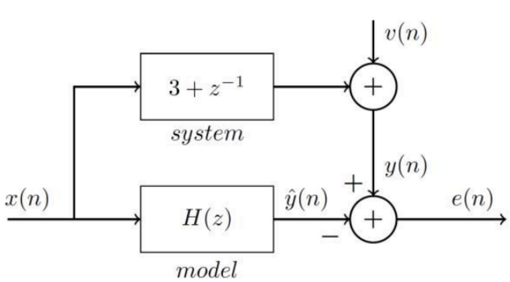
Exam June 2018

Table of Contents

Problem 1: Wiener filter........1
[✔] 1) Calcuate the autocorrelation matrix and cross-correction vector........2
[✔] 2) Calculate the expected energy........2
[✔] 3) Solve for H(z) and comment on the influence of v(n)........3
Problem 2:........4
[✔] 1) Calculate and plot the autocorrelation function of the data........4
[✔] 2) Can data be described by an MA(2) model?........5
[✔] 3) Calculate the power spectral density of the random process assuming an AR(2) model........6
Problem 3: True/False Questions........8
[✔] 1) When is a system invertible?........8
[✔] 2) The Hilbert transform of a triangular wave signal is another triangular wave signal........9
[✔] 3) Leakage can be reduced in estimates of Power Spectral Density by applying a window function to the data........11
Problem 4: LMS algorithm........11
1) Compute the coefficients for the first three iterations of the algorithm........11
2) Discuss the pros and cons of increasing the step size........11
Problem 5: Probability Density Functions........12
[✔] 1) Show that the probability density function is valid........12
[✔] 2) Calculate the marginal probabiltiy density function........12
[✔] 3) Without calculations, argue whether var(X) or var(Y) is larger........13
[✔] 4) Show that X and Y are uncorrelated........13
Functions........16

Problem 1: Wiener filter

Consider the following setup, where we aim to model the system $3+z^{-1}$ using a Wiener filter. The input signal $x(n)$ is unit variance white noise. The output of the system is corrupted by additive noise $v(n)$, which is also white with $\sigma_v^2 = 0.2$. It can be assumed that $x(n)$ and $v(n)$ are uncorrelated.



$x(n) \sim \mathrm{WN}(0,1)$, $v(n) \sim \mathrm{WN}(0, 0.2)$ and $x(n)$ and $v(n)$ are uncorrelated

```
clear variables;
```

[✔] 1) Calcuate the autocorrelation matrix and cross-correction vector

1. Calculate the 2×2 autocorrelation matrix $\mathbf{R}_x$ and the 2×1 cross-correlation vector $\mathbf{g}$.

Since $x(n) \sim \mathrm{WN}(0,1)$, its autocorrelation function is:

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

The autocorrelation matrix $\mathbf{R}_x$ is therefore

```
M = 2;
ell = 0:M-1;
r_xx = [1, 0];
R_xx = toeplitz(r_xx)
```

From the drawing, we know that the output of the system is:

$$y(n) = 3x(n) + x(n-1) + v(n)$$

Now, we can compute the cross-correlation vector g:

$$\begin{aligned} r_{xy}(\ell) &= E[x(n)y(n-\ell)] \\ &= E[x(n)(3x(n-\ell) + x(n-\ell-1) + v(n-\ell))] \\ &= E[3x(n)x(n-\ell) + x(n)x(n-\ell-1) + x(n)v(n-\ell)] \\ &= 3E[x(n)x(n-\ell)] + E[x(n)x(n-\ell-1)] + E[x(n)v(n-\ell)] \\ &= 3r_{xx}(\ell) + r_{xx}(\ell-1) + E[x(n)v(n-\ell)] \end{aligned}$$

If we assume that $x(n)$ and $v(n)$ are uncorrelated then:

$$= 3r_{xx}(\ell) + r_{xx}(\ell-1)$$

Since $r_{xx}(\ell) = \delta(\ell)$ then:

$$r_{xy}(\ell) = 3\delta(\ell) + \delta(\ell-1)$$

Thus, the cross-correlation vector g is:

$$g = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

[✔] 2) Calculate the expected energy

2. Calculate $E[y^2(n)]$.

From the drawing, we know that the output of the system is:

$$y(n) = 3x(n) + x(n-1) + v(n)$$

Let us first expand $y^2(n)$:

```
syms n x(n) v(n)
expand((3*x(n) + x(n-1) + v(n))^2)
```

Now, compute the expectation of each term:

$$E[x^2(n-1)] = E[x(n)x(n)] = r_{xx}(0) = 1 \cdot \delta(0) = 1 \cdot 1 = 1$$

$E[2x(n-1)v(n)] = 2E[x(n-1)v(n)] = 0$ (because $x(n)$ and $v(n)$ are uncorrelated)

$$E[6x(n-1)x(n)] = 6r_{xx}(1) = 6\delta(1) = 6 \cdot 0 = 0$$

$$E[v^2(n)] = r_{vv}(0) = 0.2 \cdot \delta(0) = 0.2 \cdot 1 = 0.2$$

$E[6v(n)x(n)] = 6r_{xv}(0) = 6 \cdot 0 = 0$ (because $x(n)$ and $v(n)$ are uncorrelated)

$$E[9x(n)x(n)] = 9r_{xx}(0) = 9\delta(0) = 9 \cdot 1 = 9$$

The expected energy is:

$$E[y^2(n)] = 1 + 0.2 + 9 = 10.2$$

[✓] 3) Solve for H(z) and comment on the influence of v(n)

3. Solve for $H(z)$ and comment on the result, in particular what is the influence of $v(n)$.

The second order Wiener filter for estimating the signal $y(n)$ is given by:

$$\hat{y}(n) = h_1 x(n) + h_2 x(n-1)$$

The optimum Wiener filter to estimate a random process is given by Eq. 14.109:

$$\boldsymbol{h}_o = \boldsymbol{R}_x^{-1}\boldsymbol{g}, \qquad (14.109)$$

where $\boldsymbol{R}_x$ is the correlation matrix of a random vector $\boldsymbol{x}$ and $\boldsymbol{g}$ is the cross-correlation vector between $\boldsymbol{x}$ and y

```
g = [3, 1]';
R_xx = [[1, 0];
        [0, 1]];
h_opt = R_xx\g
```

The optimal Wiener filter is:

$$H(z) = 3 + z^{-1}$$

This is exactly the system that we are trying to model.

Problem 2: Signal Modelling

```
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 $0 \leq l \leq 4$.

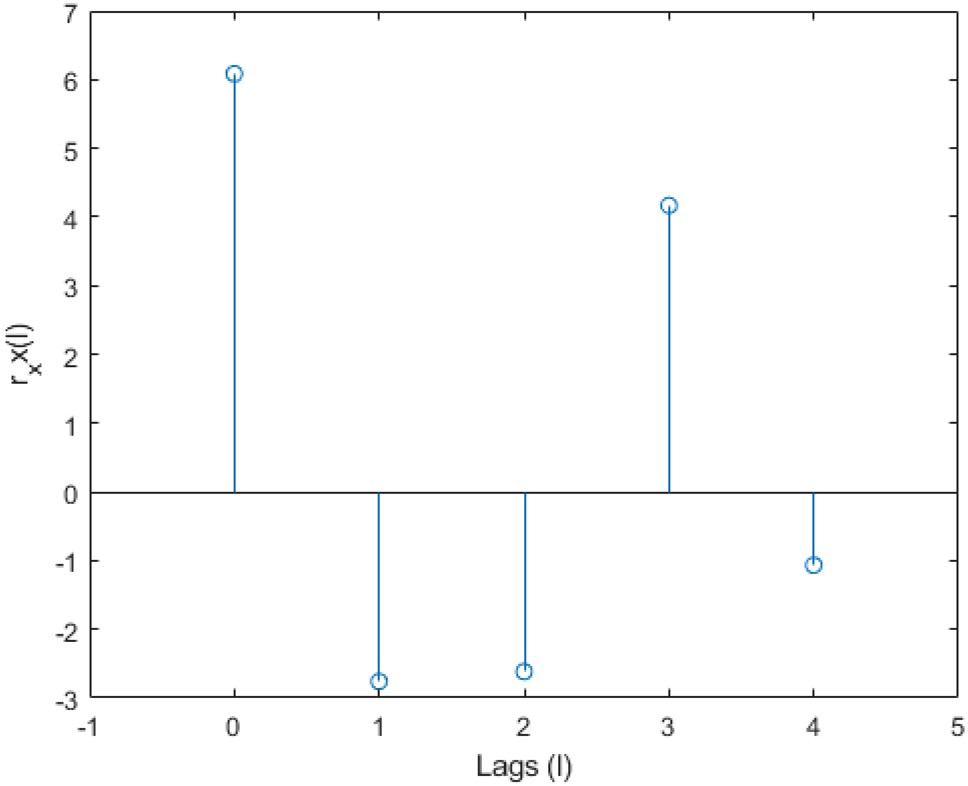
```
x = [-2.39, 2.28, 0.65, -4.68, 2.88, 0.13, -2.48, 1.78, 2.27, -4.49, 1.13, 0.9, -0.92];

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

The autocorrelation for an MA(2) process is:

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

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:

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

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:

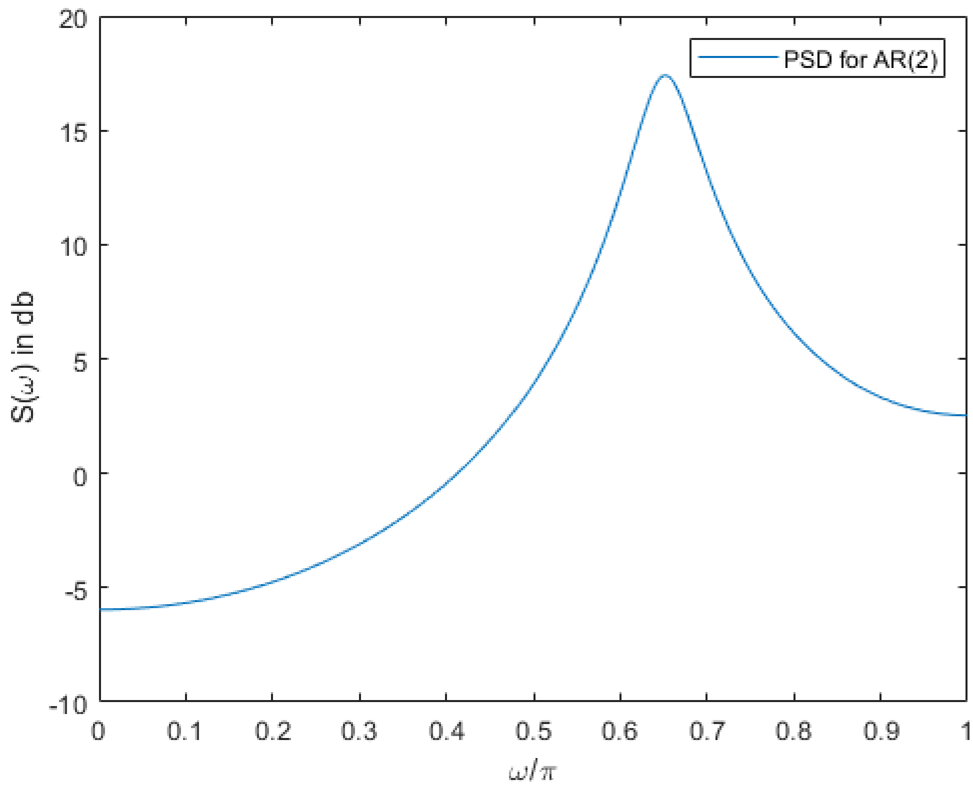
1. Find the coefficents $\{a_1, a_2, \cdots, a_p\}$ for the $\mathrm{AR}(p)$ model
2. Compute the transfer function for the $\mathrm{AR}(p)$ by computing the sum and finding its reciprocal
3. Compute the conjugate of the transfer function: $|H(e^{j\omega})|^2$
4. Multiply it with the variance σ_x^2

The method above is implemented in the 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 3: True/False Questions

```
clear variables;
```

For the statements given below, state whether they are true or false and justify your answer for each statement.

[✔] 1) When is a system invertible?

1. A system with $H(z) = \frac{1+3z^{-1}}{1+\frac{1}{4}z^{-1}}$ is invertible.

Answer: FALSE!

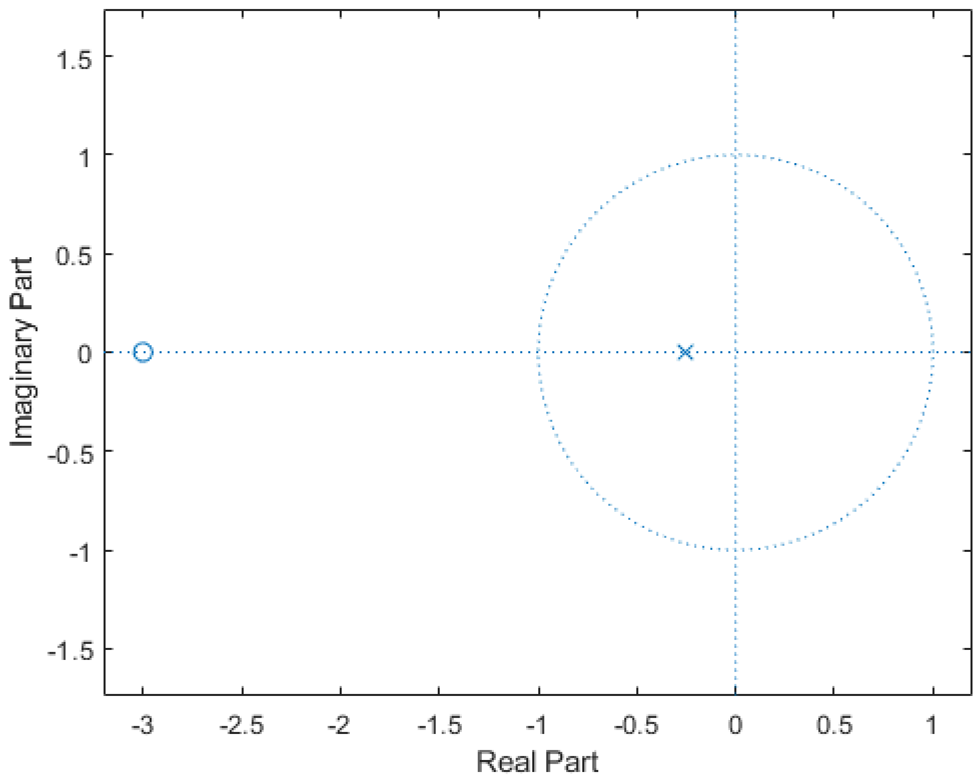
A filter $H(z) = \frac{B(z)}{A(z)}$ is said to be stable and causal if all the poles of $H(z)$ are inside the unit circle.

If the inverse filter $H_{\text{inv}}(z) = \dfrac{A(z)}{B(z)}$ has to be stable and causal, then all the poles of $H_{\text{inv}}(z)$ must be inside the unit circle or equivalently all zeros of $H(z)$ must be inside the unit circle.

In practice, we say that a system is invertible if its zeros and poles are inside the unit circle (minimum phase).

Plotting the poles and zeros of the given system function, we observe that the zero is outside the unit circle.

```
b = [1, 3];
a = [1, 1/4];
zplane(b, a)
```



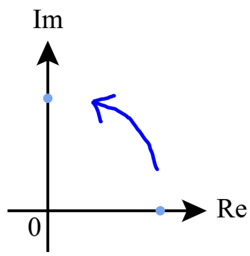
[✔] 2) The Hilbert transform of a triangular wave signal is another triangular wave signal

2. The Hilbert transform of a triangular wave signal is another triangular wave signal.

Answer: FALSE.

The Hilbert transform rotates the Fourier coefficients in the complex space which converts the real components into the imaginary component. The quarter-cycle shifted signal is then added to the original signal. In other words, a Hilbert transform takes each frequency component present in the original signal and shifts its phase by $-\frac{\pi}{2}$.

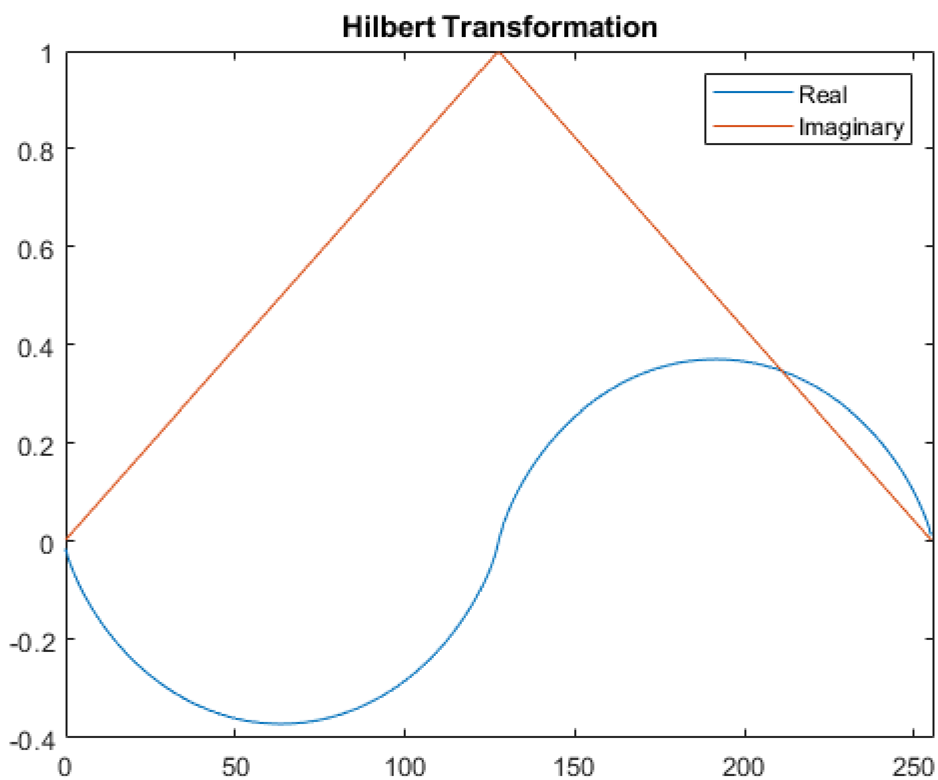
The Hilbert transform of $\cos(\theta)$ is $\sin(\theta)$ i.e., a quarter-cycle shift.



A triangular wave signal can thought of being composed of multiple frequency component. If each of these are shifted by 90 degrees then result is not a triangular wave signal.

We can experiment with it.

```
N = 256;
n = 0:N-1;
xr = triang(256);
x = hilbert(xr);
plot(n, imag(x), n, real(x))
legend('Real','Imaginary')
title('Hilbert Transformation')
xlim([0, N])
```



[✔] 3) Leakage can be reduced in estimates of Power Spectral Density by applying a window function to the data.

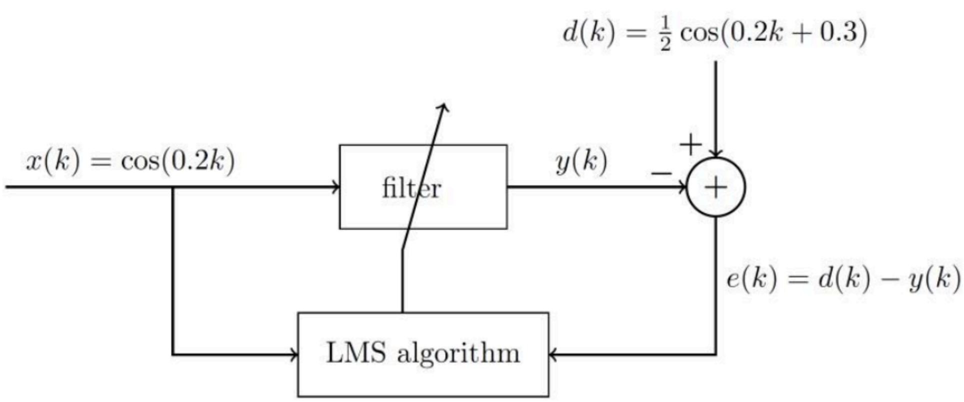
3. Leakage can be reduced in estimates of power spectral density by applying a window function to the data.

Answer: TRUE.

Leakage is an artefact of an FFT applied to a non-periodic data. Even if a signal is periodic like a sinousoid, when the signal is measured with some interval it can be non-periodic. The issue is that FFT assumes that the signal is periodic and repeats itself after the measured interval. This is not the case when the data is non-periodic, which contains sharp transitions at the end of each measured interval. These sharp changes have a broad frequency response which lead to spectral leakage. Windows can help *minimize* the effects of leakage by smoothing the time domain signal, but cannot eliminate leakage.

Problem 4: LMS algorithm

Consider the following system where an adaptive filter is used to filter an incoming signal.



1) Compute the coefficients for the first three iterations of the algorithm

In this problem the LMS algorithm is used to adjust a two-tap FIR filter.

1. Compute the filter coefficients for the first three iterations of the adaptive algorithm. Use $\mathbf{w}(0) = \mathbf{0}$, $\mu = 0.1$ and $x(k) = 0$ for $k < 0$.

2) Discuss the pros and cons of increasing the step size

2. The LMS algorithm contains a step size parameter, μ. Discuss the pros and cons of increasing the step size.

Problem 5: Probability Density Functions

Consider the following joint probability density function.

$$f_{X,Y}(x,y) = \begin{cases} \frac{1}{2} & 1 \le y \le 2 \text{ and } 1 \le x \le 3 \\ 0 & \text{elsewhere} \end{cases}$$

[✔] 1) Show that the probability density function is valid

1. Show that $\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{X,Y}(x,y) dx\, dy = 1$.

Integrate using MATLAB:

```
syms x y f(x,y) fX(x) fY(y)
f(x,y) = 1/2;
inner = int(f(x,y), x, 1, 3);
outer = int(inner, y, 1, 2)
```

```
outer = 1
```

```
% Alternative method is to numerically evaluate double integral
fun = @(x,y) 1/2 + 0*(x+y); % Hack: 0*(x+y) is added for MATLAB
integral2(fun, 1, 3, 1, 2)
```

```
ans = 1
```

[✔] 2) Calculate the marginal probabiltiy density function

2. Calculate the marginal probability density function $f_X(x)$.

The marginal probability density function is given as:

$$f_X(x) = \int_{-\infty}^{\infty} f_{X,Y}(x,y)\mathrm{d}y$$

The marginal probability for X is given as:

$$f_X(x) = \int_1^2 \frac{1}{2}\,\mathrm{d}y = \left[\frac{1}{2}y\right]_1^2 = 1 - \frac{1}{2} = \frac{1}{2}$$

Check it in MATLAB:

```
fX(x) = int(f(x,y), y, 1, 2)
```

fX(x) =

$$\frac{1}{2}$$

```
% Alternative method
fun = @(y) 1/2 + 0*y; % Hack!
integral(fun, 1, 2)
```

ans = 0.5000

The marginal probability for Y is given as:

$$f_Y(y) = \int_1^3 \frac{1}{2}\,\mathrm{dx} = \left[\frac{1}{2}x\right]_1^3 = \frac{3}{2} - \frac{1}{2} = 1$$

```
fY(y) = int(f(x,y), x, 1, 3)
```

fY(y) = 1

```
% Alternative method
fun = @(x) 1/2 + 0*x; % Hack!
integral(fun, 1, 3)
```

ans = 1.0000

[✔] 3) Without calculations, argue whether var(X) or var(Y) is larger

The variance is a measure of the spread of the distribution about its mean value. A large variance indicates that the values of the random variable are spread over a wider interval about the mean. In this problem, X has a wider span of possible values. Therefore, $\mathrm{var}(X) > \mathrm{var}(Y)$.

[✔] 4) Show that X and Y are uncorrelated

Two random variables are uncorrelated if

$$\mathrm{cov}(X, Y) = E(\mathrm{XY}) - E[X]E[Y] = 0 \quad \text{or} \quad E[\mathrm{XY}] = E[X]E[Y]$$

The simplest way to show that X and Y are uncorrelated is if the two random variables are independent.

Two random variables X and Y are said to be independent if:

$$f_{X,Y}(x, y) = f_X(x) f_Y(y)$$

The joint probability is given as:

$$f_{X,Y}(x, y) = \frac{1}{2}$$

The marginal probability for X is given as:

$$f_X(x) = \int_1^2 \frac{1}{2}\,\mathrm{dy} = \left[\frac{1}{2}y\right]_1^2 = 1 - \frac{1}{2} = \frac{1}{2}$$

The marginal probability for Y is given as:

$$f_Y(y) = \int_1^3 \frac{1}{2}\,\mathrm{dx} = \left[\frac{1}{2}x\right]_1^3 = \frac{3}{2} - \frac{1}{2} = 1$$

Since they are the random variables X and Y are indepedent they are also uncorrelated.

Another approach involves computing the covariance and showing that

$$\mathrm{cov}(X, Y) = E(\mathrm{XY}) - E[X]E[Y] = 0$$

We compute $E[X]$

$$E[X] = \int_{-\infty}^{\infty} x f_X(x)\,\mathrm{dx} = \int_1^3 \frac{1}{2}x\ \mathrm{dx} = \left[\frac{1}{4}x^2\right]_1^3 = \frac{1}{4}(3)^2 - \frac{1}{4}(1)^2 = 2$$

Check in MATLAB:

```
EX = int(f(x,y)*x, x, 1, 3)
```

```
EX = 2
```

```
% Alternative method
fun = @(x) 1/2*x;
integral(fun, 1, 3)
```

```
ans = 2
```

We compute $E[Y]$

$$E[Y] = \int_{-\infty}^{\infty} y f_Y(y)\,\mathrm{dy} = \int_1^2 y\ \mathrm{dy} = \left[\frac{1}{2}y^2\right]_1^2 = \frac{1}{2}(2)^2 - \frac{1}{2} = \frac{3}{2}$$

```
EY = int(fY(y)*y, y, 1, 2)
```

```
EY =
```

$$\frac{3}{2}$$

```
% Alternative method
fun = @(y) y;
integral(fun, 1, 2)
```

```
ans = 1.5000
```

We compute $E[\mathrm{XY}]$:

$$E[\mathrm{XY}] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} xy f_{X,Y}(x,y)\,\mathrm{dy}\,\mathrm{dx}$$

$$E[\mathrm{XY}] = \int_1^3 \left(\int_1^2 \frac{1}{2}xy\ \mathrm{dy}\right)\mathrm{dx}$$

First we compute the inner integral:

$$\int_1^2 \frac{1}{2}xy\ \mathrm{dy} = \left[\frac{1}{4}x\,y^2\right]_1^2 = \frac{1}{4}x(2)^2 - \frac{1}{4}x = \frac{3}{4}x$$

```
inner = int(x*y*f(x,y), y, 1, 2)
```

```
inner =
```

$$\frac{3\,x}{4}$$

Next, we compute the outer integral:

$$\int_1^3 \frac{3}{4}x\ \mathrm{dx} = \left[\frac{3}{8}x^2\right]_1^3 = \frac{3}{8}(3)^2 - \frac{3}{8} = \frac{27}{8} - \frac{3}{8} = \frac{24}{8} = 3$$

```
EXY = int(inner, x, 1, 3)
```

EXY = 3

Two random variables are uncorrelated if the covariance $\mathrm{cov}(X,Y) = E[\mathrm{XY}] - E[X]E[Y] = 0$.

```
EXY - EX * EY
```

ans = 0

Since $\mathrm{cov}(X,Y) = E[\mathrm{XY}] - E[X]E[Y] = 0$ then X and Y are uncorrelated.

Functions

```
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 [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
```