Pisarenko

Table of Contents

Difference Equations of Sinusoids

A signal consisting of p sinusoidal components has the difference equation:

$$x(n) = -\sum_{m=1}^{2p} a_m x(n-m) \qquad (14.5.2)$$

This corresponds to a system with the system function:

$$H(z) = \frac{1}{1 + \sum_{m=1}^{2p} a_m z^{-m}} \qquad (14.5.3)$$

From the polynomial $A(z)$, we observe that the system has $2p$ poles on the unit circle which correspond to the frequencies of the sinusoids

$$A(z) = 1 + \sum_{m=1}^{2p} a_m z^{-m} \qquad (14.5.4)$$

Now, suppose that the sinusoids are corrupted by a white noise sequence *w(n)* with zero mean and variance σ_w^2:

$$y(n) = x(n) + w(n) \qquad (14.5.5)$$

The difference equation for (14.5.5) is an $\mathrm{ARMA}(2p, 2p)$ process that can be expressed in matrix form:

$$\mathbf{Y}^t\mathbf{a} = \mathbf{W}^t\mathbf{a} \qquad (14.5.7)$$

where:

- $\boldsymbol{Y}^t = [y(n) \;\; y(n-1) \;\; \cdots \;\; y(n-2p)]$ is the observed data vector of size $2p+1$
- $\boldsymbol{W}^t = [w(n) \;\; w(n-1) \;\; \cdots \;\; w(n-2p)]$ is the noise vector of size $2p+1$
- $\boldsymbol{a} = [1 \;\; a_1 \;\; \cdots a_{2p}]$ is the coefficients vector

If we multiply (14.5.7) by $\boldsymbol{Y}$ and take the expected value, we obtain the following:

$$(\boldsymbol{\Gamma}_{yy} - \sigma_w^2\mathbf{I})\mathbf{a} = \mathbf{0} \qquad (14.5.9)$$

where:

- Γ_{yy} is the autocorrelation matrix
- σ_w^2 is an eigenvalue of the autocorrelation matrix
- $\boldsymbol{a}$ is the eigenvector associated with the eigenvalue σ_w^2

Pisarenko: harmonic decomposition method

The Pisarenko method can be used to recover the sinusoidal frequencies of a corrupted signal $x(n)$ given two assumptions:

- The signal $x(n)$ consists of p sinusoids that has been corrupted by white noise.
- The autocorrelation matrix of size $(p+1) \times (p+1)$ is known or can be estimated from data

The Pisarenko method consists of the following steps:

Step 1: Compute the autocorrelation matrix $\mathbf{R}_{\mathrm{xx}}$

Step 2: Find the eigenvector corresponding to the smallest minimum eigenvalue. The elements of this eigenvector is the parameters of the $\mathrm{ARMA}(2p, 2p)$ model

Step 3: Find the frequencies $\{f_i\}$ of the sinusoids. This can be done by computing the roots of the polynomial $A(z)$ in Eq. (14.5.4) in the book. This polynomial has $2p$ poles on the unit circle which correspond to the frequencies of the system.

$$A(z) = 1 + \sum_{m=1}^{2p} a_m z^{-m} \qquad (14.5.4)$$

***Step 4*:** Solve Eq. (14.5.11) for the signal powers $\{P_i\}$

$$\begin{bmatrix} \cos 2\pi f_1 & \cos 2\pi f_2 & \cdots & \cos 2\pi f_p \\ \cos 4\pi f_1 & \cos 4\pi f_2 & \cdots & \cos 4\pi f_p \\ \vdots & \vdots & & \vdots \\ \cos 2\pi p f_1 & \cos 2\pi p f_2 & \cdots & \cos 2\pi p f_p \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_p \end{bmatrix} = \begin{bmatrix} \gamma_{yy}(1) \\ \gamma_{yy}(2) \\ \vdots \\ \gamma_{yy}(p) \end{bmatrix} \qquad (14.5.11)$$

where

- $\gamma_{\text{yy}}(1), \gamma_{\text{yy}}(2), \cdots, \gamma_{\text{yy}}(p)$ are the estimated autocorrelation values
- $P_i = \frac{A_i^2}{2}$ is the average power of the *i*th sinusoid and A_i is the corresponding amplitude

Step 5: Estimate the amplitude $A_i = \sqrt{2P_i}$

These steps are coded up in the `pisarenko()` function (see at end of this document).

Exam 2012, Problem 3: PSD Estimation assuming sinusoidal signal in white noise

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

$\lvert m \rvert$	$r_x(m)$
0	4
1	2
2	-1

```
clear variables;
```

1) Estimate PSD assuming sinusoidal white noise

Estimate the power density spectrum under the assumption that $\{x(n)\}$ consist of a single sinusoidal signal in additive white noise.

The Pisarenko method is used to estimate the power spectrum density of a random process.

The method makes two assumptions:

- The signal $x(n)$ consists of p sinusoids that has been corrupted by white noise.
- The autocorrelation matrix of size $(p+1)\times(p+1)$ is known or can be estimated

Given these assumptions, the Pisarenko method can recover the sinusoidal frequencies of the corrupted signal using the following steps:

Step 1: Compute the autocorrelation matrix $\mathbf{R}_{\text{yy}}$

Step 2: Find the eigenvector corresponding to the smallest minimum eigenvalue. The elements of this eigenvector is the parameters of the $\text{ARMA}(2p, 2p)$ model

Step 3: Find the frequencies $\{f_i\}$ of the sinusoids. This can be done by computing the roots of the polynomial $A(z)$ in (14.5.4). This polynomial has $2p$ poles on the unit circle which correspond to the frequencies of the system.

$$A(z) = 1 + \sum_{m=1}^{2p} a_m z^{-m} \qquad (14.5.4)$$

Step 4: Solve Eq. (14.5.11) for the signal powers $\{P_i\}$

$$\begin{bmatrix} \cos 2\pi f_1 & \cos 2\pi f_2 & \cdots & \cos 2\pi f_p \\ \cos 4\pi f_1 & \cos 4\pi f_2 & \cdots & \cos 4\pi f_p \\ \vdots & \vdots & & \vdots \\ \cos 2\pi p f_1 & \cos 2\pi p f_2 & \cdots & \cos 2\pi p f_p \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_p \end{bmatrix} = \begin{bmatrix} \gamma_{yy}(1) \\ \gamma_{yy}(2) \\ \vdots \\ \gamma_{yy}(p) \end{bmatrix} \qquad (14.5.11)$$

where

- $\gamma_{\text{yy}}(1), \gamma_{\text{yy}}(2), \cdots, \gamma_{\text{yy}}(p)$ are the estimated autocorrelation values
- $P_i = \dfrac{A_i^2}{2}$ is the average power of the ith sinusoid and A_i is the corresponding amplitude

Step 5: Estimate the amplitude $A_i = \sqrt{2P_i}$

These steps are coded up in the `pisarenko()` function (see at end of this document):

```
r_xx = [4, 2, -1];
[F, A, P, lambda_min] = pisarenko(r_xx, 1)
```

```
F = 0.1490
A = 2.5970
P = 3.3723
```

```
lambda_min = 0.6277
```

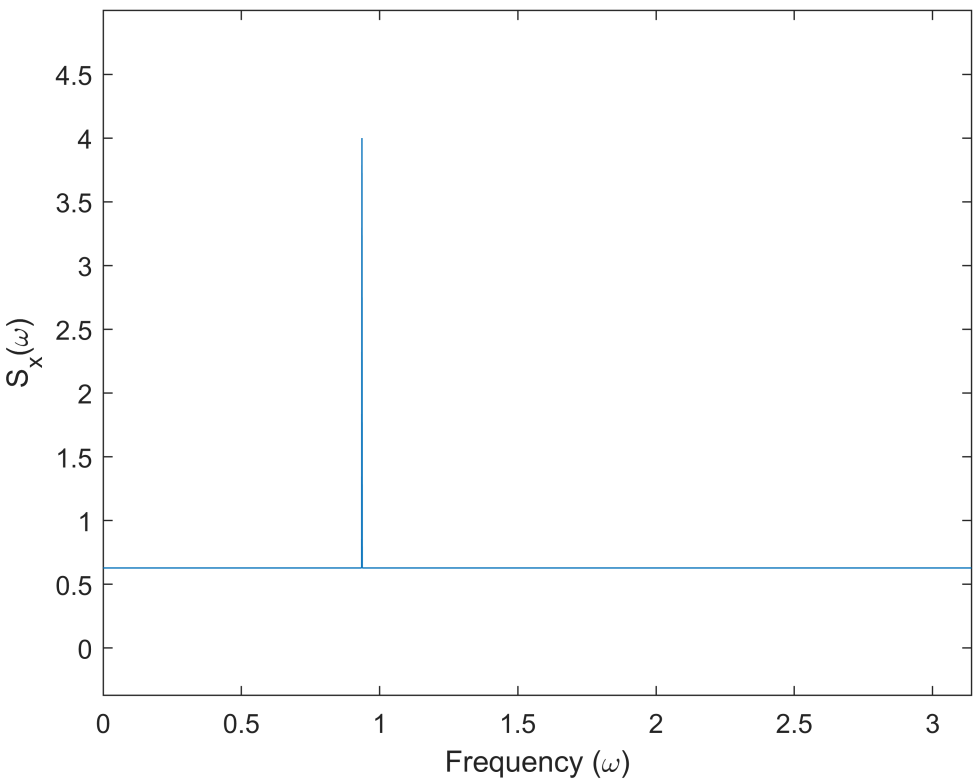
We can describe the signal as follows:

$$x(n) = 2.597 \cdot \cos(2\pi \cdot 0.149n + \phi) + w(n)$$

where $w(n)$ is white noise with variance $\sigma_w^2 = 0.6277$.

A sketch of the PSD is given as:

```
D = 3;
w = 0:10^-D:pi;
delta = @(n) round(n, D) == 0;
S = lambda_min*ones(1, numel(w)) + P*delta(w - 2*pi*F);
plot(w, S)
xlim([0, pi])
ylim([lambda_min-1, lambda_min+P+1])
xlabel('Frequency (\omega)')
ylabel('S_x(\omega)')
```



In ADSI Problem 4.4, we found that the autocorrelation of a real sinusoid given by $y(n) = A\cos(\omega n + \phi)$ where A and ω are real constants and ϕ is a random variable with $\phi \sim U(0, 2\pi)$ is:

$$r_{yy}(\ell) = \frac{A^2}{2}\cos(\omega\ell)$$

The autocorrelation function of white noise with variance σ_w^2 is given by:

$$r_{\mathrm{ww}}(\ell) = \sigma_w^2 \delta(\ell)$$

Using these results, we have the autocorrelation function of the signal:

$$r_{\mathrm{xx}}(\ell) = \frac{A^2}{2}\cos(\omega\ell) + \sigma_w^2 \delta(\ell)$$

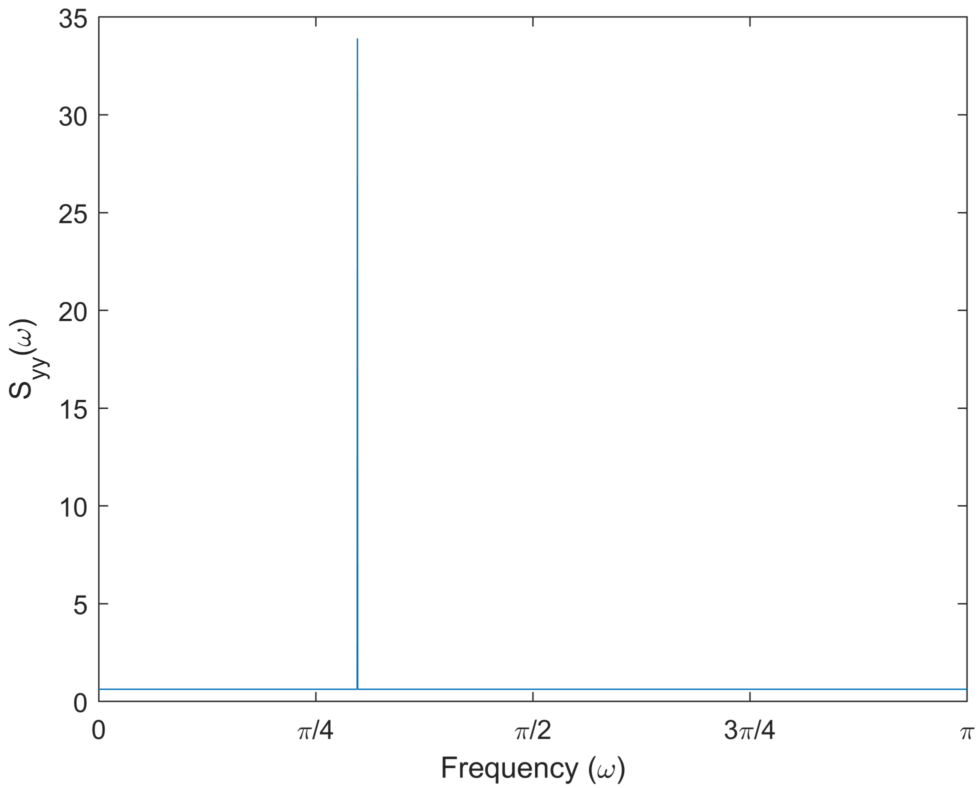
```
ell = 0:2;
(A^2/2) * cos(2*pi*F*ell) + lambda_min * (ell == 0)
```

```
ans = 1×3
    4.0000    2.0000   -1.0000
```

The power spectral density is the Fourier transform the autocorrelation function:

$$S(\omega) = \frac{A^2}{2}\pi[\delta(\omega - \omega_0) + \delta(\omega + \omega_0)] + \sigma_w^2$$

```
D = 3;
w=0:10^-(D):pi;
delta = @(n) round(n, D) == 0;
S = (A^2*pi)/2 * pi * (delta(w-2*pi*F) + delta(w+2*pi*F)) + lambda_min;
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])
```



Exam 2015 Problem 1: Estimate PSD of signal using Pisarenko

The autocorrelation function of an unknown signal is estimated as

$\lvert l \rvert$	$r_x(l)$
0	17
1	4
2	5

```
clear variables;
```

1) Estimate and sketch the PSD of sinusoidal signal in white noise

1. Estimate and sketch the power spectral density of the signal under the assumption that the signal is a single sinusoidal signal in white noise.

The Pisarenko method can be used to recover the sinusoidal frequencies of a corrupted signal $x(n)$ given two assumptions:

- The signal $x(n)$ consists of p sinusoids that has been corrupted by white noise.

- The autocorrelation matrix of size $(p+1) \times (p+1)$ is known or can be estimated from data

The Pisarenko method consists of the following steps:

Step 1: Compute the autocorrelation matrix $\mathbf{R}_{xx}$

Step 2: Find the eigenvector corresponding to the smallest minimum eigenvalue. The elements of this eigenvector is the parameters of the $\text{ARMA}(2p, 2p)$ model

Step 3: Find the frequencies $\{f_i\}$ of the sinusoids. This can be done by computing the roots of the polynomial $A(z)$ in Eq. (14.5.4) in the book. This polynomial has $2p$ poles on the unit circle which correspond to the frequencies of the system.

$$A(z) = 1 + \sum_{m=1}^{2p} a_m z^{-m} \qquad (14.5.4)$$

Step 4: Solve Eq. (14.5.11) for the signal powers $\{P_i\}$

$$\begin{bmatrix} \cos 2\pi f_1 & \cos 2\pi f_2 & \cdots & \cos 2\pi f_p \\ \cos 4\pi f_1 & \cos 4\pi f_2 & \cdots & \cos 4\pi f_p \\ \vdots & \vdots & & \vdots \\ \cos 2\pi p f_1 & \cos 2\pi p f_2 & \cdots & \cos 2\pi p f_p \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_p \end{bmatrix} = \begin{bmatrix} \gamma_{yy}(1) \\ \gamma_{yy}(2) \\ \vdots \\ \gamma_{yy}(p) \end{bmatrix} \qquad (14.5.11)$$

where

- $\gamma_{yy}(1), \gamma_{yy}(2), \cdots, \gamma_{yy}(p)$ are the estimated autocorrelation values
- $P_i = \frac{A_i^2}{2}$ is the average power of the ith sinusoid and A_i is the corresponding amplitude

Step 5: Estimate the amplitude $A_i = \sqrt{2P_i}$

These steps are coded up in the `pisarenko()` function (see at end of this document).

```
r_xx = [17, 4, 3]; % In the solution r_x(2)=3
[F, A, P, lambda_min] = pisarenko(r_xx, 1)
```

```
F = 0.0645
A = 2.9504
P = 4.3523
lambda_min = 12.6477
```

We can describe the signal as:

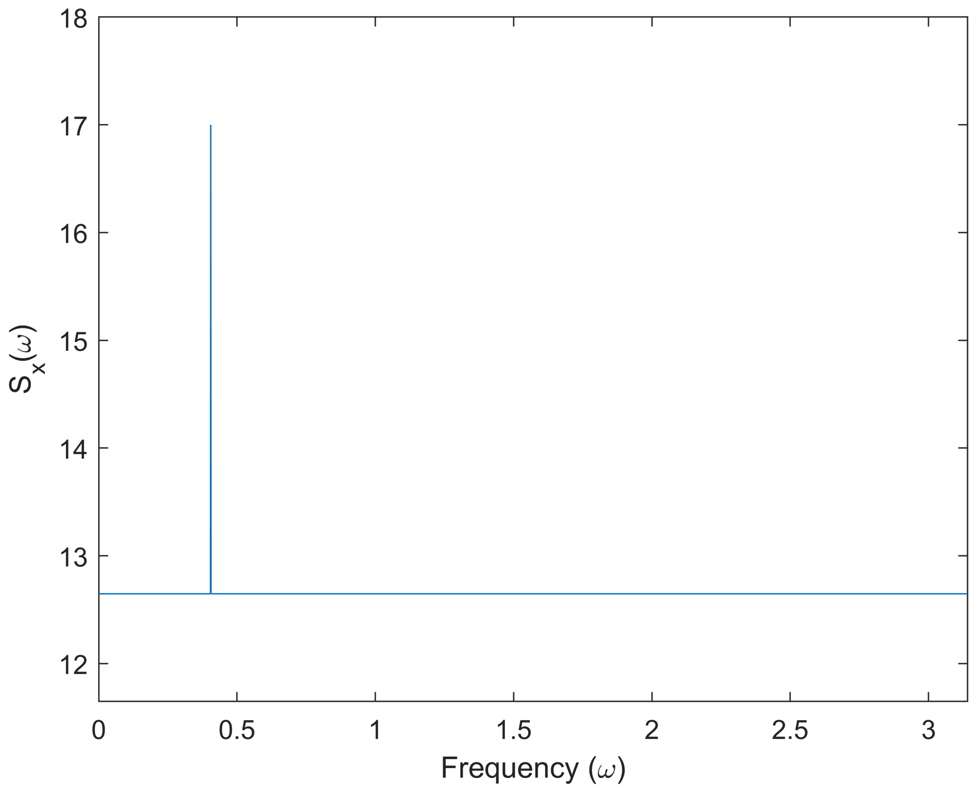
$$x(n) = 2.9504 \cos(2\pi \cdot 0.0645 + \phi) + w(n)$$

where $w(n)$ is white noise with variance $\sigma_w^2 = 12.6477$

A sketch of the PSD is given as:

```
D = 3;
```

```
w = 0:10^-D:pi;
delta = @(n) round(n, D) == 0;
S = lambda_min*ones(1, numel(w)) + P*delta(w - 2*pi*F);
plot(w, S)
xlim([0, pi])
ylim([lambda_min-1, lambda_min+P+1])
xlabel('Frequency (\omega)')
ylabel('S_x(\omega)')
```



To compute the power spectral density, we need to find the autocorrelation function of the sinusoid.

In ADSI Problem 4.4, we found that the autocorrelation of a real sinusoid given by $y(n) = A\cos(\omega n + \phi)$ where A and ω are real constants and ϕ is a random variable with $\phi \sim U(0, 2\pi)$ is:

$$r_{yy}(\ell) = \frac{A^2}{2}\cos(\omega\ell)$$

The autocorrelation function of white noise with variance σ_w^2 is given by:

$$r_{ww}(\ell) = \sigma_w^2\delta(\ell)$$

Combining these results, the general autocorrelation function of our signal is:

$$r_{xx}(\ell) = \frac{A^2}{2}\cos(\omega\ell) + \sigma_w^2\delta(\ell)$$

The power spectral density is the Fourier transform the autocorrelation function which is given by:

$$S(\omega) = \frac{A^2}{2}\pi[\delta(\omega - \omega_0) + \delta(\omega + \omega_0)] + \sigma_w^2$$

```
A, F, lambda_min
```

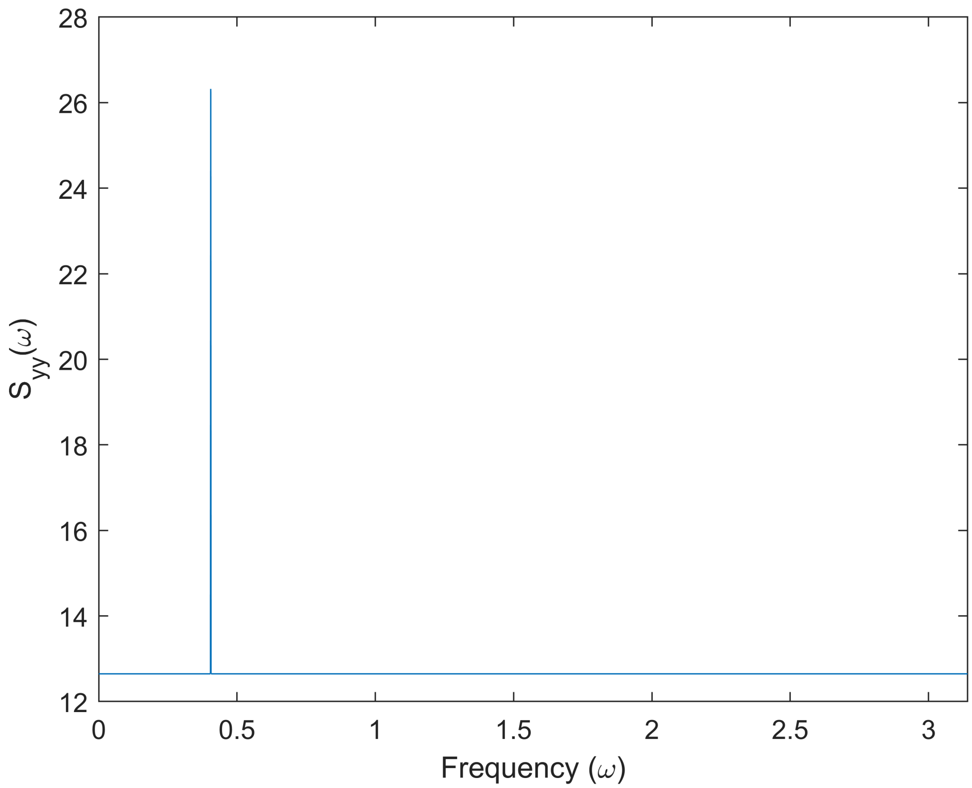
```
A = 2.9504
F = 0.0645
lambda_min = 12.6477
```

Using the values found via the Pisarenko, the PSD is:

$$S(\omega) = \frac{(2.9504)^2}{2}\pi[\delta(\omega - 0.0645 \cdot 2\pi) + \delta(\omega + 0.0645 \cdot 2\pi)] + 12.6477$$

We can plot the PSD:

```
D = 3;
w = 0:10^(-D):pi;
delta = @(n) round(n, D) == 0;
S = (A^2/2)*pi * (delta(w-(F*2*pi)) + delta(w+(F*2*pi))) + lambda_min;
%plot(w/pi, real(pow2db(S)))
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])
```



2) Calculate the signal to noise ratio

2. Calculate the signal to noise ratio.

The signal to noise ratio is the power of the signal divided by the power of the noise:

$$\text{SNR} = \frac{\text{power of signal}}{\text{power of noise}} = \frac{P}{\sigma_w^2} = 0.3441$$

```
SNR = P/lambda_min
```

```
SNR = 0.3441
```

3) Does an additional value to the ACRS agree with the signal model?

Assume that an additional value of the autocorrelation function has been estimated and $r_x(3) = 20$.

3. Is this additional value in agreement with the signal model determined in the above questions?

The additional autocorrelation value is not in agreement with the signal model.

A fundamental property of the autocorrelation function is that $r_x(0) \geq r_x(\ell)$ for all ℓ.

This property is voilated with $r_x(0) = 17$ and $r_x(3) = 20$.

Exam 2018 Problem 1: Compute PSD from data using Pisarenko (sinusoid with additive white noise)

Here's a sequence of data from a measurement.

$$\{-11,\ 21,\ 18,\ 62,\ 34,\ -5,\ -14,\ -64,\ -49,\ -35,\ -1,\ 29,\ 37,\ 49,\ 32,\ 12\}$$

[✔] 1) Compute and plot the autocorrelation for lags 0 to 4

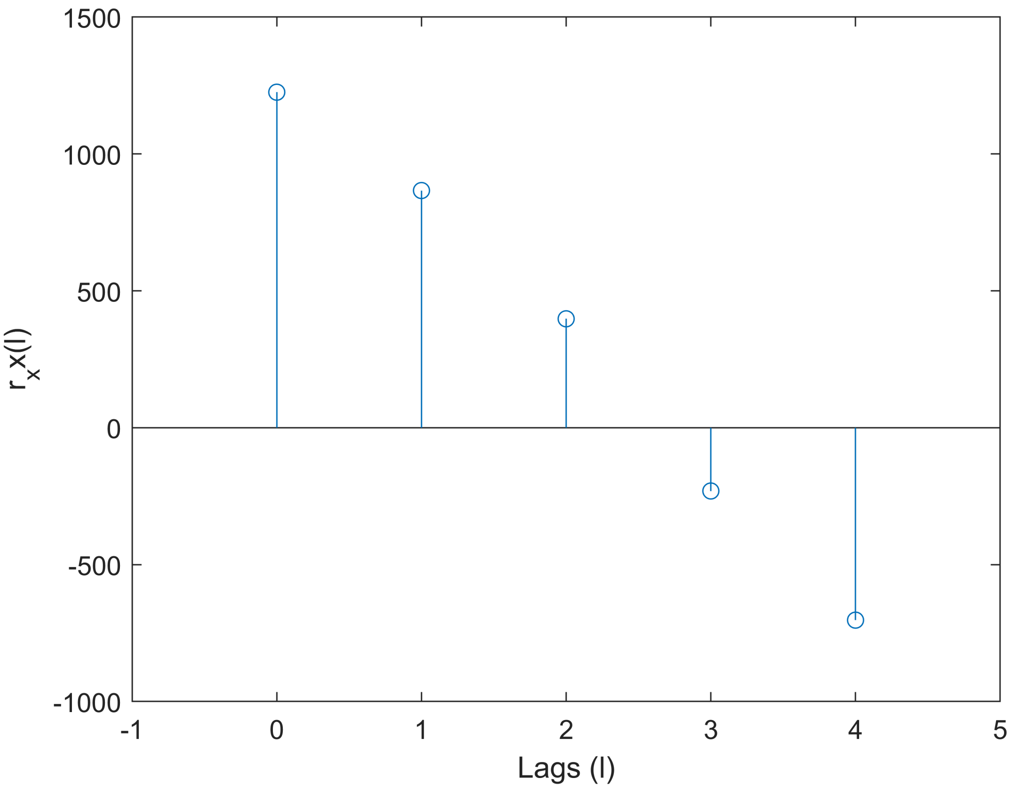
```
x = [-11, 21, 18, 62, 34, -5, -14, -64, -49, -35, -1, 29, 37, 49, 32, 12];

% 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
10^3 ×
    1.2256    0.8664    0.3979   -0.2310   -0.7027
```

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



[✔] 2) Compute the power spectral density assuming a Pisarenko model.

Assume that the signal is a sinusoid in additive white Gaussian noise.

The Pisarenko method can be used to recover the sinusoidal frequencies of a corrupted signal $x(n)$ given two assumptions:

- The signal $x(n)$ consists of p sinusoids that has been corrupted by white noise.
- The autocorrelation matrix of size $(p+1)\times(p+1)$ is known or can be estimated from data

The Pisarenko method consists of the following steps:

Step 1: Compute the autocorrelation matrix $\mathbf{R}_{\mathrm{xx}}$

Step 2: Find the eigenvector corresponding to the smallest minimum eigenvalue. The elements of this eigenvector is the parameters of the $\mathrm{ARMA}(2p,2p)$ model

Step 3: Find the frequencies $\{f_i\}$ of the sinusoids. This can be done by computing the roots of the polynomial $A(z)$ in Eq. (14.5.4) in the book. This polynomial has $2p$ poles on the unit circle which correspond to the frequencies of the system.

$$A(z) = 1 + \sum_{m=1}^{2p} a_m z^{-m} \qquad (14.5.4)$$

***Step 4*:** Solve Eq. (14.5.11) for the signal powers $\{P_i\}$

$$\begin{bmatrix} \cos 2\pi f_1 & \cos 2\pi f_2 & \cdots & \cos 2\pi f_p \\ \cos 4\pi f_1 & \cos 4\pi f_2 & \cdots & \cos 4\pi f_p \\ \vdots & \vdots & & \vdots \\ \cos 2\pi p f_1 & \cos 2\pi p f_2 & \cdots & \cos 2\pi p f_p \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_p \end{bmatrix} = \begin{bmatrix} \gamma_{yy}(1) \\ \gamma_{yy}(2) \\ \vdots \\ \gamma_{yy}(p) \end{bmatrix} \qquad (14.5.11)$$

where

- $\gamma_{\mathrm{yy}}(1), \gamma_{\mathrm{yy}}(2), \cdots, \gamma_{\mathrm{yy}}(p)$ are the estimated autocorrelation values
- $P_i = \frac{A_i^2}{2}$ is the average power of the *i*th sinusoid and A_i is the corresponding amplitude

Step 5: Estimate the amplitude $A_i = \sqrt{2P_i}$

These steps are coded up in the `pisarenko()` function (see at end of this document):

```
[F, A, P, lambda_min] = pisarenko(r_xx(mid:mid+4), 1)
```

```
F = 0.0938
A = 45.6585
P = 1.0423e+03
lambda_min = 183.2154
```

We can describe the signal as:

$$x(n) = 45.6585 \cos(2\pi \cdot 0.0938n) + w(n)$$

where $w(n)$ is white noise with variance $\sigma_w^2 = 183.2154$

To compute the power spectral density, we need to find the autocorrelation function of the sinusoid.

In ADSI Problem 4.4, we found that the autocorrelation of a real sinusoid given by $y(n) = A\cos(\omega n + \phi)$ where A and ω are real constants and ϕ is a random variable with $\phi \sim U(0, 2\pi)$ is:

$$r_{\mathrm{yy}}(\ell) = \frac{A^2}{2}\cos(\omega \ell)$$

The autocorrelation function of white noise with variance σ_w^2 is given by:

$$r_{\mathrm{ww}}(\ell) = \sigma_w^2 \delta(\ell)$$

Combining these results, the general autocorrelation function of our signal is:

$$r_{\mathrm{xx}}(\ell) = \frac{A^2}{2}\cos(\omega \ell) + \sigma_w^2 \delta(\ell)$$

The power spectral density is the Fourier transform the autocorrelation function which is given by:

$$S(\omega) = \frac{A^2}{2}\pi[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)]+\sigma_w^2$$

```
A, F, lambda_min
```

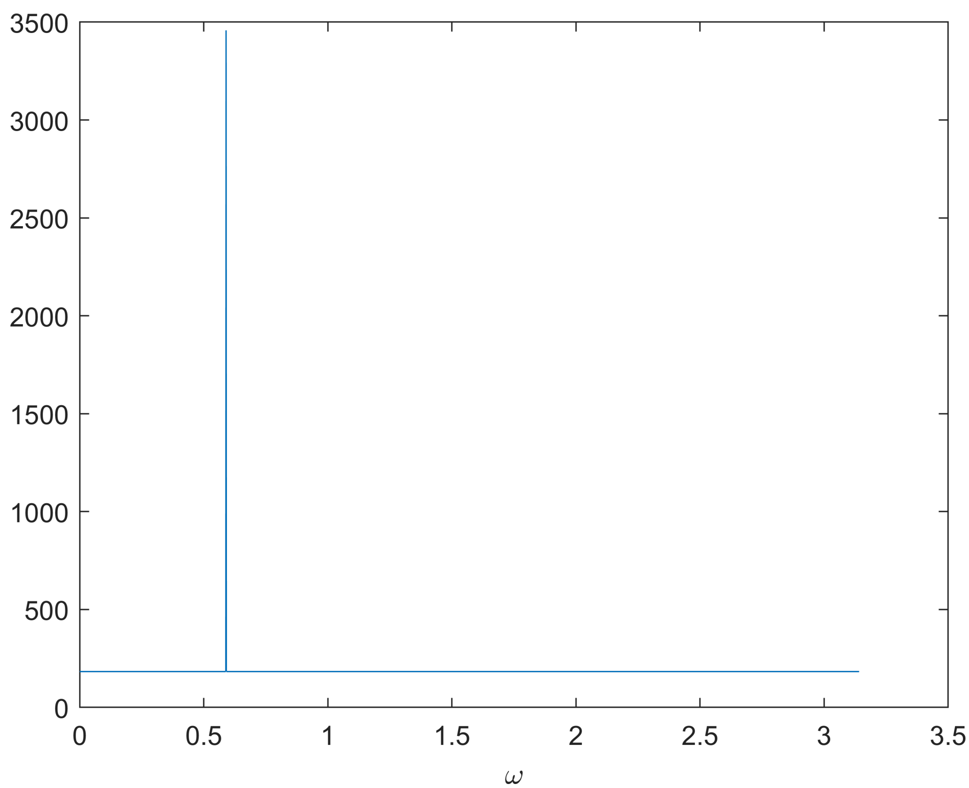
```
A = 45.6585
F = 0.0938
lambda_min = 183.2154
```

Using the values found via the Pisarenko, the PSD is:

$$S(\omega) = \frac{(45.6585)^2}{2}\pi[\delta(\omega-2\pi\cdot 0.0938)+\delta(\omega+2\pi\cdot 0.0938)]+183.2154$$

Plot the PSD:

```
D = 3;
w = 0:10^(-D):pi;
delta = @(n) round(n, D) == 0;
S = (A^2/2)*pi * (delta(w-(F*2*pi)) + delta(w+(F*2*pi))) + lambda_min;
%plot(w/pi, real(pow2db(S)))
plot(w, S);
xlabel('\omega')
```



[✔] 3. Discuss whether the Pisarenko model can be considered appropriate for the given data.

We can describe the signal as:

$$x(n) = 45.6585\cos(2\pi \cdot 0.0938n) + w(n)$$

where $w(n)$ is white noise with variance $\sigma_w^2 = 183.2154$

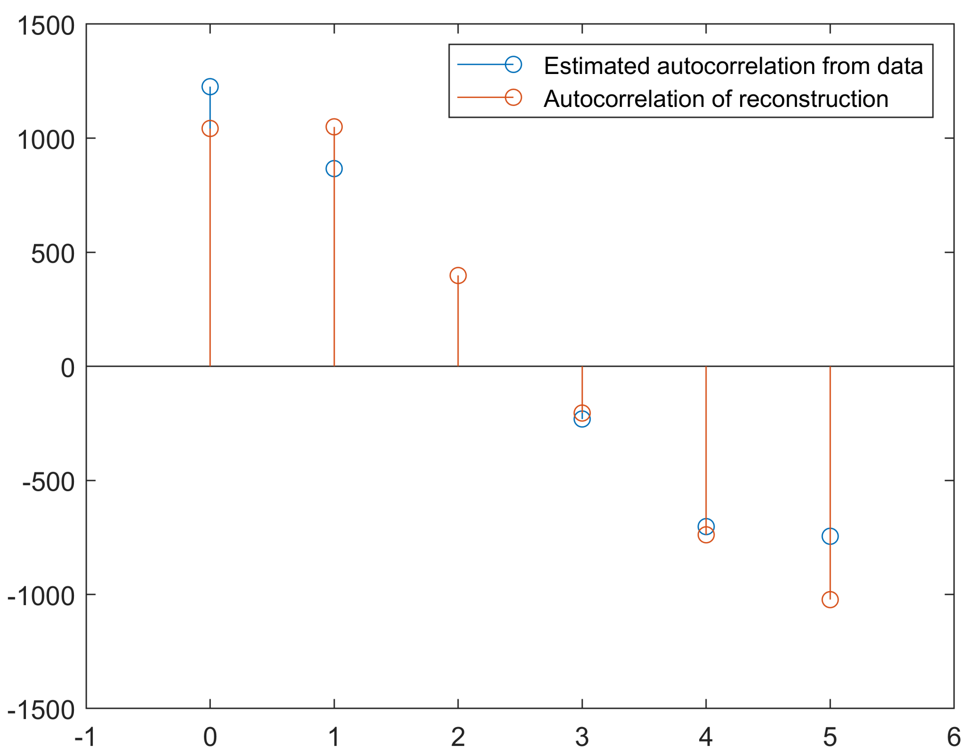
The general autocorrelation function of that signal is:

$$r_{xx}(\ell) = \frac{A^2}{2}\cos(\omega\ell) + \sigma_w^2\delta(\ell)$$

As only 16 samples are available, the estimate of the autocorrelation is quite uncertain and this uncertainty will enter into the Pisarenko model.

The plot below of the autocorrelation does resemble a slow oscillation and as such the Pisarenko model can't be ruled out on the basis of the available data.

```
% Plot the autocorrelation function of the sinusoid
ell = 0:5;
r_ww = lambda_min * (ell == 1);
r_xx_p = (A^2/2) * cos(2*pi*F*ell) + r_ww;
stem(ell, r_xx(mid:mid+max(ell)))
hold on;
stem(ell, r_xx_p)
legend('Estimated autocorrelation from data', 'Autocorrelation of reconstruction')
xlim([min(ell)-1, max(ell)+1])
hold off;
```



In addition, plotting the limited amount of samples, we see a sinusiodal shape. Since Pisarenko model assumes sinuisoidal signal that has been corrupted by white noise then it can be appropriate model.

Problem 5.1: Frequency estimation using Pisarenko's method

Find the frequency and amplitude of a single real sinusoidal signal in white noise, $y(n) = A\cos(2\pi fn + \phi) + w(n)$ when the first few values of the autocorrelation function are given by

$$r_y(l) = \begin{cases} 3, & l = 0 \\ 0, & l = \pm 1 \\ -2, & l = \pm 2 \end{cases}$$

```
clear variables;
```

Step 1: Compute the autocorrelation matrix $\mathbf{R}_{\mathrm{yy}}$

```
p = 1;
r_yy = [3, 0, -2];
R_yy = toeplitz(r_yy)
```

```
R_yy = 3×3
     3     0    -2
     0     3     0
    -2     0     3
```

Step 2: Find the minimum eigenvalue and the corresponding eigenvector. The elements of this eigenvector is the parameters of the $\mathrm{ARMA}(2p, 2p)$ model.

```
[eigvecs, eigvals] = eigs(R_yy, size(R_yy, 1), 'smallestreal');
min_eig_vec = eigvecs(:, 1)
```

```
min_eig_vec = 3×1
   -0.7071
         0
   -0.7071
```

Step 3: Find the frequencies $\{f_i\}$ of the sinusoids. This can be done by computing the roots of the polynomial $A(z)$ in (14.5.4). This polynomial has $2p$ poles on the unit circle which correspond to the frequencies of the system.

$$A(z) = 1 + \sum_{m=1}^{2p} a_m z^{-m} \qquad (14.5.4)$$

Since the given estimating the frequency of a signal real sinusoid, we have:

$$A(z) = 1 + a_1 z^{-1} + a_2 z^{-2}$$

The parameters $\{a_0, a_1, a_2\}$ corresponds to the eigenvector of the smallest eigenvalue:

```
% By definition a_0 = 1 because we want A(z)=1+....
a = min_eig_vec / min_eig_vec(1)
```

```
a = 3×1
     1
     0
     1
```

This means we have following IIR filter:

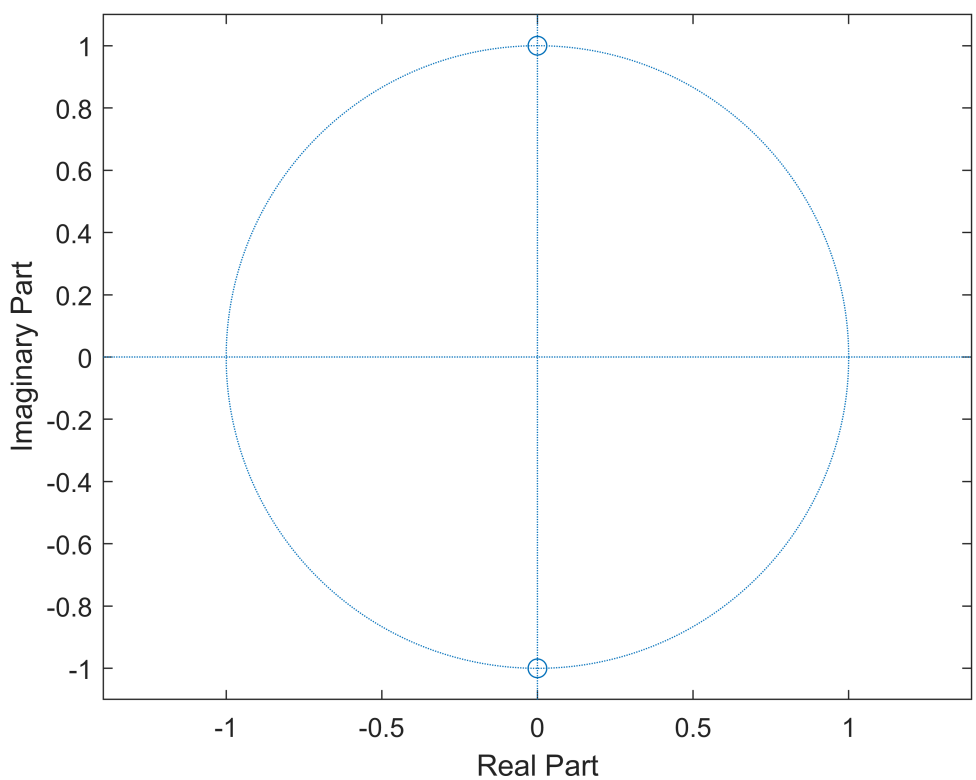
$$A(z) = 1 + z^{-2}$$

We know that the polynomial $A(z)$ in (14.5.4) has $2p = 2 \cdot 1 = 2$ poles on the unit circle. We can find the poles by finding the roots of the system $A(z) = 1 + z^{-2}$:

```
z = roots(a)
```

```
z = 2×1 complex
   0.0000 + 1.0000i
   0.0000 - 1.0000i
```

```
zplane(z)
```



The poles on the unit circle correspond to the frequencies of the system.

The two poles are complex conjucate of each other in order to create a real sinusoid. If we only had one pole then it would be complex.

Since $z = A\, e^{j2\pi f}$, we can find the frequency by the angle:

$\angle z = 2\pi f$ and $f = \frac{\angle z}{2\pi}$

```
% The poles come in pairs. Each pair is complex conjugate
% of one another. Only use one of them and find the absolution value.
f1 = abs(angle(z(2))) / (2*pi)
```

```
f1 = 0.2500
```

```
f2 = angle(z(1)) / (2*pi)
```

```
f2 = 0.2500
```

Step 4: Solve Eq. (14.5.11) to find the signal powers $\{P_i\}$

$$\begin{bmatrix} \cos 2\pi f_1 & \cos 2\pi f_2 & \cdots & \cos 2\pi f_p \\ \cos 4\pi f_1 & \cos 4\pi f_2 & \cdots & \cos 4\pi f_p \\ \vdots & \vdots & & \vdots \\ \cos 2\pi p f_1 & \cos 2\pi p f_2 & \cdots & \cos 2\pi p f_p \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_p \end{bmatrix} = \begin{bmatrix} \gamma_{yy}(1) \\ \gamma_{yy}(2) \\ \vdots \\ \gamma_{yy}(p) \end{bmatrix} \qquad (14.5.11)$$

Since we only have one frequency, the equation becomes:

$\cos(2\pi f_1)P_1 = \gamma_{yy}(1)$ where $\gamma_{yy}(1)$ is the second autocorrelation value.

We want to find P_1:

$$P_1 = \frac{\gamma_{yy}(1)}{\cos(2\pi f_1)}$$

```
P1 = r_yy(2) / cos(2*pi*f1)
```

```
P1 = 0
```

This does not work because the autocorrelation value $\gamma_{yy}(1) = 0$. Instead we need to go one lag higher.

$\cos(2\pi f_1)P_2 = \gamma_{yy}(2)$ where $\gamma_{yy}(2)$ is the third autocorrelation value.

$$P_2 = \frac{\gamma_{yy}(2)}{\cos(4\pi f_1)}$$

```
P2 = r_yy(3) / cos(4*pi*f1)
```

```
P2 = 2
```

From section 14.5.1, we are given:

$$P_i = \frac{A_i^2}{2}$$

This allows us to estimate the amplitude A_i given P_i:

$$A_i = \sqrt{2P_i}$$

```
A1 = sqrt(2*P2)
```

```
A1 = 2
```

So frequency is $f = 0.25$ and the amplitude is $A = 2$.

Problem 5.2: Pisarenko of two sinusoids (sine) in white noise

Let a signal be given by

$$x(n) = A_1 \sin(\omega_1 n + \phi_1) + A_2 \sin(\omega_2 n + \phi_2) + w(n)$$

Where the phases ϕ_1 and ϕ_2 are uncorrelated and uniformly distributed from 0 to 2π, $w(n)$ is zero mean gaussian white noise.

```
clear variables;
```

0) Preliminary, compute the autocorrelation function of a sine signal:

In ADSI Problem 4.4, we found that the autocorrelation of a complex sinusoid givey by $y(n) = A\,e^{j(\omega n+\phi)}$ where A and ω are real constants and ϕ is a random variable with $\phi \sim U(0, 2\pi)$ is:

$$r_{yy}(\ell) = A^2\, e^{j\omega\ell}$$

To use this result, we need to convert the given signal in this problem to complex exponential signal.

A complex exponential signal is always formed by the sum of two real signals:

$$Ae^{j\omega n} = A\cos(\omega n) + j\,A\sin(\omega n)$$

Therefore, we know that:

$$\sin(\omega) = \frac{1}{2j}e^{j\omega} - \frac{1}{2j}e^{-j\omega}$$

Using this relation, we can rewrite a real signal $A\sin(\omega n+\phi)$ as:

$$z(n) = A\sin(\omega n+\phi)$$

$$z(n) = \frac{A}{2j}e^{j(\omega n+\phi)} - \frac{A}{2j}e^{-j(\omega n+\phi)}$$

To compute the autocorrelation function, we square the magnitude, remove the phase and replace n with ℓ:

$$r_{\mathrm{zz}}(\ell) = \left(\frac{A}{2j}\right)^2 e^{j\omega\ell} - \left(\frac{A}{2j}\right)^2 e^{-j\omega\ell}$$

We know that $(2j)^2 = 2^2\cdot j^2 = -4$ because $j=\sqrt{-1}$ so $j^2=-1$

$$r_{\mathrm{zz}}(\ell) = \frac{A^2}{-4}e^{j\omega\ell} - \frac{A^2}{-4}e^{-j\omega\ell}$$

$$r_{\mathrm{zz}}(\ell) = -\frac{A^2}{4}e^{j\omega\ell} + \frac{A^2}{4}e^{-j\omega\ell}$$

We want to make the autocorrelation function in terms of $\cos(\cdot)$, we rewrite the expression as follows:

$$r_{\mathrm{zz}}(\ell) = \left(-\frac{A^2}{2}\right)\cdot\frac{1}{2}(e^{j\omega\ell}+e^{-j\omega\ell})$$

Since $\cos(\theta) = \frac{1}{2}(e^{j\theta}+e^{-j\theta})$, we can rewrite the expression as:

$$r_{\mathrm{zz}}(\ell) = -\frac{A^2}{2}\cos(\omega\ell)$$

Thus, the autocorrelation function of a real signal $z(n) = A\sin(\omega n+\phi)$ is

$$r_{\mathrm{zz}}(\ell) = -\frac{A^2}{2}\cos(\omega\ell)$$

1) Calculate the autocorrelation function

The autocorrelation function of a real signal $z(n) = A\sin(\omega n+\phi)$ where A and ω are real constants and ϕ is a random variable with $\phi \sim U(0,2\pi)$ is:

$$r_{\mathrm{zz}}(\ell) = -\frac{A^2}{2}\cos(\omega\ell)$$

The autocorrelation function of white noise Gassian noise with zero mean and variance σ_w^2 is:

$$r_{\mathrm{ww}}(\ell) = \sigma_w^2\delta(\ell)$$

In this problem, we are given:

Let a signal be given by

$$x(n) = A_1 \sin(\omega_1 n + \phi_1) + A_2 \sin(\omega_2 n + \phi_2) + w(n)$$

Where the phases ϕ_1 and ϕ_2 are uncorrelated and uniformly distributed from 0 to 2π, $w(n)$ is zero mean gaussian white noise.

The autocorrelation function of the signal $x(n)$ given in this problem is:

$$r_{xx}(\ell) = -\frac{{A_1}^2}{2}\cos(\omega_1 \ell) - \frac{{A_2}^2}{2}\cos(\omega_2 \ell) + \sigma_w^2 \delta(\ell)$$

or

$$r_{xx}(\ell) = -\frac{{A_1}^2}{2}\cos(2\pi f_1 \ell) - \frac{{A_2}^2}{2}\cos(2\pi f_2 \ell) + \sigma_w^2 \delta(\ell)$$

2) Choose appropriate values for the amplitudes and frequencies and for the noise power

We choose following values:

$$A_1 = 4, A_2 = 8, f_1 = \frac{1}{4\pi}, f_2 = \frac{1}{\pi}, \sigma_w^2 = 1$$

3) Calculate the eigenvalues of the autocorrelation matrix as a function of its size and compare with your expectation

```
% The values that we chose in problem 5.2
A1=4;  A2=8;  f1=1/(4*pi);  f2=1/pi;  wvar=1;

% The size of the autocorrelation sequence
% L = 64;
% l = 0:L-1;
% r_xx = -((A1^2/2) * cos(2*pi*f1*l)) - ((A2^2/2) * cos(2*pi*f2*l)) + wvar
ell_seq = 0:pi/4:2*pi;

L = numel(ell_seq);

% Initialise the autocorrelation sequence with zeros.
% This becomes very important especially when
% the size of the ACRZ (the ell variable) is changed
r_xx = zeros(L, 1);

% Compute the autocorrelation sequence based on the
% autocorrelation function derived in problem 1)
for i = 1:L
    ell = ell_seq(i);
    r_xx(i) = -((A1^2)/2 * cos(2*pi*f1*ell)) - ((A2^2)/2 * cos(2*pi*f2*ell)) + wvar;
end
r_xx
```

```
r_xx = 9×1
  -39.0000
   -6.3910
   27.3431
   -2.0615
  -31.0000
    4.0615
   38.6569
    8.3910
  -23.0000
```

```
% Compute the autocorrelation matrix
R_xx = toeplitz(r_xx)
```

```
R_xx = 9×9
  -39.0000   -6.3910   27.3431   -2.0615  -31.0000    4.0615   38.6569    8.3910 ···
   -6.3910  -39.0000   -6.3910   27.3431   -2.0615  -31.0000    4.0615   38.6569
   27.3431   -6.3910  -39.0000   -6.3910   27.3431   -2.0615  -31.0000    4.0615
   -2.0615   27.3431   -6.3910  -39.0000   -6.3910   27.3431   -2.0615  -31.0000
  -31.0000   -2.0615   27.3431   -6.3910  -39.0000   -6.3910   27.3431   -2.0615
    4.0615  -31.0000   -2.0615   27.3431   -6.3910  -39.0000   -6.3910   27.3431
   38.6569    4.0615  -31.0000   -2.0615   27.3431   -6.3910  -39.0000   -6.3910
    8.3910   38.6569    4.0615  -31.0000   -2.0615   27.3431   -6.3910  -39.0000
  -23.0000    8.3910   38.6569    4.0615  -31.0000   -2.0615   27.3431   -6.3910
```

4) Use the Pisarenko method to calculate the spectrum and compare with the expected results

```
[F, A] = pisarenko(r_xx, 2)
```

```
F = 2×1
    0.0878
    0.4166
A = 2×1
    7.0875
    8.0111
```

Problem 5.3: Frequency resolution of the Pisarenko method

In this problem the goal is to investigate the frequency resolution of the Pisarenko method. We will use your two-sinusoids-in-white-noise model from the previous problem:

Let a signal be given by

$$x(n) = A_1 \sin(\omega_1 n + \phi_1) + A_2 \sin(\omega_2 n + \phi_2) + w(n)$$

Where the phases ϕ_1 and ϕ_2 are uncorrelated and uniformly distributed from 0 to 2π, $w(n)$ is zero mean gaussian white noise.

```
clear variables;
```

1) The stability of PSD estimates using the Pisarenko method

a) Create a number of short realizations, i.e. N=64 or N=128 with different initial phases.

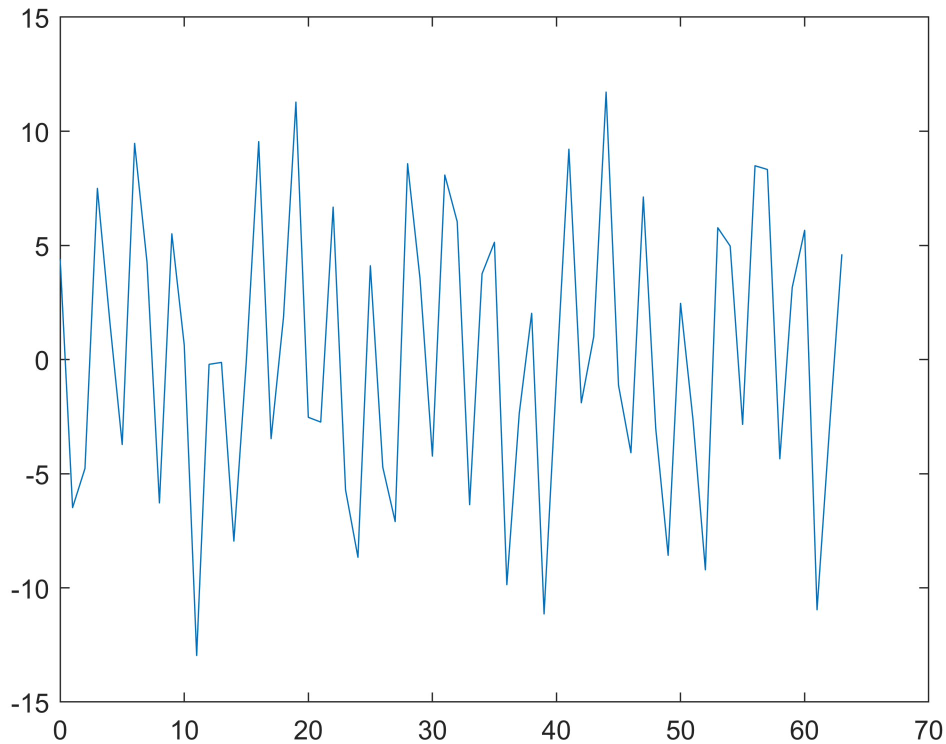
```
% The values that we chose in problem 5.2
A1=4;  A2=8;  f1=1/(4*pi);  f2=1/pi;  wvar=1;

N = 64;
n = 0:N-1;

% Generate phi(1) and phi(2) from U(0, 2*pi)
phi = 2*pi*rand(1, 2);

% Generate noise signal with chosen variance
wn = sqrt(wvar) * randn(1, N);

% Generate the signal
x1 = A1 * sin(2*pi*f1*n + phi(1)) + A2 * sin(2*pi*f2*n + phi(2)) + wn;
plot(n, x1)
```

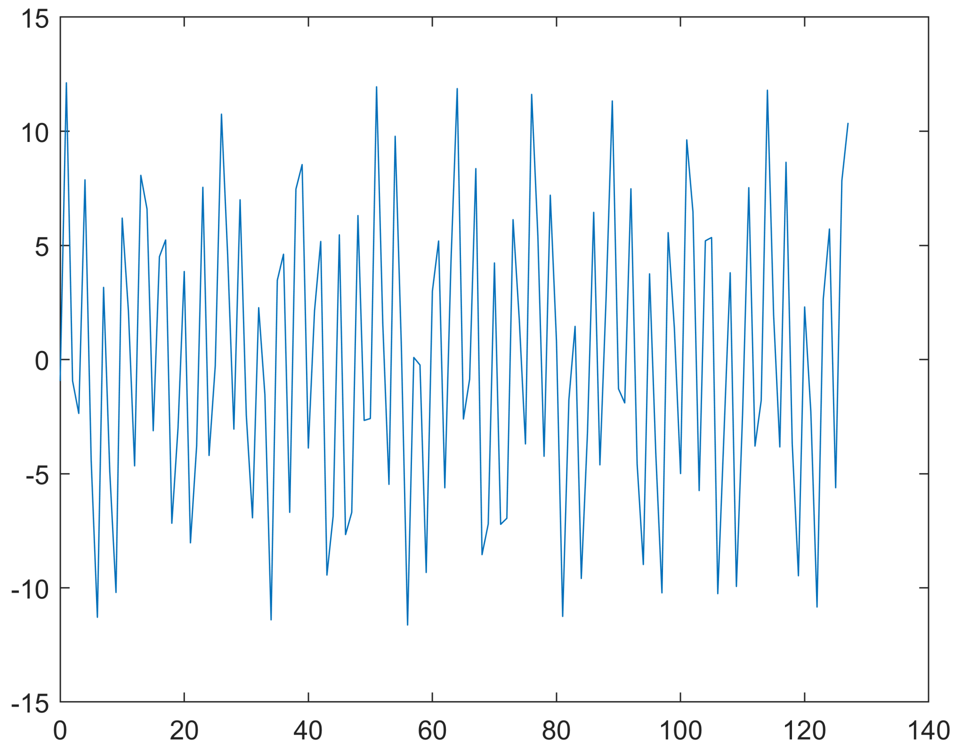



```
N = 128;
n = 0:N-1;

% Generate phi(1) and phi(2) from U(0, 2*pi)
phi = 2*pi*rand(1, 2);

% Generate noise signal with chosen variance
wn = sqrt(wvar) * randn(1, N);

% Generate the signal
x2 = A1 * sin(2*pi*f1*n + phi(1)) + A2 * sin(2*pi*f2*n + phi(2)) + wn;
plot(n, x2)
```



b) Use the Pisarenko method to compute the spectrum of the two realizations.

```
p = 2; % Number of sinusoids in the signals

% Compute the autocorrelation sequence
[r_x1, lags] = xcorr(x1', 'biased');

% Compute the autocorrelation matrix from rx(0)
mid_index = ceil(numel(lags)/2);
[F, A] = pisarenko(r_x1(mid_index:end), p)
```

```
F = 2×1
    0.3168
    0.0842
A = 2×1
    7.6815
    3.6995
```

```
% Compute the autocorrelation sequence
[r_x2, lags] = xcorr(x2', 'biased');

% Compute the autocorrelation matrix from rx(0)
mid_index = ceil(numel(lags)/2);
[F, A] = pisarenko(r_x2(mid_index:end), p)
```

```
F = 2×1
    0.3182
    0.0800
```

```
A = 2×1
    7.8739
    3.9904
```

```
f1, f2, A1, A2
```

```
f1 = 0.0796
f2 = 0.3183
A1 = 4
A2 = 8
```

c) Comment on the stability of the Pisarenko method

The Pisarenko method seem to be stable.

Problem 5.4: Pisarenko and coloured noise

In the derivation of the harmonic methods it was assumed that the signal consisted of a sinusoids or complex exponentials in white noise. In real life, the assumption of white noise is likely to be wrong in a lot of situations.

```
clear variables;
```

1) Create a MATLAB model of a sinusoidal signal in white, slightly coloured and very coloured noise.

When the power spectral density of the noise is not uniform across the entire frequency spectrum, it is called colored noise.

```
% The values that we chose in problem 5.2
A1=4;      f1=1/8;    phi1 = 2*pi*rand(); % Phase ~ U(0, 2pi)

N = 128;   n = 0:N-1;

% Generate a signal with white noise
w1 = randn(1, N);
x1 = A1 * cos(2*pi*f1*n + phi1) + w1;

% Generate a signal with slightly coloured noise
w2 = step(dsp.ColoredNoise('InverseFrequencyPower', 0.4, 'SamplesPerFrame', numel(n)))';
x2 = A1 * cos(2*pi*f1*n + phi1) + w2;

% Generate a signal with highly coloured noise
w3 = step(dsp.ColoredNoise('InverseFrequencyPower', 2, 'SamplesPerFrame', numel(n)))';
x3 = A1 * cos(2*pi*f1*n + phi1) + w3;
```

2) Compare the eigenvalues of the autocorrelation matrix for the three different scenarios.

```
L = 3; % The length of ACRS
[r_x1, ~] = xcorr(x1, L, 'biased');
R_x1 = toeplitz(r_x1(L+1:end));
eigs(R_x1, size(R_x1, 1), 'smallestreal')
```

```
ans = 4×1
    0.7408
    0.8408
   16.5341
   16.7258
```

```
[r_x2, ~] = xcorr(x2, L, 'biased');
R_x2 = toeplitz(r_x2(L+1:end));
eigs(R_x2, size(R_x2, 1), 'smallestreal')
```

```
ans = 4×1
    0.8768
    1.1592
   16.4413
   17.1596
```

```
[r_x3, ~] = xcorr(x3, L, 'biased');
R_x3 = toeplitz(r_x3(L+1:end));
eigs(R_x3, size(R_x3, 1), 'smallestreal')
```

```
ans = 4×1
    0.2948
    2.6214
   17.4658
  243.7402
```

3) Calculate the Pisarenko spectra and discuss whether Pisarenko is useful when the noise is coloured.

```
[F1, A1] = pisarenko(r_x1(L+1:end), 1)
```

```
F1 = 0.1247
A1 = 3.9887
```

```
[F2, A2] = pisarenko(r_x2(L+1:end), 1)
```

```
F2 = 0.1235
A2 = 3.9976
```

```
[F3, A3] = pisarenko(r_x3(L+1:end), 1)
```

```
F3 = 0.0401
A3 = 11.4277
```

It seems that Pisarenko yields relatively good results when the noise is slightly coloured. However, the results degrade for the noise highly coloured.

Problem 5.5: Pisarenko, wrong choice of eigenvector

In example 14.5.1 from the note the autocorrelation matrix for a process consisting of a single sinusoid in additive white noise is given as

$$\boldsymbol{R}_y = \begin{bmatrix} 3 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 3 \end{bmatrix}$$

The example proceeds to find $f_1 = \frac{1}{8}$

1) How does the use of a wrong eigenvalue influence the solution?

Repeat the example, but use the two other eigenvalues as starting points. How does the use of a wrong eigenvalue influence the solution?

Step 1: Compute the autocorrelation matrix $\mathbf{R}_{\mathrm{yy}}$

```
r_yy = [3, 1, 0];
R_yy = toeplitz(r_yy);
[eigvecs, eigvals] = eigs(R_yy, size(R_yy, 1), 'smallestreal');
```

```
% Choose which eigenvalues is used for the calculations.
% The eigenvalues are sorted in increasing order, so
% the first eigenvalue is the smallest.
chosen_eig_idx = 1;
```

Step 2: Find the minimum eigenvalue and the corresponding eigenvector. The elements of this eigenvector is the parameters of the $\mathrm{ARMA}(2p, 2p)$ model

```
vals = diag(eigvals);
eig_val_min = vals(chosen_eig_idx)
```

```
eig_val_min = 1.5858
```

Using the smallest eigenvalue, we should get:

$$\sigma_w^2 = \lambda_{\min} = 3 - \sqrt{2} \quad = 1.5858$$

```
% Get the eigenvector corresponding to the minimum eigenvalue
eig_vec_min = eigvecs(:,chosen_eig_idx);

% Ensure that a_0 = 1 (this is by definition)
eig_vec_min = eig_vec_min / eig_vec_min(1)
```

```
eig_vec_min = 3×1
    1.0000
   -1.4142
    1.0000
```

If the smallest eigenvalue is used we would get $a_0 = 1$, $a_1 = -1.4142$, $a_2 = 1$

Step 3: Find the frequencies $\{f_i\}$ of the sinusoids. This can be done by computing the roots of the polynomial $A(z)$ in (14.5.4). This polynomial has $2p$ poles on the unit circle which correspond to the frequencies of the system.

$$A(z) = 1 + \sum_{m=1}^{2p} a_m z^{-m} \qquad (14.5.4)$$

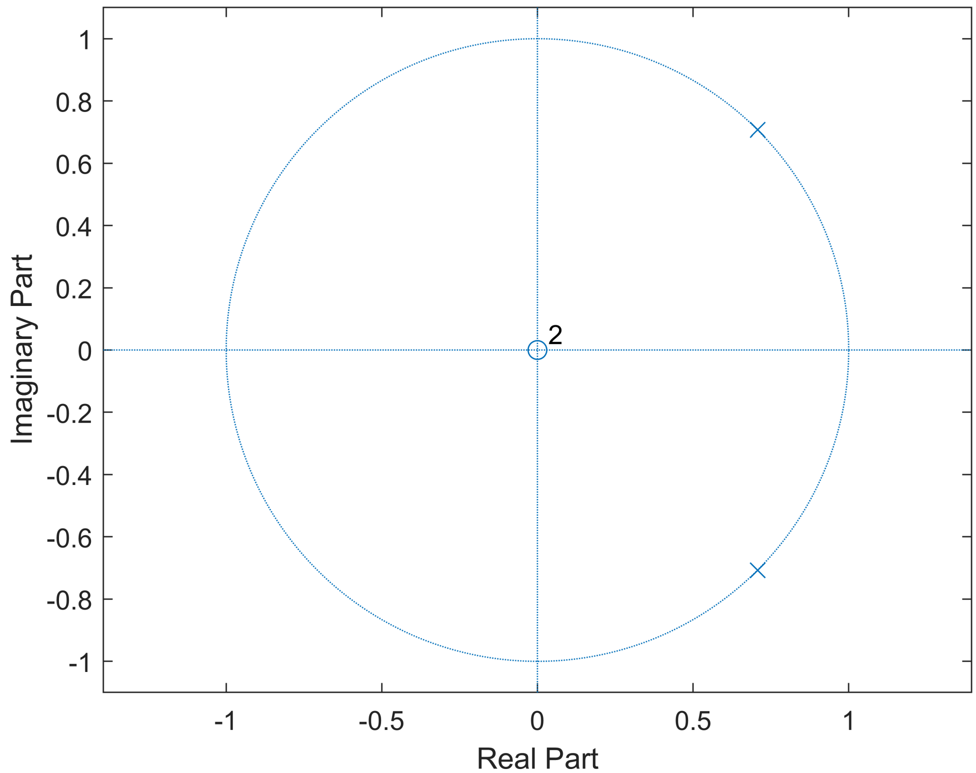
```
z = roots(eig_vec_min)
```

```
z = 2×1 complex
   0.7071 + 0.7071i
   0.7071 - 0.7071i
```

So we have $z_1 = 0.7071 + 0.7071j$ and $z_2 = 0.7071 - 0.7071j$

Note that $|z_1| = |z_2| = 1$ so poles are on the unit circle:

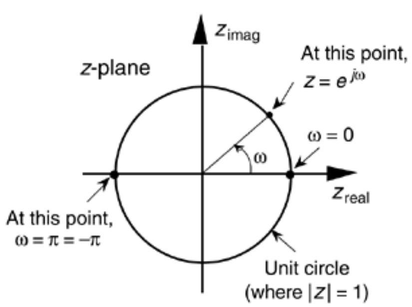
```
% norm(z(1)), norm(z(2))
% Verify that the poles are on the unit circle
zplane(1, eig_vec_min')
```



In general, the z is defined as follows:

$$z = Ae^{j2\pi f} = A[\cos(2\pi f) + j\sin(2\pi f)]$$

Recall that $\angle z = 2\pi f$ and $f = \frac{\angle z}{2\pi}$ because of the definition of the complex number z:



Since the two z quantities that we found are on the unit circle, their magnitude $A = 1$.

We also found that two pair of zs are complex conjucates of one another. So we only use one of them to compute the frequency:

```
f1 = abs(angle(z(2))) / (2*pi)
```

```
f1 = 0.1250
```

Step 4: Solve Eq. (14.5.11) for the signal powers $\{P_i\}$

$$\begin{bmatrix} \cos 2\pi f_1 & \cos 2\pi f_2 & \cdots & \cos 2\pi f_p \\ \cos 4\pi f_1 & \cos 4\pi f_2 & \cdots & \cos 4\pi f_p \\ \vdots & \vdots & & \vdots \\ \cos 2\pi p f_1 & \cos 2\pi p f_2 & \cdots & \cos 2\pi p f_p \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_p \end{bmatrix} = \begin{bmatrix} \gamma_{yy}(1) \\ \gamma_{yy}(2) \\ \vdots \\ \gamma_{yy}(p) \end{bmatrix} \tag{14.5.11}$$

Since we only have one frequency, the equation becomes:

$\cos(2\pi f_1)P_1 = \gamma_{yy}(1)$ where $\gamma_{yy}(1)$ is the second autocorrelation value:

We want to find P_1:

$$P_1 = \frac{\gamma_{yy}(1)}{\cos(2\pi f_1)}$$

```
P1 = r_yy(2) / cos(2*pi*f1)
```

```
P1 = 1.4142
```

From section 14.5.1, we are given:

$$P_i = \frac{A_i^2}{2}$$

This allows us to estimate the amplitude A_i given P_i:

$$A_i = \sqrt{2P_i}$$

```
A1 = sqrt(2*P1)
```

```
A1 = 1.6818
```

Functions

```
function [A]=test()
    A=1
end
```

```matlab
function [F, A, P, lambda_min]=pisarenko(r_xx, p)
    % Estimates the frequencies and amplitudes using the Pisarenko method
    %  r_xx: autocorrelation sequence starting from zero.
    %        The length of ACRS must be at least 2p+1.
    %  p: assumed number of sinusoids in the signal
    %  F: normalised frequencies of the sinuoids
    %  A: amplitudes of the sinuoids
    if numel(r_xx) < 2*p+1
        error(strcat('The length of ACRS must be at least ', int2str(2*p+1)));
    end

    % Compute the autocorrelation matrix
    R_xx = toeplitz(r_xx(1:2*p+1));

    % Perform the eigendecomposition
    [eigvecs, eigvals] = eigs(R_xx, size(R_xx, 1), 'smallestreal');

    eigvals = diag(eigvals);
    lambda_min = eigvals(1);

    % Find the eigenvector 'a' corresponding to the smallest eigenvalue.
    % The function eigs() sorts eigenvectors, so just pick the first column.
    a = eigvecs(:, 1);

    % Ensure that a_0 = 1 (this is by definition)
    a = a / a(1);

    % The elements of this eigenvector corresponds to the parameters
    % of an ARMA(2p, 2p) model: a_0, a_1, ..., a_2p where p: number of sinusoids
    % The polynomial A(z) in (14.5.4) has 2p poles on the unit circle.
    % Obtain the poles by finding the roots of the system.
    z = roots(a);

    % Estimate frequencies
    F = zeros(p, 1);
    for i = 1:p
        % The poles come in pairs. Each pair is complex conjugate
        % of one another. Only use one of them and find the absolute value.
        z_i = z(2*i);
        F(i) = abs(angle(z_i)) / (2*pi);
    end

    % Build the matrix of cosines
    C = zeros(p);
    for i = 1:p
        for j = 1:p
            C(i, j) = cos(i * 2*pi * F(j));
        end
    end

    % Start the ACRS from the second element according to equation (14.5.11)
    gamma = r_xx(2:p+1);

    % Solve equation (14.5.11) for P
    P = C\gamma; % Same as `inv(C)*gamma` but faster and more accurate

    % Since P=A^2/2, we can compute the amplitude A=sqrt(2*P)
    A = sqrt(2 * P);
```

```
end
```