DSP LAB 4 REPORT

Dekun Xie 210337152

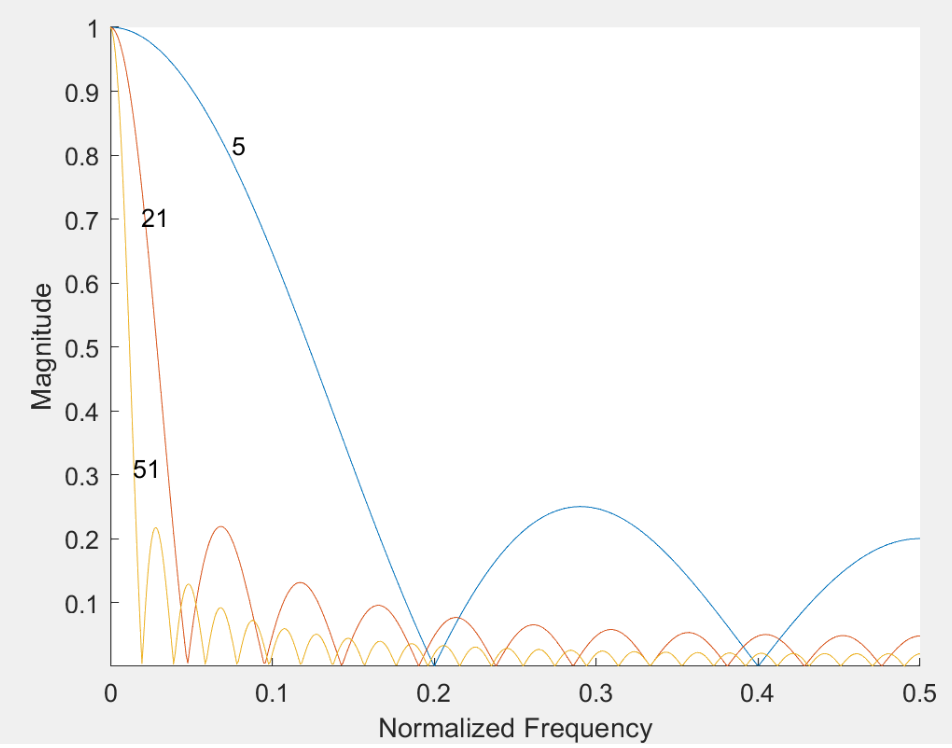
0.Introduction

In this lab, I will know about the plot and properties of FIR and IIR. In addition, the special case of M-pointer averager will be shown as well. And I will know the properties of allpass filter and the consequence of allpass filters in parallel artfully.

1.1 Knowing that an M-point averager has an impulse response of `ones(1,M)./M`, you can find its frequency response. I know you can. For M=[5, 21, 51] plot (using `plot` and not `stem`) on the same figure (`hold on`) the magnitude response versus the *normalized frequency* domain [0:0.5]. Evaluate the FFT using 1024 points, i.e., `fft(h,1024)`. ("Normalized frequency" means that a value of 1 is the sampling rate; and a value of 0.5 is the Nyquist frequency.) Use `gtext` to label each curve. Include you figure and code. It should look something like, but of course not nearly as amazing.

```
M = [5, 21, 51];
N = 1024;
for m = M
    ir = ones(1,m) ./m;
    dft = fft(ir,N);
    mag = abs(dft(1:N/2+1));
    hold on
    plot([0:N/2]/N,mag);
```

```
    gtext(num2str(m))
end
axis tight;
xlabel('Normalized Frequency')
ylabel('Magnitude')
```



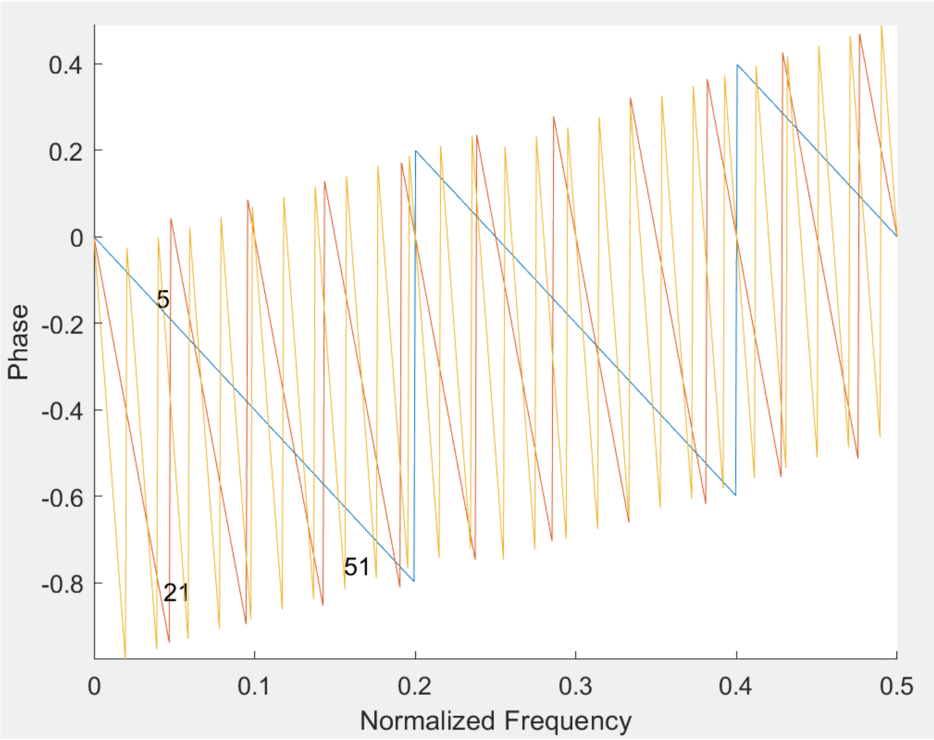
1.2 What happens to the passband of the moving average filter—those frequencies that are passed with only slight attenuation—as the order M increases? What single frequency passes completely unaltered in all three filters?

①As M increases, the passband becomes narrower.

②0 Hz in normalized Frequency.

1.3 For the same values of M, plot on one figure the phase responses in units of pi radians. This means using angle instead of abs, and <u>dividing the output of angle by pi</u>. Again, plot these

using normalized frequencies between [0:0.5]. Include your figure, but not your code.

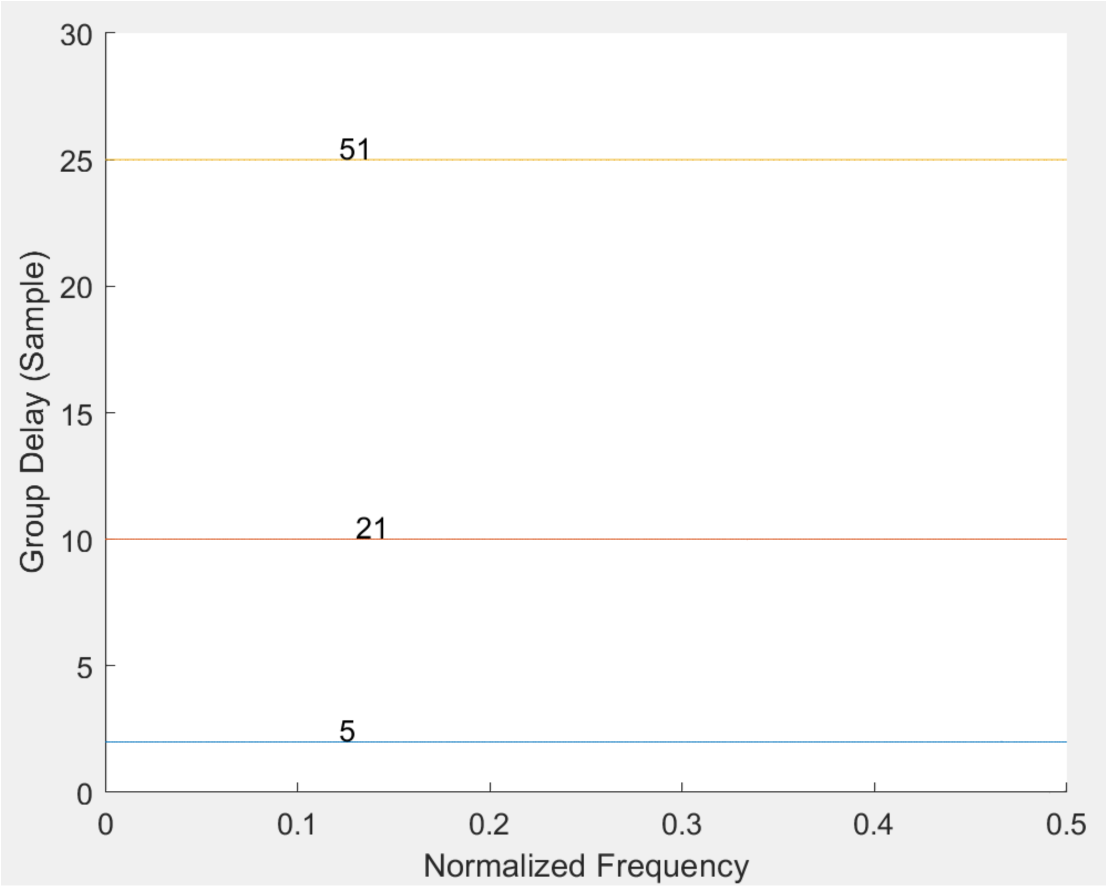


1.4 Can you explain why the phase response has a jagged appearance? Hint: look at your answer to 1.1, specifically the frequencies where the magnitude response becomes zero; and look to see what the jump in pi radians is in your plot. (Hint: if your phase response does not look jagged, seek help.)

As the phase of FIR is linear, every line is straight. Every time the magnitude of frequency becomes 0, the phase reset and jump. The jump step is 1/M.

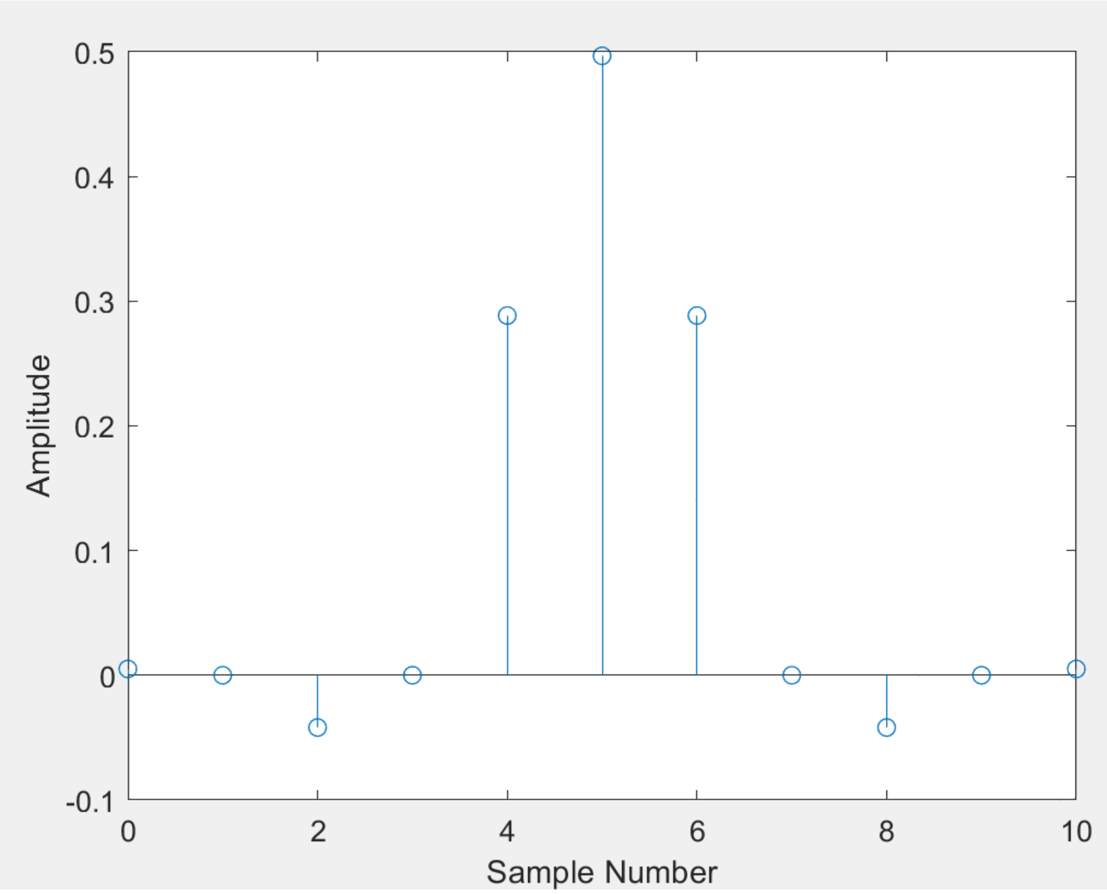
1.5 Plot the group delay of the moving average filter for each M above, by using the grpdelay routine. In this case you will say: g = grpdelay(ones(1,M)./M, 1, 1024, ‘whole’). Plot the group delay as a function of normalized frequency, and include your plot. It should look like Figure 2.

```
M = [5, 21, 51];
N = 1024;
for m = M
    ir = ones(1,m) ./m;
    g = grpdelay(ir, 1, 1024, 'whole');
    hold on
    plot([0:N/2]/N, g(1:N/2+1));
    gtext(num2str(m))
end
axis tight;
xlabel('Normalized Frequency')
ylabel('Group Delay (Sample)')
set (gca, 'YLim', [0 30])
```



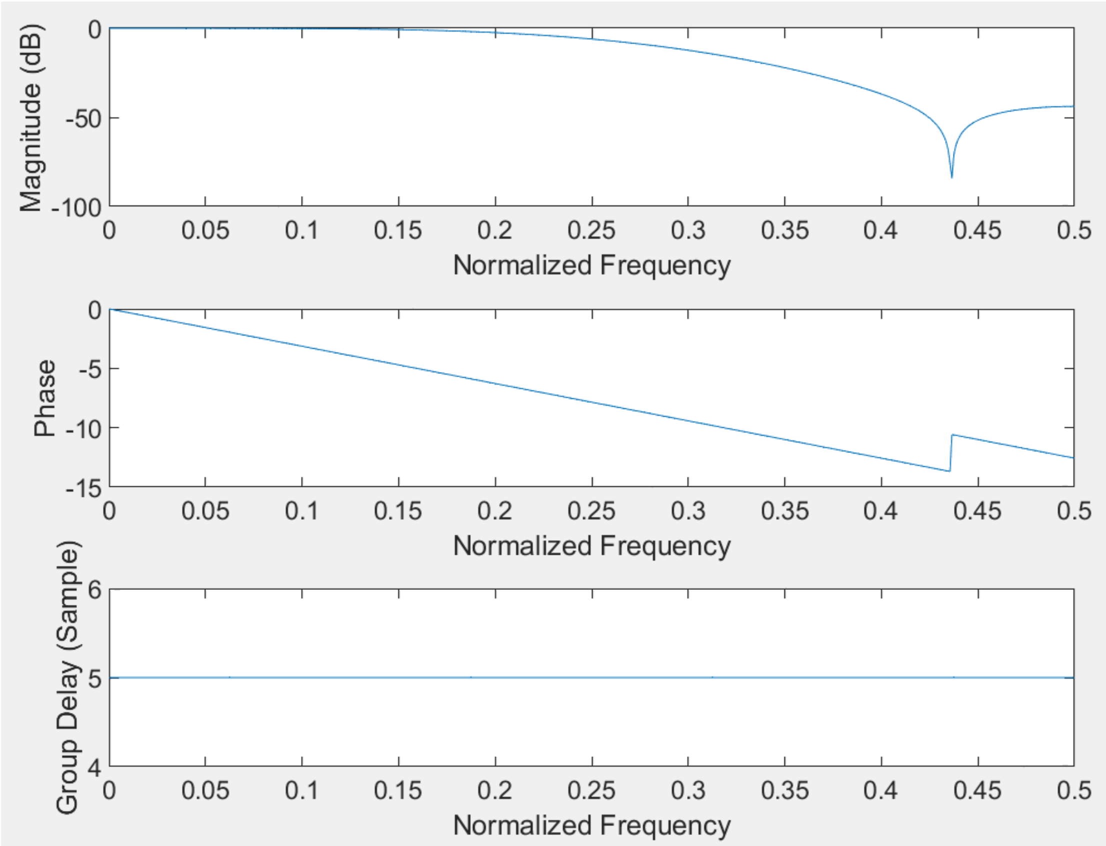
2.1 Consider the following impulse response of a 10-order, length-11 FIR system, as described by (2): $h[n]$={0.00506, 0, -0.04194, 0, 0.28848, 0.49679, 0.28848, 0, -0.04194, 0, 0.00506} for . Plot this impulse response with stem. Include your figure with appropriately labeled axes.

```
H = [0.00506, 0, -0.04194, 0, 0.28848, 0.49679, 0.28848, 0, -
0.04194, 0, 0.00506];
N = 11;
x = [0:N-1];
stem(x, H)
xlabel('Sample Number')
ylabel('Amplitude')
```



2.2 Find and plot the frequency response (magnitude and phase), and the group delay, of this system as functions of normalized frequency (between 0 and 0.5). Evaluate the FFT using 1,024 points, i.e., fft(h,1024). (This essentially zeropads the signal with enough zeros until its length is 1,024 samples, and then evaluates the FFT. Zeropadding is not always a good thing though, so get that out of your head!) Plot the magnitude response in dB. (This means something like 20*log10(abs(H)./max(abs(H))).) Make sure you unwrap the phase, i.e., unwrap(angle(H)). Include your figures and your code.

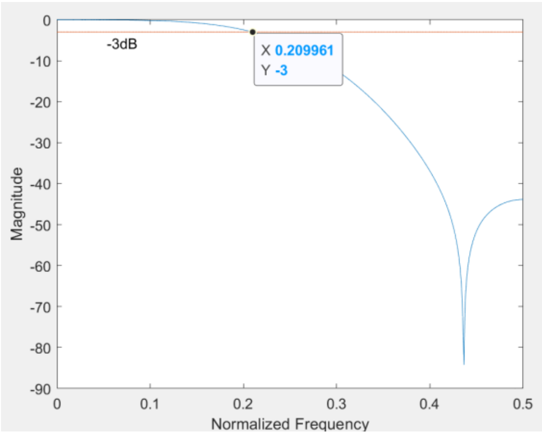
```
H = [0.00506, 0, -0.04194, 0, 0.28848, 0.49679, 0.28848, 0, -
0.04194, 0, 0.00506];
N = 1024;

dft = fft(H,N);
mag = 20 * log10( abs(dft)./max(abs(dft)));

subplot(3,1,1)
plot((0:N/2)/N, mag(1:N/2+1))
xlabel('Normalized Frequency')
ylabel('Magnitude (dB)')

subplot(3,1,2)
phase = unwrap(angle(dft));
plot((0:N/2)/N, phase(1:N/2+1))
xlabel('Normalized Frequency')
ylabel('Phase')

subplot(3,1,3)
g = grpdelay(H, 1, 1024, 'whole');
plot((0:N/2)/N, g(1:N/2+1))
xlabel('Normalized Frequency')
ylabel('Group Delay (Sample)')
```



2.3 What kind of filter is this? Find the "3 dB" point, or the normalized frequency above which the attenuation is greater than 3 dB. This is also called the cut-off frequency. (Hint: it should be a number between 0 and 0.5.) Physically, an attenuation of 3 dB corresponds to a reduction of the power by two, approximately.

①Low pass filter

②the cut-off frequency is 0.21



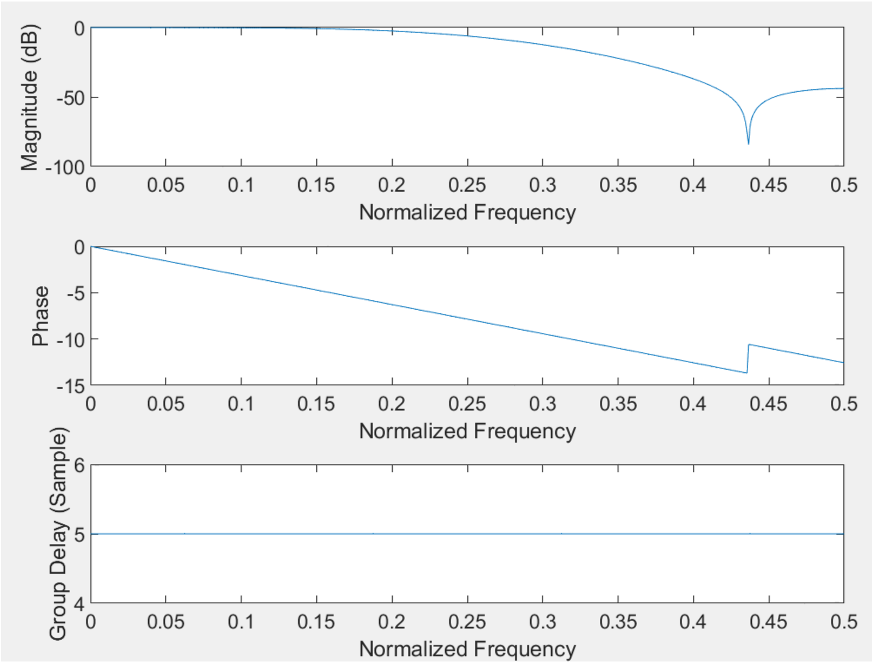
2.4 What is the group delay of this filter in samples? Would this filter destroy information contained in the envelope of a signal?

The group delay is 5.

This filter will not destroy information as the group delays are equal in every frequency. Therefore, in spite of delay, there is no change in the relative of each point.

2.5 For the given in 2.1, multiply it by a series of alternating positive and negative ones, or {1, -1, 1, -1, ⋯}. Now $h[n]$={0.00506, 0, -0.04194, 0, 0.28848, -0.49679, 0.28848, 0, -0.04194, 0, 0.00506}. Find and plot the frequency response (magnitude and phase), and the group delay, of this system as functions of normalized frequency (between 0 and 0.5). Evaluate the FFT using 1,024 points, i.e., fft(h,1024). Plot the magnitude response in dB. Make sure you unwrap the phase, i.e., unwrap(angle(H)). Include your figures, but not you code since it will basically be

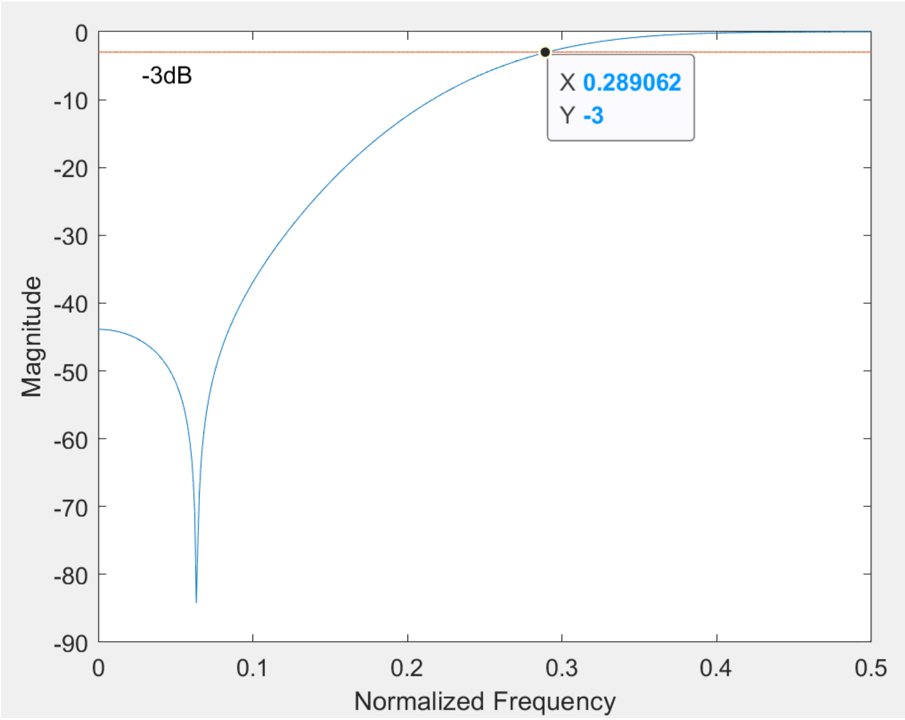
the same as 2.2.



2.6 What kind of filter is this? What is the cut-off frequency now?

①High Pass Filter

②the cut-off frequency is 0.29



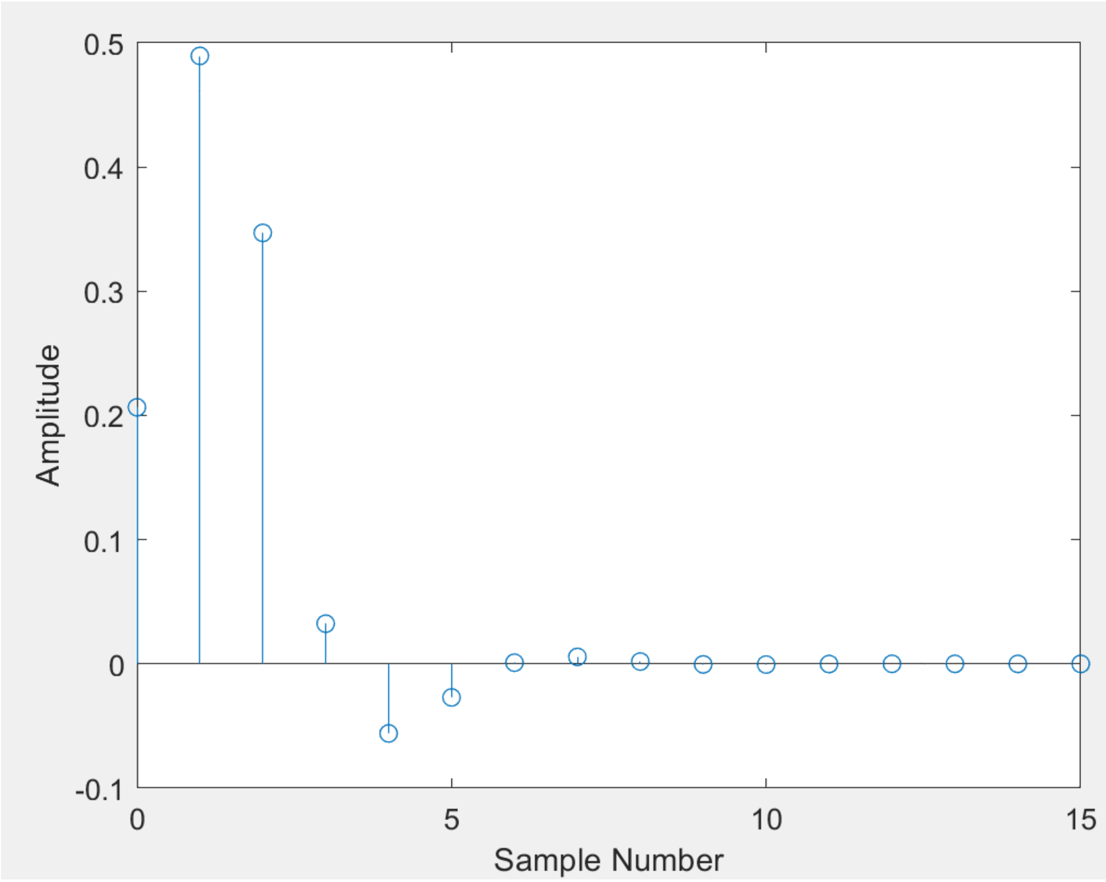
You only changed the sign of one value in the impulse response of a system and this is what happens! The former lowpass filter starts attenuating the lows and passing the highs. Its magnitude and phase responses appear to be mirror images of those of the original lowpass filter, found in 2.2. Its group delay remains the same however.

3.1 Using filter, find and plot the first sixteen samples of impulse response of this system. Include your plot and code.

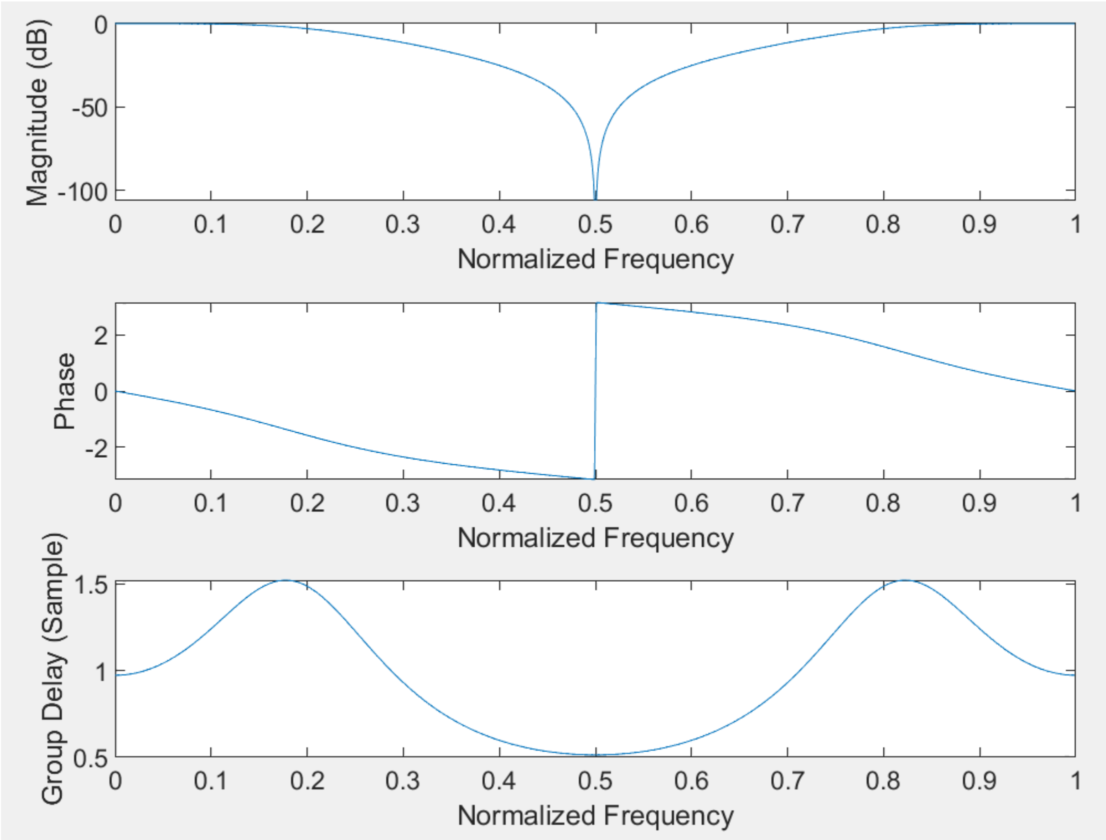
$$y[n] = 0.206572\,x[n] + 0.413144\,x[n-1] + 0.206572\,x[n-2] + 0.369527\,y[n-1] - 0.195816\,y[n-2]$$

```
N =16;
b = [0.206572, 0.413144, 0.206572];
a = [1 , -0.369527, 0.195816];
```

```
x(1:N) = 0;
x(1) = 1;
ftr = filter(b,a,x);
stem((0:N-1)s,ftr)
xlabel(Sample Number)
ylabel('Amplitude')
```



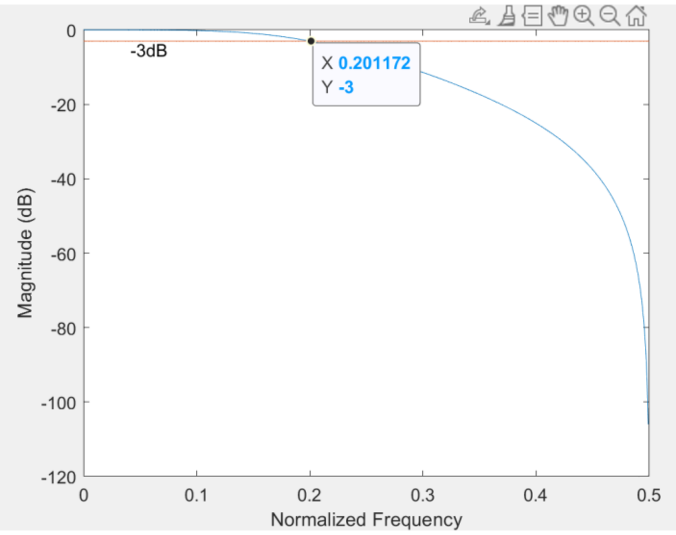
3.2 Using the code you made in 2.2, plot the frequency response and group delay for this system. This time however, use *freqz* to find the frequency response. You may still use *grpdelay*. Evaluate the frequency response at 1,024 points around the whole unit-circle. Include your figure, but not code.



3.3 Compare these results with those found for the FIR system described in 2.2. What are the differences? For instance, how do the cut-off frequencies compare? What is different about the group delay? How does the computational complexity of this filter compare to that of the FIR?

Difference:

①the cut-off is 0.201 Hz which is similar to but less than that of the FIR (0.209) in 2.2

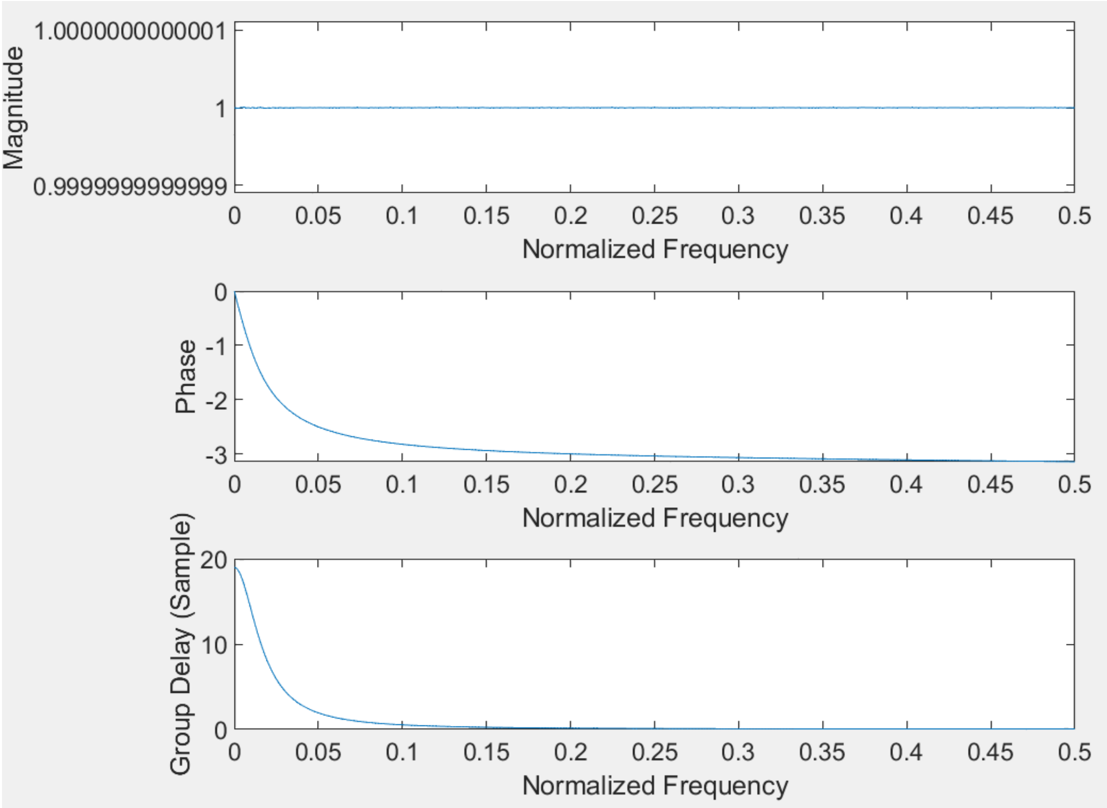


②the group delay changes with the frequency when that in 2.2 doesn’t change.

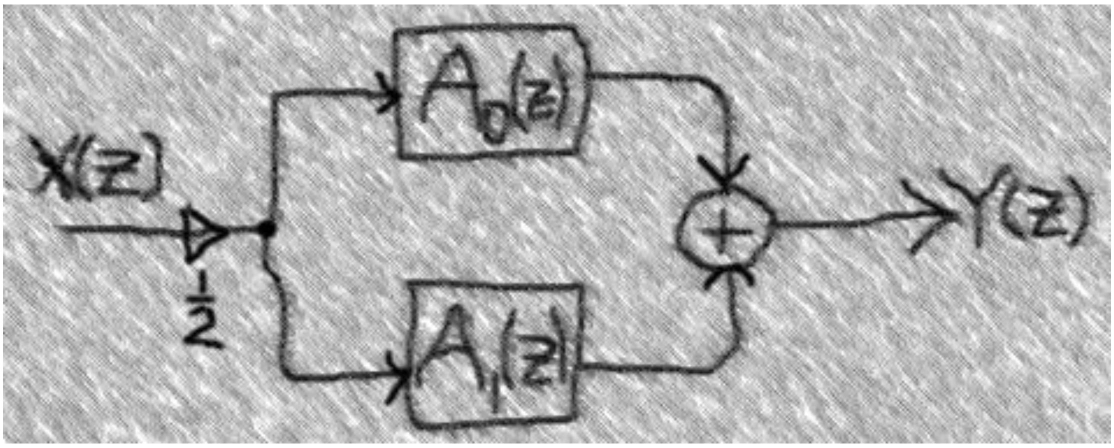
③FIR gets more computational complexity. As FIR requires more multiplies or adds.

④IIR’s phase response is not linear when FIR’s is linear.

3.4 Using MATLAB, confirm that the transfer function given by equation (5) $A_0(z)=\dfrac{a+z^{-1}}{1+az^{-1}}$ is an allpass filter. Use α= -0.9, and freqz to calculate the frequency response. Like in 2.2 and 3.2, plot the magnitude, phase response, and group delay of this filter over the normalized frequency range 0 ≤ f ≤ 0.5. This time do not plot the magnitude response in dB. Include these plots in your lab report, but not the code.



3.5 Now for our last trick, we will make something from nothing. Consider the two allpass filters in parallel artfully shown in Figure3.



With A0(z) described by (5) with α= -0.15, and A1(z) = 1, plot the frequency response and group delay of this system, just like you did in 2.2 and 3.2. Make sure your magnitude plot is in dB this time. Include your code and plot.

$$1 = \frac{\alpha + z^{-1}}{1 + \alpha z^{-1}}$$

$$1 + \alpha z^{-1} = \alpha + z^{-1} \quad (\alpha \neq z)$$

$$(z^{-1} - 1)\alpha = z^{-1} - 1$$

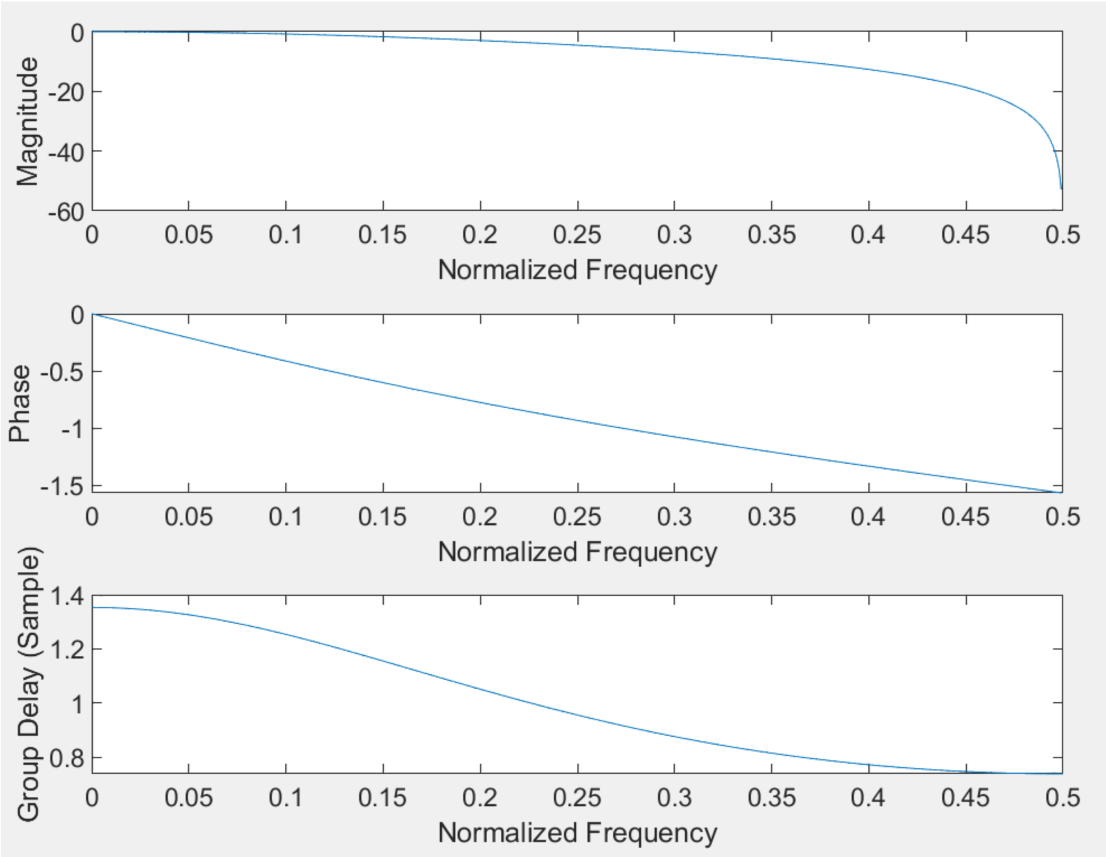
$$\alpha = \frac{z^{-1} - 1}{z^{-1} - 1}$$

$$\alpha = 1$$

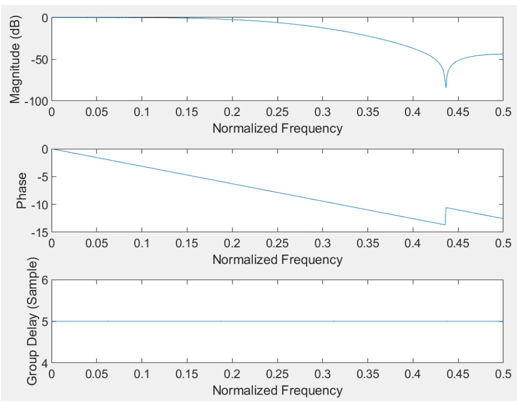
When A1(z) = 1, α = 1.

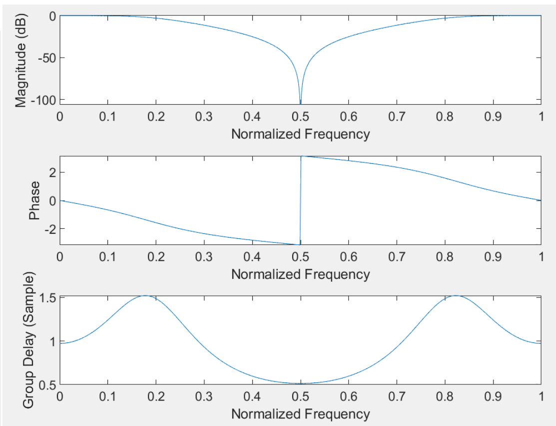
```
alpha = -0.15;
b = [alpha, 1];
a = [1, alpha];
N = 1024;

h0 = freqz(b,a,N,'whole');
h1 = freqz([1,1],[1,1],N,'whole');
h = h0 + h1;

mag = 20 * log10( abs(h)./max(abs(h)));

subplot(3,1,1)
plot((0:N/2)/N, mag(1:N/2+1))
xlabel('Frequency')
ylabel('Magnitude')

subplot(3,1,2)
phase = unwrap(angle(h));
plot((0:N/2)/N, phase(1:N/2+1))
xlabel('Frequency')
ylabel('Phase')

subplot(3,1,3)
g1 = grpdelay(b,a, N, 'whole');
g2 = grpdelay([1,1],[1,1], N, 'whole');
g = g1+g2;
plot((0:N/2)/N, g(1:N/2+1))
```

```
xlabel('Frequency')
ylabel('Group Delay (Sample)')
```



3.5





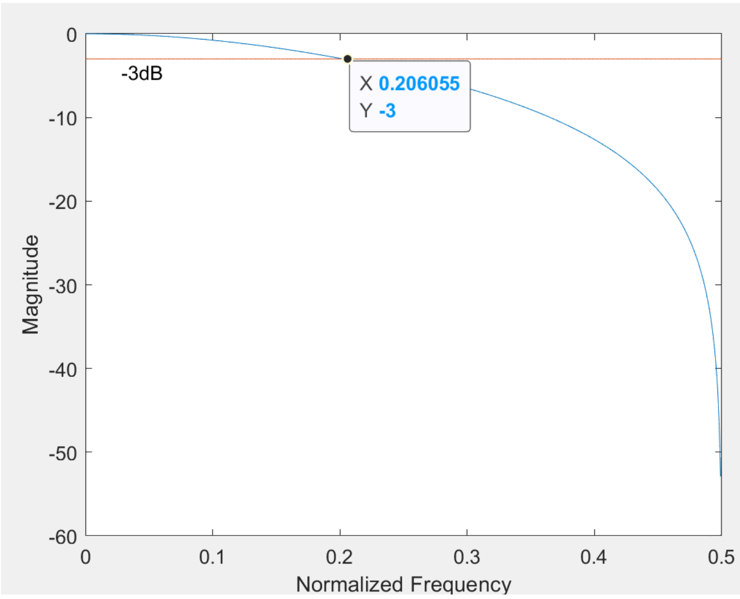
2.2 3.2

3.6 How does this lowpass filter compare with those created in 2.2 and 3.2? (Hint: if you are not getting a lowpass filter, again,

seek professional help.) For instance, how do the cut-off frequencies compare? What is different about the group delay? How does the computational complexity of this filter compare to the other two?

Comparison:

①cutoff: that is 0.206 in 3.6, which is similar to but less that in 2.2 (0.209). And it is also similar to but more than that in 3.2 (0.201)



②group delay: 3.6 less stable than FIR in 2.2, but more stable than 3.2. (Less fluctuation that 3.2)

③computational complexity: 2.2 > 3.6 > 3.2. As FIR need multiplies and adds, 2.2 > 3.6. As there are 2 IIR filter in 3.6, 3.6> 3.2.

3.7 In your own words, interpret Shakespeare's line to explain why two allpass filters in parallel may not an allpass filter make.

Although the allpass filters will let all frequency pass without change the amplitude of the frequency, they still will change the phase and group delay of the signal. When the signal pass 2 different allpass filter, it will become 2 different phase signals. And then, the sum of 2 signals will look like the filter change the amplitude of certain frequency.

3.8 Take a moment to reminisce with friends (Figure 4) about that crazy experience. This includes looking back at your answers while sipping tea and confirming that you understand now what was happening.

4.Conclution

In this lab, I benefit a lot.

Firstly, I start to know the group delay of the FIR is a constant. It means, the filter will not destroy the information in signal as the relative position of sample points will not change. In contrast, the group delay in IIR fluctuates and will destroy the information of signal.

Secondly, the M-point averager is one of the low-pass filters according to the plot. And it is one of the FIR.

Thirdly, although allpass filter can let all frequency of signal pass through without changing the magnitude, the combo of the allpass filters with different coefficients can produce different type of filters (e.g., high-pass filters, low-pass filters) by changing the phase.