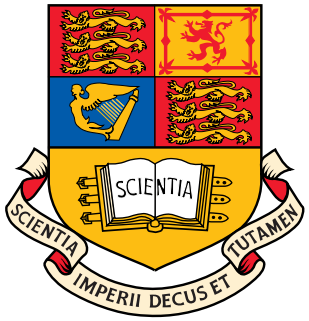
Statistical Signal Processing & Inference

Linear Stochastic Processes

Danilo Mandic

room 813, ext: 46271



Department of Electrical and Electronic Engineering

Imperial College London, UK

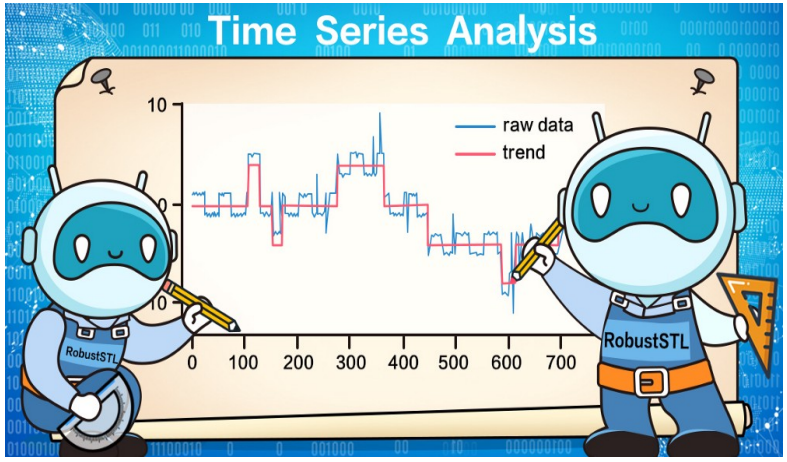
d.mandic@imperial.ac.uk, URL: www.commsp.ee.ic.ac.uk/∼mandic

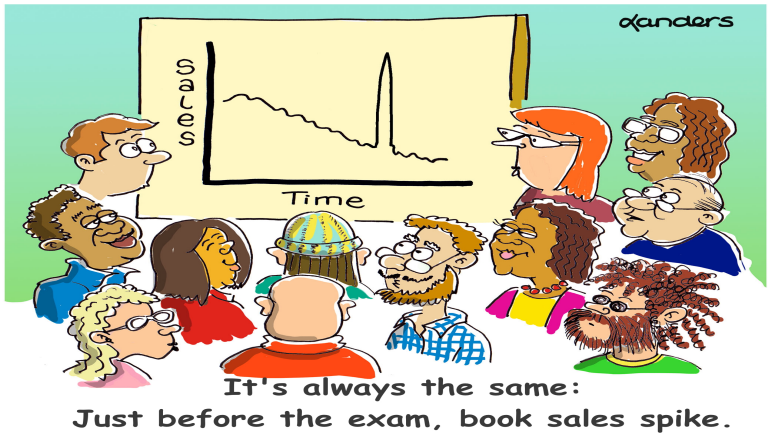
This lecture ↬ Time Series Modelling and Prediction

Q: Have you ever considered what the following tasks have in common?

- Forecasting of financial data
- Supply-demand modelling (e.g. electricity or air-ticket pricing)
- Modelling of COVID-19 spread
- Weather forecasting and modelling in astronomy (e.g. sunspots)
- Word generation by Large Language Models such as ChatGPT

A: These are time series of which the signal generating mechanisms are largely unknown or untractable. We need to make inference from such data based on historical observations (autoregression) – subject of this Lecture.





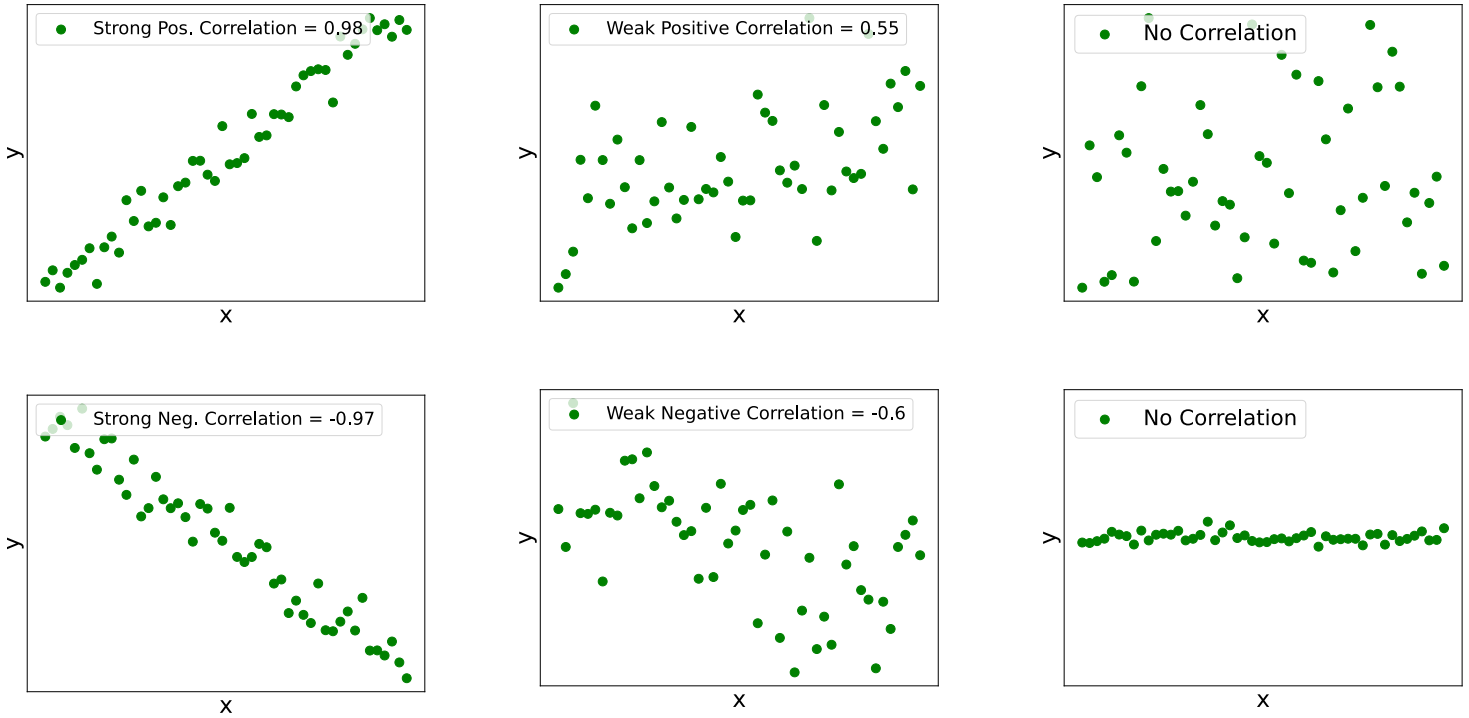
Aims of this lecture

- To introduce linear stochastic models for real world data
- Establish the general regression and auto-regression frameworks
- Understand how stochastic processes are created, and to get familiarised with the autocorrelation, variance and power spectrum of such processes
- Learn how to derive the parameters of linear stochastic ARMA models
- Introduce special cases: autoregressive (AR), moving average (MA)
- Stability conditions and model order selection (partial correlations)
- Optimal model order selection criteria (MDL, AIC, ...)
- Apply stochastic modelling to real world data (speech, environmental, finance), and address the issues of under– and over–modelling
- Provide grounding and intuition for Generative Autoregressive models

This material is a first fundamental step for real-world time series analysis

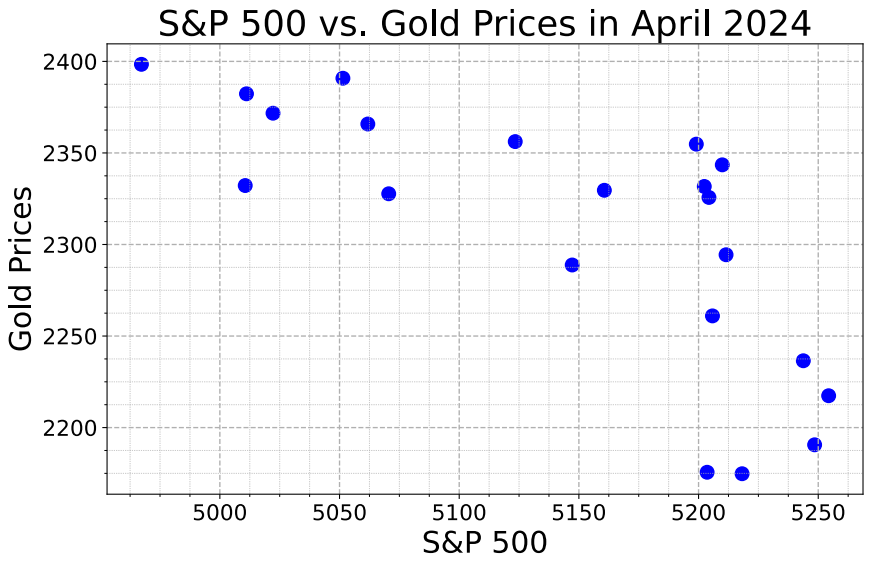
Relations between two variables: Scatter plots and correlation

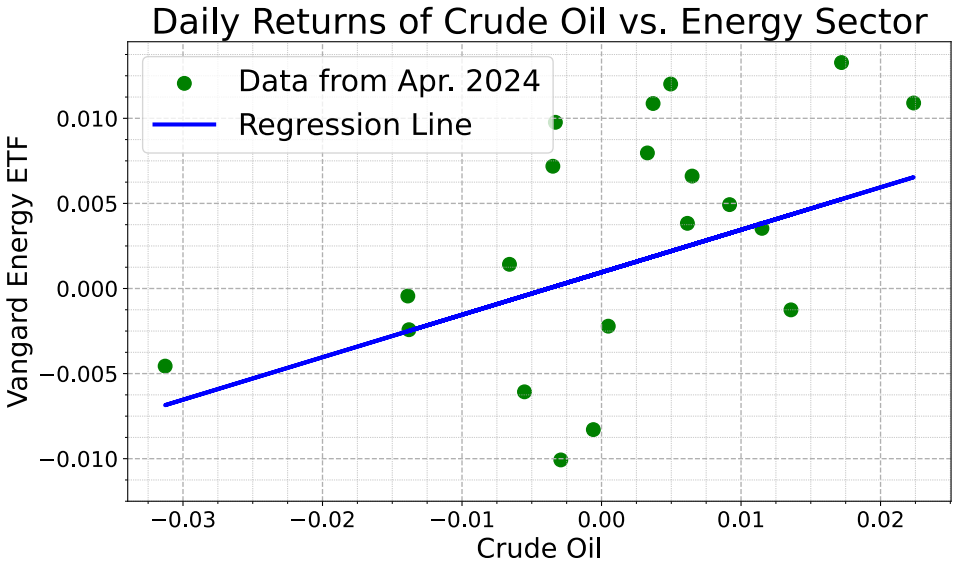
Correlation quantifies the strength (scatter) and direction of the **linear relationship** between two variables.



- In addition to the correlation between two variables, we would often like a quantitative description of how the two variables vary together.
- We would also like to perform prediction based on this knowledge $\rightarrow$ subject of **linear regression**. **Regression line is a unique line**

Correlation versus Regression





Correlation measures the strength of the relationship between the variables x and y in both the x-direction and y-direction.

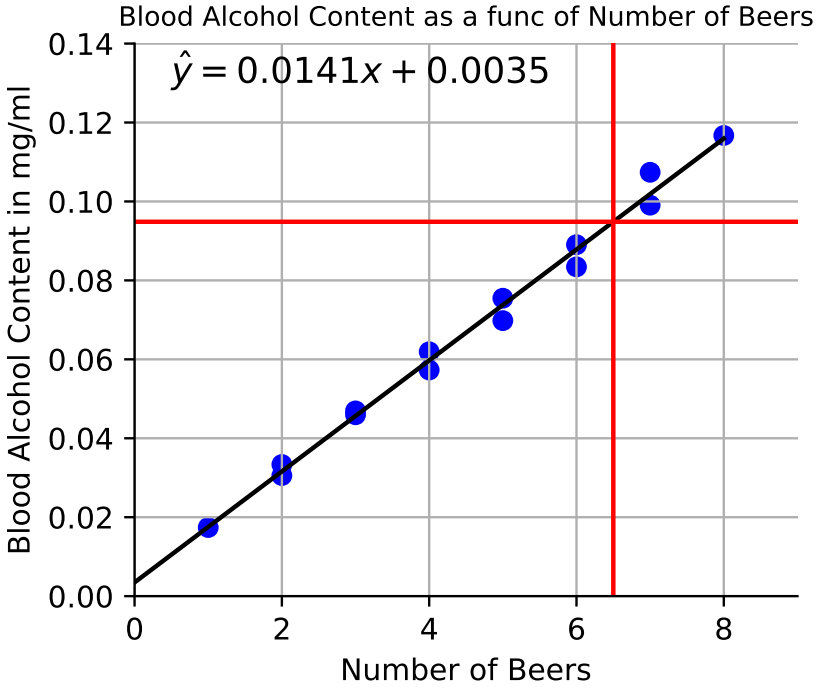
Regression examines the distance of all points from the regression line in the y-direction only; it models the variation of the **explained variable**, y, in response to the change in the **explanatory variable**, x.

- Variable x is also called a regressor, independent variable, or predictor
- Variable y is also called response, dependent var., criterion or true label

Regression: Advantages and limitations (see Lecture 6)

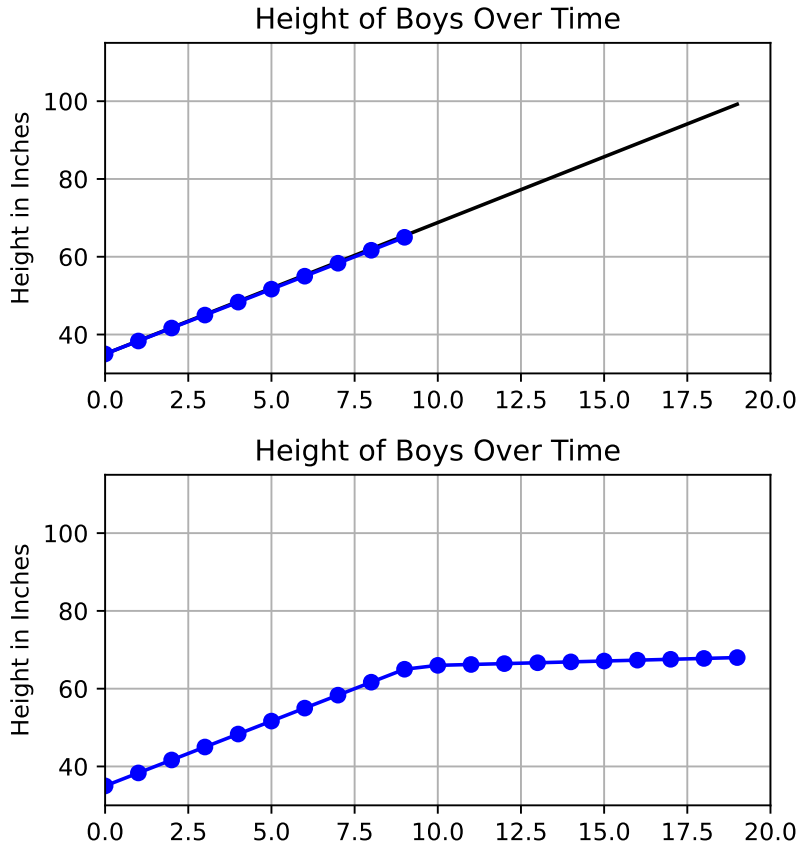
Motivation for Auto-Regression

Interpolation: Nobody in the study drunk 6.5 pints of beer, but we can still use regression to interpolate and find the estimated blood alcohol level.



Interpolation is quite accurate, as a linear fit matches the data.

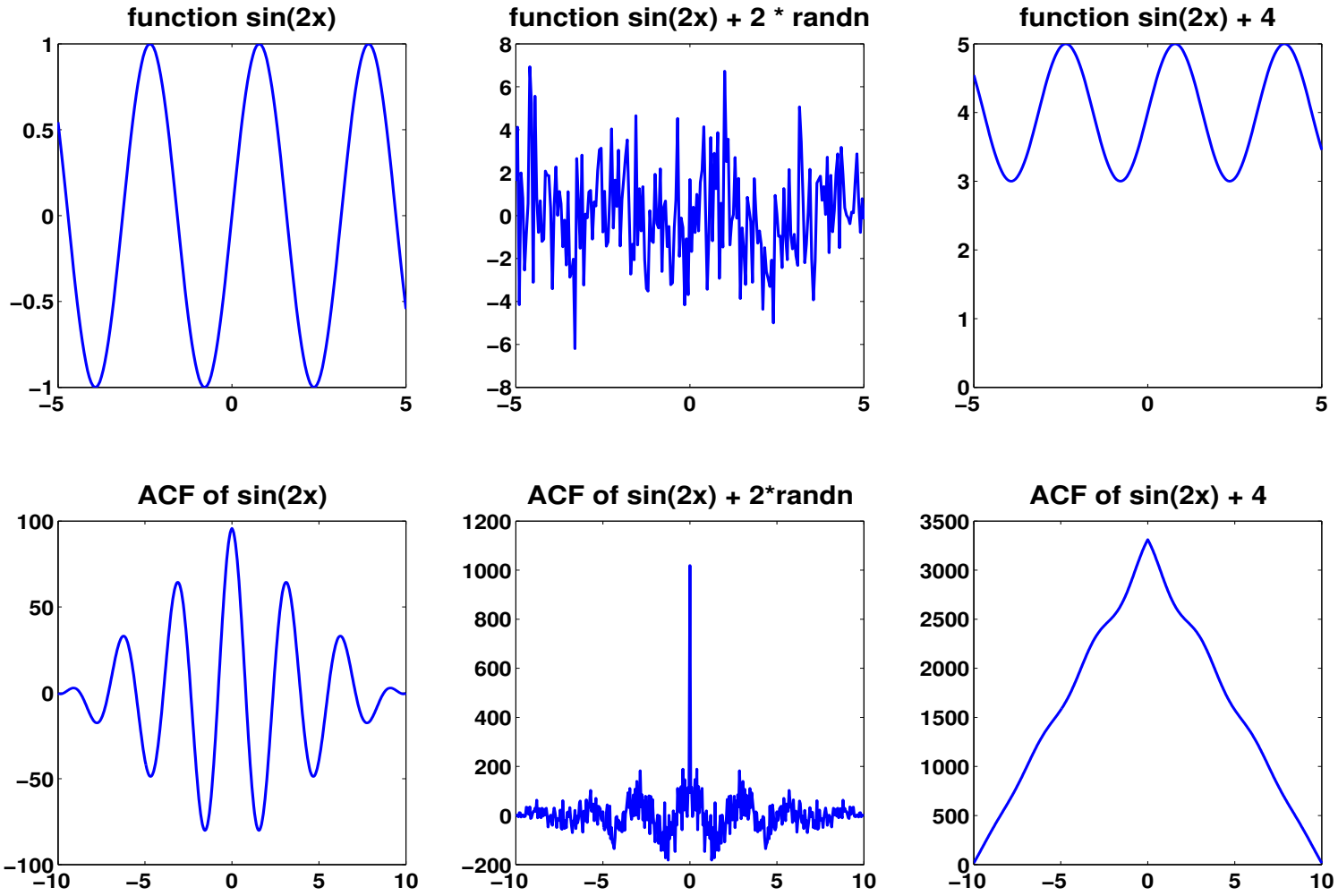
Extrapolation: Needs to be considered much more carefully.



A "piece-wise" linear fit would be more appropriate (or quadratic).

Example 1: Assessing the nature of a signal from its ACF

Windowed clean signal, signal in WGN, signal with DC offset (see also Lecture 1)



Which disturbance is more detrimental: deterministic DC or stochastic noise

Wold decomposition theorem
(Existence theorem, also mentioned in your coursework)

Wold's decomposition theorem plays a central role in time series analysis, and explicitly proves that any covariance–stationary time series can be decomposed into two different parts: **deterministic** (such as a sinewave) and **stochastic** (filtered WGN).

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

$$x[n] = x_p[n] + x_r[n] = x_p[n] + \sum_{j=1}^{q} b_j w[n-j] \qquad w \rightsquigarrow \text{white process}$$

$\Rightarrow x_r[n] \quad \not\rightsquigarrow$ regular random process

$\Rightarrow x_p[n] \quad \not\rightsquigarrow$ predictable process, with $x_r[n] \quad \perp x_p[n]$,

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

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

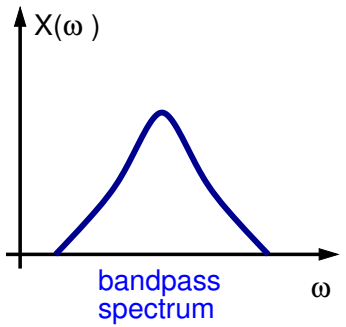
Our focus will be on the modelling of the random component

NB: Recall the difference between shift–invariance and time–invariance

How do we model a real world random signal?

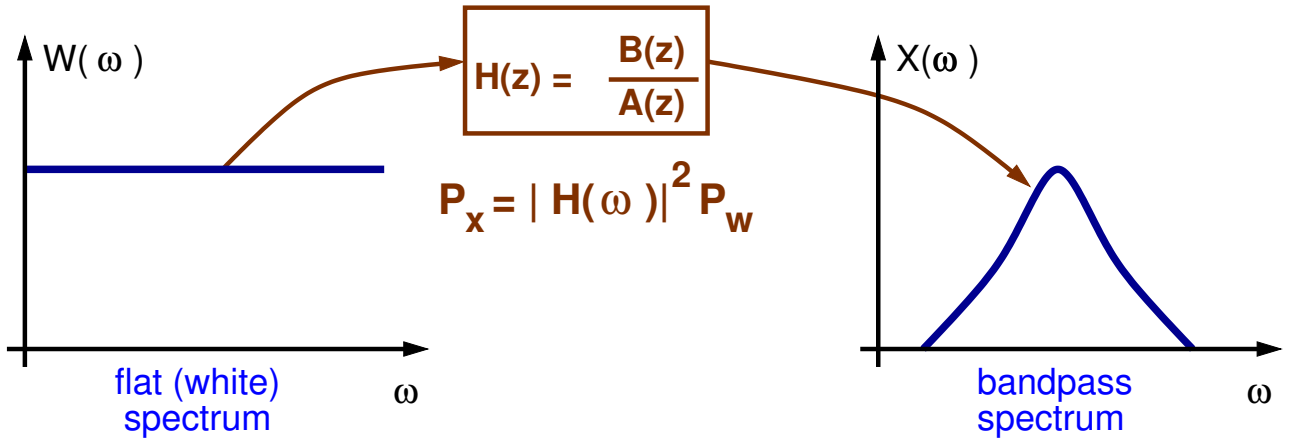
Suppose the measured real world signal has e.g. a bandpass (any other) power spectrum

Modelling: **We aim to describe the whole long signal with only very few parameters**



Q1: Can we model first and second statistics of real world signal by shaping up the white noise spectrum using some coefficients (transfer function)?

Q2: Does this produce the same second order stats as those of the original signal (mean, variance, ACF, spectrum) for any white noise input?



Can we use this linear stochastic model for prediction?

Towards linear stochastic processes

Wold's theorem implies that any purely non-deterministic covariance–stationary process can be arbitrarily well approximated by an ARMA process

From Wold's th., the power spectrum of a WSS process has the form

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

We are interested in processes generated by **filtering white noise with a linear shift–invariant filter** that has a rational system (transfer) function.

This class of digital filters includes the following system functions:

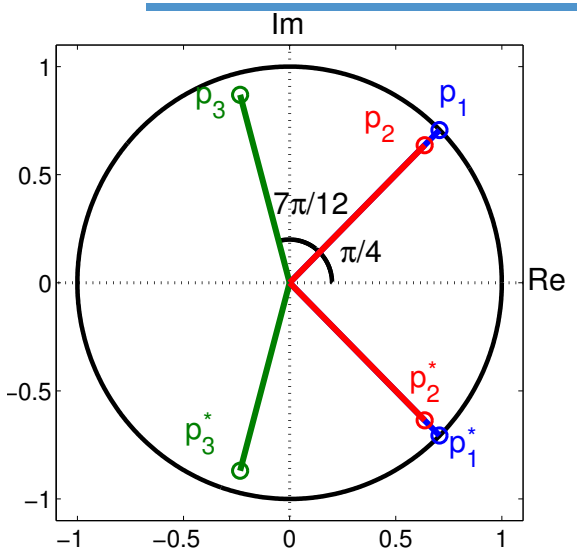
- Autoregressive (AR) $\rightarrow$ all pole system $\rightarrow$ $H(z) = 1/A(z)$
- Moving Average (MA) $\rightarrow$ all zero system $\rightarrow$ $H(z) = B(z)$
- Autoregressive Moving Average (ARMA) $\rightarrow$ poles and zeros $\rightarrow H(z) = B(z)/A(z)$

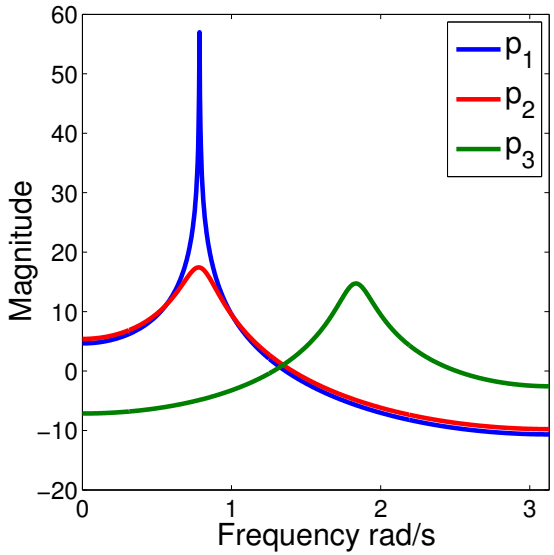
Definition: A covariance-stationary process $x[n]$ is called **(linearly) deterministic** if $p(x[n] \mid x[n-1], x[n-2], \ldots) = x[n]$.

☞ A stationary deterministic process, $x_p[n]$, can be predicted correctly (with zero error) using the entire past, $x_p[n-1], x_p[n-2], x_p[n-3], \ldots$

Example 2: Second-order all–pole systems and sinewave

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





To produce oscillations, we need two conjugate complex poles, e.g. p_1 and p_1^*, therefore

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

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

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

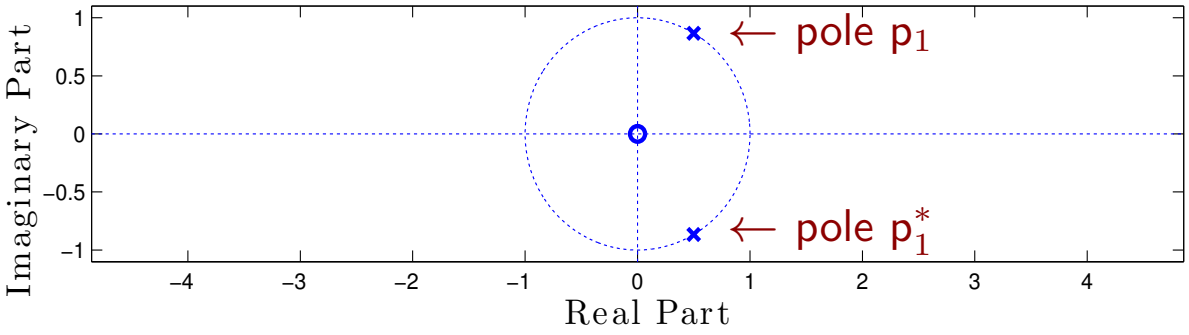
$$H(z) = \frac{1}{(1-\rho e^{j\theta} z^{-1})(1-\rho e^{-j\theta} z^{-1})} = \frac{1}{1-2\rho\cos(\theta) z^{-1} + \rho^2 z^{-2}}$$

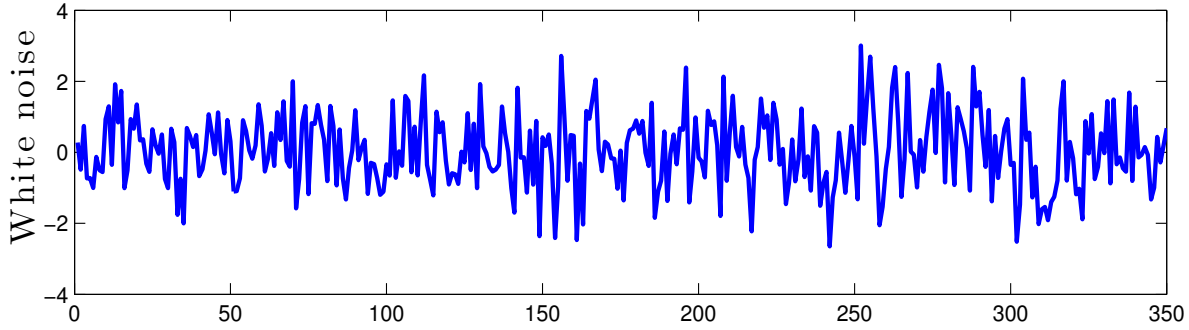
$$\text{for the sinewave } \rho = 1 \quad \Rightarrow \quad H(z) = \frac{1}{1-2\cos(\theta) z^{-1} + z^{-2}} = \frac{1}{1+a_1 z^{-1} + a_2 z^{-2}}$$

☞ **Indeed, a sine can be modelled as an autoregressive AR(2) process**

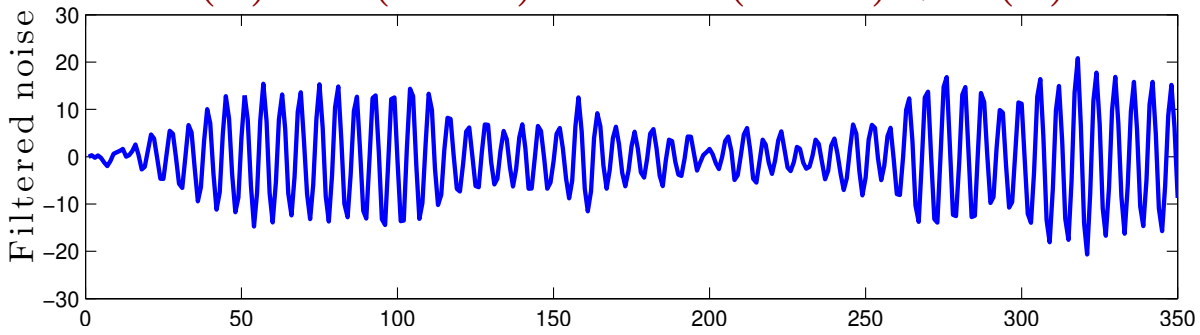
Example 2: Sinewave revisited, is it determ. or stoch.?

Is a sinewave best described as nonlinear deterministic or linear stochastic?





$$x(n) = x(n-1) - 0.98x(n-2) + w(n)$$



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

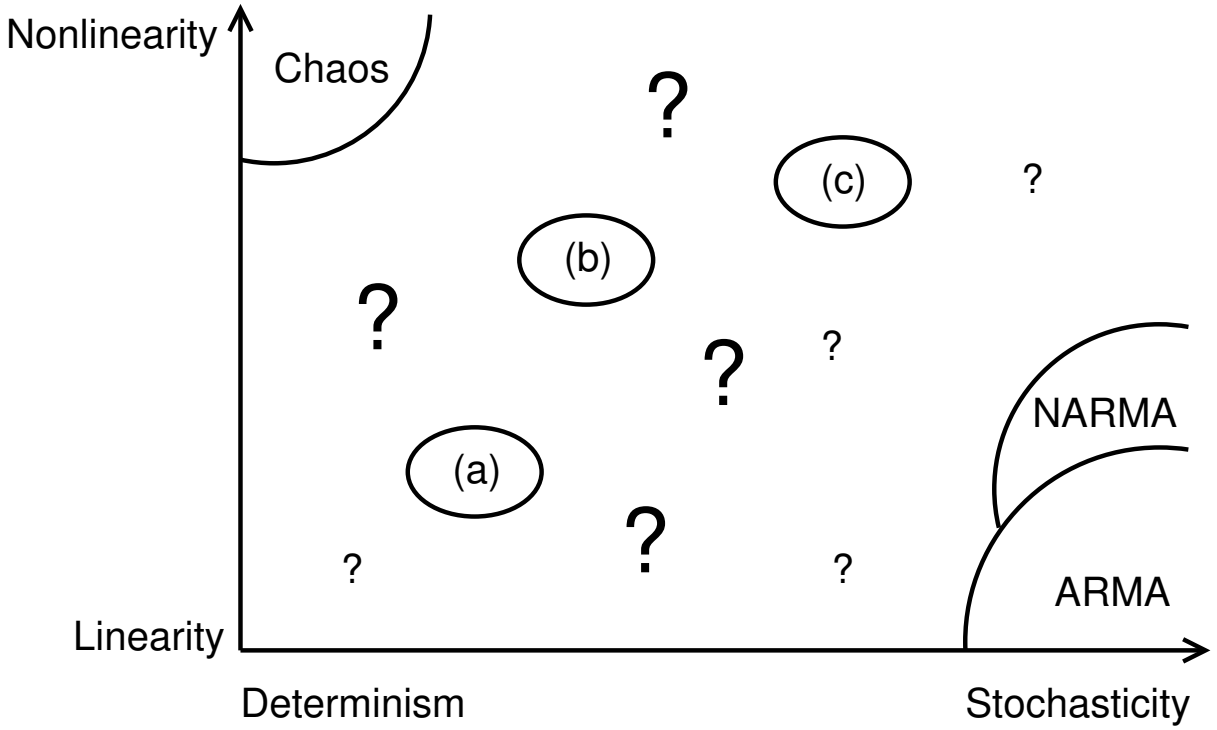
```
x(n)=a1*x(n-1)+a2*x(n-2)+w(n)
a1=1, a2=-0.98, w~N(0,1)
```

How can we categorise real–world measurements?

where would you place a DC level in WGN, $x[n] = A + w[n], \quad w \sim \mathcal{N}(0, \sigma_w^2)$

(a) Noisy oscillations, (b) Nonlinearity and noisy oscillations, (c) Random nonlinear process

(? left) Route to chaos, (? top) stochastic chaos, (? middle) mixture of sources



Our lecture is about ARMA models (linear stochastic)

How about observing the signal through a nonlinear sensor?

Spectrum of ARMA models (Recap slides and Lecture 1)

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

$ACF \equiv PSD$ **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

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

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: power at the output of a linear system $P_y = |H(z)|^2 P_x = H(z)H^*(z)P_x$**)**

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

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

Example 3: Can the shape of power spectrum tell us about the order of the polynomials $B(z)$ and $A(z)$?

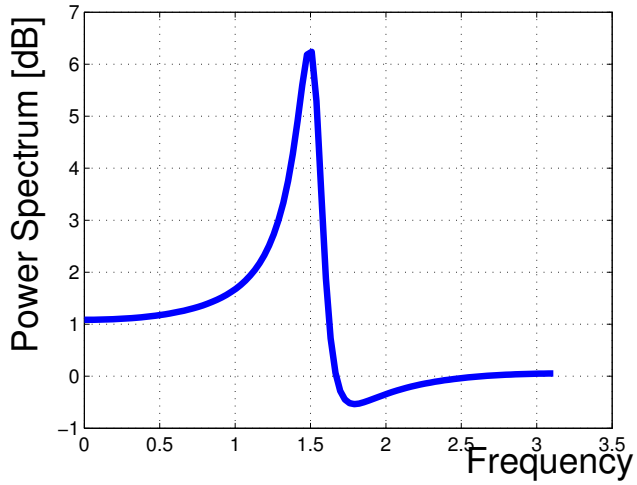
Plot the power spectrum of an ARMA(2,2) process for which

- 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)

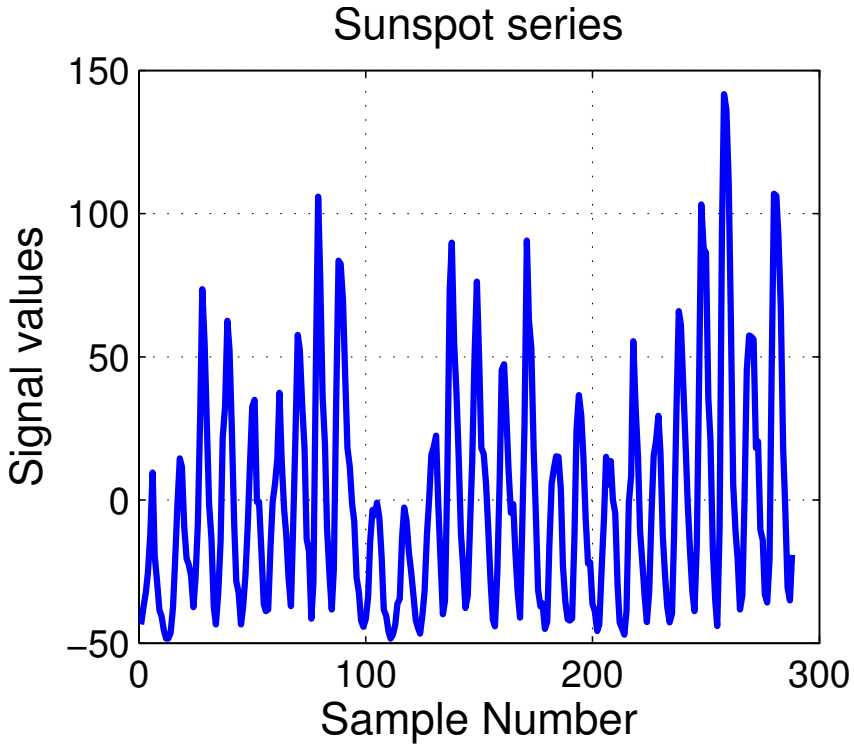
$$H(z) = \frac{X(z)}{W(z)} = \frac{1 + 0.9025z^{-2}}{1 - 0.5562z^{-1} + 0.81z^{-2}}$$

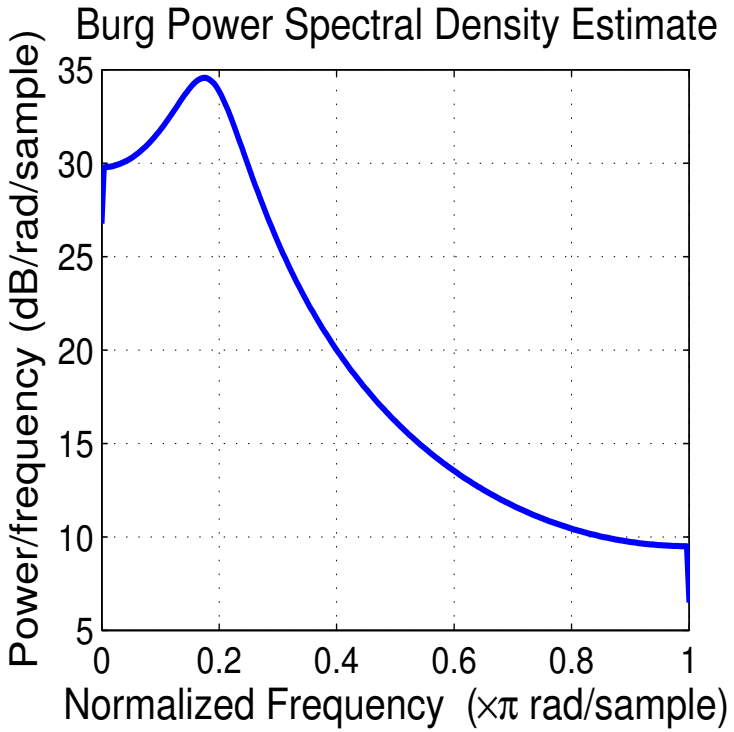
$$\Rightarrow \quad x(n) = 0.5562x(n-1) - 0.81x(n-1) + w(n) + 0.9025w(n-2)$$



Example 4: Power spectrum of real–world Sunspot data

Sunspot numbers and their PSD $\rightsquigarrow$ even their AR(2) linear model is powerful!





Recorded from about 1700 onwards

This signal is random, as sunspots originate from the explosions of helium on the Sun. Still, the number of sunspots obeys a relatively simple model and is predictable, as shown later in the Lecture.

Difference equations $\nrightarrow$ the ACF follows the data model!

(for convenience, a slight abuse in notation from A(z) to the autoregressive part)

Since $X(z) = H(z)W(z)$, the random processes $x[n]$ and $w[n]$ are related by a linear difference equation with constant coefficients, given by

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

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

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

(also Slide 18 & App. 3)

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

General linear processes: Stationarity and invertibility

Can we tell anything about the process x from the coefficients a, b (cf. h in FIR)

Consider a linear stochastic process $\looparrowright$ output from a linear filter, driven by WGN, denoted by $w[n]$. (**NB: Here w is "input" and x "output"**)

$$x[n] = w[n] + b_1 w[n-1] + b_2 w[n-2] + \cdots = w[n] + \sum_{j=1}^{\infty} b_j w[n-j]$$

that is, **a weighted sum** of past samples of **driving white noise** $w[n]$.

For this process to be a valid stationary process, the coefficients must be **absolutely summable**, that is $\sum_{j=0}^{\infty} |b_j| < \infty$.

This also implies that under stationarity conditions, $x[n]$ is also a weighted sum of past values of x, plus an added shock $w[n]$, that is

$$x[n] = a_1 x[n-1] + a_2 x[n-2] + \cdots + w[n]$$ (see also Slide 30)

- Linear Process is *stationary* if $\sum_{j=0}^{\infty} |b_j| < \infty$
- Linear Process is *invertible* if $\sum_{j=0}^{\infty} |a_j| < \infty$

Recall that $H(\omega) = \sum_{n=0}^{\infty} h(n) e^{-\jmath\omega n} \quad \rightarrow \text{for } \omega = 0 \quad \Rightarrow H(0) = \sum_{n=0}^{\infty} h(n)$

Autoregressive processes (pole–only)

A general $AR(p)$ process (autoregressive of order p) is given by

$$x[n] = a_1 x[n-1] + \cdots + a_p x[n-p] + w[n] = \sum_{i=1}^{p} a_i x[n-i] + w[n] = \mathbf{a}^T \mathbf{x}[n] + w[n]$$

Observe the auto–regression in $\{x[n]\} \looparrowright$ the past of x is used to generate the future

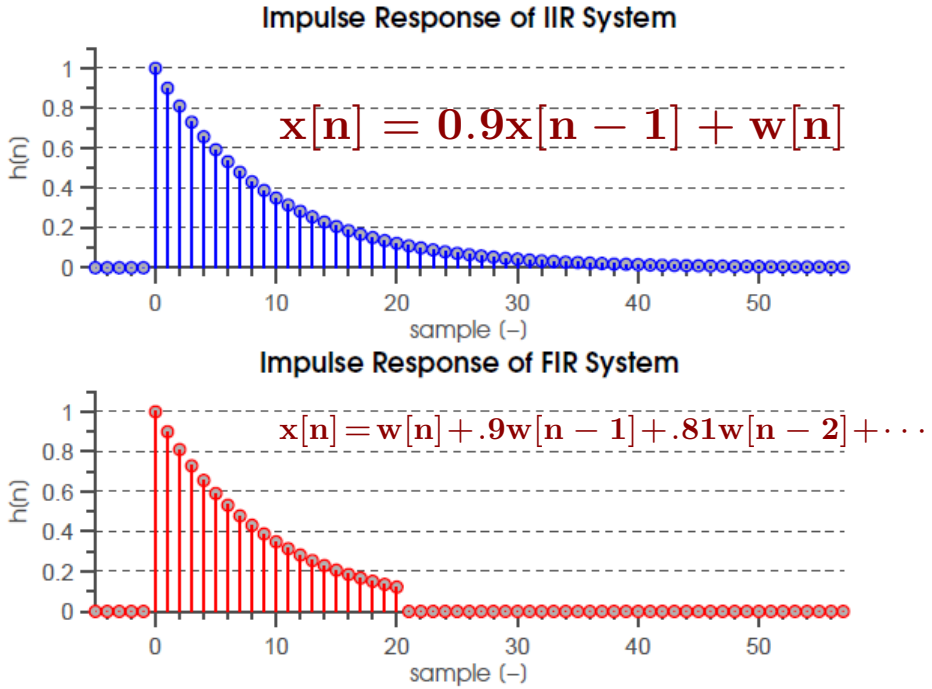
Duality between the AR and MA processes:

For example, consider the first order autoregressive process, AR(1),

$$x[n] = a_1 x[n-1] + w[n] \;\Leftrightarrow\; \sum_{j=0}^{\infty} b_j w[n-j]$$

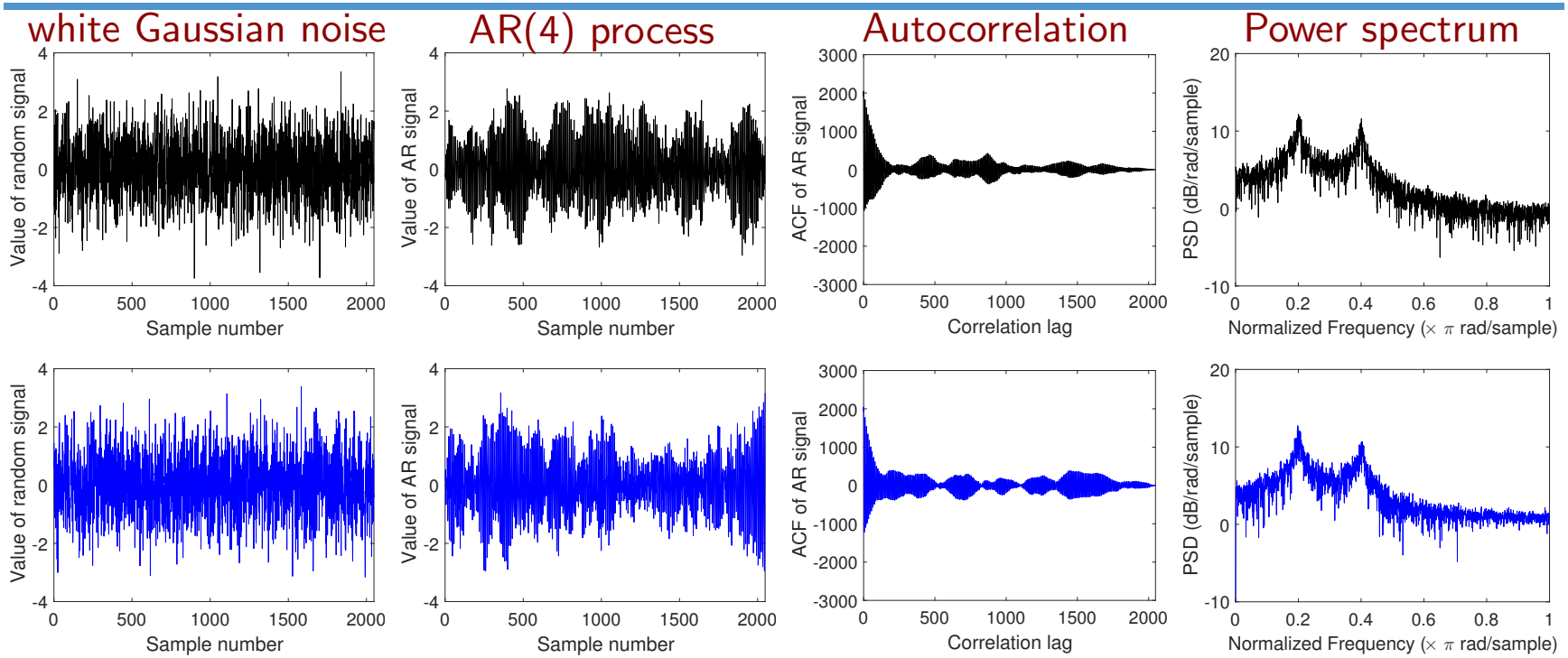
It has an MA representation, too.

This follows from the duality between IIR and FIR filters.



Example 5: Statistical properties of AR processes

Drive the AR(4) model from Example 6 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 taking a statistical approach**

ACF and normalised ACF of AR processes (see Appendix 3)

Key: ACF has the same form as the AR process in hand!

To obtain the autocorrelation function of an AR process, multiply the above equation by $x[n-k]$ to obtain (recall that $r(-m) = r(m)$)

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

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

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

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:

For $k = 0$, 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 Appendix)

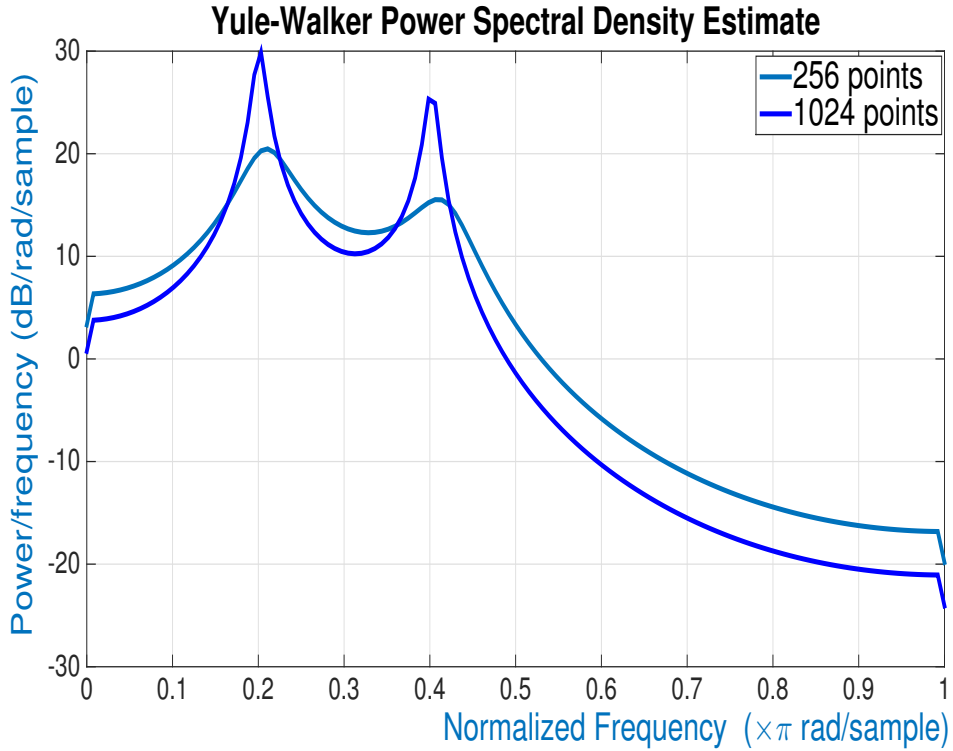
$$P_{xx}(f) = \frac{2\sigma_w^2}{|1 - a_1 e^{-\jmath 2\pi f} - \cdots - a_p e^{-\jmath 2\pi p f}|^2} \qquad 0 \leq f \leq 1/2$$

Fro more detail: *"Spectrum of Linear Systems"* from Lecture 1: Background

Example 6a: $AR(p)$ signal generation vs # data points

Consider an AR(4) process with coeff. $\mathbf{a} = [2.2137, -2.9403, 2.1697, -0.9606]$

- 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:

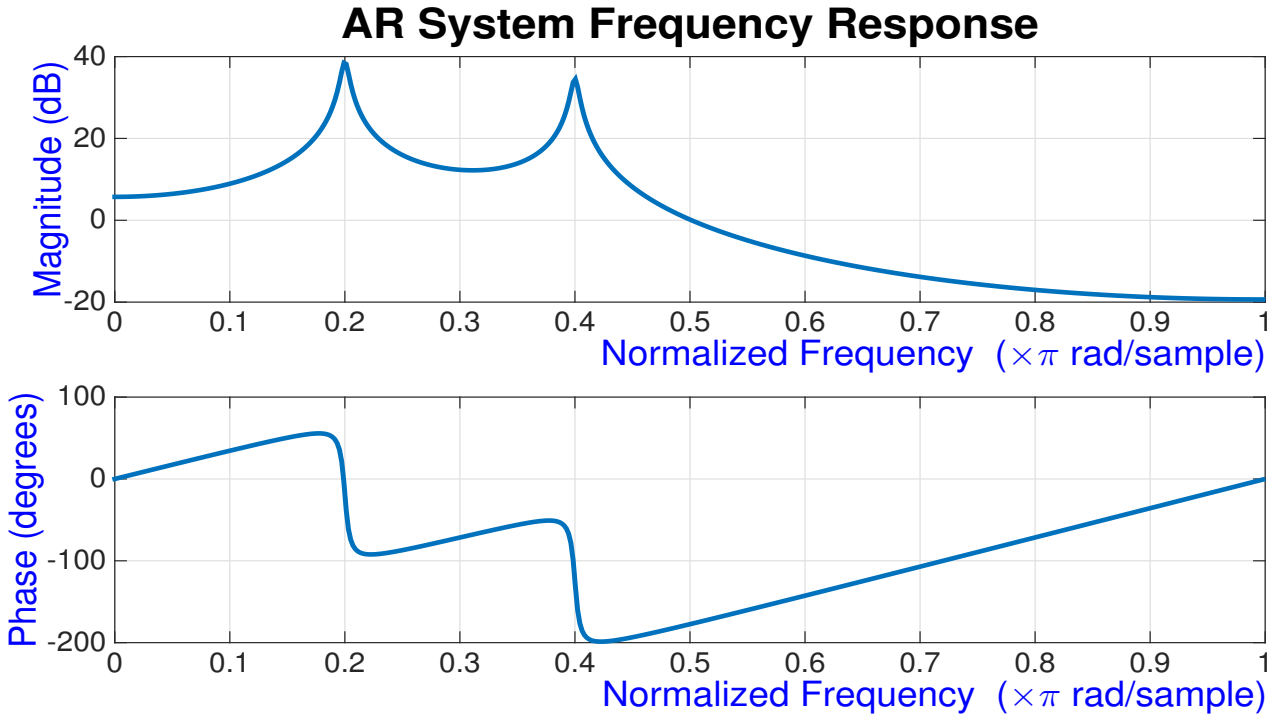
```
randn('state',1);
x = filter(1,a,randn(256,1));     % AR system output
pyulear(x,4)                      % Fourth-order estimate
```

Example 6b: Alternative AR power spectrum calculation (an alternative function in Matlab)

Consider the AR(4) system given by

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

```
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')
```



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

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

☞ Recall that we have already calculated the autocorrelation function (ACF) which follows the same form as the AR(p) model as the data, that is

$$r_{xx}(k) = a_1 r_{xx}(k-1) + a_2 r_{xx}(k-2) + \cdots + a_p r_{xx}(k-p), \qquad k > 0$$

Then, with the known $r_{xx}(0), \ldots, r_{xx}(p-1)$, we can build a system of p equations with p unknowns $(a_1, \ldots, a_p)$, and solve for $a_1, \ldots, a_p$, that is

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

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

☞ Their solution gives us the set of **autoregressive parameters**, $a_1, \ldots, a_p$, or $\mathbf{a} = [a_1, \ldots, a_p]^T$, which build the AR(p) model and **generate an AR(p) process.**

A vector–matrix form of the Yule–Walker equations

The above set of Yule–Walker equations can be expressed in a compact vector–matrix form

$$\mathbf{r}_{xx} = \mathbf{R}_{xx}\mathbf{a} \qquad \mathbf{a} = [a_1, a_2, \ldots, a_p]^T$$

since

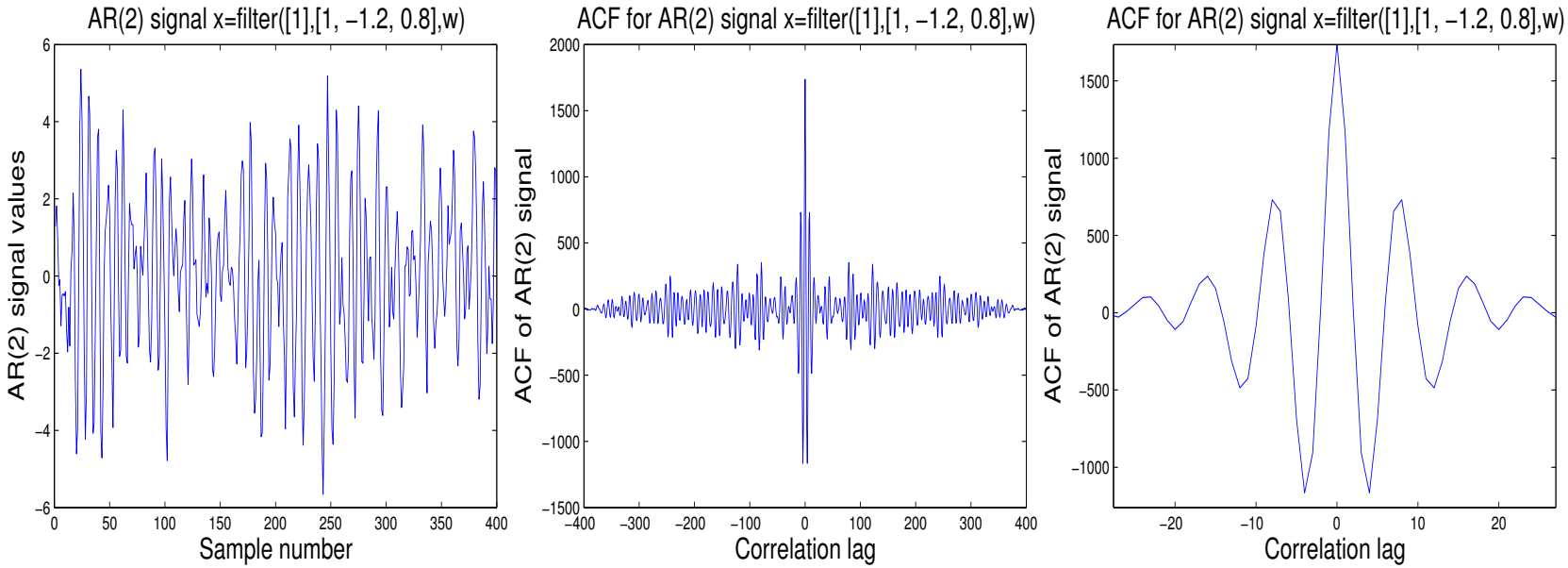
$$\mathbf{R}_{xx} = \begin{bmatrix} r_{xx}(0) & r_{xx}(1) & \cdots & r_{xx}(p-1) \\ r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_{xx}(p-1) & r_{xx}(p-2) & \cdots & r_{xx}(0) \end{bmatrix} \quad \text{and} \quad \mathbf{r}_{xx} = \begin{bmatrix} r_{xx}(1) \\ r_{xx}(2) \\ \vdots \\ r_{xx}(p) \end{bmatrix}$$

☞ The ACF matrix $\mathbf{R}_{xx}$ is positive definite (Toeplitz) which guarantees matrix inversion, so that the Yule-Walker solution for the unknown AR(p) coeficients, $[a_1, \ldots, a_p]^T = \mathbf{a}$, becomes

Yule-Walker solution: $\mathbf{a} = \mathbf{R}_{xx}^{-1}\mathbf{r}_{xx}$

Example 7: Find the parameters of an AR(2) process, $x(n)$, generated by $x[n] = 1.2x[n-1] - 0.8x[n-2] + w[n]$

Homework: Comment on the shape of the ACF for large lags



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

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

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

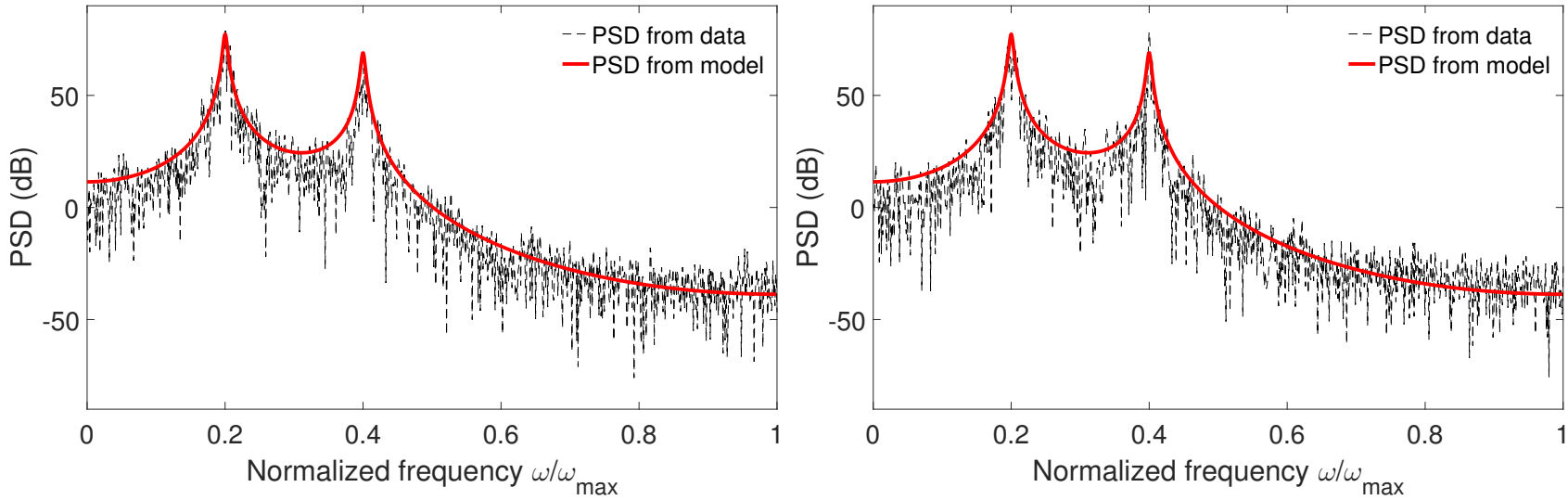
$\mathbf{a}^{(4)} = [1.1762, -0.7513, -0.0456, 0.0083]$

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

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

Example 8: Advantages of model-based analysis

Consider the PSDs for different realisations of the AR(4) process from Example 5



- The different realisations of the AR(4) process (based on different driving WGNs) lead to different Empirical 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
```

Normal equations for the autocorrelation coefficients

Standard correlations can take any value, so it is often convenient to consider correlations coefficients

$$\rho_k = r_{xx}(k)/r_{xx}(0) \qquad -1 < \rho_k < 1$$

which are normalised by $r_{xx}(0)$ (signal power), and take values $\in [-1, 1]$. Then, the Yule-Walker equations expressed in terms of ρ_k become

$$\begin{aligned}
\rho_1 &= a_1 + a_2\rho_1 + \cdots + a_p\rho_{p-1} \\
\rho_2 &= a_1\rho_1 + a_2 + \cdots + a_p\rho_{p-2} \\
\vdots &= \vdots \\
\rho_p &= a_1\rho_{p-1} + a_2\rho_{p-2} + \cdots + a_p
\end{aligned}$$

Q: When does the sequence $\{\rho_0, \rho_1, \rho_2, \ldots\}$ vanish?

Homework: Explore the command **xcorr** in Matlab

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**

Stochastic modelling: From raw data to ARMA(p,q) proc.

So far, we have assumed the model (AR, MA, or ARMA) and analysed the ACF and PSD based on known model coefficients.

In practice: DATA $\rightsquigarrow$ MODEL

This procedure is as follows:

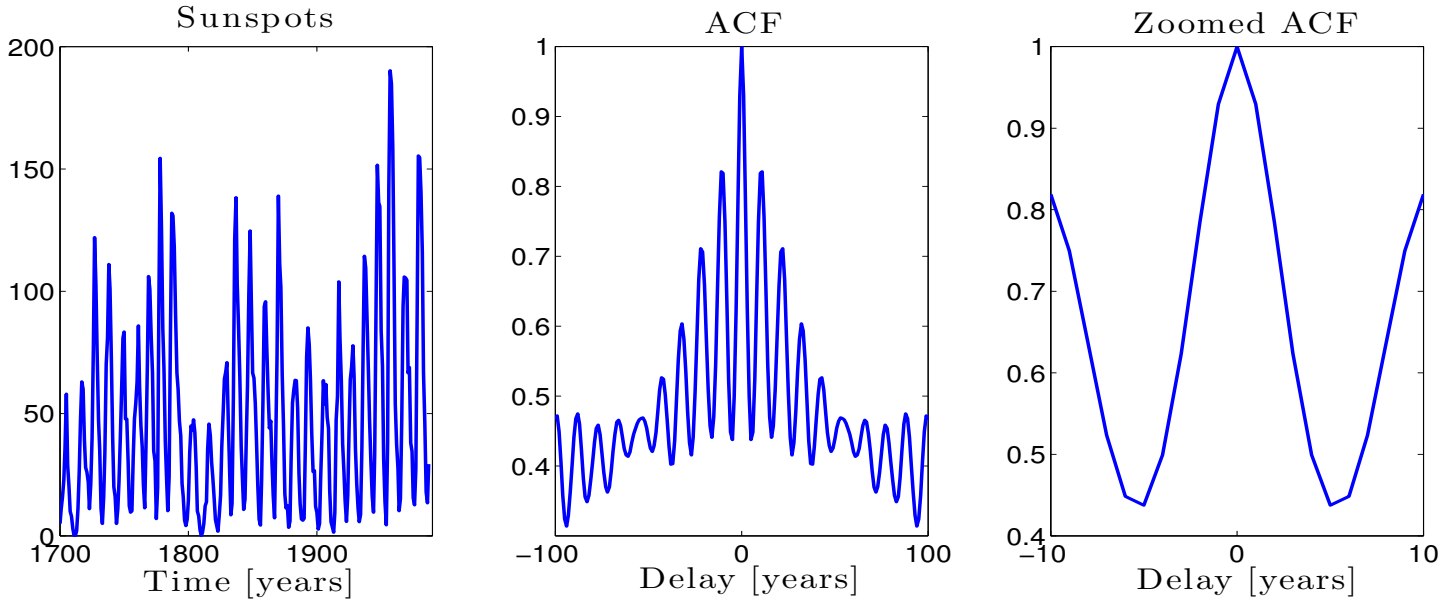
```
* record data x(k)
* find the autocorrelation of the data ACF(x)
* divide by r_xx(0) to obtain correlation coefficients \rho(k)
* write down Yule-Walker equations
* solve for the vector of AR parameters
```

The problem is that we do not know the model order p beforehand.

An intuition into the choice of the correct morel order is given in the following example.

Example 9: Sunspot number estimation

consistent with the properties of a second order AR process



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

☞ 'Best" model is AR(2): $x[n] = 1.474\,x[n-1] - 0.5857\,x[n-2] + w[n]$

Special case #1: AR(1) process (Markov)

For Markov processes, we have the **first order conditional expectation**, given by

$$p(x[n]|x[n-1], x[n-2], \ldots, x[0]) = p(x[n]|x[n-1])$$

$$\text{Then} \quad x[n] = a_1 x[n-1] + w[n] = \underbrace{w[n] + a_1 w[n-1] + a_1^2 w[n-2] + \cdots}_{\text{equivalent MA}(\infty)\text{ process}}$$

Therefore: order-1 memory

i) For the AR(1) process to be **stationary** $-1 < a_1 < 1$.

ii) **Autocorrelation Function of AR(1):** From the Yule-Walker equations

$$r_{xx}(k) = a_1 r_{xx}(k-1), \quad k > 0$$

$$r_0 = a_1 r_{xx}(1) + \sigma_w^2, \quad k = 0$$

In terms of the **correlation coefficients**, $\rho(k) = r(k)/r(0)$, with $\rho_0 = 1$

$$\rho_k = a_1^k, \quad k > 0$$

Notice the difference in the behaviour of the ACF for a positive and negative a_1

Variance and power spectrum of AR(1) process

Both can be calculated directly from the general expression for the variance and spectrum of $AR(p)$ processes, given in Slide 22.

- **Variance:** Also from a general expression for the variance of linear processes from Lecture 1

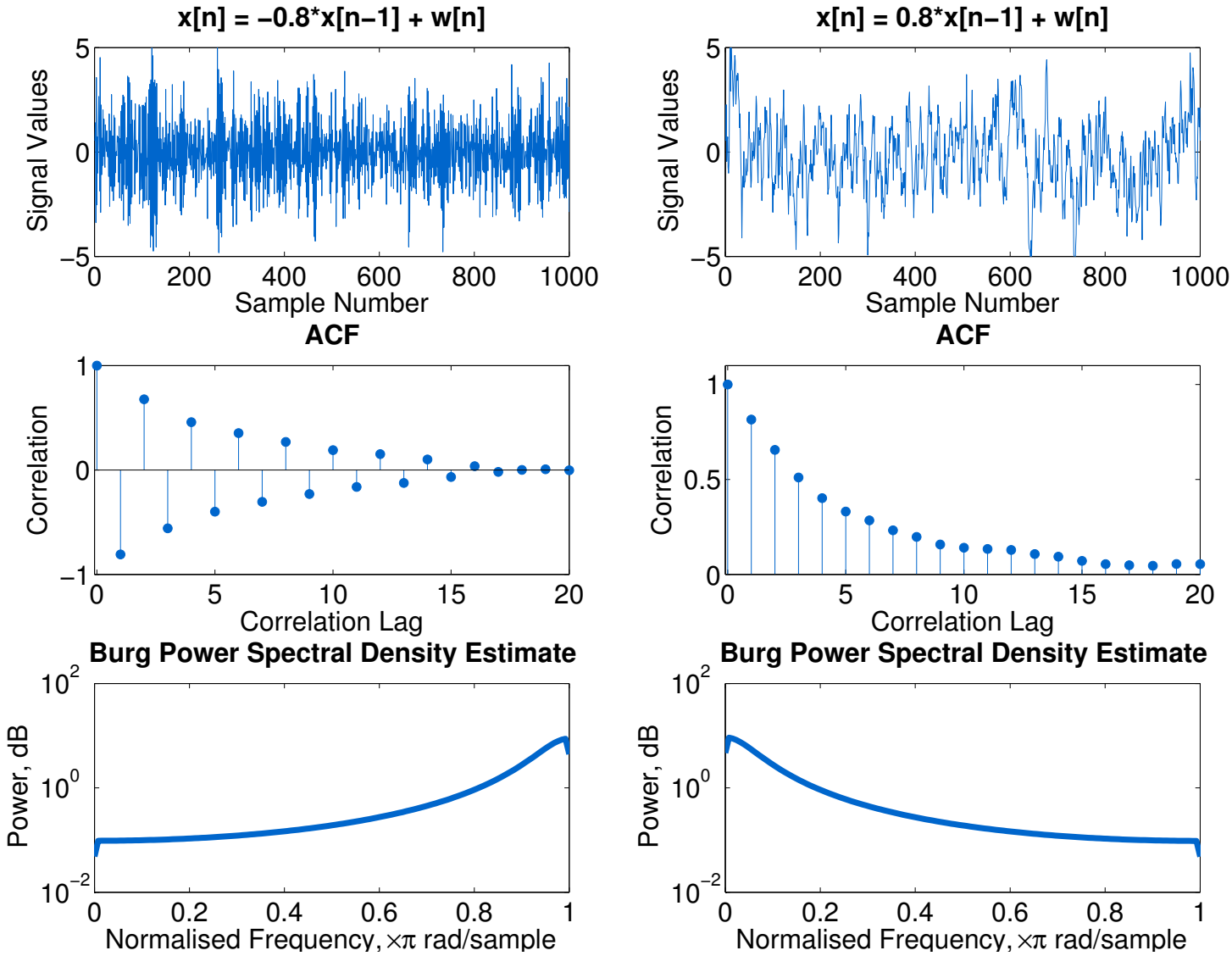
$$\sigma_x^2 = \frac{\sigma_w^2}{1 - \rho_1 a_1} = \frac{\sigma_w^2}{1 - a_1^2}$$

- **Power spectrum:** Notice how the flat PSD of WGN is shaped according to the position of the pole of $AR(1)$ model (Low-Pass for $0 < a_1 < 1$ and High-Pass for $-1 < a_1 < 0$)

$$P_{xx}(f) = \frac{2\sigma_w^2}{\left|1 - a_1 e^{-\jmath 2\pi f}\right|^2} = \frac{2\sigma_w^2}{1 + a_1^2 - 2a_1 cos(2\pi f)}$$

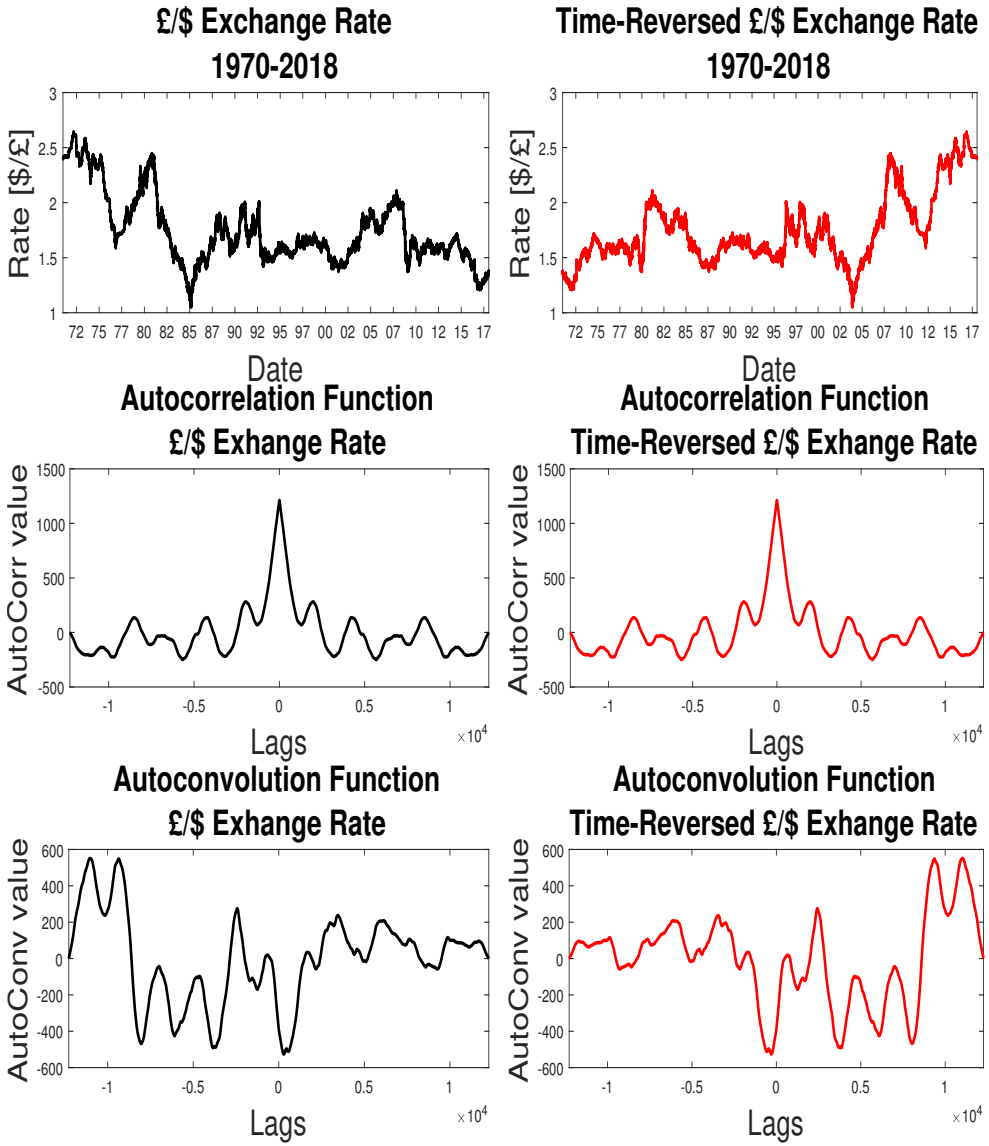
Example 10 a): ACF and spectrum of AR(1) for $a = \pm 0.8$

$a < 0 \rightarrow$ High Pass $\qquad$ $a > 0 \rightarrow$ Low Pass



Example 10 b): Model order of a financial time series

(the 'correct' and 'time-reversed' time series)



Autoregressive coefficients:

AR(1): $\mathbf{a} = [0.9994]$

AR(2): $\mathbf{a} = [.9994, -.0354]$

AR(3): $\mathbf{a} = [.9994, -.0354, -.0024]$

AR(4): $\mathbf{a} = [.9994, -.0354, -.0024, .0129]$

AR(5): $\mathbf{a} = [.9994, -.0354, -.0024, .0129, -.0129]$

AR(6): $\mathbf{a} = [.9994, -.0354, -.0024, .0129, -.0129, -.0172]$

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)

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

$$H(\omega) = H(e^{\jmath\omega}) = \frac{1}{1 - a_1 e^{-\jmath\omega} - a_2 e^{-2\jmath\omega}} \quad \rightarrow \quad P_{xx}(\omega) = |H(\omega)|^2 \sigma_w^2$$

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 $\rho_1 < \rho(0) = 1 \not\mapsto$ a **stability condition** on a_1 and a_2

Variance and power spectrum

(see Slide 22)

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

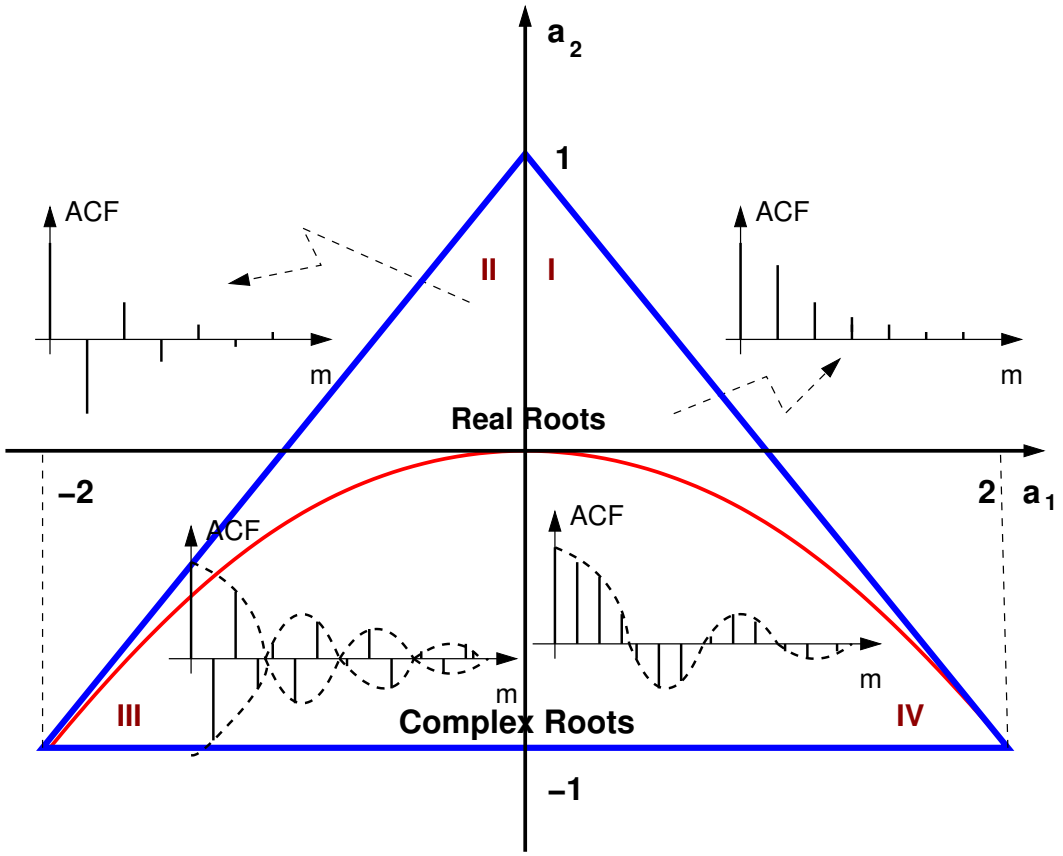
$$\begin{aligned} P_{xx}(f) &= \frac{2\sigma_w^2}{\left|1 - a_1 e^{-\jmath 2\pi f} - a_2 e^{-\jmath 4\pi f}\right|^2} \\ &= \frac{2\sigma_w^2}{1 + a_1^2 + a_2^2 - 2a_1(1 - a_2\cos(2\pi f) - 2a_2\cos(4\pi f))}, \quad 0 \le f \le 1/2 \end{aligned}$$

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"

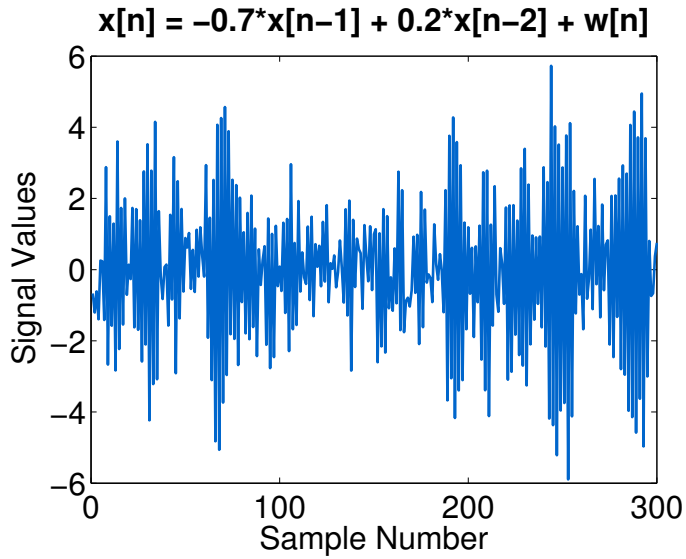
Stability triangle

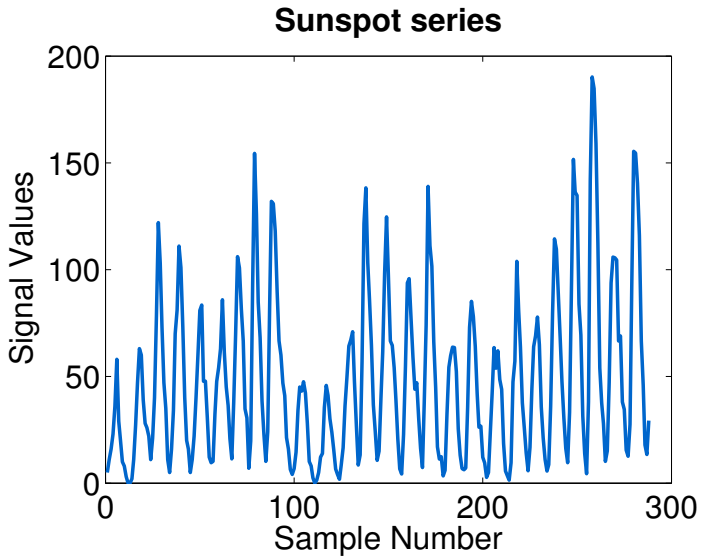


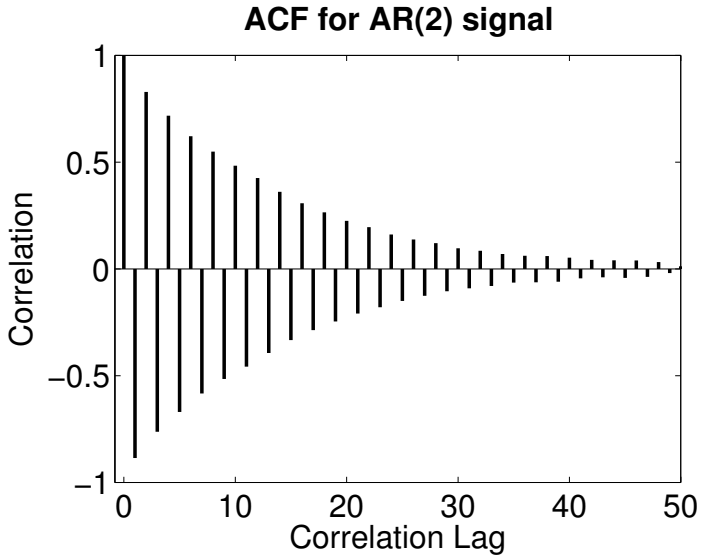
i) **Real roots** Region 1: Monotonically decaying ACF
ii) **Real roots** Region 2: Decaying oscillating ACF
iii) **Complex roots** Region 3: Oscillating pseudo-periodic ACF
iv) **Complex roots** Region 4: Pseudo-periodic ACF

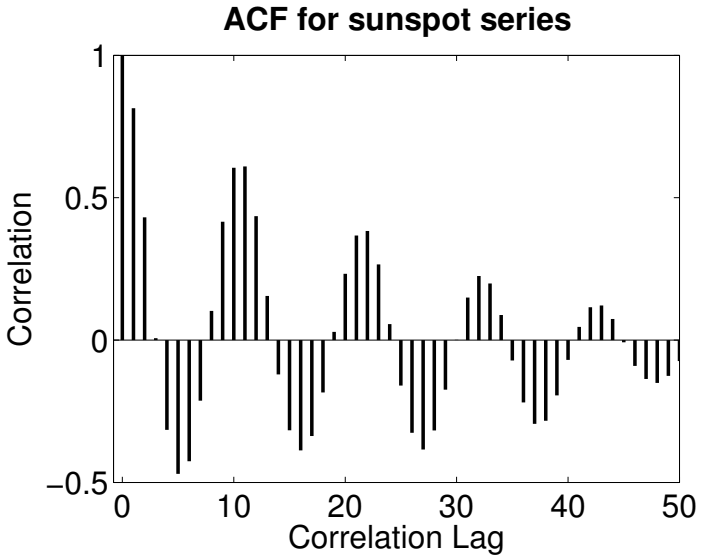
Example 11: Stability triangle and ACFs of AR(2) signals

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









Determining regions in the stability triangle

let us examine the autocorrelation function of AR(2) processes

The ACF

$$\rho_k = a_1\rho_{k-1} + a_2\rho_{k-2} \qquad k > 0$$

- **Real roots:** $\Rightarrow (a_1^2 + 4a_2 > 0)$ ACF $\looparrowright$ mixture of damped exponentials
- **Complex roots:** $\Rightarrow (a_1^2 + 4a_2 < 0) \Rightarrow$ ACF exhibits a pseudo–periodic behaviour

$$\rho_k = \frac{D^k \sin(2\pi f_0 k + \Phi)}{\sin \Phi}$$

D - damping factor, of a sinewave with frequency f_0 and phase Φ

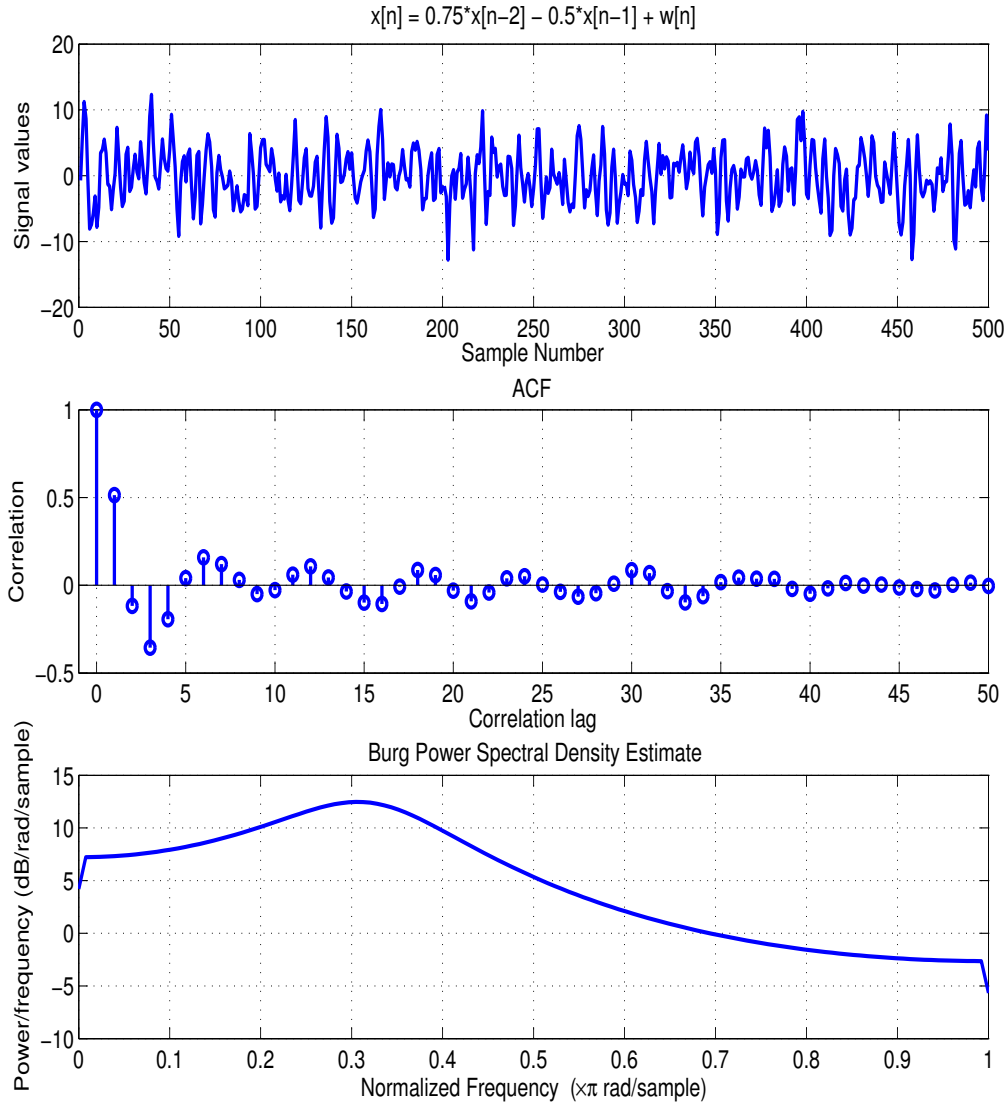
$$D = \sqrt{-a_2}$$

$$\cos(2\pi f_0) = \frac{a_1}{2\sqrt{-a_2}}$$

$$\tan(\Phi) = \frac{1 + D^2}{1 - D^2}\tan(2\pi f_0)$$

Example 12: AR(2) where $a_1 > 0$, $a_2 < 0$ $\quad \rightsquigarrow \quad$ Region 4

Consider: $x[n] = 0.75x[n-1] - 0.5x[n-2] + w[n]$



The damping factor

$$D = \sqrt{0.5} = 0.71,$$

Frequency

$$f_0 = \frac{\cos^{-1}(0.5303)}{2\pi} = \frac{1}{6.2}$$

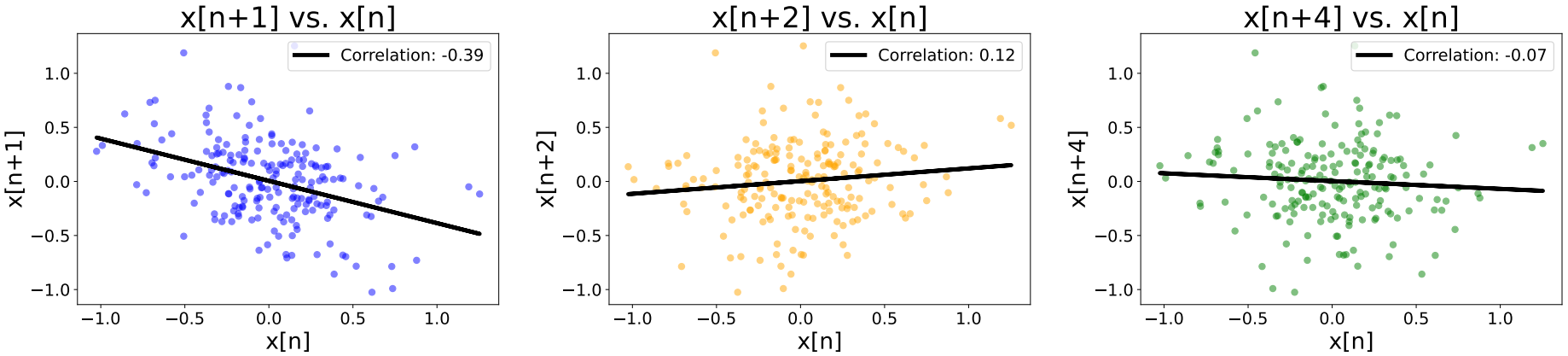
The fundamental period of the autocorrelation function is therefore

$$T_0 = 6.2.$$

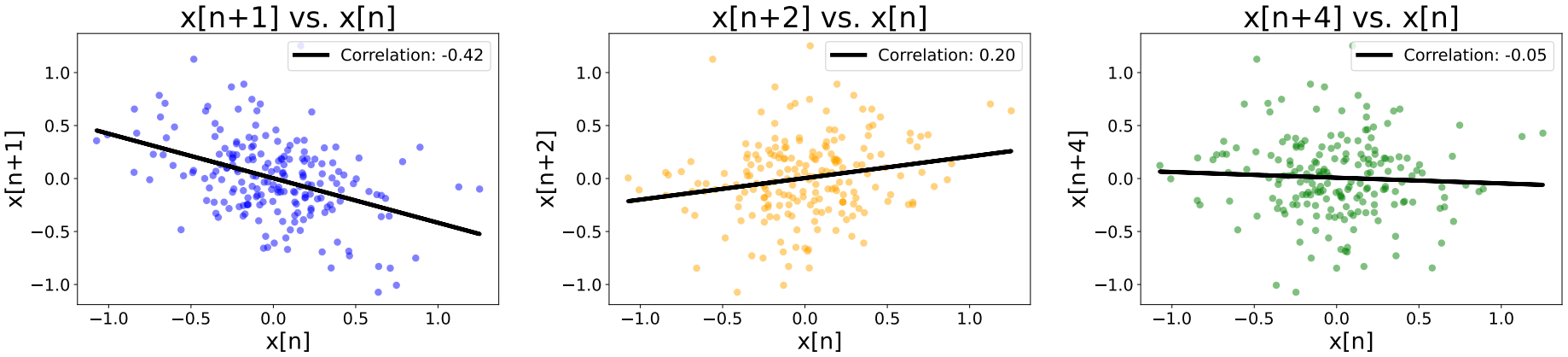
Model order selection ↬ Intuition

Consider an AR(1) process with $a_1 = -0.3$, and an AR(2) with $a_1 < 0, a_2 > 0$

The nature of an AR process may be inferred through scatter plots of pairs $x[n], x[m+n]$, separated by an interval (lag), m.



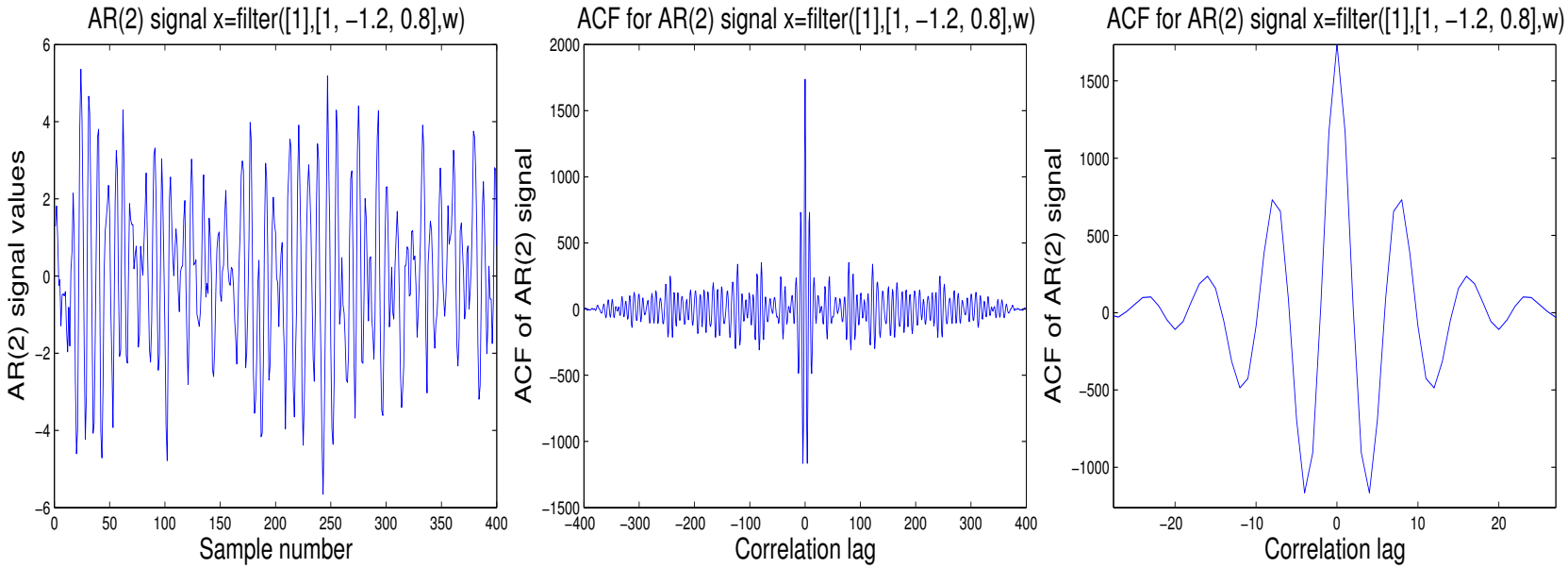
AR(1) process $x[n] = -0.3x[n-1] + w[n]$



AR(2) process $x[n] = -0.3x[n-1] + 0.1x[n-2] + w[n]$

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
see Appendix 5 for more detail

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 equations, 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 equations is the use of the symbols a_{ki} to denote the AR coefficient $a_i \not\mapsto k$ indicating the model order

Finding partial ACF coefficients

AR_2_Highpass_Circularity.m

Solving these equations for $k = 1, 2, \ldots$ successively, we obtain

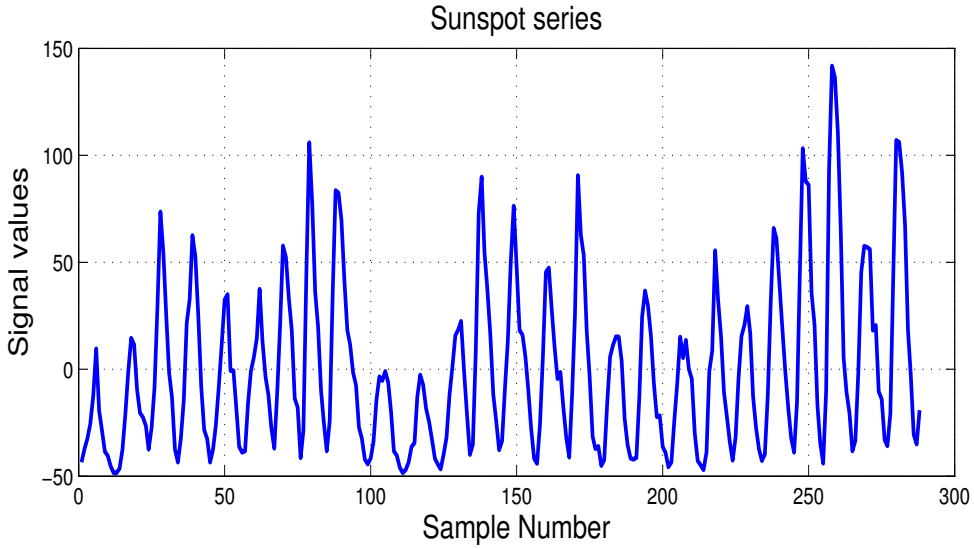
$$a_{11} = \rho_1, \qquad a_{22} = \frac{\rho_2 - \rho_1^2}{1 - \rho_1^2}, \qquad a_{33} = \frac{\begin{vmatrix} 1 & \rho_1 & \rho_1 \\ \rho_1 & 1 & \rho_2 \\ \rho_2 & \rho_1 & \rho_3 \end{vmatrix}}{\begin{vmatrix} 1 & \rho_1 & \rho_2 \\ \rho_1 & 1 & \rho_1 \\ \rho_2 & \rho_1 & 1 \end{vmatrix}}, \qquad \text{etc}$$

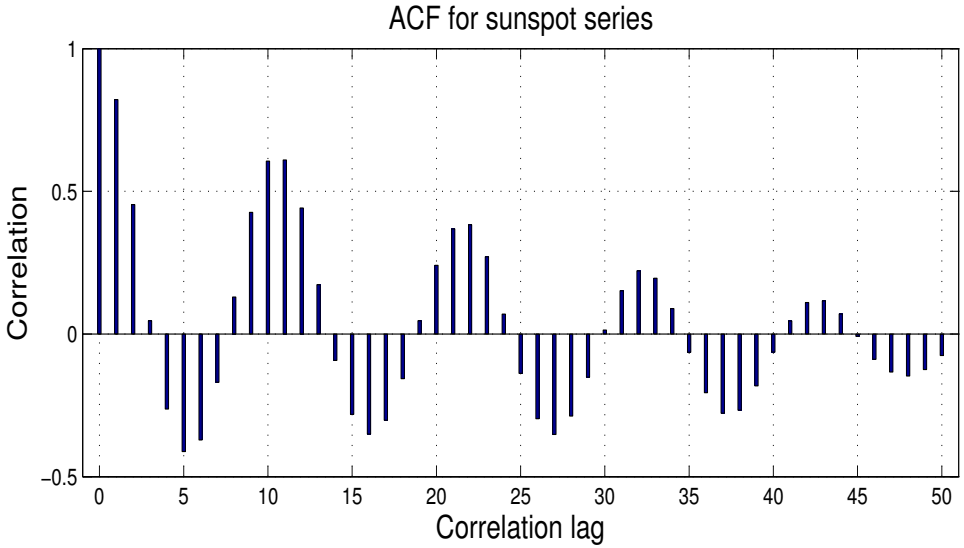
- The quantity a_{kk}, regarded as a function of the model order k, is called the **partial autocorrelation function (PAC)** (more details in Appendix 5)
- The PAC, a_{kk}, measures the linear correlation between $x(n)$ and $x(n-k)$, once we have removed the influence of $x_{n-1}, \ldots, x_{n-k+1}$
- For an AR(p) process, the PAC a_{kk} will be nonzero for $k \leq p$ and zero-valued for $k > p$ $\quad \rightarrowtail$ **indicates the order of an $AR(p)$ process.**

In practice, we introduce a small threshold, as for real world data it is difficult to guarantee that $a_{kk} = 0$ for $k > p$. (see your Coursework)

Example 13: 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]

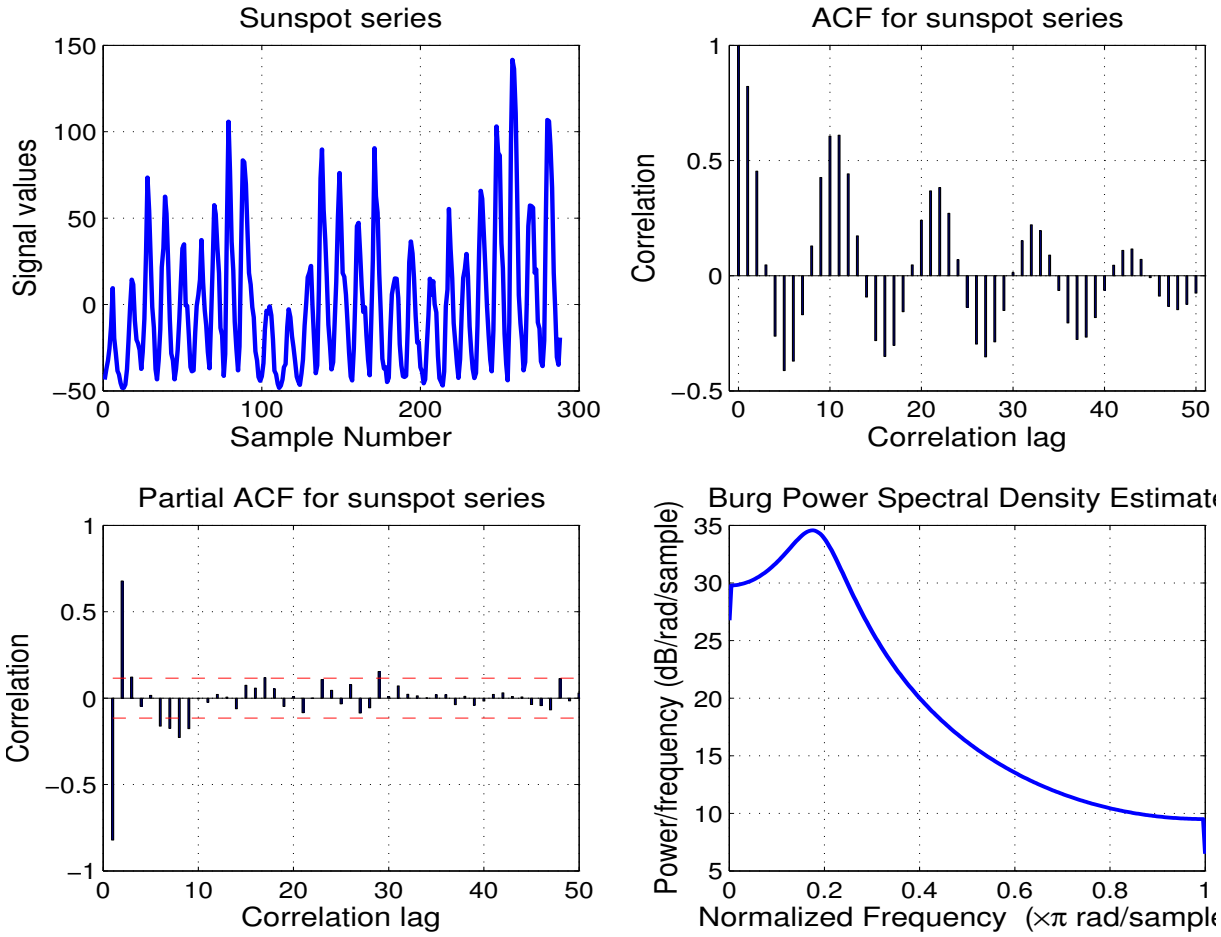
$\mathbf{a}_5$=[1.4773,-0.5377,-0.1739, 0.0174,0.1555]

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

See also Slide 9

Example 13 (contd.): Model order for sunspot numbers

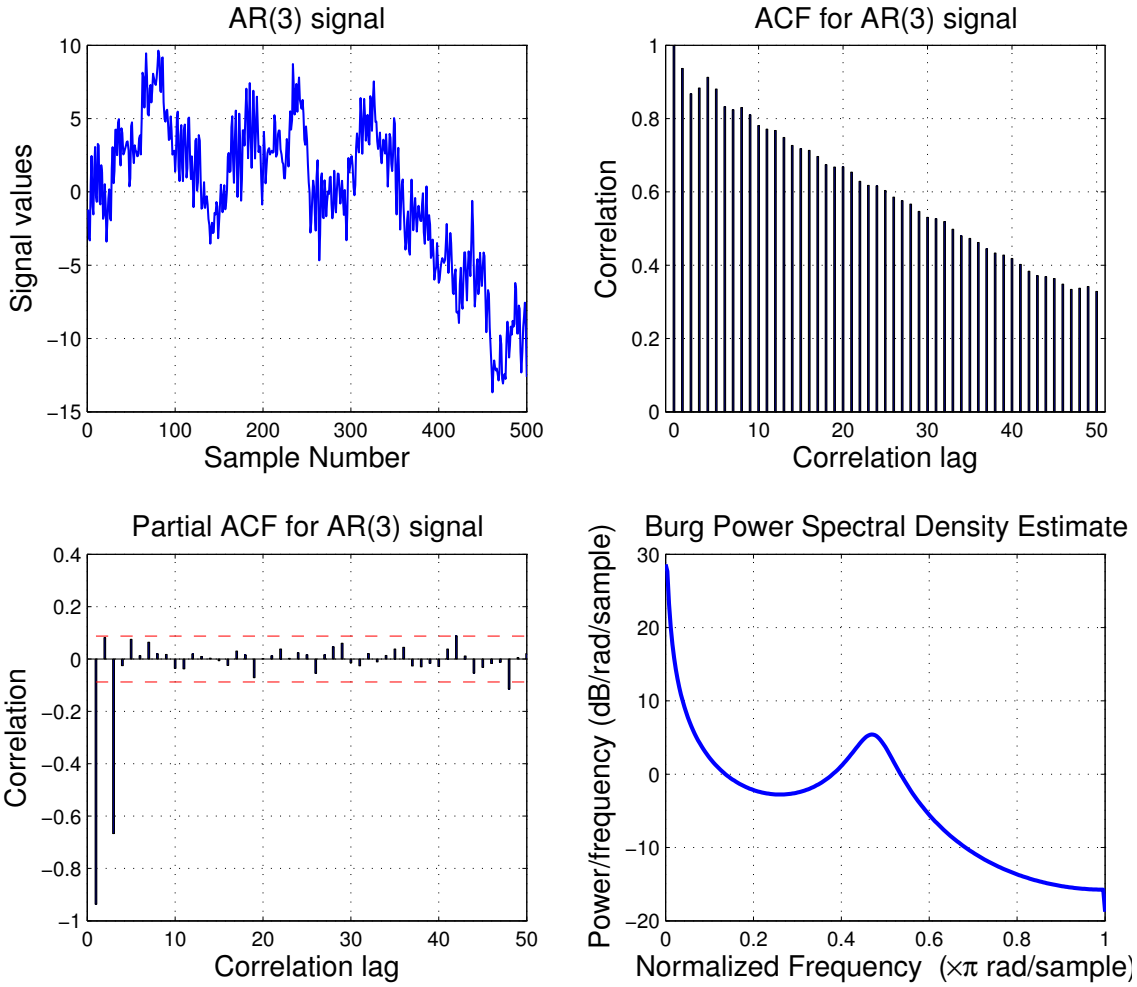
After $k = 2$ the partial correlation function (PAC) is very small, indicating $p = 2$



The broken red lines denote the 95% confidence interval which has the value $\pm 1.96/\sqrt{N}$, and where $PAC \approx 0$ (see Appendix 5)

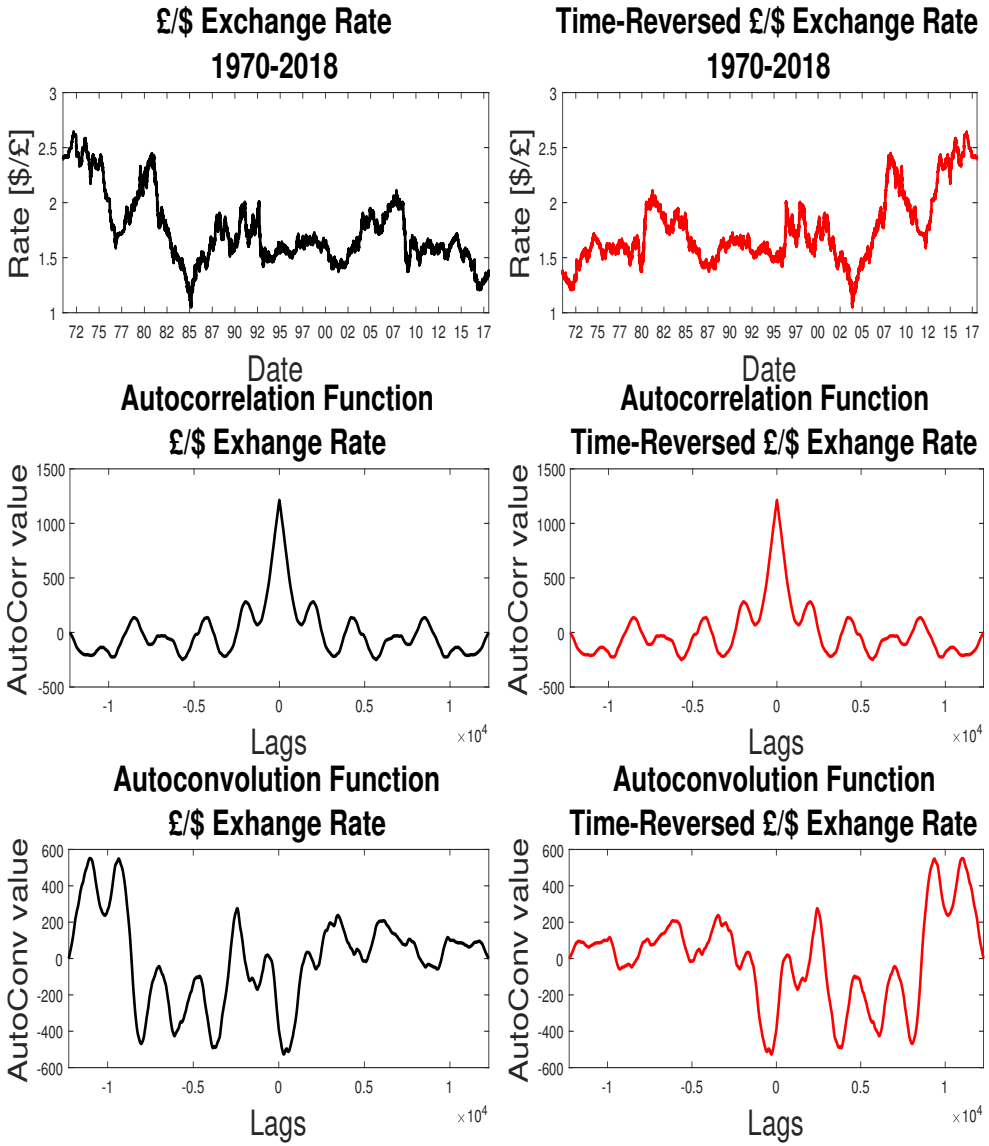
Example 14: Model order for an AR(3) process

An AR(3) process realisation, its ACF, and partial autocorrelation (PAC)



After lag $k = 3$, the PAC becomes very small (broken line $\rightsquigarrow$ conf. int.)

Example 15: The Partial Correlation view $\hookrightarrow$ model order of a financial time series (the 'correct' and 'time-reversed' time series)



Partial correlations:

AR(1): $\mathbf{a} = [0.9994]$

AR(2): $\mathbf{a} = [.9994, -.0354]$

AR(3): $\mathbf{a} = [.9994, -.0354, -.0024]$

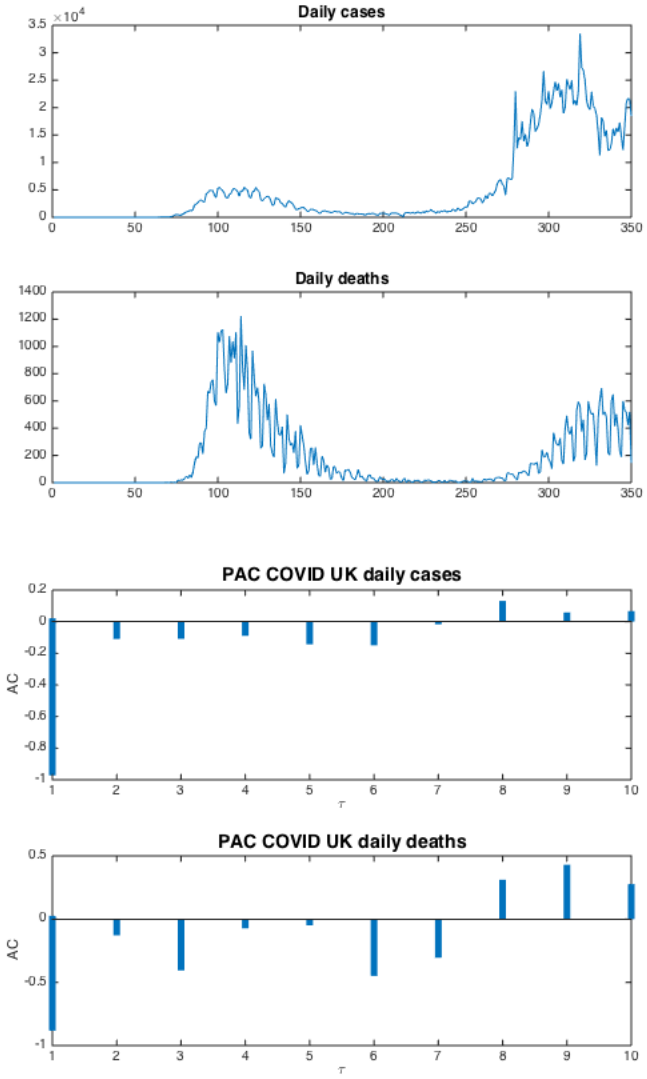
AR(4): $\mathbf{a} = [.9994, -.0354, -.0024, .0129]$

AR(5): $\mathbf{a} = [.9994, -.0354, -.0024, .0129, -.0129]$

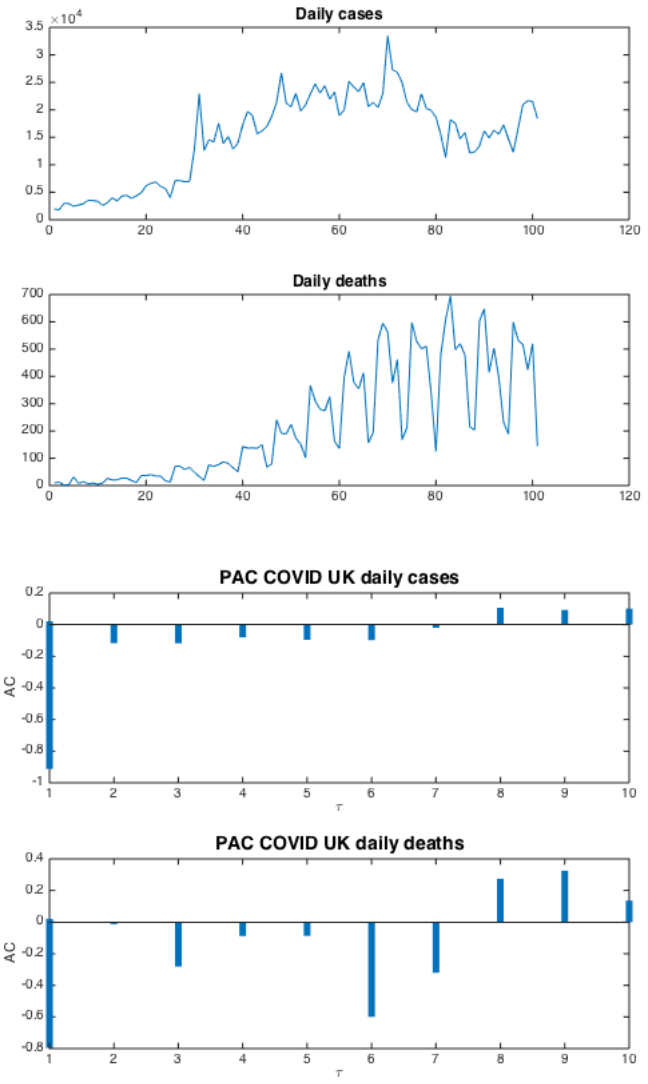
AR(6): $\mathbf{a} = [.9994, -.0354, -.0024, .0129, -.0129, -.0172]$

Example 16: ARMA(p,q) modelling of COVID-19 data?

COVID-19 time series in the UK



Second wave, UK COVID-19



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:

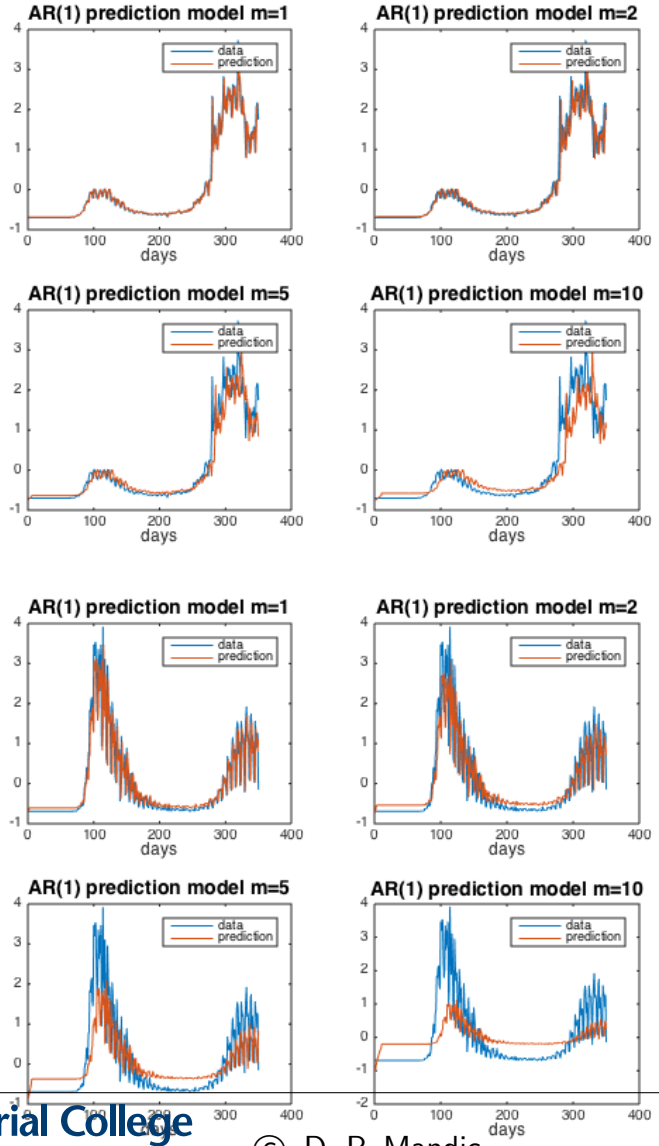
`ARYULE`

and functions

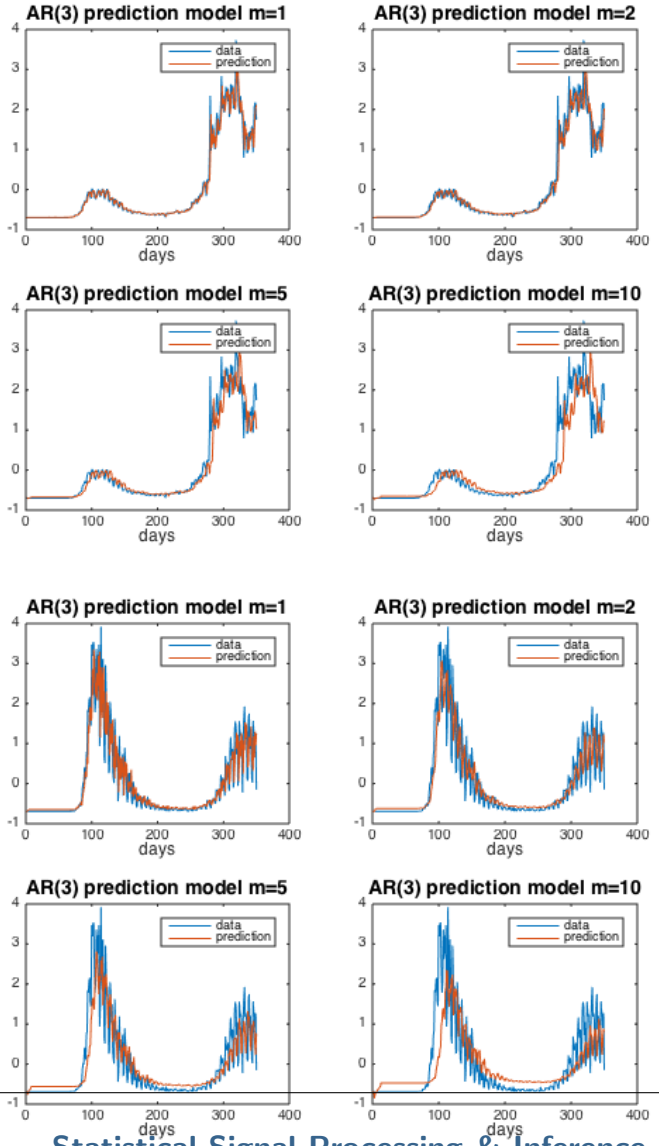
`PYULEAR, ARMCOV, ARBURG, ARCOV, LPC, PRONY`

Example 17: AR(1) and AR(3) prediction of COVID-19 data

AR(1) T: cases, B: Deaths



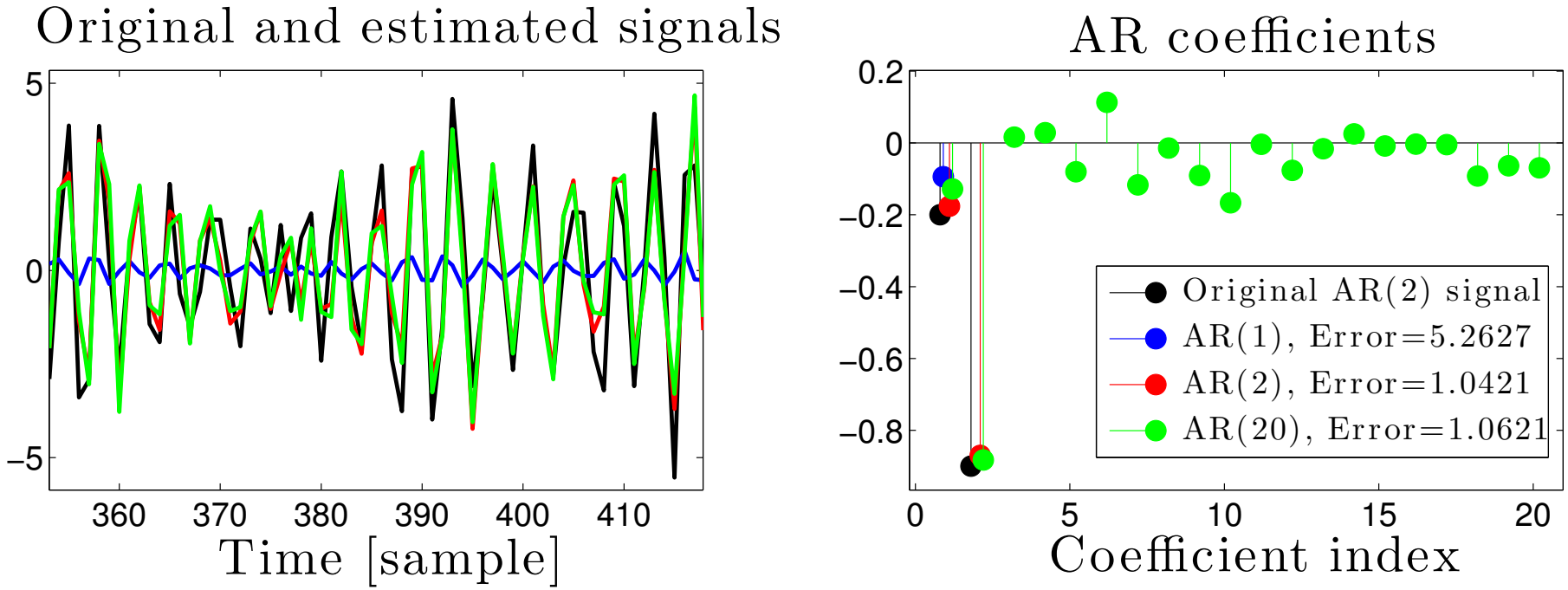
AR(3) T: cases, B: Deaths



Example 18: Under– vs Over–fitting a model ↬ Estimation of the parameters of an AR(2) process

Consider AR(2): $x[n] = -0.2x[n-1] - 0.9x[n-2] + w[n]$, $w[n] \sim \mathcal{N}(0,1)$

☞ We perform its prediction using AR(1), AR(2) and AR(20) models:



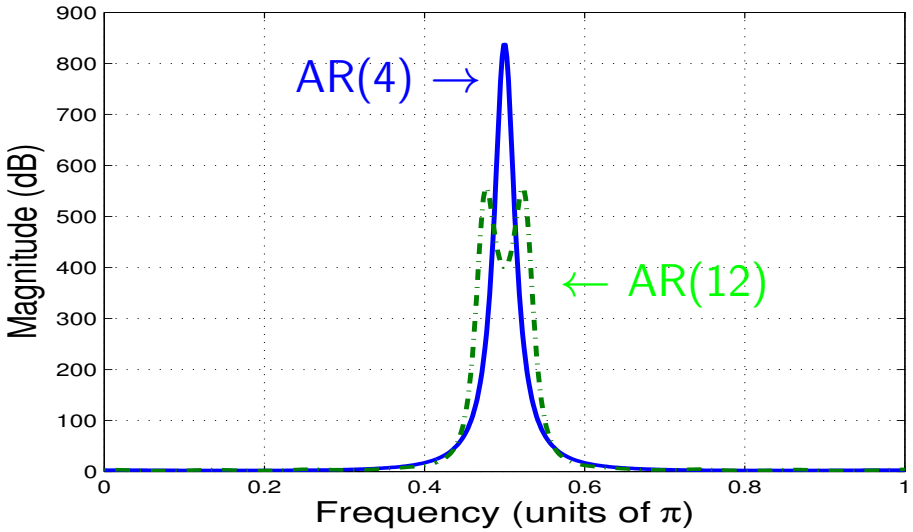
The *higher order* coefficients of the AR(20) model are close to zero and therefore do not contribute significantly to the estimate, while the AR(1) does not have sufficient degrees of freedom. (see also Appendix 3)

Effects of over-modelling on autoregressive spectra: Spectral line splitting

Consider an AR(2) signal

$$x(n) = -0.9x(n-2) + w(n) \text{ with } w \sim \mathcal{N}(0,1)$$

We have $N = 64$ data samples, and model orders $p = 4$ (solid blue) and $p = 12$ (broken green). AR_2_Highpass_Circularity.m



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 (see Appendix 7)

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 a "penalty" for high model orders. Some of the criteria for model order selection are:

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

AIC: The Akaike information criterion (AIC)

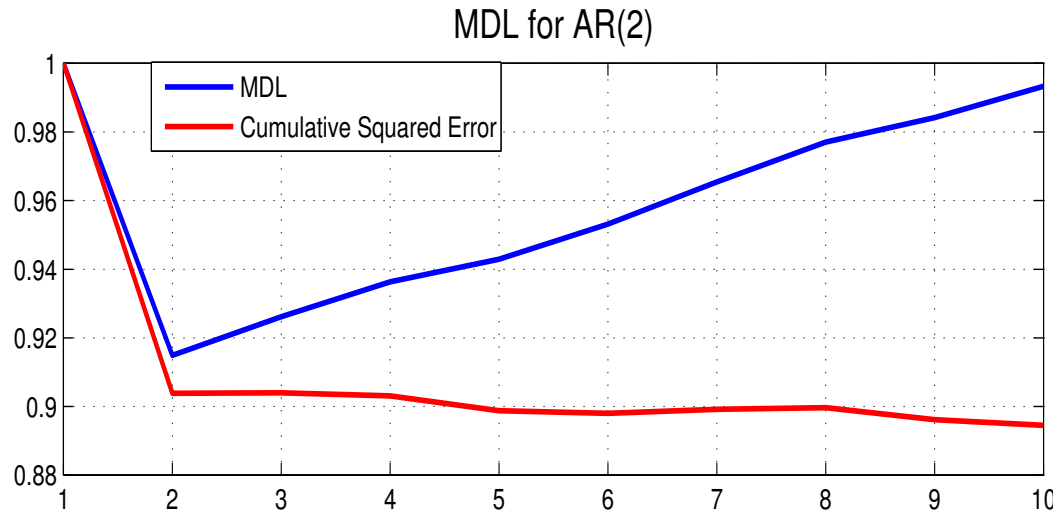
$$\textbf{MDL} \quad p_{opt} = \min_{p} \left[log(E) + \frac{p * log(N)}{N} \right]$$

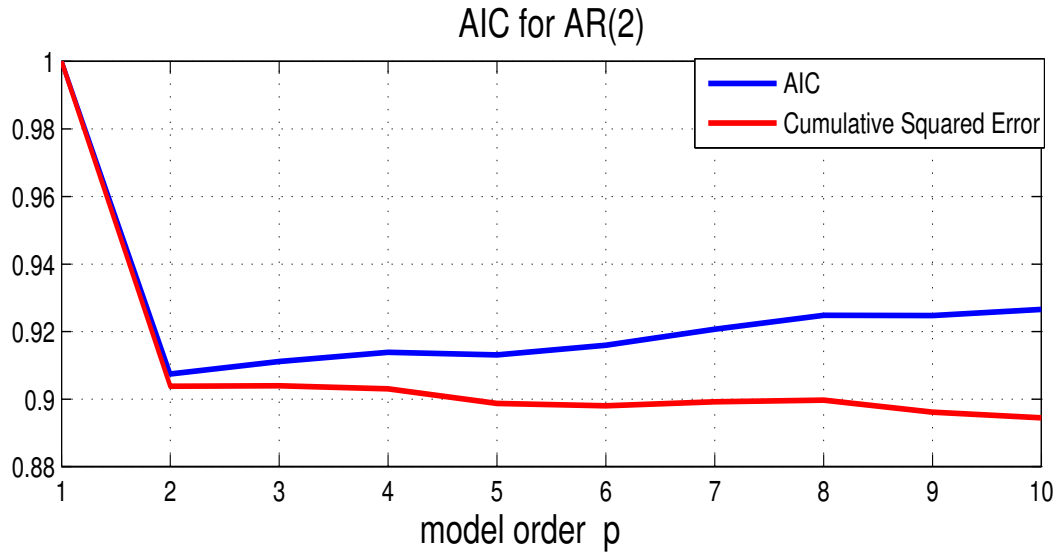
$$\textbf{AIC} \quad p_{opt} = \min_{p} \left[log(E) + 2p/N \right]$$

$E \rightsquigarrow$ 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 19: Model order selection ↬ 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 **(prediction 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/fitting)

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)

$$c_k = E\big[(w[n] + b_1 w[n-1] + \cdots + b_q w[n-q]) \times x[n-k]\big]$$

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

Hence the, for $k = 0$, the variance of the MA(q) process becomes

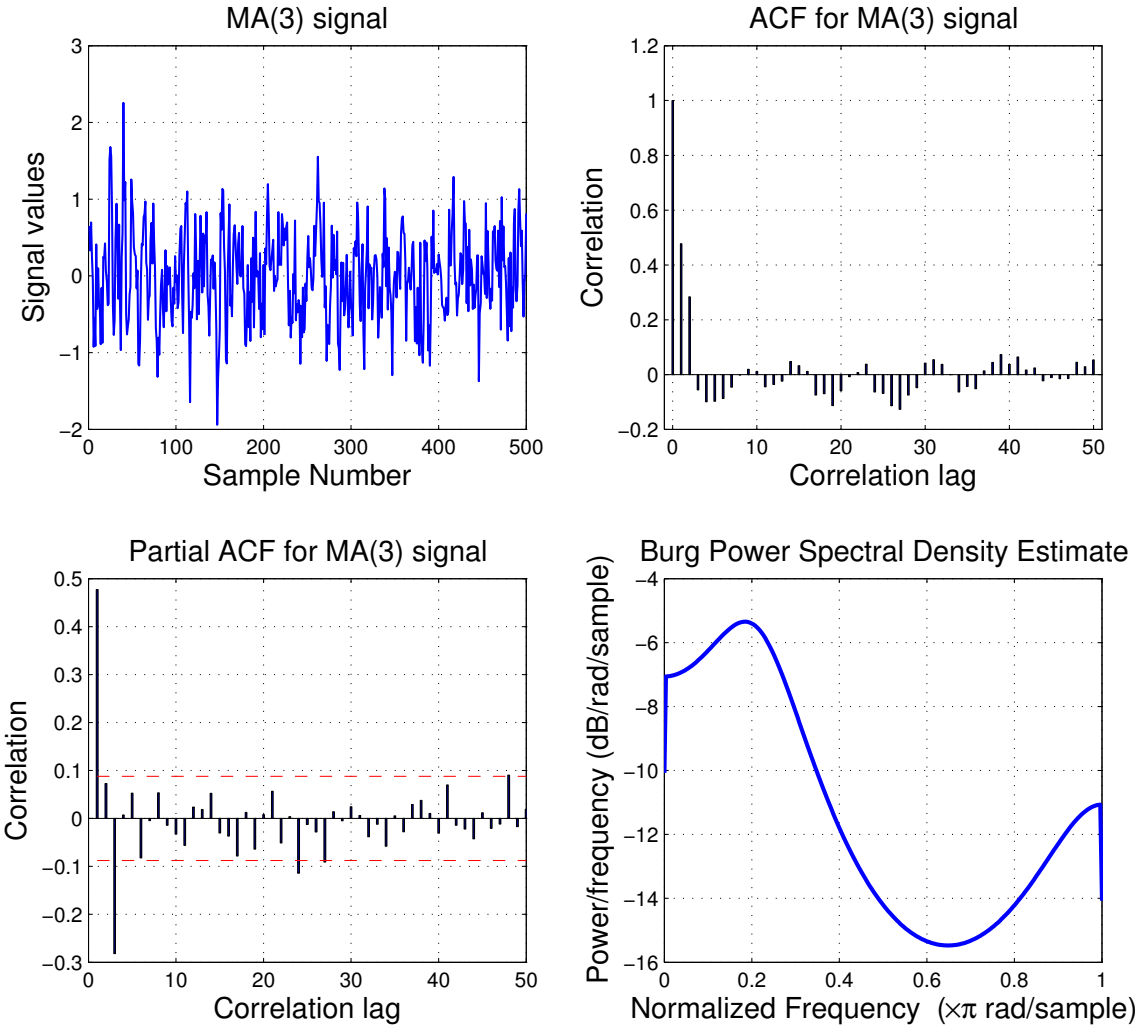
$$c_0 = (1 + b_1^2 + \cdots + b_q^2)\sigma_w^2$$

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

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

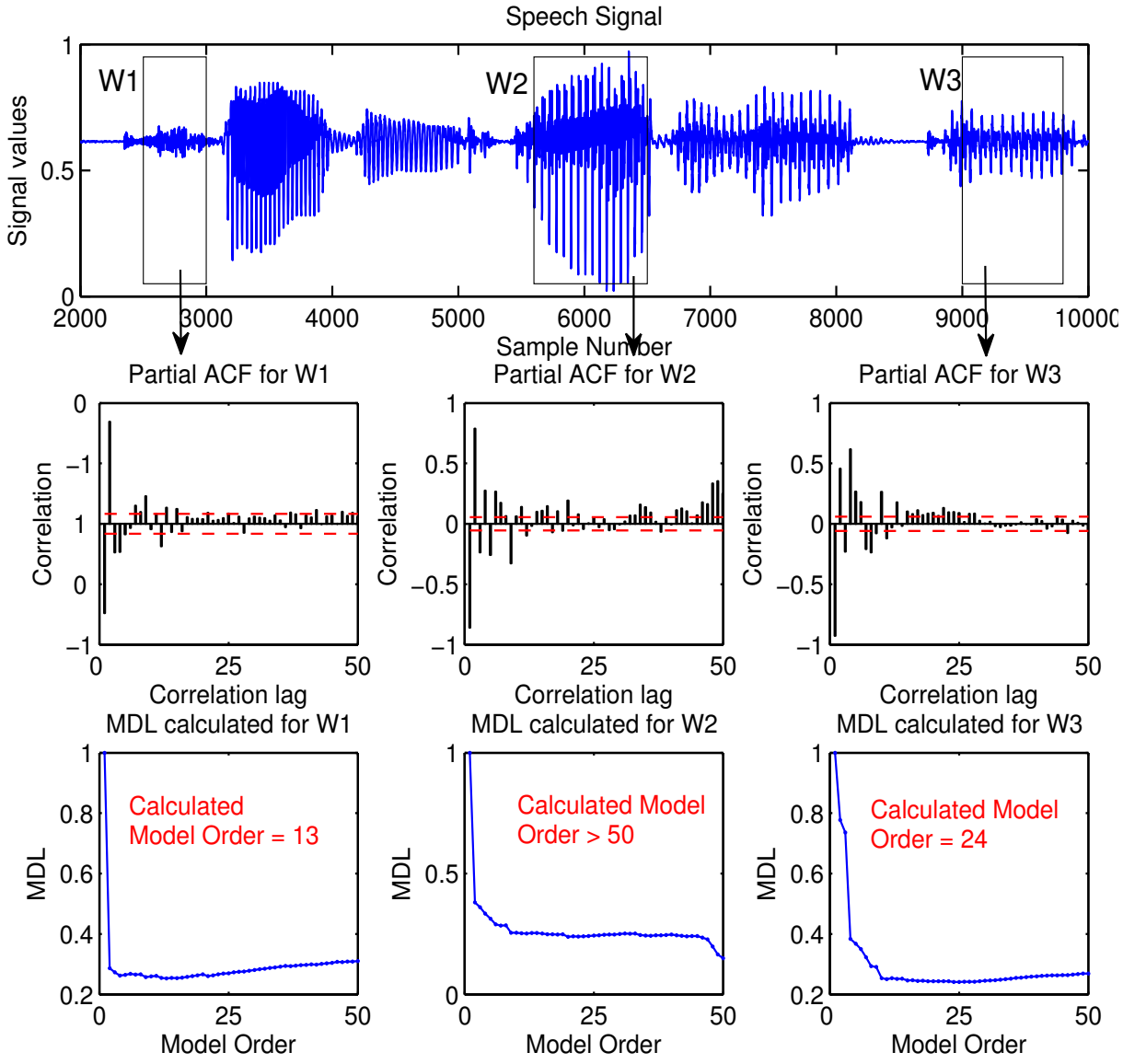
Example 20: Third order moving average MA(3) process

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



After lag $k = 3$, the PAC becomes very small (broken line $\rightsquigarrow$ conf. int.)

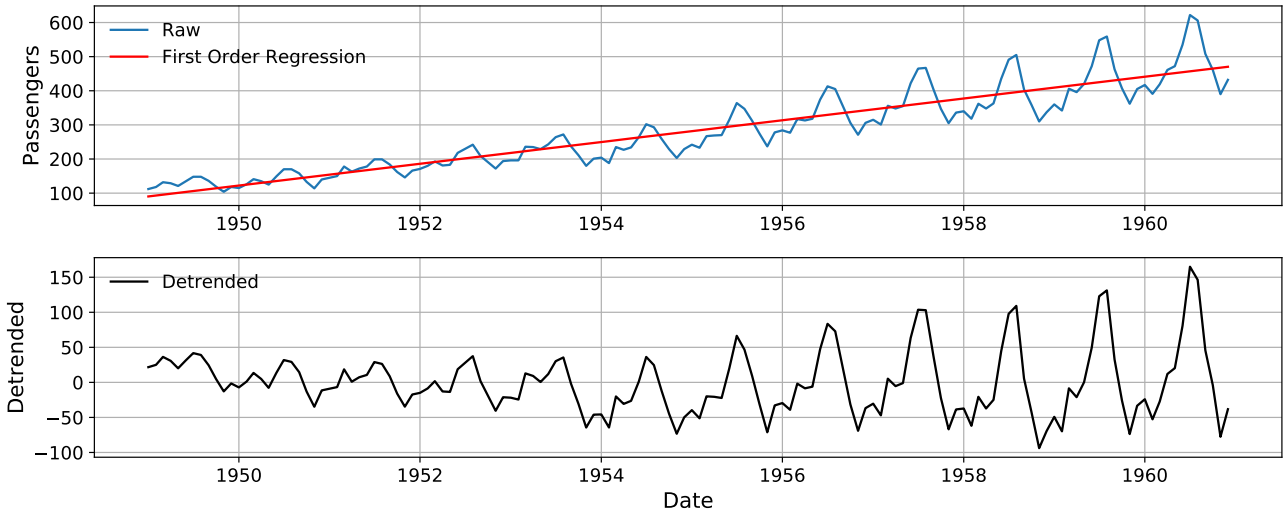
Example 21: Analysis of nonstationary signals

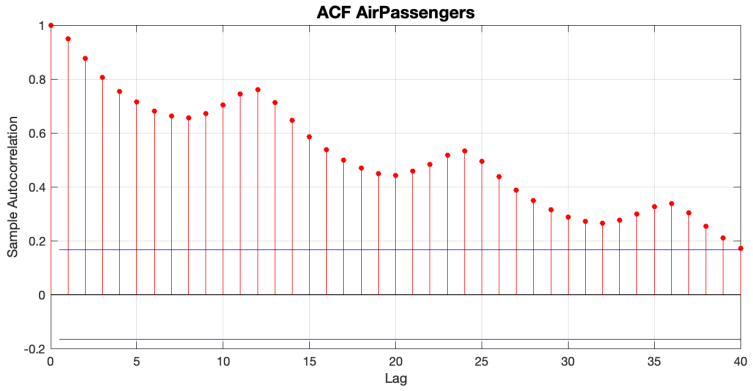


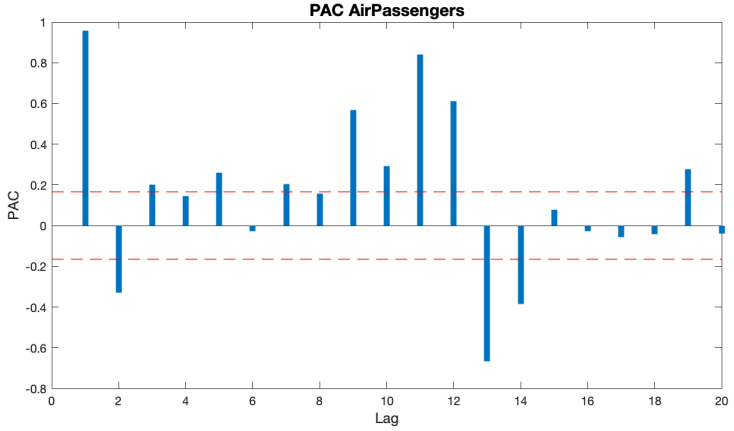
- Consider a real–world speech signal, and thee different segments with different statistical properties
- **Different AR model orders required for different segments of speech ↬ opportunity for content analysis!**
- To deal with nonstationarity we need short sliding data windows

Example 22: Problems with nonstationary data

The nonstat. air passengers time series has trend, cyclical and seasonal comp.







☞ This is reflected in the autocorrelation and PAC functions (trend, seasonal)

Dealing with nonstationarity in data: Autoregresive Integrated Moving Average models, ARIMA(p,d,q)

- ARMA models should be used when the data is stationary
- When data shows elements of non-stationarity, a generalisation of ARMA models may be used which accounts for nonstationarity, referred to as the **autoregressive integrated moving average** (ARIMA) model
- The form of ARIMA models is same as that of ARMA models, but with additional differencing of the input data in order to remove elements of non-stationarity (e.g. drifts or trends)
- This differencing corresponds to the "integrated" part of the model
- ARIMA(p,d,q) means: AR of order p, MA of order q, $d\times$ differentiation

$$y(n)=\sum_{i=1}^{p}a_i y(n-i)+\sum_{j=1}^{q}b_j w(n-j)+w(n)$$

☞ where $y(n)$ is the d-th difference of $x(n)$. Therefore,

- For $d=0$, we have $(y(n)=x(n)$ - For $d=1$, $y(n)=x(n)-x(n-1)$
- For $d=2$, that is, for an ARIMA(p,2,q) model, we have

$y(n)=[x(n)-x(n-1)]-[x(n-1)-x(n-2)]=x(n)-2x(n-1)+x(n-2)$

Example 22a: ARMA vs ARIMA, nonstationary data

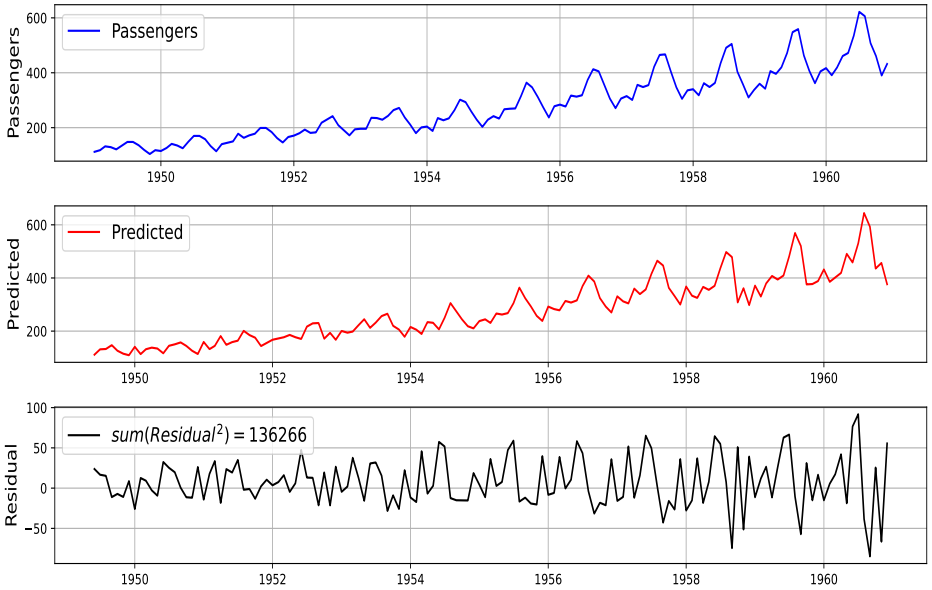
One-step prediction: ARMA(2,4) vs ARIMA(2,1,4) model, air passenger data

For ARIMA(2,1,4) modelling and subsequent prediction (inference), the non-statationary airline passenger time series was first differentiated as

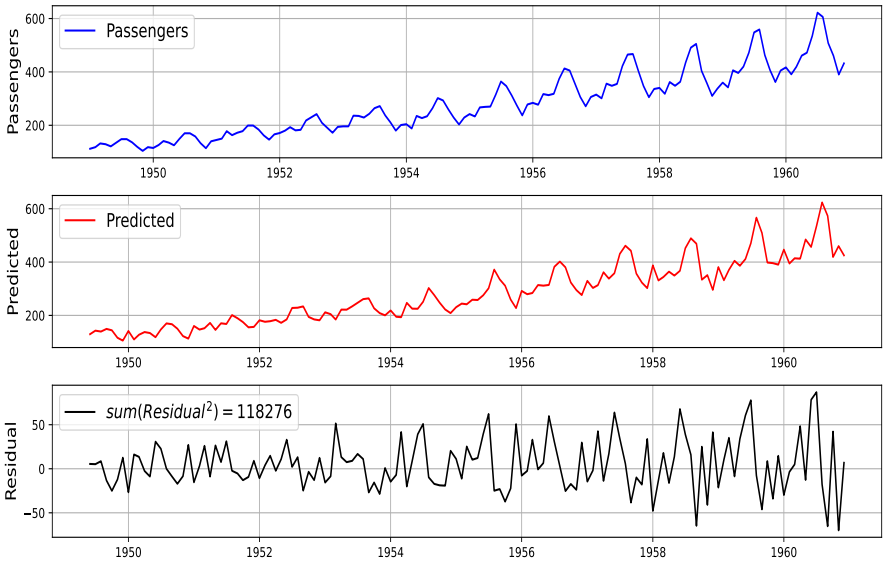
$$y(n) = x(n) - x(n-1) \qquad \text{for} \quad n = 1, \ldots, N-1$$

The ARIMA(2,1,4) model was then found, in the form

$$y(n) = a_1 y(n-1) + a_2 y(n-2) + b_1 w(n-1) + \cdots + b_4 w(n-4) + w(n)$$



ARMA$(2,4)$ one-step prediction



ARIMA$(2,1,4)$ one-step prediction

The ARIMA(2,1,4) model was able to deal better with the nonstationarity input, with $error^2 = 118\,k$ as opposed to $error^2 = 135\,k$ for ARMA(2,4).

Example 22b: ARMA vs ARIMA, nonstationary data

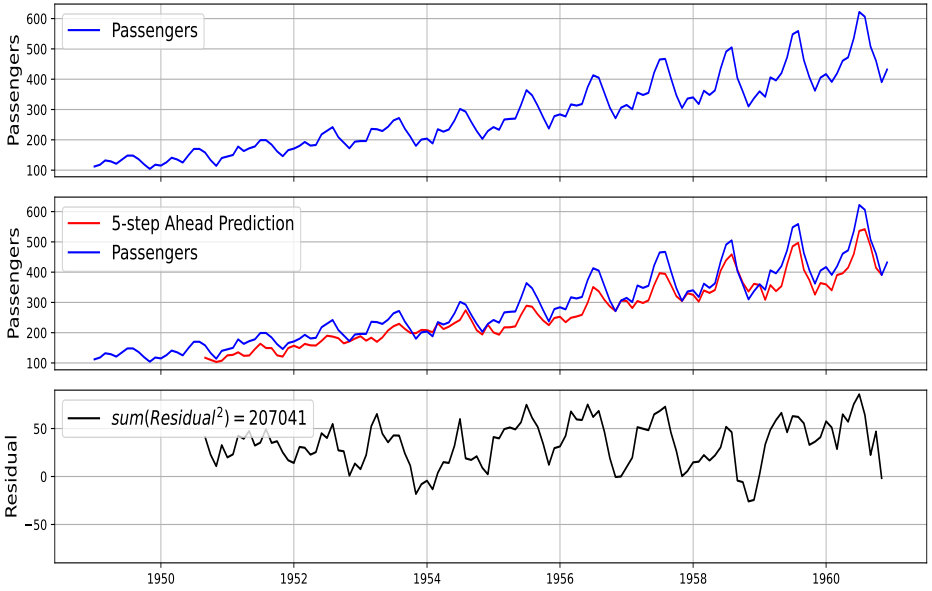
5-step ahead prediction: ARMA(12,4) versus ARIMA(12,1,4) model

For ARIMA(12,1,4) modelling and subsequent prediction (inference), the airline passenger time series was first differentiated as

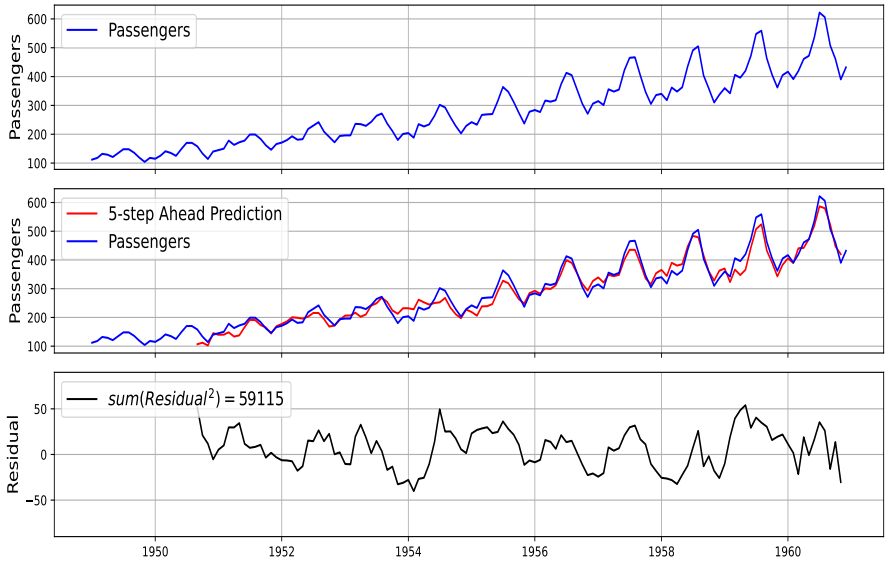
$$y(n) = x(n) - x(n-1) \qquad \text{for} \quad n = 1, \ldots, N-1$$

The ARIMA(12,1,4) model was then found, of the form

$$y(n) = a_1 y(n-1) + \cdots + a_{12} y(n-12) + b_1 w(n-1) + \cdots + b_4 w(n-4) + w(n)$$



ARMA$(12, 4)$, 5-step prediction



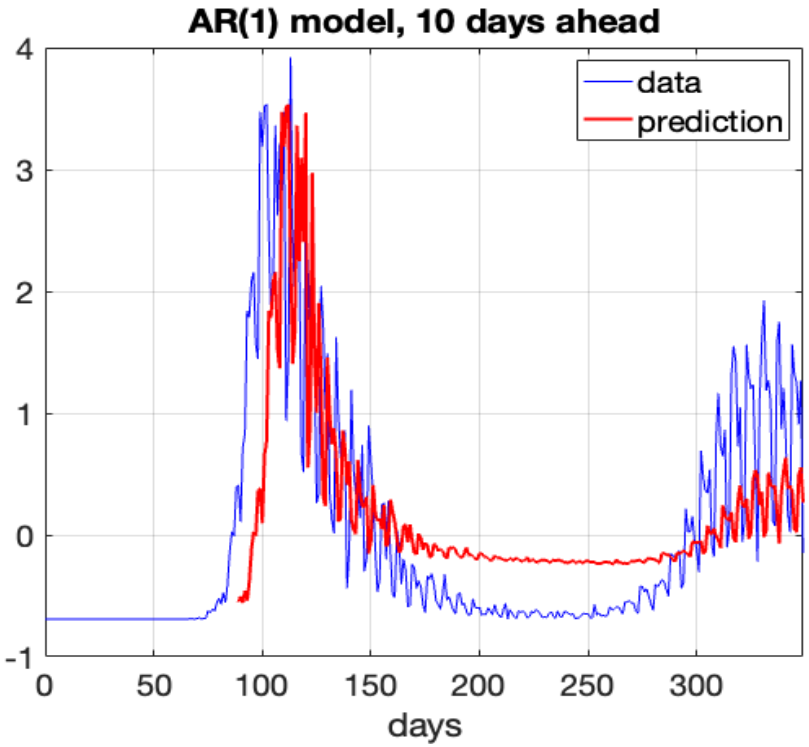
ARIMA$(12, 1, 4)$, 5-step prediction

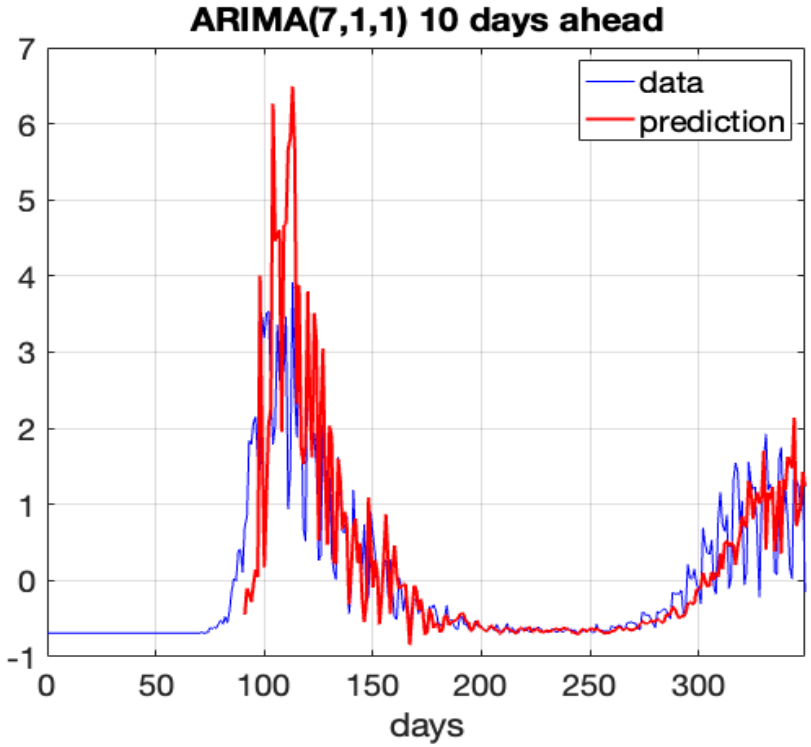
☞ The ARIMA(12,1,4) model yields much better inference than ARMA(12,4)

Example 22c: ARIMA prediction of COVID-19 data

(see Lecture 3 for the Bias–Variance trade–off)

Consider the prediction of COVID-19 death rates in the UK.



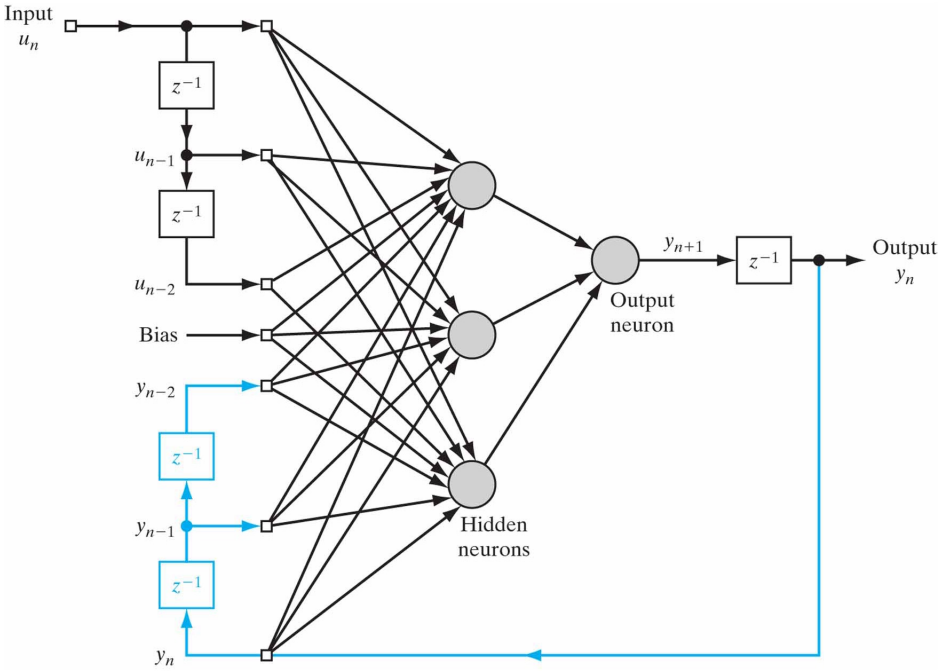


◦ The AR(1) prediction exhibits bias, as the mean of the predicted data (in red) is "off-set" from the mean of true data (in blue) for most of the plot

◦ The ARIMA(7,1,1) prediction is almost unbiased, and with similar variance as AR(1) prediction (which one do you prefer)

Nonlinear autoregressive models and Neural Networks

Nonlinear Autoregressive with Exogenous Inputs (NARX)



Recall the ARMA model

$$y(n) = a_1 y(n-1) + \cdots + a_p y(n-p) + w(n) + b_1 w(n-1) + \cdots + b_1 w(n-q)$$

This model provides two forms of geometric invariance: 1) scale invariance and 2) time translation $\not\hookrightarrow$ very useful in Neural Networks

The above NARMA(3,2) RNN has three hidden neurons and performs mapping $\hat{y}_{n+1} = \Phi(y_n, y_{n-1}, y_{n-2}, u_n, u_{n-1}, u_{n-2})$

Summary: AR and MA Processes

- A stationary AR(p) process can be represented as an infinite order MA process. A finite MA process has a dual infinite AR process.
- A finite MA(q) process has an ACF that is zero beyond lag q. For an AR process, the ACF is infinite in length and consists of mixture of damped exponentials and/or damped sinusoids.
- Finite MA processes are always stable, and there is no requirement on the coefficients of MA processes for stationarity. For invertibility, the roots of the characteristic equation must lie inside the unit circle.
- AR processes produce spectra with sharp peaks (two poles of $A(z)$ per peak), whereas MA processes cannot produce peaky spectra.
- For Vector Autoregressive (VAR) models, see Appendix 8.

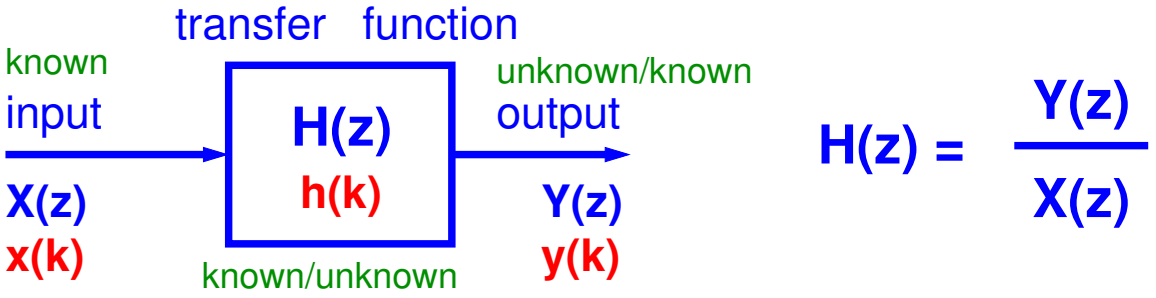
ARMA modelling is a classic technique which has found a tremendous number of practical applications.

Even Large Language Models (LLM) such as ChatGPT perform a form of auto-regression when generating new words (see Appendix 10).

Summary: Wold's decomposition, ARMA, ARIMA

- Every stationary time series can be represented as a sum of a perfectly predictable process and a feasible moving average process
- Two time series with the same Wold representations are the same, as the Wold representation is unique
- Since any MA process also has an ARMA representation, working with ARMA models is not an arbitrary choice but is physically justified
- The causality and stationarity on ARMA processes depend entirely on the AR parameters and not on the MA parameters
- An MA process is not uniquely determined by its ACF
- An AR(p) process is always invertible, even if it is not stationary
- An MA(q) process is always stationary, even if it is non-invertible
- **For non-stationary data we may employ ARIMA(p,d,q) models**

Recap: Linear systems



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

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

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

The next two slides show how to calculate the power of the output, $y(n)$.

Recap: Linear systems – statistical properties $\hookrightarrow$ mean and variance

i) **Mean**

$$E\{y[n]\} = E\left\{\sum_{r=-\infty}^{\infty} h(r)x[n-r]\right\} = \sum_{r=-\infty}^{\infty} h(r)E\{x[n-r]\}$$

$$\Rightarrow \quad \mu_y = \mu_x \sum_{r=-\infty}^{\infty} h(r) = \mu_x H(0)$$

[NB: $H(\theta) = \sum_{r=-\infty}^{\infty} h(r)e^{-jr\theta}$. For $\theta = 0$, then $H(0) = \sum_{r=-\infty}^{\infty} h(r)$]

ii) **Cross–correlation**

$$r_{yx}(m) = E\{y[n]x[n+m]\} = \sum_{r=-\infty}^{\infty} h(r)E\{x[n-r]x[n+m]\}$$

$$= \sum_{r=-\infty}^{\infty} h(r)r_{xx}(m-r) \quad \textbf{convolution of input ACF and } \{\mathbf{h}\}$$

$\Rightarrow$ **Cross-power spectrum** $S_{yx}(f) = \mathcal{F}(r_{yx}) = S_{xx}(f)H(f)$

Recap: Lin. systems – statistical properties $\rightarrow$ output

These are key properties $\rightarrow$ used in AR spectrum estimation

From $r_{xy}(m) = r_{yx}(-m)$ we have
$r_{xy}(m) = \sum_{r=-\infty}^{\infty} h(r) r_{xx}(m-r)$. Now we write

$$r_{yy}(m) = E\{y[n]y[n+m]\} = \sum_{r=-\infty}^{\infty} h(r) E\{x[n-r]y[n+m]\}$$

$$= \sum_{r=-\infty}^{\infty} h(r) r_{xy}(m+r) = \sum_{r=-\infty}^{\infty} h(-r) r_{xy}(m-r)$$

by taking Fourier transforms we have

$$\begin{aligned} S_{xy}(f) &= S_{xx}(f)H(f) \\ S_{yy}(f) &= S_{xy}(f)H(-f) \quad \rightsquigarrow \quad \text{function of} \quad r_{xx} \end{aligned}$$

or

$$\mathbf{S_{yy}(f) = H(f)H(-f)S_{xx}(f) = |H(f)|^2 S_{xx}(f)}$$

Output power spectrum = input power spectrum $\times$ squared transfer function

More on Wold Decomposition (Representation) Theorem

Example: A "paradox", can we talk about a deterministic random process

Consider a stochastic process given by

$$x[n] = A\cos[n] + B\sin[n]$$

where A, B $\in \mathcal{N}(0, \sigma^2)$ and A is independent of B (A and B are independent normal random variables).

This process is deterministic because it can be written as

$$x[n] = \frac{\sin(2)}{\sin(1)}\, x[n-1] - x[n-2]$$

that is, based on the history of $x[n]$. Therefore

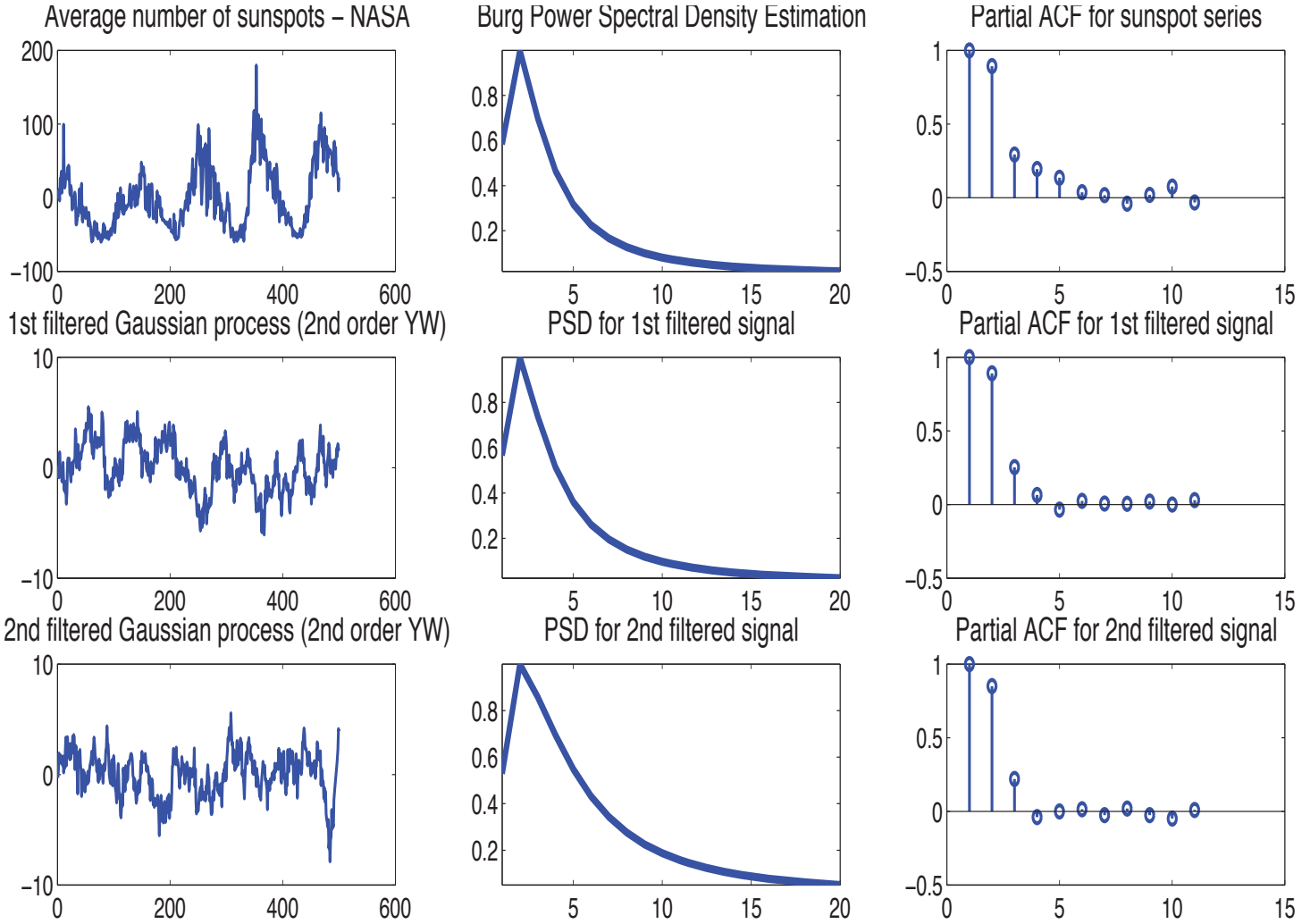
$$p(x[n] \,|\, x[n-1], x[n-2], \ldots) = \frac{\sin(2)}{\sin(1)}\, x[n-1] - x[n-2] = x[n]$$

Remember: Deterministic does not mean that $x[n]$ is non-random

Appendix 1: Sunspot numbers (recorded since 1874)

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

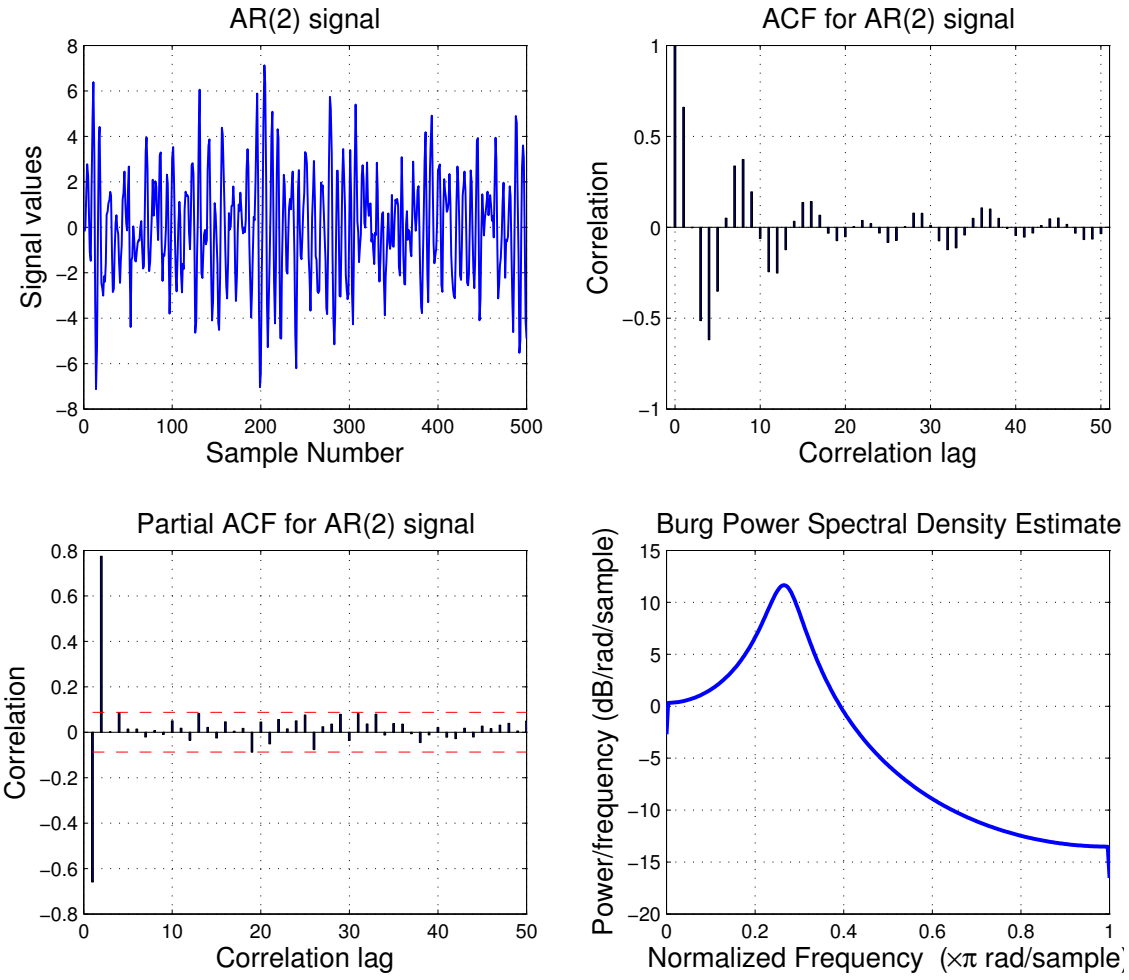
Left: time Middle:spectrum Right: autocorrelation



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 (broken line $\rightsquigarrow$ conf. int.)

Appendix 3: Obtaining the ACF of a general AR(p) process

Consider the AR(p) process, given by

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

To obtain the autocorrelation function of this AR process, multiply the above equation by $x[n-k]$ to obtain (recall that $r(-m) = r(m)$)

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

Apply the statistical expectation operator (the coefficients a_i go in front)

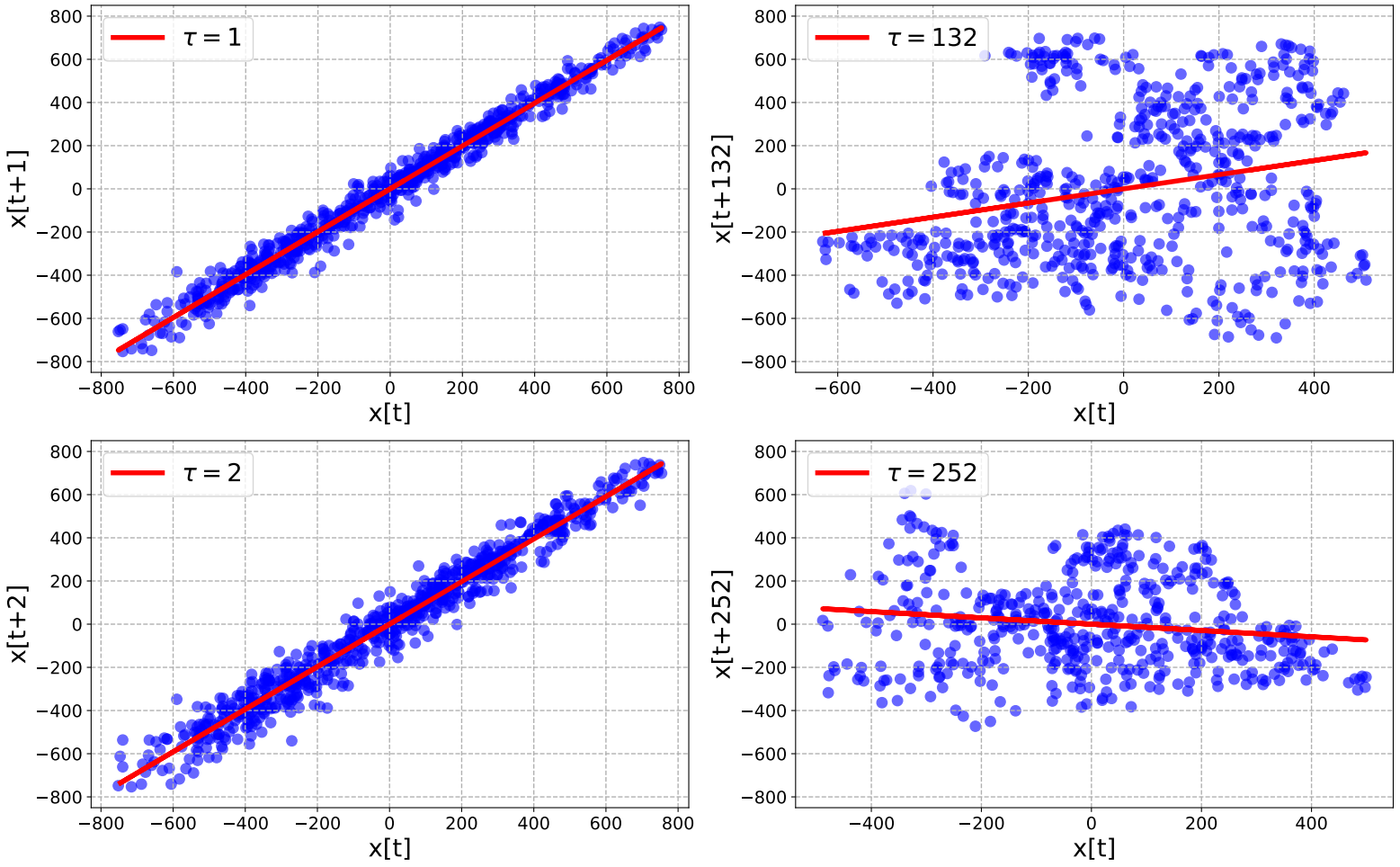
$$\underbrace{E\{x[n-k]x[n]\}}_{r_{xx}(k)} = a_1 \underbrace{E\{x[n-k]x[n-1]\}}_{r_{xx}(k-1)} + a_2 \underbrace{E\{x[n-k]x[n-2]\}}_{r_{xx}(k-2)} + \cdots + a_p \underbrace{E\{x[n-k]x[n-p]\}}_{r_{xx}(k-p)} + \underbrace{E\{x[n-k]w[n]\}}_{r_{xw}(k)=0}$$

☞ $r_{xw}(k) = 0$ since $x[n-k] = a_1 x[n-k-1] + \cdots + a_p x[n-k-p] + w[n-k]$.

As $w[n-x] \perp w[n]$, then $E\{x[n-k]w[n]\}$ vanishes for $k > 0$, to give

$$r_{xx}(k) = a_1 r_{xx}(k-1) + a_2 r_{xx}(k-2) + \cdots + a_p r_{xx}(k-p), \qquad k > 0$$

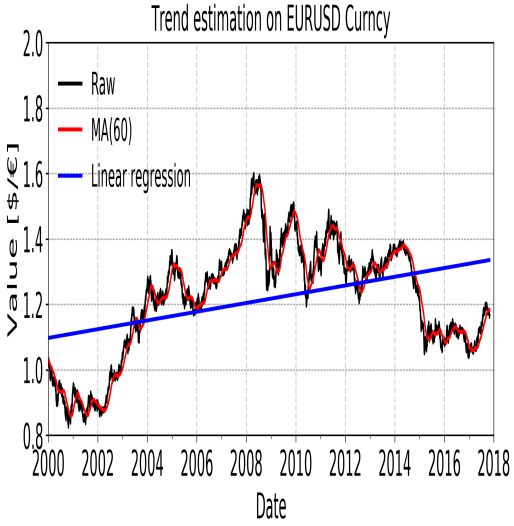
Appendix 4a): Scatter plots of the detrended S&P 500 financial index

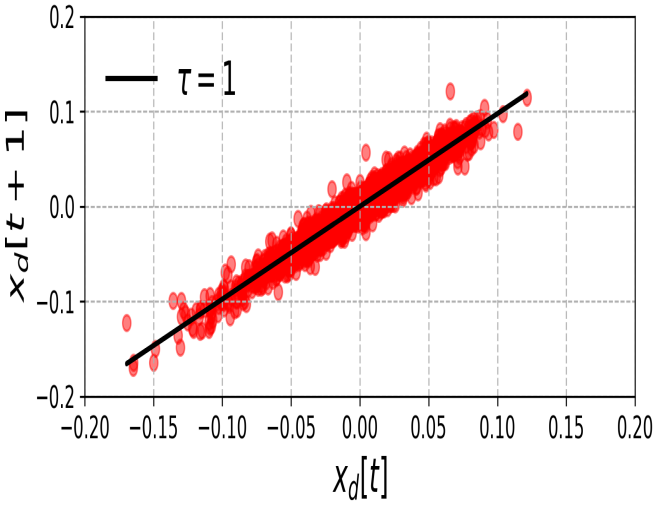


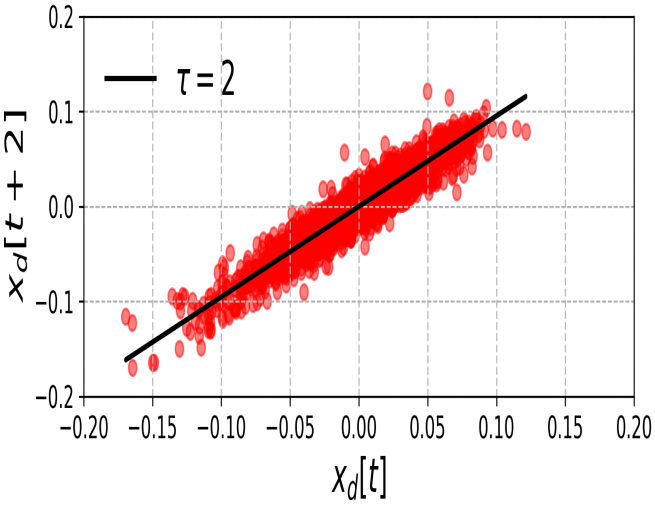
The detrended S&P 500 time series shows strong correlations for small lags in the scatter plot.

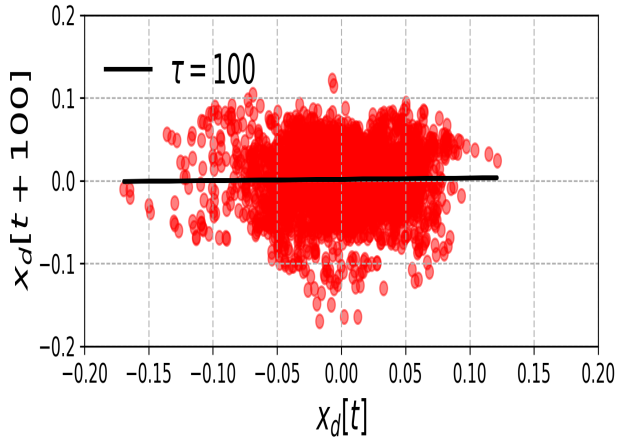
Appendix 4b): Euro vs USD currency exchange

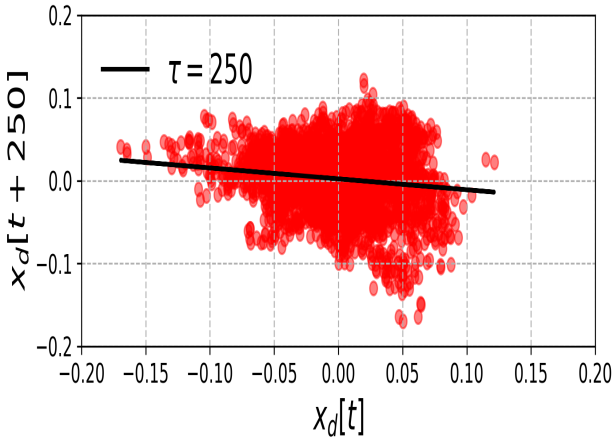
Scatter plots of a detrended EUR/USD exchange rate vs its τ days lagged version

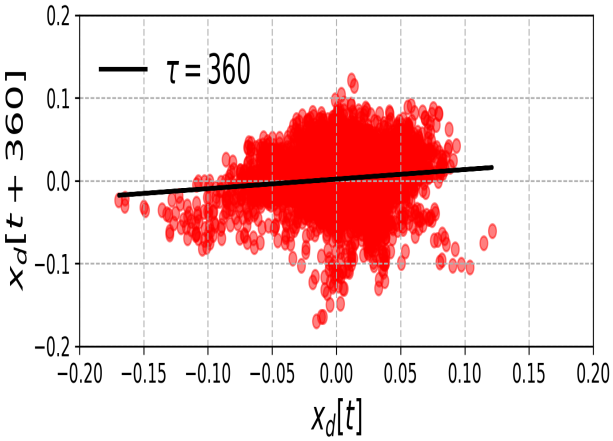












Appendix 5: More on the Partial Autocorrelation Function (PACF)

The PACF of a stationary process is a vector, $\boldsymbol{\pi}$, defined as

$$\pi(k) = \begin{cases} \pi(0) = 1 \\ \pi(k) = a_{kk}, \quad \text{for} \quad k \geq 1 \end{cases}$$

where a_{kk} is the last component of $\mathbf{a}_k = [a_{k1}, a_{k2}, \ldots, a_{kk}]^T, k = 1, 2, \ldots p$ which are calculated from $\mathbf{a}_k = \mathbf{R}_k^{-1}\mathbf{r}_k$ (Yule-Walker, see Slide 23)

☞ It is possible to show that its value $\pi(k) = a_{kk}, \text{for} \quad k \geq 1$ is equivalent to the correlation coefficient between the residuals of the regressions

$$x(n) - \hat{x}(n) = x(n) - E\{x(n) \,|\, x(n-1), \ldots, x(n-k+1)\} \quad \text{and}$$

$$x(n-k) - \hat{x}(n-k) = x(n-k) - E\{x(n-k) \,|\, x(n-k+1), \ldots, x(n-1)\}$$

☞ $\pi(k)$ (or equally the AR coefficient a_{kk}) **measures the linear dependence** between $x(n)$ and $x(n-k)$, **once we have removed the influence of** $x_{n-1}, \ldots, x_{n-k+1}$, i.e. $a_{kk} = corr\big(x(n) - \hat{x}(n), x(n-k) - \hat{x}(n-k)\big)$.

Appendix 5: Confidence intervals for PACFs (intuition)

Quenouille (1949) showed that on the hypothesis that the process is AR(p), the estimated partial autocorrelations of order $p+1$, and higher, are approximately independently and normally distributed with zero mean.

With N observations, we then have $var(\hat{a}_{kk}) \approx 1/N, \quad k \geq p+1.$

Thus, the standard error (SE) of the estimated PAC $\hat{a}_{kk}$ is

$$SE(\hat{a}_{kk}) = \hat{\sigma}(\hat{a}_{kk}) \approx 1/\sqrt{N}, \quad k \geq p+1.$$

Intuition: Let us establish whether a time series, $\{x_1, \ldots, x_N\}$, is an independent identically distributed process, that is, $x \sim i.i.d.(0, \sigma^2)$

To achieve this, we need a decision rule, for example

Reject the null hypothesis $H_0 : \rho_k = 0$ if $|\hat{\rho}_k| > c$, with c a constant.

Constant c is a threshold, arising e.g. from a statistical significance test

$$P(|\hat{\rho}_k| > c|H_0) = 0.05 \quad \Rightarrow \quad P(|\hat{\rho}_k| > c|H_0) = 1 - P(|\hat{\rho}_k| \leq c|H_0) = 0.05$$

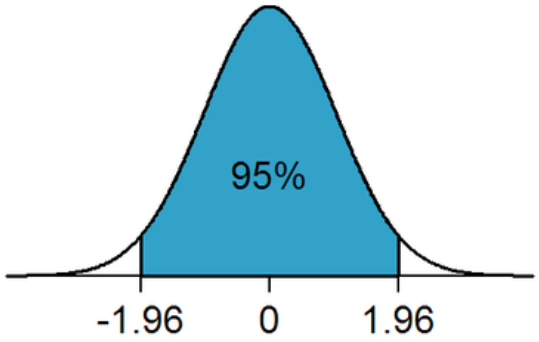
This implies that $P(|\hat{\rho}_k| \leq c|H_0) = P(-c\sqrt{N} \leq \sqrt{N}\hat{\rho}_k \leq c\sqrt{N}) = 0.95$

Appendix 5: Confidence intervals for PACFs (intuition)

If x_n is $i.i.d.(0, \sigma^2)$, then $\sqrt{N}\hat{\rho}_k \to \mathcal{N}(0,1)$ and, for large N, a normal distribution is a good approximation to the true distribution of $\sqrt{N}\hat{\rho}_k$. s.t.

$$P(-c\sqrt{N} \leq \sqrt{N}\hat{\rho}_k \leq c\sqrt{N}) = 0.95 \quad \text{iff} \quad c\sqrt{N} = 1.96 \quad \Rightarrow \quad c = \frac{1.96}{\sqrt{N}}$$

We can **reject** H_0 **if** $|\hat{\rho}_k| > 1.96/\sqrt{N}$ **that is, if** $\hat{\rho}_k \notin [-1.96/\sqrt{N}, 1.96/\sqrt{N}]$



In a two-tailed test, the rejection region for a significance level of 0.05 is at both ends of the distribution (in our case Gaussian), and amounts to up to 5% of the area under the curve (white regions).

☞ If the data, $x_1, \ldots, x_N$ were indeed generated by an $i.i.d.$ process, then $\approx$ 95% of sample ACFs, $\hat{\rho}_1, \ldots, \hat{\rho}_n$, should be within the bounds $\pm 1.96/\sqrt{N}$.

In other words, about 5% of the sample correlations should be outside the broken red lines in the PACF plots. For example, if 20 values of $\hat{\rho}_k$ are calculated, then only one of its values should lie outside these limits.

Appendix 5: Confidence intervals for PACFs (intuition)

☞ The PACF basically finds correlation between the residuals at time n and time $n-k$ (after removing the effects which are already explained by the earlier lags).

This is why it is called "partial" as the already found variations are removed before calculating the next correlation.

If there is any hidden information left in the residual, we might have a good correlation at the next lag, so we keep exploring along the lags.

☞ Too many correlated features are not desirable, as this can create collinearity issues $\not\hookrightarrow$ we should retain only the relevant features.

The null hypothesis is the "default" assumption that there has been no change in statistical behaviour.
To determine whether a result is statistically significant, we calculate a p-value, which is the probability of a more extreme statistical behaviour given that the null hypothesis is true.
The null hypothesis is rejected if the p-value is less than (or equal to) a predetermined level – the significance level – which is usually set at 5%.

Appendix 6: A note on over–parametrisation

Consider the linear stochastic process given by

$$x[n] = x[n-1] - 0.16x[n-2] + w[n] - 0.8w[n-1]$$

It clearly has an ARMA(2,1) form. Consider now its coefficient vectors written as polynomials in the z–domain

$$a(z) = 1 - z^{-1} + 0.16z^{-2} = (1 - 0.8z^{-1})(1 - 0.2z^{-1})$$

$$b(z) = 1 - 0.8z^{-1}$$

These polynomials have a common factor $(1 - 0.8z^{-1})$, and therefore after cancelling these terms, we have the resulting lower–order polynomials

$$a(z) = 1 - 0.2z^{-1}$$

$$b(z) = 1$$

The above process is therefore an AR(1) process, given by

$$x[n] = 0.2x[n-1] + w[n]$$

and the original ARMA(2,1) version was over–parametrised.

Appendix 7: More on model order selection

A criterion is said to be **consistent** if the correct model is chosen with probability one as the number of data points asymptotically approaches ∞.

- MDL is consistent whereas AIC is not.
- Hannan and Quinn (1979) proposed the Hannan-Quinn information criterion (HQC) as a means of improving the consistency of AIC.
- Small-sample properties of AIC lead to over-estimating the model order. Hurvich and Tsai (1989) derived a 'corrected' AIC, referred to as AICc, in order to compensate for the small-sample over-fitting.

More detail in e.g. "Regression and Time Series Model Selection" by McQuarrie and Tsai.

$$\text{MDL} = \log E_p + \frac{p \log N}{N} \qquad \text{AIC} = \log E_p + \frac{2p}{N} \tag{1}$$

$$\text{HQC} = \log E_p + \frac{2p \log\log N}{N} \qquad \text{AICc} = AIC + \frac{2p(p+1)}{N-p-1} \tag{2}$$

where p is the model order, E_p is the loss function for the model with p parameters and N is the number of estimated data points.

Appendix 7: More on model order selection

The results below show that the AIC_c criterion was able to identify the most parsimonious model order of $p = 2$.

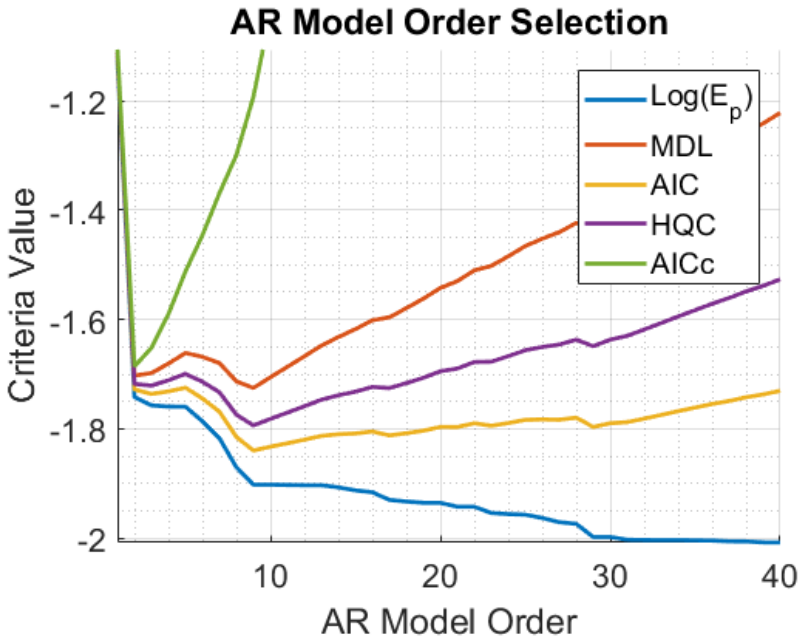


Figure 1: Information criteria for AR model order selection, with cumulative squared error as the loss function. A short segment of an AR(2) process was considered, which affected the reliability of these information criteria.

Appendix 8: Vector Autoregressive models (VAR)

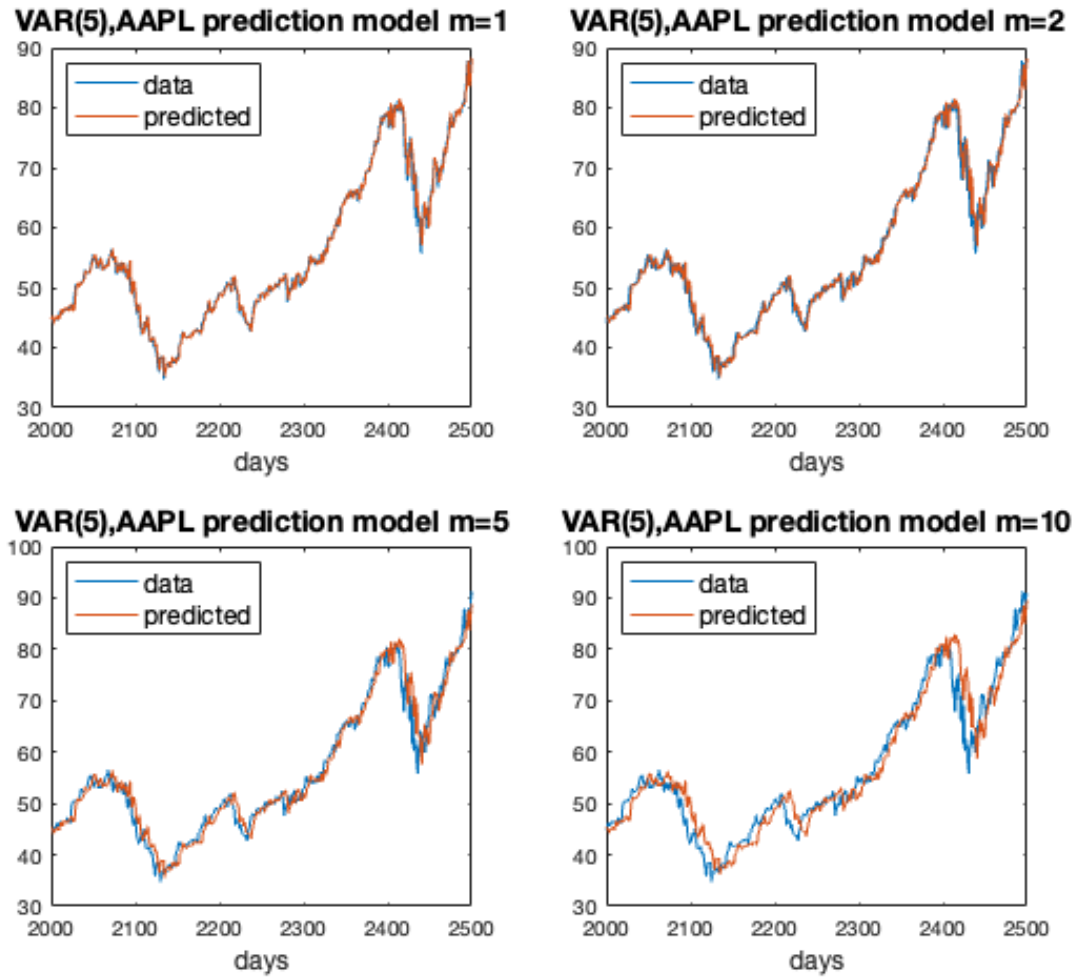
- The multivariate, also called Vector Autoregressive (VAR), processes generalise the standard AR (and ARMA) models.
- This allows us to make inference from multiple data channels together
- The quantities $\mathbf{x}$, $\mathbf{a}$ and $\mathbf{w}$ now become matrices, so e.g. the VAR(1) process can be expressed as

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

- Something to think about: Would the inverse of the multichannel correlation matrix depend on 'how similar' the data channels are; Explain this 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 are the advantages of such a model?
- How would you express an AR(p) process as a state-space model; What kind of the transition matrix between the states would you have?

Appendix 8:Multivariate inference often helps

For a rigorous account of multivariate inference, see Lecture 4



Apple stock prediction using a vector autoregressive VAR(5) model (Apple as one variate and 5 other stocks from S&P 500 as other variates)

Appendix 9: From stochastic Autoregression to Autoregressive Generative AI

Recall: The term 'autoregressive' originates from the field of time-series forecasting, where future predictions are based on the past observations.

Such a "sequence prediction" has been routinely used in e.g. natural language processing (NLP).

☞ Autoregressive generative models are much more complex, as e.g. even a standard image of 1,000 $\times$ 1,000 pixels has a whopping 10^6 pixels!

Basis of Autoregressive Generative models

For an n-dimensional dataset to learn from, the joint distribution of data is

$$p(x_0, x_1, \ldots, x_{n-1}) = p(x_0)p(x_1|x_0)p(x_2|x_1, x_0) \cdots p(x_{n-1}|x_{n-2}, \ldots, x_1, x_0)$$

This precisely depicts the operation of large autoregressive models (no assumption of conditional independence between variables).

Current Deep Autoregressive Generative models (2025) include Pixel CNN, Pixel RNN, Character CNN, Character RNN, Wave–Net.

Pro's and Con's: **Pos:** Intuitive, well understood supervised learning process; **Neg:** Needs ordering of random variables, sequential generation

Consider also: Fourier transform as a filtering operation

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 $F(\omega) = \int_{-\infty}^{\infty} x(t)e^{-\jmath\omega t}dt$

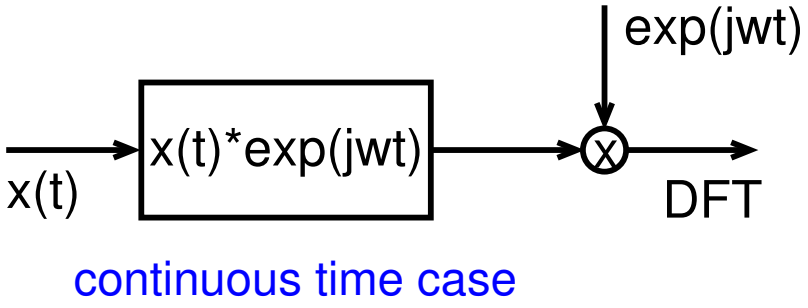
Let us now swap variables $t \to \tau$ and multiply by $e^{\jmath\omega t}$, to give

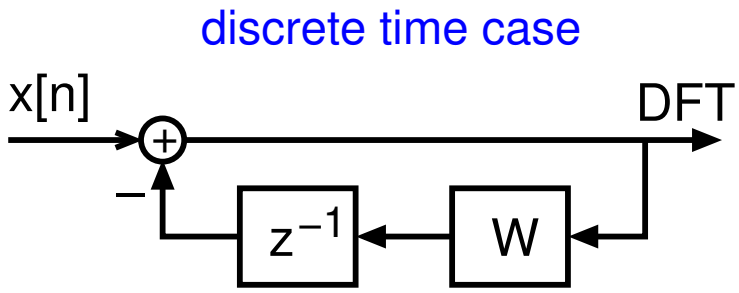
$$e^{\jmath\omega t}\int x(\tau)e^{-\jmath\omega\tau}d\tau = \int x(\tau)\underbrace{e^{\jmath\omega(t-\tau)}}_{h(t-\tau)}d\tau = x(t) * e^{\jmath\omega t} \qquad (= x(t) * h(t))$$

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) $H(z) = \frac{1}{1-z^{-1}W} = \frac{1-z^{-1}W^*}{1-2\cos\theta_k z^{-1}+z^{-2}}$





Notes

Notes