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

- To introduce the concept of least squares estimation (LSE)
- Establish parallels with the ML estimation, BLUE, MVUE, and CRLB
- Geometry of LS: The signal, noise, and measurement subspaces
- Show how to exploit the orthogonality of the signal space and the estimation error
- Linear least squares, nonlinear least squares, separable least squares, constrained least squares, order recursive least squares
- Move from block-based estimation to estimation based on streaming data: Sequential least squares, link with state space models
- Weighted least squares, confidence levels in data samples
- Practical applications

The method of Least Squares

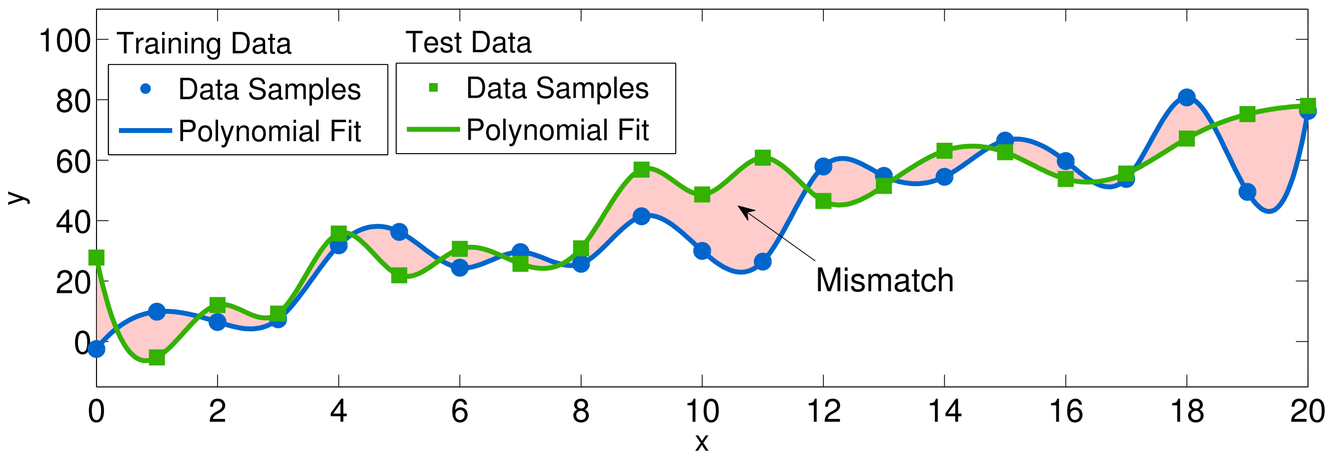
This class of estimators has, generally, no optimality properties

- But, do we necessarily desire optimality ↬ an optimal estimator may be mathematically intractable or computationally too complex
- Makes good sense for many practical problems ↬ this dates back to Gauss who in 1795 introduced the method to study planetary motions
- **LS is not statistically based** ↬ no probabilistic assumptions are made about the data, no need for a pdf model
- We only need to assume a deterministic signal model
- Usually easy to implement, either in a block–based or sequential manner, 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

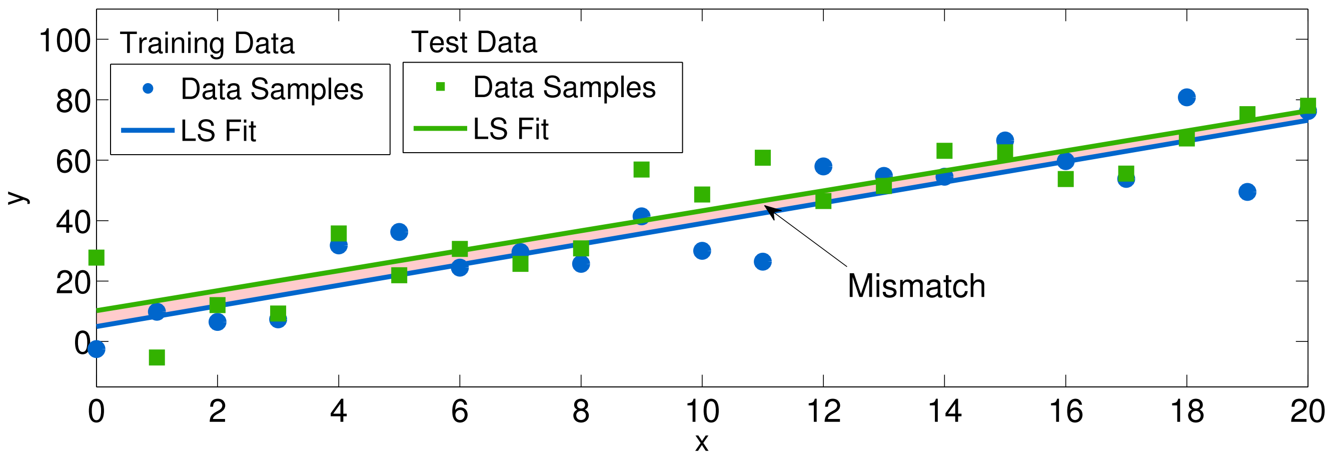
Motivation: A simpler model often generalises better

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

Curve Fitting using Piecewise Polynomials



Curve Fitting using Linear Least Squares (LS)

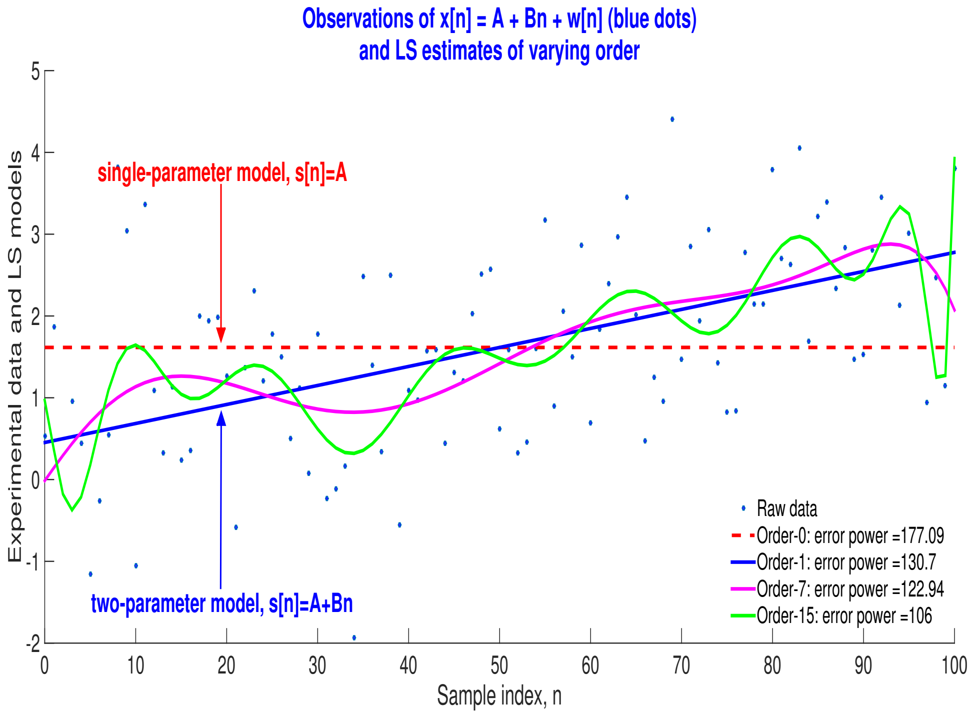


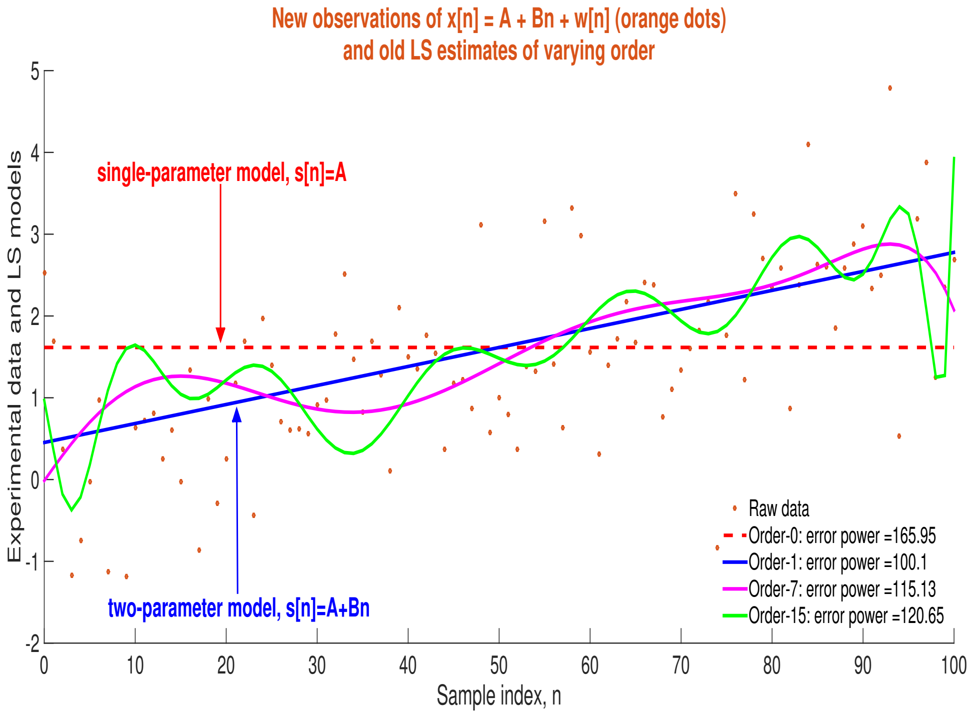
☞ **Observe the usefulness of a model over an exact fit!** Least_Squares_Order.m

But, careful with over–fitting

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)



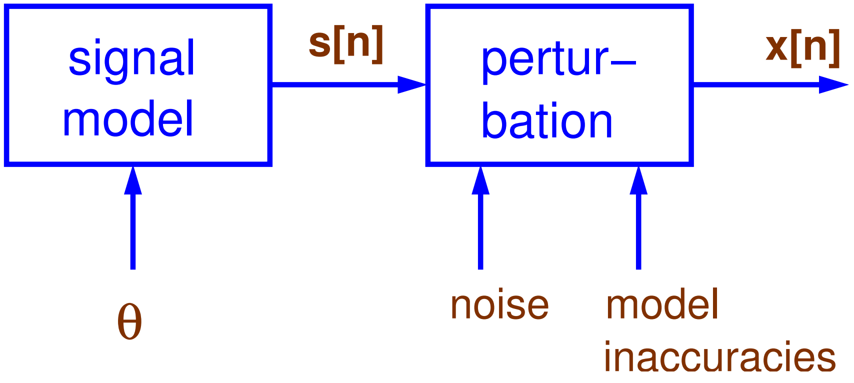


☞ Order–7 and Order–15 Least Squares (LS) fits to the data gave a lower “within–sample” error power than the correct Order–1 fit (122.9 and 106 versus 130.7) (left panel)

☞ But this leads to over–fitting, i.e. worse extrapolation (prediction) 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 purely deterministic, generated by a model which depends upon an unknown parameter θ or a vector parameter $\boldsymbol{\theta}$.



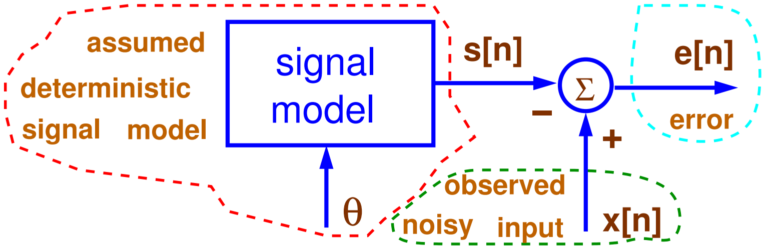
Least squares data model

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

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

No probabilistic assumptions ☺

Only signal model assumed $\looparrowright$ wide range of applications



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

$$\text{LSE}: \qquad \min_{\theta} J(\theta) \qquad \text{(our objective)}$$

The LS estimator of the unknown parameter θ finds the value of θ that makes the model output $s[n]$ closest to the observed data $x[n]$; the closeness is measured by the LS error criterion (error power)

Example 1: DC Level in WGN

Our old example: DC level in WGN (in MLE, we needed a pdf!)

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

Measurement model: $x[n] = s[n] + q[n] = A + q[n], \quad q[n] \looparrowright$ any noise

LSE formulation:

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

LSE solution:

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

$$\text{the LS estimator}: \qquad \hat{A} = \frac{1}{N}\sum_{n=0}^{N-1} x[n]$$

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 $\boldsymbol{x[n] = A + q[n] + q'[n]}$

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

☞ 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 **deterministic signal** (described by a 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

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

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 use the LS method, as A is "linear in the data"
- ○ 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 the minimisation of J over f_0 only **(separable least squares)**.

Geometric interpretation & Example: Fourier analysis

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

Example 2: Consider a sig. 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, b.

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

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

☞ We must assume that $\mathbf{H}$ is full rank 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 $\boldsymbol{\varepsilon} = \mathbf{x} - \mathbf{s}$

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

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

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

Geometric interpretation ↬ continued

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

This can be rewritten in a more elegant form.

Recall that the Euclidean length $\| \cdot \|_2$ of an $N \times 1$ 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 (recall that $\| \mathbf{a} - \mathbf{b} \|$ is the distance between the vectors $\mathbf{a}$ and $\mathbf{b}$)

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

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

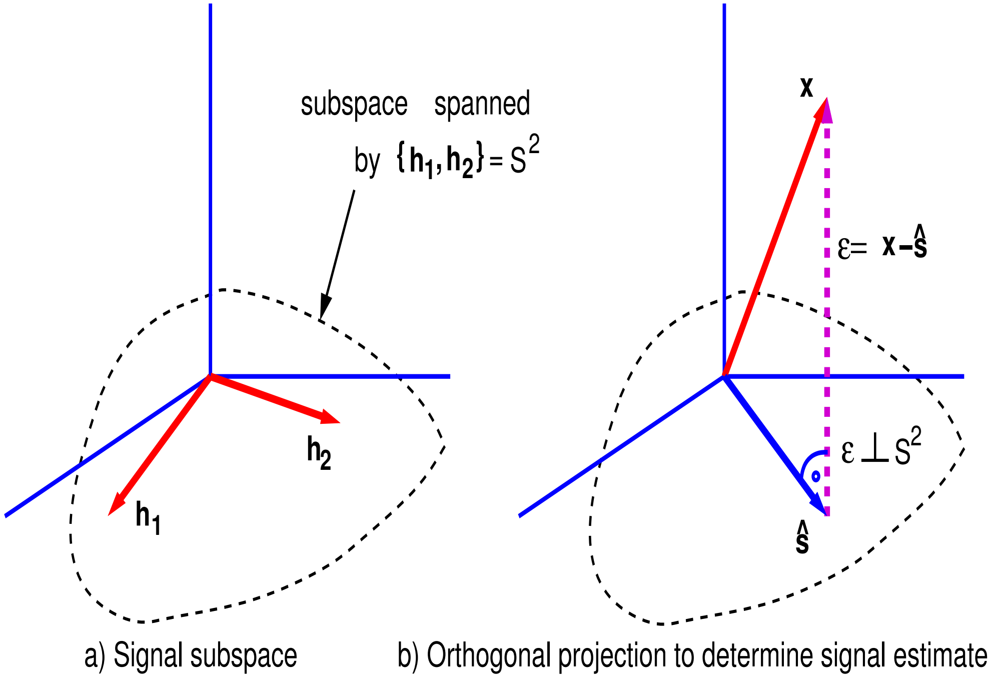
$$\hat{\mathbf{s}} = \textstyle\sum_{i=1}^{p} \hat{\theta}_i \mathbf{h}_i$$

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

Geometry of LSE: Vector space projections

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

The vector $\mathbf{x} \in \mathbb{R}^{N\times 1}$, however, all signal vectors must lie in a p-dimen. subspace of $S^p \subset \mathbb{R}^N$. 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, $\hat{\mathbf{s}} = \mathbf{P}\mathbf{x}$, **P** ↬ projection matrix.

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

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

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

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

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

Finally: LS solution (through geometry, no derivatives)

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

Letting $$\mathbf{s} = \theta_1 \mathbf{h_1} + \theta_2 \mathbf{h_2} = \mathbf{H}\boldsymbol{\theta}$$

the conditions (a) and (b) from the previous slide, we have

$$\begin{aligned} \left(\mathbf{x} - \theta_1\mathbf{h_1} - \theta_2\mathbf{h_2}\right)^{\mathbf{T}}\mathbf{h_1} &= 0 \qquad \equiv \qquad \boldsymbol{\varepsilon}^T\mathbf{h}_1 = 0 \\ \left(\mathbf{x} - \theta_1\mathbf{h_1} - \theta_2\mathbf{h_2}\right)^{\mathbf{T}}\mathbf{h_2} &= 0 \qquad \equiv \qquad \boldsymbol{\varepsilon}^T\mathbf{h}_2 = 0 \end{aligned}$$

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

from $\boldsymbol{\varepsilon}^T\mathbf{H} = \mathbf{0}^T$ we have $\mathbf{H}^T\boldsymbol{\varepsilon} = \mathbf{0}$ so that $\mathbf{H}^T(\mathbf{x} - \mathbf{H}\boldsymbol{\theta}) = \mathbf{0}$

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

We can now solve for the unknown vector parameter, $\boldsymbol{\theta}$, to yield the **Least Squares Estimate (LSE)**

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

where $\mathbf{H}$ is the $(N \times p)$-dimensional measurement (observation) matrix.

Remark: Benefits of having orthogonal columns of $\mathbf{H}$

The Least Squares estimator finds the coefficient vector in the form

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

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

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

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

☞ **Easy, no inversion needed!** Also, for every unknown parameter, $\theta_i = \mathbf{h}_i^T\mathbf{x}$

☞ $\mathbf{h}_i^T\mathbf{x}$ is a projection of the observed data $\mathbf{x}$ onto each column of $\mathbf{H}$

Example 2: Fourier analysis $\looparrowright$ continued

For more detail see Example 9 in Lecture 4

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)

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

$$\text{while} \qquad \mathbf{h}_1^T \mathbf{h}_1 = \frac{N}{2} \qquad \mathbf{h}_2^T \mathbf{h}_2 = \frac{N}{2}$$ (not orthonormal)

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 $\mathbf{H}$ are not orthogonal, and the signal estimate

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

Linear least squares in a nutshell

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

$$\begin{aligned} J(\boldsymbol{\theta}) &= \sum_{n=0}^{N-1} \left(x[n] - s[n,\theta]\right)^2 = \Big(\mathbf{x} - \underbrace{\mathbf{H}\boldsymbol{\theta}}_{\hat{\mathbf{s}}}\Big)^T \Big(\mathbf{x} - \underbrace{\mathbf{H}\boldsymbol{\theta}}_{\hat{\mathbf{s}}}\Big) \\ &= \mathbf{x}^T\mathbf{x} - 2\mathbf{x}^T\mathbf{H}\boldsymbol{\theta} + \boldsymbol{\theta}^T\mathbf{H}^T\mathbf{H}\boldsymbol{\theta} \qquad (\mathbf{H} \text{ is full rank}) \end{aligned}$$

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

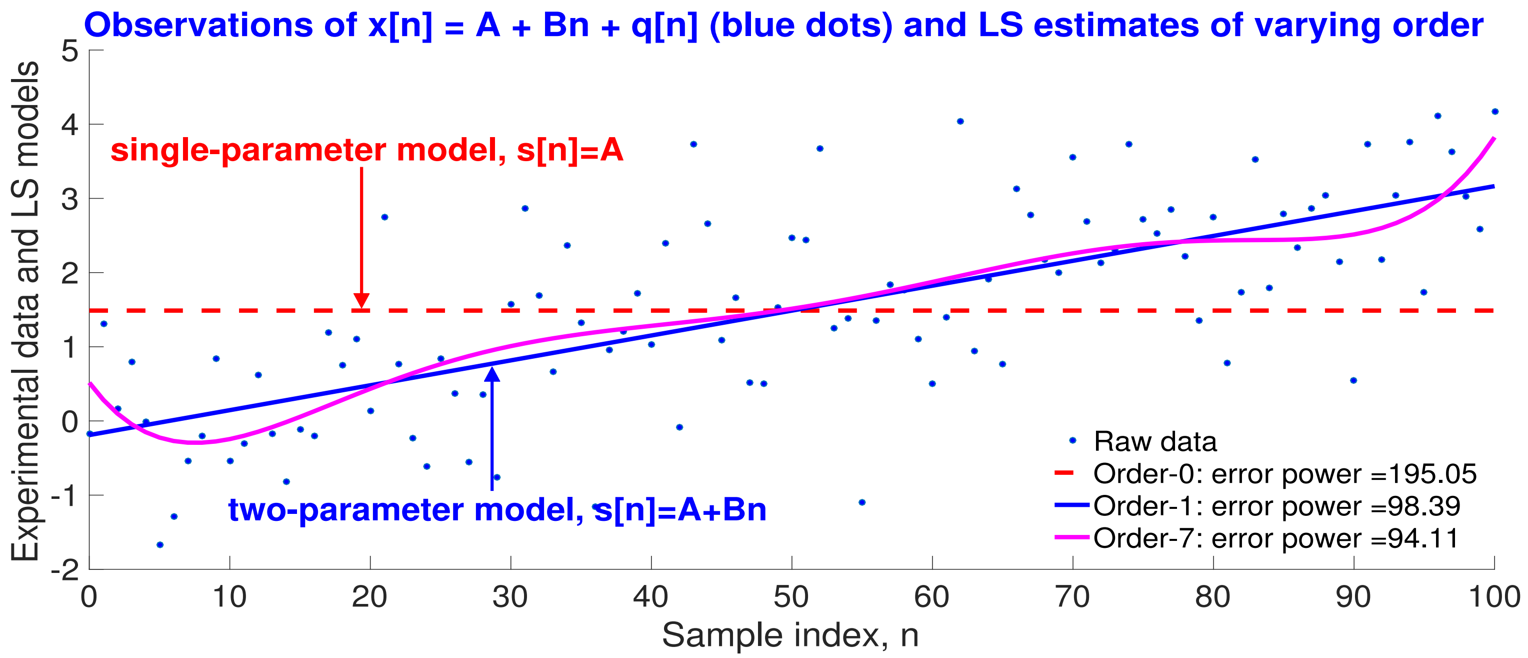
2. The minimum LS cost (replace $\hat{\boldsymbol{\theta}}$ into $J(\boldsymbol{\theta})$ above) is therefore

$$J_{min} = J(\hat{\boldsymbol{\theta}}) = \mathbf{x}^T\Big[\mathbf{I} - \mathbf{H}\underbrace{\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\Big]\mathbf{x}}_{\boldsymbol{\theta}} = \mathbf{x}^T\underbrace{\Big(\mathbf{x} - \underbrace{\mathbf{H}\boldsymbol{\theta}}_{\hat{\mathbf{s}}}\Big)}_{\boldsymbol{\varepsilon}} = \mathbf{x}^T\mathbf{x} - \mathbf{x}^T\mathbf{H}\boldsymbol{\theta} \qquad (\leq \| \mathbf{x} \|_2^2)$$

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

think of the AR process order $x(n) = a_1x(n-1) + \cdots + a_px(n-p) + q(n)$

Follows naturally from the problem of fitting a polynomial to the data (recall the Weierstrass theorem $\hookrightarrow$ 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
- The choice $p = N$ is a perfect fit to the data, but this way we also fit the noise (see the previous slide and also Slide 4)
- Recall the MDL and AIC in AR modelling $\hookrightarrow$ 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 also use an order-recursive LS algorithm to compute the model of order $(p+1)$ from the model of order p**

Weighted Least Squares (WLS)

see also Example 5 in Lecture 5, and Quadratic Forms in the Appendix here

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

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

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.

Opportunities in practical applications ↬ 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 ↬ 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 ↬ parameter transformation.** **Example:** Consider a nonlinear problem of estimating the amplitude and phase of a sinusoid $\quad s[n] = A\cos(\omega n + \phi), \quad n = 0, \ldots, N-1$

⇝ 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 $\hat{\alpha} = \big(\mathbf{H}^T\mathbf{H}\big)^{-1}\mathbf{H}^T\mathbf{x}$ **(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	$\mathbf{x} = \mathbf{H}\boldsymbol{\theta} + \boldsymbol{\varepsilon}$	no probabilistic assumptions	$\hat{\boldsymbol{\theta}}_{ls} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{x}$
BLUE	$\mathbf{x} = \mathbf{H}\boldsymbol{\theta} + \mathbf{q}$	q is white with 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 know *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 $\mathbf{P}^2 = \mathbf{P}$, it is symmetric and **singular** with rank p (many $\mathbf{x}(n)$ can have the same projection).

Sequential least squares

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 therefore need to obtain a sequence of LS estimators over time.

The problem:

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 solution $\hat{\boldsymbol{\theta}}_{N-1}$ sequentially, based only on $\hat{\boldsymbol{\theta}}_{N-1}$ and $x[N]$, that is

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

Example 4: DC level in uncorrelated zero mean noise

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

Consider the problem of estimating the DC level in noise, for which we have obtained the LSE

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

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

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

$$\hat{A}[N] = \frac{N}{N+1}\hat{A}[N-1] + \frac{1}{N+1}x[N]$$ ↬ **a recursive estimate!**

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

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

upon arrival of $x[N]$, re-arrange

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

Example 4: DC level in noise ↬ 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

$$\begin{aligned} \hat{A}[N] &= \hat{A}[N-1] + \frac{1}{N+1}\Big(x[N] - \hat{A}[N-1]\Big) \\ \textbf{new estimate} &= \textbf{old estimate} + \underbrace{\textbf{gain} \times \textbf{error}}_{\textbf{correction}} \end{aligned}$$

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

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

☞ 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.

Example 5: Weighted LS for the estimation of DC level in noise in a sequential form (see Example 9 in Lecture 4 & Slide 16)

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 choose $w_n = 1/\sigma_n^2$, to give[1]

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

Its corresponding sequential form then becomes

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

or **new estimate** $=$ **old estimate** $+$ **gain** $\times$ **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 $[\mathbf{W}]_{ii} = \frac{1}{\sigma_i^2}$.

Some observations about weighted LS

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

Two extreme cases:

- ○ If $\sigma_N^2 \rightarrow \infty$, i.e. the new sample is extremely noisy, then we do not correct the previous LSE
- ○ If $\sigma_N^2 \rightarrow 0$, that is, the new sample is noise–free, then $\hat{A} \rightarrow x[N]$, and we discard all the previous samples

☞ 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

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

Weighted LS: Recursive calculation of gain and variance

- The gain for the N-th update can be written 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

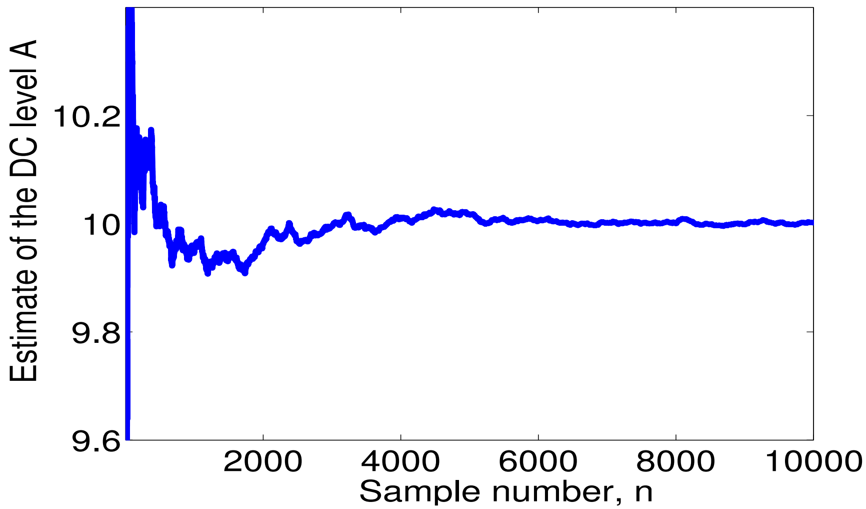
Estimator update: $\hat{A}[N] = \hat{A}[N-1] + K[N]\Big(x[N] - \hat{A}[N-1]\Big)$

$$\text{where} \quad K[N] = \frac{var\big(\hat{A}[N-1]\big)}{var\big(\hat{A}[N-1]\big) + \sigma_N^2}$$

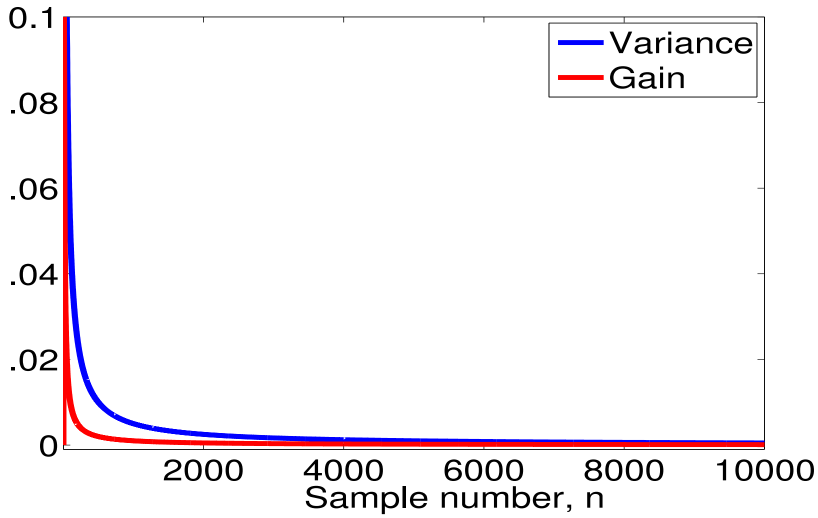
Variance update: $var\big(\hat{A}[N]\big) = \big(1 - K[N]\big)\, var\big(\hat{A}[N-1]\big)$

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

Example 6: 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

Consider a gain a noisy line (slides 26 & 27, Lecture 4)

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

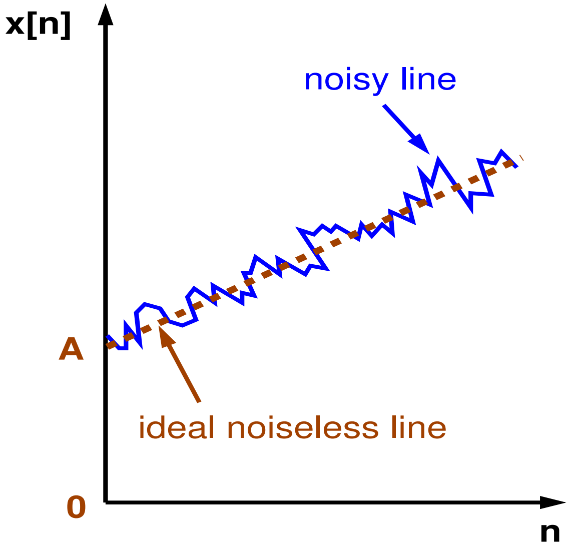
where $\mathbf{x} = [x_0, x_1, \ldots, x_{N-1}]^T$, $\mathbf{q} = [q_0, q_1, \ldots, q_{N-1}]^T$, and $\boldsymbol{\theta} = [A \quad B]^T$

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

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



For $N+1$ data point

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

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 $\mathbb{H}$ grows with time.

- **Estimator update:**

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

where the **gain factor** is given by

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

- **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 7: Sequential LS for the parameters of a line

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

- We model $\mathbf{x}[n] = A + Bn + \mathbf{q}[n]$, then 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):**

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

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

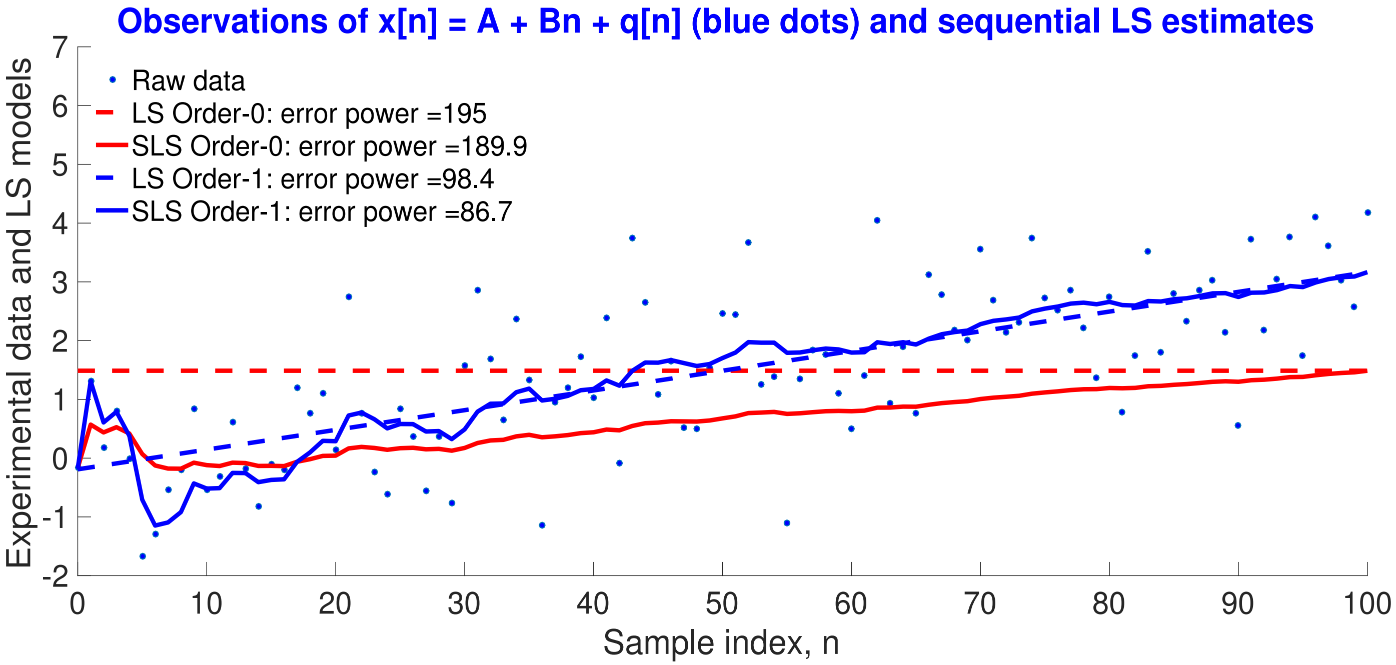
$$\mathbf{C}[n] = \Big(\mathbf{I} - \mathbf{K}[n]\mathbf{h}^T[n]\Big)\mathbf{M}[n]$$

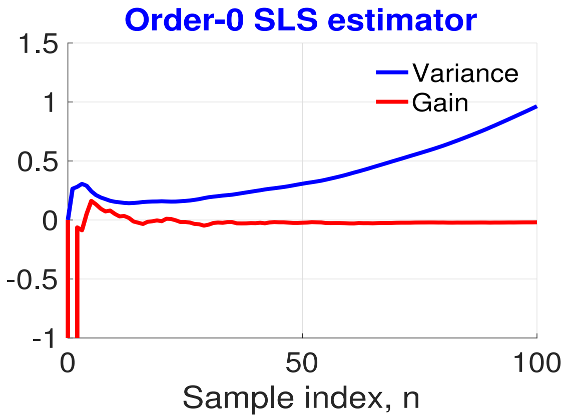
- The **gain factor** is updated as $\mathbf{K}[n] = \begin{bmatrix} \frac{2(2n-1)}{n(n+1)} \\ \frac{6}{n(n+1)} \end{bmatrix}$

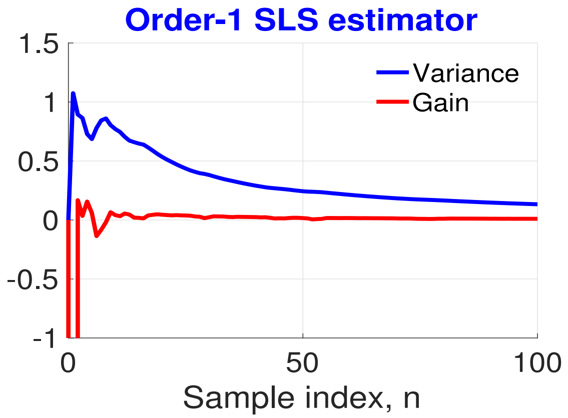
  and the **covariance matrix** as $\mathbf{C}[n] = \begin{bmatrix} \frac{2(2n-1)}{n(n+1)}\sigma_n^2 & 0 \\ 0 & \frac{12}{n(n^2+1)}\sigma_n^2 \end{bmatrix}$

Example 7: Continued

Matlab: Sequential_LS_Order_Interactive_Local.m





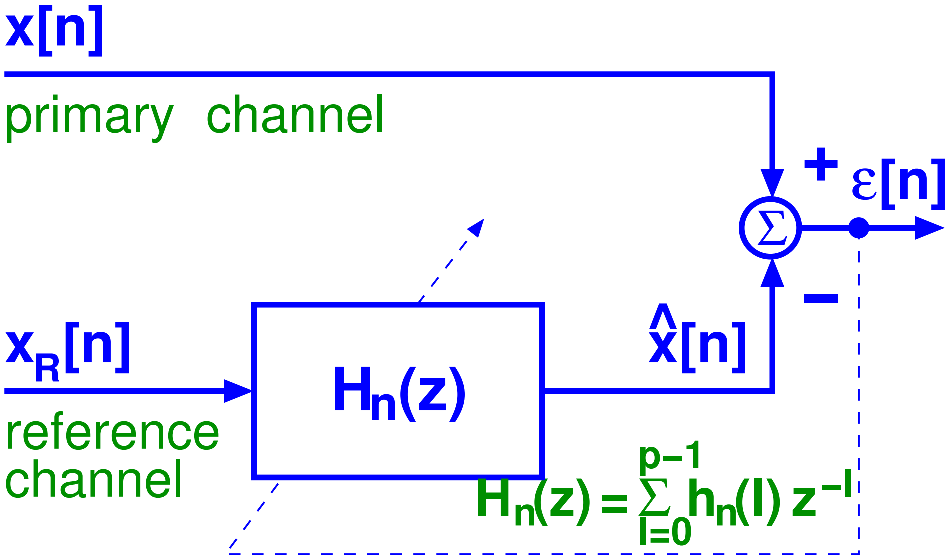


Case study: Adaptive Noise Canceller (ANC)

a common application of signal processing in order to reduce unwanted noise

Example 1: We may wish to remove background noise in aircraft and car audio systems (noise cancelling headphones, road noise cancellation)

Example 2: Another common problem is the removal of 50Hz mains artefact in biomedical instrumentation



The configuration of a sequential noise canceller

The reference channel takes the role of the traditional input and the primary channel is the noisy signal of interest, $x[n] = s[n; \boldsymbol{\theta}] + q[n]$

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 (nonzero correlation)
- Filter coefficients are updated sequentially to make $\hat{x}[n]$ as close to $x[n]$ as possible, in the LS sense
- 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 of the sequential LS problem

ANC $\looparrowright$ some practical considerations

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

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

☞ 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. Notice that here $\mathrm{h}[n]$ from Slides 26–27 (data in measurement model) is replaced by $\mathrm{x}_R[n]$, to avoid confusion with impulse response, h_n

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:

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

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:

$$\begin{aligned} e[n] &= x[n] - \sum_{l=0}^{p-1} \hat{h}_{n-1}(l) x_R[n-l] \\ \mathbf{K}[n] &= \frac{\mathbf{C}[n-1]\mathbf{x}_R[n]}{\lambda^n + \mathbf{x}_R^T[n]\mathbf{C}[n-1]\mathbf{x}_R[n]} \\ \mathbf{C}[n] &= \big(\mathbf{I} - \mathbf{K}[n]\mathbf{x}^T[n]\big)\mathbf{C}[n-1], \quad \text{typically} \quad 0.9 < \lambda < 1 \end{aligned}$$

In LS methods we do not know the probability densities or σ_n^2 for every sample $x[n]$.

☞ we replace them with a forgetting factor λ^n. This favours most recent samples 👍

Example 8: ANC for line noise removal (0.1Hz sinus. interfer.)

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 $q[n] = 10\cos(2\pi(0.1)n + \pi/4$.

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

○ 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 $\lambda = 0.99$

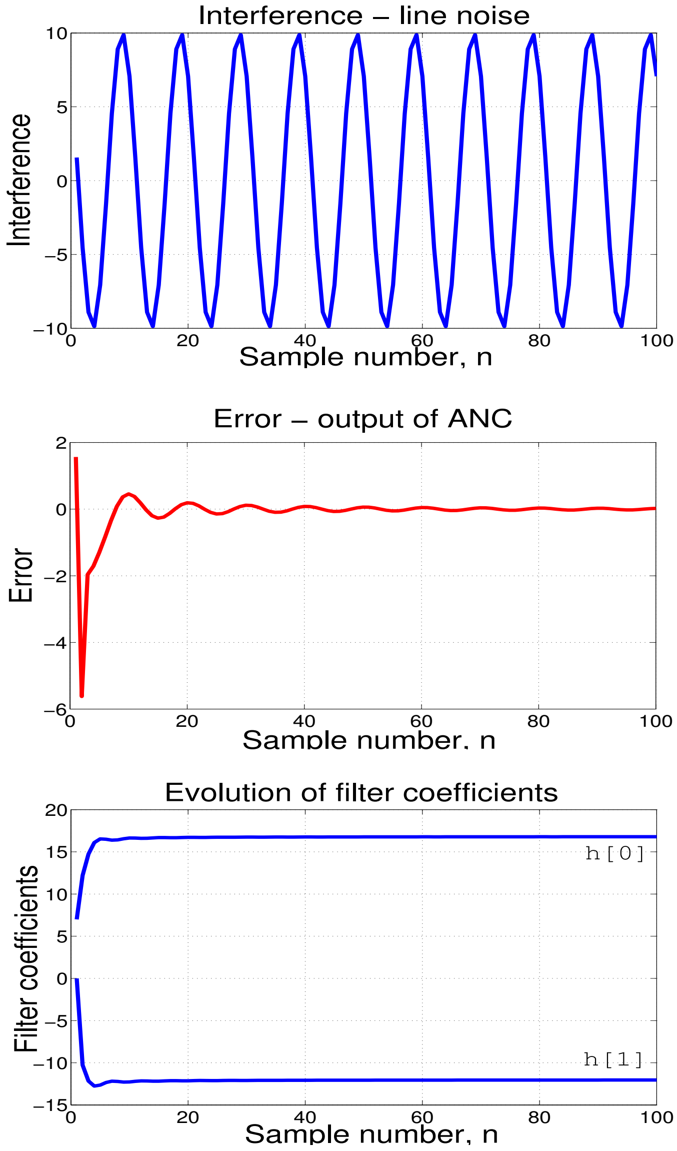
○ We need two filter coefficients to model the amplitude and phase of the interference, that is
$$\mathcal{H}[exp(2\pi(0.1))] = 10exp(\jmath\pi/4)$$

$\rightsquigarrow$ the noise canceller must increase the gain of the reference by 10 and phase by $\pi/4$ to match the interference.

Upon solving, (ANC performance on the right)

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

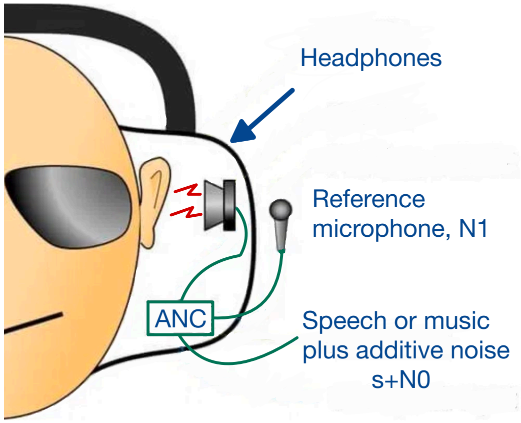
which results in $h[0] = 16.8$ and $h[1] = -12$.

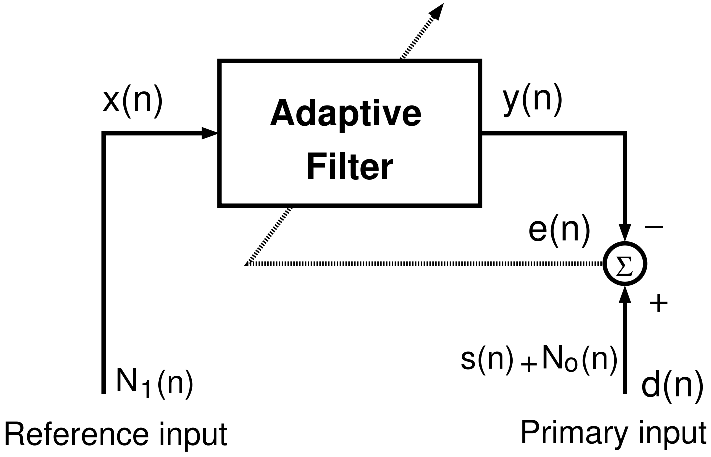


Applications: Adaptive noise cancellation with reference

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

In the adaptive noise cancellation configuration (below right), 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, $y = \hat{N}_0$.

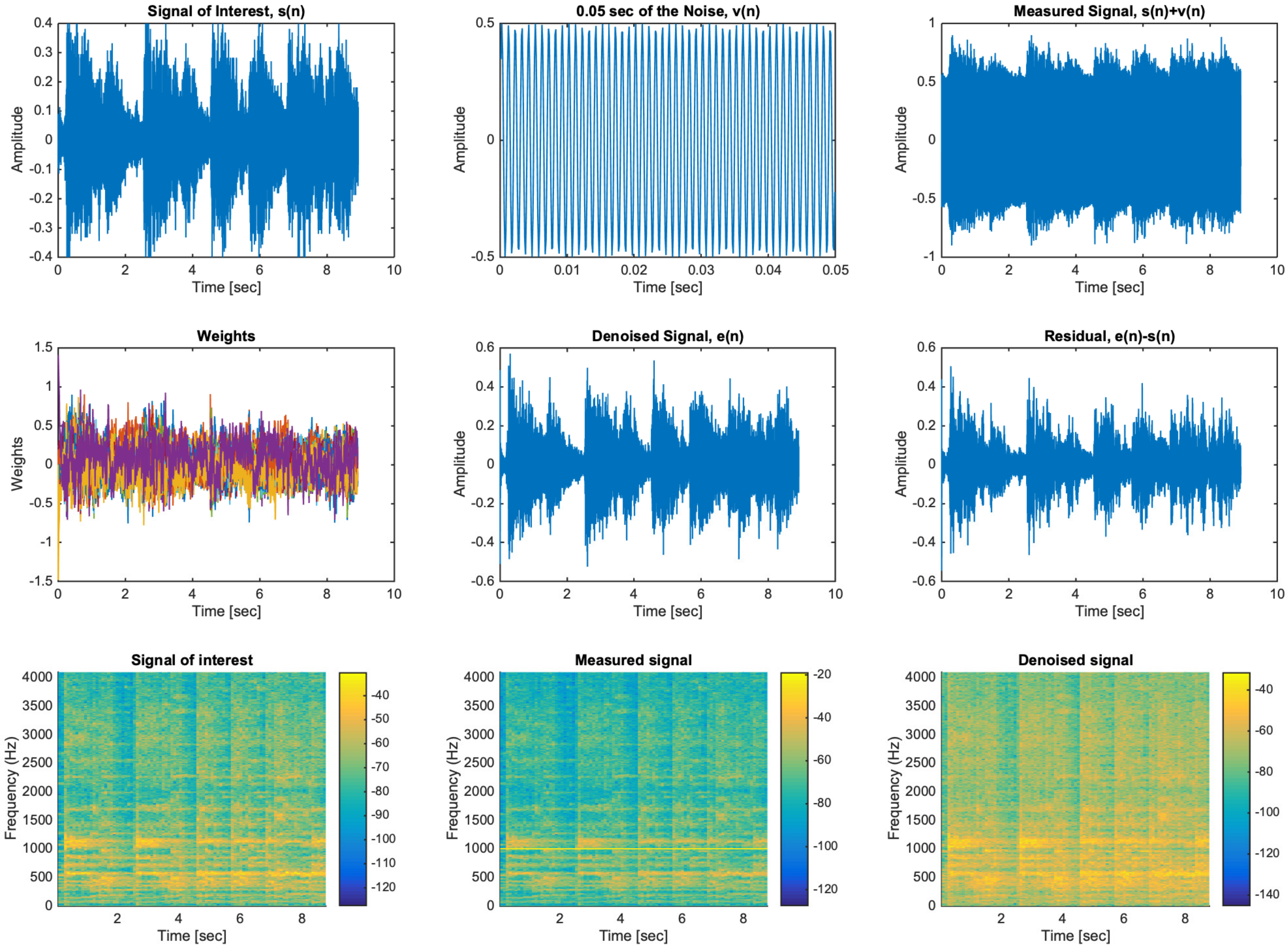
Teaching signal, $d(n)$, is the noise-polluted signal of interest, $s(n) + N_0(n)$, which serves as the Primary Input to the filter. Since $s \perp N_1$, the filter can only yield $y = \hat{N}_0$.

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, $e = s + N_0 - \hat{N}_0$, serves as a **"system output"**, whereby the adaptive filter aims to achieve $e \approx s$. In other words, the standard e serves as an output, $e = \hat{s}$.

Example 9: Noise cancelling headphones and acoustic feedback cancellation $(\lambda = 0.99)$

Denoising_SLS_GUI.m

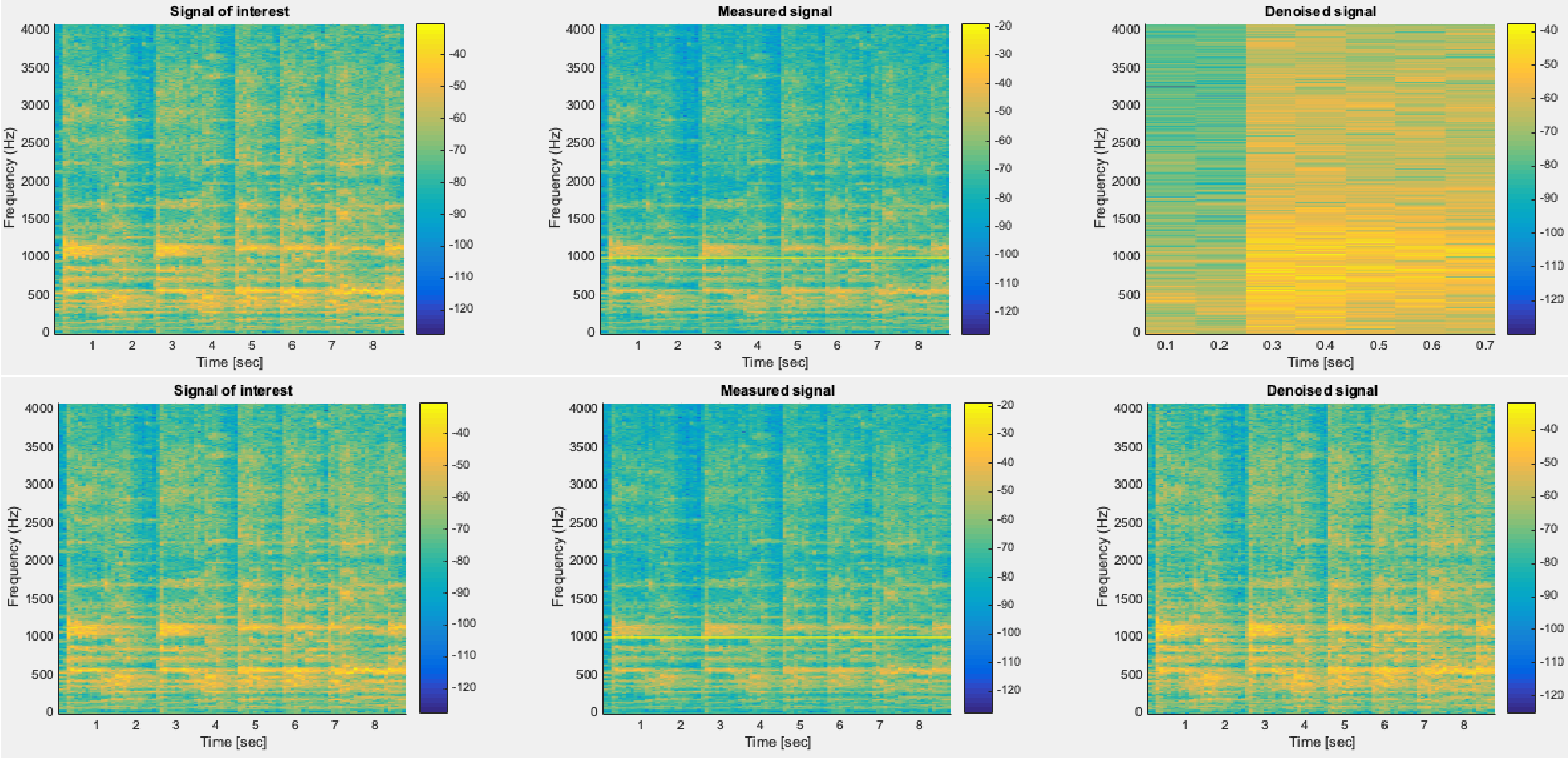


More in the Adaptive Signal Processing and Machine Intelligence course

Example 9: Adaptive noise cancellation: Role of the forgetting factor

Denoising_SLS_GUI.m

Top panels: Forgetting factor $\lambda = 0.9$

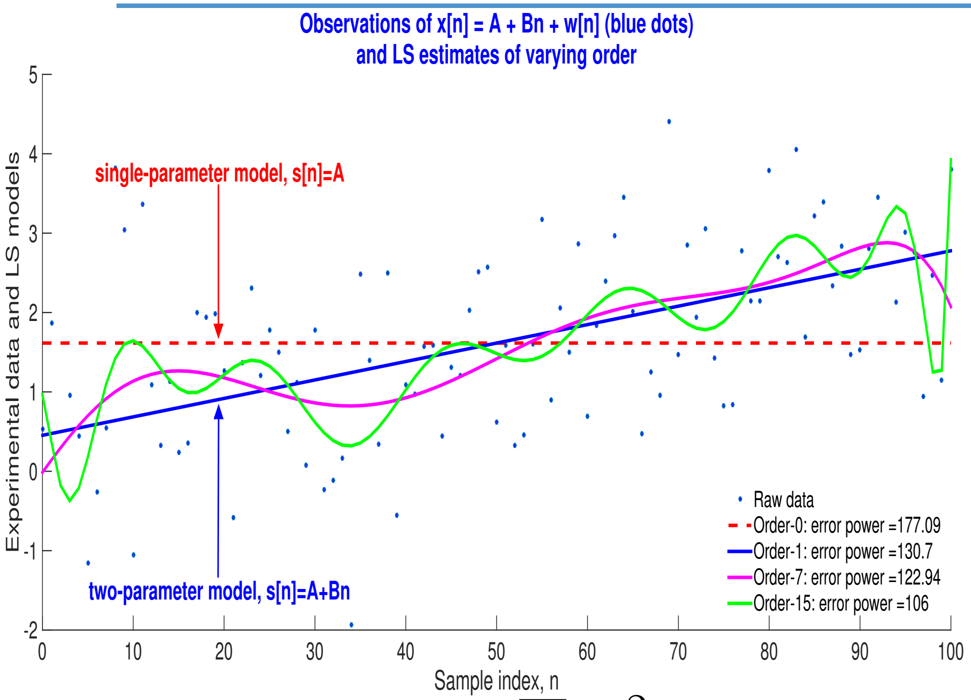


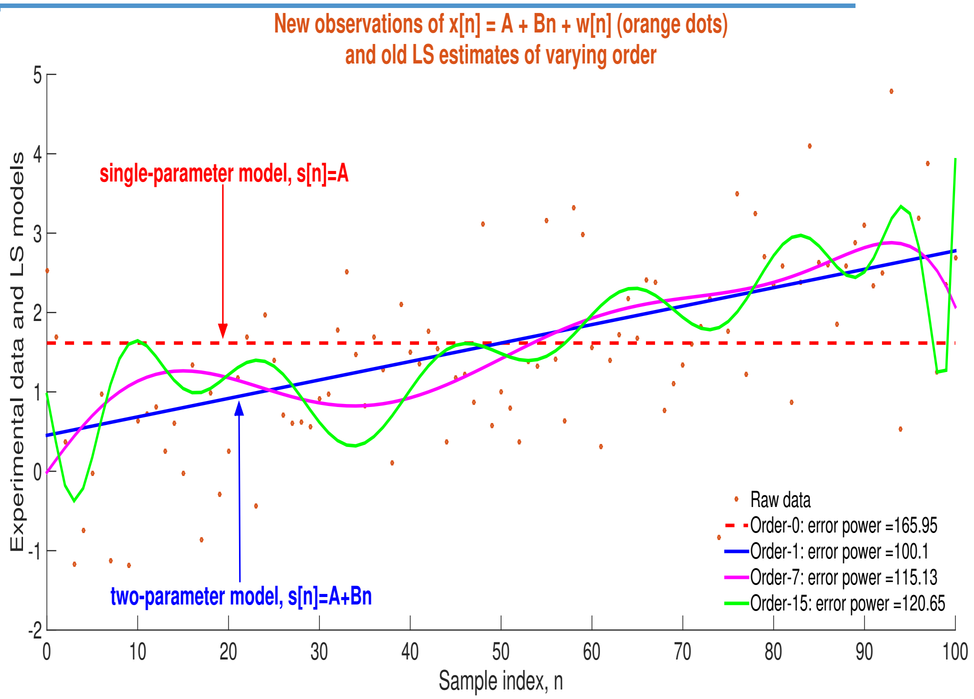
Bottom panels: Forgetting factor $\lambda = 0.995$

Lecture summary

- The method of least squares is extremely important for practical applications. **Least Squares does not mean fitting 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.
- Estimation error is orthogonal to the signal model space.
- Method of LS is easy to implement and straightforward to interpret.
- Sequential solutions to the LS problem are very practical.
- Weighted least squares allow to assign "confidence" to samples, that is to de–emphasise unrealiable samples.
- We can also use a forgetting factor to deal with time-varying statistics
- A number of applications of LS theory: Adaptive noise cancellation, digital filter design, Prony type spectral estimation, and many more ...

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





☞ 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 fit N data points. **Warning: Do not fit 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 $J_{min}(p) - J_{min}(p-1) > \varepsilon$ (user threshold)

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

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

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

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

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

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

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

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

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

Appendix: 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 ↬ 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

Example: Consider N coarse-quantised data points, $x[0], \ldots, x[N-1]$. The signal has $M \ll N$ possible amplitude values, $V_1, \ldots, V_M$, with the corresponding relative frequencies, $N_1, \ldots, N_M$. Calculate the mean, $\bar{x}$.

Solution:

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

Appendix: 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**

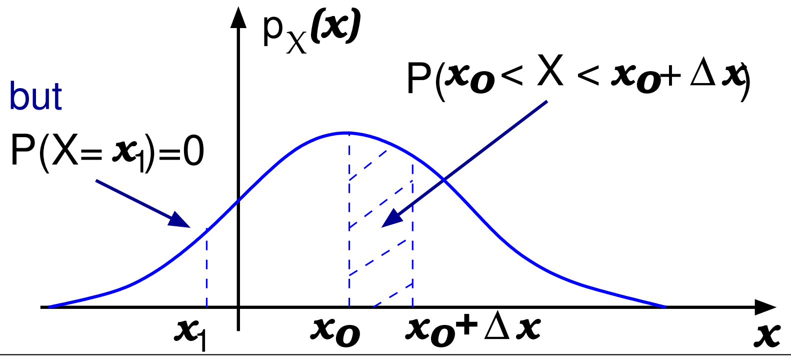
for example: $$average = E\{X\} = \int_{-\infty}^{\infty} x\, p_X(x)\, dx$$ no data here

Notice that there is no explicit mention of data here $\looparrowright$ 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

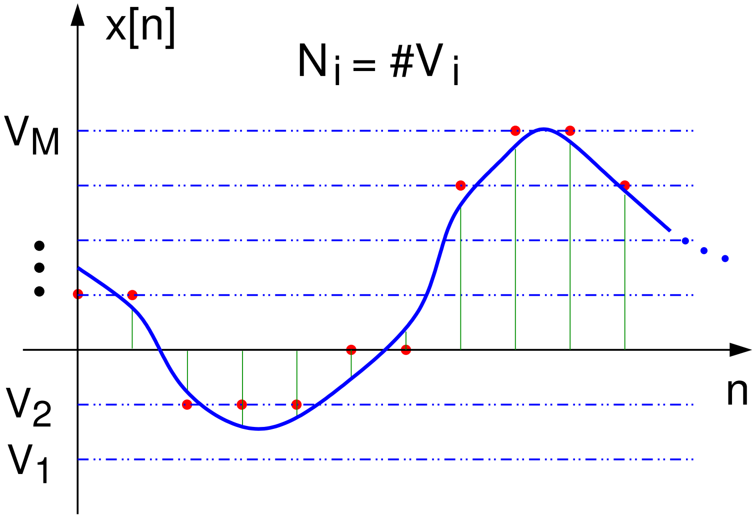
$$P(X = x_1) = 0$$

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

Appendix: Statistics vs. Probability

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, $\bar{x}$.

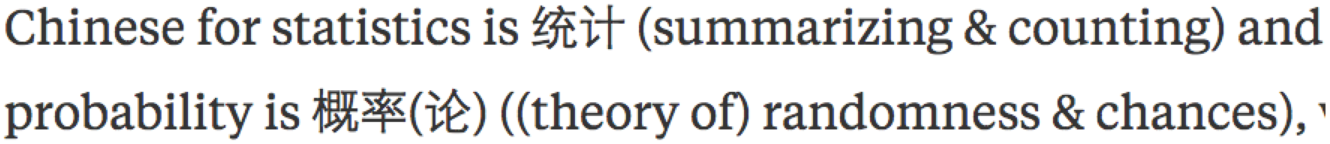


Solution:

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

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

Appendix: 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 to randomly draw 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: Range of a matrix, span of a set of vectors

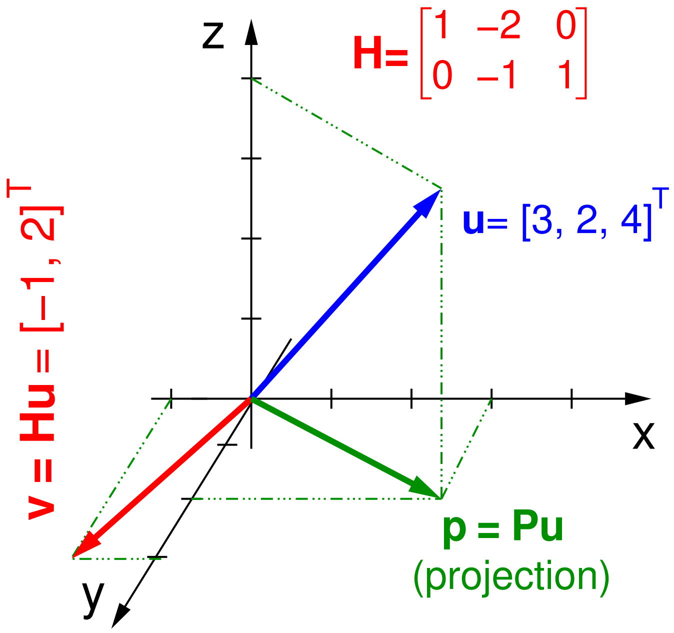
(a wide matrix transforms a vector space into another lower-dimensional one)

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

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

Then,

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


z
$\mathbf{H} = \begin{bmatrix} 1 & -2 & 0 \\ 0 & -1 & 1 \end{bmatrix}$
$\mathbf{u} = [3, 2, 4]^T$
$\mathbf{v} = \mathbf{Hu} = [-1, 2]^T$
x
y
$\mathbf{p} = \mathbf{Pu}$
(projection)

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 $\mathbf{p} = \mathbf{Pu}$, where $\mathbf{P} = \mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$

Quadratic forms and positive–(semi)definite matrices

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

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

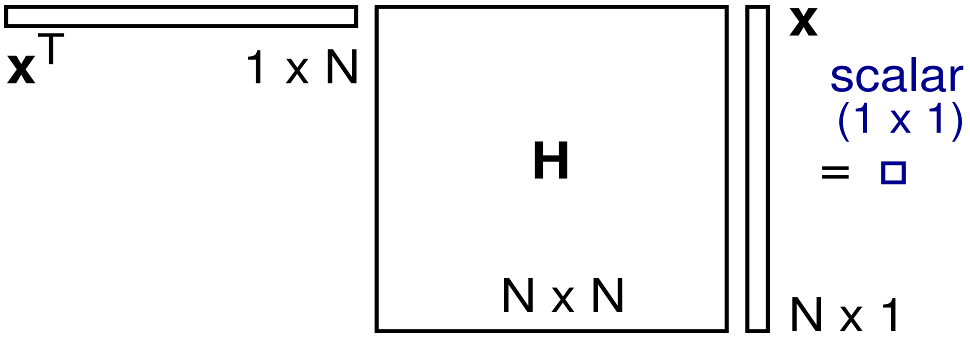
For simplicity, consider a 2nd order case, where

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

$$\uparrow variable \ \ vector \qquad \uparrow fixed \ \ matrix$$

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

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



○ If $Q_{\mathbf{H}}(\mathbf{x}) \geq 0$, for any $\mathbf{x} \neq \mathbf{0}$ then the matrix $\mathbf{H}$ is called positive semidefinite

○ The matrix $\mathbf{H}$ is positive definite if $Q_{\mathbf{H}}(\mathbf{x}) > 0, \forall \mathbf{x} \neq \mathbf{0}$

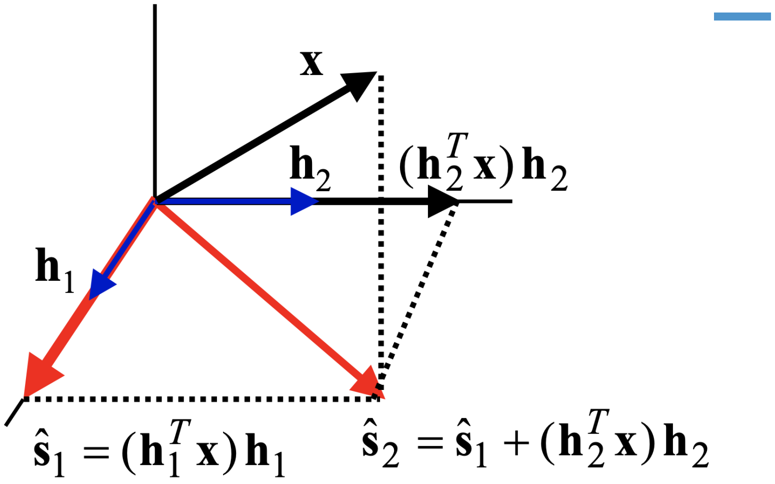
Appendix: Order Recursive Least Squares (ORLS)

(If 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



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

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

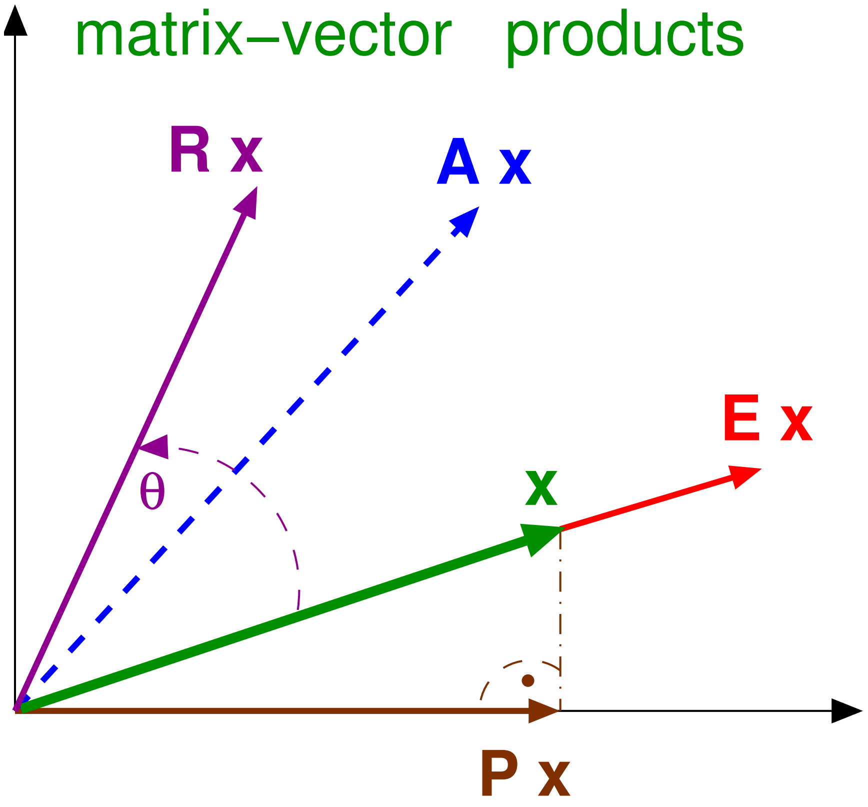
☞ **We can then use p-order model to compute the $(p+1)$-order model!**

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: What is that a matrix does to a vector?



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

A ⇝ any general matrix

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

$\mathbf{E}\mathbf{x} = \lambda \mathbf{x}$ ⇝ eigenanalysis

P ⇝ 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?

Notes:

- ○

Notes:

○