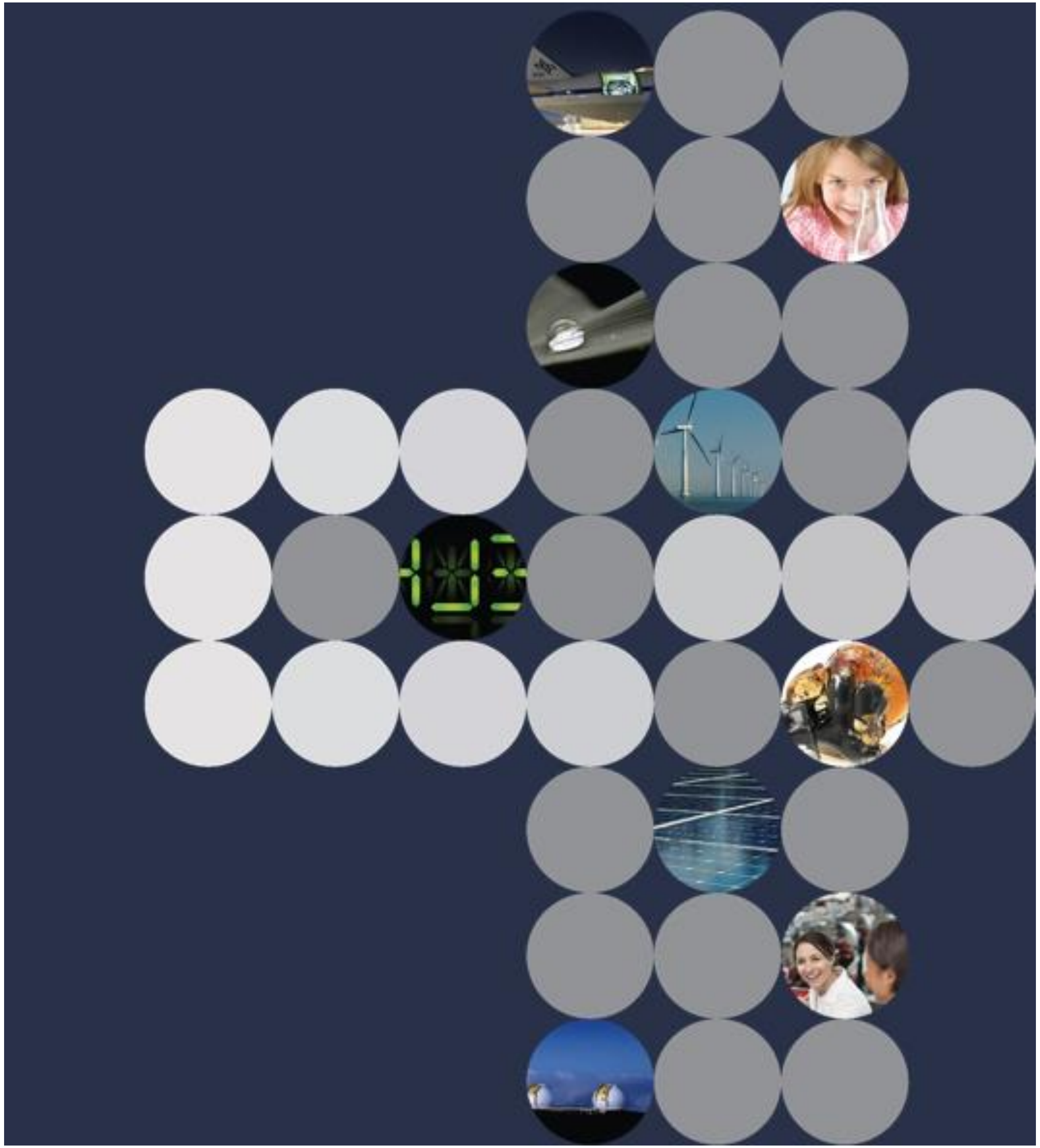
EE512 – Applied Biomedical Signal Processing

Linear Models II

João Jorge

CSEM Signal Processing Group



Outline

Theory

- Autoregressive (AR) models
  - Parametric estimates: power spectrum
- Moving average (MA) models
  - Parameter estimation
- Autoregressive moving average (ARMA) models
  - Parameter estimation
  - Parametric estimates: power spectrum
- Harmonic models
  - Parameter estimation
- Linear system identification
  - Parameter estimation
  - Pre-analysis

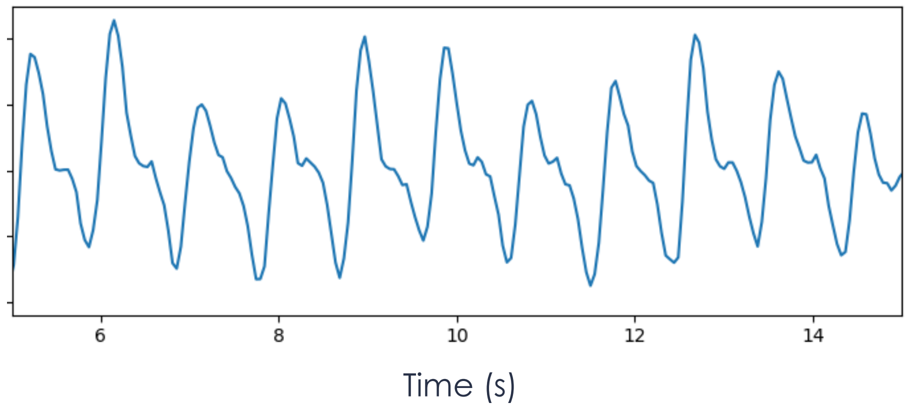
Practical session

- Based on Jupyter notebook
- See instructions in the notebook

AR models: power spectral density

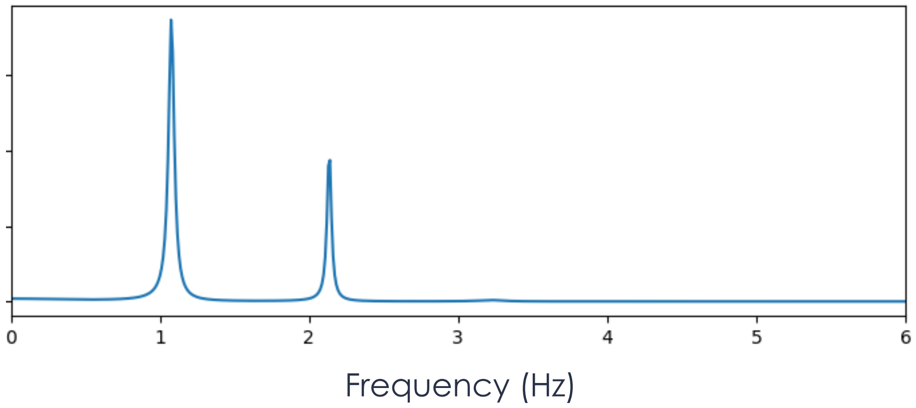
- AR models also provide a parametric estimate for the **power spectral density (PSD)**, $\hat{P}_x(f)$, of the approximated signal $x(n)$

$\boldsymbol{x(n)}$ (a PPG signal in normal heart function)



To frequency domain

$\boldsymbol{\hat{P}_x(f)}$ (spectrum dominated by heart rate and 1st harmonic)



AR models: power spectral density

- AR models also provide a parametric estimate for the **power spectral density (PSD)**, $\hat{P}_x(f)$, of the approximated signal $x(n)$
- Once the AR coefficients a_l , $l = 1, \ldots, p$ and excitation variance σ_ε^2 are obtained, then:

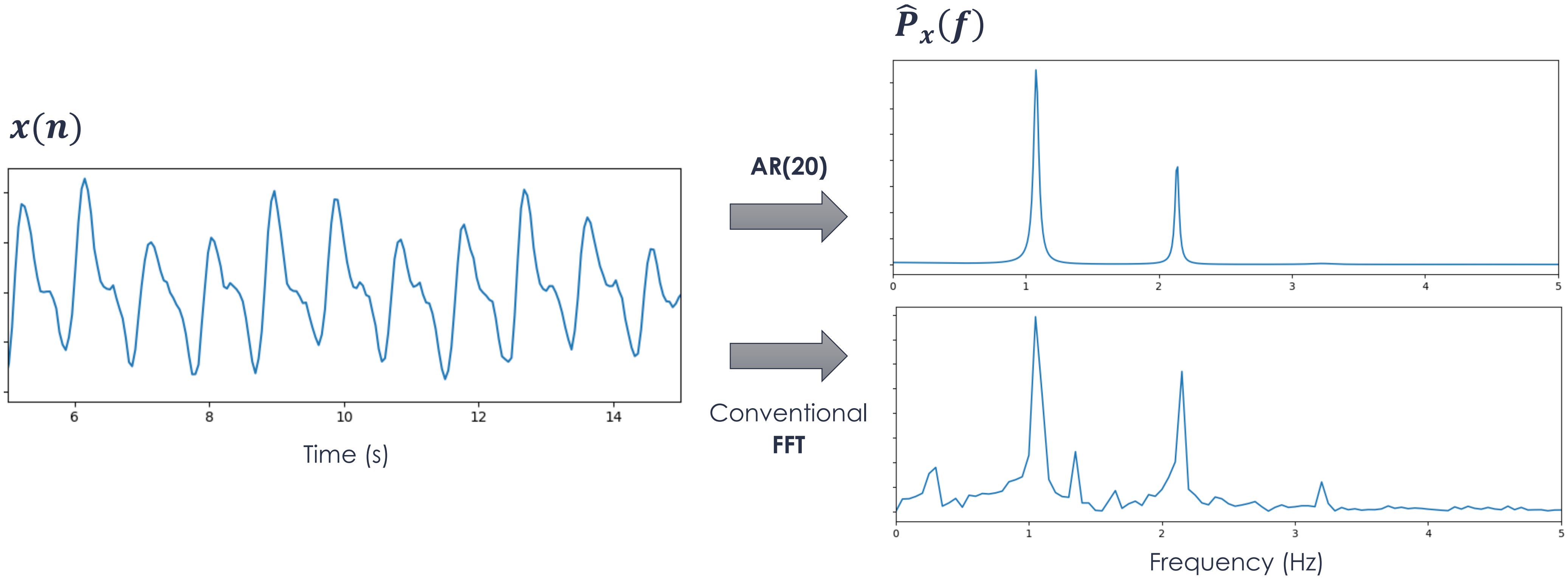
$$\hat{P}_x(f) = \frac{\sigma_\varepsilon^2}{f_s \left| 1 + \sum_{l=1}^{p} a_l e^{-2\pi l j \frac{f}{f_s}} \right|^2}$$

with f the frequency in Hz, and f_s the sampling frequency in Hz

A derivation can be found in:
Hayes, *Statistical Digital Signal Processing and Modeling*, Wiley, 1996, chapter 3.6

AR models: power spectral density

- AR models also provide a parametric estimate for the **power spectral density (PSD)**, $\hat{P}_x(f)$, of the approximated signal $x(n)$



Autoregressive moving average (ARMA) models

- The ARMA model can be simplified when appropriate:

Autoregressive moving average (ARMA)

$$x(n) = \sum_{l=0}^{q} b(l)\varepsilon(n-l) - \sum_{l=1}^{p} a(l)x(n-l)$$

$a(l) = 0$

$b(l) = \delta(l)$

Moving average (MA)

$$x(n) = \sum_{l=0}^{q} b(l)\varepsilon(n-l)$$

Autoregressive (AR)

$$x(n) = \varepsilon(n) - \sum_{l=1}^{p} a(l)x(n-l)$$

Autoregressive moving average (ARMA) models

- The ARMA model can be simplified when appropriate:

Autoregressive moving average (ARMA)

$$x(n) = \sum_{l=0}^{q} b(l)\varepsilon(n-l) - \sum_{l=1}^{p} a(l)x(n-l)$$

$a(l) = 0$

$b(l) = \delta(l)$

Moving average (MA)

$$x(n) = \sum_{l=0}^{q} b(l)\varepsilon(n-l)$$

Autoregressive (AR)

$$x(n) = \varepsilon(n) - \sum_{l=1}^{p} a(l)x(n-l)$$

Moving average (MA) models

$$\varepsilon(n) \longrightarrow \boxed{H(z)} \longrightarrow x(n)$$

$$H(z) = \frac{A(z)}{1} = \sum_{k=0}^{q} b(k)z^{-k}$$

Why use MA models?

- They are not that widely used ($H(z)$ can be seen as a causal FIR filter);
- Nonetheless, they are an important component of the ARMA model.

Moving average (MA) models

In general:

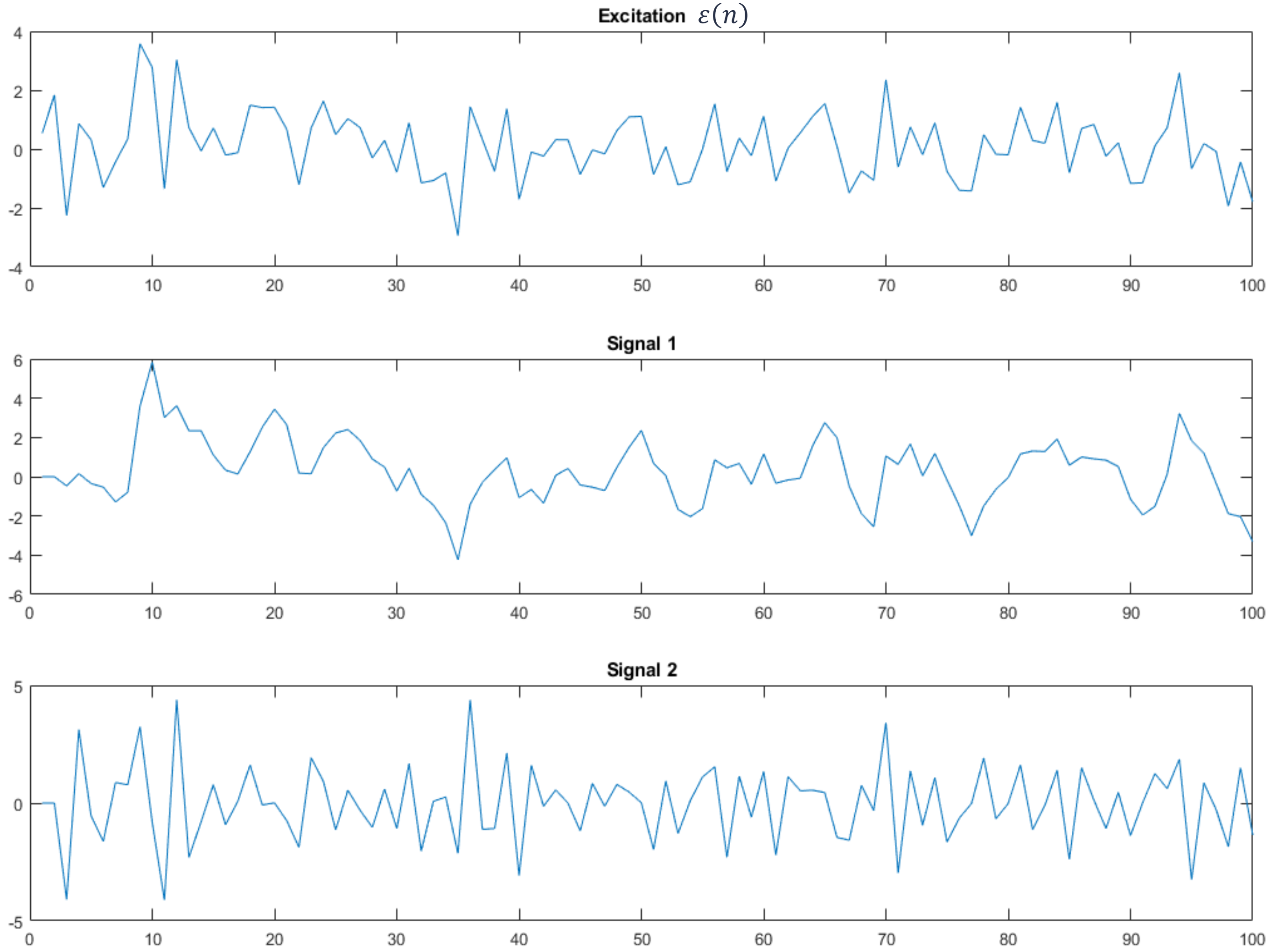
$$x(n) = \sum_{l=0}^{q} b(l)\varepsilon(n-l)$$

Example 1

$$x(n) = +1.0\ \varepsilon(n) + 0.8\ \varepsilon(n-1) + 0.6\ \varepsilon(n-2)$$

Example 2

$$x(n) = +1.0\ \varepsilon(n) - 1.0\ \varepsilon(n-1)$$



MA models: Yule-Walker equations

How can we estimate the parameters?

- Following the Yule-Walker approach, we obtain:

$$R_{xx}(k) = \begin{cases} \sigma_\varepsilon^2 \sum_{l=1}^{q-k} b_l b_{l+k} & \text{for } k = 0, \dots, q \\ 0 & \text{for } k > q \end{cases}$$

- Unfortunately, these equations are **non-linear** with respect to b_l, and less straight-forward to solve.

MA models: parameter estimation

- As an alternative, we first consider approximating the MA model by an infinite-order AR(∞) model:

$$B(z) = \sum_{k=0}^{q} b(k) z^{-k} = \frac{1}{A_\infty(z)} = \frac{1}{\alpha + \sum_{l=1}^{\infty} a_l z^{-l}}$$

Note: this approach is enabled by the **duality** between MA and AR: a finite-order MA process can be represented by an infinite-order AR, and vice-versa. Consider that:

$$\sum_{k=0}^{q} b(k) z^{-k} = b(0) \prod_{i=1}^{q} (1 - z_i\, z^{-1}) = b(0) \prod_{i=1}^{q} \frac{1}{\sum_{l=0}^{\infty} (z_i\, z^{-1})^l} \rightarrow \frac{1}{\alpha + \sum_{l=1}^{\infty} a_l z^{-l}}$$

$$\sum_{l=0}^{\infty} d^l = \frac{1}{1-d}$$, for $|d| < 1$

MA models: parameter estimation

$$B(z) = \frac{1}{A_\infty(z)}$$

- Given the above relationship, we have $B(z)A_\infty(z) = 1$. The inverse Z-transform of this product is a convolution, and that of the constant 1 is the impulse function. This yields:

$$a_m + \sum_{k=1}^{q} b_k a_{m-k} = \delta(m)$$

- This can generate a set of equations by varying the value of m.

MA models: parameter estimation

- **In theory**, we would then estimate the parameters of the AR(∞) model for the signal x, write a set of q equations for q values of m, and solve the equation set for the $b_{1,\dots,q}$.

- **In practice**, the AR model must have a finite order m, with $m \gg q$. The relationship $B(z)A_m(z) \approx 1$ can translate into:

$$a_m + \sum_{k=1}^{q} b_k a_{m-k} = e_{MA}(m)$$

- Remind the linear prediction problem in AR models (previous lecture): $e(n) = x(n) + \sum_{l=1}^{p} a_l x(n-l)$. With a playing the role of x, the equation above can be tackled similarly, by minimizing e_{MA}^2, i.e. the MSE.

Autoregressive moving average (ARMA) models

Autoregressive moving average (ARMA)

$$x(n) = \sum_{l=0}^{q} b(l)\varepsilon(n-l) - \sum_{l=1}^{p} a(l)x(n-l)$$

$a(l) = 0$

$b(l) = \delta(l)$

Moving average (MA)

$$x(n) = \sum_{l=0}^{q} b(l)\varepsilon(n-l)$$

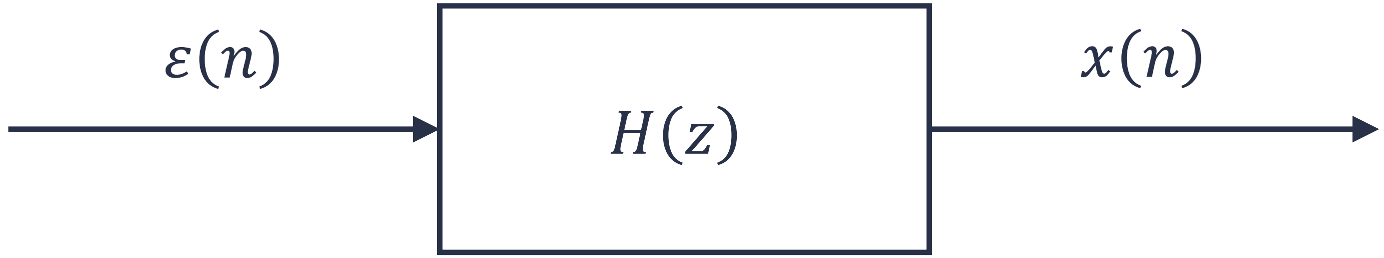
Autoregressive (AR)

$$x(n) = \varepsilon(n) - \sum_{l=1}^{p} a(l)x(n-l)$$

Autoregressive moving average (ARMA) models

Autoregressive moving average (ARMA)

$$x(n) = \sum_{l=0}^{q} b(l)\varepsilon(n-l) - \sum_{l=1}^{p} a(l)x(n-l)$$



$$H(z) = \frac{A(z)}{B(z)} = \frac{\sum_{k=0}^{q} b(k)z^{-k}}{1 + \sum_{k=1}^{p} a(k)z^{-k}} \quad \begin{matrix} \} \ MA(q) \\ \} \ AR(p) \end{matrix}$$

ARMA models: Yule-Walker equations

How can we estimate the parameters?

- The most robust approach consists in estimating the AR and MA parts separately. We apply the Y.W. approach on the difference equation:

$$x(n) = -\sum_{i=1}^{p} a_i x(n-i) + \sum_{j=0}^{q} b_j \varepsilon(n-j)$$

$$R_{xx}(k) = E\{x(n)x(n-k)\}$$
$$= -\sum_{i=1}^{p} a_i E\{x(n-i)x(n-k)\} + \sum_{j=0}^{q} b_j E\{\varepsilon(n-j)x(n-k)\}$$

ARMA models: Yule-Walker equations

- Again defining a transfer function G:

$$\varepsilon(n) \longrightarrow \boxed{G(z)} \longrightarrow x(n)$$

$$x(n) = \sum_{m=0}^{\infty} g(m)\,\varepsilon(n-m)$$

- Then, for the term:

$$E\{\varepsilon(n-j)x(n-k)\} = \sum_{m=0}^{\infty} g(m)E\{\varepsilon(n-j)\varepsilon(n-k-m)\}$$

$$= \sum_{m=0}^{\infty} g(m)R_{\varepsilon\varepsilon}(k+m-j)$$

$$= \begin{cases} g(j-k)\sigma_{\varepsilon}^{2} & if\ j \geq k \\ 0 & otherwise \end{cases}$$

Note that:

- $R_{\varepsilon\varepsilon}(l)$ is only non-zero at $l = 0$;
- $m \geq 0$

ARMA models: Yule-Walker equations

- We then obtain the **extended Yule-Walker equations**:

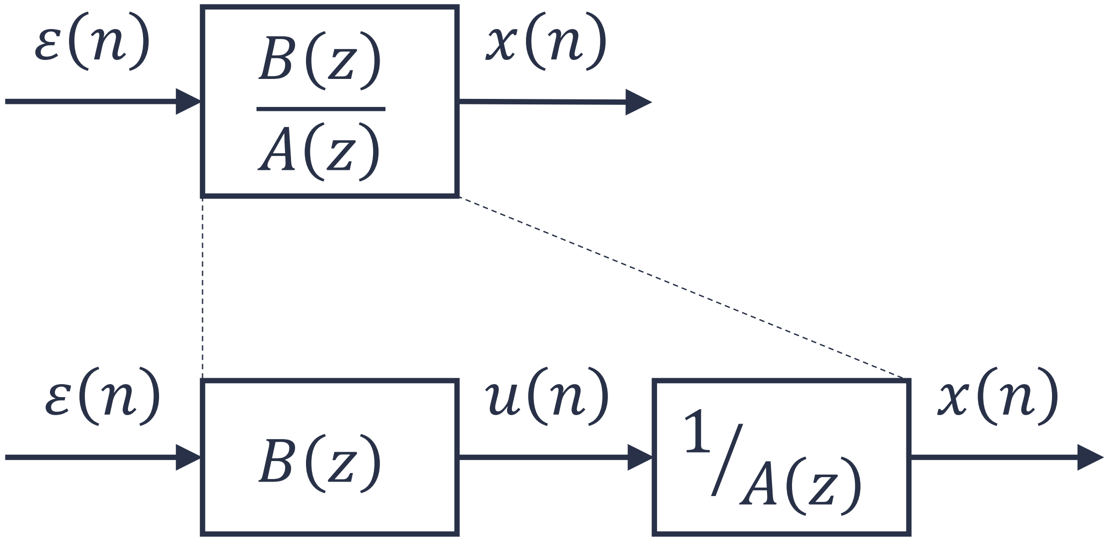
$$R_{xx}(k) = \begin{cases} -\sum_{i=1}^{p} a_i R_{xx}(k-i) + \sigma_\varepsilon^2 \sum_{j=0}^{q} b_j g(j-k) & \text{for } 0 \leq k \leq q \\ -\sum_{i=1}^{p} a_i R_{xx}(k-i) & \text{for } k > q \end{cases}$$

Note that g is not known. How to tackle this?

- First, we can use p equations from $k > q$ to estimate the AR parameters;

ARMA models: Yule-Walker equations

- Then, we approach the ARMA model in two steps:



- Having $A(z)$, the intermediate signal $u(n)$ can be computed;
- The MA part $B(z)$ can then be estimated as previously proposed.

ARMA models: order selection

Selection criteria

Several statistical criteria have been proposed:

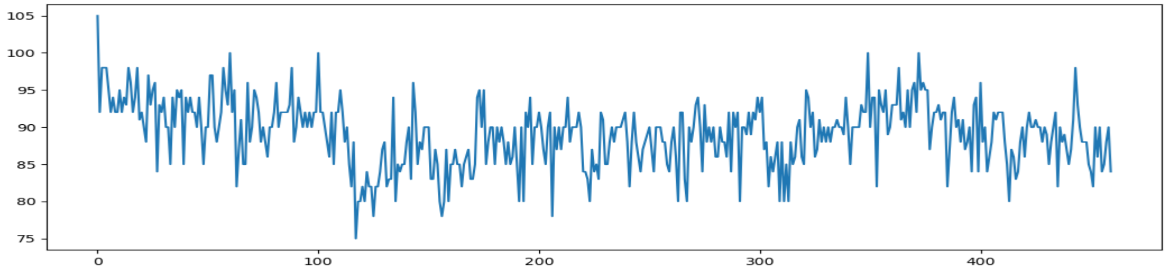
- **Akaike information criterion (AIC):** $$AIC(p,q) = \underbrace{N \ln\left(\sigma_{p,q}^2\right)}_{\text{Decreases with } \Uparrow p,q} + \underbrace{2(p+q)}_{\text{Increases with } \Uparrow p,q}$$

- **Minimum description length (MDL):** $$MDL(p,q) = N \ln\left(\sigma_{p,q}^2\right) + (p+q)\ln(N)$$

- More complex since two parameters (p,q) need to be jointly optimized;
- The autocorrelation, $R_{xx}(k)$ and $R_{uu}(k)$, may provide guidance.

Example: modeling a blood pressure signal

Measured daily systolic blood pressure



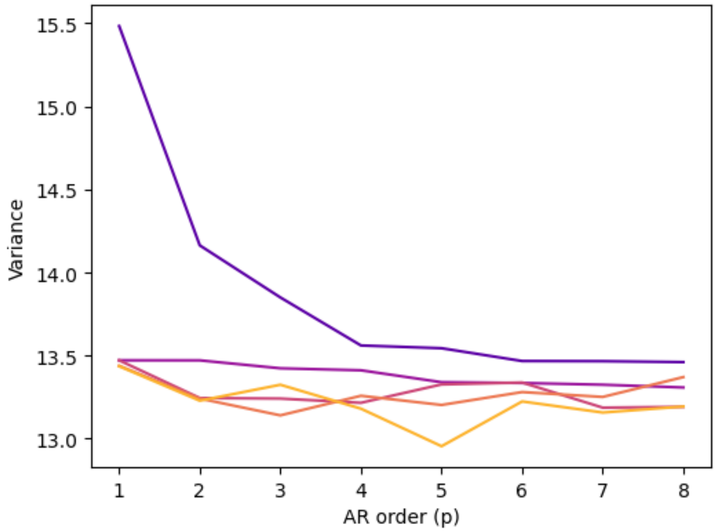
Estimate ARMA models of varying (p,q) order

$$x(n) = \sum_{l=0}^{q} b(l)\varepsilon(n-l) - \sum_{l=1}^{p} a(l)x(n-l)$$

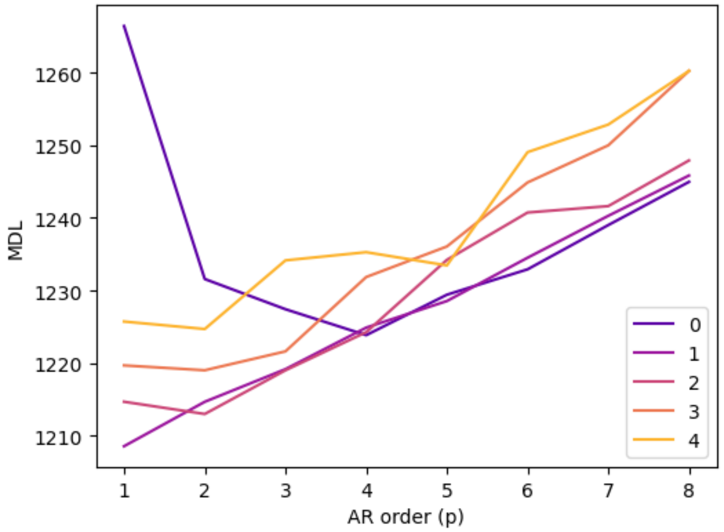
Python: *statsmodel.tsa.arima*

For each (p,q), estimate $\boldsymbol{\sigma^2_{p,q}}$ and **MDL**

Excitation variance $\boldsymbol{\sigma^2_{p,q}}$



MDL

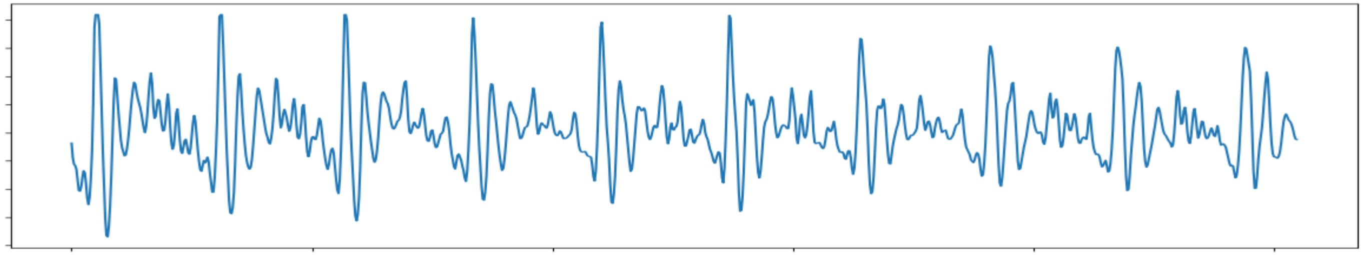


MA order (q)

EPFL csem

Example: modeling a speech signal

Measured signal from speech (/a/ sound)



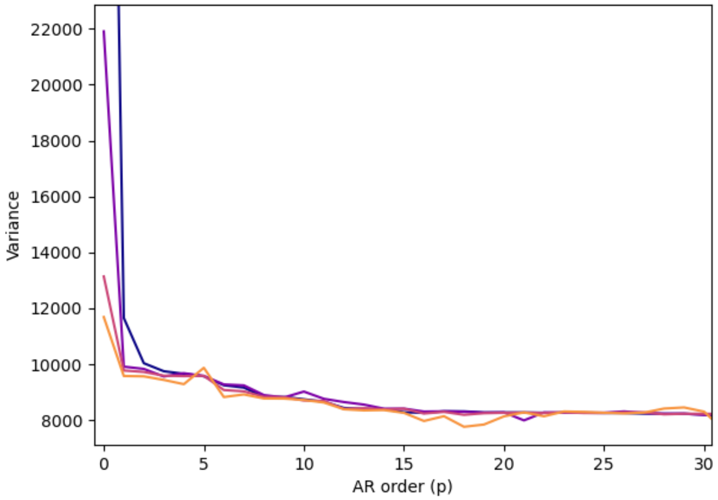
Estimate ARMA models of varying (p,q) order

$$x(n) = \sum_{l=0}^{q} b(l)\varepsilon(n-l) - \sum_{l=1}^{p} a(l)x(n-l)$$

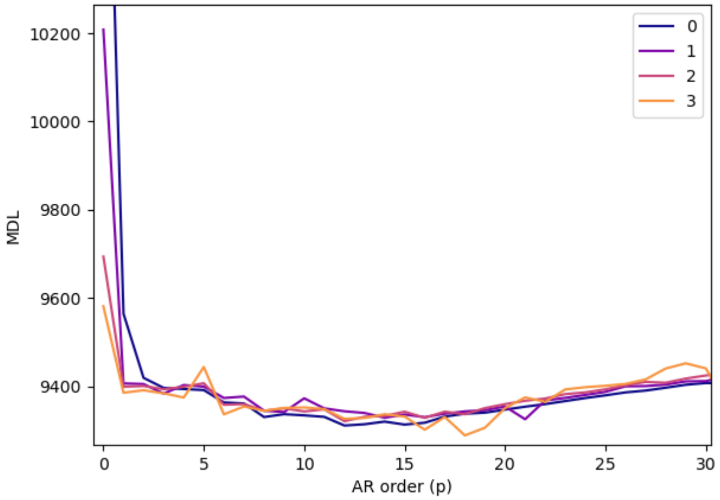
Python: *statsmodel.tsa.arima*

For each (p,q), estimate $\boldsymbol{\sigma^2_{p,q}}$ and **MDL**

Excitation variance $\boldsymbol{\sigma^2_{p,q}}$



MDL



MA order (q)

EPFL csem

ARMA models: power spectral density

- ARMA models also provide a parametric estimate for the **power spectral density** of the approximated signal x.
- Once all the model coefficients a_i, b_j and the excitation variance σ_ε^2 are obtained, then:

$$\hat{P}_x(f) = \frac{\sigma_\varepsilon^2 \left|\sum_{m=1}^{q} b_m e^{-2\pi m j \frac{f}{f_s}}\right|^2}{f_s \left|1 + \sum_{n=1}^{p} a_n e^{-2\pi n j \frac{f}{f_s}}\right|^2}$$

with f the frequency in Hz, and f_s the sampling frequency in Hz

A derivation can be found in:
Hayes, *Statistical Digital Signal Processing and Modeling*, Wiley, 1996, chapter 3.6

Exam-like question

A signal x with sampling frequency f_s is generated using the following AR model:

$$x(n) = 0.9\, x(n-1) + \varepsilon(n)$$

with ε being white noise with variance σ_ε.

1. Write the expression for the power spectral density of x in terms of these parameters.

2. At which positive normalized frequency does the power spectral density of x have a peak?

Exam-like question

$$x(n) = 0.9\,x(n-1) + \varepsilon(n)$$

1. Write the expression for the power spectral density of x in terms of these parameters.

From the expression we can see this is an AR($p = 1$) model, with $a_1 = -0.9$.

The general expression for the power spectral density is:

$$\hat{P}_x(f) = \frac{\sigma_\varepsilon^2}{f_s \left| 1 + \sum_{l=1}^{p} a_l e^{-2\pi l j \frac{f}{f_s}} \right|^2}$$

And so for this case the expression becomes:

$$\hat{P}_x(f) = \frac{\sigma_\varepsilon^2}{f_s \left| 1 - 0.9 e^{-2\pi j \frac{f}{f_s}} \right|^2}$$

Exam-like question

2. At which positive normalized frequency does the power spectral density of x have a peak?

The denominator can be developed as follows:

$$\begin{aligned}\left|1-0.9e^{-2\pi j\frac{f}{f_s}}\right|^2 &= \left(1-0.9e^{-2\pi j\omega}\right)^*\left(1-0.9e^{-2\pi j\omega}\right)\\ &= \left(1-0.9e^{+2\pi j\omega}\right)\left(1-0.9e^{-2\pi j\omega}\right)\\ &= 1-0.9e^{-2\pi j\omega}-0.9e^{+2\pi j\omega}+0.81\\ &= 1.81-1.8\cos(2\pi\omega)\end{aligned}$$

We can see that this term is lowest at $\omega = 0$, and then increases monotonically (driven by the cosine) to its highest value at $\omega = \frac{1}{2}$. Since this term is the denominator of the power spectral density, we can then conclude, inversely, that the PSD has its peak at $\omega = 0$ (where the denominator is lowest).

Note: Looking at the model, $x(n) = 0.9\, x(n\text{-}1) + \varepsilon(n)$, we can intuitively see that this result makes sense : each element $x(n)$ depends almost as much on the previous sample as on the current excitation value, leading to a behavior similar to that of a low-pass filter, and thus higher power towards low frequencies.

Harmonic models

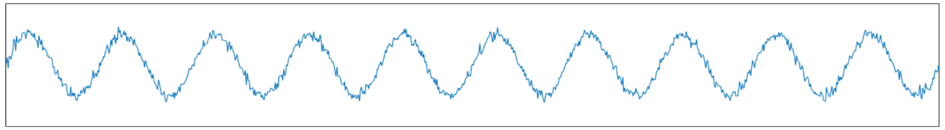
- Another useful class of linear models is the sum of **complex exponentials** in white noise:

$$x(n) = \sum_{i=1}^{p} A_i e^{j2\pi f_i n} + w(n) \qquad \text{with } A_i \text{ also complex}$$

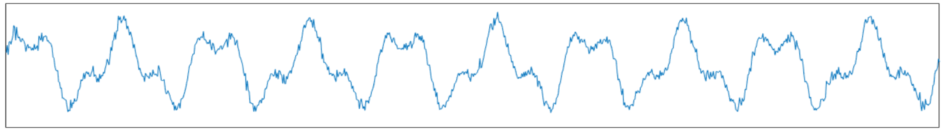
- A specific case of this is the (real) combination of sinusoids:

$$x(n) = \sum_{j=1}^{q} B_j sin\left(2\pi f_j n + \phi_j\right) + w(n)$$

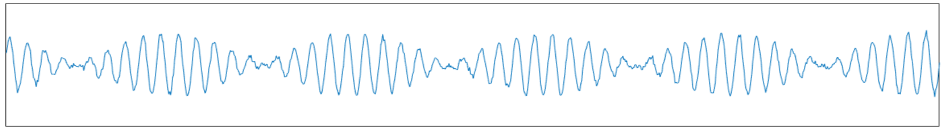
$(\omega_1 = 10)$



$(\omega_1 = 10, \omega_2 = 25)$



$(\omega_1 = 50, \omega_2 = 55)$



Reminder: $\cos\theta = \dfrac{e^{j\theta} + e^{-j\theta}}{2} \qquad \sin\theta = \dfrac{e^{j\theta} - e^{-j\theta}}{2j}$

Harmonic models: eigendecomposition methods

How can we estimate the parameters?

- To provide some intuition: for a complex exponential with noise, $x(n) = A_1 e^{jn\omega_1} + w(n)$, the autocorrelation of x takes the form:

$$r_x(k) = P_1 e^{jk\omega_1} + \sigma_w^2 \delta(k) \qquad \text{with } P_1 = |A_1|^2$$

- Reminder: autocorrelation and its matrix (up to a lag p) are defined as:

$$r_x(k) = E\{x(n)x^*(n-k)\} \qquad R_x = \begin{bmatrix} r_x(0) & r_x^*(1) & \cdots & r_x^*(p) \\ r_x(1) & r_x(0) & \cdots & r_x^*(p-1) \\ \vdots & \vdots & \ddots & \vdots \\ r_x(p) & r_x(p-1) & \cdots & r_x(0) \end{bmatrix}$$

Harmonic models: eigendecomposition methods

- In this case we get: $R_x = R_s + R_n$ defined up to a given lag $M-1$

$$R_s = P_1 \begin{bmatrix} 1 & e^{-j\omega_1} & \cdots & e^{-j(M-1)\omega_1} \\ e^{j\omega_1} & 1 & \cdots & e^{-j(M-2)\omega_1} \\ \vdots & \vdots & \ddots & \vdots \\ e^{j(M-1)\omega_1} & e^{j(M-2)\omega_1} & \cdots & 1 \end{bmatrix} \qquad R_n = \sigma_w^2 \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

- Due to their structure, it can be shown that:
  - R_s has a rank of 1, and thus 1 non-zero eigenvalue: $\lambda_{s1} = MP_1$;
  - R_n has full rank, with eigenvalues equal to σ_w^2 ;
- As a result, the largest eigenvalue of R_x will be $\lambda_{max} = MP_1 + \sigma_w^2$, and the remaining $M-1$ eigenvalues will be equal to σ_w^2.

Harmonic models: eigendecomposition methods

- **Eigendecomposition methods** use those properties to estimate σ_w^2 and P_1 based on the eigenvalues, and then ω_1 from the eigenvector.

- For **sums of p complex exponentials**, we obtain:
  - p eigenvectors from R_s that form the **"signal subspace"**, and,
  - $M - p$ eigenvectors that form the **"noise subspace"**.

  The two are orthogonal; this property can be used to estimate the ω_i.

- The **Pisarenko harmonic retrieval** approach (described next) makes use of these properties.

How can we estimate the parameters of sinusoids embedded in noise?

- Reminder: a sinusoid can be represented by an AR(2) process:

$$x(n) = \sin(2\pi f n) = 2\cos(2\pi f)x(n-1) - x(n-2)$$

Using: $\sin(a+b) + \sin(a-b) = 2\sin(a)\cos(b)$

with $a = 2\pi f(n-1)$ and $b = 2\pi f$

- The denominator of the transfer function $H(z)$ is then:

$$1 - 2\cos(2\pi f)\, z^{-1} + z^{-2}$$

The poles (zeros of this denominator) are equal to $e^{\pm j2\pi f}$.

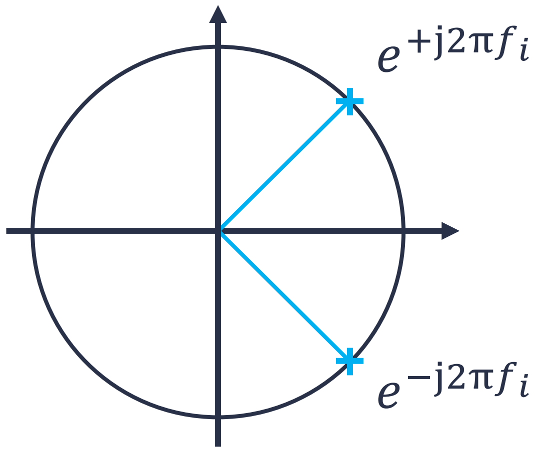
Harmonic models: Pisarenko harmonic retrieval

- A signal composed of a sum of p sinusoids can then be represented by an AR($2p$) model without excitation:

$$x(n) = -\sum_{i=1}^{2p} a_i x(n-i)$$

with a_i the coefficients of the polynomial:

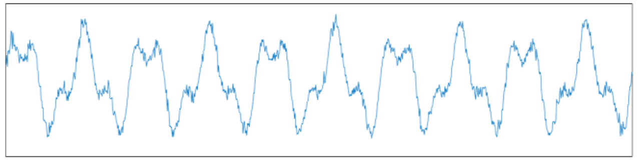
$$z^{2p} + a_1 z^{2p-1} + \cdots + a_{2p-1} z + a_{2p} = \prod_{i=1}^{p} (z - z_i)\,(z - \overline{z_i})$$



The roots z_i are on the unit circle, and the angle relates to the frequency f_i.

Harmonic models: Pisarenko harmonic retrieval

- We now consider a signal with sinusoids perturbed by white noise:



$$y(n) = \underbrace{x(n)}_{\substack{p \text{ sinusoids} \\ \text{described as AR}(2p)}} + \underbrace{w(n)}_{\text{white noise}} = w(n) - \sum_{i=1}^{2p} a_i x(n-i)$$

with $R_{ww}(k) = \sigma_w^2 \delta(k)$ and $E\{x(n)w(n-k)\} = 0$

- Replacing $x(n-i)$ with $y(n-i) - w(n-i)$, one can write:

$$\sum_{i=0}^{2p} a_i y(n-i) = \sum_{i=0}^{2p} a_i w(n-i) \qquad \text{with } a_0 = 1$$

Harmonic models: Pisarenko harmonic retrieval

- This equation can be re-written in matrix form:

$$\boldsymbol{y}^T\boldsymbol{a} = \boldsymbol{w}^T\boldsymbol{a}$$

with:

$$\boldsymbol{y}^T = [y(n) \quad y(n-1) \quad \cdots \quad y(n-2p)]$$

$$\boldsymbol{w}^T = [w(n) \quad w(n-1) \quad \cdots \quad w(n-2p)]$$

$$\boldsymbol{a}^T = [1 \quad a_1 \quad \cdots \quad a_{2p}]$$

- Multiplying by $\boldsymbol{y}$ and getting the expectation:

$$E\{\boldsymbol{y}\boldsymbol{y}^T\boldsymbol{a}\} = E\{\boldsymbol{y}\boldsymbol{w}^T\boldsymbol{a}\}$$

$$E\{\boldsymbol{y}\boldsymbol{y}^T\}\boldsymbol{a} = E\{\boldsymbol{y}\boldsymbol{w}^T\}\boldsymbol{a}$$

Harmonic models: Pisarenko harmonic retrieval

- Note that:

$$E\{\boldsymbol{y}\boldsymbol{y}^T\} = \begin{bmatrix} R_{yy}(0) & R_{yy}(1) & \cdots & R_{yy}(2p) \\ R_{yy}(1) & R_{yy}(0) & \cdots & R_{yy}(2p-1) \\ \vdots & \vdots & \ddots & \vdots \\ R_{yy}(2p) & R_{yy}(2p-1) & \cdots & R_{yy}(0) \end{bmatrix} = R_{yy}$$

- For the right-side, defining: $\boldsymbol{x}^T = [x(n) \quad x(n-1) \quad \cdots \quad x(n-2p)]$, we can write:

$$E\{\boldsymbol{y}\boldsymbol{w}^T\} = E\{(\boldsymbol{x}+\boldsymbol{w})\boldsymbol{w}^T\} = E\{\boldsymbol{w}\boldsymbol{w}^T\} = \sigma_w^2 I$$

$$E\{x(n)w(n-k)\} = 0$$

Harmonic models: Pisarenko harmonic retrieval

- With these results on $E\{\boldsymbol{y}\boldsymbol{y}^T\}$ and $E\{\boldsymbol{w}\boldsymbol{y}^T\}$, we therefore obtain:

$$R_{yy}\boldsymbol{a} = \sigma_w^2\boldsymbol{a}$$

Autocorrelation matrix of measured signal y (from 0 to $2p$)

Coefficients of AR($2p$) - unknown

Variance of noise - unknown

This equality shows that $\boldsymbol{a}$ is an eigenvector to the eigenvalue σ_w^2.

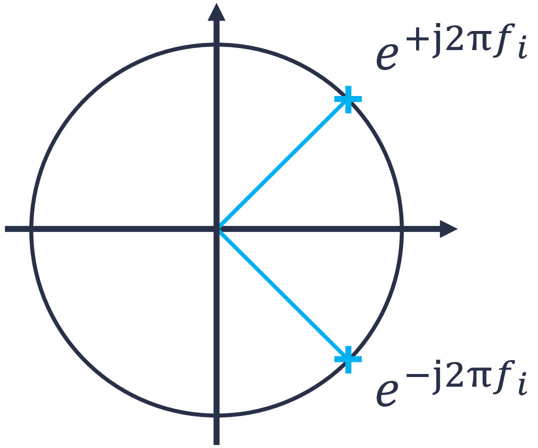
- As discussed before, σ_w^2 is the smallest eigenvalue of R_{yy}. And if R_{yy} is larger than $(2p+1) \times (2p+1)$, then σ_w^2 is repeated (the "noise subspace").
- If the number of sinusoids p is not known a priori, we can estimate it by computing the eigenvalues of R_{yy} for increasingly larger values of p until it stabilizes.

Harmonic models: Pisarenko harmonic retrieval

- Once $\boldsymbol{a}$ has been obtained, the **frequencies** f_i are obtained from the roots z_i of:

$$z^{2p} + a_1 z^{2p-1} + \cdots + a_{2p-1} z + a_{2p} = 0$$

The roots should be of the form: $z_i = e^{\pm j2\pi f_i}$



- The **sinusoid powers** $\boldsymbol{P_i}$ can also be estimated (see next slide).

- The **phases** cannot be obtained in this way (as the approach is based on autocorrelation).

Harmonic models: Pisarenko harmonic retrieval

- To estimate the **sinusoid powers $\boldsymbol{P_i}$** , it can be shown that the autocorrelation of y obeys:

$$R_{yy}(0) = \sigma_w^2 + \sum_{i=1}^{p} P_i$$

$$R_{yy}(k) = \sum_{i=1}^{p} P_i \cos 2\pi f_i k$$

- The equation for $R_{yy}(0)$ expresses the fact that the total power of y is equal to the sum of the sinusoid powers and the noise power.

Harmonic models: Pisarenko harmonic retrieval

- The equation for $R_{yy}(k)$ yields a set of linear questions:

$$\begin{bmatrix} R_{xx}(1) \\ R_{xx}(2) \\ \vdots \\ R_{xx}(p) \end{bmatrix} = \begin{bmatrix} \cos(2\pi f_1) & \cos(2\pi f_2) & \cdots & \cos(2\pi f_p) \\ \cos(2\pi f_1 2) & \cos(2\pi f_2 2) & \cdots & \cos(2\pi f_p 2) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(2\pi f_1 p) & \cos(2\pi f_2 p) & \cdots & \cos(2\pi f_p p) \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_p \end{bmatrix}$$

- The noise power can then be obtained through the equation for $R_{yy}(0)$.

Harmonic models: Pisarenko harmonic retrieval

- **Summary of procedure:**

We start with a measured signal $y(n)$

1. Choose number of sinusoids p for our model
2. Estimate $R_{yy}(k = 0, \dots, 2p)$
3. Estimate the eigenvalues of R_{yy}
4. If the smallest eigenvalues are not stable, repeat from 1. with higher p
5. Estimate eigenvector $\boldsymbol{a}$ for smallest eigenvalue of R_{yy}
6. Estimate roots of the z-polynomial with $\boldsymbol{a}$ coefficients
7. Extract **sinusoid frequencies** f_i from the roots z_i (use: $z_i = e^{\pm j 2\pi f_i}$)
8. Extract **sinusoid powers** P_i from $R_{yy}(k) = \sum_{i=1}^{p} P_i \cos 2\pi f_i k$, $k = 1, \dots, p$
9. Extract **noise power** σ_w^2 from $R_{yy}(0) = \sigma_w^2 + \sum_{i=1}^{p} P_i$

Harmonic models: Pisarenko harmonic retrieval

- **Summary of procedure:**

…

7. Extract **sinusoid frequencies** f_i from the roots z_i (use: $z_i = e^{\pm j2\pi f_i}$)
8. Extract **sinusoid powers** P_i from $R_{yy}(k) = \sum_{i=1}^{p} P_i \cos 2\pi f_i k$, $k = 1, \dots, p$
9. Extract **noise power** σ_w^2 from $R_{yy}(0) = \sigma_w^2 + \sum_{i=1}^{p} P_i$

$$x(n) = \sum_{i=1}^{q} B_i sin(2\pi \boldsymbol{f}_i n + \phi_i) + w(n)$$

Linear system identification

- **Linear system identification** is the determination of the linear filter that relates two signals optimally.

Why is identification useful?

- Can provide information on the relationship between two signals (in the time and/or the frequency domain);
- Can allow predicting future samples of one of the signals.

Linear system identification

- Consider two known signals x and y, assumed to be jointly stationary. We attempt to model the relationship by:

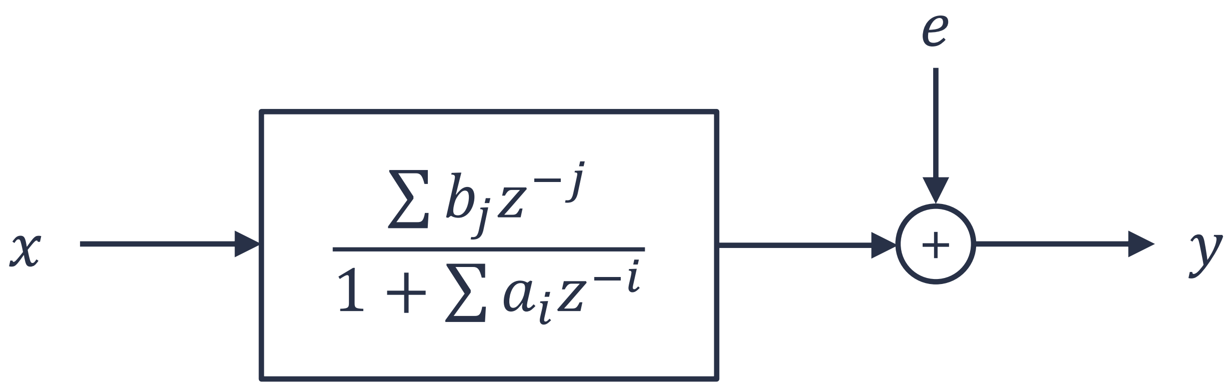
$$y(n) = -\sum_{i=1}^{p} a_i y(n-i) + \sum_{j=-q_0}^{q_1} b_j x(n-j) + e(n)$$

with e a stationary white noise process, uncorrelated with y.

- One can have $q_0 > 0$, i.e. the filter may be non-causal.

Linear system identification

- Schematically:



- The signal e can be seen as an additive interference on y, or as the error due to x and y not being perfectly linearly related – in practice, it is often a mixture of both.
- The variance of e can provide information on the linearity of the relationship between x and y.

Linear system identification

How can we estimate the parameters?

- We can apply a least-squares approach. If N sample pairs are observed: $\{x(n), y(n)\}, n = 1, \ldots, N$, a matrix-form equation can be written:

$$y(n) = -\sum_{i=1}^{p} a_i y(n-i) + \sum_{j=-q_0}^{q_1} b_j x(n-j) + e(n)$$

$$A\boldsymbol{c} = \boldsymbol{d} + \boldsymbol{e}$$

Linear system identification

$$\boldsymbol{A}\boldsymbol{c} = \boldsymbol{d} + \boldsymbol{e}$$

with: $A = [Y \quad X]$

$$Y = \begin{bmatrix} y(i_1 - 1) & \cdots & y(i_1 - p) \\ y(i_1) & \cdots & y(i_1 - p + 1) \\ \vdots & \ddots & \vdots \\ y(N - i_0 - 1) & \cdots & y(N - i_0 - q_1) \end{bmatrix} \qquad X = \begin{bmatrix} x(i_1 + q_0) & \cdots & x(i_1 - q_1) \\ x(i_1 + q_0 + 1) & \cdots & x(i_1 - q_1 + 1) \\ \vdots & \ddots & \vdots \\ x(N - i_0 + q_0) & \cdots & x(N - i_0 - q_1) \end{bmatrix}$$

with:

$$\boldsymbol{c} = \begin{bmatrix} -a_1 \\ \vdots \\ -a_p \\ b_{-q_0} \\ \vdots \\ b_{q_1} \end{bmatrix} \qquad \boldsymbol{d} = \begin{bmatrix} y(i_1) \\ \vdots \\ y(N - i_0) \end{bmatrix} \qquad \boldsymbol{e} = \begin{bmatrix} e(i_1) \\ \vdots \\ e(N - i_0) \end{bmatrix}$$

and the additional indices: $i_0 = \max(q_0, 0) \qquad i_1 = \max(q_1, p) + 1$

Linear system identification

- We then minimize $\boldsymbol{e}$ in the least-squares sense:

$$||\boldsymbol{e}||^2 = (A\boldsymbol{c} - \boldsymbol{d})^T(A\boldsymbol{c} - \boldsymbol{d}) = \boldsymbol{c}^T A^T A\boldsymbol{c} - 2\boldsymbol{d}^T A\boldsymbol{c} + \boldsymbol{d}^T\boldsymbol{d}$$

- The above is clearly a quadratic form on $\boldsymbol{c}$. The minimum (unique if $A^T A$ is positive definite) is obtained by zeroing the gradient:

$$\frac{\partial ||\boldsymbol{e}||^2}{\partial \boldsymbol{c}} = 0 \iff 2A^T A\boldsymbol{c} - 2A^T\boldsymbol{d} = \boldsymbol{0}$$

$$\iff \boldsymbol{c} = (A^T A)^{-1} A^T\boldsymbol{d}$$

Linear system identification

$$y(n) = -\sum_{i=1}^{p} a_i y(n-i) + \sum_{j=-q_0}^{q_1} b_j x(n-j) + e(n)$$

Some remarks:

- Naturally, simplified models can be considered; e.g. a finite impulse response (FIR) filter, by setting $a_i = 0$.

$$y(n) = \sum_{j=-q_0}^{q_1} b_j x(n-j) + e(n)$$

- Other signals than x can be added to the model, by adding columns to A and respective coefficients to $\boldsymbol{c}$.

$$y(n) = -\sum_{i=1}^{p} a_i y(n-i) + \sum_{j=-q_0}^{q_1} b_j x(n-j) + \sum_{k=-l_0}^{l_1} c_k w(n-k) + \cdots + e(n)$$

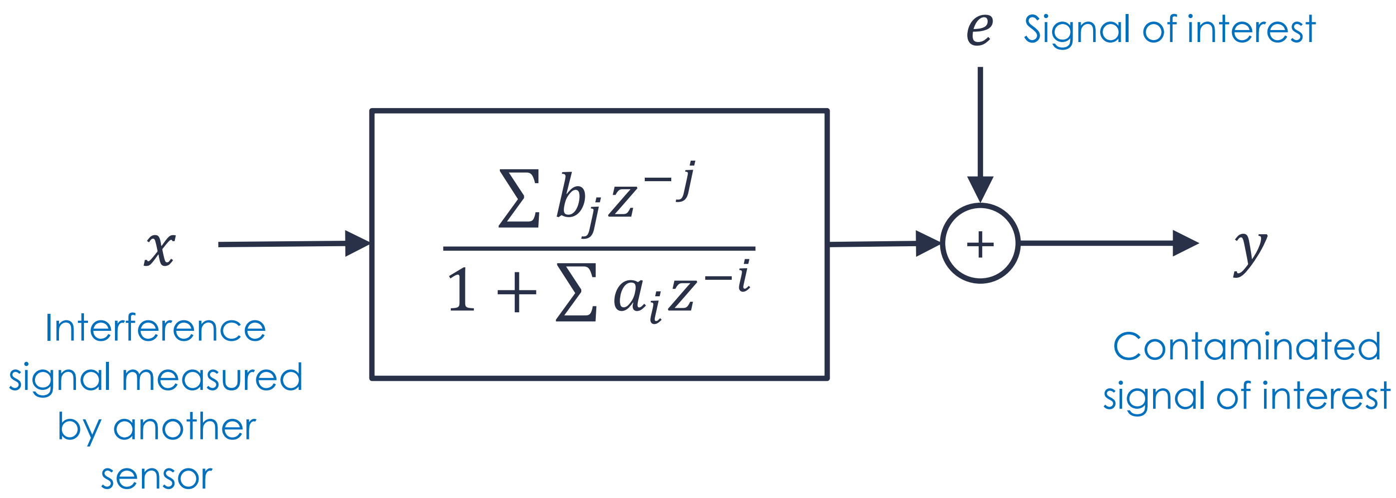
Linear system identification

$$y(n) = -\sum_{i=1}^{p} a_i y(n-i) + \sum_{j=-q_0}^{q_1} b_j x(n-j) + e(n)$$

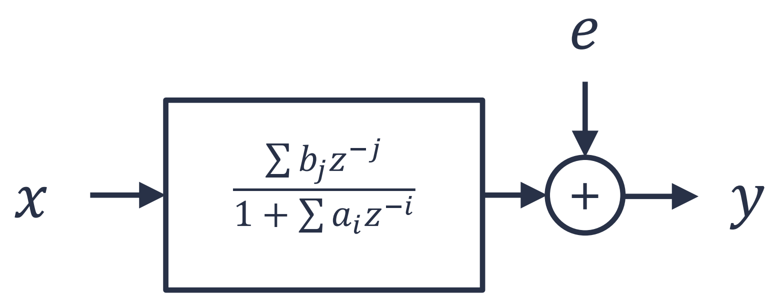
Some remarks:

- In non-stationary problems, the relationship between x and y may be expected to vary over time. This can be addressed with so-called **adaptive identification** techniques.
- We can also use variants of this approach to estimate and remove interferences from the signal – so-called **interference cancellation**.

$$y(n) = -\sum_{i=1}^{p} a_i(n) y(n-i) + \sum_{j=-q_0}^{q_1} b_j(n) x(n-j) + e(n)$$

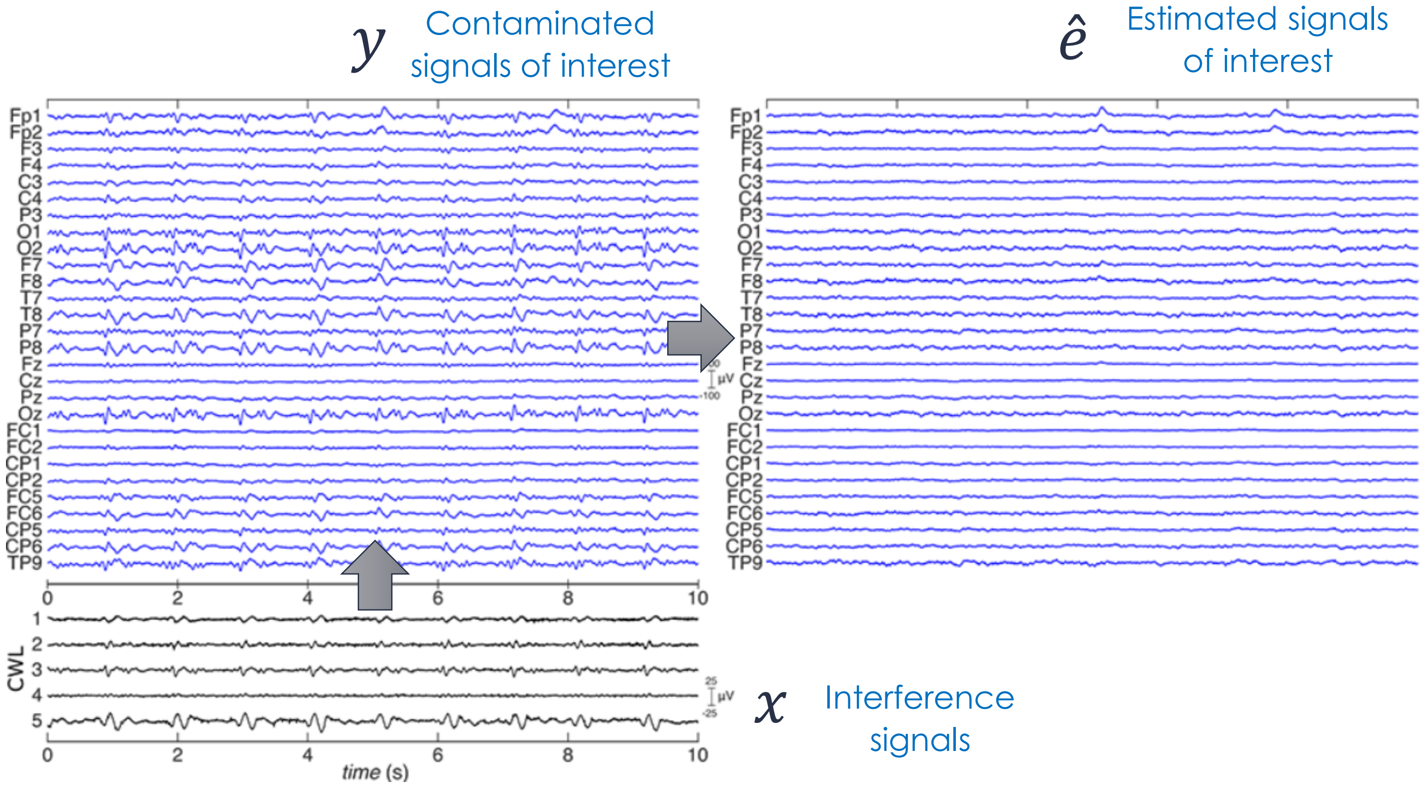


Example: artifact cancellation for EEG recorded inside MRI

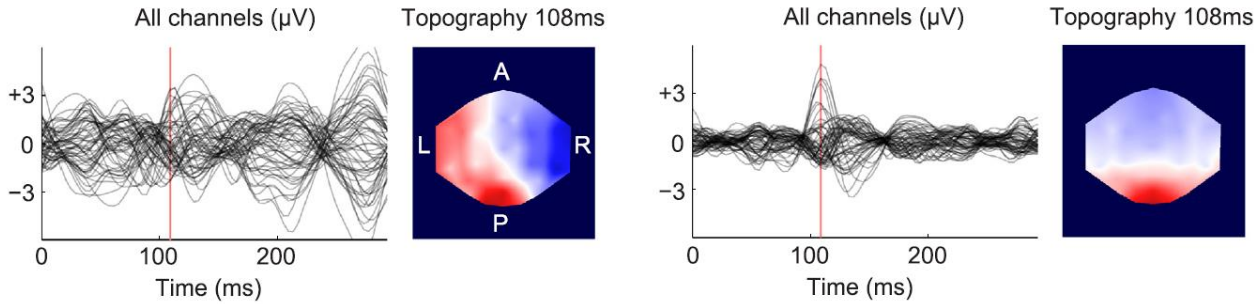


EEG sensors

Loop sensors measure only artifacts



Recovery of visual responses:



Linear system identification: pre-analysis

- Before estimating parameters, **pre-analysis** techniques can be applied to determine if the signals can indeed be related by a linear filter.
- A useful example is the **coherence function**:

$$K_{xy}(f) = \frac{\left|P_{xy}(f)\right|^2}{P_x(f)P_y(f)}$$

with $P_x(f)$ and $P_y(f)$ the signal power, and $P_{xy}(f)$ the cross-spectral density, defined as the Fourier transform of the cross-correlation $R_{xy}(k)$:

$$P_{xy}(f) = \sum_{k=-\infty}^{+\infty} R_{xy}(k)\, e^{-2\pi j f k}$$

Linear system identification: pre-analysis

- It can be shown that $0 \leq K_{xy}(f) \leq 1$ (and the higher, the “better”).
- $K_{xy}(f)$ allows investigating at which frequencies/frequency bands a linear relationship might exist; typically, a value of $K_{xy}(f) > 0.5$ suggests a linear relationship at f.
- When a linear relationship is identified:
  - the parameters q_0 and q_1 can be sought based on the estimate of the cross-correlation $R_{xy}(k)$.
  - If the filter has a recursive part, the autocorrelation $R_{yy}(k)$ can help looking for the parameter p.

$$y(n) = -\sum_{i=1}^{p} a_i y(n-i) + \sum_{j=-q_0}^{q_1} b_j x(n-j) + e(n)$$

Linear system identification: pre-analysis

- The previous insights can thus help building a starting model. The "fine-tuning" can be performed using a criterion such as the **minimum description length (MDL)**:

$$MDL = \underbrace{N\ln(\hat{\sigma}_e^2)}_{\text{Decreases with complexity}} + \underbrace{(p + q_0 + q_1 + 1)\ln(N)}_{\text{Increases with complexity}}$$

with $\hat{\sigma}_e^2$ the estimated variance of the error on the available data.

Some final thoughts

- **Real signals** can contain many contributions. Example for an EEG:
  - Electric potentials from neuronal activity of interest;
  - Random noise;
  - (Very) structured "noise", e.g. other ongoing neuronal activity, power line, head/body movement, cardiac potentials, muscular potentials, etc…
- Because of this, the signal properties can easily deviate from theoretical assumptions and models:
  - Some approaches may seem appropriate in theory but work poorly in practice;
  - Some approaches may seem inappropriate (e.g. too simplistic), but actually work sufficiently well (accurately, robustly) in practice;
- Try different approaches for your problem, see how they behave, understand the limitations at play, don't get discouraged!

References

- M. H. Hayes, *Statistical Digital Signal Processing and Modeling*, Wiley, 1996.
- R. Shiavi, *Introduction to Applied Statistical Signal Analysis*, 2nd Ed., Academic Press 1999.
- S. M. Kay and S. L. Marple, *Spectrum analysis – A modern perspective*, Proc. IEEE, vol. 69, no. 11, Nov. 1981, pp. 1380-1419.

- And for other sources, check **Moodle FAQ**