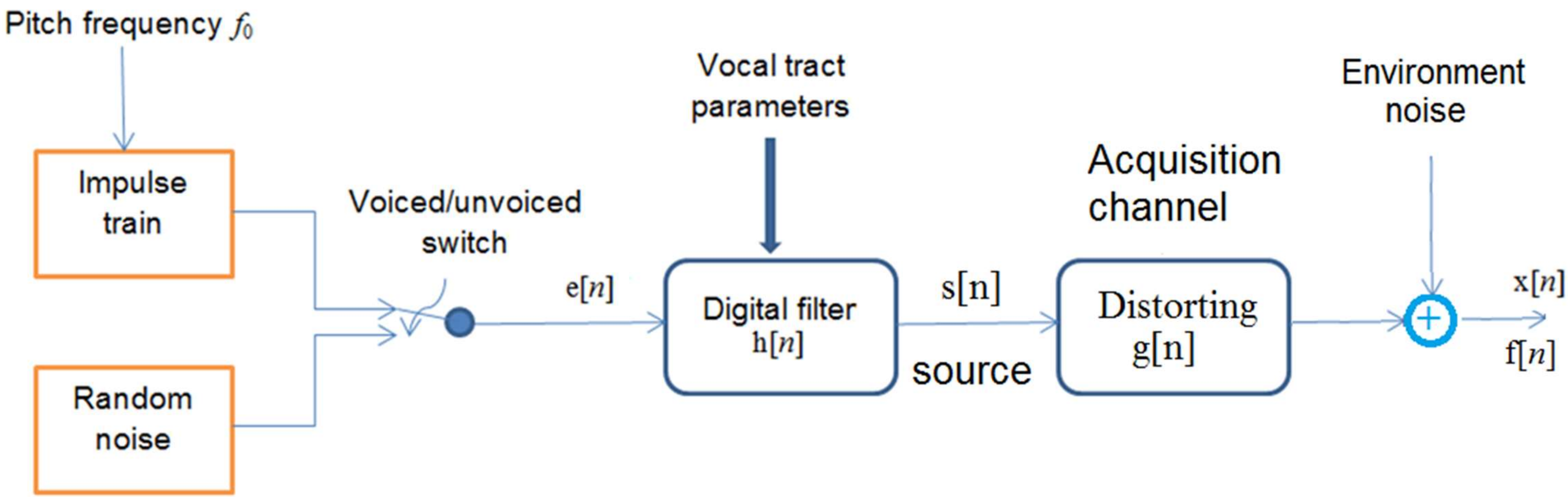
# SPRO 8
# Custom filter

# 1. Speech synthesis and (acquisition) distortion model




- The sound excitations and the frequency response h[$n$] – which is varying in time – is modelling the speech articulation tract.
- The distorting filter g[n] and noise are modeling the distortions appearing in the signal acquisition process.

# Source signal reconstruction

A signal acquisition of distorted source signal is modeled as:

$$f = s \otimes g + n$$

where: $f$ – measured signal, $s$ – the ideal, non-distorted source,

$g$ – an unknown distortion convoluted with the source,

$n$ – an unknown additive noise.

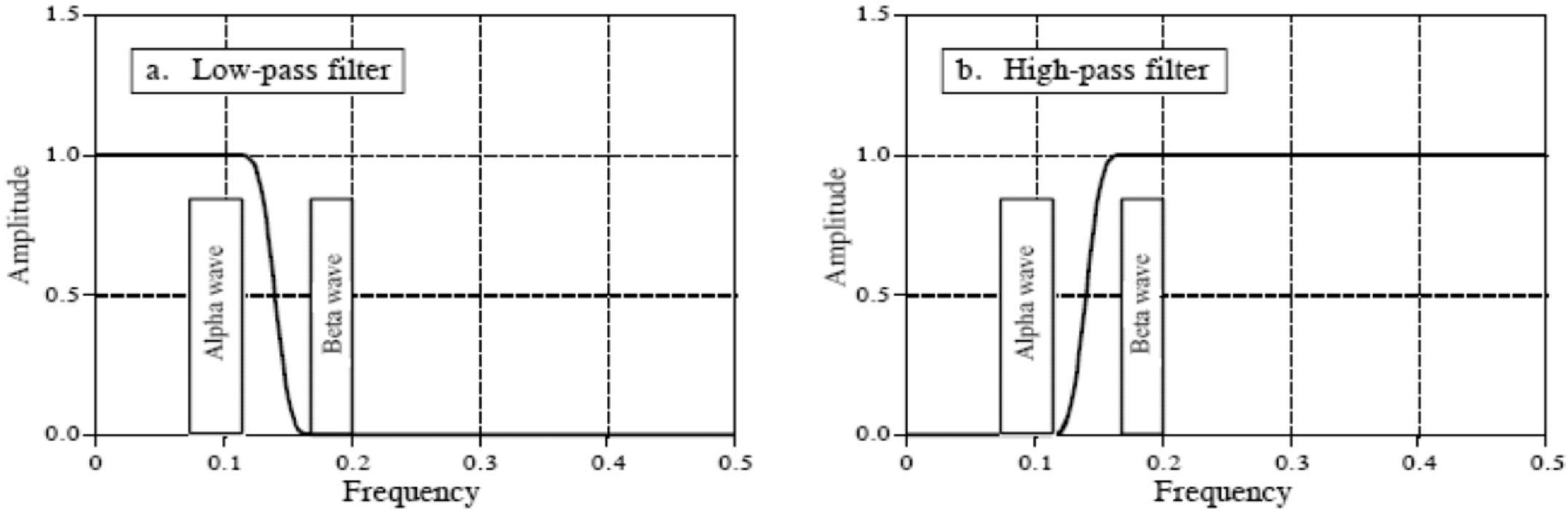
Firstly, we concentrate on additive **noise reduction (cancellation)**, when $f = s + n$

- High-frequency noise can be cancelled by applying a low-pass filter with appropriate cut-off frequency.
- DC component can be cancelled by a notch (band-reject) filter.
- Structured noise can be cancelled by the method of spectral subtraction.
- Random (white) noise can be reduced by a smoothing filter or Wiener filter

# 2. Signal denoising

Signal spectra are separable

If the frequency spectra of the useful source and distortions are clearly disjoint a band-pass frequency („notch") filter can be applied to separate the signals:



# Color noise - spectral subtraction

**Color noise estimation and spectral subtraction**

if the noise characteristics is unknown, but it can be measured.

Iteratively estimate the Fourier coefficients of the noise signal in frames which obviously include only the noise signal:

$$|\hat{N}_{m,j}|^2 = (1-\gamma)\,|\hat{N}_{m-1,j}|^2 + \gamma\,|F_{m,j}|^2$$

where the parameter $\gamma \approx 0.2$ for frames without speech and $\gamma = 0.0$ for frames with speech.

Subtract the estimated noise energy from the total signal energy in frames, where we detect speech:

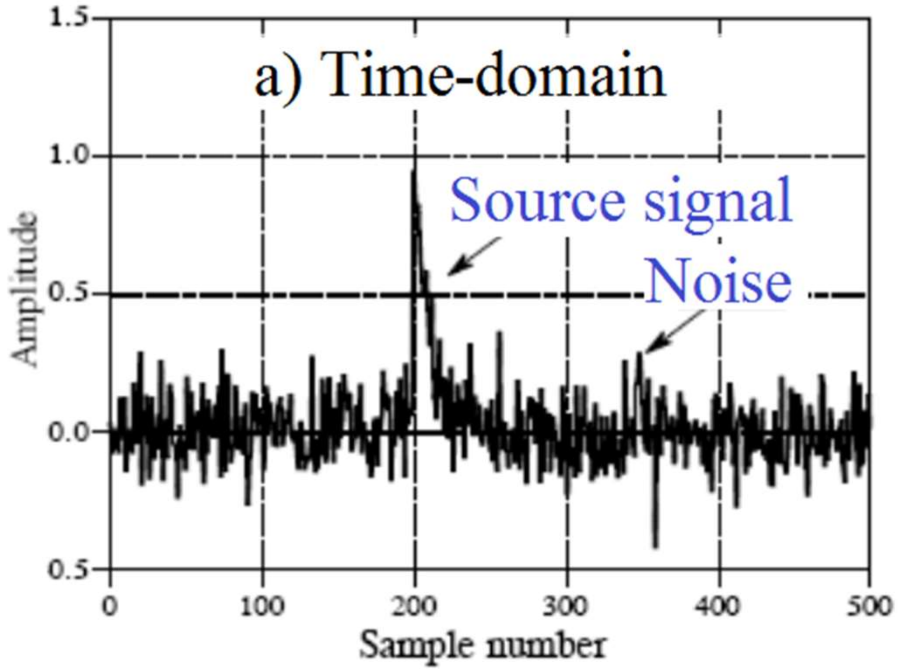
$$|\hat{S}_{m,j}|^2 = \begin{cases} |F_{m,j}|^2 - \alpha\,|\hat{N}_{m,j}|^2 & : if\ |F_{m,j}|^2 - \alpha\,|\hat{N}_{m,j}|^2 > \beta\,|F_{m,j}|^2 \\ \beta\,|F_{m,j}|^2 & : otherwise \end{cases}$$

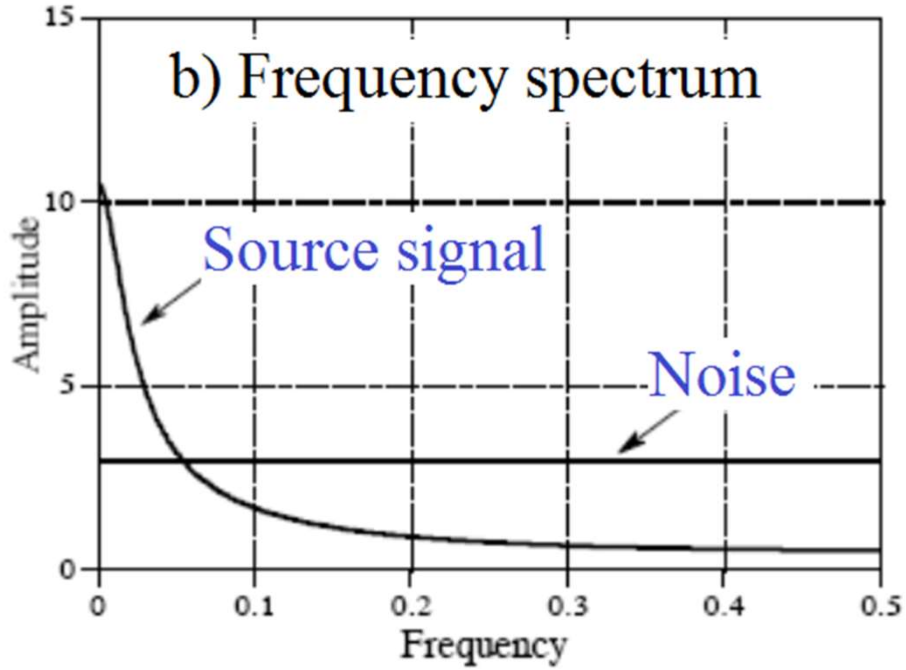
where the parameters $\alpha \approx 0.9$, $\beta \approx 0.15$.

# White noise

<u>White noise problem</u>

Trying to extract a waveform (e.g., an exponential pulse) buried in random noise is not easy even in the frequency domain, as the spectrum of the noise is **white** (the same amplitude at all frequencies) and the spectra of the signal and noise overlap.



# White noise filtering

<u>The moving average filter</u>

• The filter kernel is a rectangular pulse with an amplitude equal to one divided by the number of points in the average;

• This filter is optimal in the sense that it provides the fastest step response for a given noise reduction.

<u>The matched filter</u>

• The filter kernel is the same as the target signal being detected, except it has been flipped left-for-right. This flip is required to perform correlation using convolution,

• A point in the output signal is a measure of how well the filter kernel matches the corresponding section of the input signal,

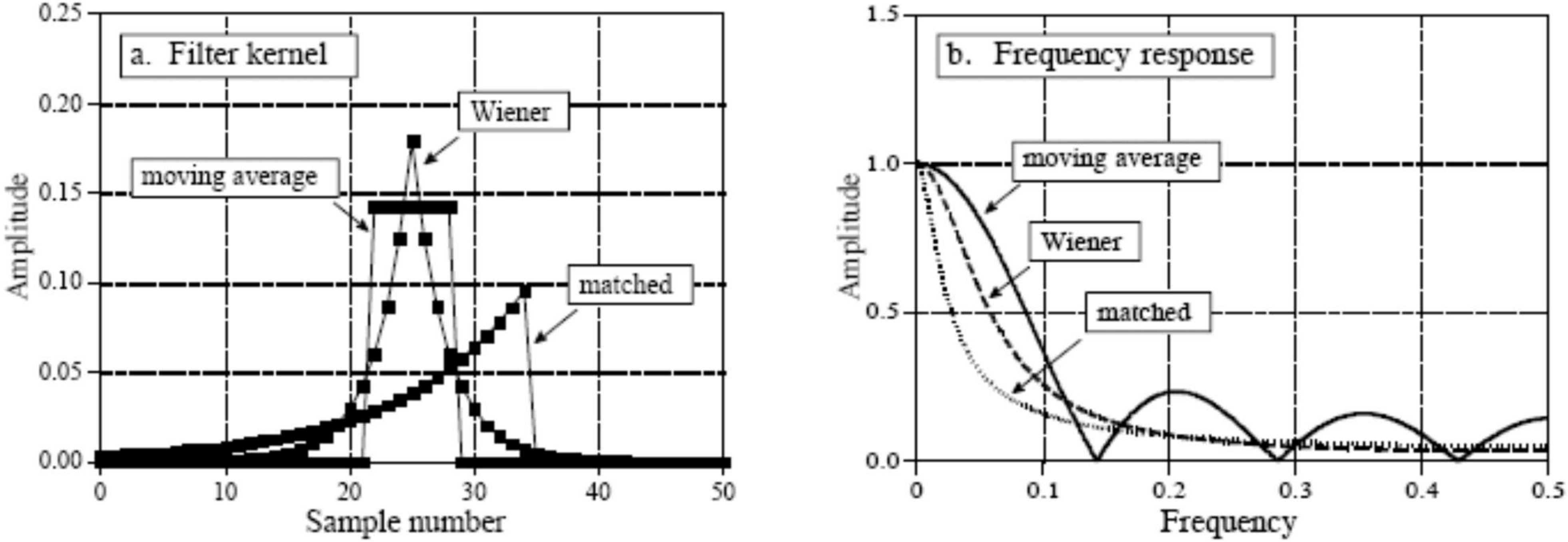
• The shape of the target signal must already be known.

• This filter is optimal in the sense that the top of the peak is farther above the noise than can be achieved with any other linear filter.

# Wiener filter

The Wiener filter is optimal in the sense that it estimates the ratio of the signal power to the noise power (over the length of the signal). It separates signals from noise, as the gain of the filter at each frequency represents the relative amount of signal and noise at that frequency:

$$H[f] = \frac{S[f]^2}{S[f]^2 + N[f]^2}$$



# 3. Wiener filter

The goal of the **Wiener** filter is to compute an estimate of an unknown signal using a related signal as an input and by filtering that observed signal to estimate the source.

- **Random noise** cancellation by a Wiener filter

Let the unknown source signal be corrupted by additive noise. The Wiener filter can be used to filter out the noise from the corrupted signal to estimate the source signal.

- The design of an approriate Wiener filter is based on the minimum mean square error (MMSE) estimator.
- Contrary to pre-determined filter kernels, the parameters of the Wiener filter depend on the measured data and some assumption about the unknown source properties:
  - the spectra of the source signal and noise are known, or
  - auto- and cross-correlation coefficents are known.

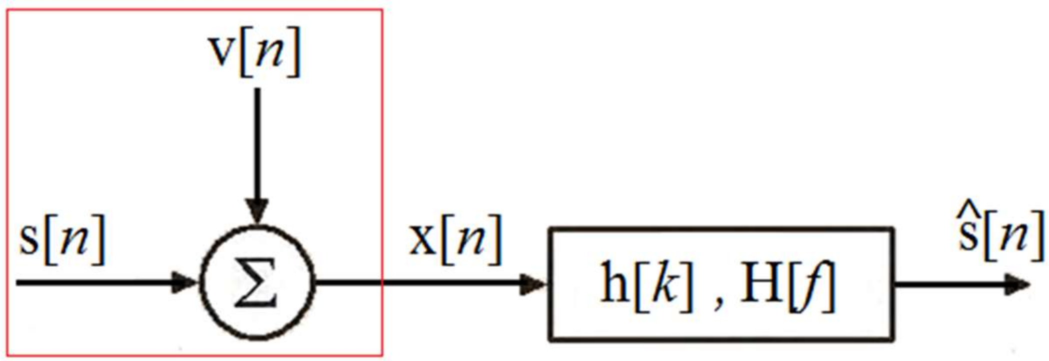
# The Wiener filter model

The SISO model (single input-single output) of the Wiener filter:

given an observation, $x[n] = s[n] + v[n]$, where

- $s[n]$ is some original signal (unknown) at time $n$,
- $v[n]$ is some unknown additive noise, independent of $s[n]$,
- $x[n]$ is the observed signal,

the goal of a Wiener filter design is to define an impulse response $h[k]$ (or frequency response $H[f]$) in order to estimate the unknown source:



# The Wiener filter

The goal is to find some $\mathrm{h}[k]$ so that one can estimate $\hat{s}[n]$ as follows: $\hat{s}[n] = h[k] \otimes x[n]$, that minimizes the mean square error

$$\mathrm{J} = E\{|s[n] - \hat{s}[n]|^2\}$$

Equivalent criterion can be formulated in the frequency domain:

$$\varepsilon(f) = E\{|S[f)] - \hat{S}[f]|^2\}$$

It is possible to estimate such impulse response $\mathrm{h}[k]$, given $x[n]$, under the assumption, that $x[n]$ and the random noise signal $v[k]$ are zero-mean and stationary and additionally:

- either the auto- and cross-correlation functions $\mathrm{R_{ss}}$, $\mathrm{R_{vv}}$ (in the time domain) or
- the corresponding spectral functions, $\mathrm{S}[f]$, $\mathrm{V}[f]$, are known.

# A. Wiener filter in frequency domain

The **Wiener filter** is optimal in the sense that it minimises the **mean-square error** (over the signal length).

**In the frequency domain** it separates signals from noise, as the gain of the filter *at each frequency* represents the relative amount of signal and noise *at that frequency*:

$$H(f) = \frac{S(f)^2}{S(f)^2 + V(f)^2}$$

Remarks:

1. A general formula – continuous frequency domain
2. Only the magnitudes are important; all the phases are zero.

# Derivation of Wiener filter in the frequency domain

$$\varepsilon(f) = E\{|S(f) - \hat{S}(f)|^2\}$$
$$= E\{|S(f) - H(f)X(f)|^2\}$$
$$= E\{|S(f) - H(f)[S(f) + V(f)]|^2\}$$
$$= E\{|[1 - H(f)]S(f) - H(f)V(f)|^2\}$$
$$= [1 - H(f)][1 - H(f)]^* E\{S(f)^2\} + H(f)H(f)^* E\{V(f)^2\}$$

$$\frac{d\varepsilon(f)}{dH(f)} = H(f)^* V_m(f)^2 - [1 - H(f)]^* S_m(f)^2 = 0$$

$$H(f) = \frac{S_m(f)^2}{S_m(f)^2 + V_m(f)^2}$$

# B. Wiener-Hopf filter (time-domain)

Define the error signal $e[n]$, such that the error sample at time $i$ is:

$$e_i = s_i - \hat{s}_i = s_i - \hat{\mathbf{h}}_k^T \cdot \boldsymbol{x}_{i,k}$$

where

$$\hat{\mathbf{h}}_k^T = [\hat{h}_0, \hat{h}_1, \cdots, \hat{h}_{k-1}]$$

is an unknown filter kernel of length $k$ ($k \leq \mathrm{L}$, and $\mathrm{L}$ length of observation ), while

$$\boldsymbol{x}_{i,k}^T = [x_i, x_{i-1}, \cdots, x_{i-k-1}]$$

is a vector of $k$ observed samples.

To find the optimal Wiener filter, we need to **minimize a cost function** related to above error signal $e[n]$. The usual choice for the goal function is the **mean-square error (MSE)** :

$$J(\hat{\mathbf{h}}_k) = \mathrm{E}\{e[n]^2\}$$

# Wiener-Hopf filter (cont.)

The optimal Wiener filter cancels the gradient of the goal function:

$$\frac{\partial J(\hat{\mathbf{h}}_k)}{\partial \hat{\mathbf{h}}_k} = \mathbf{0}_{k\times 1}$$

$$\frac{\partial J(\hat{\mathbf{h}}_k)}{\partial \hat{\mathbf{h}}_k} = 2\mathrm{E}\left\{e[n]\frac{\partial e[n]}{\partial \hat{\mathbf{h}}_k}\right\} = -2\mathrm{E}\{e[n]\cdot \boldsymbol{x}_k[n]\}$$

Therefore, at the optimum we have

$$\mathrm{E}\{e^{(o)}[n]\cdot \boldsymbol{x}_k[n]\} = \mathbf{0}_{k\times 1} \qquad (2)$$

where

$$e^{(o)}[n] = s[n] - \hat{\mathbf{h}}_k^T \cdot \boldsymbol{x}_k[n] \qquad (3)$$

# Wiener-Hopf (cont.)

The optimal estimate of $s[n]$ is: (4)

$$\hat{s}[n] = \hat{\mathbf{h}}_k^T \cdot \boldsymbol{x}_k[n]$$

It appears that also:

$$\hat{\mathbf{h}}_k^T \cdot \mathrm{E}\{e^{(o)}[n] \cdot \boldsymbol{x}_k[n]\} = \hat{\mathbf{h}}_k^T \cdot \mathbf{0}_{k\times 1}$$

$$\mathrm{E}\{e^{(o)}[n] \cdot \hat{\mathbf{h}}_k^T \cdot \boldsymbol{x}_k[n]\} = 0$$

$$\mathrm{E}\{e^{(o)}[n] \cdot \hat{s}[n]\} = 0$$

# Wiener-Hopf equation

Substitute $e_o(k)$ from eq. (3) to $\mathrm{E}\{.\}$ in eq. (2):

$$\mathrm{E}\{(s[n] - \hat{\mathbf{h}}_k^T \cdot \boldsymbol{x}_k[n]) \cdot \boldsymbol{x}_k[n]\} = \mathbf{0}_{k\times 1}$$

$$\mathrm{E}\{(s[n] \cdot \boldsymbol{x}_k[n])\} - \mathrm{E}\{\hat{\mathbf{h}}_k^T \cdot \boldsymbol{x}_k[n] \cdot \boldsymbol{x}_k[n]\} = \mathbf{0}_{k\times 1}$$

$$\mathbf{r}_{sx}^{(k)} = \mathbf{R}_{xx}^{(k)} \quad \hat{\mathbf{h}}_{k,o} \qquad (6)$$

Equation (6) is called the **Wiener-Hopf equation**:

where $$\mathbf{R}_{xx}^{(k)} = \mathrm{E}\{\boldsymbol{x}_k[n] \cdot \boldsymbol{x}_k^T[n]\}$$

is the auto-correlation matrix of the signal $x[n]$,

and $$\mathbf{r}_{sx}^{(k)} = \mathrm{E}\{\mathbf{x}_k[n] \cdot s[n]\}$$

is the cross-correlation vector of length $k$ between $\mathbf{x}_k[n]$ and $s[n]$.

# Wiener-Hopf equation (cont.)

The auto-correlation matrix is:

$$\mathbf{R}_{xx} = \begin{pmatrix} r(0) & r(1) & \cdots & r(k-1) \\ r(1) & r(0) & \cdots & r(k-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(k-1) & r(k-2) & \cdots & r(0) \end{pmatrix}$$

with $r(l) = \mathrm{E}\{x[n] \cdot x[n-l]\}, \quad l = 0,1,\cdots,k-1$

Assuming $\mathbf{R}_{xx}$ is non-singular (it is in fact symmetric and semi-positive and a Toeplitz matrix also), the **Wiener filter** of order $k$ is:

$$\hat{\mathbf{h}}_{k,o} = {\mathbf{R}_{xx}}^{-1}\mathbf{r}_{sx}$$

The efficient Levinson-Durbin algorithm can be applied to solve this equation.

It can be shown that, $r_{sx}(l) = r_{ss}(l)$ (if $\mathrm{s}[n]$ and $\mathrm{v}[n]$ are un-correlated), while $r_{xx}(l) = r_{ss}(l) + r_{vv}(l)$.

Thus, knowing $r_{vv}$ , equation (6) can be defined and solved.

# 4. Deconvolution

Unwanted **convolution** is an inherent problem of real-life signals.

Examples:

- Echoes in long distance telephone calls;
- The finite bandwidth of analog sensors and electronic devices.

**Deconvolution** is the process of filtering a signal to compensate for an undesired convolution. The goal of deconvolution is to reconstruct the original signal *before* the convolution took place.

**Deconvolution** requires the characteristics of the convolution (i.e., the impulse or frequency response) to be known.

When the filter's kernel and frequency response are unknown this makes a ***blind deconvolution*** problem.

**Blind deconvolution** – the parameters of the convolution operation are *not* known.

# Deconvolution

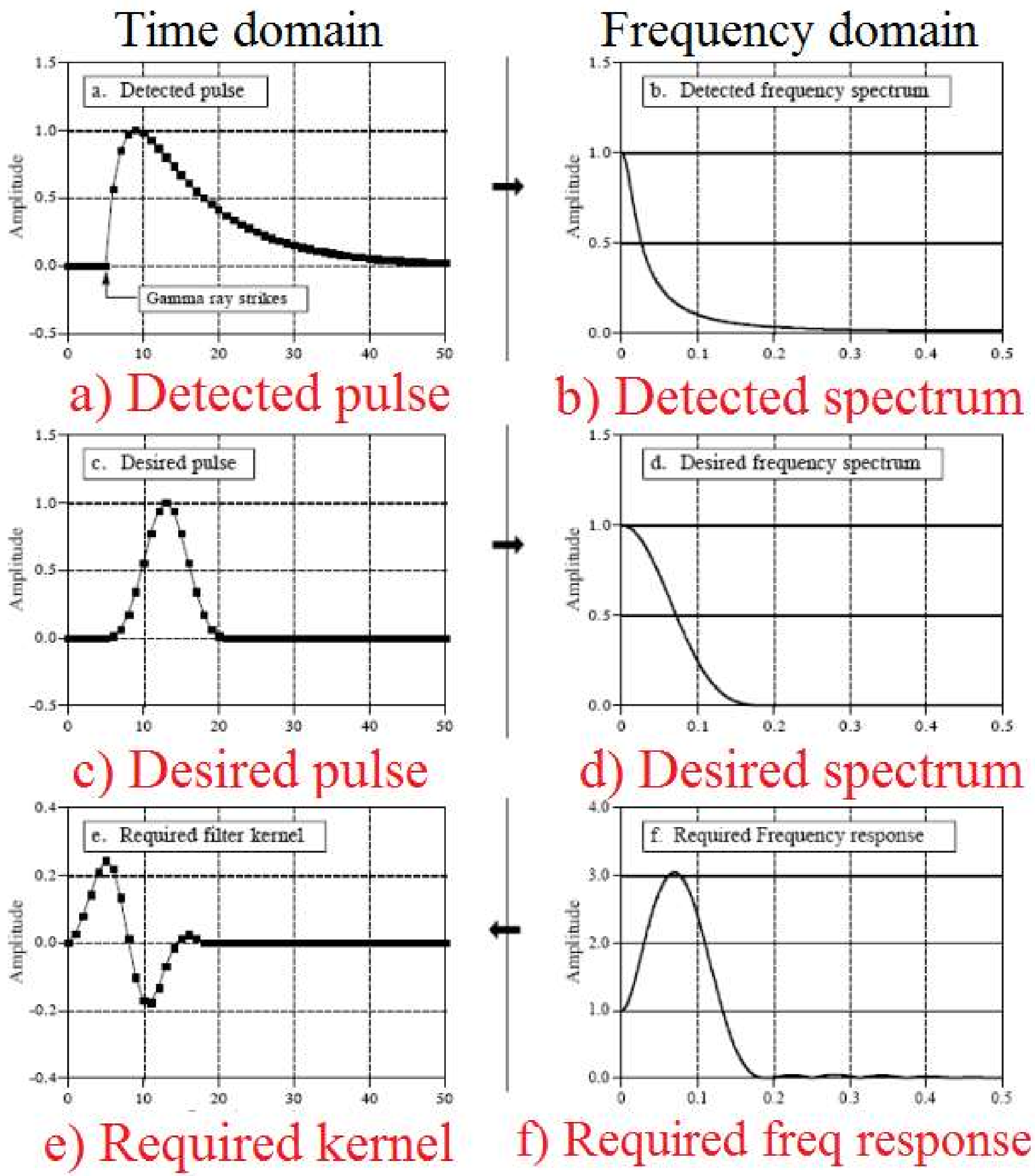
Assume, our goal is to find a filter kernel, (e), that when convolved with the signal in (a), produces the signal in (c), i.e.

**if** $a \otimes e = c$ , and **given** *a* and *c*, **find *e***.

Fortunately, this problem is simpler in the frequency domain:

**if** $b \cdot f = d$, and **given** *b* and *d*, **find *f***.

The unknown frequency response of the filter, (f), is the frequency spectrum of the desired pulse, (d), *divided* by the frequency spectrum of the detected pulse, (b).



# Blind deconvolution

A ***blind* deconvolution** problem: how to find the signal (a), given only measured signal (c): $a \otimes e = c$, or in the frequency domain - given only (d), how can we determine (b): $b \cdot f = d$

The solution: make an estimate or assumption about the unknown parameters.

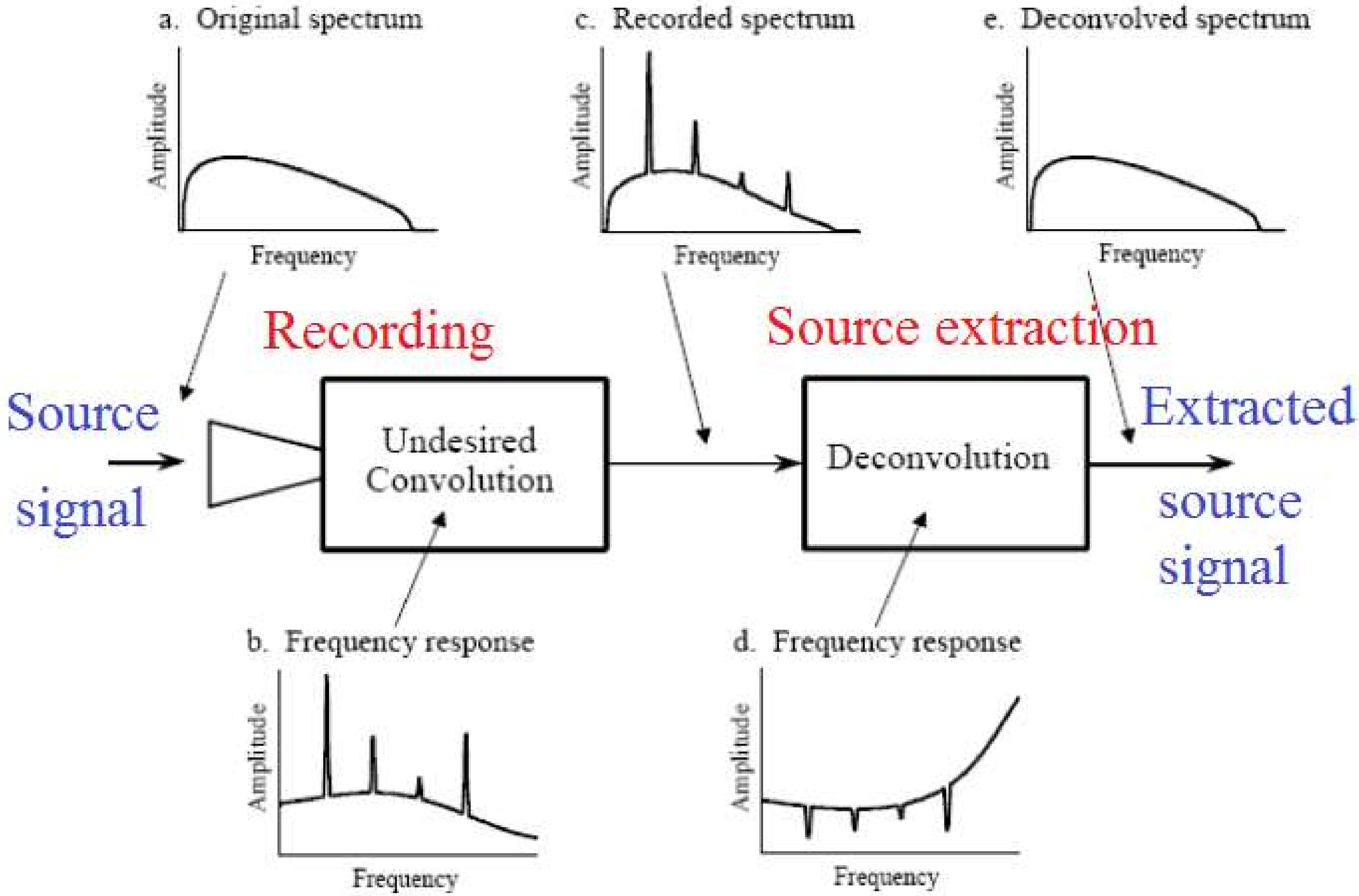
Example

The *average spectrum* of the original music is assumed to match the *average spectrum* of the same music performed by a present day singer using modern equipment. The *average spectrum* is found by:

- break the signal into a large number of segments,
- take the DFT of each segment,
- convert into polar form, and then average the magnitudes together.

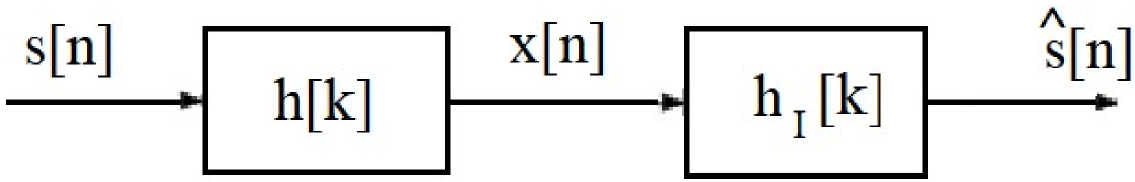
# Example of deconvolution

Deconvolution of old phonograph recordings. (a) The frequency spectrum produced by the original singer. (b) Resonance peaks in the primitive equipment produce distortions. (c) The recorded frequency spectrum. (d) The frequency response of the deconvolution filter is designed to counteracts the undesired convolution, restoring the original spectrum, (e).



# 5. System identification (for channel equalization)

A cascade connection of a system $h[n]$ and its inverse $h_I[n]$ is an identity system: $h[n] \otimes h_I[n] = \delta[n]$




In the frequency or Z-domain: $H_I(f) = \frac{1}{H(f)}$, $H_I(z) = \frac{1}{H(z)}$ .

Problem: system identification h[k].

Simple method: excite the system with known input signal and observe the output: $H(z) = \frac{X(z)}{S(z)}$ .

A practical method is based on signal correlation coefficients:

$$H(z) = \frac{P_{xs}(z)}{P_{ss}(z)}$$

where $P_{xs}(z)$ and $P_{ss}(z)$ are power spectra densities obtained from signal correlation values.

# System identification

Cross-correlation of two signals:

s$[n]$ with x$[n]$: $r_{sx}(l) = \sum_{n=-\infty}^{\infty} s_n \cdot x_{n-l}, \quad l = 0, \pm 1, \pm 2, \ldots$

or x$[n]$ with x$[n]$: $r_{xs}(l) = \sum_{n=-\infty}^{\infty} x_n \cdot s_{n-l}, \quad l = 0, \pm 1, \pm 2, \ldots$

Thus: $r_{sx}(l) = r_{xs}(-l)$

Correlation vs. convolution: $r_{sx}(l) = s[n] \otimes x[-(n-l)]$

As, $x[n] = s[n] \otimes h[k]$, it becomes: $r_{sx}(l) = h[-k] \otimes r_{ss}(l)$

In the frequency- or Z-domain it becomes:

$$P_{sx}(z) = H^*(z) \cdot P_{ss}(z),$$

where $P_{ss}(z)$ is the power spectral density of s$[n]$ and * is the complex conjugate operator. If $r_{sx}(l)$ is replaced by $r_{xs}(-l)$ we get:

$$H(z) = \frac{P_{xs}(z)}{P_{ss}(z)}$$

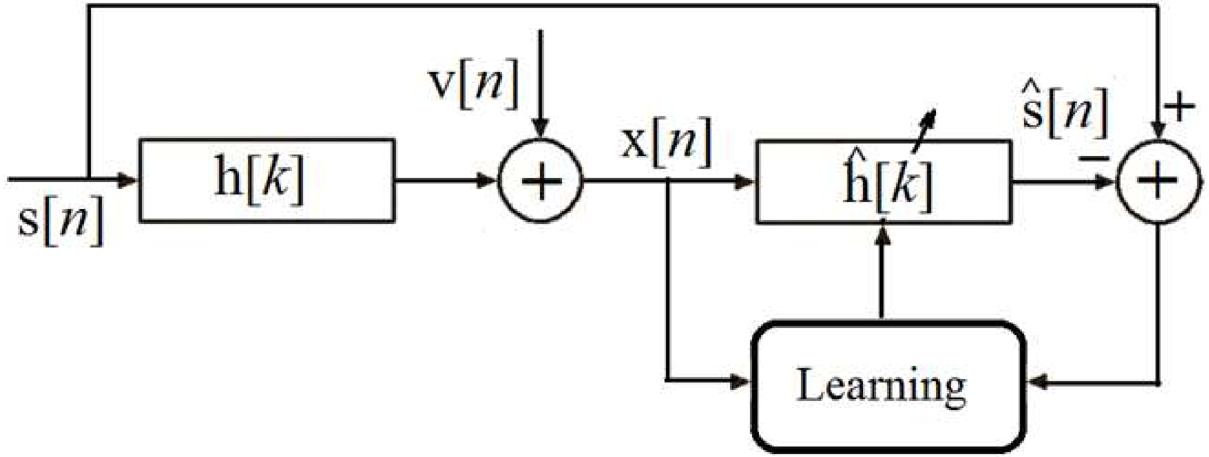
(e.g., this method is applied in the MATLAB function **tfe**).

# 6. Adaptive channel equalization

There are many algorithms for adaptive channel equalization.

For example, the problem can be solved by a Wiener filter (the **Wiener deconvolution**), which minimizes the mean-square error between the estimate of source and the desired source:

$$J = E\{|s[n] - \hat{s}[n]|^2\}$$




The solution in the frequency domain is:

$$\hat{H}(f) = \frac{H^{*}(f)\, S_m(f)^2}{H(f)^2\, S_m(f)^2 + V_m(f)^2}$$

# Wiener deconvolution

Derivation:

$$\varepsilon(f) = E\{|S(f) - \hat{S}(f)|^2\} = E\{\left|S(f) - \hat{H}(f)X(f)]\right|^2\} =$$

$$= E\{\left|S(f) - \hat{H}(f)[H(f)S(f) + V(f)]\right|^2\}$$

$$= E\{\left|[1 - \hat{H}(f)H(f)]S(f) - \hat{H}(f)\, V(f)\right|^2\}$$

$$= [1 - \hat{H}(f)H(f)][1 - \hat{H}(f)H(f)]^*\, E\{S(f)^2\}$$

$$+ \hat{H}(f)\hat{H}(f)^*\, E\{V(f)^2\}$$

$$\frac{d\varepsilon(f)}{d\hat{H}(f)} = \hat{H}(f)^* V_m(f)^2 - H(f)[1 - \hat{H}(f)H(f)]^*\, S_m(f)^2 = 0$$

$$\hat{H}(f) = \frac{H(f)^*\, S_m(f)^2}{H(f)H(f)^*\, S_m(f)^2 + V_m(f)^2}$$

# Wiener-Hopf deconvolution

Define the error signal: $e[n] = s[n] - \hat{s}[n] = s[n] - \hat{\mathbf{h}}_k^T \cdot \boldsymbol{x}_k[n]$

where $\hat{\mathbf{h}}_k^T = [\hat{h}_0, \hat{h}_1, \cdots, \hat{h}_{k-1}]$ is an unknown filter kernel of length $k \leq \mathrm{L}$, while $\boldsymbol{x}_k^T[n] = [x_n, x_{n-1}, \cdots, x_{n-k-1}]$ is a vector of $k$ observed samples.

The goal function: $J(\hat{\mathbf{h}}_k) = \mathrm{E}\{e[n]^2\}$

Solution: $\dfrac{\partial J(\hat{\mathbf{h}}_k)}{\partial \hat{\mathbf{h}}_k} = \mathbf{0}_{k\times 1}$

$$\frac{\partial J(\hat{\mathbf{h}}_k)}{\partial \hat{\mathbf{h}}_k} = 2\mathrm{E}\left\{e[n]\frac{\partial e[n]}{\partial \hat{\mathbf{h}}_k}\right\} = -2\mathrm{E}\{e[n] \cdot \boldsymbol{x}_k[n]\}$$

Therefore, at the optimum we have

$$\mathrm{E}\{e_o[n] \cdot \mathbf{x}_k[n]\} = \mathbf{0}_{k\times 1}$$

where

$$e_o[n] = s[n] - \hat{\mathbf{h}}_{k,o}^T \cdot \boldsymbol{x}_k[n]$$

# Wiener-Hopf deconvolution

Like for the Wiener denosing, we get the **Wiener-Hopf equation**:

$$\mathbf{r}_{sx}{}^{(k)} = \mathbf{R}_{xx}{}^{(k)} \cdot \hat{\mathbf{h}}_{k,o}$$

where

$$\mathbf{R}_{xx}{}^{(k)} = \mathrm{E}\{\mathbf{x}_k[n] \cdot \mathbf{x}_k^T[n]\}$$

is the auto-correlation matrix of the signal $x[n]$, and

$$\mathbf{r}_{sx}{}^{(k)} = \mathrm{E}\{\mathbf{x}_k[n] \cdot s[n]\}$$

is the cross-correlation vector of length $k$ between $\mathbf{x}_{\boldsymbol{k}}[n]$ and $s[n]$.

# Wiener-Hopf equation

The auto-correlation matrix is:

$$\mathbf{R}_{xx} = \begin{pmatrix} r(0) & r(1) & \cdots & r(k-1) \\ r(1) & r(0) & \cdots & r(k-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(k-1) & r(k-2) & \cdots & r(0) \end{pmatrix}$$

with $\quad r(l) = \mathrm{E}\{x[n] \cdot x[n-l]\}, \quad l = 0,1,\cdots,k-1$

Assuming that $\mathbf{R}_{xx}$ is non-singular (it is in fact symmetric and semi-positive and a Toeplitz matrix also), the **Wiener filter** of order $k$ is:

$$\hat{\mathbf{h}}_{k,o} = \mathbf{R}_{xx}{}^{-1}\mathbf{r}_{sx}$$

The auto- and cross-correlation coefficients must be estimated from the two known signals $\mathrm{s}[n]$ and $\mathrm{x}[n]$.

The efficient Levinson-Durbin algorithm can be applied to solve the above equation.

# Exercise 8.1

## Wiener filter

Suppose we have a signal s(n) with autocorrelation function $r_{ss}(k)=0.95^{|k|}$.

The signal is observed in the presence of additive white noise with variance $\sigma_{vv}^2 = 2$. Hence $r_{vv}(k)=2\delta(k)$.

s(n) and v(n) are uncorrelated, zero-mean, stationary random processes.

We would like to design the optimum second-order Wiener filter response $h$.

# Exercise 8.2

## Wiener deconvolution (channel equalization)

Assume an observed signal is given as: $x[k] = h[n] \otimes s[n] + v[n]$.

We assume that $x[n]$ and the random noise signal $v[n]$ are zero-mean and stationary, and s[$n$] and v[$n$] are not correlated.

Using a **Wiener filter**, it is possible to perform channel equalization, i.e., to estimate the impulse response $g[n]$ of the deconvolving filter, given $s[n]$ and $x[n]$, by solving the Wiener-Hopf equation:

$$\boldsymbol{g}_{k,o} = \boldsymbol{R}_k{}^{-1}\boldsymbol{p}_k$$

where: $$\boldsymbol{R}_k = \begin{pmatrix} r(0) & r(1) & \cdots & r(k-1) \\ r(1) & r(0) & \cdots & r(k-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(k-1) & r(k-2) & \cdots & r(0) \end{pmatrix}$$

# Exercise 8.2 (cont.)

With $r(l) = E\{x[n] \cdot x[n-l]\},\ l = 0,1,\cdots,k-1.$

$\boldsymbol{R}_k$ is the auto-correlation matrix of the signal $x[n]$, and

$\boldsymbol{p}_k = E\{\boldsymbol{x}_k[n] \cdot s[n]\}$ is the cross-correlation vector between $\boldsymbol{x}_k[n]$ and $s[n]$.

Estimate $\boldsymbol{g}_{3,o} = [g_0, g_1, g_2]$ for the following data:

$x[k] = [3 \quad 2 \quad -2 \quad -2 \quad 2 \quad 2 \quad -2 \quad -2 \quad -1]$;

$s[k] = [\ 1, 0, -1, 0, 1, 0, -1\ ]$ ;

$f = 3$;

*Remark:* $h = [3, 2, 1]$