Lab 4

Linear Systems and Digital Filtering

0. Abstract

A linear system is a mathematical model of a system based on the use of a linear operator. Linear systems typically exhibit features and properties that are much simpler than the nonlinear case. In this lab, we will mostly focus on the different kinds of filters(i e. FIR,IIR) and analyze their frequency response and group delay. And we will get to know the use of all pass filter.

1. Sections

1.1. The FFT using 1024 points of the frequency response

The code for the is as follows:

```
function exe1_1()
omega = 0:1/800:0.5;
a = omega * 2 * pi
H5 = (1/5)*(1-exp(-i*a*5))./(1-exp(-i*a));
H21 = (1/21)*(1-exp(-i*a*21))./(1-exp(-i*a));
H51 = (1/51)*(1-exp(-i*a*51))./(1-exp(-i*a));
plot(omega,[abs(H5);abs(H21);abs(H51)])
xlabel('Normalized Frequency')
ylabel('Magnitude')
gtext({'5';'21';'51'})
axis([0, 0.5, 0, 1])
```

The frequency response of the M-point averager :

$H(\omega) = (1/M)\,(1 - e^{(-j\omega M)})/(1 - e^{(-j\omega)})$.

We may be interested in the magnitude of this function in order to determine which frequencies get through the filter unattenuated and which are attenuated. Below is a plot of the magnitude of this function for M = 5 (blue), 21 (red), and 51 (orange).

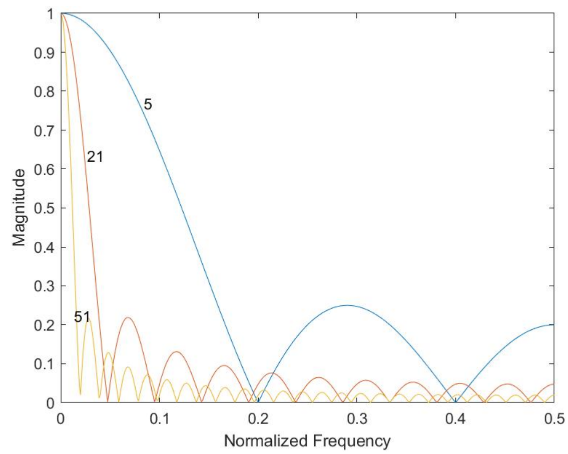


Figure 1: The FFT of the frequency response of M-point averager

1.2. Answer for jagged appearance

The passband decreases as the M in the moving averager increasing. When the frequency is less than 0.02 Normalized Frequency, it can be unaltered in all three filters.

1.3. The phase responses in units of pi radians

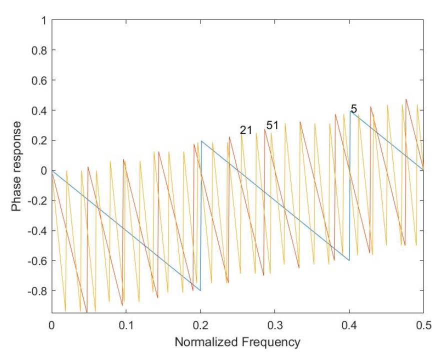


Figure 2: The phase responses in units of pi radians

1.4. Explanation for the jagged appearance

Because frequency response of a causal length NN moving average filter is:

$$H(\omega) = \frac{\sin\left(\frac{N\omega}{2}\right)}{N\sin\left(\frac{\omega}{2}\right)} e^{-j\omega(N-1)/2} = A(\omega)e^{j\phi(\omega)}$$

However the phase we plotted in matlab is not ϕ(ω)ϕ(ω), but ϕ^(ω)ϕ^(ω) as defined by:

$$H(\omega) = |A(\omega)|e^{j\hat{\phi}(\omega)}$$

The difference between ϕ(ω) and ϕ^(ω) is that whenever A(ω) crosses zero, a phase jump of ±π±π occurs in ϕ^(ω), corresponding to a sign change in A(ω) and this phase jump is the reason why the phase response has a jagged appearance.

1.5. Plot the magnitude response of the two-dimensional moving average filter

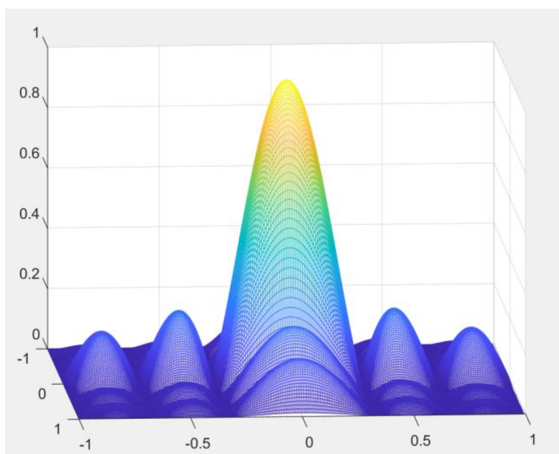
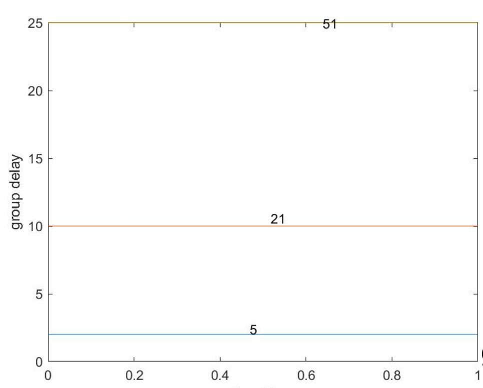


Figure 3: The phase responses in units of pi radians

The is like extending the figure we find in 1.1 into 4 directions. When we just look the figure when y =0, x>0, the figure is exactly the same as figure created in 1.1.

1.6. Plot the group delay of the average filter



g

Figure 4: The group delay of the moving average filter in different M

1.7 Load the sound file and zeropad each of them

The L1 and L2 represent the length of the ‘drumloop2.wav’ and ‘KillaloeCathedral.wav ’ respectively. We make the length of the F1 and F2 the same is to make pointwise multiply possible. The code and comments for each line are as follow:

```
function exe1_7()
[x1,Fs1] = audioread('drumloop2.wav');
[x2,Fs2] = audioread('KillaloeCathedral.wav');
L1 = length(x1);
L2 = length(x2);
L = L1 + L2 - 1;
F1 = fft(x1,L);
F2 = fft(x2,L);
F_mul = F1 .* F2;
a = ifft(F_mul);
soundsc(a)
```

1.8 Listen and explain the results

What we do previous is to fft both wav first and then pointwise multiply then use the IFFt to play this sound. The first original songs is clear and melodiou, and the second song is like something falls onto the ground. So after multiplying them, the sounds becomes gravy.

2.1 Plot the impulse response

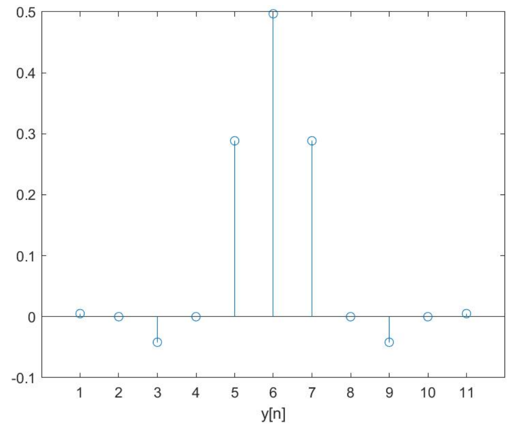


Figure 6: The impulse response

2.2 Find and plot the frequency response and the group delay

The code for the frequency response and the group delay are as follow:

```
function exe2_2()
N = 1024;
fs = 1
f=(0:N-1)*fs/N;
n=0:N-1;
x1 = [0.00506, 0, -0.04194, 0, 0.28848, 0.49679, 0.28848, 0, -0.04194, 0,
0.00506]
y1=fft(x1,N);
subplot(2,1,1);
n=0:N-1;
mag = 20 * log10(abs(y1) ./ max(abs(y1)))
T = ones(1,N) .* (11 - 1) ./ (2 )
subplot(2,2,1); plot(f(1:N/4),T(1:N/4),'-')
xlabel('Normalized frequency')
ylabel('Group delay(samples)')
subplot(2,2,2);
plot(f(1:N/2),mag(1:N/2));
title('Magnitude response');
xlabel('Normalized Frequency');
ylabel('dB');
subplot(2,2,[3 4]);
plot(f(1:N/2),angle(y1(1:N/2)),'.');
title('phase response');
xlabel('Normalized Frequency');
ylabel('arg');
```

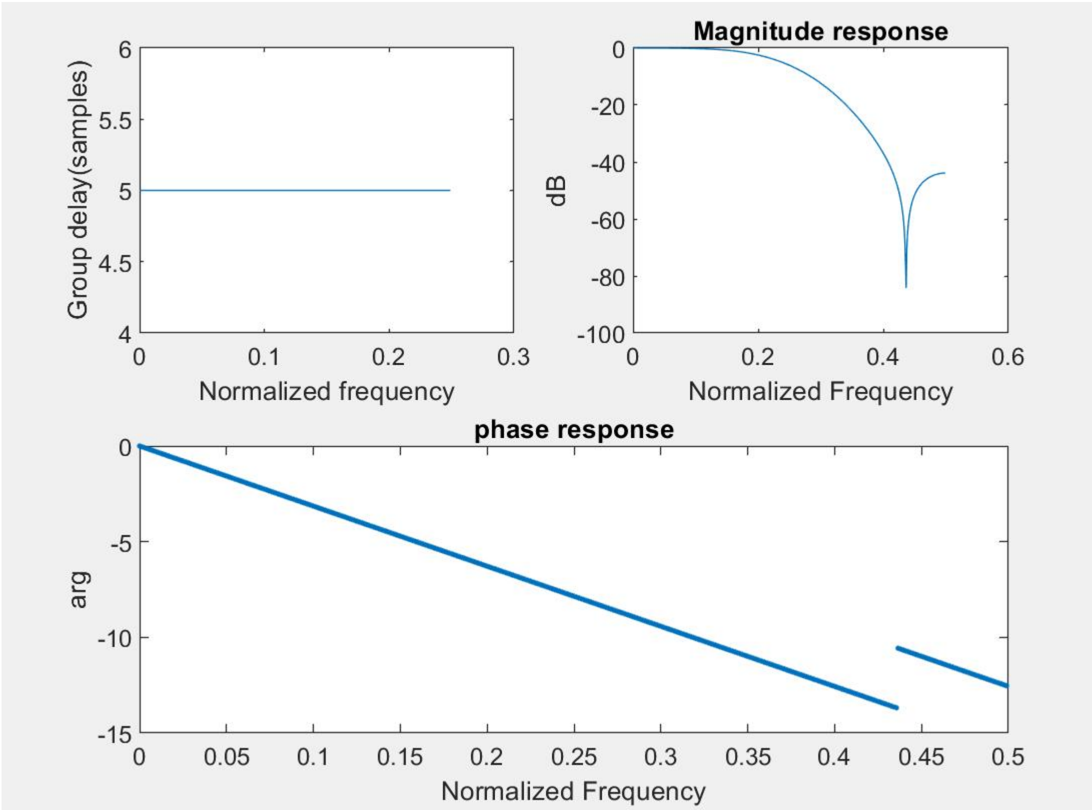


Figure 6: (1) The group delay (2) The frequency magnitude response (3) The frequency phase response

2.3 Answer for question

This filter is a low-path filter because most signal with lower frequencies. When x is equal to 0.21 normalized frequency, it is approximately the “3dB” point.

2.4 The group delay for all frequencies

The group delay this filter is about 5 samples delay, this kind of delay won’t destroy the information since it is a linear phase filter.

2.5 The graph multiplied a series of alternating positive and negative ones

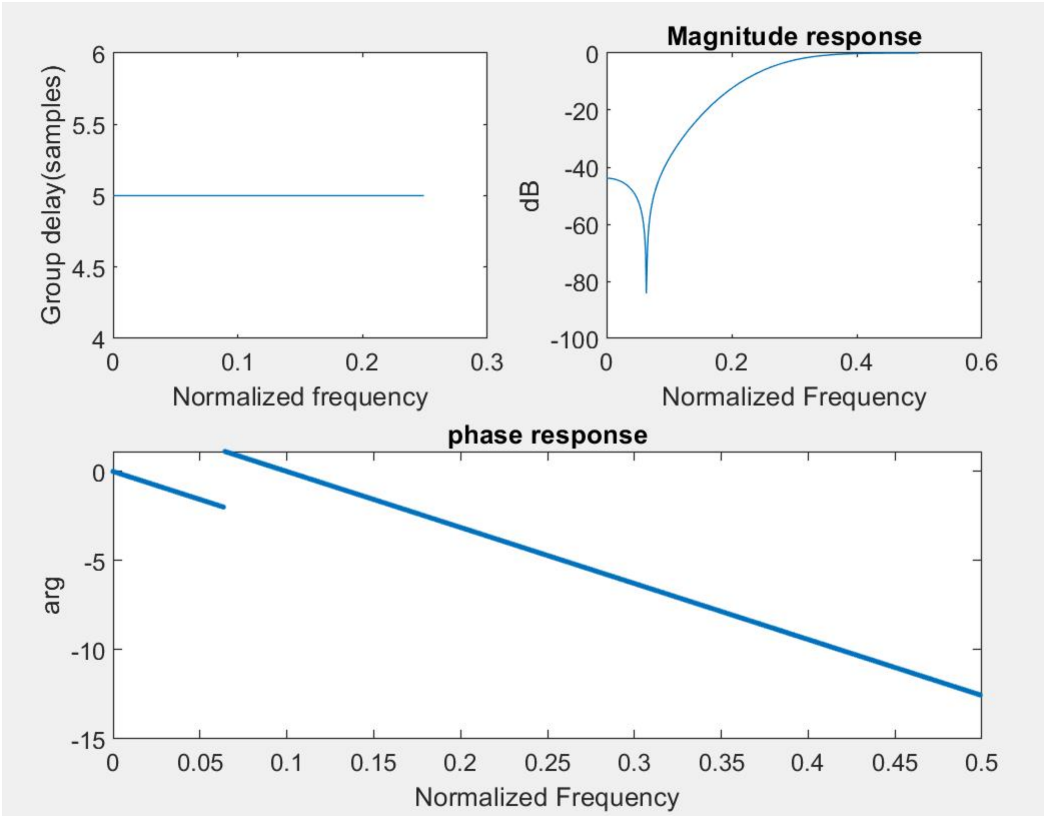


Figure 7: (1) The group delay (2) The frequency magnitude response (3) The frequency phase response

2.6 Answer for the question

After multiplying by a series of alternating positive and negative one, it has become a high-pass filter now. The cut-off frequency is approximately to 0.3 normalized frequency.

2.7 Explain what is happening

Because when we multiply the impulse respnse of the lowpss FIR filter by a series of alternating positive and negative 1's. And since the impulse response is symmetric and the value on the even size is zero and the only response it changes is h[6]. Because it turns from the positive to negative, it causes the filter to be a high pass filter.

2.8 Apply the filter to the sound

After using the conv function in the Matlab, we use different h[n] to filter it. When we use the h1[n](which is the low path filter), the sound is very similar to the original one. But when wee use h2[n] (which is a high path filter), the sound is very low and it filters a lot of low-frequency sound.

2.9 Find the band of frequency above the cut-off frequencies

The frequency Fs is 44100Hz. When using the low path filter, the frequency is lower than 9261Hz can be preserved from the original drum signal. When using the high path filter, the frequency that is higher 13230Hz is preserved

3.1 Find and plot the first sixteen samples of impulse response of the system

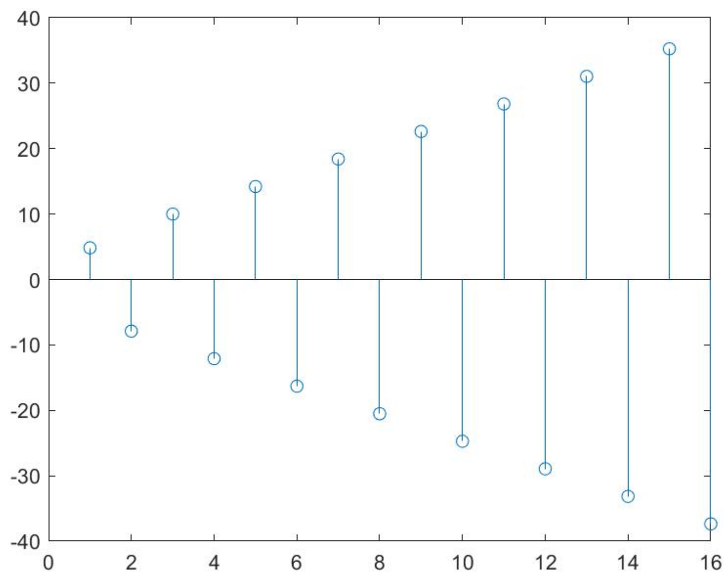


Figure 7: The first sixteen samples of impulse response of this system

3.2 Plot the frequency response and group delay for this system

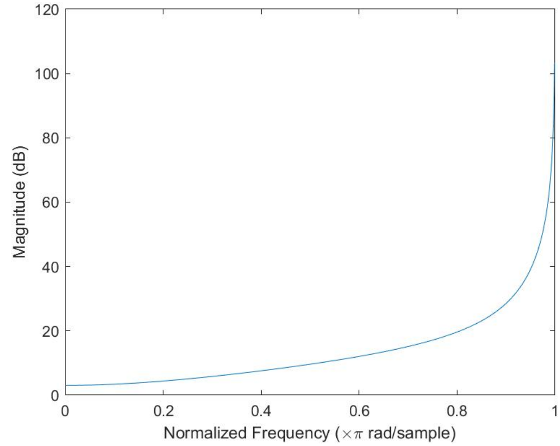


Figure 8: The frequency response

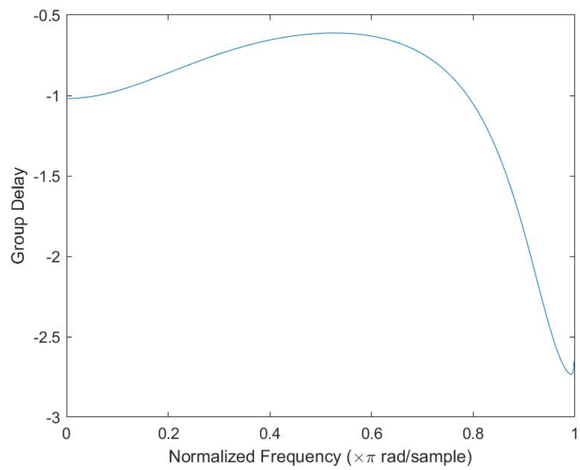


Figure 9: The group delay for the IIR system

3.3 Compare the results with FIR system

In the FIR filter, it is a linear pase filter so the delay is the same in all frequencies. But in IIR, the group delay is different in different frequencies and it also has a narrower band. The computational complexity of IIR filter is less than FIR. If the carrier has been modulated in amplitude, the filter can not filter the signal interest.

3.4 Plot the magnitude, phase response, and group delay of the filter

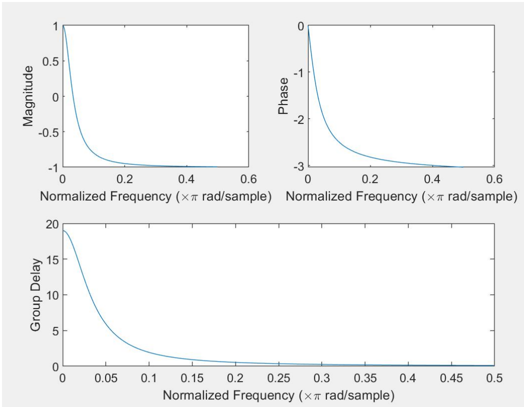


Figure 9: The magnitude, phase response, and group delay of the filter

3.5 The allpass filter in parallel artfully

The code is as follows:

```
function exe3_5()
a = [1 -0.9 ];
b = [-0.9 1];
a1 = [1];
b1 = [1]
[h1,w2] = freqz(b1,a1,1024);
[h,w] = freqz(b,a,1024);
[gp,w1] = grpdelay(b,a,1024);
[gp1,w3] = grpdelay(b,a,1024);
N = length(h)
N1 = length(gp)
subplot(2,2,1);plot(w(1:(N/2))/pi,0.5 * (h(1:(N/2))+h(1:(N/2))));
xlabel('Normalized Frequency (\times\pi rad/sample)')
ylabel('Magnitude')
subplot(2,2,2);plot(w(1:(N/2))/pi,unwrap(0.5
*(angle(h(1:N/2))+angle(h1(1:N/2)))))
xlabel('Normalized Frequency (\times\pi rad/sample)')
ylabel('Phase')
subplot(2,2,[3 4]); plot(w1(1:(N1/2))/pi,0.5 * (gp(1:(N1/2))+gp1(1:(N1/2))))
xlabel('Normalized Frequency (\times\pi rad/sample)')
ylabel('Group Delay')
```

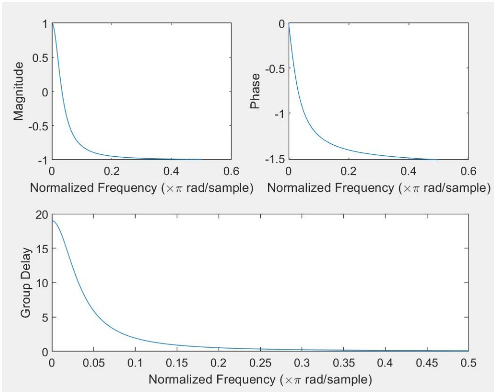


Figure 10: The magnitude, phase response, and group delay of the filter

3.6 Answer to the question

Both this filter and the one in the 3.2 are low pass filter. This filter has relative less group delay and the cutoff frequency is lower than what it is in the 3.2. This filter has the less computational complexity compare to the other two.

3.7 Interpret the verse

An allpass filter is a filter with a unity gain across all frequencies. This means that no frequency passing through that filter will be boosted or attenuated. It introduces, however, a frequency-dependent delay.

3.8 Answer to the question

I have known the importance of group delay.

2. Conclusion

In lab4, we get start with the moving average filter and then we use the function to calculate the group delay. Then we move to the Finite Impulse Response Filters and Infinite Impulse Response Filters. Then comparing the computational complexity between them. At last we analyze the frequency response of the all pass filter and get to know the use of the all pass filter.

3. Acknowledgments

Thanks for the TA gives the instruction on Lab4.