 x(n-1) + \cdots + a_p x(n-p) + q(n)$

Follows naturally from the problem of fitting a polynomial to the data (recall the Weierstrass theorem $\looparrowright$ any continuous differentiable function can be approximated arbitrarily well with a high-enough order polynomial)

- Observe from the previous slide that J_{min} is a **non-increasing function of the model order** p. (for basic a basic idea on regularisation, see Appendix 9)
- The choice $p = N$ is a perfect fit to the "in-sample" data, but overall we overfit, that is, we also fit the noise (see Slide 18 and Slide 5).
- Recall the penalty-based MDL and AIC strategies in AR modelling $\looparrowright$ we choose the **simplest model order** p that is adequate for the data.
- **In practice, if we have a specified** J_{min}, then we can gradually increase p until we reach the required J_{min}.
- To save on computation, we can further exploit the geometry of learning in OLS, and use an order-recursive LS algorithm to compute the model of order $(p+1)$ from the model of order p. (see Appendix 7)

Other forms of Least Squares ↬ towards weighted LS

(assigning importance to data samples)

Recall that the LS estimator aims to minimise $J = \sum_{n=0}^{N-1} e^2(n) = \mathbf{e}^T\mathbf{e}$.

$$\mathbf{e}^T\mathbf{e} = [e_0, e_1, \ldots, e_{N-1}]\begin{bmatrix} e_0 \\ e_1 \\ \vdots \\ e_{N-1} \end{bmatrix} = \mathbf{e}^T\,\mathbf{I}\,\mathbf{e} = [e_0, e_1, \ldots, e_{N-1}]\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}\begin{bmatrix} e_0 \\ e_1 \\ \vdots \\ e_{N-1} \end{bmatrix}$$

We can also assign weighting to the errors, $J = \sum_{n=0}^{N-1} w_n\, e^2(n) = \mathbf{e}^T\mathbf{W}\mathbf{e}$

$$[e_0, e_1, \ldots, e_{N-1}]\begin{bmatrix} w_0 & 0 & \cdots & 0 \\ 0 & w_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_{N-1} \end{bmatrix}\begin{bmatrix} e_0 \\ e_1 \\ \vdots \\ e_{N-1} \end{bmatrix} = \mathbf{e}^T\,\mathbf{W}\,\mathbf{e}$$

For $w_n = \dfrac{1}{\sigma_n^2}$ $\Rightarrow$ $\mathbf{e}^T\,\mathbf{W}\,\mathbf{e} = [e_0, e_1, \ldots, e_{N-1}]\begin{bmatrix} \frac{1}{\sigma_0^2} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sigma_1^2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{\sigma_{N-1}^2} \end{bmatrix}\begin{bmatrix} e_0 \\ e_1 \\ \vdots \\ e_{N-1} \end{bmatrix}$

↑ see Slide 21 and L5 Example 5

☞ If w is a forgetting factor, λ, then $\mathrm{J(n)} = \sum_{k=0}^{n} \lambda^{n-k} e^2(k)$ and $\mathbf{W} = \mathrm{diag}(\lambda^o, \ldots, \lambda^n)$

Weighted Least Squares (WLS)

see also Example 5 in Lecture 5, and Quadratic Forms in Appendix 6 & in Lec. 1

To emphasize the contribution of those data samples that are deemed to be more reliable, we can include an $N \times N$ positive definite (and hence symmetric) **diagonal weighting matrix**, $\mathbf{W}$, so that

$$J(\boldsymbol{\theta}) = \mathbf{e}^T\mathbf{W}\,\mathbf{e} = \big(\mathbf{x} - \mathbf{H}\boldsymbol{\theta}\big)^T\mathbf{W}\big(\mathbf{x} - \mathbf{H}\boldsymbol{\theta}\big)$$

It is now straightforward to show that the weighted least squares solution

$$\hat{\boldsymbol{\theta}} = \big(\mathbf{H}^T\mathbf{W}\mathbf{H}\big)^{-1}\mathbf{H}^T\mathbf{W}\mathbf{x} \quad \& \quad J_{min} = \mathbf{x}^T\Big(\mathbf{W} - \mathbf{W}\mathbf{H}\big(\mathbf{H}^T\mathbf{W}\mathbf{H}\big)^{-1}\mathbf{H}^T\mathbf{W}\Big)\mathbf{x}$$

☞ **Example 3:** For a diagonal $\mathbf{W}$ with elements $[\mathbf{W}]_{ii} = w_i > 0$, the LS error of the DC level estimator becomes

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

If $x[n] = A + q[n]$, where the zero-mean **uncorrelated** noise (not i.i.d., any distribution) $q[n] \sim (0, \sigma_n^2)$, it is reasonable to choose $w_n = 1/\sigma_n^2$, to give

$$\hat{A} = \Big(\sum_{n=0}^{N-1}\frac{x[n]}{\sigma_n^2}\Big)\Big(\sum_{n=0}^{N-1}\frac{1}{\sigma_n^2}\Big)^{-1}$$

Remark: If we take $\mathbf{W} = \mathbf{C}^{-1}$, then the WLS yields the BLUE estimator.

Exponentially weighted LS $\looparrowright$ LS with a "forgetting factor" or "fading memory"

(see Slide 20)

◦ The standard LS cost function is best suited for statistically stationary environments, as it takes into account the whole data history.

◦ In the original cost function all the errors are weighted equally. This is not adequate in statistically nonstationary environments, where distant past is not contributing to learning nor is it statistically relevant.

◦ In order to deal with nonstationary environments, we can modify the LS error criterion to promote forgetting of old data (weighted LS), as

$$J(n) = \sum_{k=0}^{n} \lambda^{n-k} e^2(k) = \begin{bmatrix} e_n, e_{n-1}, \ldots, e_0 \end{bmatrix} \begin{bmatrix} \lambda^0 & 0 & \cdots \\ 0 & \lambda^1 & \cdots \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda^n \end{bmatrix} \begin{bmatrix} e_n \\ e_{n-1} \\ \vdots \\ e_0 \end{bmatrix} = \mathbf{e}^T \mathbf{W} \mathbf{e}$$

◦ The forgetting factor $\lambda \in (0, 1]$, but typically $\lambda > 0.95$.

◦ Through the forgetting factor, λ, the 'old' and often irrelevant/unreliable information is gradually forgotten $\looparrowright$ suitable for non–stationary environ.

◦ The forgetting factor introduces an effective data window length of $\frac{1}{1-\lambda}$

LSE: Opportunities in practical applications $\looparrowright$ numerous

- **Constrained least squares.** We can incorporate a set of linear constraints in the form $\mathbf{A}\boldsymbol{\theta} = \mathbf{c}$, to have a constrained LS criterion
$$J_c(\boldsymbol{\theta}) = \big(\mathbf{x} - \mathbf{H}\boldsymbol{\theta}\big)^T\big(\mathbf{x} - \mathbf{H}\boldsymbol{\theta}\big) - \boldsymbol{\lambda}\big(\mathbf{A}\boldsymbol{\theta} - \mathbf{c}\big)$$
using e.g. Lagrange optimisation as above (first term $\looparrowright$ LS solution $\hat{\boldsymbol{\theta}}$).
- **Nonlinear least squares.** The signal model is nonlinear, i.e. $\mathbf{s} \neq \mathbf{H}\boldsymbol{\theta}$
We can either linearise the problem (e.g. using Taylor series expansion) or solve it numerically in some iterative or recursive fashion. These methods are often prone to convergence problems if highly nonlinear.
- **Dealing with nonlinear least squares $\looparrowright$ parameter transformation.**

Example: Consider a nonlinear problem of estimating the amplitude and phase of a sinusoid $\quad s[n] = A\cos(\omega n + \phi), \quad n = 0, \ldots, N-1$

$\rightsquigarrow$ Transform the problem into $\boldsymbol{A\cos(\omega n + \phi) = A\cos\phi\cos\omega n - A\sin\phi\sin\omega n}$

Variable swap. Let $\alpha_1 = A\cos\phi$ and $\alpha_2 = -A\sin\phi$, and $\boldsymbol{\alpha} = [\alpha_1, \alpha_2]^T$.

Now, the signal model becomes linear in $\boldsymbol{\alpha}$, that is, $\mathbf{s} = \mathbf{H}\boldsymbol{\alpha}$

Use LS to obtain $\hat{\boldsymbol{\alpha}} = \big(\mathbf{H}^T\mathbf{H}\big)^{-1}\mathbf{H}^T\mathbf{x}$ (see Lecture 5 Example 10)

where $A = \sqrt{\alpha_1^2 + \alpha_2^2}$ and $\phi = \arctan(-\alpha_2/\alpha_1)$

LS estimation in the big picture of estimators

Consider the linear model $\mathbf{x} = \mathbf{H}\boldsymbol{\theta} + \mathbf{w}$

Estimator	Model	Assumption	Estimate
LSE	$\mathbf{x} = \mathbf{H}\boldsymbol{\theta} + \mathbf{q}$	no probabilistic assumptions	$\hat{\boldsymbol{\theta}}_{ls} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{x}$
BLUE	$\mathbf{x} = \mathbf{H}\boldsymbol{\theta} + \mathbf{q}$	SOS of q, unknown pdf	$\hat{\boldsymbol{\theta}}_{blue} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{x}$
MLE	$\mathbf{x} = \mathbf{H}\boldsymbol{\theta} + \mathbf{q}$	need to assume pdf of q	$\hat{\boldsymbol{\theta}}_{mle} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{x}$
MVUE	$\mathbf{x} = \mathbf{H}\boldsymbol{\theta} + \mathbf{q}$	need to know pdf of q	$\hat{\boldsymbol{\theta}}_{mvu} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{x}$

LSE and orthogonal projections:

Signal model is $\mathbf{s} = \mathbf{H}\boldsymbol{\theta}$ $\looparrowright$ the estimate is a projection of $\mathbf{x}$ onto $S^p \in \mathbb{R}^p \subset \mathbb{R}^N$

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

where $\mathbf{P} = \mathbf{H}\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T$ is called the **projection matrix**. Since the estimated signal $\hat{\mathbf{s}} = \mathbf{P}\mathbf{x} \in S^p$, it follows that $\mathbf{P}(\mathbf{P}\mathbf{x}) = \mathbf{P}\mathbf{x}$.

Therefore, any projection matrix is **idempotent**, that is $\mathbf{P}^2 = \mathbf{P}$, and it is symmetric and **singular** with rank p (many $\mathbf{x}(n)$ can have the same projection).

From Autoregression to Least Squares regression

Auto-regression: We regress a variable $x[n]$ onto its own past values, $x[n-1], \ldots, x[n-p]$, in the form (with $q[n]$ as driving white noise)

$$\text{Model:} \quad x[n] = a_1 x[n-1] + a_2 x[n-2] + \cdots + a_p x[n-p] + q[n]$$

$$\text{Estimate:} \quad \hat{x}[n] = \hat{a}_1 x[n-1] + \hat{a}_2 x[n-2] + \cdots + \hat{a}_p x[n-p]$$

Multiple regression onto p different variables: The **population model** has the following general form, where $e[n]$ are the residuals

$$\text{Model:} \quad y[n] = \alpha + \beta_1 x_1[n] + \beta_2 x_2[n] + \cdots + \beta_p x_p[n] + e[n]$$

The estimate $\hat{y}[n]$ based on the **multiple regression model** is then

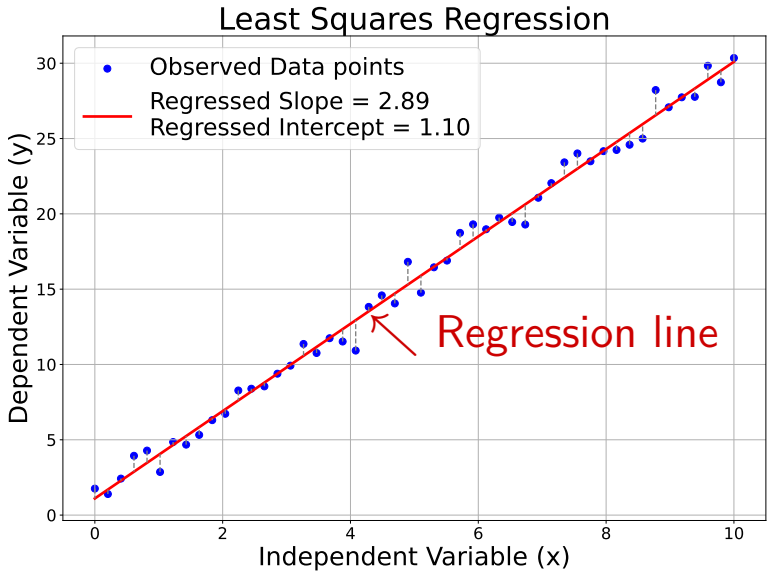
$$\hat{y}[n] = \hat{\alpha} + \hat{\beta}_1 x_1[n] + \hat{\beta}_2 x_2[n] + \cdots + \hat{\beta}_p x_p[n]$$

As before, α represents the intercept, but the β's are now the **partial correlation coefficients.** (for more detail, see Lecture 8)

Least Squares Regression (LSR): A brief summary

Linear regression $\looparrowright$ relationship between two variables based on a line of best fit

Consider a line fit: $\boldsymbol{y} = \beta\,\boldsymbol{x} + \boldsymbol{e} \Longleftrightarrow y_i = \beta\,x_i + e_i \quad i \in \{1, \ldots, N\}$



- Least Squares regression (LSR) aims to minimise the sum of the squares of the differences between the observed and predicted values

$$\underset{\beta}{\operatorname{argmin}} ||\boldsymbol{y} - \beta\boldsymbol{x}||_2^2 \Longleftrightarrow \underset{\beta}{\operatorname{argmin}} ||\boldsymbol{e}||_2^2$$

- We say that we regress y onto x, with β as the regression coefficient.

Common terminologies for Least Squares Regression

	Econometrics	**Statistics**	**Machine Learning**
$\boldsymbol{y}$	Dependent Var., Estimate	Explained V., Response, Regressand	True Label, Criterion
β	Coefficients	Coefficients	Parameters
$\boldsymbol{x}$	Independent Var., Predictor	Explanatory Var. Regressor	Features, Predictors
$\boldsymbol{e}$	Residual	Error	Prediction Error

Example 4a: Capital Asset Pricing Model (CAPM)

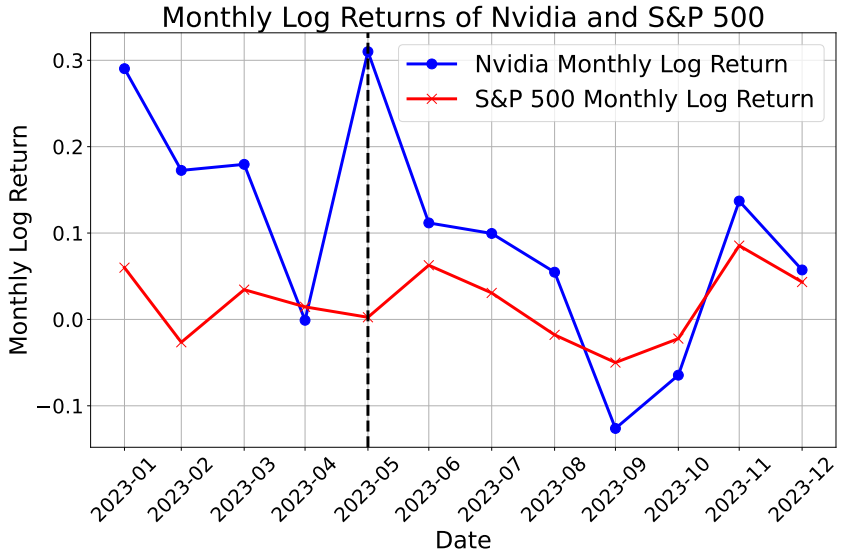
W. Sharpe was awarded the Nobel Prize in Economic Sciences in 1990 for CAPM

The CAPM is given by the regression model

$$E(R_i) = R_f + \beta\left(E(R_m) - R_f\right) + e$$

expected return of asset i ↗ risk-free ↑ ↑ exposure to market ↖ residual (unpredictable)

- R_f is the **risk-free** rate of interest, e.g. interest arising from government bonds; R_f is assumed to be e.g. the 3% Annual Percentage Rate (APR);
- β (the beta) ↬ sensitivity of the expected excess asset returns, $E(R_i)$-R_f, to excess market returns, $E(R_m)$-R_f, **(β=exposure to market)**.
- $\left(E(R_i) - R_f\right)$ is known as the **risk premium**;
- $E(R_m)$ is the expected return of the market;
- $\left(E(R_m) - R_f\right)$ is the **market premium** or **excess return of the market** (difference between the expected market return and the risk free).



↬ We assume that the market is the S&P 500 index and regress for β.

☞ **So CAPM is actually fitting a line to noisy data!** ↬ LS regression

Large β ↬ a less resilient company Small β ↬ lower exposure to market risk

Example 4a: Capital Asset Pricing Model (CAPM), cntd.

We here employ a block-LS approach, over blocks of 22 days (see Appendix 10)

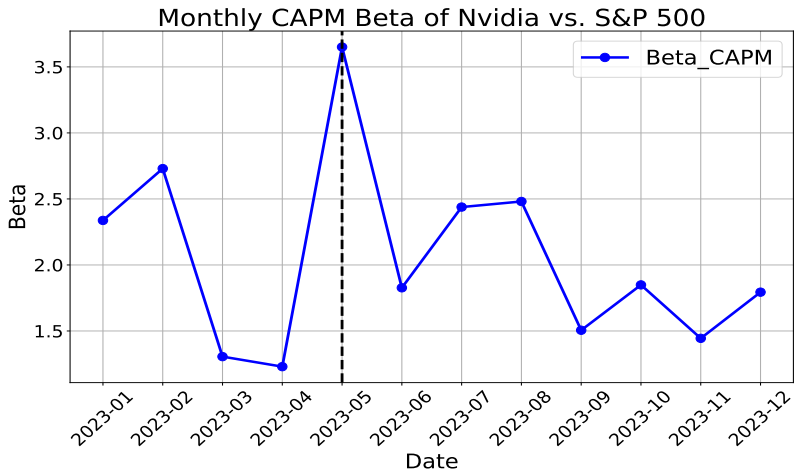
Asset return R_i, risk-free interest rate R_f, and market return R_m (S&P500 return) are all known. We consider log-returns.

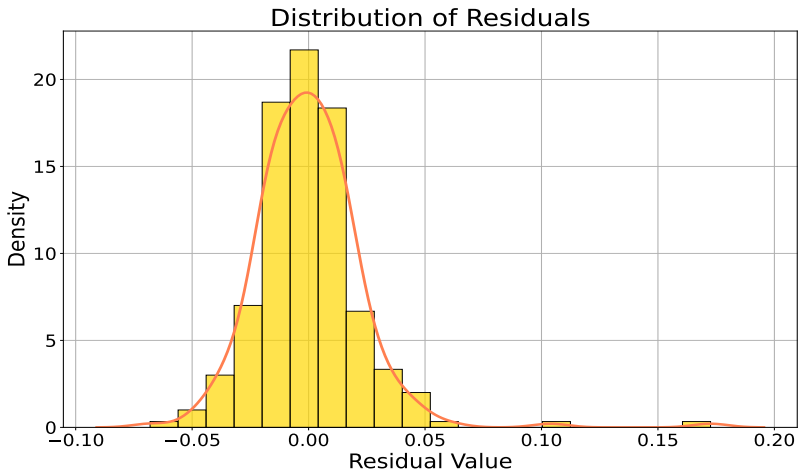
☞ We can now perform LS regression to obtain the value of β.

Each month has 22 trading days. Then, the CAPM states that

$$\begin{bmatrix} R_{i;day1} - R_f \\ R_{i;day2} - R_f \\ \vdots \\ R_{i;day22} - R_f \end{bmatrix} = \beta \begin{bmatrix} R_{m;day1} - R_f \\ R_{m;day2} - R_f \\ \vdots \\ R_{m;day22} - R_f \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_{22} \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{r_i = \beta\, r_m + e}$$

Therefore, the LS estimate: $\hat{\beta} = (\boldsymbol{r_m}^{\mathbf{T}} \boldsymbol{r_m})^{-1} \boldsymbol{r_m}^{\mathbf{T}} \boldsymbol{r_i}$





Example 4b: Fama-French three-factor model (Problem Sets)

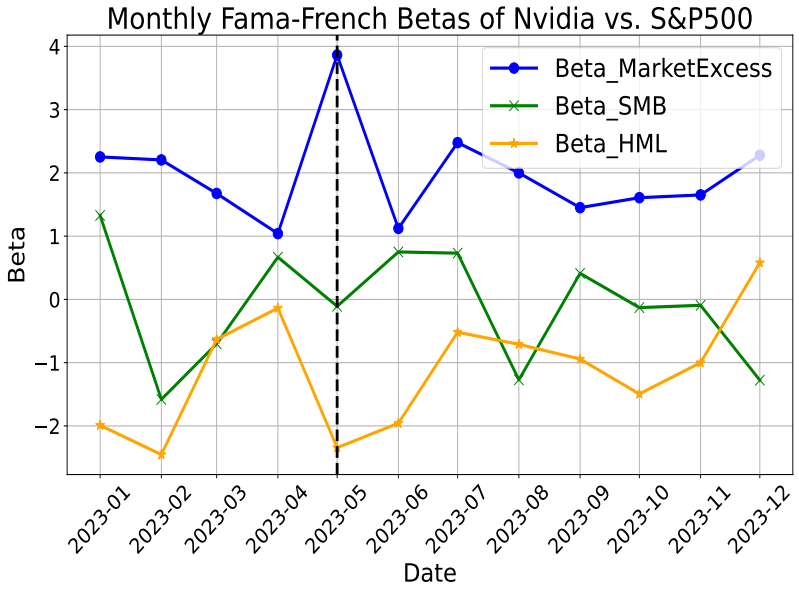
NB: β here is not equal to β in CAPM, due to two additional factors

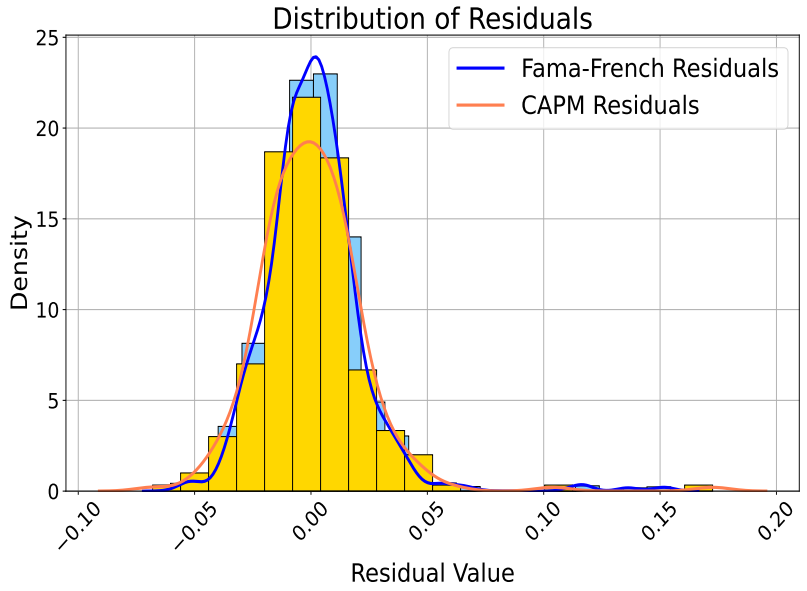
The model is given by (E. Fama won Nobel Prize in Economics in 2013)

$$R_i = R_f + \beta\,(R_m - R_f) + bs \cdot SMB + bv \cdot HML + e$$

where SMB measures the historic excess returns of small caps over big caps and HML the value stocks over growth stocks, bs and bv are coeffs.

LS Regression of Fama-French: We regress for the three beta's: The market is the S&P 500 index; R_f is assumed to be 3% APR; Intercept $= 0$.





Sequential least squares (SLS)

Oftentimes data are collected sequentially (streaming data), namely one point at a time. To process such data, we can either:

- Wait until all the data points (samples) are collected and make an estimate of the unknown parameters $\looparrowright$ **block-based approach**, or
- Refine our estimate as each new sample arrives $\looparrowright$ **sequential approach**

We shall now modify the LS method from a batch to a sequential mode.

Objective:

Suppose we have a least squares estimate, $\hat{\boldsymbol{\theta}}_{N-1}$, which is based on the full signal history $\{x[0], x[1], \ldots, x[N-1]\}$.

We wish to produce a new estimate, $\hat{\boldsymbol{\theta}}_N$, upon observing the new data sample, $x[N]$, but without using full dataset $\{x[0], \ldots, x[N]\}$.

Question: Can we update the existing estimate, $\hat{\boldsymbol{\theta}}_{N-1}$, into the new estimate, $\hat{\boldsymbol{\theta}}_N$, sequentially, based on only $\hat{\boldsymbol{\theta}}_{N-1}$ and $x[N]$, that is

$$\hat{\boldsymbol{\theta}}_N = f\left(\hat{\boldsymbol{\theta}}_{N-1}, x[N]\right)$$

Example 5: DC level in uncorrelated zero mean noise

(new notation, $\hat{A}[N]$ = "estimate of A at a time instant N")

Consider the problem of LS estimation the DC level in noise, for which we have obtained

$$\hat{A}[N-1] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]$$

If we now observe the new sample $x[N]$, then the new, enhanced, estimate

$$\hat{A}[N] = \frac{1}{N+1}\sum_{n=0}^{N} x[n] = \frac{1}{N+1}\Big(\sum_{n=0}^{N-1} x[n] + x[N]\Big)$$

$$\hat{A}[N] = \frac{N}{N+1}\hat{A}[N-1] + \frac{1}{N+1}x[N] \quad \rightsquigarrow \textbf{ a recursive estimate!}$$

☞ **Similarly, to compute the minimum LS error recursively** (Appendix 2)

$$\text{from} \qquad J_{min}[N-1] = \sum_{n=0}^{N-1}\big(x[n] - \hat{A}[N-1]\big)^2$$

Upon arrival of $x[N]$, re-arrange $\qquad J_{min}[N] = \sum_{n=0}^{N}\big(x[n] - \hat{A}[N]\big)^2 \qquad (*)$

Example 5: DC level in noise $\rightarrowtail$ a more convenient form of the sequential estimator and the associated MSE

Clearly, the new estimate $\hat{A}[N]$ can be calculated from the old estimate $\hat{A}[N-1]$, upon receiving the new observation $x[N]$.

The solution can be rewritten in a more physically insightful form, as

new estimate $\searrow$ old estimate $\downarrow$ gain $\searrow$ $\downarrow$ new data $\swarrow$ new estimate

$$\hat{A}[N] = \hat{A}[N-1] + \frac{1}{N+1}\Big(x[N] - \hat{A}[N-1]\Big)$$

$$\textbf{new estimate} = \textbf{old estimate} + \underbrace{\textbf{gain} \times \textbf{error}}_{\textbf{correction}}$$

The minimum LS error then becomes (show yourselves, or see Appendix 2)

$$J_{\min}[N] = J_{\min}[N-1] + \frac{N}{N+1}\Big(x[N] - \hat{A}[N-1]\Big)^2$$

☞ Notice that $J_{\min}$ is "cumulative" and increases with the number of data points, N, as we are trying to fit more points with the same number of parameters (over-determined system).

Example 6: Weighted LS for the estimation of DC level in noise in a sequential form (see Example 9 in Lecture 4 & Slide 21)

Start from

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

If $x[n] = A + q[n]$, where the zero-mean **uncorrelated** noise (any distribution) $q[n] \sim (0, \sigma_n^2)$, it is reasonable to choose $w_n = 1/\sigma_n^2$, to give[1]

$$\textbf{Standard LS solution :} \qquad \hat{A}[N] = \frac{\sum_{n=0}^{N} \frac{x[n]}{\sigma_n^2}}{\sum_{n=0}^{N} \frac{1}{\sigma_n^2}}$$

Its corresponding sequential form then becomes

$$\hat{A}[N] = \hat{A}[N-1] + \frac{\frac{1}{\sigma_N^2}}{\sum_{n=0}^{N} \frac{1}{\sigma_n^2}} \big(x[N] - \hat{A}[N-1]\big)$$

or **new estimate** = **old estimate** + **gain** × **error**

In practice, we may employ a forgetting factor $\lambda < 1$, to give $J(A) = \sum_{n=0}^{N-1} \lambda^{N-1-n} e^2(n)$

[1] In standard weighted LS, with a diagonal weighting matrix $\mathbf{W}$, we would have $[\mathbf{W}]_{ii} = \frac{1}{\sigma_i^2}$.

Observations about weighted LS: Noisy sample

How does a new noisy sample, $x[N]$, with a large σ_N^2 influence the estimation?

Notice that the gain reflects a **relative match between the current estimate and the new data**, and **depends on our confidence** in the new data sample, $x[N]$, given by $1/\sigma_N^2$.

Two extreme cases:

- If $\sigma_N^2 \to \infty$, i.e. the new sample is extremely noisy, then we automatically do not correct the previous LS estimate (LSE)
- If $\sigma_N^2 \to 0$, that is, the new sample is noise–free, then $\hat{A} \to x[N]$, and all previous samples are discarded

☞ If we assume $x[n] = A + q[n]$, with $\{q[n]\}$ zero mean uncorrelated noise for which the variance of each $q[n]$ is $\sigma_n^2,\ n = 0, \ldots, N-1$, then the LSE is also the BLUE and

$$var\big(\hat{A}[N-1]\big) = \frac{1}{\sum_{n=0}^{N-1} \frac{1}{\sigma_n^2}}$$

(Lecture 5 slide 28)

Weighted LS: Influence of "goodness" of the estimate $\hat{A}$

- The gain for the N-th update can be rewritten as $(0 \leq K[N] \leq 1)$

$$K[N] = \frac{\frac{1}{\sigma_N^2}}{\sum_{n=0}^{N} \frac{1}{\sigma_n^2}} = \frac{\frac{1}{\sigma_N^2}}{\frac{1}{\sigma_N^2} + \frac{1}{var\left(\hat{A}[N-1]\right)}} = \frac{var\left(\hat{A}[N-1]\right)}{var\left(\hat{A}[N-1]\right) + \sigma_N^2}$$

- **Bad estimate, good data.** If $var\left(\hat{A}[N-1]\right) \gg \sigma_N^2$, then new data is very useful, $K[N] \approx 1$, and the correction based on new data is large

- **Good estimate, bad data.** Conversely, is $var\left(\hat{A}[N-1]\right) \ll \sigma_N^2$, then new data has little use, $K[N] \approx 0$, and the correction is small

- The recursive expression for the variance can be calculated as

$$var\left(\hat{A}[N]\right) = \Big(1 - K[N]\, var\left(\hat{A}[N-1]\right)\Big)$$

☞ **Notice that the gain $K[n]$ is also a random variable.**

Summary of sequential DC level estimators, both weighted and standard

DClevel.m

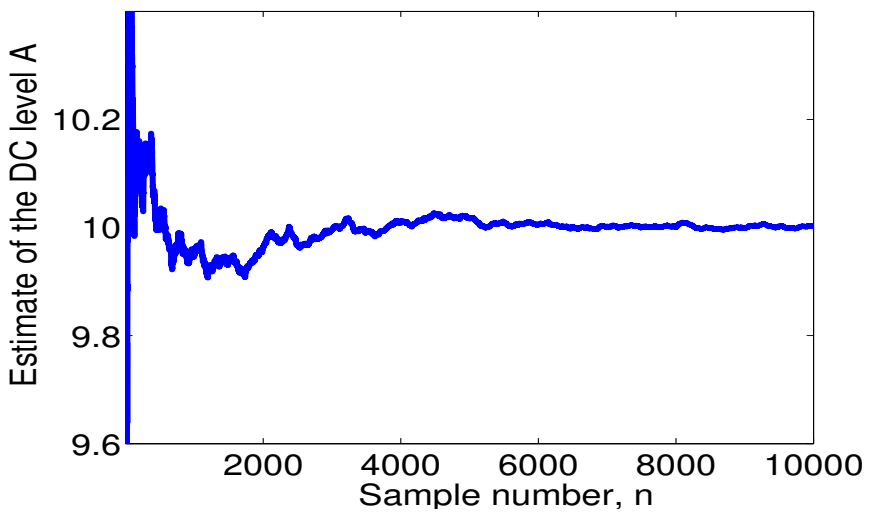
Estimator update: $\hat{A}[N] = \hat{A}[N-1] + K[N]\Big(x[N] - \hat{A}[N-1]\Big)$

Weighted: $K[N] = \dfrac{var\big(\hat{A}[N-1]\big)}{var\big(\hat{A}[N-1]\big) + \sigma_N^2}$ Standard: $K[N] = \dfrac{1}{N+1}$

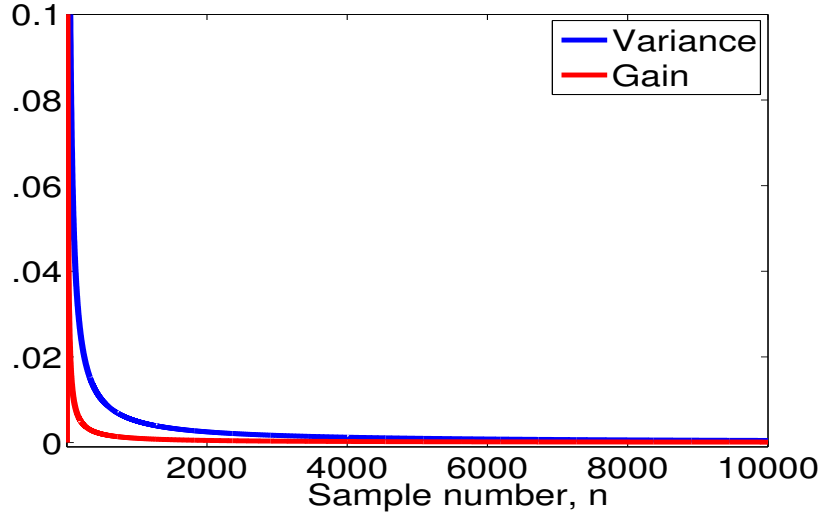
Variance update: $var\big(\hat{A}[N]\big) = \big(1 - K[N]\big)\, var\big(\hat{A}[N-1]\big)$

Initialisation: $\hat{A}[0] = x[0], \quad var\big(\hat{A}[0]\big) = \sigma_0^2$

Example 7: Perform sequential DC level estimation for $A = 10$, $\sigma^2 = 5$



Evolution of the estimate $\hat{A}$



Variance and gain

Towards the vector parameter case: A noisy line example

(see Slides 26 – 28 here, and Lecture 4)

The observed data: $x[n] = A + Bn + q(n) \quad \equiv \quad \mathbf{x} = \mathbf{H}\boldsymbol{\theta} + \mathbf{q}$

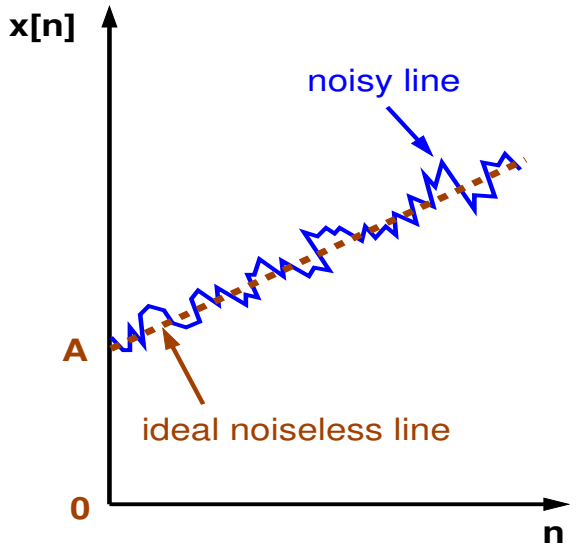
where $\mathbf{x} = [x_0, x_1, \ldots, x_{N-1}]^T$, $\mathbf{q} = [q_0, q_1, \ldots, q_{N-1}]^T$, and $\boldsymbol{\theta} = [A \quad B]^T$

Then, for N data points

$$\mathbf{H}_{N-1} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ \vdots & \vdots \\ 1 & N-1 \end{bmatrix}_{N \times 2}$$

while, for $N+1$ data points

$$\mathbf{H}_N = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ \vdots & \vdots \\ 1 & N \end{bmatrix}_{(N+1) \times 2}$$



Thus, for $(N+1)th$ data point

$$\mathbf{H}_N = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ \vdots & \vdots \\ 1 & N-1 \\ 1 & N \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{N-1} \\ 1 \quad N \end{bmatrix}_{(N+1) \times 2}$$

↖ grows with N

Sequential LSE for a vector parameter

Consider an input $\mathbf{x}[n] = \big[x[0], x[1], \ldots, x[n]\big]^T \rightsquigarrow \quad \mathbf{H}[n] = \begin{bmatrix} \mathbf{H}[n-1]_{n\times p} \\ \\ \mathbf{h}^T[n]_{1\times p} \end{bmatrix}$

Note that the size of the observation matrix $\mathbf{H}$ grows with time.

- **Estimator update:**

$$\hat{\boldsymbol{\theta}}[n] = \hat{\boldsymbol{\theta}}[n-1] + \mathbf{K}[n]\Big(\underbrace{x[n] - \mathbf{h}^T[n]\,\hat{\boldsymbol{\theta}}[n-1]}_{error}\Big)$$

new estimate ↑ ↑ old estimate ↖ gain

where the **gain factor** is given by

$$\mathbf{K}[n] = \mathbf{C}[n-1]\mathbf{h}[n]\Big[\sigma_n^2 + \mathbf{h}^T[n]\mathbf{C}[n-1]\mathbf{h}[n]\Big]^{-1}$$

↑ var. of the most recent sample

- **Covariance matrix update:**

$$\mathbf{C}[n] = \Big(\mathbf{I} - \mathbf{K}[n]\mathbf{h}^T[n]\Big)\mathbf{C}[n-1]$$

- **Initialisation:** $\mathbf{C}[-1] = \alpha\,\mathbf{I}, \quad \alpha \to \text{large}, \qquad \boldsymbol{\theta}[-1] = \mathbf{0}$

Example 8: Sequential LS for the parameters of a line

zero- and first-order sequential least-squares estimator for $x[n] = A + Bn + q[n]$

- Measurement $x[n] = A + Bn + q[n]$ and the vector parameter $\hat{\boldsymbol{\theta}}[n] = \left[\hat{A}, \quad \hat{B}\right]^{\top}$
- **Estimator update:** $\hat{\boldsymbol{\theta}}[n] = \hat{\boldsymbol{\theta}}[n-1] + \mathbf{K}[n]\Big(x[n] - \mathbf{h}^T[n]\boldsymbol{\Phi}[n]\hat{\boldsymbol{\theta}}[n-1]\Big)$

  where $\boldsymbol{\Phi}[n] = \begin{bmatrix} 1 & n \\ 0 & 1 \end{bmatrix}$ and $\mathbf{h}[n] = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
- **Initialisation:** $\mathbf{C}[-1] = \alpha\,\mathbf{I}, \qquad \alpha > 100\,\sigma_0^2, \qquad \hat{\boldsymbol{\theta}}[-1] = [0,\ 0]^T$
- **Update (Ricatti equations):**

$$\mathbf{M}[n] = \boldsymbol{\Phi}[n]\mathbf{C}[n-1]\boldsymbol{\Phi}^T[n]$$

$$\mathbf{K}[n] = \mathbf{M}[n]\mathbf{h}[n]\left[\mathbf{h}^T[n]\mathbf{M}[n]\mathbf{h}[n] + \sigma_n^2\right]^{-1}$$

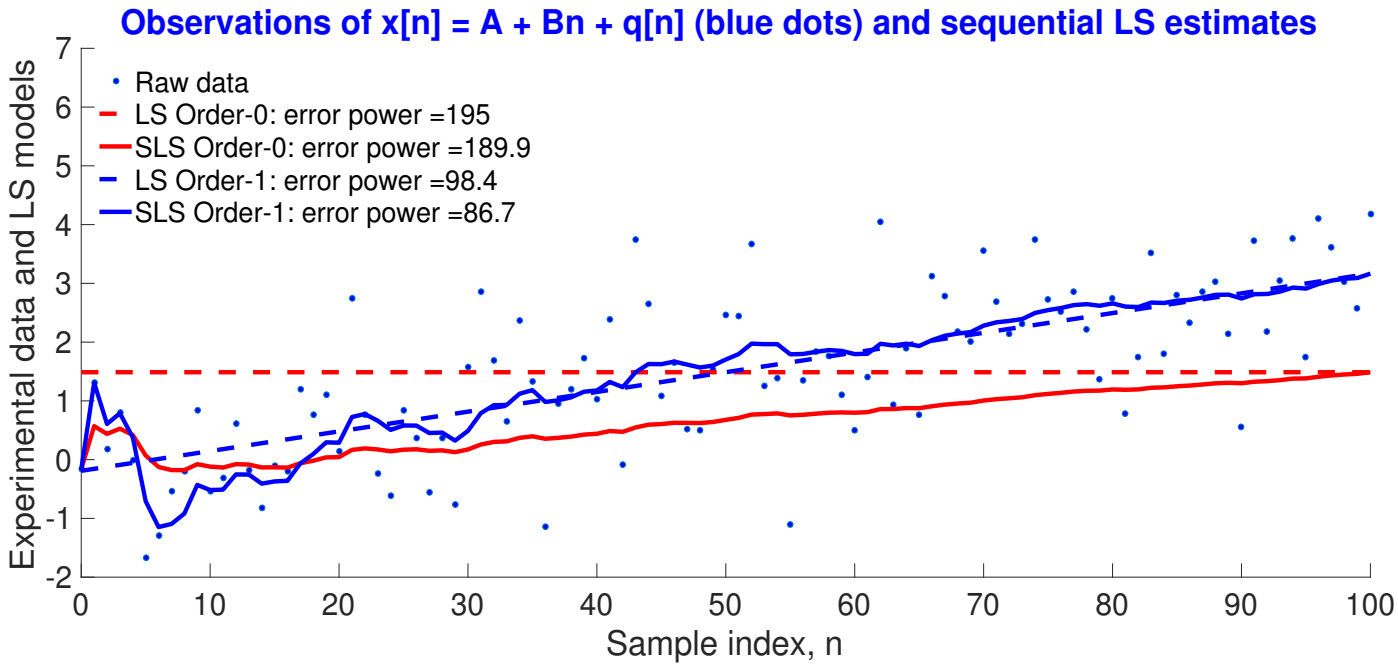
$$\mathbf{C}[n] = \Big(\mathbf{I} - \mathbf{K}[n]\mathbf{h}^T[n]\Big)\mathbf{M}[n]$$

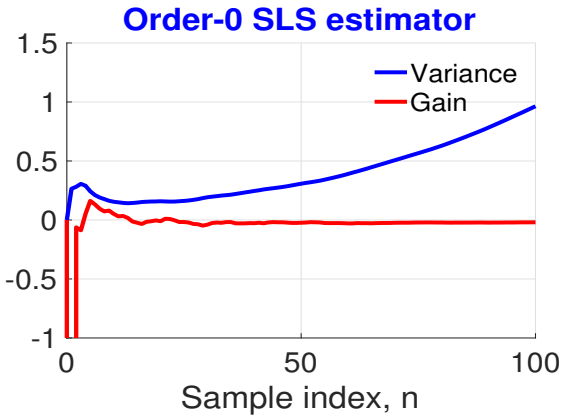
- The **gain factor** is updated as $\mathbf{K}[n] = \begin{bmatrix} \frac{2(2n-1)}{n(n+1)} \\ \frac{6}{n(n+1)} \end{bmatrix}$

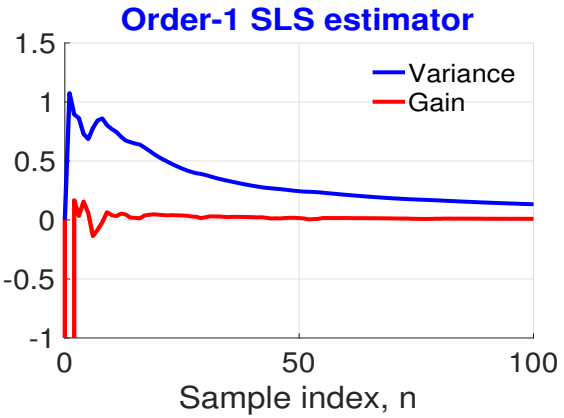
  and the **covariance matrix** as $\mathbf{C}[n] = \begin{bmatrix} \frac{2(2n-1)}{n(n+1)}\,\sigma_n^2 & 0 \\ 0 & \frac{12}{n(n^2+1)}\,\sigma_n^2 \end{bmatrix}$

Example 8: Continued

Matlab: Sequential_LS_Order_Interactive_Local.m







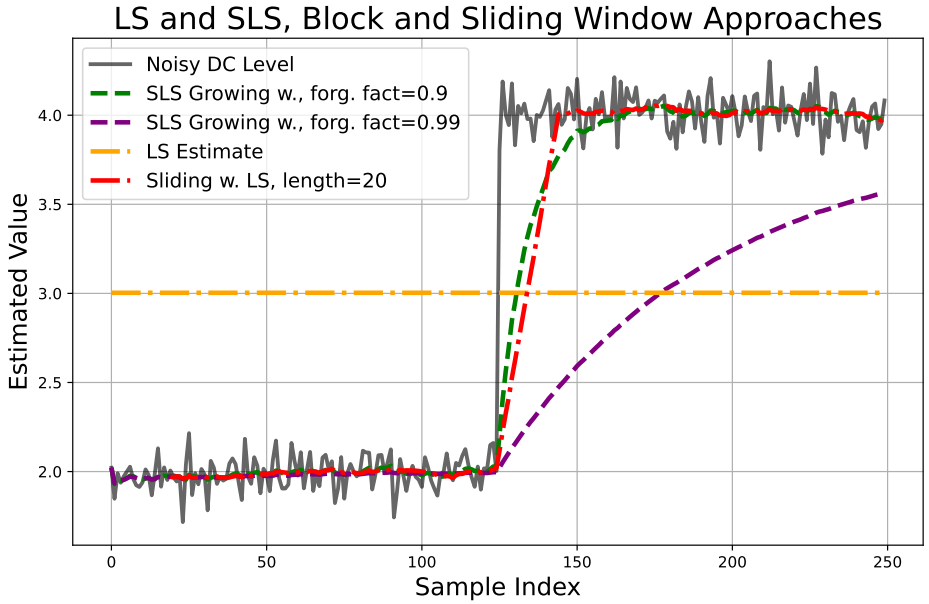
Example 9: Least Squares (LS) and Sequential Least Squares (SLS) for non-stationary data

LS_and_SLS_1.ipynb

Consider the case where the DC level changes its value from $A = 2$ to $A = 4$, at the sample index $n = 125$. This is a source of non–stationarity.

LS: $\hat{A} = \frac{1}{N}\sum_{n=1}^{N-1} x[n]$ **SLS:** $\hat{A}[N] = \hat{A}[N-1] + \frac{1}{N+1}\Big(x[N] - \hat{A}[N-1]\Big)$

Sliding window LS: Calculate the LS estimate over an L=20 samples long sliding window, termed Sliding LS, start as $\hat{A}[19] = \frac{1}{20}\sum_{n=0}^{19} x[n]$. Plot the estimate, then shift the window by one sample, and repeat until $N = 250$.



The forgetting factor, λ helps with nonstationary data

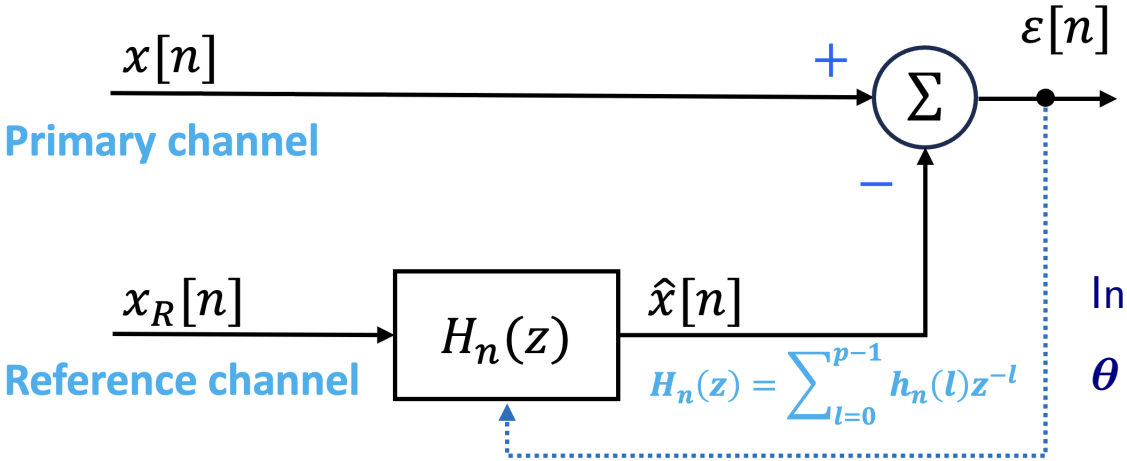
A sliding LS with data window of $L = 20$ is a good alternative

Case study: Adaptive Noise Canceller (ANC)

A common application of adaptive learning in order to reduce unwanted noise

Case Study 1: A common problem is the removal of artefacts in sensor data in biomedicine, or removal of 50 Hz interference in instrumentation.

Case Study 2: We may wish to remove background noise in audio systems in aircrafts or cars (noise cancelling headphones, road noise cancellation).



Configuration of a general noise canceller

- The **reference channel** takes the role of the **traditional input**
- The **primary channel**, that is the noisy signal of interest, takes the role of the **desired input (teaching signal).**
- The residual, ε, takes the role of the "system output".

ANC $\looparrowright$ line interference removal

- **Primary channel:** 'signal' + 'noise to be cancelled' (for example, the 50 Hz mains interference in an acquired ECG signal)
- **Reference channel:** Noise source which is related to the noise in the primary channel (non-zero correlation)
- Filter coefficients are updated sequentially to make $\hat{x}[n]$ as close to $x[n]$ as possible, in the LS sense, with $\hat{x}[n] = \sum_{l=0}^{p-1} h_n(l) x_R(k-l)$
- We therefore desire to minimise the power of the residual, $\varepsilon[n]$, that is

$$\begin{aligned} J[n] &= \sum_{k=0}^{n} \varepsilon^2[k] = \sum_{k=0}^{n} \big(x[k] - \hat{x}[k]\big)^2 \\ &= \sum_{k=0}^{n} \Big(x[k] - \sum_{l=0}^{p-1} h_n(l) x_R[k-l]\Big)^2 \end{aligned}$$

- Filter coefficients (weights) can then be determined as a solution to the sequential LS (SLS) problem

ANC ↬ some practical considerations

The signal and noise are typically statistically nonstationary, and to deal with that we introduce the weighting in the form of a **"forgetting factor"** λ, for which the range is $0 < \lambda < 1$, so that the cost function becomes

$$J[n] = \sum_{k=0}^{n} \lambda^{n-k} \Big(x[k] - \sum_{l=0}^{p-1} h_n(l) x_R[k-l] \Big)^2$$

The solution will not change if we minimise instead (see S. Kay's book)

$$J'[n] = \sum_{k=0}^{n} \frac{1}{\lambda^k} \Big(x[k] - \sum_{l=0}^{p-1} h_n(l) x_R[k-l] \Big)^2$$

Notice that $J[n]$ is different from $J'[n]$, but the solutions are identical.

☞ This is also the form of the standard weighted LS problem.

The sequential LS vector estimator of the filter coefficients is denoted by

$$\hat{\boldsymbol{\theta}}[n] = \big[\hat{h}_n(0), \hat{h}_n(1), \ldots, \hat{h}_n(p-1) \big]^T$$

ANC summary ↬ Follows from Slide 38

Notice that here $\mathbf{h}[n]$ from Slides 42–43 is replaced by $\hat{\theta}$, to avoid confusion

Input reference vector: $\mathbf{x}_R[n] = \big[x_R[n], x_R[n-1], \ldots, x_R[n-p+1]\big]^T$

Weights: $\sigma_n^2 = \lambda^n$ weighting coefficients w ☞ forgetting factor λ

Error:

$e[n] = x[n] - \sum_{l=0}^{p-1} \hat{h}_{n-1}(l) x_R[n-l] = x[n] - \mathbf{x}_R^T[n]\,\hat{\boldsymbol{\theta}}[n-1] = e_{n|n-1}$

error at time [n] based on parameters at time [n-1] ↑

Estimator update: $\hat{\boldsymbol{\theta}}[n] = \hat{\boldsymbol{\theta}}[n-1] + \mathbf{K}[n] e[n]$

where:

$$e[n] = x[n] - \sum_{l=0}^{p-1} \hat{h}_{n-1}(l) x_R[n-l]$$

$$\mathbf{K}[n] = \frac{\mathbf{C}[n-1]\mathbf{x}_R[n]}{\lambda^n + \mathbf{x}_R^T[n]\mathbf{C}[n-1]\mathbf{x}_R[n]}$$

$$\mathbf{C}[n] = \big(\mathbf{I} - \mathbf{K}[n]\mathbf{x}^T[n]\big)\mathbf{C}[n-1], \quad \text{typically} \quad 0.9 < \lambda < 1$$

☞ In LS methods we typically do not know the variances σ_n^2 for every $x[n]$. They may be replaced with a forgetting factor λ^n. This favours most recent samples. (see Slide 22)

Example 10: Line noise removal ANC_Line_Noise_Complex_Valued

Reference x_R is correlated with interference but has different amplitude and phase

Consider interference estimation only, that is, $s[n;\boldsymbol{\theta}] = 0$ and $q[n] = 10\cos\left(2\pi(0.1)n + \pi/4\right)$

Primary ch.: $x[n] = 10\cos\left(2\pi(0.1)n + \pi/4\right)$

Reference channel: $x_R[n] = \cos(2\pi(0.1)n)$

Initialisation: $\hat{\boldsymbol{\theta}}[-1] = \mathbf{0}$, $\mathbf{C}[-1] = 10^5\,\mathbf{I}$, and $\lambda = 0.99$.

◦ We need two filter coefficients to model the amplitude and phase of the interference, that is

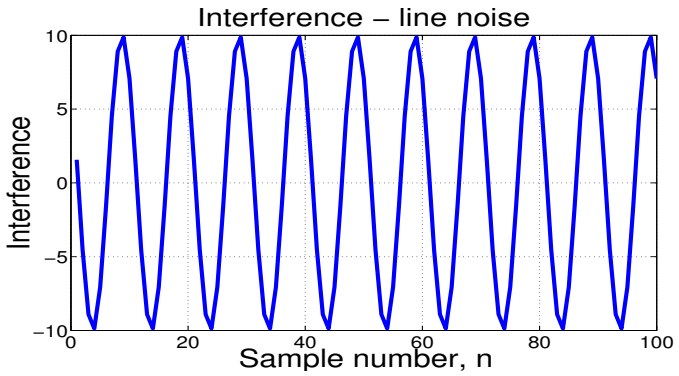
$$\mathcal{H}[exp(2\pi(0.1))] = 10\,exp(\jmath\pi/4)$$

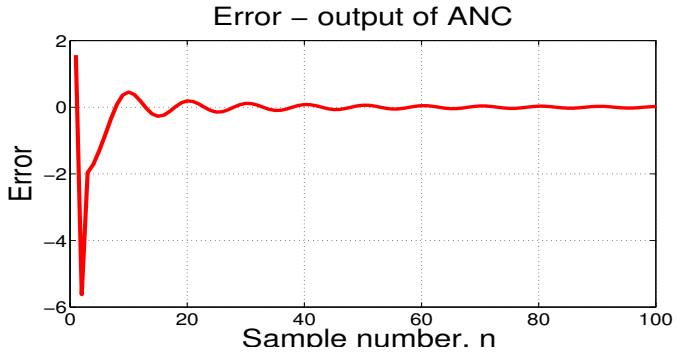
☞ The noise canceller must increase the gain of the reference, $x_R[n]$, by a factor of 10 and phase by $\pi/4$ to match the interference.

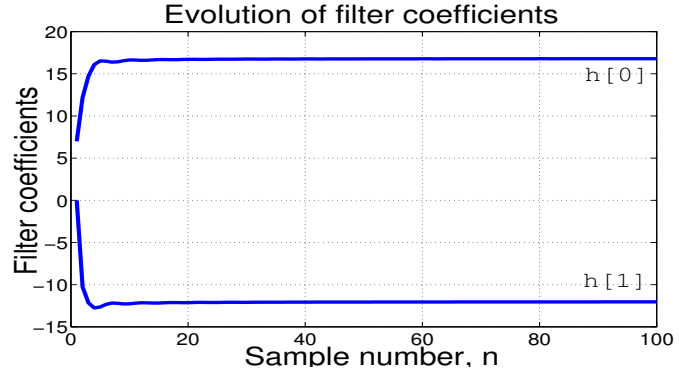
Upon solving (ANC performance is on the right)

$h[0] + h[1]exp(-2\jmath\pi(0.1)) = 10exp(\jmath\pi/4)$

which results in $h[0] = 16.8$ and $h[1] = -12$.



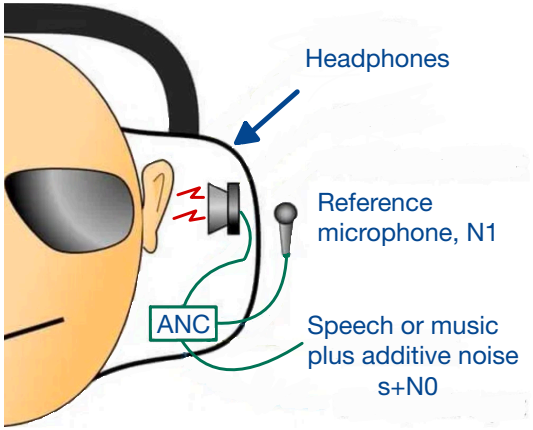


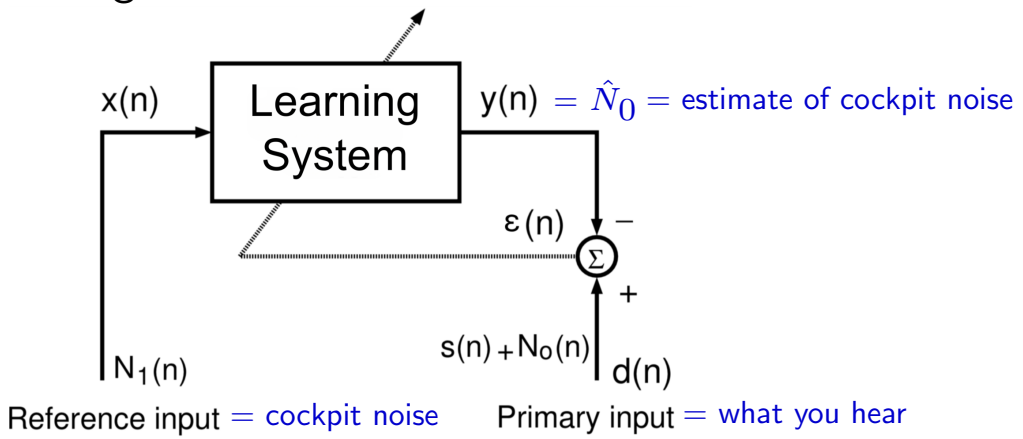


Applications: Adaptive noise cancellation with reference

(such as in noise-canceling headphones on an airplane)

In the adaptive noise cancellation configuration (below right, more details in Lecture 7), the variables in the adaptive filter have the following roles.





Input to the filter, is the Reference Noise signal, that is, $x(n) = N_1(n)$. The only requirement is that N_1 is correlated with the measurement noise, N_0, but not with the signal of interest, $s(n)$. The filter aims to estimate N_0 from N_1, that is, $y = \hat{N}_0$.

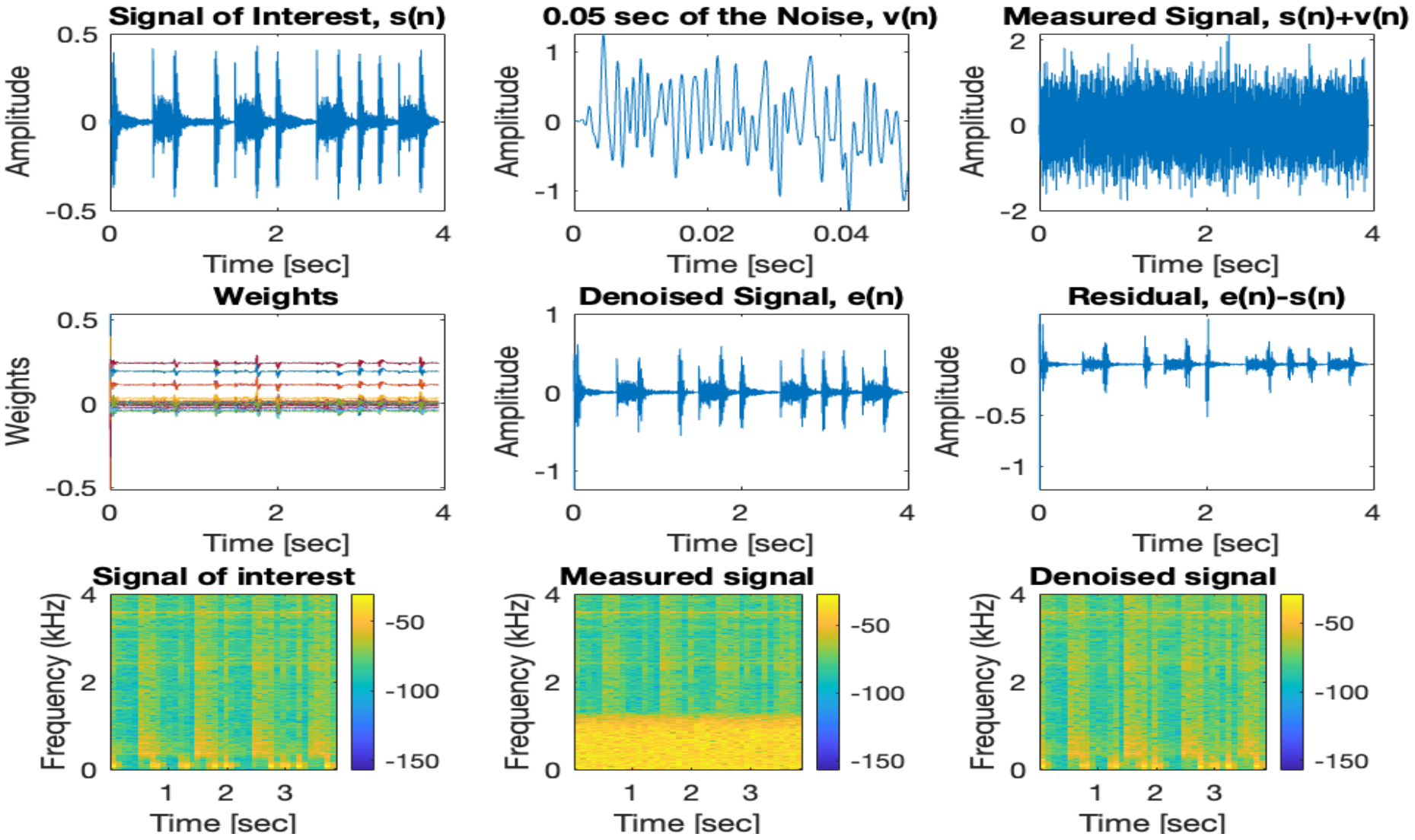
Teaching signal, $d(n)$, is the noise-polluted signal of interest, $s(n) + N_0(n)$, which serves as the Primary Input to the filter. Since $s \perp N_1$, the filter can only yield $y = \hat{N}_0$.

Filter output, $y = \hat{N}_0$, provides the best MSE estimate of the measurement noise, N_0, from the reference noise, N_1. The more correlated N_1 and N_0 the faster the convergence.

Output error, $\varepsilon = s + N_0 - \hat{N}_0$, serves as a **"system output"**, whereby the adaptive filter aims to achieve $\varepsilon = \hat{s} \approx s$, that is to obtain noise-free music signal, $s(n)$.

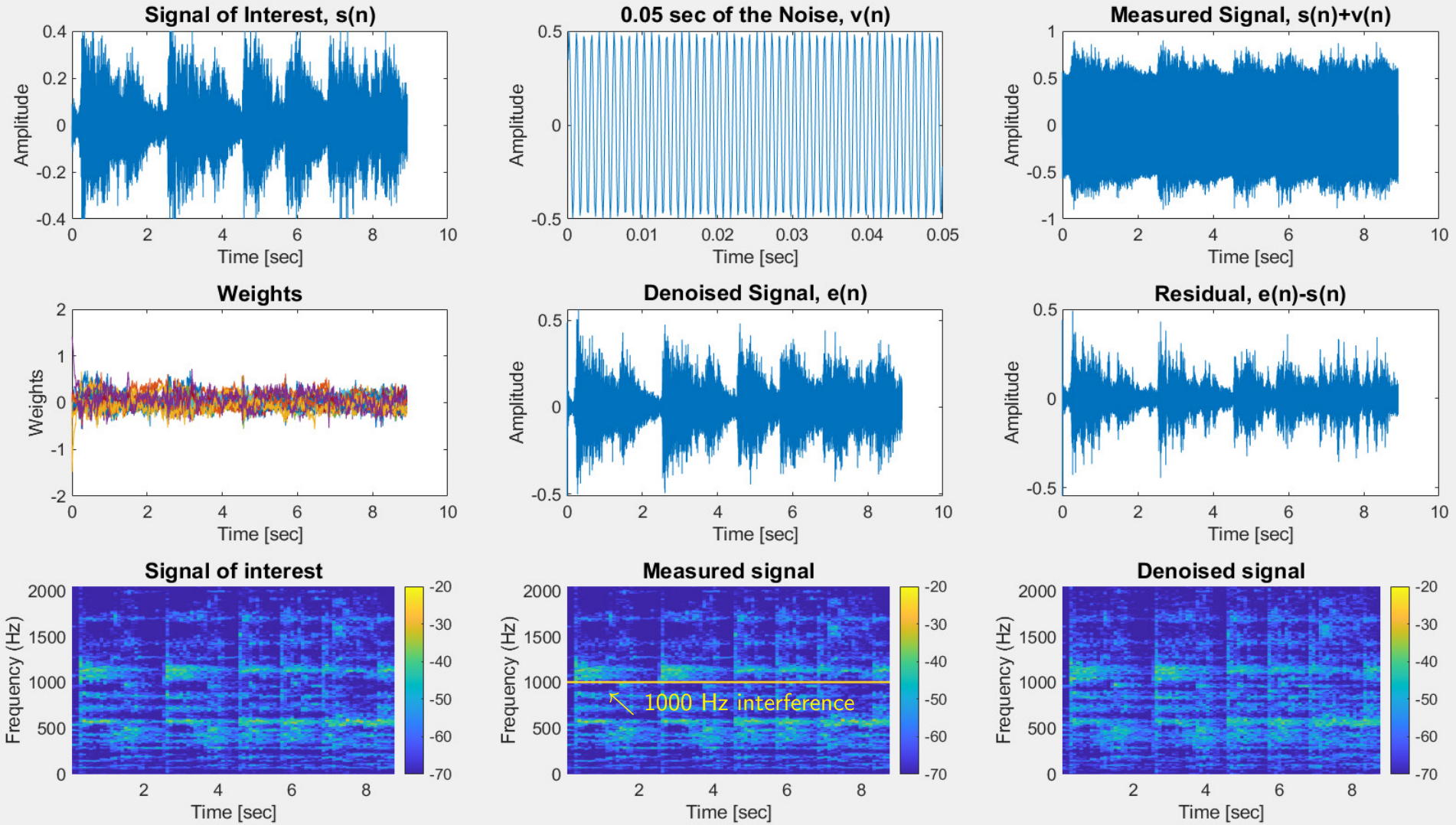
Example 11: Noise cancelling headphones ($\lambda = 0.99$)

Denoising_SLS_GUI.m



Example 12: Acoustic feedback cancellation ($\lambda = 0.995$)

Denoising_SLS_GUI.m

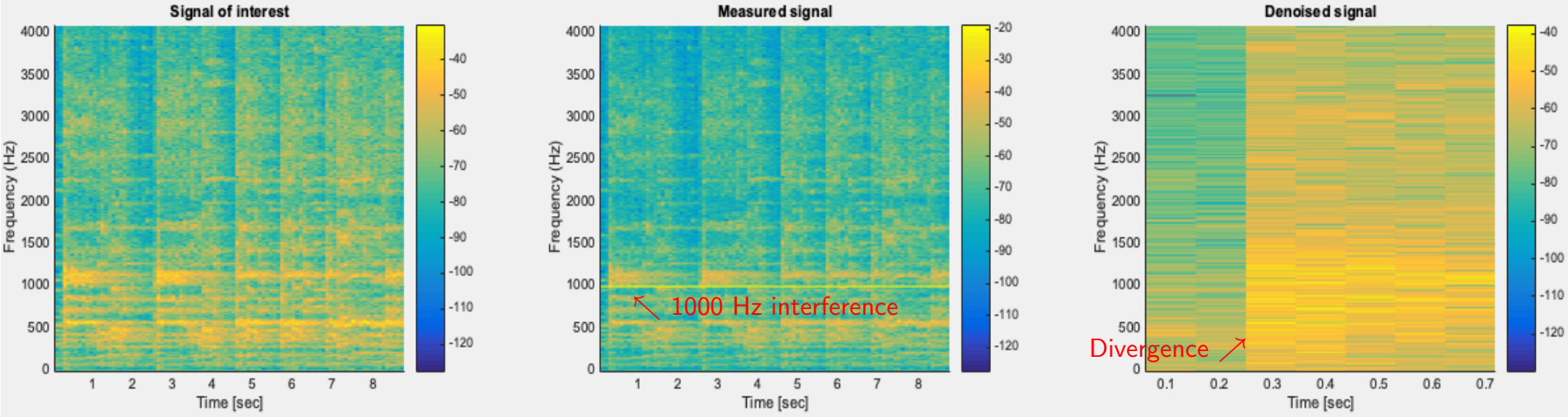


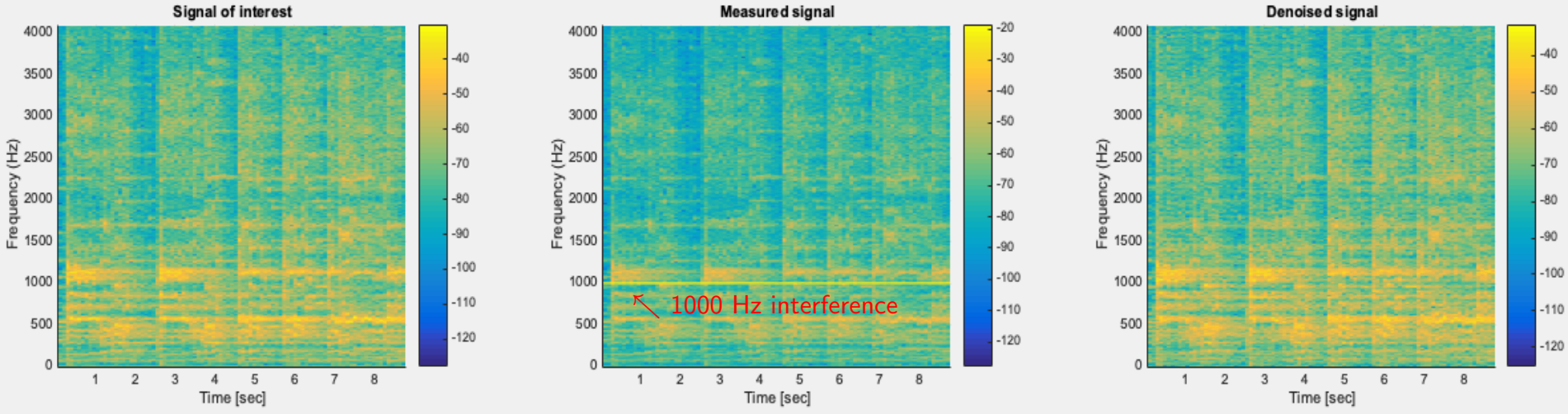
More in the Adaptive Signal Processing and Machine Intelligence course

Example 12: Acoustic feedback cancellation $\hookrightarrow$ role of the forgetting factor λ

Denoising_SLS_GUI.m

Top panels: Forgetting factor $\lambda = 0.9$





Bottom panels: Forgetting factor $\lambda = 0.995$

Lecture summary

- The method of least squares is extremely important for practical applications. Least Squares **does not mean** fitting a line to the data!
- **Do not need:** Any assumption on the PDF or any other statistics.
- **Do need:** The assumed signal model (which is deterministic). If the signal model is inaccurate, the LS estimator will be biased & not MVU.
- Principle of orthogonality $\perp\!\!\!\rightarrow$ underpins any subspace method.
- Method of LS is easy to implement and straightforward to interpret.
- Sequential solutions to the LS problem are very practical, while Weighted Least Squares allows us to assign "confidence" to samples, that is, to de–emphasise the contribution from unrealiable samples.
- We can also use a forgetting factor to deal with time-varying statistics.
- Established methods for dealing with outliers in data (see Appendix 9)
- A number of applications of LS theory: Factor models in finance, noise cancellation, Prony type spectral estimation, Recursive Least Squares, ...

Summary: Least Squares vs MLE

NB: The optimal MVU estimator and CRLB do not always exist or are impossible to find

Least Squares Estimator

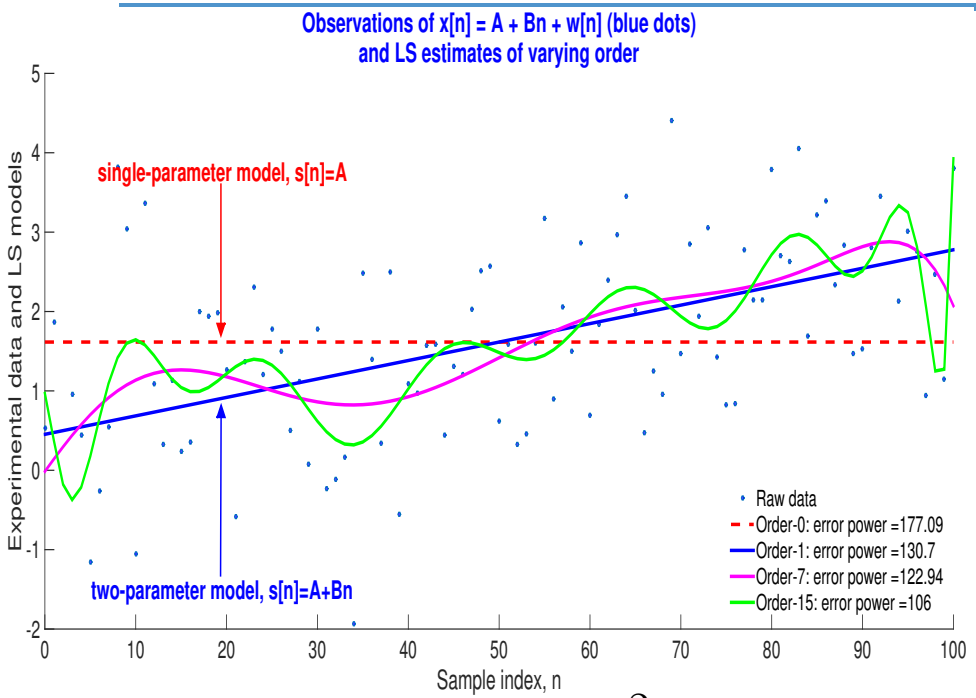
- LSE operates when the *pdf* of data is unknown; instead, it assumes a data model
- Quite intuitive and has rigorous geometrical interpretation
- No MVU guarantee, but tends to work well for $N > p$
- Recursive ways to calc. standard LSE, also admits sequential LS
- Allows for the incorporation of prior– and domain–knowledge (forgetting factor λ, uncertainty)
- Sliding window LS and λ–SLS can work on non–stationary data

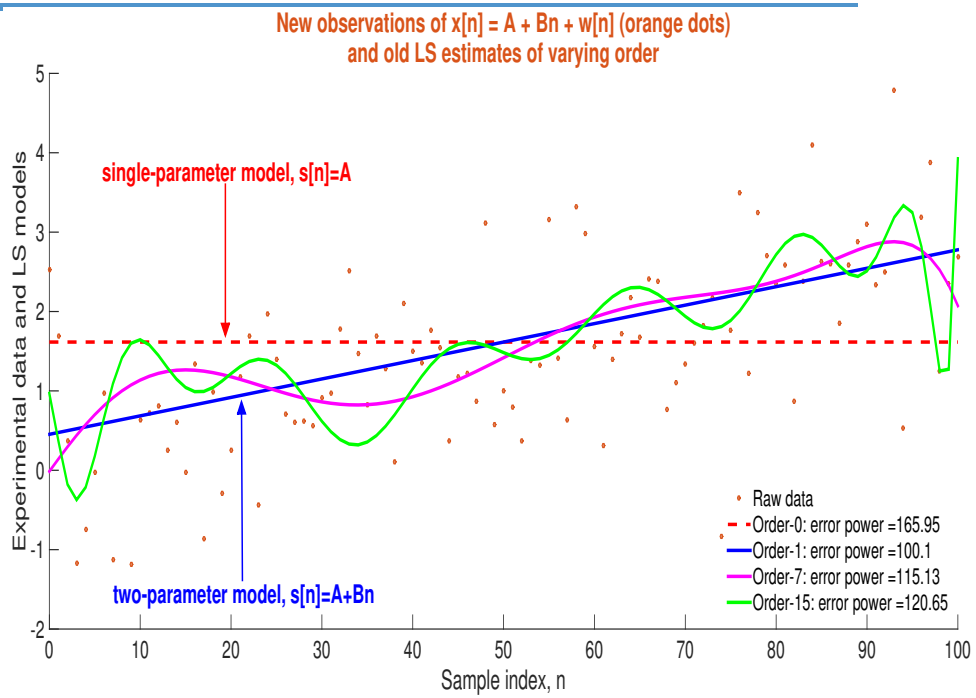
Maximum Likelihood Estimator

- Can always be applied once the *pdf* is assumed, and does not restrict the data model (*cf.* LSE)
- It is asymptotically optimal and MVU (for large data size)
- Can be computationally complex (numerical methods required)
- Often biased for small data size; no guarantee to obtain MVU
- Sensitive to outliers, can produce biased estimates in the presence of extreme events
- It is always possible to find an MLE, but it may be suboptimal

Appendix 1: Choosing the correct model order (see Slide 5)

Least_squares_overfitting.m





☞ The LS cost $J = \sum_i e_i^2$ is monotonically non–increasing with an increase in p. In our example: $J_0 = 177.09,\ J_1 = 130.7,\ J_7 = 122.94,\ J_{15} = 106,\ \ldots$

Reason: Model order $p = N$ defines a polynomial $a_0 + a_1 x + \cdots + a_N x^N$ which will perfectly fits N data points. **Warning: It also fits the noise!**

☞ Indeed, when these models are applied to unseen data (inference), the LS costs are $J_0 = 165.95,\ \mathbf{J_1 = 100.1},\ J_7 = 115.13,\ J_{15} = 120.65,\ \ldots$

In practice, increase order only if $J_{min}(p) - J_{min}(p-1) >$ user threshold

Appendix 2: Derivation of the MMSE and variance for the sequential estimator of a DC level in noise

$$J_{min}[N] = \sum_{n=0}^{N} \left(x[n] - \hat{A}[N]\right)^2 \qquad J_{min}[N-1] = \sum_{n=0}^{N-1} \left(x[n] - \hat{A}[N-1]\right)^2$$

$$= \sum_{n=0}^{N-1} \left[x[n] - \hat{A}[N-1] - \frac{1}{N+1}\left(x[N] - \hat{A}[N-1]\right)\right]^2 + \left(x[N] - \hat{A}[N]\right)^2$$

$$= J_{min}[N-1] - \frac{2}{N+1}\sum_{n=0}^{N-1}\left(x[n] - \hat{A}[N-1]\right)\left(x[N] - \hat{A}[N-1]\right)$$

$$+\frac{N}{\left(N+1\right)^2}\left(x[N] - \hat{A}[N-1]\right)^2 + \left(x[N] - \hat{A}[N]\right)^2$$

$$J_{min}[N] = J_{min}[N-1] + \frac{N}{N+1}\left(x[N] - \hat{A}[N-1]\right)^2$$

$$\mathrm{var}\left(\hat{A}[N]\right) = \frac{1}{\sum_{n=0}^{N}\frac{1}{\sigma_n^2}} = \frac{1}{\sum_{n=0}^{N-1}\frac{1}{\sigma_n^2} + \frac{1}{\sigma_N^2}} = \frac{1}{\frac{1}{\mathrm{var}(\hat{A}[N-1])} + \frac{1}{\sigma_N^2}}$$

$$= \frac{\mathrm{var}(\hat{A}[N-1])\ \sigma_N^2}{\mathrm{var}(\hat{A}[N-1]) + \sigma_N^2} = \left(1 - \frac{\mathrm{var}(\hat{A}[N-1])}{\mathrm{var}(\hat{A}[N-1]) + \sigma_N^2}\right)\mathrm{var}(\hat{A}[N-1])$$

$$= \left(1 - K[N]\right)\mathrm{var}(\hat{A}[N-1])$$

Appendix 3: Probability vs. Statistics

(for discrete RVs, $E\{X\} = \sum_{i=1}^{I} x_i P_X(x_i)$, where P_X is the probability function)

Probability: A data modelling view, describes how data **will likely behave**

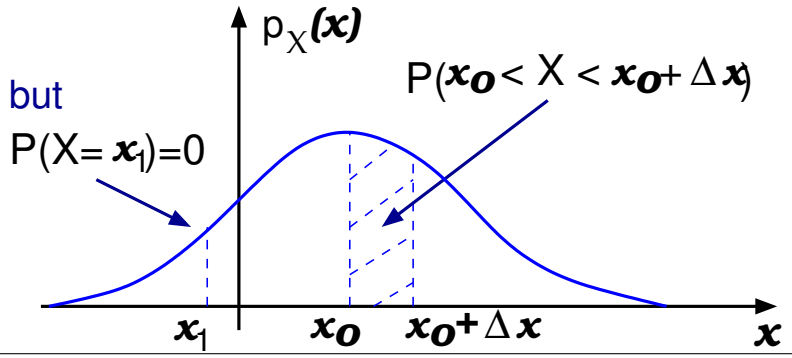
for example: $$average = E\{X\} = \int_{-\infty}^{\infty} x\, p_X(x)\, dx$$ no data here

Notice that there is no explicit mention of data here $\not\mapsto$ x is a dummy variable and p_X is the pdf of a random variable X.

Statistics: A data analysis view, determines how data **did behave**

for example: $$average = \frac{1}{N} \sum_{n=0}^{N-1} x[n]$$ no pdf here

Vagaries of probability: $P(x_0 < X < x_0 + \Delta x) = \int_{x_0}^{x_0+\Delta x} p_X(x) dx$



Notice that, for any x_1

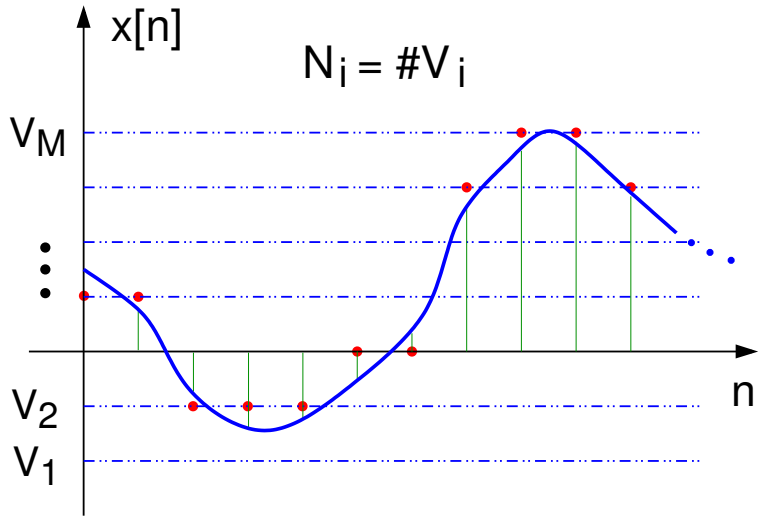
$$P(X = x_1) = 0$$

This appears odd, but otherwise the probabilities sum up to ∞

Appendix 3: Statistics vs. Probability, cont.

Statistical inference ↬ based on the observed data and supported by prob. theory

Vagaries of statistics: Consider N coarse-quantised data points, $x[0], \ldots, x[N-1]$. The quantised signal has $M \ll N$ possible amplitude values, $V_1, \ldots, V_M$, for which the corresponding relative frequencies are $N_1 = \#V_1, \ldots, N_M = \#V_M$. Calculate the mean, $\bar{x}$.



Solution:

$$\bar{x} = \frac{1}{N}\sum_{n=0}^{N-1} x[n] \quad = \quad \frac{1}{N}\sum_{m=1}^{M} V_m N_m \quad = \quad \sum_{m=1}^{M} V_m \underbrace{\frac{N_m}{N}}_{\approx\ P(x=V_m)}$$

☞ Clearly, the factor $1/N$ does not imply "uniform distribution"

Appendix 4: Statistical inference

Chinese for statistics is 统计 (summarizing & counting) and probability is 概率(论) ((theory of) randomness & chances),

Probability: Assumes perfect knowledge about the "population" of random data (through the pdf).

Typical question: There are 100 books on a bookshelf, 40 with red cover, 30 with blue cover, and 20 with green cover. What is the probability of randomly drawing a blue book from the shelf?

Statistics: No knowledge about the types of books on the shelf, we need to infer properties about the "population" based on random samples of "objects" on the shelf $\looparrowright$ **statistical inference**.

Typical question: A random sampling of 20 books from the bookshelf produced X red books, Y blue books and Z green books. What is the total proportion of red, blue, and green books on the shelf?

Statistical inference is applied in many different contexts under the names of: data analysis, data mining, machine learning, classification, pattern recognition, clustering, regression, classification

Appendix 5: Range of a matrix, span of a set of vectors

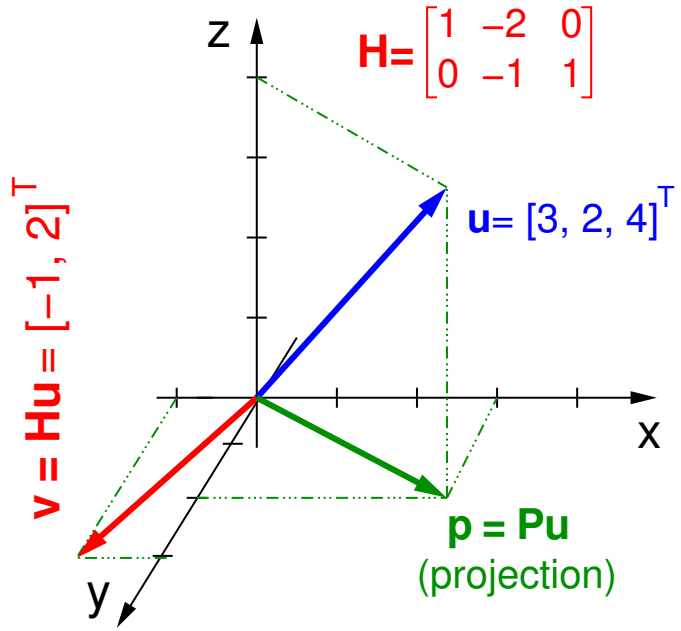
(a wide matrix transforms a vector space into another lower-dimensional one)

Consider a general 2×3 matrix $\mathbf{H}$ and a 3×1 vector $\mathbf{u}$

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \end{bmatrix} = [\mathbf{h}_1 \,|\, \mathbf{h}_2 \,|\, \mathbf{h}_3] \quad \text{where} \quad \mathbf{h}_i = \begin{bmatrix} h_{1i} \\ h_{2i} \end{bmatrix} \; i = 1, 2, 3$$

Then,

$$\mathbf{v} = \mathbf{H}\,\mathbf{u} = [\mathbf{h}_1 \,|\, \mathbf{h}_2 \,|\, \mathbf{h}_3] \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = u_1\mathbf{h}_1 + u_2\mathbf{h}_2 + u_3\mathbf{h}_3 \; \in \mathbb{R}^{2\times 1}$$



Example: $\mathbf{H} \in \mathbb{R}^{2\times 3}, \mathbf{u} \in \mathbb{R}^{3\times 1}$

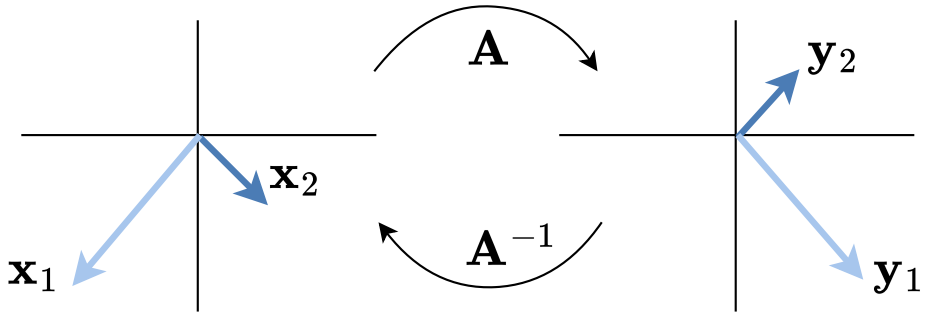
- Clearly, $\mathbf{v}$ is a linear combination of the columns of the matrix $\mathbf{H}$, $\mathbf{h}_i \in \mathbb{R}^{2\times 1}$
- Vector $\mathbf{v} = [-1, 2]^T$ therefore lies in the span of the columns of $\mathbf{H}$, i.e. in $\mathbb{R}^2$

☞ This dimensionality reduction is not a projection $\mathbf{p} = \mathbf{P}\mathbf{u}$, where $\mathbf{P} = \mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$

Appendix 5b: Orthogonal vs Non-Orthogonal mappings

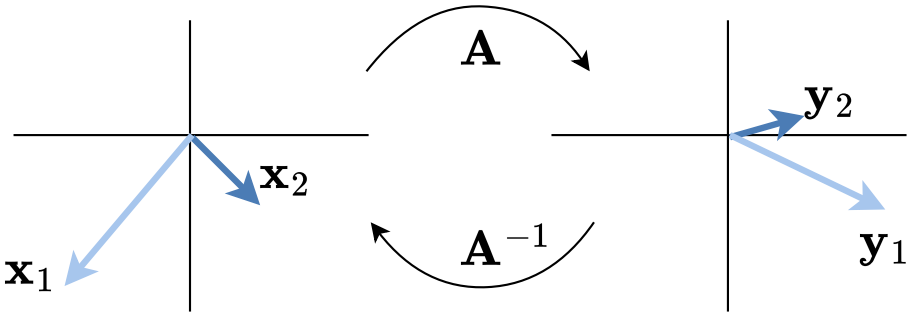
Orthogonal transformations preserve information $\not\mapsto$ useful in Principal Component Analysis (PCA), various embeddings, and optimisation.

(a) Orthogonal mappings ($\mathbf{A}^T\mathbf{A} = \mathbf{I}$):
$||\mathbf{Ax}|| = ||\mathbf{x}||, \langle \mathbf{Ax}, \mathbf{Ay} \rangle = \langle \mathbf{x}, \mathbf{y} \rangle$



✅ **Preserves**:
(1) vectors norms; (2) angles between vectors

(b) Non-Orthogonal mappings ($\mathbf{A}^T\mathbf{A} \neq \mathbf{I}$):
$||\mathbf{Ax}|| \neq ||\mathbf{x}||, \langle \mathbf{Ax}, \mathbf{Ay} \rangle \neq \langle \mathbf{x}, \mathbf{y} \rangle$



⚠ **Distorts**:
(1) vectors norms; (2) angles between vectors

The orthogonal mapping above, $\mathbf{A}$, rotates the original vectors $\mathbf{x}_1, \mathbf{x}_2$ by $\pi/2$, and the geometry of the new space, spanned by $\mathbf{y}_1, \mathbf{y}_2$ is preserved.

$$\mathbf{A} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad \mathbf{A}^{-1} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{A}\mathbf{A}^{-1} = \mathbf{A}\mathbf{A}^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \mathbf{I}$$

Appendix 6: Quadratic forms and positive–(semi)definite matrices

Quadratic forms appear often in data analysis, and are expressed as

$$\mathbf{x}^T\mathbf{H}\mathbf{x} \qquad \mathbf{x} \in \mathbb{R}^{N\times 1},\ \mathbf{H} \in \mathbb{R}^{N\times N}$$

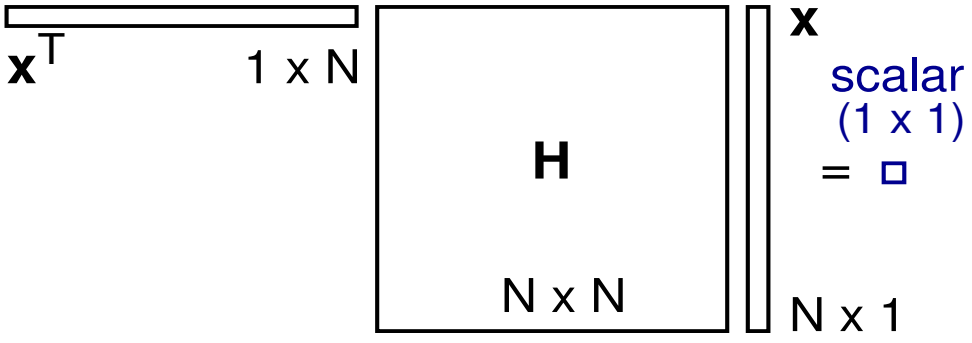
For simplicity, consider a 2nd order case, where

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \mathbf{H} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix}$$

$\uparrow$ *variable vector* $\quad\uparrow$ *fixed matrix*

The quadratic form $Q_{\mathbf{H}}(\mathbf{x}) = Q_{\mathbf{H}}(x_1, x_2)$ of a matrix $\mathbf{H}$ is a scalar given by

$$Q_{\mathbf{H}}(x_1, x_2) = \mathbf{x}^T\,\mathbf{H}\,\mathbf{x} = \sum_{i=1}^{2}\sum_{j=1}^{2} h_{ij}x_i x_j = h_{11}x_1^2 + h_{22}x_2^2 + (h_{12}+h_{21})x_1x_2$$



- If $Q_{\mathbf{H}}(\mathbf{x}) \geq 0$, for any $\mathbf{x} \neq \mathbf{0}$ then the matrix $\mathbf{H}$ is called positive semi-definite
- The matrix $\mathbf{H}$ is positive definite if $Q_{\mathbf{H}}(\mathbf{x}) > 0, \forall \mathbf{x} \neq \mathbf{0}$

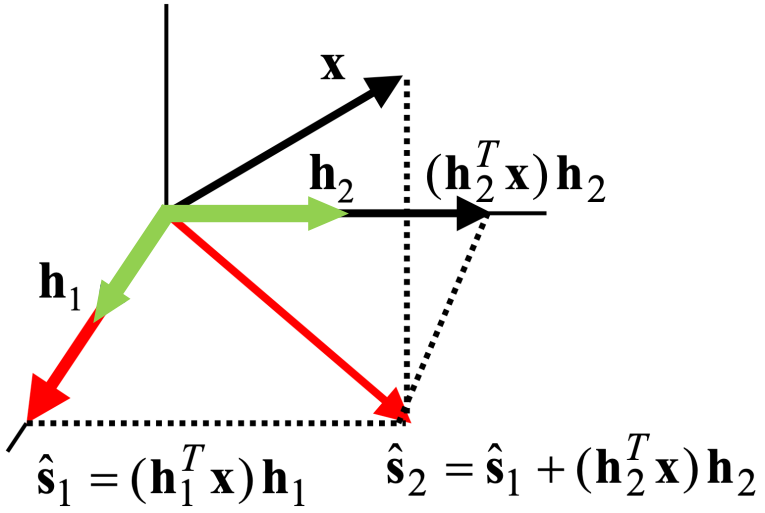
Appendix 7: Order Recursive Least Squares (ORLS)

(If $\mathbf{h}_i$ are NOT $\perp$ ORLS is harder but possible)

For orthonormal columns of $\mathbf{H}$,

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

Denote by θ_i the projections on the individual columns of $\mathbf{H}$ (coordinates in S). Then, we can find projections on each of those 1D subspaces separately, and add them to give



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

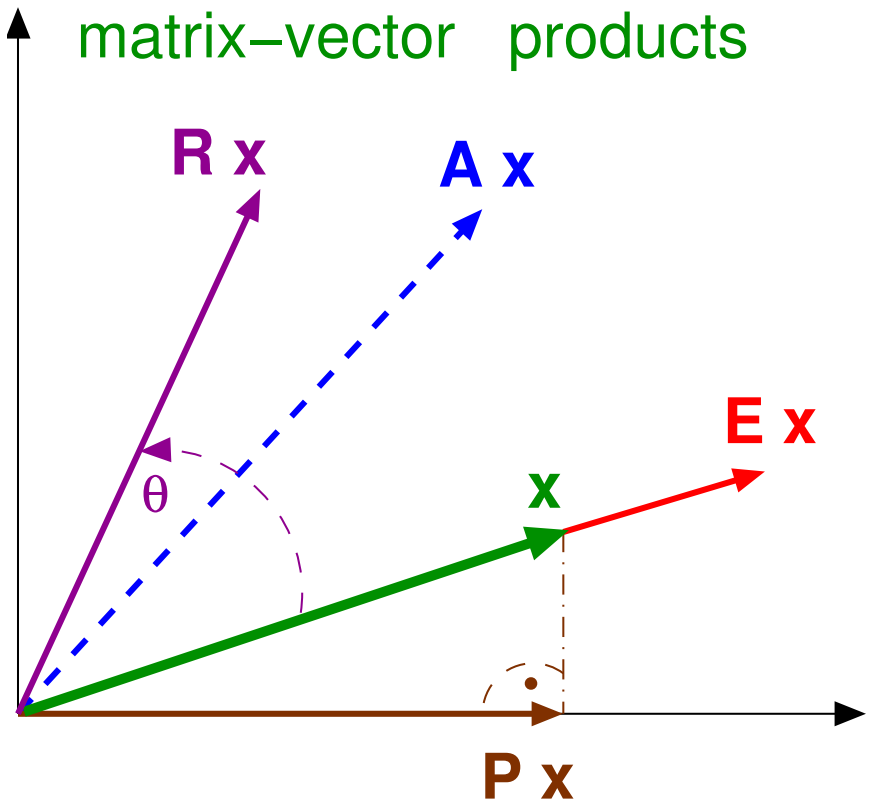
☞ **Can we use an p-order model to compute the $(p+1)$-order model?**

Indeed, denote by $\mathbf{H}_1 = \mathbf{h}_1, \quad \mathbf{H}_2 = \left[\mathbf{h}_1 \,|\, \mathbf{h}_2\right] \quad \cdots \quad \mathbf{H}_{p+1} = \left[\mathbf{H}_p \,|\, \mathbf{h}_{p+1}\right]$

For $p=1 \rightarrow \hat{s}_1 = (\mathbf{h}_1^T\mathbf{x})\mathbf{h}_1$ For $p=2 \rightarrow \hat{s}_2 = (\mathbf{h}_1^T\mathbf{x})\mathbf{h}_1 + (\mathbf{h}_2^T\mathbf{x})\mathbf{h}_2 = \hat{\mathbf{s}}_1 + (\mathbf{h}_2^T\mathbf{x})\mathbf{h}_2$

Order Recursive Least Squares: $\hat{\mathbf{s}}_{p+1} = \hat{\mathbf{s}}_p + (\mathbf{h}_{p+1}^T\mathbf{x})\mathbf{h}_{p+1}$

Appendix 8: How does a matrix transform a vector?



Ampli-twist: A matrix $\mathbf{A}$ which multiplies a vector $\mathbf{x}$
(i) stretches or shortens the vector
(ii) rotates the vector

A $\rightsquigarrow$ any general matrix

R $\rightsquigarrow$ a rotation matrix ($\mathbf{R}^T = \mathbf{R}^{-1}$ and $\det \mathbf{R} = 1$)

$\mathbf{Ex} = \lambda \mathbf{x}$ $\rightsquigarrow$ eigenanalysis

P $\rightsquigarrow$ projection matrix

An example of a rotation matrix

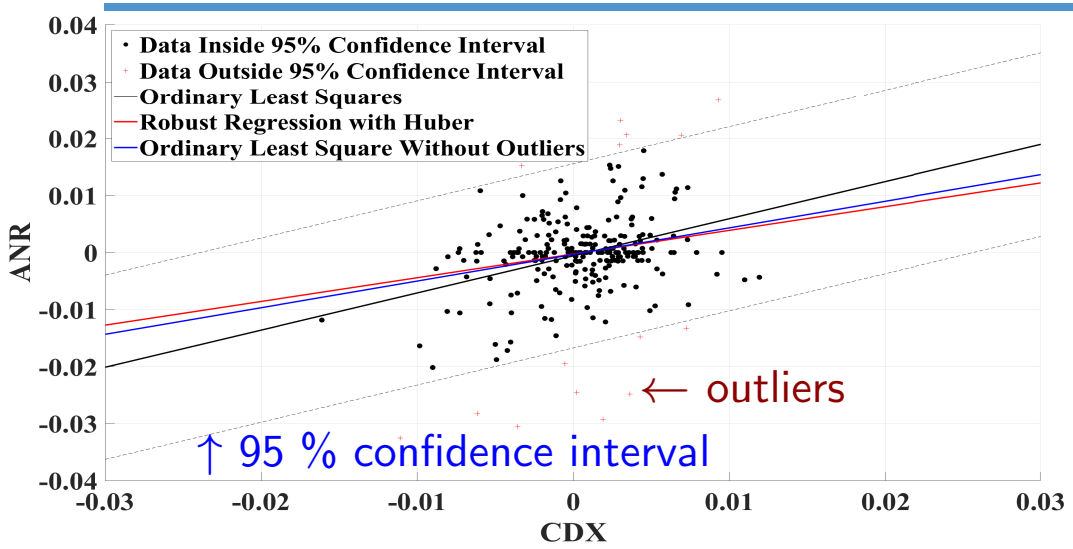
$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

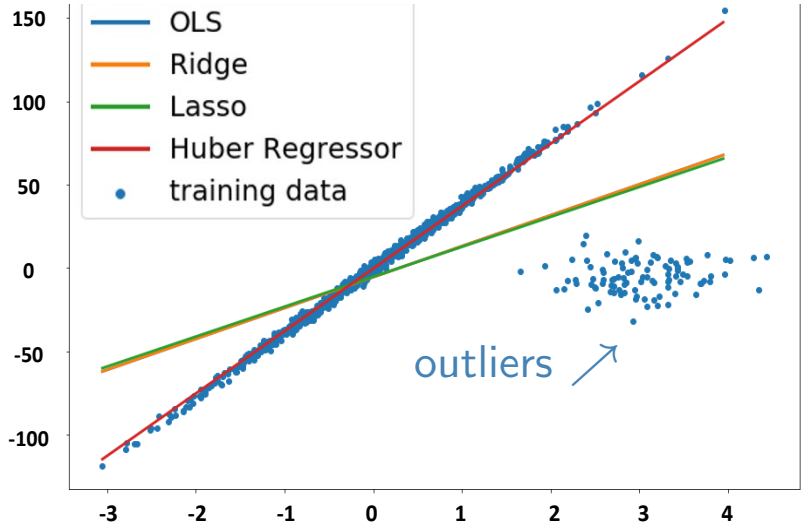
What can we say about the properties of the matrix $\mathbf{A}$, matrix $\mathbf{E}$ and the projection matrix $\mathbf{P}$ (rank, invertibility, ...)?

Is the projection matrix invertible?

Appendix 9: Sensitivity to outliers of the ordinary Least Squares (OLS)

(role of regularisation and robust estimators)





Regression of daily returns of Altona Energy (ANR) corporate bond on the credit default swap (CDX).

Regression under outliers

Huber estimator is robust

Ridge regression:

$$\mathcal{J}_n(\mathbf{w}) = \underbrace{(d_n - \mathbf{w}_n^T \mathbf{x}_n)^2}_{\text{standard cost}} + \underbrace{\lambda_1 \|\mathbf{w}_n\|_2^2}_{L_2 \text{ penalty}} = e_n^2 + \lambda_1 \mathbf{w}_n^T \mathbf{w}_n$$

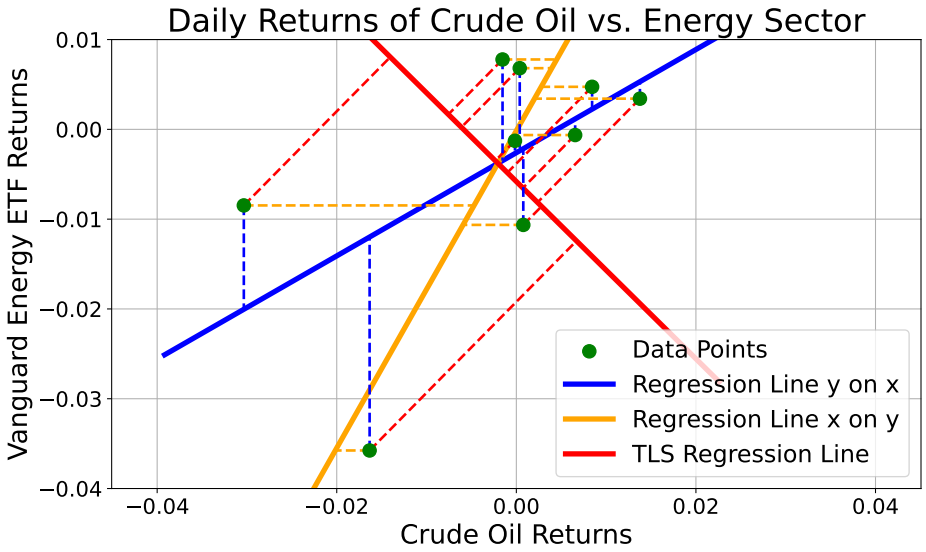
LASSO (sparsity promoting):

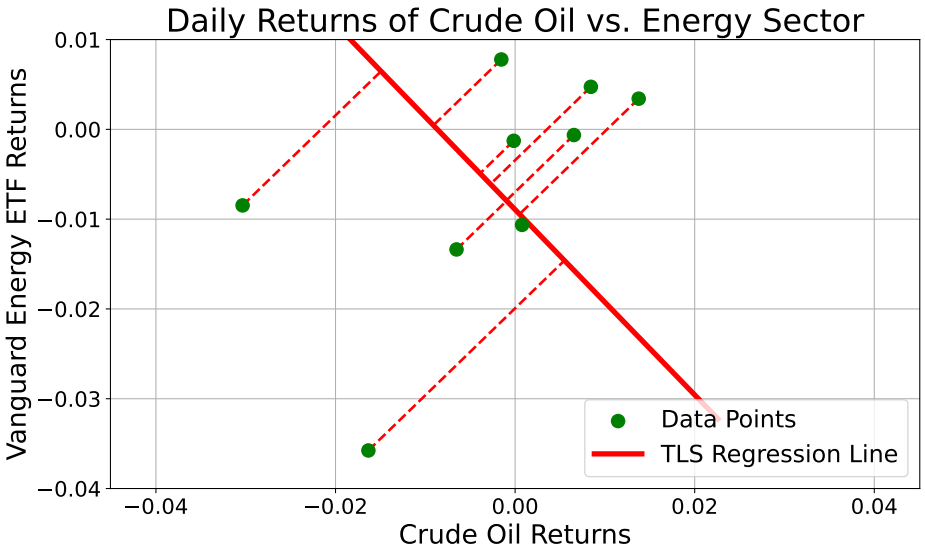
$$\mathcal{J}_n(\mathbf{w}) = \underbrace{(d_n - \mathbf{w}_n^T \mathbf{x}_n)^2}_{\text{standard cost}} + \underbrace{\lambda_2 \|\mathbf{w}_n\|_1}_{L_1 \text{ penalty}}$$

- Ridge: Penalises for large weights (but does not reduce system dimensionality)
- Least absolute shrinkage and selection operator (LASSO) enforces insignificant weights to go to zero, and thus promotes sparsity and aids **interpretability** ($\lambda_1, \lambda_2 \hookrightarrow$ param's.)

Appendix 9a: The Total Least Squares (TLS) method

Instead of the "vertical distance" (regression of y onto x), we can also use the "shortest distance" (orthogonal projection) between the observed data and the regression line $\looparrowright$ the method of Total Least Squares (TLS).





The projection operator (see Lecture 6), that is, modelling based on the 'orthogonal distance', is more complicated than the modelling based on the 'vertical' or 'horizontal' distance but generally yields more accurate models.

In the 2D case, the TLS regression line is equivalent to the first principal component of the data.

Appendix 10: A note on CAPM and Factor Models

Progression from one-factor intuition → multi-factor realism → modern finance

- W. Sharpe's seminal article *"Capital Asset Prices: A Theory of Market Equilibrium under Risk"* was initially rejected in The Journal of Finance
- One of the reviewers commented that the *"assumptions are so preposterous that all subsequent conclusions are uninteresting"*
- Finally accepted in 1964, this article laid the foundations for CAPM, earning Sharpe the Nobel Prize in Economics in 1990
- Footnote 22 in this article has 17 lines of equations (the footnote that won a Nobel Prize)
- CAPM is usually pronounced *cap-em*, but W. Sharpe prefers *C-A-P-M*
- In 1992, Eugene Fama and Kenneth French introduced the "*three–factor model*", a generalisation of CAPM from 1 to 3 factors
- Eugene F. Fama was awarded the Nobel Prize in Economics in 2013
- In 2015, Fama and French introduced the "five-factor model"
- CAPM can explain up to 70% of diversified portfolio returns, the 3-factor model up to 90%, and the 5-factor model even more

Notes:

○

Notes:

○

Notes:

○