Adaptive SP & Machine Intelligence

Lecture supplement: ARMA processes

Danilo Mandic

room 813, ext: 46271

Department of Electrical and Electronic Engineering

Imperial College London, UK

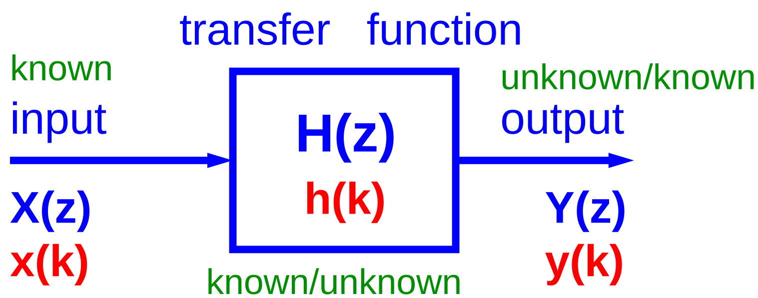
d.mandic@imperial.ac.uk, URL: www.commsp.ee.ic.ac.uk/∼mandic

Aims of this lecture supplement

- To introduce stochastic models for real world data
- Learn how stochastic models shape the spectrum of white noise
- Understand how an ensemble of realisations of a stochastic process is created
- Learn how to derive parameters of linear stochastic models
- Introduce some special cases (autoregressive, moving average)
- Learn how to determine the model order
- Address optimal model order selection criteria
- Apply stochastic modelling to real world data (speech, environmental, finance)

This material is a first fundamental step for the modelling of real world data

Linear systems



$$H(z) = \frac{Y(z)}{X(z)}$$

Described by their impulse response $h(n)$ or the transfer function $H(z)$

In the frequency domain (remember that $z = e^{\jmath\theta}$) the transfer function is

$$H(\theta) = \sum_{n=-\infty}^{\infty} h(n)e^{-\jmath n\theta} \qquad \{x[n]\} \rightarrow \left| \begin{array}{c} \{h(n)\} \\ H(\theta) \end{array} \right| \rightarrow \{y[n]\}$$

$$\textbf{that is} \qquad y[n] = \sum_{r=-\infty}^{\infty} h(r)x[n-r] = h * x$$

Then, the output PSD $S_{yy}(f) = H(f)H(-f)S_{xx}(f) = |H(f)|^2 S_{xx}(f)$.

Motivation: Wold decomposition theorem (also mentioned in your coursework)

The most fundamental justification for time series analysis is due to Wold's decomposition theorem, where it is explicitly proved that any (stationary) time series can be decomposed into two different parts: **deterministic** and **stochastic**.

Therefore, a general random process can be written a sum of two processes

$$x[n] = x_p[n] + x_r[n]$$

$\Rightarrow x_r[n] \quad \looparrowright \quad$ regular random process
$\Rightarrow x_p[n] \quad \looparrowright \quad$ predictable process, with $x_r[n] \quad \perp x_p[n]$,

$$E\{x_r[m]x_p[n]\} = 0$$

$\Rightarrow$ we can treat **separately** the **predictable** part (e.g. a deterministic signal) and the **random** signal.

Our focus will be on the modelling of the random component

Linear stochastic processes

It therefore follows that the general form for the power spectrum of a WSS process is

$$P_x(e^{\jmath\omega}) = P_{x_r}(e^{\jmath\omega}) + \sum_{k=1}^{N} \alpha_k \delta(\omega - \omega_k)$$

We are interested in processes generated by **filtering white noise with a linear shift–invariant filter** that has a rational system function. This class of digital filters includes the following system functions:

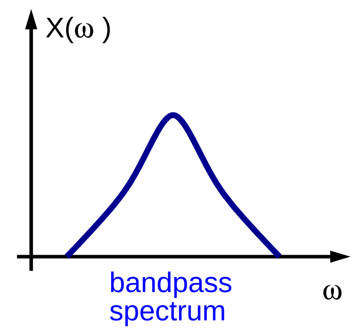
- Autoregressive (AR) $\rightarrow$ all pole system $\rightarrow$ $\quad H(z) = 1/A(z)$
- Moving Average (MA) $\rightarrow$ all zero system $\rightarrow$ $\quad H(z) = B(z)$
- Autoregressive Moving Average (ARMA) $\rightarrow$ poles and zeros $\rightarrow H(z) = B(z)/A(z)$

Notice the difference between shift–invariance and time–invariance

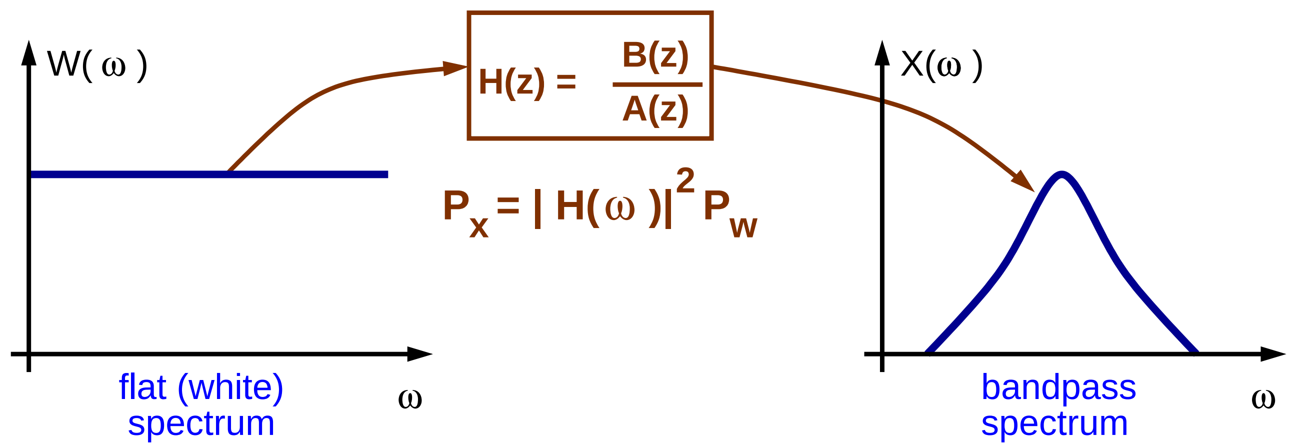
How do we model a real world signal?

Suppose the measured real world signal has e.g. a bandpass (any other) power spectrum

We desire to describe the whole long signal with very few parameters



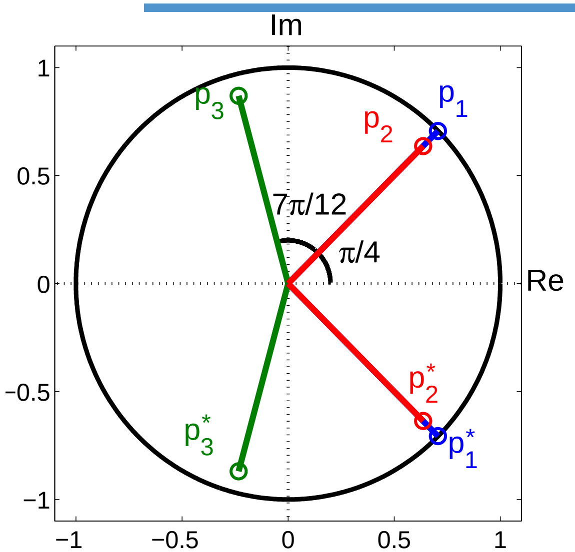
1. Can we model first and second statistics of real world signal by shaping the white noise spectrum using some transfer function?
2. Does this produce the same second order properties (mean, variance, ACF, spectrum) for any white noise input?

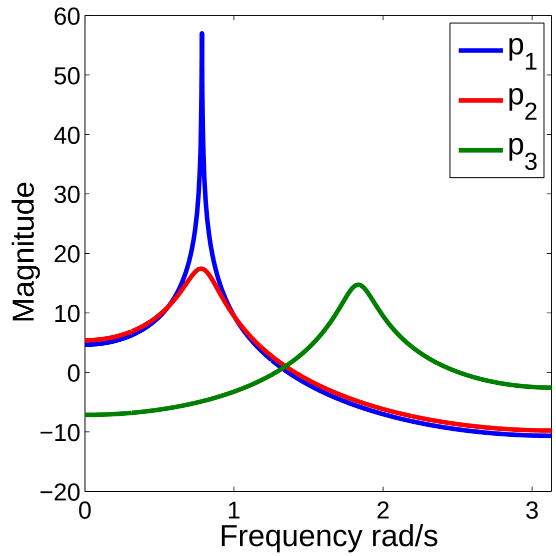


Can we use this linear stochastic model for prediction?

Example 1: Second-order all–pole transfer functions

$p_1 = 0.999exp(j\pi/4)$, $p_2 = 0.9exp(j\pi/4)$, $p_3 = 0.9exp(j7\pi/12)$





We have two conjugate complex poles, e.g. p_1 and p_1^*, therefore

$$H(z) = \frac{1}{(z-p_1)(z-p_1^*)}$$

$$= \frac{z^{-2}}{(1-p_1z^{-1})(1-p_1^*z^{-1})}$$

Transfer function for $p = \rho e^{j\theta}$ (ignoring z^{-2} in the numerator on the RHS):

$$H(z) = \frac{1}{(1-\rho e^{j\theta}z^{-1})(1-\rho e^{-j\theta}z^{-1})} = \frac{1}{1-2\rho\cos(\theta)z^{-1}+\rho^2z^{-2}}$$

$$\text{for the sinewave } \rho = 1 \quad \Rightarrow \quad H(z) = \frac{1}{1-2\cos(\theta)z^{-1}+z^{-2}} = \frac{1}{1+a_1z^{-1}+a_2z^{-2}}$$

$\Rightarrow$ **The sinewave can be modelled as an autoregressive process** (later)

Spectrum of ARMA models

recall that two conjugate complex poles of $A(z)$ give one peak in the spectrum

$ACF \quad \equiv \quad PSD$ **in terms of the information available**

In ARMA modelling we filter white noise $w[n]$ **(so called driving input)** with a causal linear shift–invariant filter with the transfer function $H(z)$, a rational system function with p poles and q zeros given by

$$X(z) = H(z)W(z) \quad \looparrowright \quad H(z) = \frac{B_q(z)}{A_p(z)} = \frac{\sum_{k=0}^{q} b_k z^{-k}}{1 + \sum_{k=1}^{p} a_k z^{-k}}$$

For a stable $H(z)$, the **ARMA(p,q) stochastic process** $x[n]$ will be wide–sense stationary. For the **driving noise** power $P_w = \sigma_w^2$, the power of the stochastic process $x[n]$ is **(recall** $P_y = |H(z)|^2 P_x = H(z)H^*(z)P_x$**)**

$$P_x(z) = \sigma_w^2 \frac{B_q(z)B_q(z^{-1})}{A_p(z)A_p(z^{-1})} \quad \Rightarrow \quad P_z(e^{\jmath\theta}) = \sigma_w^2 \frac{\left|B_q(e^{\jmath\theta})\right|^2}{\left|A_p(e^{\jmath\theta})\right|^2} = \sigma_w^2 \frac{\left|B_q(\omega)\right|^2}{\left|A_p(\omega)\right|^2}$$

Notice that "$(\cdot)^*$" in analogue frequency corresponds to "z^{-1}" in "digital freq."

Difference equation representation $\looparrowright$ the ACF follows the data model!

Random processes $x[n]$ and $w[n]$ are related by a linear difference equation with constant coefficients, given by

$$H(z) = \frac{X(z)}{W(z)} = \frac{B(z)}{A(z)} \quad \leftrightarrow \text{ARMA(p,q)} \leftrightarrow \quad x[n] = \underbrace{\sum_{l=1}^{p} a_l x[n-l]}_{autoregressive} + \underbrace{\sum_{l=0}^{q} b_l w[n-l]}_{moving\ average}$$

Notice that the autocorrelation function of $x[n]$ and crosscorrelation between the **stochastic process** $x[n]$ and **the driving input** $w[n]$ follow the same difference equation, i.e. if we multiply both sides of the above equation by $x[n-k]$ and take the statistical expectation, we have

$$r_{xx}(k) = \underbrace{\sum_{l=1}^{p} a_l\, r_{xx}(k-l)}_{\text{easy to calculate}} \; + \; \underbrace{\sum_{l=0}^{q} b_l\, r_{xw}(k-l)}_{\text{can be complicated}}$$

Since x is WSS, it follows that $x[n]$ and $w[n]$ are jointly WSS

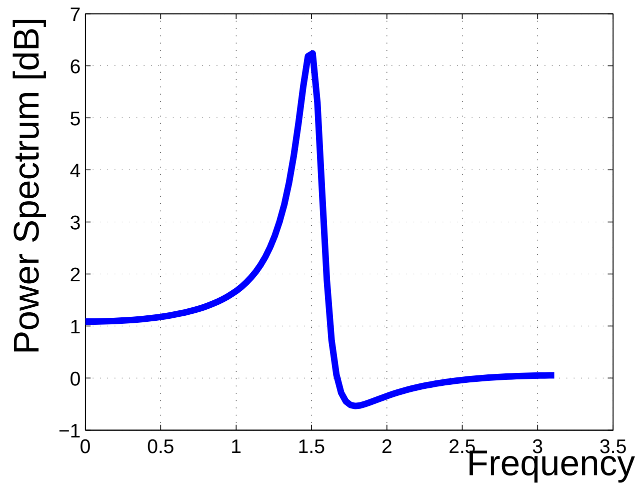
Example 2: Can the shape of power spectrum tell us about the order of the polynomials $B(z)$ and $A(z)$?

Plot the power spectrum of an ARMA(2,2) process for which

- the zeros of $H(z)$ are $z = 0.95e^{\pm \jmath\pi/2}$
- poles are at $z = 0.9e^{\pm \jmath 2\pi/5}$

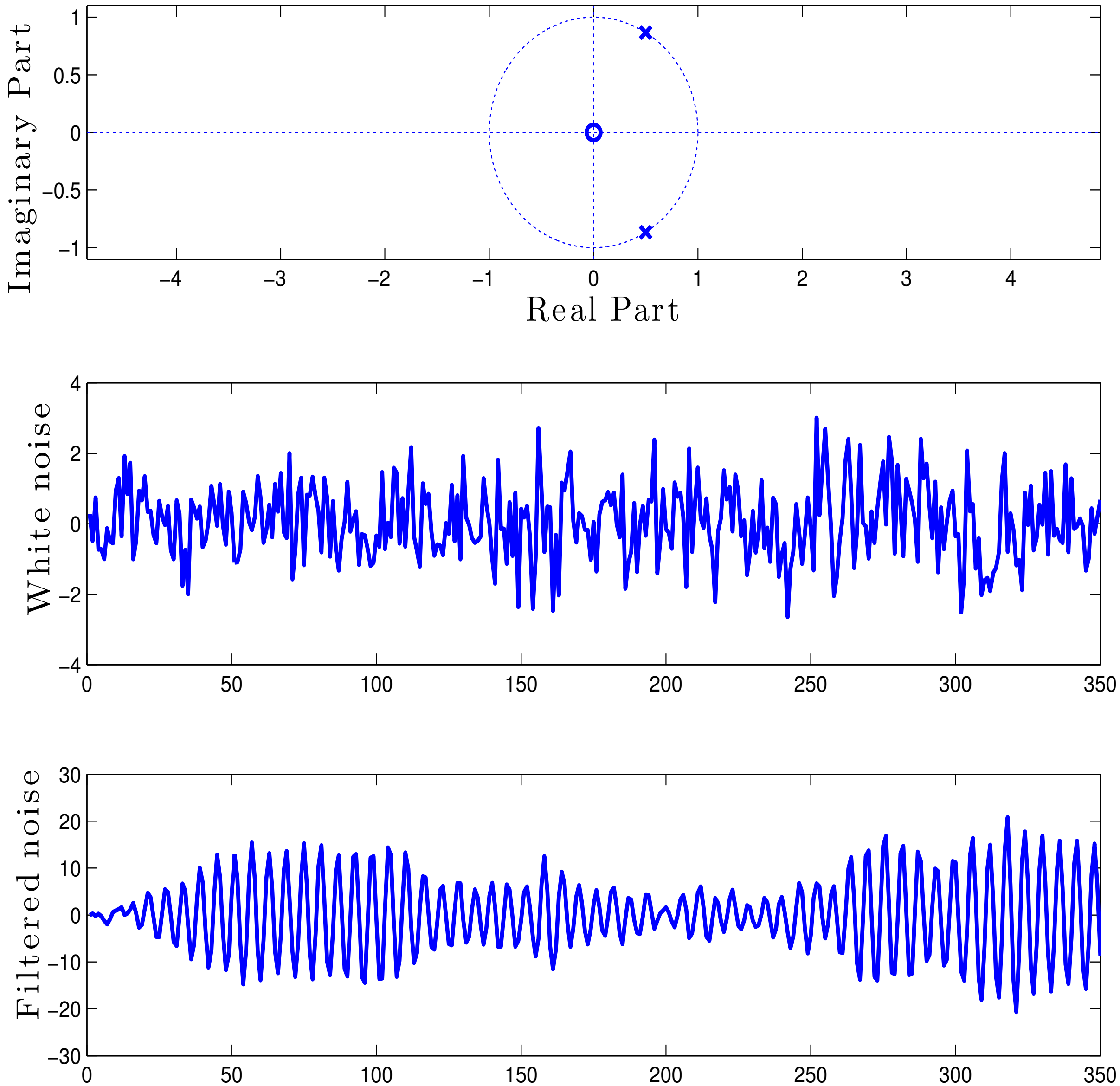
Solution: The system function is (poles and zeros – resonance & sink)

$$H(z) = \frac{1 + 0.9025z^{-2}}{1 - 0.5562z^{-1} + 0.81z^{-2}}$$



Example 3: How about a sinewave?

Is it nonlinear deterministic or linear stochastic?



Matlab code:

```
z1=0;
p1=[0.5+0.866i,0.5-0.866i];
[num1,den1]=zp2tf(z1,p1,1);
zplane(num1,den1);
s=randn(1,1000);
s1=filter(num1,den1,s);
figure;
subplot(311),plot(s),
subplot(313),plot(s1),
subplot(312),;
zplane(num1,den1)
```

The AR model of a sinewave

```
x(k)=a1*x(k-1)+a2*x(k-2)+w(k)
a1=-1, a2=0.98, w~N(0,1)
```

ACF and normalised ACF of AR processes

To obtain the autocorrelation function of an AR process, multiply the above equation by $x[n-k]$ to obtain

$$x[n-k]x[n] = a_1 x[n-k]x[n-1] + a_2 x[n-k]x[n-2] + \cdots$$
$$+ a_p x[n-k]x[n-p] + x[n-k]w[n]$$

Notice that $E\{x[n-k]w[n]\}$ vanishes when $k > 0$. Therefore we have

$$\begin{aligned} r_{xx}(0) &= a_1 r_{xx}(1) + a_2 r_{xx}(2) + \cdots + a_p r_{xx}(p) + \sigma_w^2, \qquad k = 0 \\ r_{xx}(k) &= a_1 r_{xx}(k-1) + a_2 r_{xx}(k-2) + \cdots + a_p r_{xx}(k-p), \qquad k > 0 \end{aligned}$$

On dividing throughout by $r_{xx}(0)$ we obtain

$$\rho(k) = a_1 \rho(k-1) + a_2 \rho(k-2) + \cdots + a_p \rho(k-p) \quad k > 0$$

Quantities $\rho(k)$ are called **normalised correlation coefficients**

Variance and spectrum of AR processes

Variance:
When $k = 0$ the contribution from the term $E\{x[n-k]w[n]\}$ is σ_w^2, and

$$r_{xx}(0) = a_1 r_{xx}(-1) + a_2 r_{xx}(-2) + \cdots + a_p r_{xx}(-p) + \sigma_w^2$$

Divide by $r_{xx}(0) = \sigma_x^2$ to obtain

$$\sigma_x^2 = \frac{\sigma_w^2}{1 - \rho_1 a_1 - \rho_2 a_2 - \cdots - \rho_p a_p}$$

Power spectrum: (recall that $P_{xx} = |H(z)|^2 P_{ww} = H(z)H^*(z)P_{ww}$, the expression for the output power of a linear system $\rightarrow$ see Slides 4 – 6)

$$P_{xx}(f) = \frac{2\sigma_w^2}{\left|1 - a_1 e^{-\jmath 2\pi f} - \cdots - a_p e^{-\jmath 2\pi p f}\right|^2} \qquad 0 \leq f \leq 1/2$$

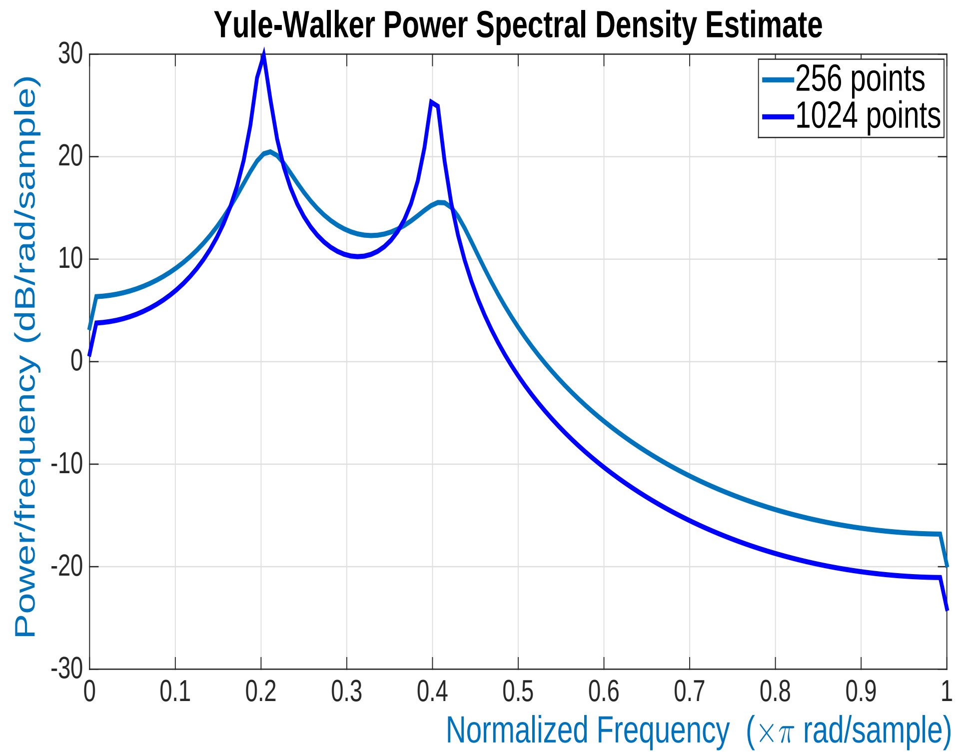
Look at *"Spectrum of Linear Systems"* from Lecture 1: Background

Example 4a: $AR(p)$ signal generation $\mathbf{a} = [2.2137, -2.9403, 2.1697, -0.9606]$

- Generate the input signal $\mathbf{x}$ by filtering white noise through the AR filter

- Estimate the PSD of $\mathbf{x}$ based on a fourth-order AR model

- **Careful!** The Matlab routines require the AR coeff. $\mathbf{a}$ in the format $\mathbf{a} = [1, -a_1, \ldots, -a_p]$



Notice the dependence on data length

Solution:

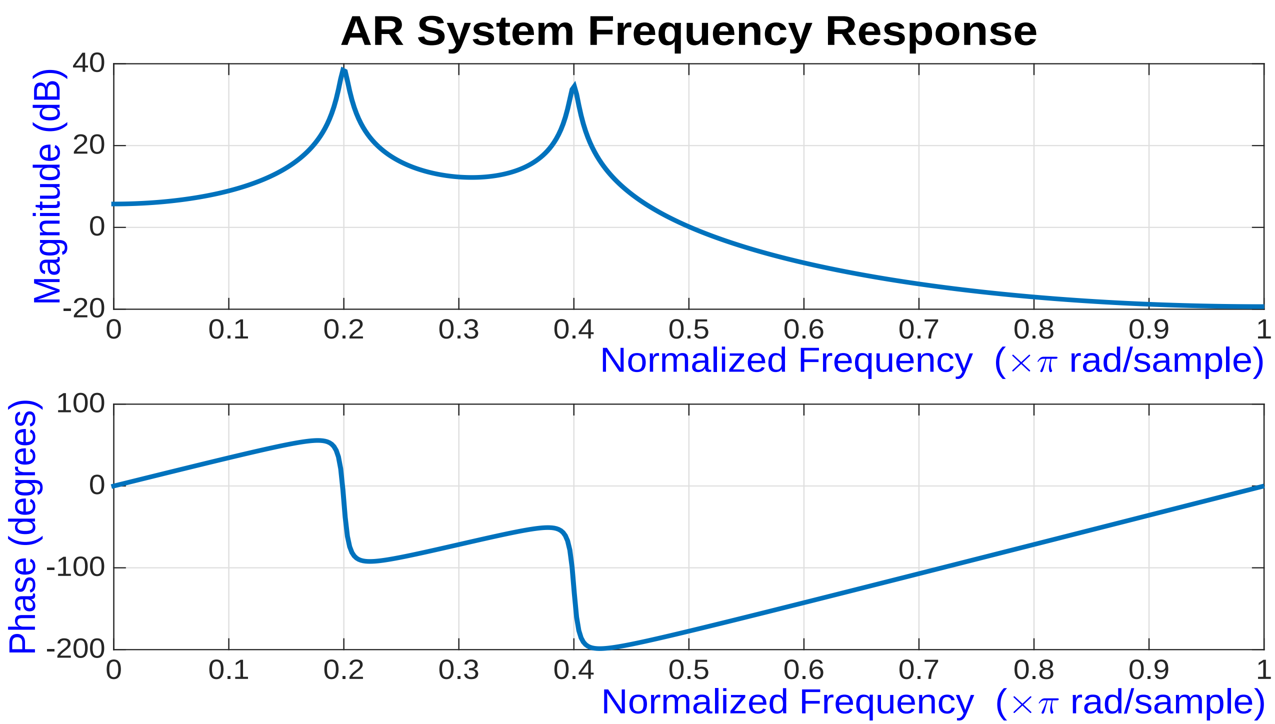
```
randn('state',1);
x = filter(1,a,randn(256,1));     % AR system output
pyulear(x,4)                      % Fourth-order estimate
```

Example 4b: Alternative AR power spectrum calculation

Consider the AR(4) system given by

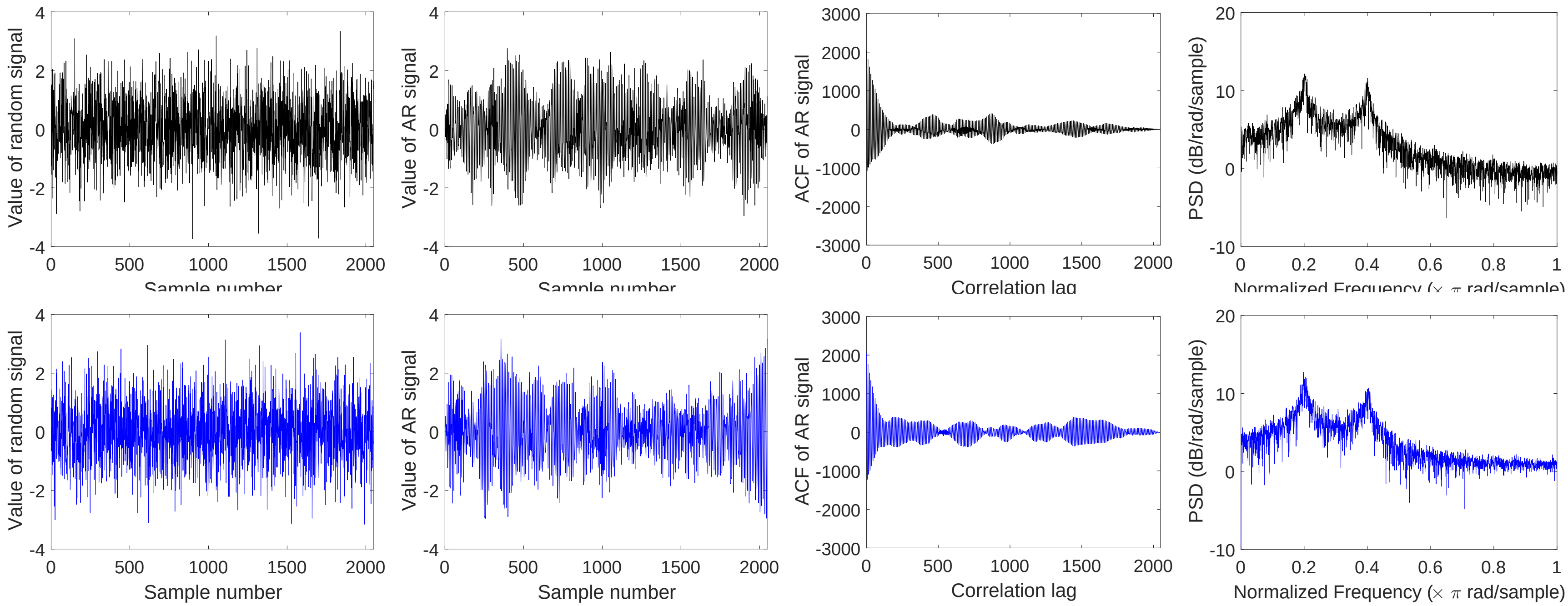
$$y[n] = 2.2137y[n-1] - 2.9403y[n-2] + 2.1697y[n-3] - 0.9606y[n-4] + w[n]$$

```
a = [1 -2.2137 2.9403 -2.1697 0.9606];  % AR filter coefficients
freqz(1,a)  % AR filter frequency response
title('AR System Frequency Response')
```



Example 5: Statistical properties of AR processes

Drive the AR(4) model from Example 4 with two different WGN realisations $\sim \mathcal{N}(0,1)$




```
r = wgn(2048,1,1);
a = [2.2137, -2.9403, 2.1697, -0.9606];
a = [1 -a];
x = filter(1,a,r);
xacf = xcorr(x);
xpsd = abs(fftshift(fft(xacf)));
```

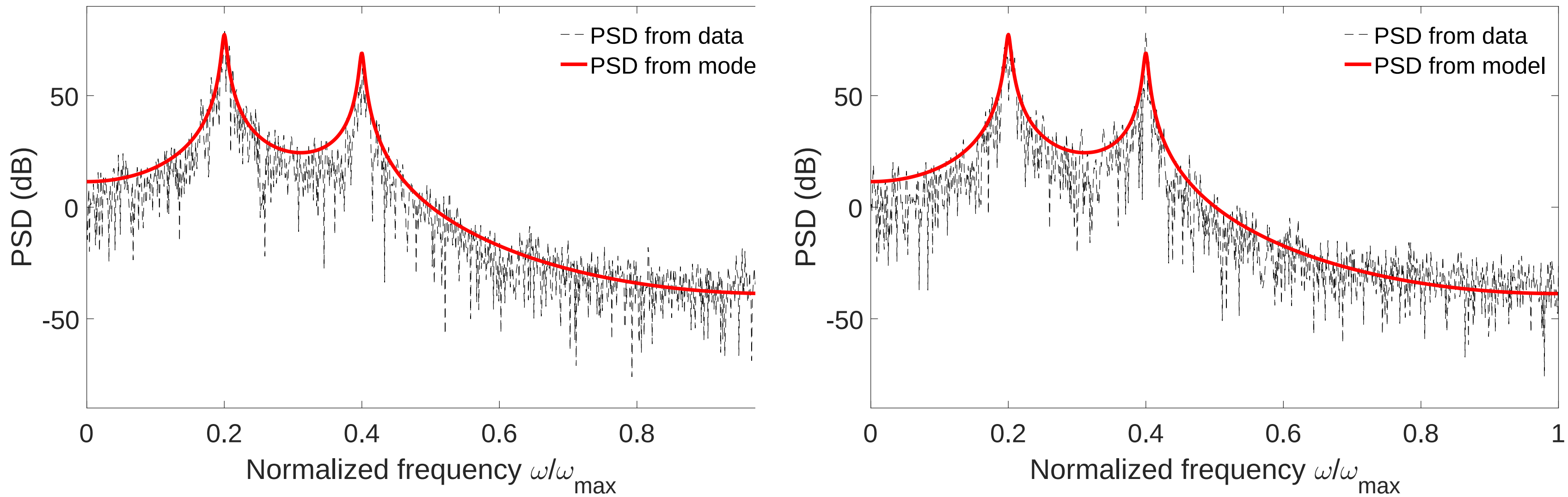
○ The time domain random AR(4) processes look different

○ The ACFs and PSDs are exactly the same (2nd-order stats)!

○ **This signifies the importance of our statistical approach**

Example 5b: Advantages of model-based analysis

Consider the PSD's for different realisations of the AR(4) process from Example 5



- The different realisations lead to different Emprical PSD's (in thin black)
- The theoretical PSD from the model is consistent regardless of the data (in thick red)

```
N = 1024;
w = wgn(N,1,1);
a = [2.2137, -2.9403, 2.1697, -0.9606];       % Coefficients of AR(4) process
a = [1 -a];
x = filter(1,a,w);
xacf = xcorr(x);                              % Autocorrelation of AR(4) process
dft = fft(xacf);
EmpPSD = abs(dft/length(dft)).^ 2;            % Empirical PSD obtained from data
ThePSD = abs(freqz(1,a,N,1)).^ 2 ;            % Theoretical PSD obtained from model
```

Key: Finding AR coefficients $\looparrowright$ the Yule–Walker eqns

(there are several similar forms – we follow the most concise one)

For $k = 1, 2, \ldots, p$ from the general autocorrelation function, we obtain a set of equations:

$$\begin{aligned} r_{xx}(1) &= a_1 r_{xx}(0) + a_2 r_{xx}(1) + \cdots + a_p r_{xx}(p-1) \\ r_{xx}(2) &= a_1 r_{xx}(1) + a_2 r_{xx}(0) + \cdots + a_p r_{xx}(p-2) \\ \vdots &= \vdots \\ r_{xx}(p) &= a_1 r_{xx}(p-1) + a_2 r_{xx}(p-2) + \cdots + a_p r_{xx}(0) \end{aligned}$$

These equations are called the **Yule–Walker or normal equations.**

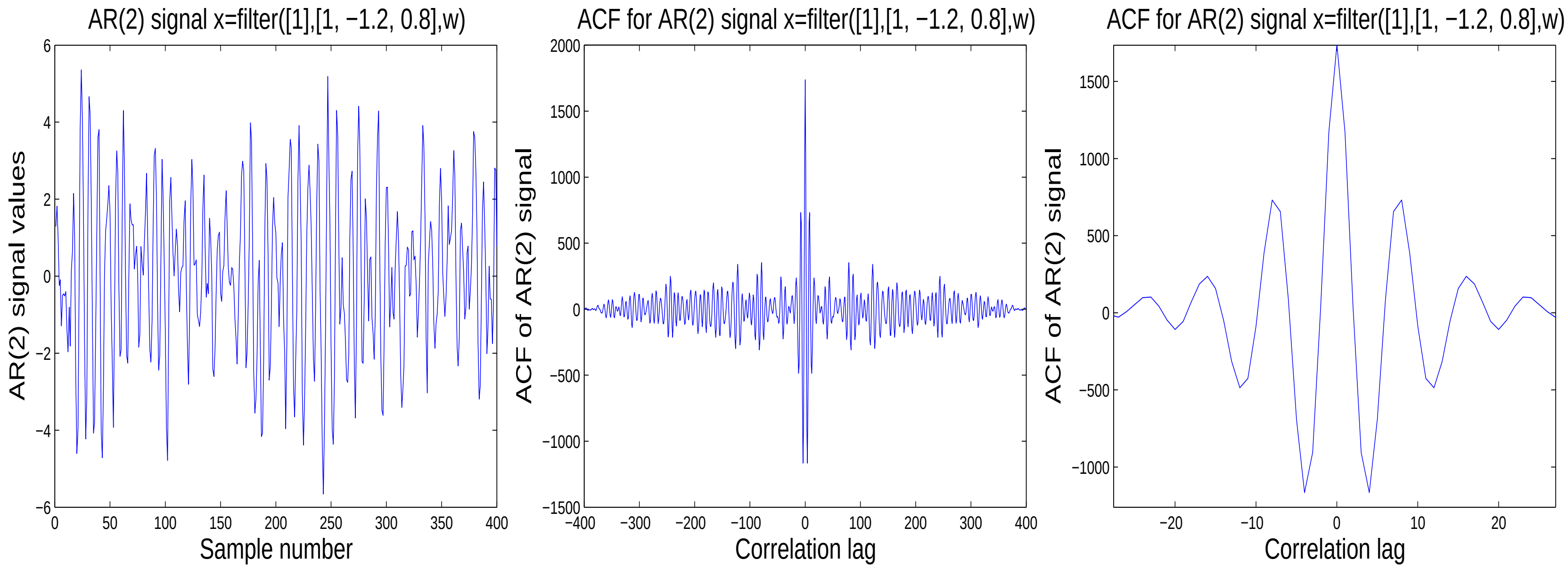
Their solution gives us the set of **autoregressive parameters** $\mathbf{a} = [a_1, \ldots, a_p]^T$.
The above can be expressed in a vector–matrix form as

$$\mathbf{r}_{xx} = \mathbf{R}_{xx}\mathbf{a} \quad \Rightarrow \quad \mathbf{a} = \mathbf{R}_{xx}^{-1}\mathbf{r}_{xx}$$

The ACF matrix $\mathbf{R}_{xx}$ is positive definite (Toeplitz) which guarantees matrix inversion

Example 6: Find the parameters of an AR(2) process, $x(n)$, generated by $x[n] = 1.2x[n-1] - 0.8x[n-2] + w[n]$

Coursework: comment on the shape of the ACF for large lags



Matlab: `for i=1:6; [a,e]=aryule(x,i); display(a);end`

$\mathbf{a}^{(1)} = [0.6689]$ $\qquad$ $\mathbf{a}^{(2)} = [1.2046, -0.8008]$

$\mathbf{a}^{(3)} = [1.1759, -0.7576, -0.0358]$

$\mathbf{a}^{(4)} = [1.1762, -0.7513, -0.0456, 0.0083]$

$\mathbf{a}^{(5)} = [1.1763, -0.7520, -0.0562, 0.0248, -0.0140]$

$\mathbf{a}^{(6)} = [1.1762, -0.7518, -0.0565, 0.0198, -0.0062, -0.0067]$

Example 7: Yule–Walker modelling in Matlab

In Matlab – Power spectral density using Y–W method *pyulear*

```
Pxx = pyulear(x,p)
[Pxx,w] = pyulear(x,p,nfft)
[Pxx,f] = pyulear(x,p,nfft,fs)
[Pxx,f] = pyulear(x,p,nfft,fs,'range')
[Pxx,w] = pyulear(x,p,nfft,'range')
```

Description:

```
Pxx = pyulear(x,p)
```

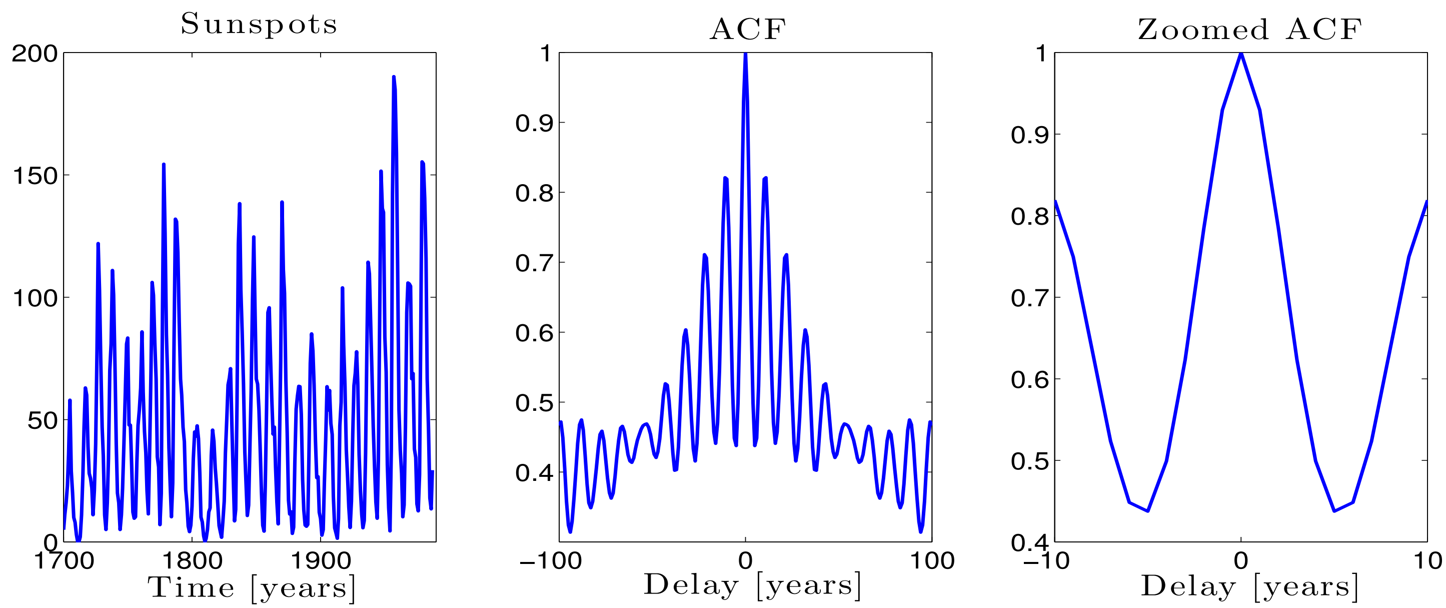
implements the Yule-Walker algorithm, and returns Pxx, an estimate of the power spectral density (PSD) of the vector $\mathbf{x}$.

To remember for later $\rightarrow$ This estimate is also an estimate of the maximum entropy.

See also **aryule, lpc, pburg, pcov, peig, periodogram**

Example 8: Sunspot number estimation

consistent with the properties of a second order AR process



$$\mathbf{a}_1 = [0.9295] \qquad \mathbf{a}_2 = [1.4740, -0.5857]$$
$$\mathbf{a}_3 = [1.5492, -0.7750, 0.1284]$$
$$\mathbf{a}_4 = [1.5167, -0.5788, -0.2638, 0.2532]$$
$$\mathbf{a}_5 = [1.4773, -0.5377, -0.1739, 0.0174, 0.1555]$$
$$\mathbf{a}_6 = [1.4373, -0.5422, -0.1291, 0.1558, -0.2248, 0.2574]$$

↬ The sunspots model is $\quad x[n] = 1.474\, x[n-1] - 0.5857\, x[n-2] + w[n]$

Special case #2: Second order autoregressive processes, $p = 2$, $q = 0$, hence the notation AR(2)

The input–output functional relationship is given by ($w[n] \sim$ any white noise)

$$\begin{aligned} x[n] &= a_1 x[n-1] + a_2 x[n-2] + w[n] \\ X(z) &= \left(a_1 z^{-1} + a_2 z^{-2}\right) X(z) + W(z) \\ \Rightarrow \quad H(z) &= \frac{X(z)}{W(z)} = \frac{1}{1 - a_1 z^{-1} - a_2 z^{-2}} \\ H(\omega) = H(e^{\jmath\omega}) &= \frac{1}{1 - a_1 e^{-\jmath\omega} - a_2 e^{-2\jmath\omega}} \end{aligned}$$

Y-W equations for p=2

$$\rho_1 = a_1 + a_2 \rho_1$$

$$\rho_2 = a_1 \rho_1 + a_2$$

when solved for a_1 and a_2, we have

$$a_1 = \frac{\rho_1(1-\rho_2)}{1-\rho_1^2} \qquad a_2 = \frac{\rho_2 - \rho_1^2}{1-\rho_1^2}$$

Connecting a's and ρ's

$$\rho_1 = \frac{a_1}{1-a_2}$$

$$\rho_2 = a_2 + \frac{a_1^2}{1-a_2}$$

Since $\rho_1 < \rho(0) = 1 \hookrightarrow$ a **stability condition** on a_1 and a_2

Variance and power spectrum

Both readily obtained from the general AR(2) process equation!

Variance

$$\sigma_x^2 = \frac{\sigma_w^2}{1 - \rho_1 a_1 - \rho_2 a_2} = \left(\frac{1 - a_2}{1 + a_2}\right) \frac{\sigma_w^2}{(1 - a_2)^2 - a_1^2}$$

Power spectrum

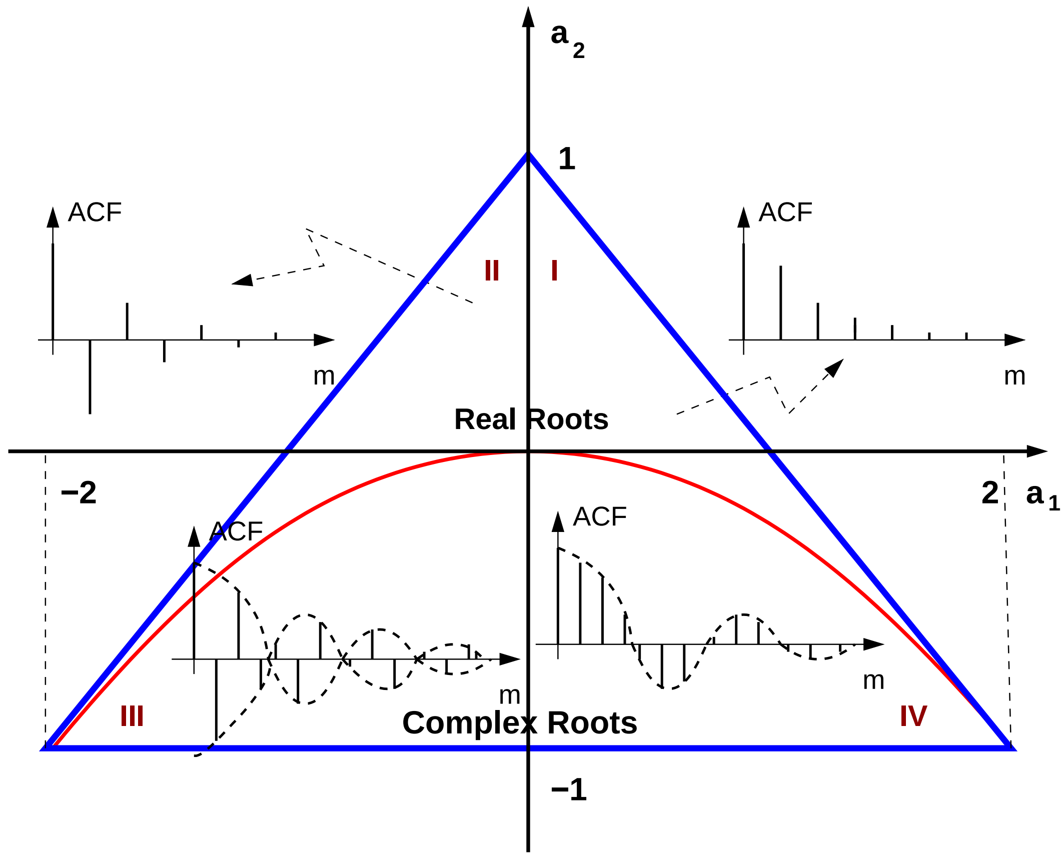
$$\begin{aligned} P_{xx}(f) &= \frac{2\sigma_w^2}{\left|1 - a_1 e^{-\jmath 2\pi f} - a_2 e^{-\jmath 4\pi f}\right|^2} \\ &= \frac{2\sigma_w^2}{1 + a_1^2 + a_2^2 - 2a_1(1 - a_2\cos(2\pi f) - 2a_2\cos(4\pi f))}, \quad 0 \le f \le 1/2 \end{aligned}$$

Stability conditions $\rightsquigarrow$ (Condition 1 can be obtained from the denominator of variance, Condition 2 from the expression for ρ_1, etc.)

$$\begin{aligned} \text{Condition } 1: &\quad a_1 + a_2 < 1 \\ \text{Condition } 2: &\quad a_2 - a_1 < 1 \\ \text{Condition } 3: &\quad -1 < a_2 < 1 \end{aligned}$$

This can be visualised within the so–called "stability triangle"

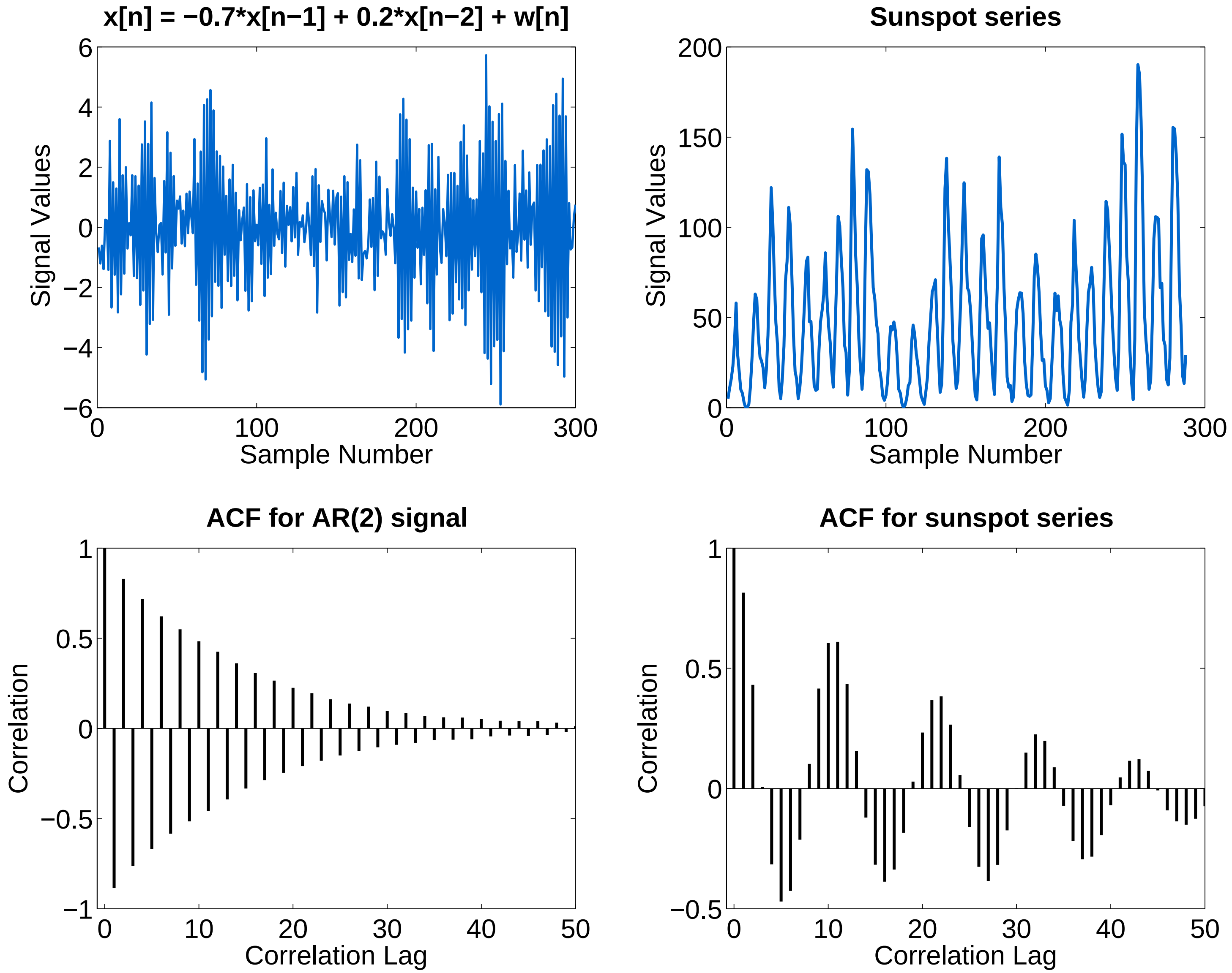
Stability triangle



i) **Real roots** Region 1: Monotonically decaying ACF
ii) **Real roots** Region 2: Decaying oscillating ACF
iii) **Complex roots** Region 3: Oscillating pseudo-periodic ACF
iv) **Complex roots** Region 4: Pseudo-periodic ACF

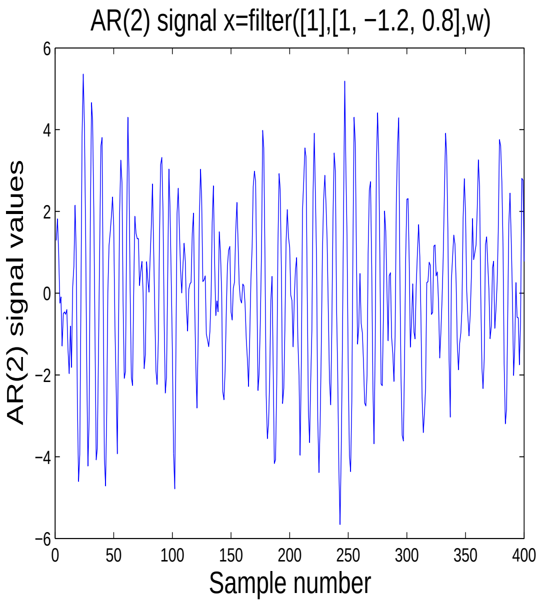
Example 10: Stability triangle and ACFs of AR(2) signals

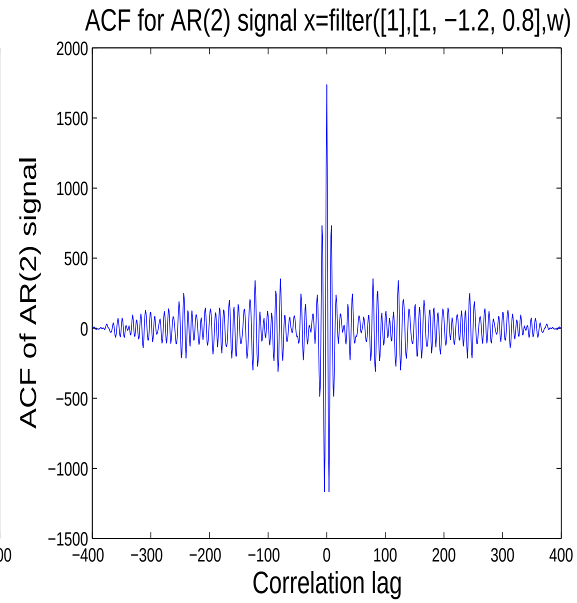
Left: $\mathbf{a} = [-0.7, 0.2]$ **(region 2)** **Right:** $\mathbf{a} = [1.474, -0.586]$ **(region 4)**

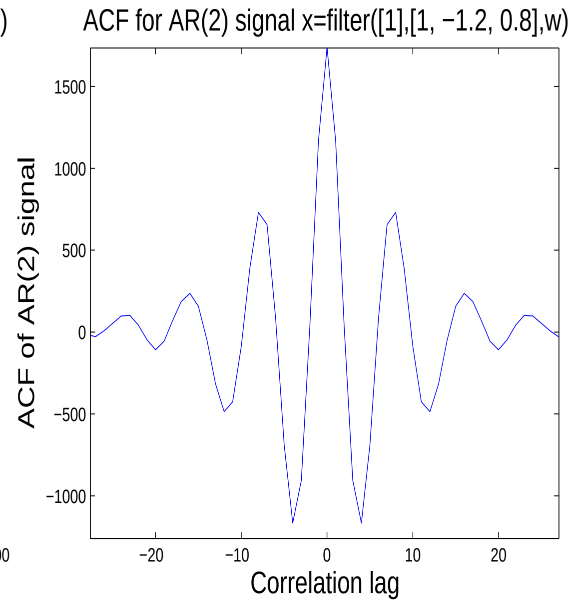


Model order selection: Partial autocorrelation function

Consider an earlier example using a slightly different notation for AR coefficients







`To find p, first re-write AR coeffs. of order p as [a_p1,...,a_pp]`

$$\mathbf{p = 1} \looparrowright [0.6689] \quad = \quad a_{11} \qquad \mathbf{p = 2} \looparrowright [1.2046, -0.8008] = [a_{21}, a_{22}]$$

$$\mathbf{p = 3} \looparrowright [1.1759, -0.7576, -0.0358] \quad = \quad [a_{31}, a_{32}, a_{33}]$$

$$\mathbf{p = 4} \looparrowright [1.1762, -0.7513, -0.0456, 0.0083] \quad = \quad [a_{41}, a_{42}, a_{43}, a_{44}]$$

$$\mathbf{p = 5} \looparrowright [1.1763, -0.7520, -0.0562, 0.0248, -0.0140] = [a_{51}, \ldots, a_{55}]$$

$$\mathbf{p = 6} \looparrowright [1.1762, -0.7518, -0.0565, 0.0198, -0.0062, -0.0067]$$

Partial autocorrelation function: Motivation

Notice: ACF of $AR(p)$ infinite in duration, **but** can by be described in terms of p nonzero functions ACFs.

Denote by a_{kj} the jth coefficient in an autoregressive representation of order k, so that a_{kk} is the last coefficient. Then

$$\rho_j = a_{kj}\rho_{j-1} + \cdots + a_{k(k-1)}\rho_{j-k+1} + a_{kk}\rho_{j-k} \qquad j = 1, 2, \ldots, k$$

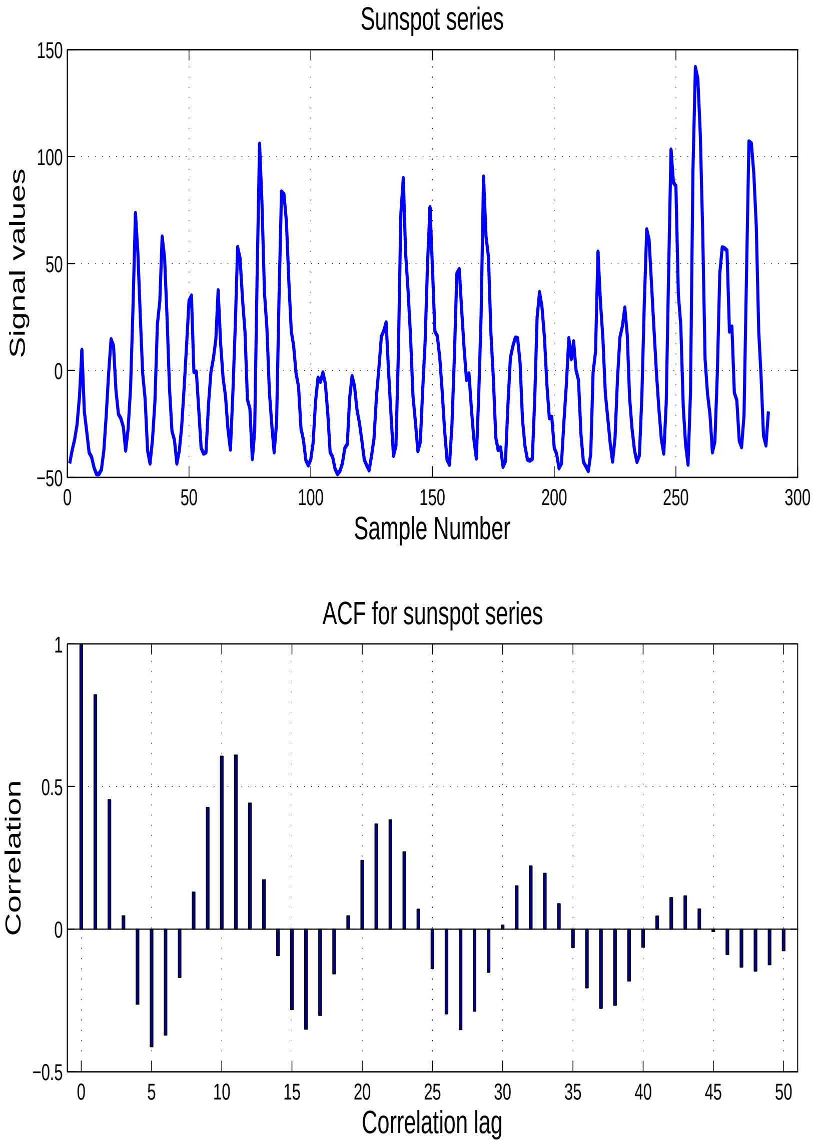
leading to the Yule–Walker equation, which can be written as

$$\begin{bmatrix} 1 & \rho_1 & \rho_2 & \cdots & \rho_{k-1} \\ \rho_1 & 1 & \rho_1 & \cdots & \rho_{k-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \rho_{k-3} & \cdots & 1 \end{bmatrix} \begin{bmatrix} a_{k1} \\ a_{k2} \\ \vdots \\ a_{kk} \end{bmatrix} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_k \end{bmatrix}$$

The only difference from the standard Y-W equtions is the use of the symbols a_{ki} to denote the AR coefficient a_i $\looparrowright$ indicating the model order.

Example 12: Work by Yule $\looparrowright$ model of sunspot numbers

Recorded for $>$ 300 years. To study them in 1927 Yule invented the $AR(2)$ model



We first center the data, as we do not wish to model the DC offset (determinisic component), but the stochastic component (AR model driven by white noise)!

Using the Y-W equations we obtain:

$\mathbf{a}_1 = [0.9295]$

$\mathbf{a}_2 = [1.4740, -0.5857]$

$\mathbf{a}_3 = [1.5492, -0.7750, 0.1284]$

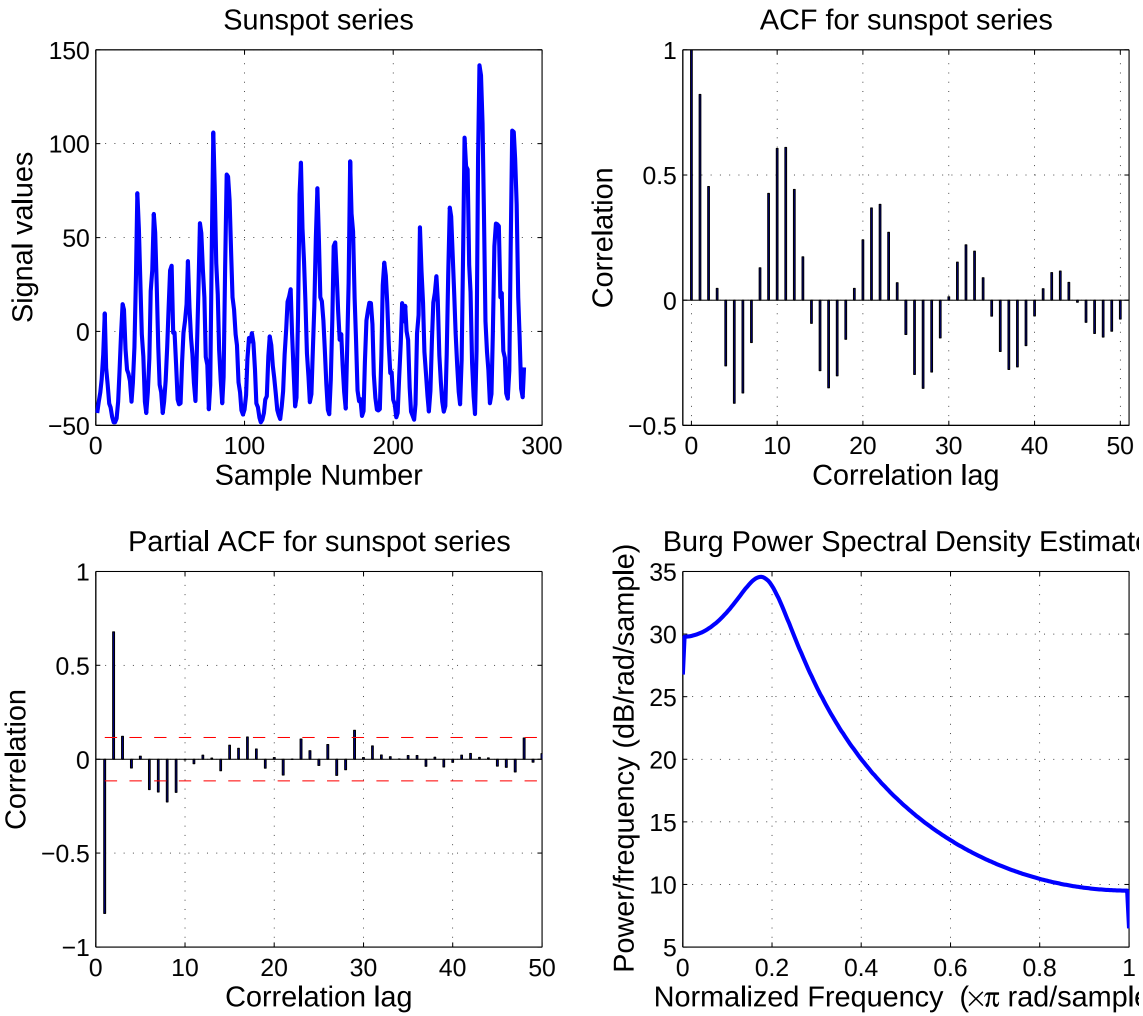
$\mathbf{a}_4$=[1.5167,-0.5788,-0.2638,0.2532]

$\mathbf{a}_5$=[1.4773,-0.5377,-0.1739, 0.0174,0.1555]

$\mathbf{a}_6$=[1.4373, -0.5422, -0.1291, 0.1558, -0.2248, 0.2574]

Example 12 (contd.): Model order for sunspot numbers

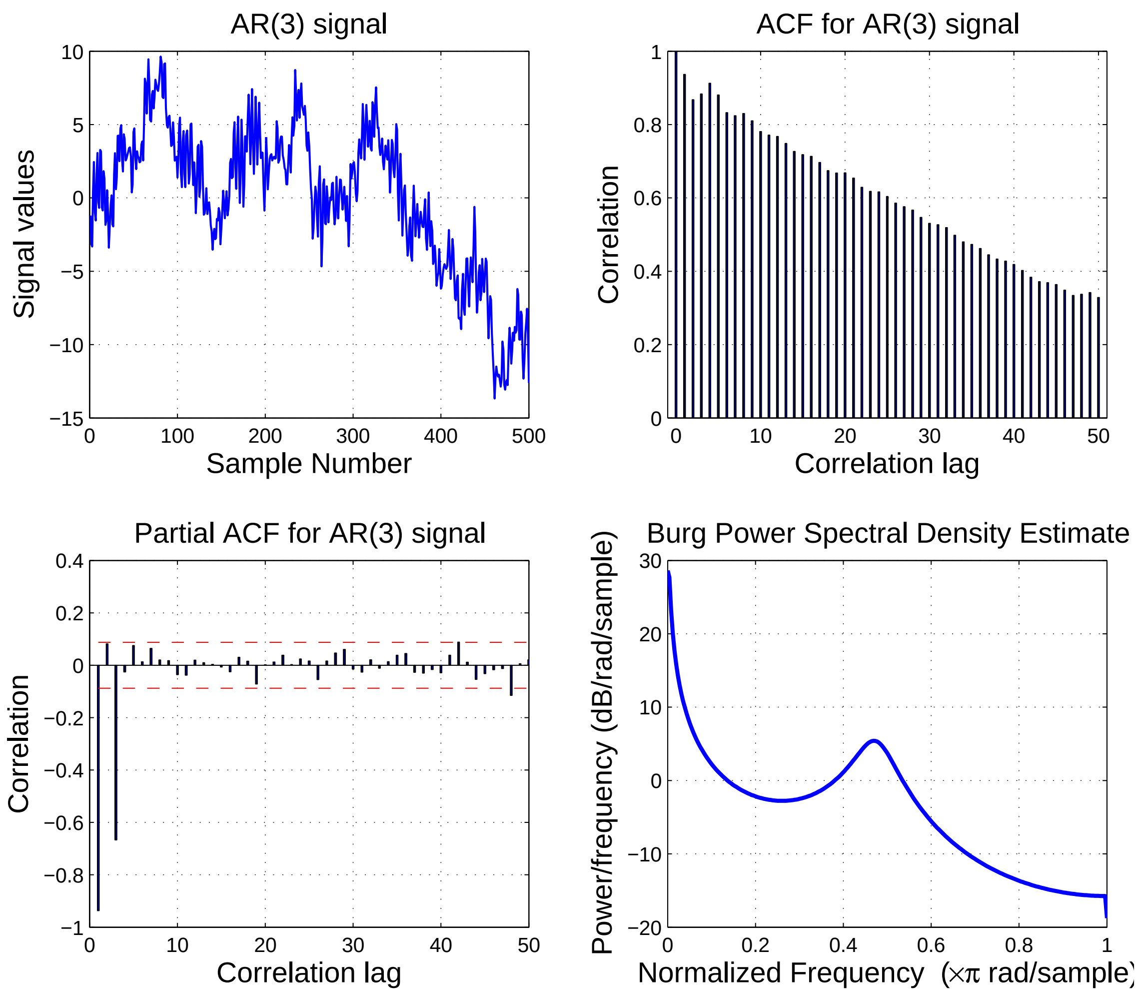
Sunspot numbers, their ACF and partial autocorrelation (PAC)



After lag $k = 2$, the PAC becomes very small

Example 13: Model order for an AR(3) process

AR(3) signal, its ACF, and partial autocorrelation (PAC)



After lag $k = 3$, the PAC becomes very small

AR model based prediction: Importance of model order

For a zero mean process $x[n]$, the best **linear predictor** in the **mean square error** sense of $x[n]$ based on $x[n-1], x[n-2], \ldots$ is

$$\hat{x}[n] = a_{k-1,1}x[n-1] + a_{k-1,2}x[n-2] + \cdots + a_{k-1,k-1}x[n-k+1]$$

(apply the $E\{\cdot\}$ operator to the general $AR(p)$ model expression, and recall that $E\{w[n]\} = 0$)

(Hint:
$E\{x[n]\} = \hat{x}[n] = E\left\{a_{k-1,1}x[n-1] + \cdots + a_{k-1,k-1}x[n-k+1] + w[n]\right\} = a_{k-1,1}x[n-1] + \cdots + a_{k-1,k-1}x[n-k+1])$)

whether the process is an AR or not

In MATLAB, check the function:

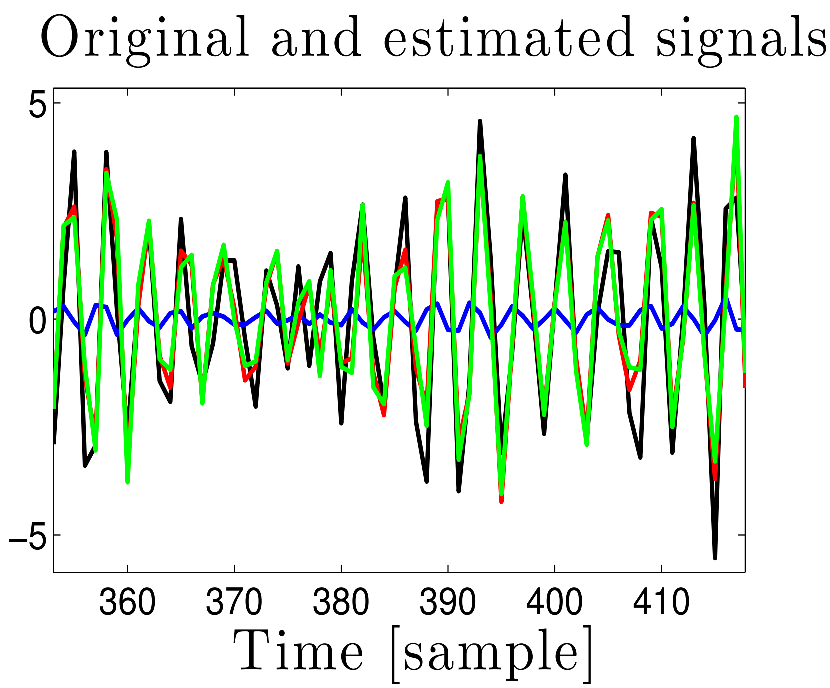
`ARYULE`

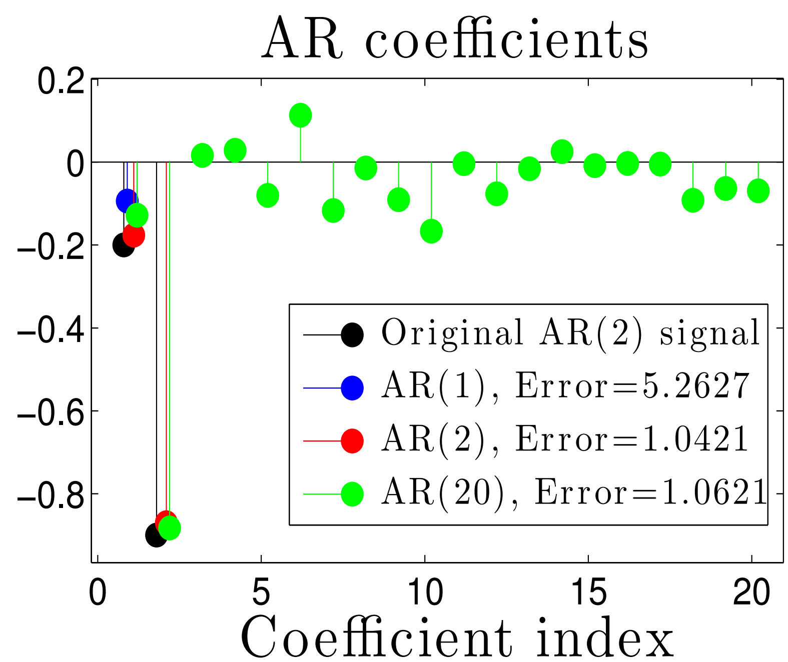
and functions

`PYULEAR, ARMCOV, ARBURG, ARCOV, LPC, PRONY`

Estimation of the parameters of an AR(2) process using models of different orders (under- vs. over-fitting)

Original AR(2) process $x[n] = -0.2x[n-1] - 0.9x[n-2] + w[n]$, $w[n] \sim \mathcal{N}(0,1)$, estimated using AR(1), AR(2) and AR(20) models:

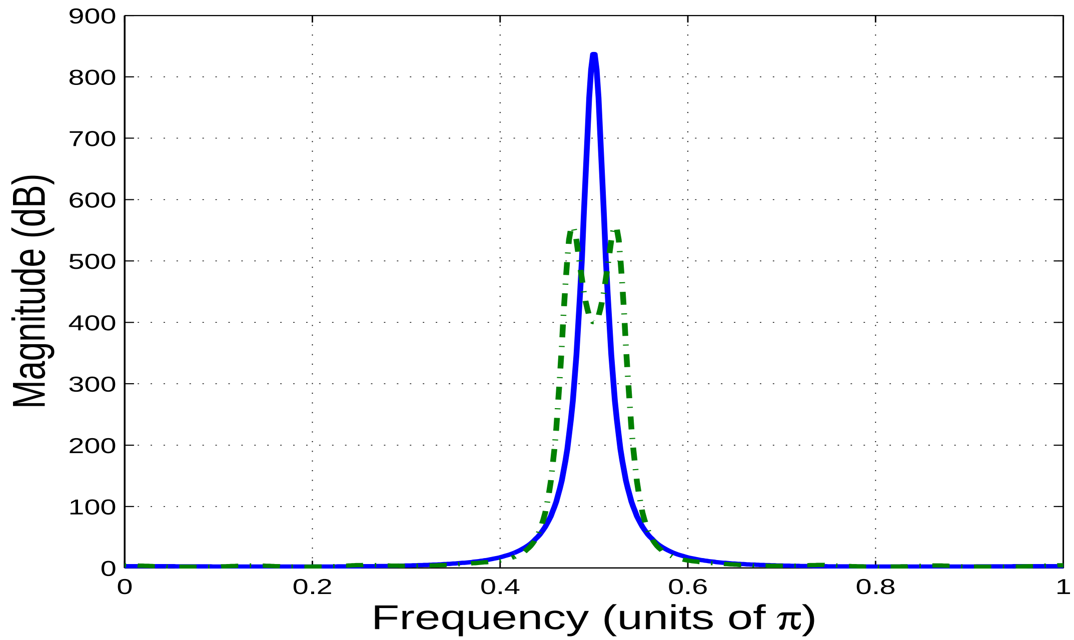




The *higher order* coefficients of the AR(20) model are close to zero and therefore do not contribute significantly to the estimate, while the AR(1) does not have sufficient degrees of freedom.

Effects of over-modelling on autoregressive spectral estimation: Spectral line splitting

Consider an AR(2) signal $x(n) = -0.9x(n-2) + w(n)$ with $w \sim \mathcal{N}(0,1)$. $N = 64$ data samples, model orders $p = 4$ (solid blue) and $p = 12$ (broken green).



Notice that this is an AR(2) model!

Although the true spectrum has a single spectral peak at $\omega - \pi/2$ (blue), when over-modelling using $p = 12$ this peak is split into two peaks (green).

Model order selection $\looparrowright$ practical issues

In practice: the greater the model order the greater accuracy & complexity

Q: When do we stop? What is the optimal model order?

Solution: To establish a trade–off between computational complexity and model accuracy, we introduce "penalty" for a high model order. Such criteria for model order selection are:

MDL: The minimum description length criterion (MDL) (by Rissanen),

AIC: The Akaike information criterion (AIC)

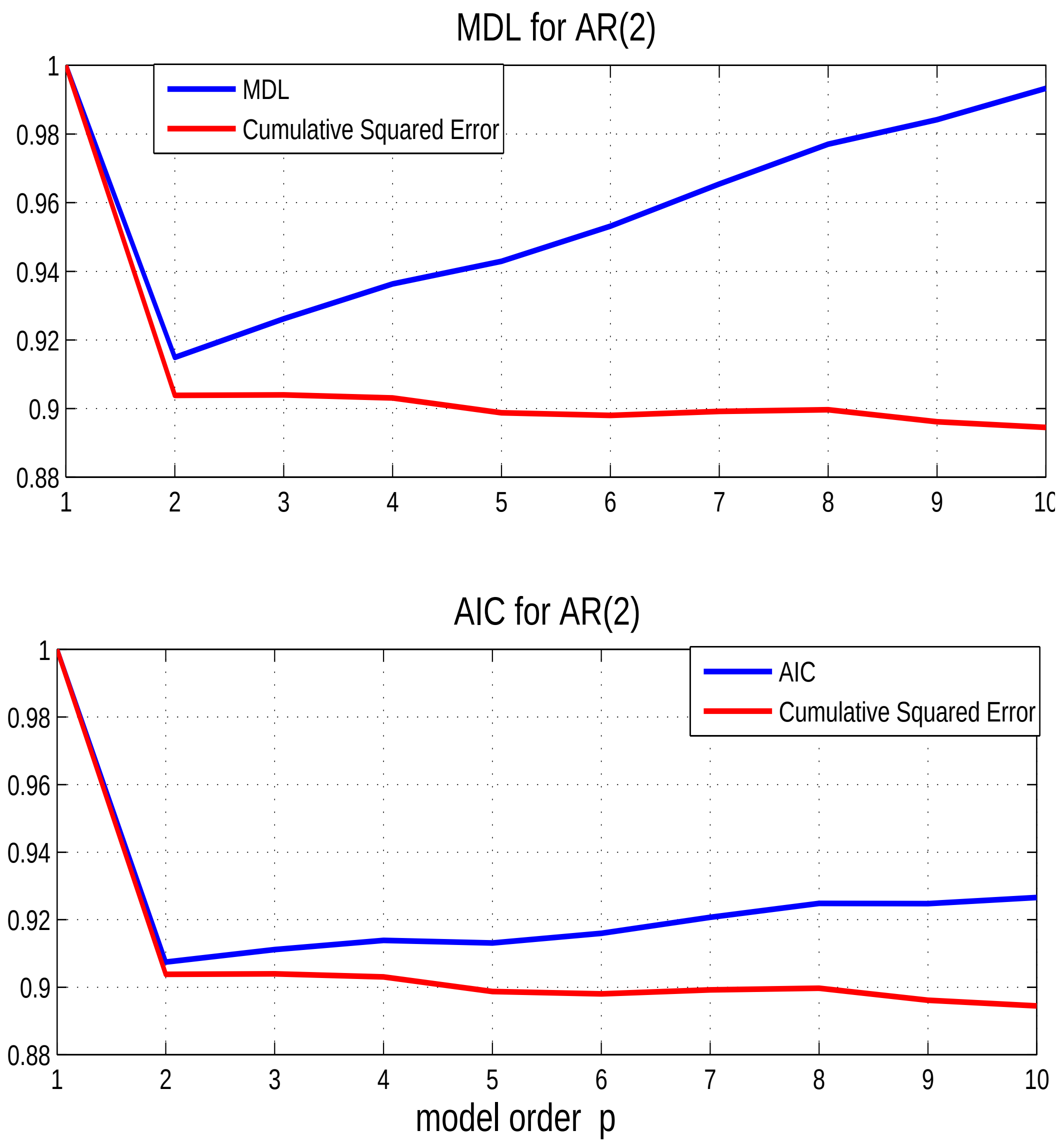
$$\mathbf{MDL} \quad p_{opt} = \min_p \left[log(E) + \frac{p * log(N)}{N} \right]$$

$$\mathbf{AIC} \quad p_{opt} = \min_p \left[log(E) + 2p/N \right]$$

$E \rightsquigarrow$ the loss function (typically cumulative squared error),
$p \rightsquigarrow$ the number of estimated parameters (model order),
$N \rightsquigarrow$ the number of available data points.

Example 15: Model order selection $\hookrightarrow$ MDL vs AIC

MDL and AIC criteria for an AR(2) model with $a_1 = 0.5 \qquad a_2 = -0.3$



The graphs on the left show the **(model error)**^{2} (vertical axis) versus the **model order** p (horizontal axis). Notice that $p_{opt} = 2$.

The curves are **convex**, i.e. a monotonically decreasing **error**^{2} with an increasing **penalty term** (MDL or AIC correction).

Hence, we have a unique minimum at $p = 2$, reflecting the correct model order (no over-modelling)

Moving average processes, MA(q)

A general MA(q) process is given by

$$x[n] = w[n] + b_1 w[n-1] + \cdots + b_q w[n-q]$$

Autocorrelation function: The autocovariance function of MA(q)

$$\begin{aligned} c_k = E[(w[n] + b_1 w[n-1] + \cdots + b_q w[n-q])(w[n-k] \\ + b_1 w[n-k-1] + \cdots + b_q w[n-k-q])] \end{aligned}$$

Hence the variance of the process

$$c_0 = (1 + b_1^2 + \cdots + b_q^2)\sigma_w^2$$

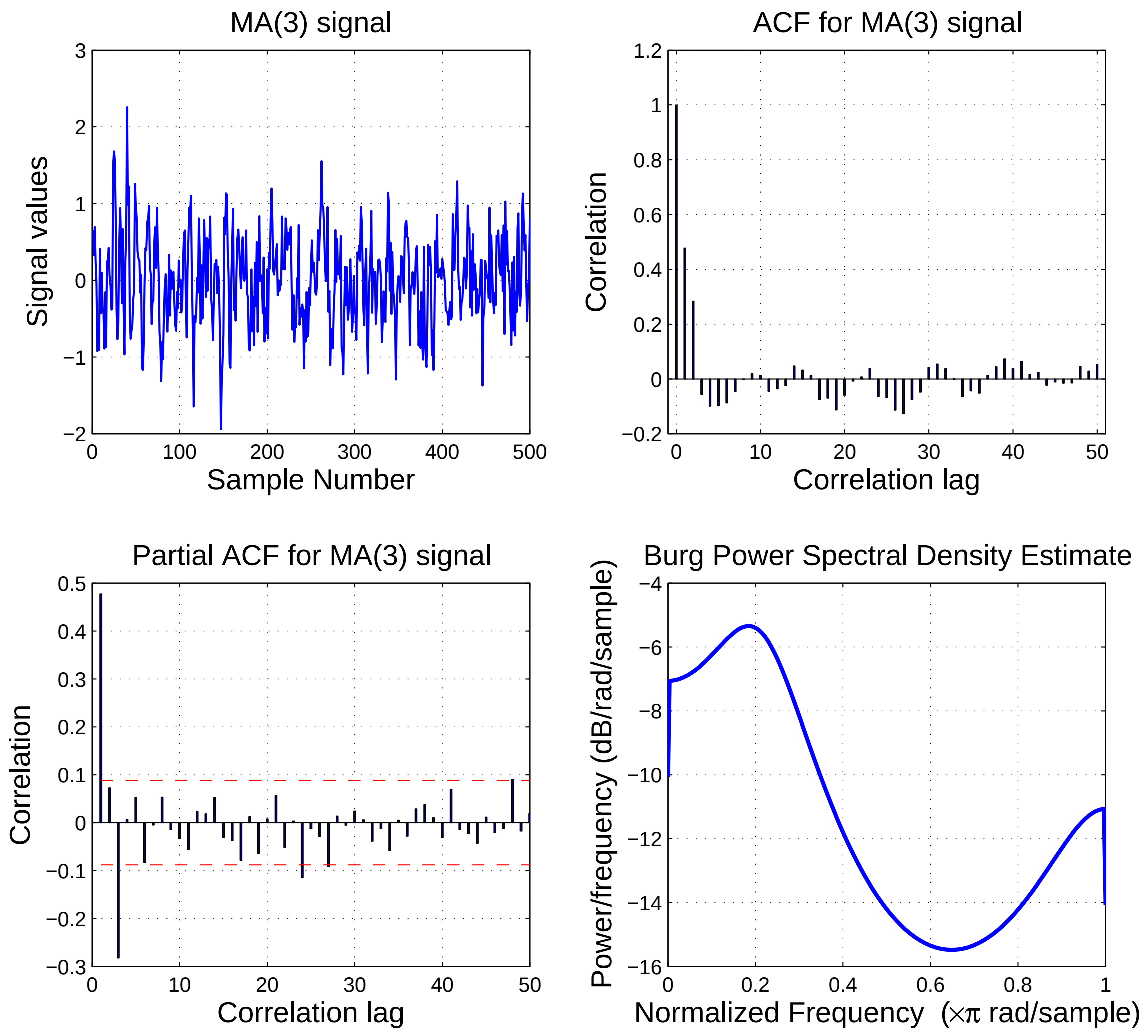
The ACF of an MA process has a cutoff after lag q.

Spectrum: All–zero transfer function $\Rightarrow$ struggles to model 'peaky' PSDs

$$P(f) = 2\sigma_w^2 \left|1 + b_1 e^{-\jmath 2\pi f} + b_2 e^{-\jmath 4\pi f} + \cdots + b_q e^{-\jmath 2\pi q f}\right|^2$$

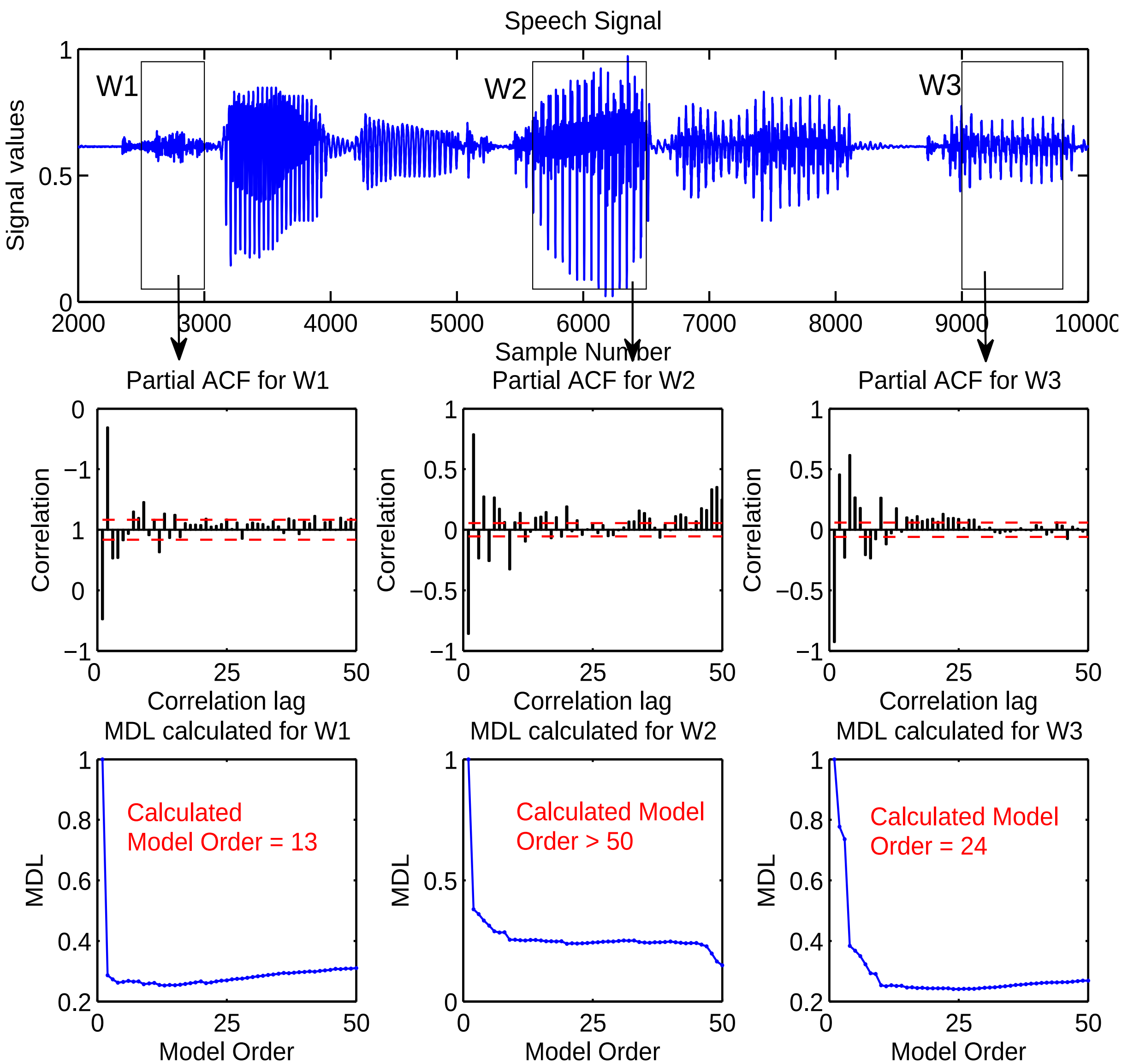
Example 16: Third order moving average MA(3) process

An MA(3) process and its autocorrel. (ACF) and partial autocorrel. (PAC) fns.



After lag $k = 3$, the ACF becomes very small

Analysis of nonstationary signals



- Consider a real–world speech signal, and thee different segments with different statistical properties
- **Different AR model orders required for different segments of speech ↬ opportunity for content analysis!**
- To deal with nonstationarity we need short sliding data windows

Summary: AR and MA Processes

i) A stationary finite AR(p) process can be represented as an infinite order MA process. A finite MA process can be represented as an infinite AR process.

ii) The finite MA(q) process has an ACF that is zero beyond q. For an AR process, the ACF is infinite in length and consists of mixture of damped exponentials and/or damped sine waves.

iii) Finite MA process are always stable, and there is no requirement on the coefficients of MA processes for stationarity. However, for invertibility, the roots of the characteristic equation must lie inside the unit circle.

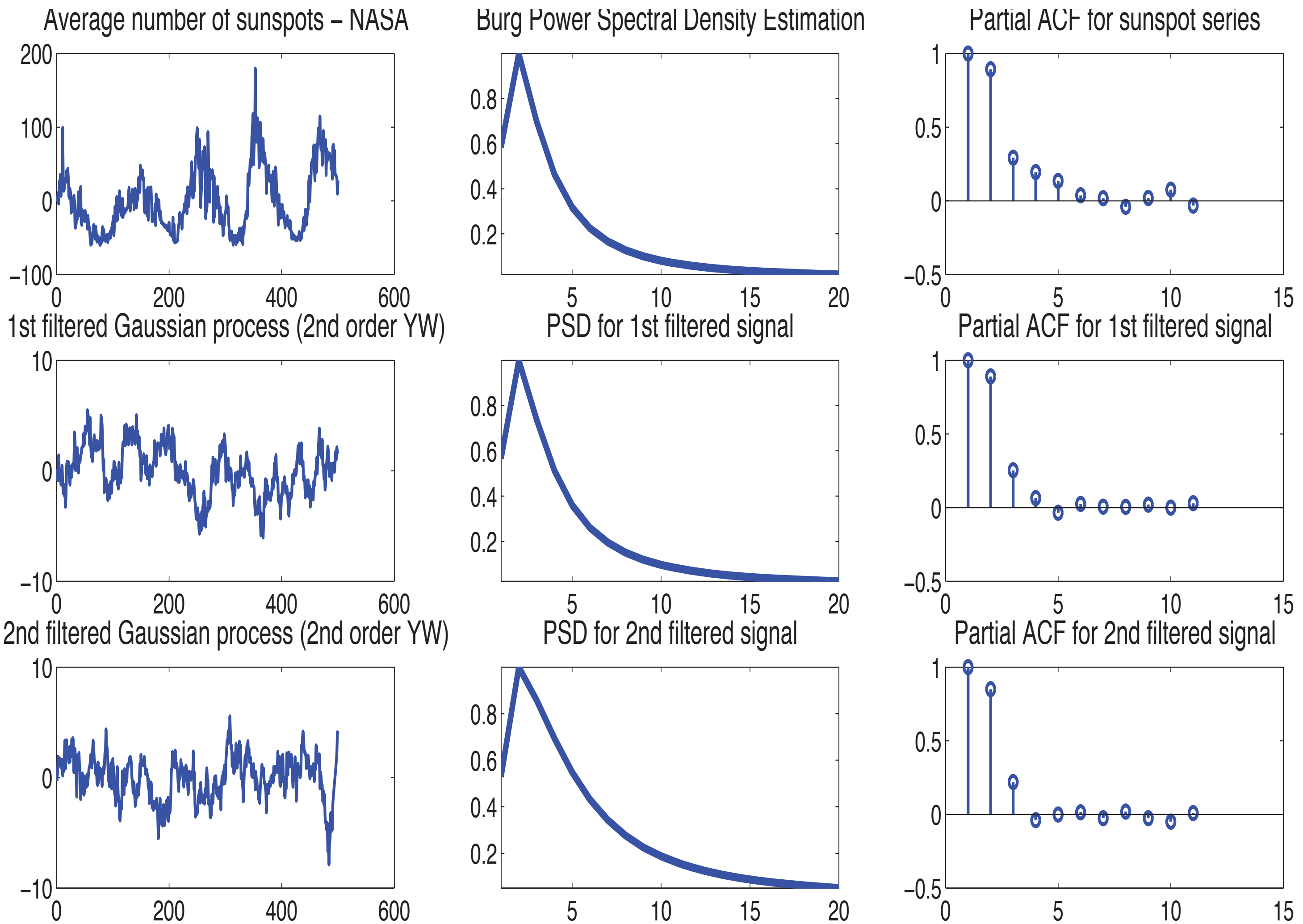
iv) AR processes produce spectra with sharp peaks (two poles of $A(z)$ per peak), whereas MA processes cannot produce peaky spectra.

ARMA modelling is a classic technique which has found a tremendous number of applications

Appendix 1: More on numbers (recorded since 1874)

Top: original sunspots   Middle and Bottom: AR(2) representations

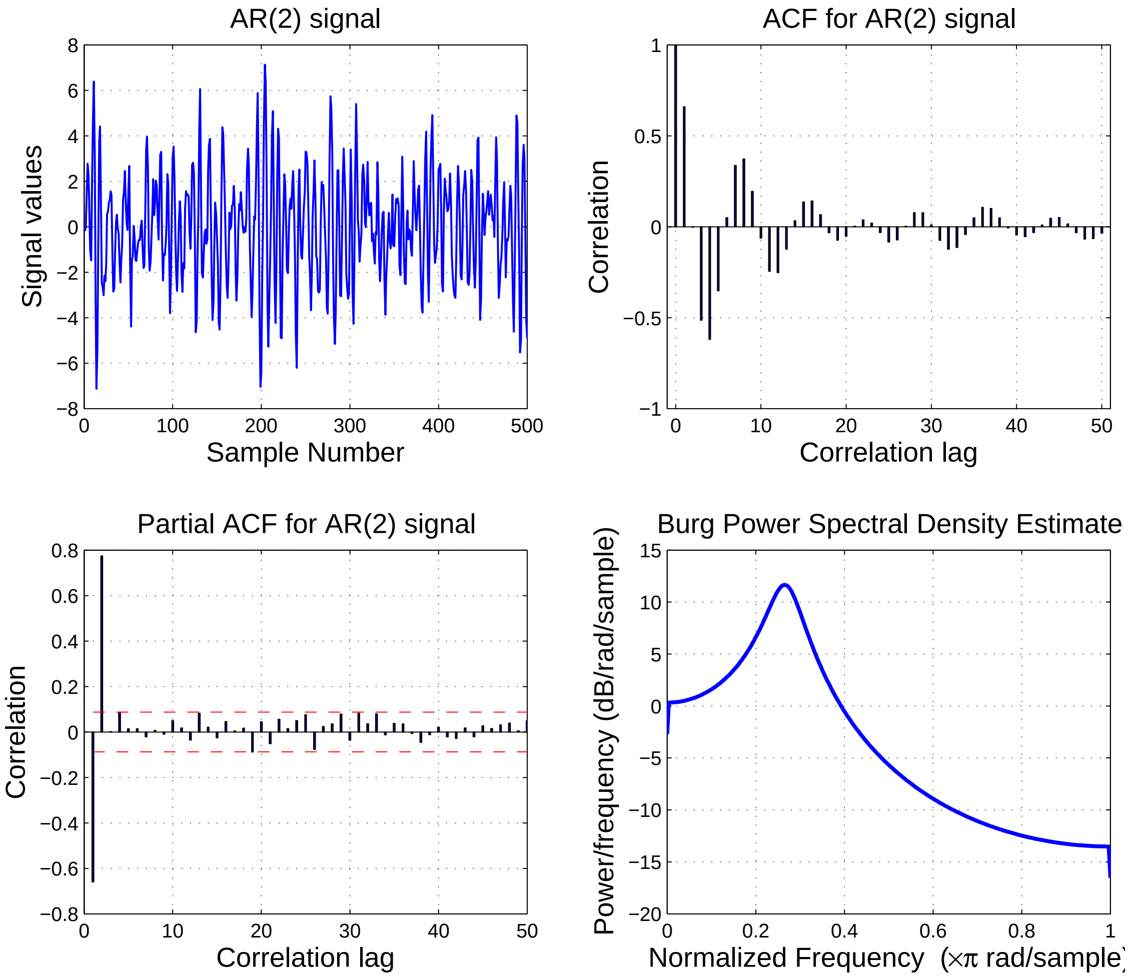
Left: time   Middle:spectrum   Right: autocorrelation



Top: original   Middle: first AR(2) model   Bottom: second AR(2) model

Appendix 2: Model order for an AR(2) process

An AR(2) signal, its ACF, and its partial autocorrelations (PAC)



After lag $k = 2$, the PAC becomes very small

Something to think about ...

- What would be the properties of a multivariate (MV) autoregressive, say MVAR(1), process, where the quantities $\mathbf{w}$, $\mathbf{x}$, $\mathbf{a}$ now become matrices, that is

$$\mathbf{X}(n) = \mathbf{A}\mathbf{X}(n-1) + \mathbf{W}(n)$$

- Would the inverse of the multichannel correlation matrix depend on 'how similar' the data channels are? Explain also in terms of eigenvalues and 'collinearity'.

- Threshold autoregressive (TAR) models allow for the mean of a time series to change along the blocks of data. What would be the advantages of such a model?

- Try to express an AR(p) process as a state-space model. What kind of the transition matrix between the states do you observe?

Consider also: Fourier transform as a digital filter

We can see FT as a convolution of a complex exponential and the data (under a mild assumption of a one-sided h sequence, ranging from 0 to ∞)

1) Continuous FT. For a continuous FT $\quad F(\omega) = \int_{-\infty}^{\infty} x(t) e^{-\jmath\omega t} dt$

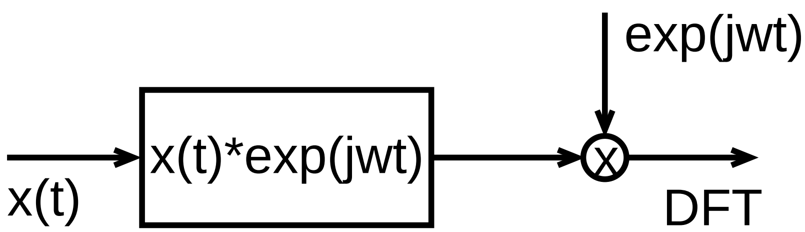
Let us now swap variables $t \rightarrow \tau$ and multiply by $e^{\jmath\omega t}$, to give

$$e^{\jmath\omega t} \int x(\tau) e^{-\jmath\omega\tau} dt = \int x(\tau) \underbrace{e^{\jmath\omega(t-\tau)}}_{h(t-\tau)} d\tau = x(t) * e^{\jmath\omega t} \qquad (= x(t) * h(t))$$

2) Discrete Fourier transform. For DFT, we have a filtering operation

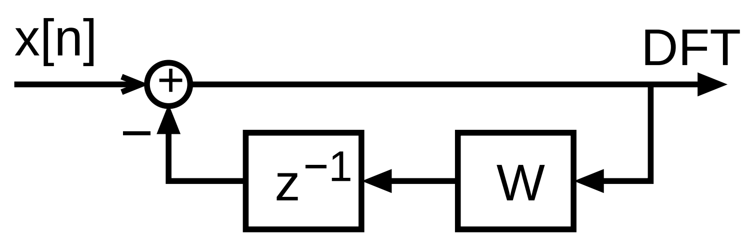
$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-\jmath\frac{2\pi}{N}nk} = \underbrace{x(0) + W\Big[x(1) + W\big[x(2) + \cdots\Big]}_{\text{cumulative add and multiply}} \qquad W = e^{-\jmath\frac{2\pi}{N}n}$$

with the transfer function (large N) $\quad H(z) = \frac{1}{1 - z^{-1}W} = \frac{1 - z^{-1}W^*}{1 - 2\cos\theta_k z^{-1} + z^{-2}}$



continuous time case

discrete time case



Notes

Notes

Notes