ARMA(p, q) process

Table of Contents

ARMA(p, q) model 1
Autocorrelation of ARMA(p,q) process 2
Quiz: why can we choose to set b0=1? 2
AR(p) process 2
Autocorrelation of AR(p) process 3
Computing AR(p) coefficients from data 3
MA(q) process 4
Problems 4
Estimate PSD from an ACRS assuming MA(1) process 4
Estimate PSD from an ACRS assuming MA(2) process 5
How to generate a realisation of an AR(3) process? 7
ADSI Problem 4.9: MA(q) processes 7
1) Full expressions of MA(0) - MA(3) processes 7
2) Calcute the autocorrelation for the MA processes 8
3) Calculate power spectral density 9
4) Plot the power density spectra given coefficients 10
ADSI Problem 4.10: MA(2) processes and phase properties 11
1) What is the order q of the processes 12
2) Compare the phase properties of the two systems 12
3) Compute and plot the power density spectra of the two processes 13
ADSI Problem 4.12: MA(q) spectral estimation from autocorrelation 15
1) Compute and plot the power density spectrum 15
2) Calculate the model parameters 16
Plot the power density spectrum an AR(3) process 18
Exam 2017 Problem 4: Prove the autocorrelation function for an MA(1) process 19
ADSI Problem 6.2: Autocorrelation expression for an AR(1) process (proof) 20
Problems True/False 23
The autocorrelation function of an AR(p) is equal to zero for lags larger than p 23
Quiz: Is the comb filter an MA(1) proces? 23
Quiz: Determine kind of model based on the inpulse response 23
[✓] Exam 2012, Problem 2: Construct an AR(2) model using a signal from a random process 24
Exam 2016 Problem 1: AR(p) signal modelling 26
1) Compute AR(2) model coefficient given autocorrelation sequence 27
2) Is the random process better modelled as an AR(3) or higher-order process? 29
Exam 2018 Problem 2: Signal Modelling using MA(2) or AR(2) 30
[✓] 1) Calculate and plot the autocorrelation function of the data 30
2) Can data be described by an MA(2) model? 31
3) Calculate the power spectral density of the random process assuming an AR(2) model 32
Problem 1: Signal modelling assuming AR(2) process 34
1) Compute AR(2) model coefficient given autocorrelation sequence 34
2) Is the random process better modelled as an AR(3) or higher-order process? 37
Functions 38

ARMA(p, q) model

An ARMA(p,q) process is defined by the difference equation:

$$y[n] = \underset{\text{feedback autoregressive}}{\boxed{-\sum_{k=1}^{p} a_k y[n-k]}} + \underset{\text{feed forward moving average}}{\boxed{\sum_{k=0}^{q} b_k x[n-k],}} \qquad (13.132)$$

where $x[n] \sim \mathrm{WN}\left(0, \sigma_x^2\right)$

Lesson: we always assume that the input to an ARMA(p,q) process is white noise.

Autocorrelation of ARMA(p,q) process

The autocorrelation of an ARMA(p,q) process is given by:

$$r_{yy}[\ell] = -\sum_{k=1}^{p} a_k r_{yy}[\ell - k] + \sigma_x^2 \sum_{k=0}^{q} b_k h[k-\ell], \quad \text{all } \ell. \qquad (13.137)$$

However, we have an issue with this expression. The impulse response $h[n]$ is actually some function of the coefficients $\{a_k, b_k\}$. The issue is that this function is a non-linear. So solving the analytical expression becomes troublesome.

However, if we get rid of the second term by setting $q = 0$, the relation becomes linear and we have an AR(p) process:

$$r_{yy}[\ell] = -\sum_{k=1}^{p} a_k r_{yy}[\ell - k] + \sigma_x^2 b_0 h[-\ell]. \quad \text{all } \ell \qquad (13.138)$$

We only consider positive lags $\ell > 0$ because $h(\ell) = 0$ for $\ell < 0$ i.e., the impulse response does not exists of negative times because it is a causal system.

$$r_{yy}[\ell] = -\sum_{k=1}^{p} a_k r_{yy}[\ell - k], \;\; \ell > 0 \qquad (13.140)$$

***Quiz:* why can we choose to set b0=1?**

Quiz: why can we choose to set $b_0 = 1$?

AR(p) process

AR(p) model is given by:

$$y(n) = -\sum_{k=1}^{p} \left[a_k \, y(n-k)\right] + b_0 x(n)$$

where the input is white noise with zero mean $x(n) \sim \mathrm{WN}\left(0, \sigma_x^2\right)$.

Autocorrelation of AR(p) process

The autocorrelation of an ARMA(p,q) process is given by:

$$r_{yy}[\ell] = -\sum_{k=1}^{p} a_k r_{yy}[\ell - k] + \sigma_x^2 \sum_{k=0}^{q} b_k h[k - \ell], \quad \text{all } \ell. \qquad (13.137)$$

However, we have an issue with this expression. The impulse response $h[n]$ is actually some function of the coefficients $\{a_k, b_k\}$. The issue is that this function is a non-linear. So solving the analytical expression becomes troublesome.

However, if we get rid of the second term by setting $q = 0$, the relation becomes linear and we have an AR(p) process:

$$r_{yy}[\ell] = -\sum_{k=1}^{p} a_k r_{yy}[\ell - k] + \sigma_x^2 b_0 h[-\ell]. \quad \text{all } \ell \qquad (13.138)$$

We only consider positive lags $\ell > 0$ because $h(\ell) = 0$ for $\ell < 0$ i.e., the impulse response does not exists of negative times because it is a causal system.

$$r_{yy}[\ell] = -\sum_{k=1}^{p} a_k r_{yy}[\ell - k], \; \ell > 0 \qquad (13.140)$$

This is useful because we can use it to find the coefficients $\{a_k\}$ of an AR(q) model. Given an output signal $y(n)$, we can compute the autocorrelation $r_{yy}(\ell)$ numerically in MATLAB. This means that the expression in Eq. 13.141 becomes a set of p linear equations.

Computing AR(p) coefficients from data

For example, if we want to model second-order AR model, $p = 2$ we get two equations with two unknowns:

$$r_{yy}(1) = -a_1 r_{yy}(0) - a_2 r_{yy}(-1)$$

$$r_{yy}(2) = -a_1 r_{yy}(1) - a_2 r_{yy}(0)$$

We can write it into matrix form:

$$\begin{bmatrix} r_{yy}(0) & r_{yy}(1) \\ r_{yy}(1) & r_{yy}(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = -\begin{bmatrix} r_{yy}(1) \\ r_{yy}(2) \end{bmatrix}$$

In the general case, it becomes:

$$\begin{bmatrix} r_{yy}[0] & r_{yy}[1] & \dots & r_{yy}[p-1] \\ r_{yy}[1] & r_{yy}[0] & \dots & r_{yy}[p-2] \\ \vdots & \vdots & \ddots & \vdots \\ r_{yy}[p-1] & r_{yy}[p-2] & \dots & r_{yy}[0] \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = -\begin{bmatrix} r_{yy}[1] \\ r_{yy}[2] \\ \vdots \\ r_{yy}[p] \end{bmatrix},$$

More compactly written as *Yule–Walker equations*:

$$\boldsymbol{R}_y \boldsymbol{a} = -\boldsymbol{r}_y. \qquad (13.143)$$

The input noise variance can be compute as follows:

$$\sigma_x^2 = \sigma_y^2 + \boldsymbol{a}^T \boldsymbol{r}_y = \sigma_y^2 - \boldsymbol{r}_y^T \boldsymbol{R}_y^{-1} \boldsymbol{r}_y \leq \sigma_y^2. \qquad (13.144)$$

We can write a general MATLAB function to find the coefficients $\{a_k\}$ for any AR(p) model `arfit`.

MA(q) process

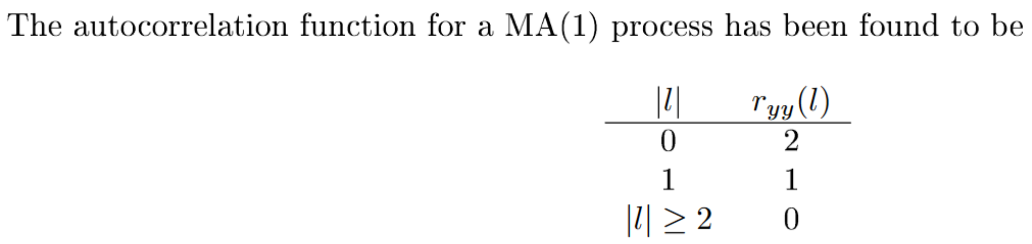
MA(q) process is given by:

$$y(n) = \sum_{k=0}^{q} b_k \, x(n-k)$$

This is the ARMA(p, q) model where the feedback part is excluded i.e., all values of a_k is set to zero.

Problems

Estimate PSD from an ACRS assuming MA(1) process



The autocorrelation function for a MA(1) process has been found to be

$\lvert l \rvert$	$r_{yy}(l)$
0	2
1	1
$\lvert l \rvert \geq 2$	0

```
clear variables;
```

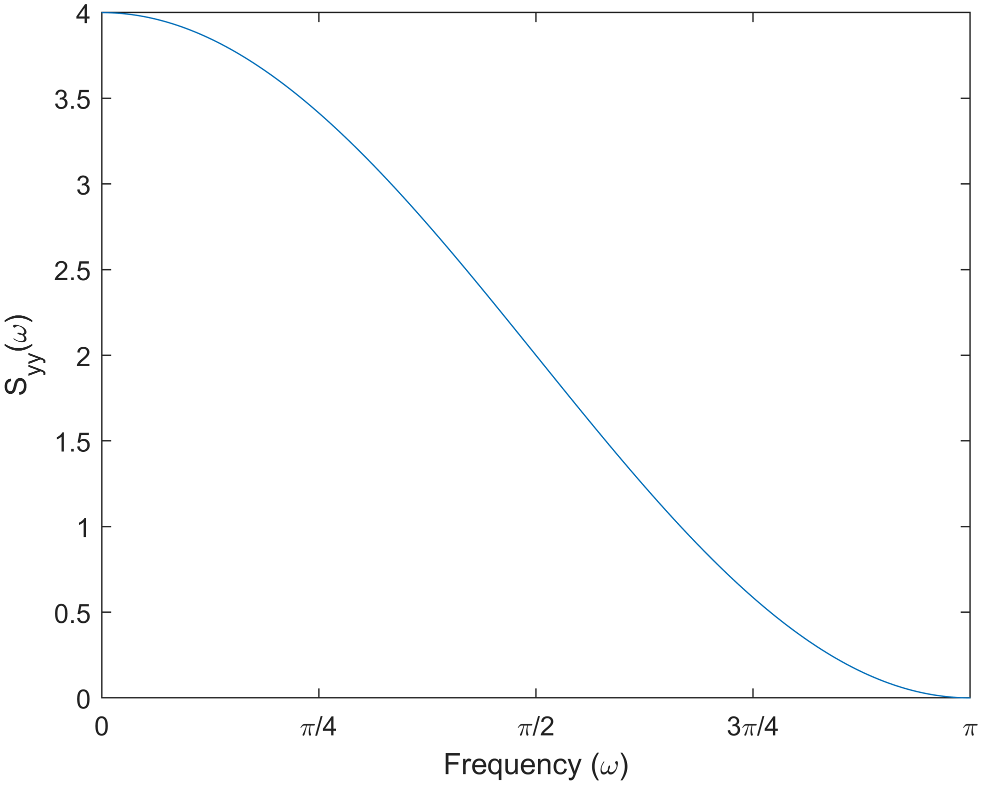
The PSD for an MA(q) process can be computed using Eq. (13.119):

$$S_{xx}(\omega) = r_{xx}[0] + 2\sum_{\ell=1}^{\infty} r_{xx}[\ell] \cos \omega\ell, \qquad (13.119)$$

Plugging in our values, we get:

$$S_{yy}(\omega) = 2 + 2(1\cos(1\omega)) = 2 + 2\cos(\omega)$$

```
w = 0:0.001:2*pi;
r_xx = [2, 1, 0];
S = r_xx(1);
for l = 2:numel(r_xx)
    S = S + 2 * r_xx(l)*cos(w*(l-1));
end
plot(w, S);
xlabel('Frequency (\omega)');
ylabel('S_{yy}(\omega)');
set(gca,'XTick',0:pi/4:pi)
set(gca,'XTickLabel',{'0', '\pi/4', '\pi/2', '3\pi/4', '\pi'})
xlim([0, pi])
```



Estimate PSD from an ACRS assuming MA(2) process

For a given random process $\{x(n)\}$ the autocorrelation has been estimated and is given by:

$\|m\|$	$r_x(m)$
0	4
1	2
2	-1

Estimate the power density spectrum under the assumption that $\{x(n)\}$ can be described as a $MA(2)$ process.

```
clear variables;
```

The PSD for an MA(q) process can be computed using Eq. (13.119):

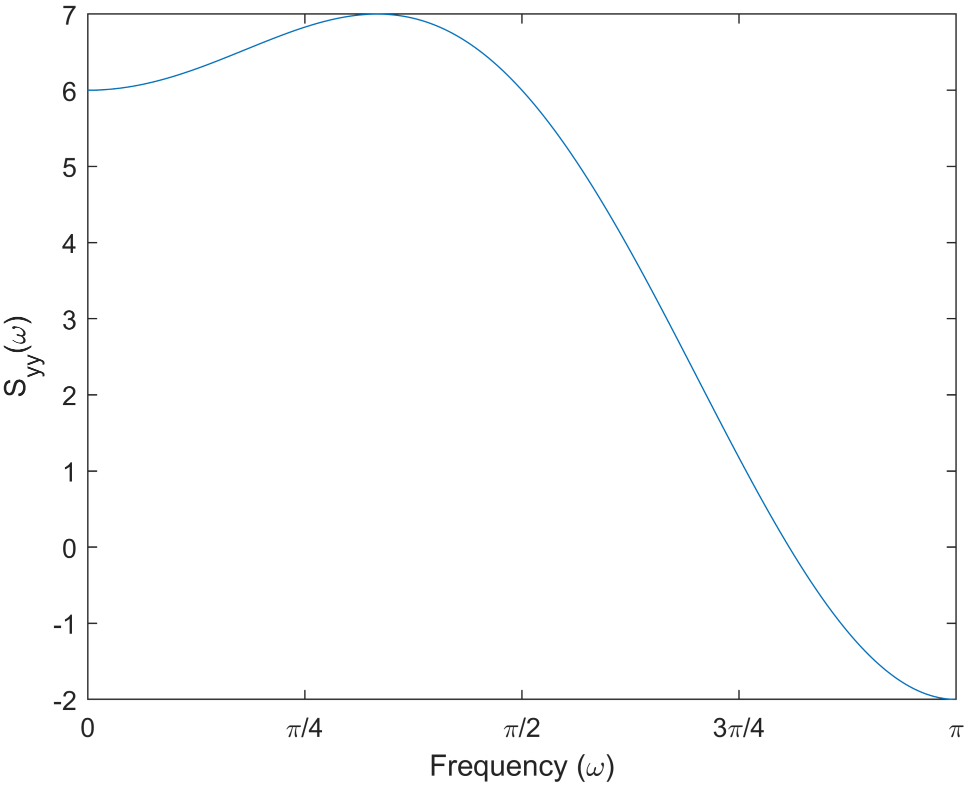
$$S_{xx}(\omega) = r_{xx}[0] + 2\sum_{\ell=1}^{\infty} r_{xx}[\ell]\cos\omega\ell, \qquad (13.119)$$

This gives following expression:

$$S_{xx}(\omega) = 4 + 2(2\cos(w) + (-1)\cos(2\omega))$$

$$= 4 + 4\cos(w) - 2\cos(2\omega)$$

```
w=0:0.001:pi;
r_xx = [4, 2, -1];

S = r_xx(1);
for l = 2:numel(r_xx)
    S = S + 2 * r_xx(l)*cos(w*(l-1));
end

plot(w, S);
xlabel('Frequency (\omega)');
ylabel('S_{yy}(\omega)');
set(gca,'XTick',0:pi/4:pi)
set(gca,'XTickLabel',{'0', '\pi/4', '\pi/2', '3\pi/4', '\pi'})
xlim([0, pi])
```



How to generate a realisation of an AR(3) process?

The signal is disturbed by additive noise from an AR(3) process given by

$$v(n) = -0.5v(n-1) - 0.5v(n-2) - 0.25v(n-3) + w(n)$$

where $w(n) \sim WGN(0, 4)$.

```
% Generate white noise w[n] with zero mean and variance 4
w_var = 4;
w = sqrt(w_var) * randn(10000, 1);

% Generate noise from an AR(3) process
b = 1;
a = [1, 0.5, 0.5, 0.25];
v = filter(b, a, w);
```

ADSI Problem 4.9: MA(q) processes

In this problem we will investigate the MA(q) process as defined by Eq. (13.132) by excluding the feedback part

$$y[n] = \sum_{k=0}^{q} b_k x[n-k]$$

where the input is a zero-mean Gaussian white noise process with unit variance.

1) Full expressions of MA(0) - MA(3) processes

Write out the full expressions for MA(0), MA(1), MA(2) and MA(3) processes.

The general $\text{ARMA}(p, q)$ is given by the difference equation:

$$y[n] = -\sum_{k=1}^{p} a_k y[n-k] + \sum_{k=0}^{q} b_k x[n-k], \qquad (13.132)$$

When the feedback part is excluded, all values of a_k is set to zero, we are left with:

$$y[n] = \sum_{k=0}^{q} b_k\, x[n-k]$$

We can write out the full expressions for the difference processes as follows:

$$\text{MA}(0) \;\rightarrow\; y[n] = b_0\, x[n]$$

$$\text{MA}(1) \;\rightarrow\; y[n] = b_0\, x[n] + b_1\, x[n-1]$$

$$\mathrm{MA}(2) \quad \rightarrow \quad y[n] = b_0\,x[n] + b_1\,x[n-1] + b_2\,x[n-2]$$

$$\mathrm{MA}(3) \quad \rightarrow \quad y[n] = b_0\,x[n] + b_1\,x[n-1] + b_2\,x[n-2] + b_3\,x[n-3]$$

2) Calcute the autocorrelation for the MA processes

Calculate the autocorrelation $r_{yy}[l]$ for the MA(0) to MA(3) processes.

The autocorrelation function of a random process is defined as:

$$r_{\mathrm{yy}}(\ell) = E[y(n)y(n-\ell)]$$

To compute the autocorrelation for the $\mathrm{MA}(0)$, just plug its difference equation into this equation:

$$r_{\mathrm{yy}}(\ell) = E[b_0\,x(n)\cdot b_0\,x(n-\ell)]$$

$$r_{\mathrm{yy}}(\ell) = b_0^2 E[x(n)\cdot x(n-\ell)]$$

$$r_{\mathrm{yy}}(\ell) = b_0^2\,r_{\mathrm{xx}}(\ell)$$

The autocorrelation of a white noise signal $w(n)$ is: $r_{\mathrm{ww}}(\ell) = \sigma_w^2\delta(\ell)$ where σ_w^2 is the signal's variance (see week 6 notes)

We are given that the input signal $x(n)$ is Gaussian white noise with unit variance. This means $r_{\mathrm{xx}}(\ell) = \delta(\ell)$

Therefore, the autocorrelation for MA(0) process is:

$$r_{\mathrm{yy}}(\ell) = b_0^2\,\delta(\ell)$$

The autocorrelation for the $\mathrm{MA}(1)$ process is:

$$r_{\mathrm{yy}}(\ell) = E[(b_0\,x[n] + b_1\,x[n-1])\cdot(b_0\,x[n-\ell] + b_1\,x[n-\ell-1])]$$

Multiply the two factors:

$$r_{\mathrm{yy}}(\ell) = E\left[b_0^2\,x[n]x[n-\ell] + b_0b_1x[n]x[n-\ell-1] + b_0b_1\,x[n-1]\,x[n-\ell] + b_1^2x[n-1]x[n-\ell-1]\right]$$

Rearrange the coefficients:

$$r_{\mathrm{yy}}(\ell) = E\left[b_0^2\,x[n]x[n-\ell] + b_0b_1(x[n]x[n-\ell-1] + x[n-1]\,x[n-\ell]) + b_1^2x[n-1]x[n-\ell-1]\right]$$

Split the expectation values and move constants outside $E[\cdot]$

$$r_{\mathrm{yy}}(\ell) = b_0^2\,E[x[n]x[n-\ell]] + b_0b_1\,E[(x[n]x[n-\ell-1] + x[n-1]\,x[n-\ell])] + b_1^2\,E[x[n-1]x[n-\ell-1]]$$

Split the expectation value in the second term:

$$r_{\mathrm{yy}}(\ell) = b_0^2\,E[x[n]x[n-\ell]] + b_0b_1\,(E[x[n]x[n-\ell-1]] + E[x[n-1]\,x[n-\ell]]) + b_1^2\,E[x[n-1]x[n-\ell-1]]$$

Replace expectation values with autocorrelation functions:

$$r_{\mathrm{yy}}(\ell) = b_0^2\, r_{\mathrm{xx}}(\ell) + b_0 b_1 \left(r_{\mathrm{xx}}(\ell-1) + r_{\mathrm{xx}}(\ell+1)\right) + b_1^2\, r_{\mathrm{xx}}(\ell)$$

Rearrange the terms

$$r_{\mathrm{yy}}(\ell) = \left(b_0^2 + b_1^2\right) r_{\mathrm{xx}}(\ell) + b_0 b_1 \left(r_{\mathrm{xx}}(\ell-1) + r_{\mathrm{xx}}(\ell+1)\right)$$

As before $r_{\mathrm{xx}}(\ell) = \delta(\ell)$ because the input signal is a white noise with unit variance. Substitute:

$$r_{\mathrm{yy}}(\ell) = \left(b_0^2 + b_1^2\right) \delta(\ell) + b_0 b_1 \left(\delta(\ell-1) + \delta(\ell+1)\right)$$

The autocorrelation for an MA(2) process is:

$$r_{\mathrm{yy}}(\ell) = \left(b_0^2 + b_1^2 + b_2^2\right)\delta(\ell) + (b_0 b_1 + b_1 b_2)\delta(\ell+1) + (b_0 b_1 + b_1 b_2)\delta(\ell-1) + b_0 b_2 \delta(\ell+2) + b_0 b_2 \delta(\ell-2)$$

The autocorrelation for an MA(3) process is:

$$\begin{aligned} r_{yy}(l) = (b_0^2 + b_1^2 + b_2^3 + b_3^2)\delta(l) + (b_0 b_1 + b_1 b_2 + b_2 b_3)\delta(l+1) + (b_0 b_1 + b_1 b_2 + b_2 b_3)\delta(l-1) \\ + (b_0 b_2 + b_1 b_3)\delta(l+2) + (b_0 b_2 + b_1 b_3)\delta(l-2) + b_0 b_3 \delta(l+3) + b_0 b_3 \delta(l-3) \end{aligned}$$

3) Calculate power spectral density

Calculate the general expressions for power density spectra of MA(0), MA(1) and MA(2) processes using Eq. (13.112).

The power spectral density (PSD) is defined as:

$$S_{xx}(\omega) \triangleq \sum_{\ell=-\infty}^{\infty} r_{xx}[\ell] \mathrm{e}^{-\mathrm{j}\ell\omega}. \qquad (13.112)$$

In 2) we found that the autocorrelation for $\mathrm{MA}(0)$ is $r_{\mathrm{yy}}(\ell) = b_0^2\, \delta(\ell)$, so plug it in this PSD formula:

$$S_{\mathrm{yy}}^{\mathrm{MA}(0)}(\omega) = \sum_{\ell=-\infty}^{\infty} r_{\mathrm{yy}}(\ell) e^{-\mathrm{j}\ell\omega} = \sum_{\ell=-\infty}^{\infty} b_0^2\, \delta(\ell) e^{-\mathrm{j}\ell\omega}$$

We notice that $r_{\mathrm{yy}}(\ell)$ has non-zero value only when $\ell = 0$. Therefore, the terms for when $\ell \neq 0$ are zero. We can, therefore, remove the infinite sum. Since $\delta(0) = 1$ and $e^{-\mathrm{j}0\omega} = 1$, we have:

$$S_{\mathrm{yy}}^{\mathrm{MA}(0)}(\omega) = b_0^2\, \delta(0) e^{-\mathrm{j}0\omega} = b_0^2 \cdot 1 \cdot 1 = b_0^2$$

We do similar computation to find the power density spectrum of MA(1) process.

In 2) we found that the autocorrelation for MA(1) is:

$$r_{\mathrm{yy}}(\ell) = \left(b_0^2 + b_1^2\right)\delta(\ell) + b_0 b_1\left(\delta(\ell-1) + \delta(\ell+1)\right)$$

There are three non-zero values of the autocorrelation function so the infinite sum has only three terms

$$S_{\mathrm{yy}}^{\mathrm{MA}(1)}(\omega) = \sum_{\ell=-\infty}^{\infty}\left(\left(b_0^2 + b_1^2\right)\delta(\ell) + b_0 b_1\left(\delta(\ell-1) + \delta(\ell+1)\right)\right)e^{-\mathrm{j}\ell\omega}$$

There are three non-zero values of the autocorrelation function so the infinite sum has only three terms:

- When $\ell = 0$: $\left(b_0^2 + b_1^2\right)\delta(0)e^{-j(0)\omega} = \left(b_0^2 + b_1^2\right)\cdot 1\cdot 1 = \left(b_0^2 + b_1^2\right)$
- When $\ell = -1$: $b_0 b_1\,\delta(0)e^{-j(-1)\omega} = b_0 b_1\,(1)e^{j\omega} = b_0 b_1\,e^{j\omega}$
- When $\ell = 1$: $b_0 b_1\,\delta(0)e^{-j(1)\omega} = b_0 b_1\,(1)e^{-j\omega} = b_0 b_1\,e^{-j\omega}$

Therefore, we have:

$$S_{\mathrm{yy}}^{\mathrm{MA}(1)}(\omega) = \left(b_0^2 + b_1^2\right) + b_0 b_1 e^{j\omega} + b_0 b_1 e^{-j\omega}$$

$$S_{\mathrm{yy}}^{\mathrm{MA}(1)}(\omega) = \left(b_0^2 + b_1^2\right) + b_0 b_1\left(e^{j\omega} + e^{-j\omega}\right)$$

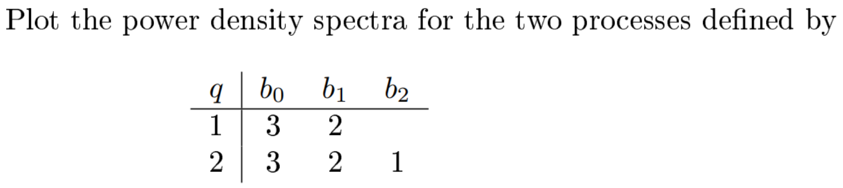
Since $\cos(\theta) = \frac{1}{2}\left(e^{j\theta} + e^{-j\theta}\right)$ and $2\cdot\cos(\theta) = e^{j\theta} + e^{-j\theta}$, we write:

$$S_{\mathrm{yy}}^{\mathrm{MA}(1)}(\omega) = \left(b_0^2 + b_1^2\right) + 2b_0 b_1\cos(\omega)$$

In similar fashion, the PSD of $\mathrm{MA}(2)$ is:

$$S_{\mathrm{yy}}^{\mathrm{MA}(2)}(\omega) = \left(b_0^2 + b_1^2 + b_2^2\right) + 2\left(b_0 b_1 + b_1 b_2\right)\cos(\omega) + 2b_0 b_1\cos(2\omega)$$

4) Plot the power density spectra given coefficients



Plot the power density spectra for the two processes defined by

q	b_0	b_1	b_2
1	3	2	
2	3	2	1

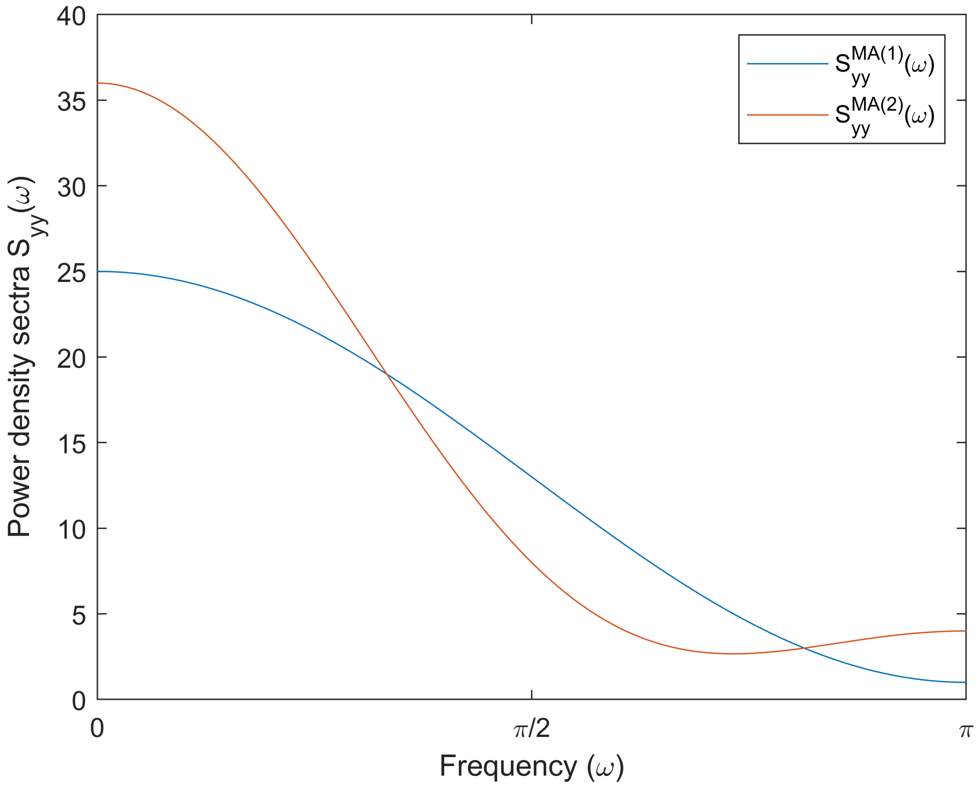
We are given the coefficients for $S_{\mathrm{yy}}^{\mathrm{MA}(1)}(\omega)$ and $S_{\mathrm{yy}}^{\mathrm{MA}(2)}(\omega)$. We use results from 3) to plot the power density spectra for the two processes.

$$S_{\mathrm{yy}}^{\mathrm{MA}(1)}(\omega) = \left(b_0^2 + b_1^2\right) + 2b_0 b_1\cos(\omega)$$

$$S_{\mathrm{yy}}^{\mathrm{MA}(2)}(\omega) = \left(b_0^2 + b_1^2 + b_2^2\right) + 2\left(b_0 b_1 + b_1 b_2\right)\cos(\omega) + 2b_0 b_1\cos(2\omega)$$

```
w=0:0.001:2*pi;
```

```
% Define the coefficients
b0 = 3; b1 = 2; b2 = 1;

% The power desnity spectrum for the MA(1) process
S1 = (b0^2 + b1^2) + 2*b0*b1*cos(w);
% The power desnity spectrum for the MA(2) process
S2 = (b0^2 + b1^2 + b2^2) + 2*(b0*b1 + b1*b2)*cos(w) + 2*b0*b2*cos(2*w);

plot(w, S1, w, S2);
legend('S_{yy}^{MA(1)}(\omega)','S_{yy}^{MA(2)}(\omega)');
xlabel('Frequency (\omega)');
ylabel('Power density sectra S_{yy}(\omega)');
set(gca,'XTick',0:pi/2:2*pi)
set(gca,'XTickLabel',{'0','\pi/2','\pi','3\pi/2','2\pi'})
xlim([0,pi])
```



ADSI Problem 4.10: MA(2) processes and phase properties

Consider the following two MA(q) systems and assume that they are excited by zero-mean white Gaussian noise with unit variance.

$$y_1[n] = 2x[n] - x[n-2] \quad \text{and} \quad y_2[n] = x[n] - 2x[n-2]$$

1) What is the order q of the processes

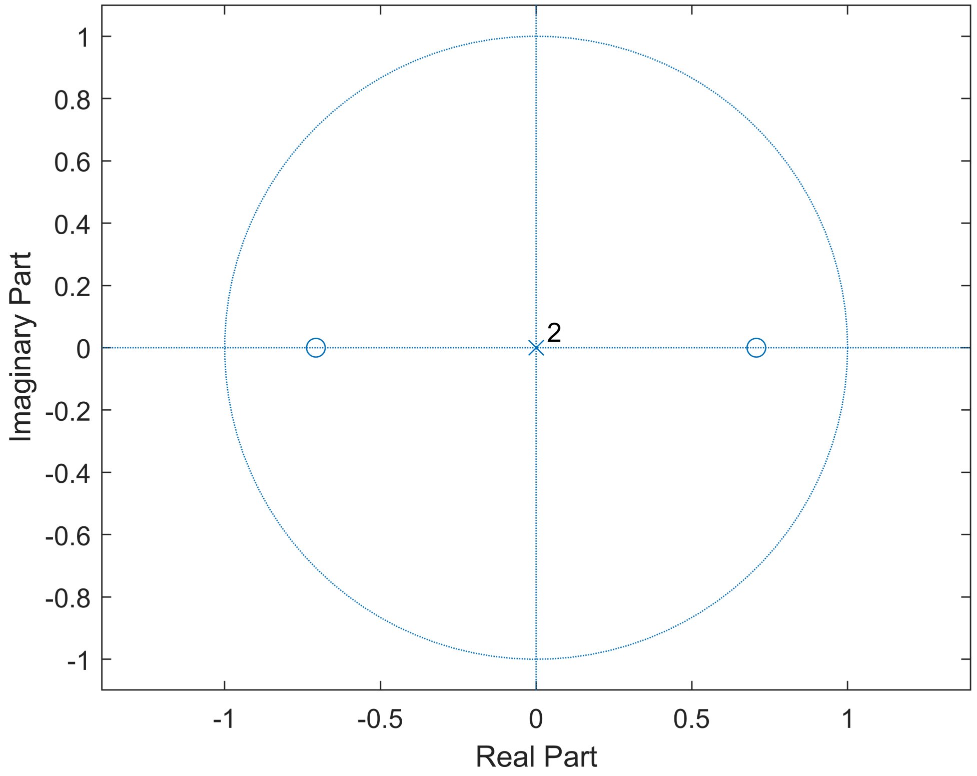
The order q of an $\mathrm{MA}(q)$ process is given by the longest delay, i.e., $x[n-q]$. Therefore, both systems are of order 2.

2) Compare the phase properties of the two systems

One important phase property of a system is whether it is invertible. For a MA(q) system, this implies that the all zeros should be inside the unit circle. If all zeros are located within the unit circle, then we have a minimum-phase system.

We can use zero-pole plot in MATLAB:

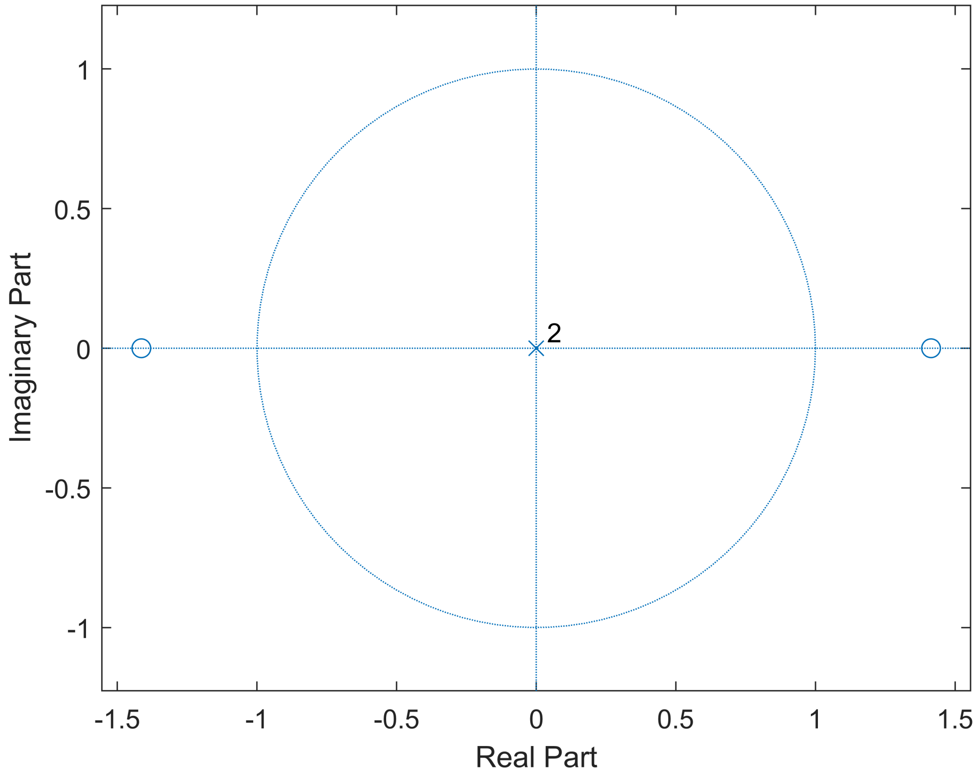
```
h1 = [2, 0, -1]; % Impulse response for y1[n]
zplane(h1);
```



From the zero-pole plot, we observe that all the zeros are within the unit circle. Therefore, we can conclude that the system $y_1[n]$ is a minimum-phase.

Let us make the zero-plot for the system $y_2[n] = x[n] - 2x[n-2]$

```
h2 = [1, 0, -2]; % Impulse response for y2[n]
zplane(h2);
```



Since both zeros are outside the unit circle, we can conclude the system $y_2[n]$ is a maximum-phase system.

3) Compute and plot the power density spectra of the two processes

We are given two MA(q) systems:

$$y_1(n) = 2x(n) - x(n-2)$$

$$y_2(n) = x(n) - 2x(n-2)$$

where the input is $x(n) \sim \mathrm{WGN}(0, 1)$ i.e., white Gaussian noise with unit variance.

We are asked to compute the power density spectra of both systems.

Before we can compute the power density spectrum, we need to find the autocorrelation.

Since we know impulse response, we can use Eq. (13.104) to compute the autocorrelation:

$$r_{yy}[\ell] = r_{hh}[\ell] * r_{xx}[\ell], \qquad (13.104)$$

where

$$r_{hh}[m] = h[-m] * h[m], \qquad (13.103)$$

The autocorrelation of white noise is $\sigma_x^2\delta(\ell)$ so the ACRS of the input signal:

$$r_{xx}(\ell) = \sigma_x^2\delta(\ell) = 1 \cdot \delta(\ell)$$

Since $r_{xx}(\ell) = \delta(\ell)$ then Eq. (13.104) becomes:

$$r_{yy}(\ell) = r_{hh}(\ell)$$

This means that if we compute $r_{hh}(\ell)$ then we have the autocorrelation.

We can use MATLAB to compute $r_{h1}(\ell)$ and $r_{h2}(\ell)$:

```
r_h1 = conv(fliplr(h1), h1)
```

```
r_h1 = 1×5
    -2     0     5     0    -2
```

```
r_h2 = conv(fliplr(h2), h2)
```

```
r_h2 = 1×5
    -2     0     5     0    -2
```

Now, we have the autocorrelation of both systems.

$\lvert\ell\rvert$	r_{y1}	r_{y2}
0	5	5
1	0	0
2	−2	−2
≥ 3	0	0

We observe that they are the same. This means that the power spectrum density will be the same.

Whenever we need to compute the PSD based on an autocorrelation sequence, we use Eq. (13.119):

$$S_{xx}(\omega) = r_{xx}[0] + 2\sum_{\ell=1}^{\infty} r_{xx}[\ell]\cos\omega\ell, \qquad (13.119)$$

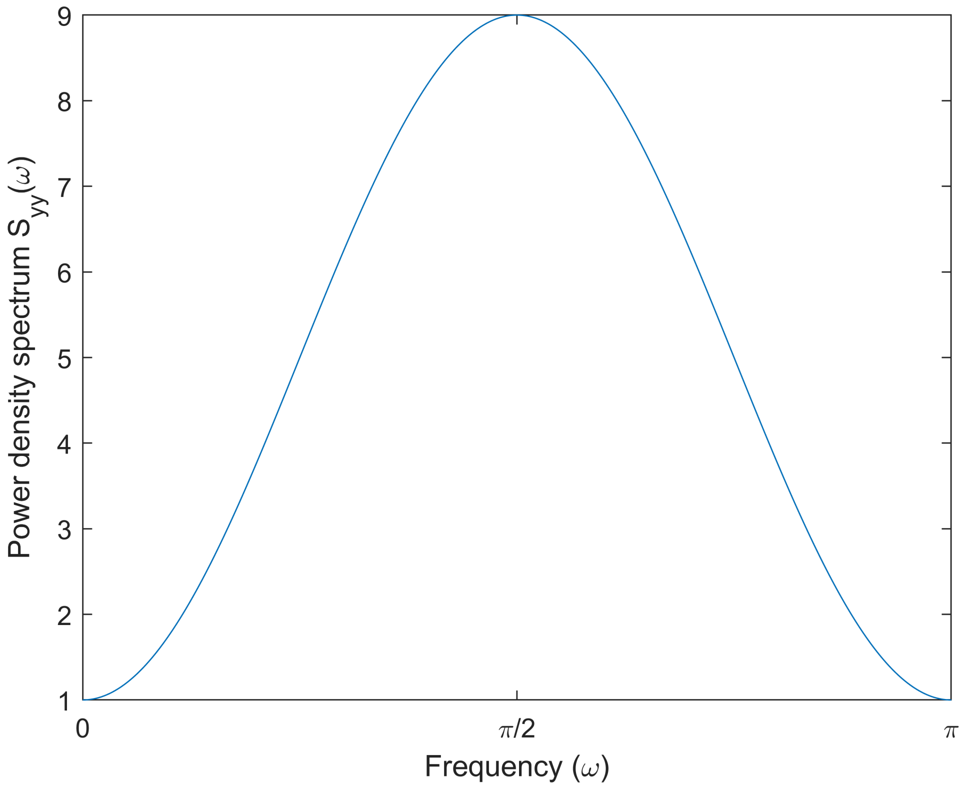
Plugging in our values, we get:

$$S_{yy}(\omega) = 5 + 2(0 + (-2)\cos(2\omega))$$

$$S_{yy}(\omega) = 5 - 4\cos(2\omega)$$

We can plot it in MATLAB:

```
w=0:0.001:2*pi;
S_yy = 5 - 4*cos(2*w);
plot(w, S_yy);
xlabel('Frequency (\omega)');
ylabel('Power density spectrum S_{yy}(\omega)');
set(gca,'XTick',0:pi/2:2*pi)
set(gca,'XTickLabel',{'0','\pi/2','\pi','3\pi/2','2\pi'})
xlim([0, pi])
```



ADSI Problem 4.12: MA(q) spectral estimation from autocorrelation

The autocorrelation function for a MA(1) process has been found to be

$\|l\|$	$r_{yy}(l)$
0	2
1	1
$\|l\| \geq 2$	0

```
clear variables;
```

1) Compute and plot the power density spectrum

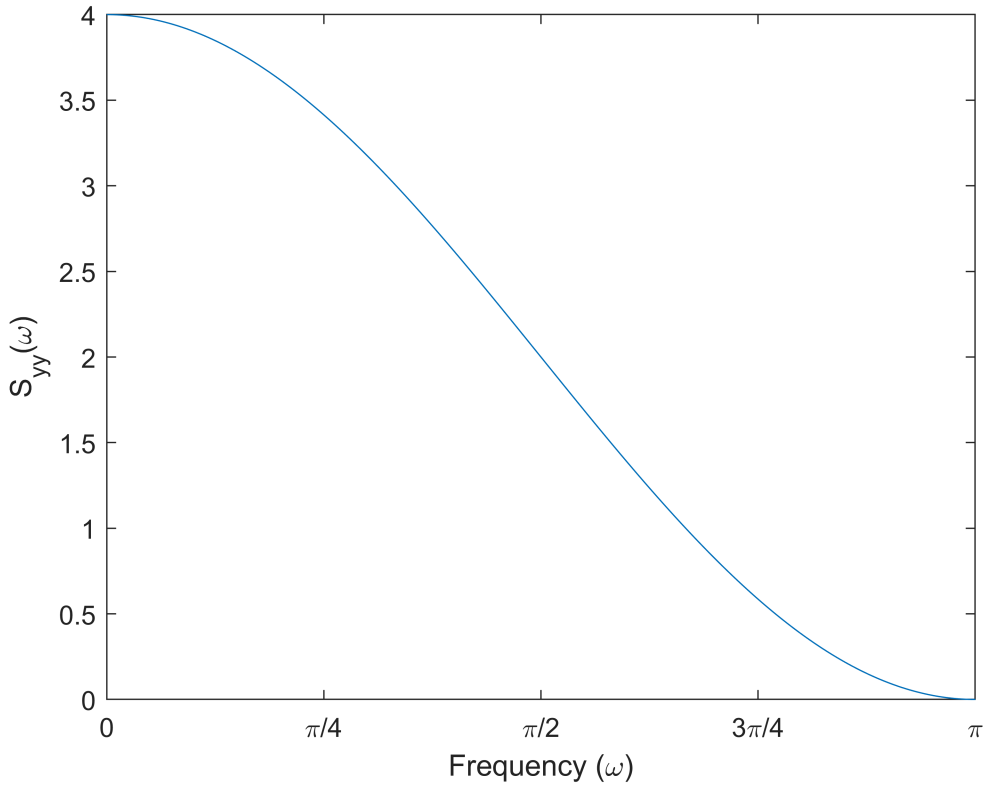
Whenever we need to compute the PSD based on an autocorrelation sequence, we use Eq. (13.119):

$$S_{xx}(\omega) = r_{xx}[0] + 2\sum_{\ell=1}^{\infty} r_{xx}[\ell]\cos\omega\ell, \qquad (13.119)$$

Plugging in our values, we get:

$$S_{yy}(\omega) = 2 + 2(1\cos(1\omega)) = 2 + 2\cos(\omega)$$

```
w=0:0.001:2*pi;
S_yy = 2 + 2*cos(w);
plot(w, S_yy);
xlabel('Frequency (\omega)');
ylabel('S_{yy}(\omega)');
set(gca,'XTick',0:pi/4:pi)
set(gca,'XTickLabel',{'0','\pi/4','\pi/2','3\pi/4','\pi'})
xlim([0, pi])
```



2) Calculate the model parameters

We are asked to find b_0 and b_1 for an MA(1) process given its autocorrelation sequence.

In 1) we found that:

$$S_{yy}(\omega) = 2 + 2\cos(\omega)$$

In ADSI Problem 4.9, we found that:

$$S_{\mathrm{yy}}^{\mathrm{MA}(1)}(\omega) = \left(b_0^2 + b_1^2\right) + 2b_0b_1\cos(\omega)$$

This implies that:

$$b_0^2 + b_1^2 = 2 \text{ and } b_0b_1 = 1$$

Rewriting the second equation:

$$b_0 = \frac{1}{b_1}$$

We can substitute in the first equation:

$$\frac{1}{b_1^2} + b_1^2 = 2$$

From the equation above, we see that $b_1 = \pm 1$. We turn our attention the second equation:

$$b_0 = 1 \text{ if } b_1 = 1$$

$$b_0 = -1 \text{ if } b_1 = -1$$

So the model parameters are $b_0 = 1, b_1 = 1$ or $b_0 = -1, b_1 = -1$.

```matlab
% Verify in MATLAB
syms b0 b1
solution = solve([b0^2 + b1^2 == 2, b0*b1 == 1], [b0, b1]);
solution.b0
```

```
ans =
```

$$\begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

```matlab
solution.b1
```

```
ans =
```

$$\begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

Plot the power density spectrum an AR(3) process

```
clear variables;
```

Here is an AR(3) process plus some white Gaussian noise:

$$v(n) = -0.5v(n-1) - 0.5v(n-2) - 0.25v(n-3) + w(n)$$
$$\text{where } w(n) \sim WGN(0,4).$$

We can determine the Power Density Spectrum of an ARMA(p,q) process is given by

$$S_{yy}(\omega) = \sigma_x^2 \left|H(\mathrm{e}^{\mathrm{j}\omega})\right|^2 = \sigma_x^2 \left| \frac{\sum_{k=0}^{q} b_k \mathrm{e}^{-\mathrm{j}\omega k}}{1 + \sum_{k=1}^{p} a_k \mathrm{e}^{-\mathrm{j}\omega k}} \right|^2 . \qquad (13.133)$$

The power spectrum of an AR(p) process is given by:

$$S_{\mathrm{yy}}(\omega) = \sigma_x^2 \left| \frac{1}{1 + \sum_{k=1}^{p} a_k e^{-j\omega k}} \right|^2$$

For this problem, we have:

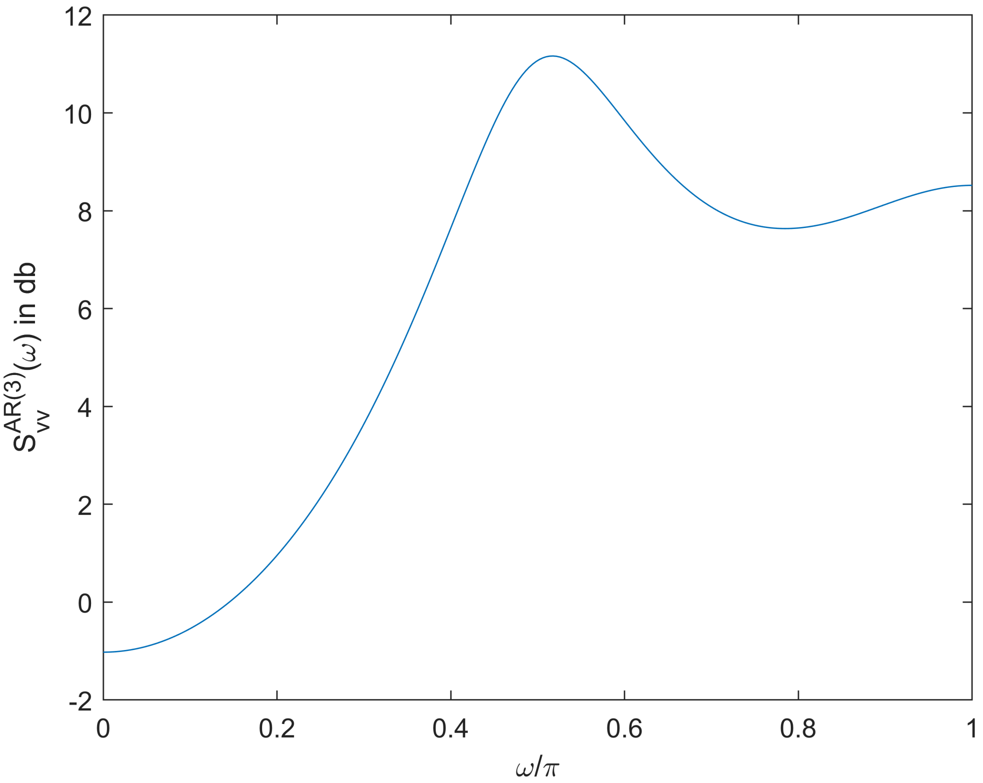
$$S_{\mathrm{vv}}(\omega) = 4 \left| \frac{1}{1 + 0.5e^{-j\omega} + 0.5e^{-\mathrm{j}2\omega} + 0.25e^{-\mathrm{j}3\omega}} \right|^2$$

The algorithm is as follows:

1. Use the coefficents $\{a_1, a_2, \cdots, a_p\}$ for the $\mathrm{AR}(p)$ model,
2. Compute the transfer function for the $\mathrm{AR}(p)$ by computing the sum and finding its reciprocal
3. Compute the conjugate of the transfer function: $|H(e^{j\omega})|^2$
4. Multiply it with the variance σ_x^2

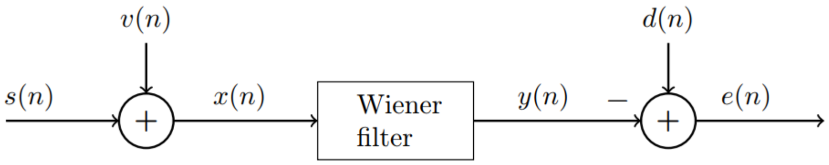
The algorithm is implemented in the functions `ar2psd()` function:

```
N = 256;

a = [0.5, 0.5, 0.25]; % The coefficients of the AR(3) model
w_var = 4; % The variance of white noise
[S_vv, w] = ar2psd(a, w_var, N); % Compute the PSD of AR(3) model

plot(w/pi, pow2db(S_vv))
xlabel('\omega/\pi')
ylabel('S_{vv}^{AR(3)}(\omega) in db')
```



Exam 2017 Problem 4: Prove the autocorrelation function for an MA(1) process

Consider the Wiener filtering problem shown below. A signal $s(n)$ is corrupted by additive, uncorrelated noise $v(n)$ giving the signal $x(n) = s(n) + v(n)$. It is desired to use a 2-tap Wiener filter to recover the signal i.e. $d(n) = s(n)$.



The noise $v(n)$ is generated in an MA(1) process as

$$v(n) = w(n) + w(n-1)$$

where $w(n)$ is a zero-mean gaussian white noise sequence with $\sigma_w^2 = 1$. The autocorrelation function of the signal is

$$r_s(l) = 4 \cdot 0.25^{|l|}.$$

```
clear variables;
```

1. Show that the autocorrelation function of the noise $v(n)$ is given by

$$r_v(l) = \delta(l-1) + 2\delta(l) + \delta(l+1).$$

In ADSI Problem 4.9, we found that the autocorrelation for an MA(1) process where the input signal is a white noise with unit variance can be described as:

$$r_v(\ell) = \left(b_0^2 + b_1^2\right)\delta(\ell) + b_0 b_1 \left(\delta(\ell-1) + \delta(\ell+1)\right)$$

In this problem $b_0 = b_1 = 1$:

$$r_v(\ell) = \delta(\ell-1) + 2\delta(\ell) + \delta(\ell+1)$$

```
r_vv = [2, 1, 0]
```

```
r_vv = 1×3
     2     1     0
```

ADSI Problem 6.2: Autocorrelation expression for an AR(1) process (proof)

Consider an AR(1) process given by $y(n) = -ay(n-1) + x(n)$ with $-1 < a < 1$ and $x(n)$ $WN(0, \sigma_x^2)$.

1. Show that the autocorrelation of the AR(1) process is given by

$$r_{yy}(l) = \frac{\sigma_x^2}{1-a^2}(-a)^{|l|}$$

Hint: Use equation (13.138) and (13.140).

The autocorrelation of an AR(p) process is given by Eq. 13.138:

$$r_{yy}[\ell] = -\sum_{k=1}^{p} a_k r_{yy}[\ell - k] + \sigma_x^2 b_0 h[-\ell]. \text{ all } \ell \qquad (13.138)$$

where $h[n]$ is the impulse response of an all-pole system.

The impulse response of an all-pole system satisfies the difference equation

$$h[n] = -\sum_{k=1}^{p} a_k h[n-k] + b_0 \delta[n], \; n \geq 0 \qquad (13.139)$$

where $h[n] = 0$ for $n < 0$

Equation 13.138 is an general expression for all ℓ. However, since $h[\ell] = 0$ for negative values of ℓ then we know that $h[-\ell] = 0$. Therefore, Eq. 13.138 can be reduced to:

$$r_{yy}[\ell] = -\sum_{k=1}^{p} a_k r_{yy}[\ell - k], \; \ell > 0 \qquad (13.140)$$

For an AR(1) process, Eq. 13.138 simplifies to:

$$r_{yy}[\ell] = -a_1 r_{yy}[\ell - 1] + \sigma_x^2 b_0 h[-\ell] \text{ for all } \ell$$

For an AR(1) process, Eq. 13.140 simplifies to:

$$r_{yy}[\ell] = -a_1 r_{yy}[\ell - 1] \text{ for } \ell > 0$$

Setting $\ell = 0$ in the first equation, we get:

$$r_{yy}[0] = -a_1 r_{yy}[-1] + \sigma_x^2 b_0 h[0]$$

The book says that $h[0] = b_0 = 1$ so we are left with:

$$r_{yy}[0] = -a_1 r_{yy}[-1] + \sigma_x^2$$

We can use the symmetry property of autocorrelation function i.e., $r_{yy}[-\ell] = r_{yy}[\ell]$:

$$r_{\mathrm{yy}}[0] = -a_1 r_{\mathrm{yy}}[1] + \sigma_x^2$$

To find an expression for $r_{\mathrm{yy}}[1]$, we set $\ell = 1$ in the second equation:

$$r_{\mathrm{yy}}[1] = -a_1 r_{\mathrm{yy}}[0]$$

We insert the second equation into the first equation:

$$r_{\mathrm{yy}}[0] = -a_1(-a_1 r_{\mathrm{yy}}[0]) + \sigma_x^2$$

$$r_{\mathrm{yy}}[0] = a_1^2 r_{\mathrm{yy}}[0] + \sigma_x^2$$

$$\sigma_x^2 = r_{\mathrm{yy}}[0] - a_1^2 r_{\mathrm{yy}}[0]$$

$$\sigma_x^2 = r_{\mathrm{yy}}[0]\left(1 - a_1^2\right)$$

$$r_{\mathrm{yy}}[0] = \frac{\sigma_x^2}{1 - a_1^2}$$

Now, we need to find an expression for $\ell > 0$. We can do this by using the second equation.

First, we set $\ell = 1$ in the second equation:

$$r_{\mathrm{yy}}[1] = -a_1 r_{\mathrm{yy}}[0] \Leftrightarrow -a_1 \frac{\sigma_x^2}{1 - a_1^2}$$

$$r_{\mathrm{yy}}[2] = -a_1 r_{\mathrm{yy}}[1] \Leftrightarrow -a_1 \left(-a_1 \frac{\sigma_x^2}{1 - a_1^2}\right) \Leftrightarrow (-a_1)^2 \frac{\sigma_x^2}{1 - a_1^2}$$

$$r_{\mathrm{yy}}[3] = -a_1 r_{\mathrm{yy}}[2] \Leftrightarrow -a_1 \left((-a_1)^2 \frac{\sigma_x^2}{1 - a_1^2}\right) \Leftrightarrow (-a_1)^3 \frac{\sigma_x^2}{1 - a_1^2}$$

This means that in general, we have:

$$r_{\mathrm{yy}}[\ell] = (-a_1)^{\ell} \frac{\sigma_x^2}{1 - a_1^2}$$

If we apply the symmetric property of the autocorrelation, we get:

$$r_{\mathrm{yy}}[\ell] = (-a_1)^{|\ell|} \frac{\sigma_x^2}{1 - a_1^2}$$

Problems True/False

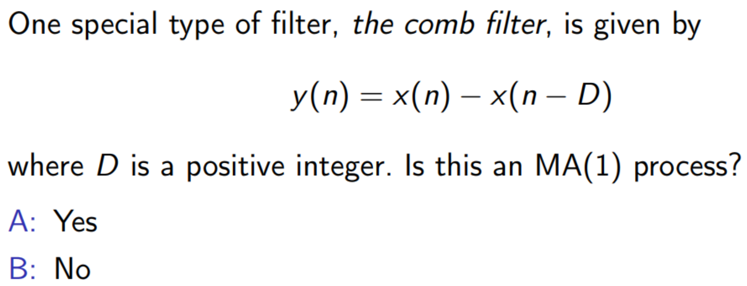
The autocorrelation function of an AR(*p*) is equal to zero for lags larger than *p*.

Answer: FALSE!

An AR(*p*) process is effectively white noise sent through an IIR filter. This implies that a correlation will exist between infinitely distant samples. While the autocorrelation function can be zero for any lag it is generally different from zero for almost all lags.

Note though, that for all AR(2) processes encountered in real life, the autocorrelation values gets exponentially close to zero for large lags.

Quiz: Is the comb filter an MA(1) proces?



One special type of filter, *the comb filter*, is given by

$$y(n) = x(n) - x(n - D)$$

where D is a positive integer. Is this an MA(1) process?

A: Yes

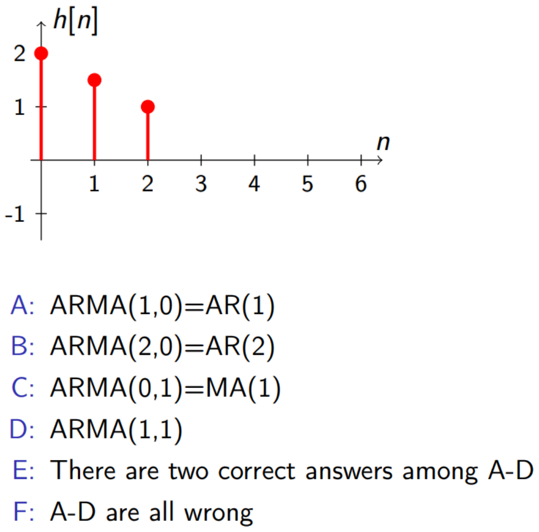
B: No

The answer is no!

Quiz: Determine kind of model based on the inpulse response

The impulse response for a specific ARMA(*p*,*q*) model is shown below.

What kind of model is shown?



A: ARMA(1,0)=AR(1)

B: ARMA(2,0)=AR(2)

C: ARMA(0,1)=MA(1)

D: ARMA(1,1)

E: There are two correct answers among A-D

F: A-D are all wrong

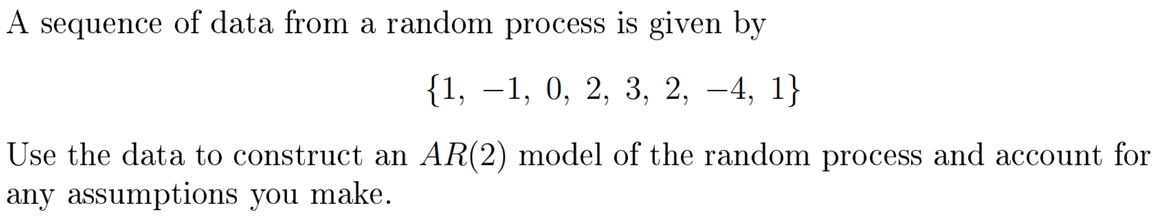
The correct answer is F.

The autoregressive part of the ARMA(p,q) model has infinite impulse response. What we see in the figure is finite impulse response with three elements. Therefore, it must be an MA(q) model.

Since the figure shows three filter coefficients, the system must be an MA(2) process.

If the figure indicated infinite filter coefficients, we could not have determined the order of the model. In this case, it would be either an autoregressive model or a high-order MA(q) model.

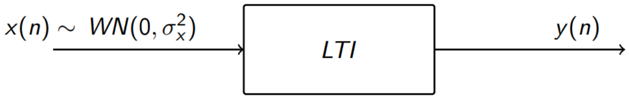
[✔] Exam 2012, Problem 2: Construct an AR(2) model using a signal from a random process


A sequence of data from a random process is given by

$$\{1,\ -1,\ 0,\ 2,\ 3,\ 2,\ -4,\ 1\}$$

Use the data to construct an $AR(2)$ model of the random process and account for any assumptions you make.

We assume that the given sequence is the output of a process whose input was white Gaussian noise with zero mean and unit variance:



Given the input is white noise with zero mean $x(n) \sim \mathrm{WN}\left(0, \sigma_x^2\right)$, the autocorrelation of an AR(q) model is given by Eq. 13.141:

$$r_{\mathrm{yy}}(\ell) = -\sum_{k=1}^{p} a_k r_{\mathrm{yy}}(\ell - k), \quad \ell > 0$$

This is useful because we can use it to find the coefficients $\{a_k\}$ of an AR(q) model. Given an output signal $y(n)$, we can compute the autocorrelation $r_{\mathrm{yy}}(\ell)$ numerically in MATLAB. This means that the expression in Eq. 13.141 becomes a set of p linear equations.

For a second-order $p = 2$ AR model, we get two equations with two unknowns:

$$r_{\mathrm{yy}}(1) = -a_1 r_{\mathrm{yy}}(0) - a_2 r_{\mathrm{yy}}(-1)$$

$$r_{\mathrm{yy}}(2) = -a_1 r_{\mathrm{yy}}(1) - a_2 r_{\mathrm{yy}}(0)$$

We can write it into matrix form:

$$\begin{bmatrix} r_{\mathrm{yy}}(0) & r_{\mathrm{yy}}(1) \\ r_{\mathrm{yy}}(1) & r_{\mathrm{yy}}(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = -\begin{bmatrix} r_{\mathrm{yy}}(1) \\ r_{\mathrm{yy}}(2) \end{bmatrix}$$

Solving it will give us the AR(2) model coefficients $\{a_1, a_2\}$.

This is implemented in the arfit() function:

```
y = [1, -1, 0, 2, 3, 2, -4, 1];
[a, ~] = arfit(y, 2)
```

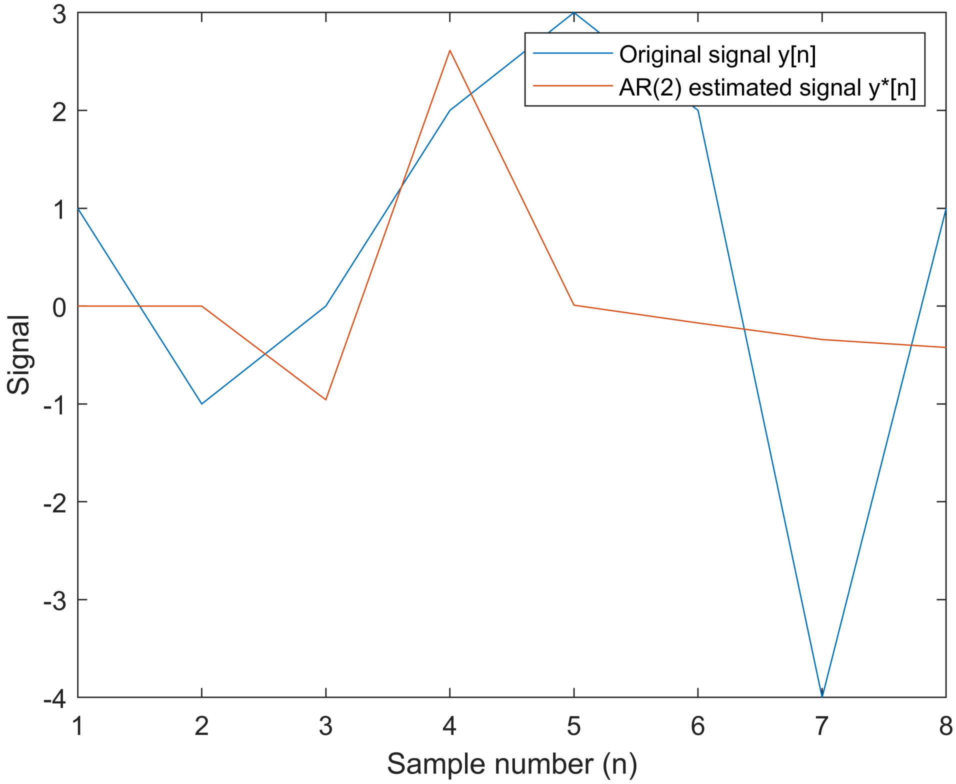
```
a = 2×1
    0.0340
    0.2232
```

The coefficients for the AR(2) model is; $a_1 = 0.0340$ and $a_2 = 0.2232$.

$$y(n) = -[0.0340y(n-1) + 0.2232y(n-2)] + b_0x(n)$$

We can plot the signal and the estimated AR(2) signal:

```
N = 8;

% Generate the input
x = randn(N, 1)'; % Random noise with zero mean and unit variance.
b0 = 1;

% Estimate the output signal using the AR(2) model
y_hat = zeros(N, 1);
n = 3:N; % Trick to make it work in MATLAB
y_hat(n) = -(a(1)*y(n-1) + a(2)*y(n-2)) + b0*x(n);

% Plot the two
n = 1:N;
plot(n, y(n), n, y_hat(n))
legend('Original signal y[n]', 'AR(2) estimated signal y*[n]');
xlabel('Sample number (n)')
ylabel('Signal')
```



Exam 2016 Problem 1: AR(p) signal modelling

The first few values of the autocorrelation function of a random process has been determined as

$\lvert l \rvert$	$r_x(l)$
0	11.26
1	9.70
2	6.00
3	1.49
4	-2.53

Initially, it can be assumed that the random process can be modeled as an AR(2) process.

```
clear variables;
```

1) Compute AR(2) model coefficient given autocorrelation sequence

1. Calculate the model parameters and plot the power spectral density.

An AR(p) model is given by:

$$y(n) = -\sum_{k=1}^{p} [a_k\, y(n-k)] + b_0 x(n)$$

The autocorrelation of AR(q) model was derived in Eq. 13.141 as:

$$r_{\text{yy}}(\ell) = -\sum_{k=1}^{p} a_k r_{\text{yy}}(\ell - k), \quad \ell > 0$$

This is useful because we can use it to find the coefficients $\{a_k\}$ of an AR(q) model using the autocorrelation $r_{\text{yy}}(\ell)$ computed numerically in MATLAB. This means that the expression in Eq. 13.141 becomes a set of p linear equations.

An AR(2) model is given by:

$$y(n) = -(a_1\, y(n-1) + a_2\, y(n-2)) + b_0 x(n)$$

We can estimate the model parameters of a second-order AR model $p = 2$ by creating two equations with two unknowns:

$$r_{\text{yy}}(1) = -a_1 r_{\text{yy}}(0) - a_2 r_{\text{yy}}(-1)$$

$$r_{\text{yy}}(2) = -a_1 r_{\text{yy}}(1) - a_2 r_{\text{yy}}(0)$$

We can write it into matrix form using the Toeplitz matrix:

$$\begin{bmatrix} r_{\text{yy}}(0) & r_{\text{yy}}(-1) \\ r_{\text{yy}}(1) & r_{\text{yy}}(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = -\begin{bmatrix} r_{\text{yy}}(1) \\ r_{\text{yy}}(2) \end{bmatrix}$$

In the general case, it becomes:

$$\begin{bmatrix} r_{yy}[0] & r_{yy}[1] & \dots & r_{yy}[p-1] \\ r_{yy}[1] & r_{yy}[0] & \dots & r_{yy}[p-2] \\ \vdots & \vdots & \ddots & \vdots \\ r_{yy}[p-1] & r_{yy}[p-2] & \dots & r_{yy}[0] \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = -\begin{bmatrix} r_{yy}[1] \\ r_{yy}[2] \\ \vdots \\ r_{yy}[p] \end{bmatrix},$$

More compactly written as *Yule–Walker equations*:

$$\boldsymbol{R}_y \boldsymbol{a} = -\boldsymbol{r}_y. \qquad (13.143)$$

The input noise variance can be compute as follows:

$$\sigma_x^2 = \sigma_y^2 + \boldsymbol{a}^T\boldsymbol{r}_y = \sigma_y^2 - \boldsymbol{r}_y^T\boldsymbol{R}_y^{-1}\boldsymbol{r}_y \leq \sigma_y^2. \qquad (13.144)$$

This is coded in MATLAB function (see end of document):

```
p = 2;
r_xx = [11.26, 9.70, 6.00, 1.49, -2.53]';
[a, v] = ar_from_acrs(r_xx, p)
```

```
a = 2×1
   -1.5604
    0.8114
v = 0.9922
```

The power spectrum of an ARMA(p, q) process is given by:

$$S_{yy}(\omega) = \sigma_x^2\left|H(\mathrm{e}^{\mathrm{j}\omega})\right|^2 = \sigma_x^2\left|\frac{\sum_{k=0}^{q} b_k \mathrm{e}^{-\mathrm{j}\omega k}}{1+\sum_{k=1}^{p} a_k \mathrm{e}^{-\mathrm{j}\omega k}}\right|^2. \qquad (13.133)$$

The power spectrum of an AR(p) process is given by:

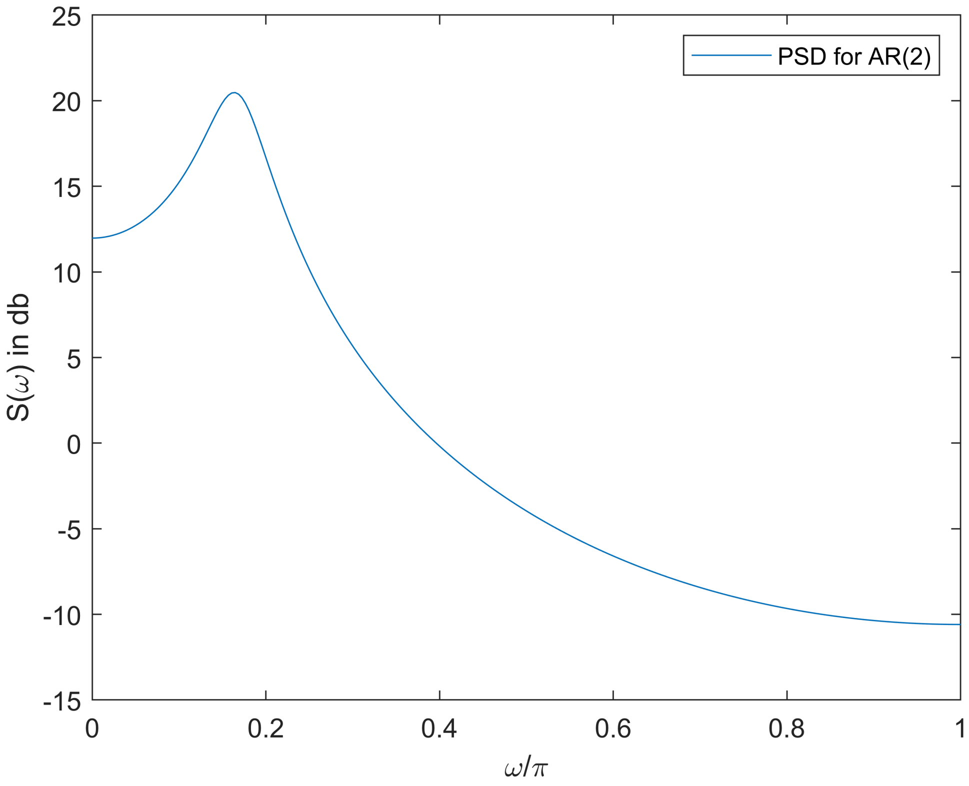
$$S_{yy}(\omega) = \sigma_x^2\left|\frac{1}{1+\sum_{k=1}^{p} a_k e^{-j\omega k}}\right|^2$$

To solve this problem, the method is as follows:

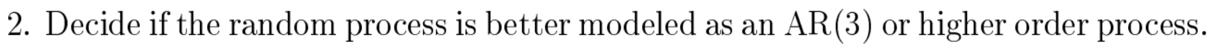
1. Find the coefficents $\{a_1, a_2, \cdots, a_p\}$ for the $\mathrm{AR}(p)$ model
2. Compute the transfer function for the $\mathrm{AR}(p)$ by computing the sum and finding its reciprocal
3. Compute the conjugate of the transfer function: $|H(e^{j\omega})|^2$
4. Multiply it with the variance σ_x^2

The method above is implemented in the function `ar2psd` (see at the end of document):

```
[S, w] = ar2psd(a, v, 256);
plot(w/pi, real(pow2db(S)))
legend('PSD for AR(2)')
xlabel('\omega/\pi')
ylabel('S(\omega) in db')
```



2) Is the random process better modelled as an AR(3) or higher-order process?


2. Decide if the random process is better modeled as an AR(3) or higher order process.

The variane σ_x^2 calculated above corresponds to the minimum Mean Squared Error (MSE) for the AR(2) model. So for a higher order model to be a better model the MSE must be lower than for the AR(2) model.

We can solve the equations for different AR(q) models:

```
MSE_ar2 = v
```

```
MSE_ar2 = 0.9922
```

```
[~, MSE_ar3] = ar_from_acrs(r_xx, 3)
```

```
MSE_ar3 = 0.9922
```

```
[~, MSE_ar4] = ar_from_acrs(r_xx, 4)
```

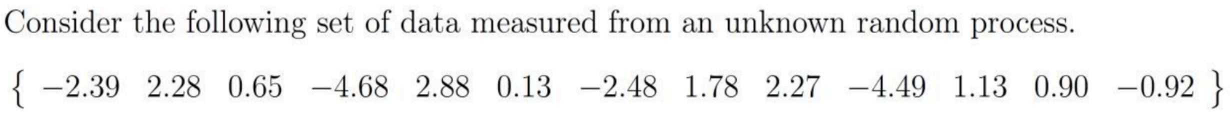
```
MSE_ar4 = 0.9919
```

The MSE is lowered very slightly when the model order is increased.

Because of the slight change we can be conclude that the process is not better modeled as an AR(3) or AR(4) process.

Exam 2018 Problem 2: Signal Modelling using MA(2) or AR(2)

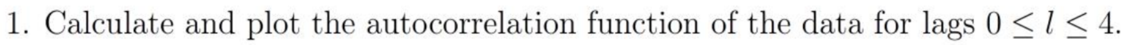
```
clear variables;
```



Consider the following set of data measured from an unknown random process.

$$\{ \; -2.39 \quad 2.28 \quad 0.65 \quad -4.68 \quad 2.88 \quad 0.13 \quad -2.48 \quad 1.78 \quad 2.27 \quad -4.49 \quad 1.13 \quad 0.90 \quad -0.92 \; \}$$

[✔] 1) Calculate and plot the autocorrelation function of the data

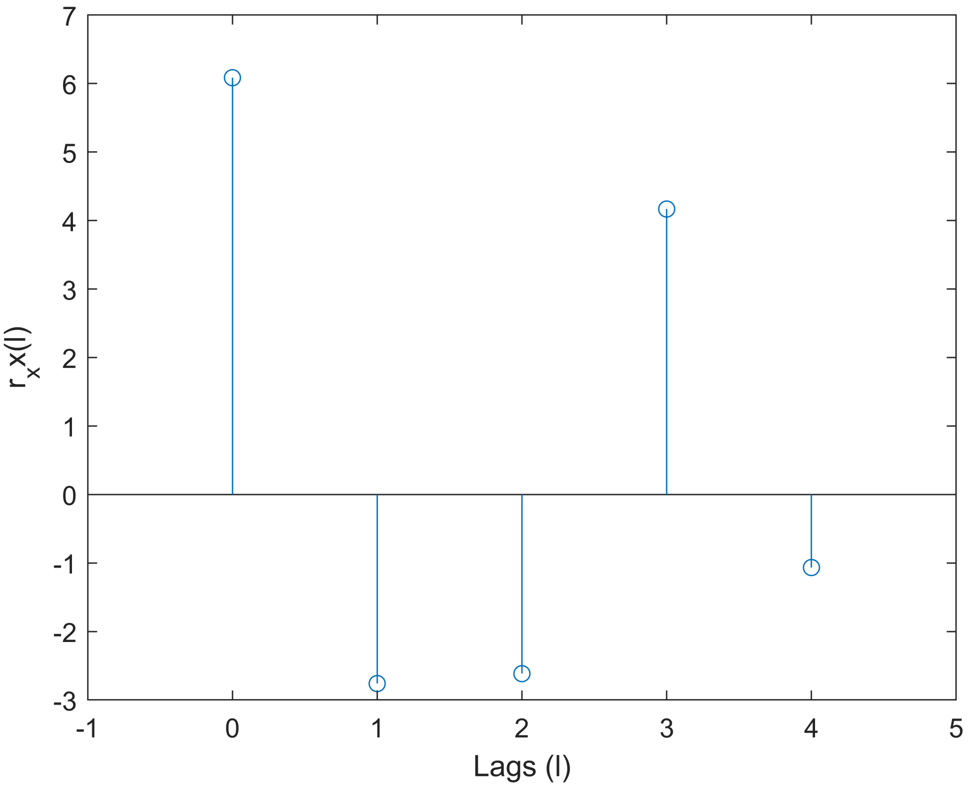


1. Calculate and plot the autocorrelation function of the data for lags $0 \leq l \leq 4$.

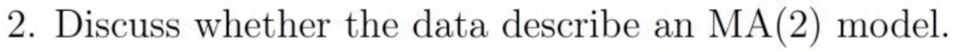
```
x = [-2.39, 2.28, 0.65, -4.68, 2.88, 0.13, -2.48, 1.78, 2.27, -4.49, 1.13, 0.9, -0.92];

% Estimate autocorrelation using data
[r_xx, ell] = xcorr(x, 'biased');

% Print out the lags 0 to 4
mid = floor(numel(ell)/2)+1;
r_xx(mid:mid+4)
```

```
ans = 1×5
    6.0855   -2.7605   -2.6160    4.1680   -1.0665
```

```
% Plot the results
stem(0:4, r_xx(mid:mid+4))
xlim([-1, 5])
xlabel('Lags (l)')
ylabel('r_xx(l)')
```



2) Can data be described by an MA(2) model?



2. Discuss whether the data describe an MA(2) model.

MA(2) model is given by:

$$y[n] = b_0\, x[n] + b_1\, x[n-1] + b_2\, x[n-2]$$

where $x[n] \sim \mathrm{WN}\left(0, \sigma_x^2\right)$

The autocorrelation for an MA(2) process is:

$$r_{yy}(\ell) = \left(b_0^2 + b_1^2 + b_2^2\right)\delta(\ell) + (b_0 b_1 + b_1 b_2)\delta(\ell+1) + (b_0 b_1 + b_1 b_2)\delta(\ell-1) + b_0 b_2 \delta(\ell+2) + b_0 b_2 \delta(\ell-2)$$

Clearly, the autocorrelation for an MA(2) for $\ell > 2$ is zero.

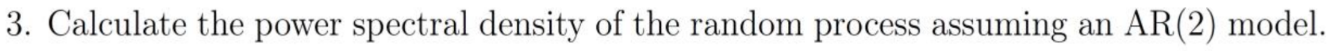
The estimated autocorrelation for the data is:

```
r_xx(mid:mid+4)
```

```
ans = 1×5
    6.0855   -2.7605   -2.6160    4.1680   -1.0665
```

Since the estimated autocorrelation has non-zero values for $\ell > 2$, it cannot be described by an MA(2) model.

3) Calculate the power spectral density of the random process assuming an AR(2) model



3. Calculate the power spectral density of the random process assuming an AR(2) model.

Before we can compute the PSD, we need to find the AR(2) model coefficients.

An AR(p) model is given by:

$$y(n) = -\sum_{k=1}^{p} [a_k\, y(n-k)] + b_0 x(n)$$

The autocorrelation of AR(q) model was derived in Eq. 13.141 as:

$$r_{\mathrm{yy}}(\ell) = -\sum_{k=1}^{p} a_k r_{\mathrm{yy}}(\ell - k), \quad \ell > 0$$

This is useful because we can use it to find the coefficients $\{a_k\}$ of an AR(q) model using the autocorrelation $r_{\mathrm{yy}}(\ell)$ computed numerically in MATLAB. This means that the expression in Eq. 13.141 becomes a set of p linear equations.

An AR(2) model is given by:

$$y(n) = -(a_1\, y(n-1) + a_2\, y(n-2)) + b_0 x(n)$$

We can estimate the model parameters of a second-order AR model $p = 2$ by creating two equations with two unknowns:

$$r_{\mathrm{yy}}(1) = -a_1 r_{\mathrm{yy}}(0) - a_2 r_{\mathrm{yy}}(-1)$$

$$r_{\mathrm{yy}}(2) = -a_1 r_{\mathrm{yy}}(1) - a_2 r_{\mathrm{yy}}(0)$$

We can write it into matrix form using the Toeplitz matrix:

$$\begin{bmatrix} r_{\mathrm{yy}}(0) & r_{\mathrm{yy}}(-1) \\ r_{\mathrm{yy}}(1) & r_{\mathrm{yy}}(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = -\begin{bmatrix} r_{\mathrm{yy}}(1) \\ r_{\mathrm{yy}}(2) \end{bmatrix}$$

In the general case, it becomes:

$$\begin{bmatrix} r_{yy}[0] & r_{yy}[1] & \dots & r_{yy}[p-1] \\ r_{yy}[1] & r_{yy}[0] & \dots & r_{yy}[p-2] \\ \vdots & \vdots & \ddots & \vdots \\ r_{yy}[p-1] & r_{yy}[p-2] & \dots & r_{yy}[0] \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = -\begin{bmatrix} r_{yy}[1] \\ r_{yy}[2] \\ \vdots \\ r_{yy}[p] \end{bmatrix},$$

More compactly written as:

$$\boldsymbol{R}_y \boldsymbol{a} = -\boldsymbol{r}_y$$

This is a system of linear equations which can be solved in MATLAB using `arfit` function (see bottom of the document).

The power spectrum of an ARMA(p, q) process is given by:

$$S_{yy}(\omega) = \sigma_x^2 \left|H(\mathrm{e}^{\mathrm{j}\omega})\right|^2 = \sigma_x^2 \left| \frac{\sum_{k=0}^{q} b_k \mathrm{e}^{-\mathrm{j}\omega k}}{1 + \sum_{k=1}^{p} a_k \mathrm{e}^{-\mathrm{j}\omega k}} \right|^2 . \tag{13.133}$$

The power spectrum of an AR(p) process is given by:

$$S_{\mathrm{yy}}(\omega) = \sigma_x^2 \left| \frac{1}{1 + \sum_{k=1}^{p} a_k e^{-j\omega k}} \right|^2$$

To solve this problem, the method is as follows:

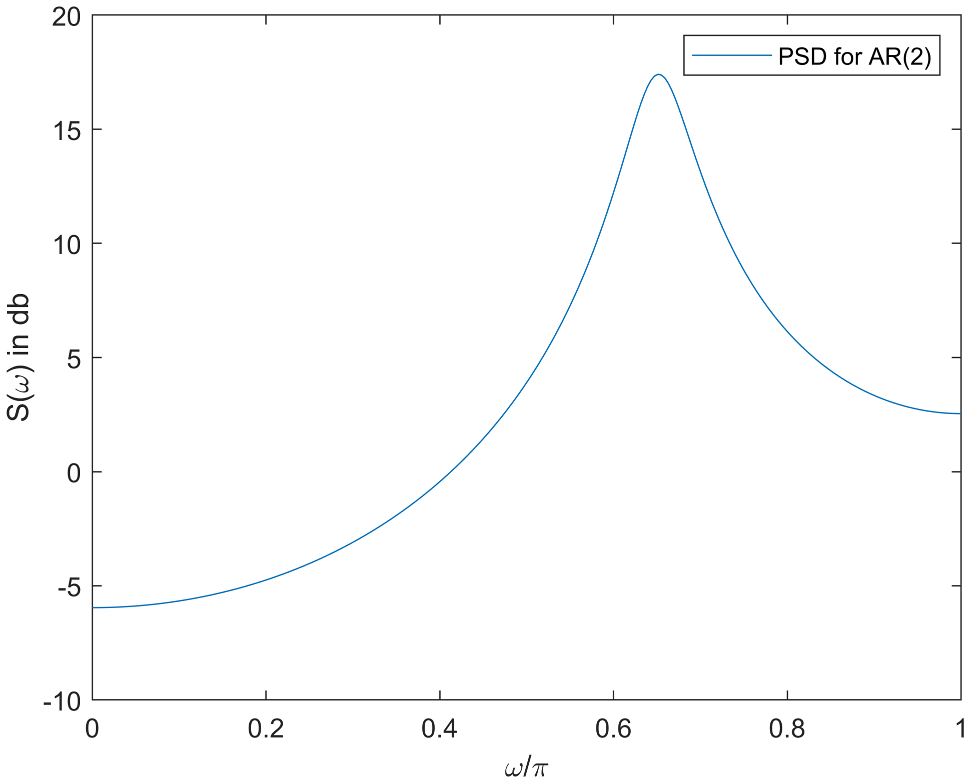
1. Find the coefficents $\{a_1, a_2, \cdots, a_p\}$ for the $\mathrm{AR}(p)$ model
2. Compute the transfer function for the $\mathrm{AR}(p)$ by computing the sum and finding its reciprocal
3. Compute the conjugate of the transfer function: $|H(e^{j\omega})|^2$
4. Multiply it with the variance σ_x^2

The method above is implemented in the functions `arfit()` and `ar2psd()`:

```
[a, v] = arfit(x, 2)
```

```
a = 2×1
    0.8167
    0.8003
v = 1.7374
```

```
[S, w] = ar2psd(a, v, 256);
plot(w/pi, real(pow2db(S)))
legend('PSD for AR(2)')
xlabel('\omega/\pi')
ylabel('S(\omega) in db')
```



Problem 1: Signal modelling assuming AR(2) process

The first few values of the autocorrelation function of a random process has been determined as

$\lvert l \rvert$	$r_x(l)$
0	11.26
1	9.70
2	6.00
3	1.49
4	-2.53

Initially, it can be assumed that the random process can be modeled as an AR(2) process.

```
clear variables;
```

1) Compute AR(2) model coefficient given autocorrelation sequence

1. Calculate the model parameters and plot the power spectral density.

An AR(p) model is given by:

$$y(n) = -\sum_{k=1}^{p} [a_k\, y(n-k)] + b_0 x(n)$$

The autocorrelation of AR(q) model was derived in Eq. 13.141 as:

$$r_{\mathrm{yy}}(\ell) = -\sum_{k=1}^{p} a_k r_{\mathrm{yy}}(\ell - k), \quad \ell > 0$$

This is useful because we can use it to find the coefficients $\{a_k\}$ of an AR(q) model using the autocorrelation $r_{\mathrm{yy}}(\ell)$ computed numerically in MATLAB. This means that the expression in Eq. 13.141 becomes a set of p linear equations.

An AR(2) model is given by:

$$y(n) = -(a_1\, y(n-1) + a_2\, y(n-2)) + b_0 x(n)$$

We can estimate the model parameters of a second-order AR model $p = 2$ by creating two equations with two unknowns:

$$r_{\mathrm{yy}}(1) = -a_1 r_{\mathrm{yy}}(0) - a_2 r_{\mathrm{yy}}(-1)$$

$$r_{\mathrm{yy}}(2) = -a_1 r_{\mathrm{yy}}(1) - a_2 r_{\mathrm{yy}}(0)$$

We can write it into matrix form using the Toeplitz matrix:

$$\begin{bmatrix} r_{\mathrm{yy}}(0) & r_{\mathrm{yy}}(-1) \\ r_{\mathrm{yy}}(1) & r_{\mathrm{yy}}(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = -\begin{bmatrix} r_{\mathrm{yy}}(1) \\ r_{\mathrm{yy}}(2) \end{bmatrix}$$

In the general case, it becomes:

$$\begin{bmatrix} r_{yy}[0] & r_{yy}[1] & \dots & r_{yy}[p-1] \\ r_{yy}[1] & r_{yy}[0] & \dots & r_{yy}[p-2] \\ \vdots & \vdots & \ddots & \vdots \\ r_{yy}[p-1] & r_{yy}[p-2] & \dots & r_{yy}[0] \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = -\begin{bmatrix} r_{yy}[1] \\ r_{yy}[2] \\ \vdots \\ r_{yy}[p] \end{bmatrix},$$

More compactly written as *Yule–Walker equations*:

$$\boldsymbol{R}_y \boldsymbol{a} = -\boldsymbol{r}_y. \qquad (13.143)$$

The input noise variance can be compute as follows:

$$\sigma_x^2 = \sigma_y^2 + \boldsymbol{a}^T \boldsymbol{r}_y = \sigma_y^2 - \boldsymbol{r}_y^T \boldsymbol{R}_y^{-1} \boldsymbol{r}_y \le \sigma_y^2. \qquad (13.144)$$

This is coded in MATLAB function (see end of document):

```
p = 2;
```

```
r_xx = [11.26, 9.70, 6.00, 1.49, -2.53]';
[a, v] = ar_from_acrs(r_xx, p)
```

```
a = 2×1
   -1.5604
    0.8114
v = 0.9922
```

The power spectrum of an ARMA(p, q) process is given by:

$$S_{yy}(\omega) = \sigma_x^2 \left|H(\mathrm{e}^{\mathrm{j}\omega})\right|^2 = \sigma_x^2 \left| \frac{\sum_{k=0}^{q} b_k \mathrm{e}^{-\mathrm{j}\omega k}}{1+\sum_{k=1}^{p} a_k \mathrm{e}^{-\mathrm{j}\omega k}} \right|^2 . \qquad (13.133)$$

The power spectrum of an AR(p) process is given by:

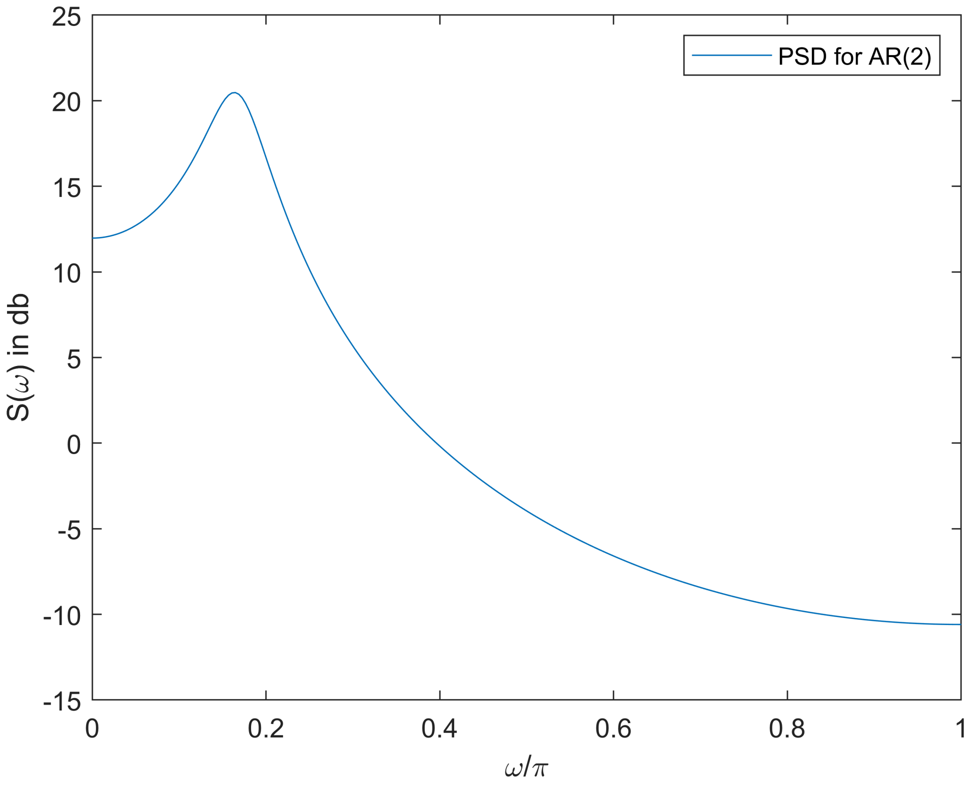
$$S_{\mathrm{yy}}(\omega) = \sigma_x^2 \left| \frac{1}{1+\sum\limits_{k=1}^{p} a_k e^{-j\omega k}} \right|^2$$

To solve this problem, the method is as follows:

1. Find the coefficents $\{a_1, a_2, \cdots, a_p\}$ for the $\mathrm{AR}(p)$ model
2. Compute the transfer function for the $\mathrm{AR}(p)$ by computing the sum and finding its reciprocal
3. Compute the conjugate of the transfer function: $|H(e^{j\omega})|^2$
4. Multiply it with the variance σ_x^2

The method above is implemented in the function `ar2psd` (see at the end of document):

```
[S, w] = ar2psd(a, v, 256);
plot(w/pi, real(pow2db(S)))
legend('PSD for AR(2)')
xlabel('\omega/\pi')
ylabel('S(\omega) in db')
```



2) Is the random process better modelled as an AR(3) or higher-order process?

2. Decide if the random process is better modeled as an AR(3) or higher order process.

The variane σ_x^2 calculated above corresponds to the minimum Mean Squared Error (MSE) for the AR(2) model. So for a higher order model to be a better model the MSE must be lower than for the AR(2) model.

We can solve the equations for different AR(q) models:

```
MSE_ar2 = v
```

```
MSE_ar2 = 0.9922
```

```
[~, MSE_ar3] = ar_from_acrs(r_xx, 3)
```

```
MSE_ar3 = 0.9922
```

```
[~, MSE_ar4] = ar_from_acrs(r_xx, 4)
```

```
MSE_ar4 = 0.9919
```

The MSE is lowered very slightly when the model order is increased.

Because of the slight change we can be conclude that the process is not better modeled as an AR(3) or AR(4) process.

Functions

```
function [S, w] = ar2psd(a, v, N)
% AR2PSD Compute the Power Spectral Density from AR(p) coefficients
%   [S, w] = ar2psd(a, v, N)
% a: AR(p) coefficients
% v: the variance
% N: number of points in the range [1, pi]
% S: the estimated power spectrum
% w: frequencies
    w = linspace(0, 1, N) * pi;

    % Compute the transfer function
    % Used Eq. (13.133) in the book
    H = ones(N, 1);
    for k=1:numel(a)
        H = H + a(k)*exp(-1j * w' * k);
    end
    H = 1./H;

    % Finally compute the PSD
    S = v * H.*conj(H);
end

function [a,v] = arfit(x,p)
    % fit AR(p) model from data
    % x: data
    % p: model order
    % a: a coefficients
    % v: variance
    if isrow(x)
        x = x'; % Convert to a column vector
    end

    % Compute the autocorrelation
    [r_xx, lags] = xcorr(x, p, 'biased');

    % Select elements r_xx[0] to r_xx[p-1]
    R_elems = r_xx(p+1:2*p);

    % Create the Toeplitz matrix
    R = toeplitz(R_elems);

    % Select elements r_xx[1] to r_xx[p]
    r = r_xx(p+2:2*p+1);

    % Solve systems of linear equations using mldivide function
    a = mldivide(R, -r);
    % Compute the variance
    v = r_xx(p+1) + a'*r;
end


function [a, v] = ar_from_acrs(r_xx, p)
    R_xx = toeplitz(r_xx(1:p));

    % Select elements r_xx[1] to r_xx[p]
    r = r_xx(2:p+1);

    % Solve Yule-Walker equation (13.143)
    a = mldivide(R_xx, -r);

    % Compute the variance according to Eq. (13.144)
    v = r_xx(1) + a'*r;
end
```