

# EE141 Digital Signal Processing

## Lab 3: Basic Second Order Filters

### Lab Section: 022

Name: Buddy Ugwumba SID: 862063029

#### Objective

Gain hands on experience processing a real-world music signal.

```
% Loads the MATLAB file
load ride
% Use audioplayer function to load the file
p = audioplayer(audio_sample, fs);
play(p)
```

a) For general  $a$  and  $c$ , determine the impulse response  $h[n]$

$$H(z) = \frac{C}{(1-az^{-1})(1-a^*z^{-1})}$$

$a^*$  = same  $x$  position  
but  $y$  is opposite

$$= \frac{A}{1-az^{-1}} + \frac{B}{1-a^*z^{-1}} \Rightarrow A a^n u[n] + B(a^*)^n u[n]$$

$$= A(1-a^*z^{-1}) + B(1-az^{-1}) \Rightarrow 0$$

$$A - Aa^*z^{-1} + B - Baz^{-1} \Rightarrow 0$$

$$-Aa^*z^{-1} - Baz^{-1} + A + B$$

$$\begin{aligned} c: & \quad A = \frac{C}{1-\frac{a^*}{a}} & B = \frac{C}{1-\frac{a}{a^*}} \\ H & \quad = \frac{C}{\frac{a-a^*}{a}} & = \frac{C}{\frac{a^*-a}{a^*}} \\ & \quad = \frac{aC}{a-a^*} & = \frac{a^*C}{a^*-a} \end{aligned}$$

b) Appropriately choosing  $a$  and  $c$  to make  $h[n]$  a lowpass filter.

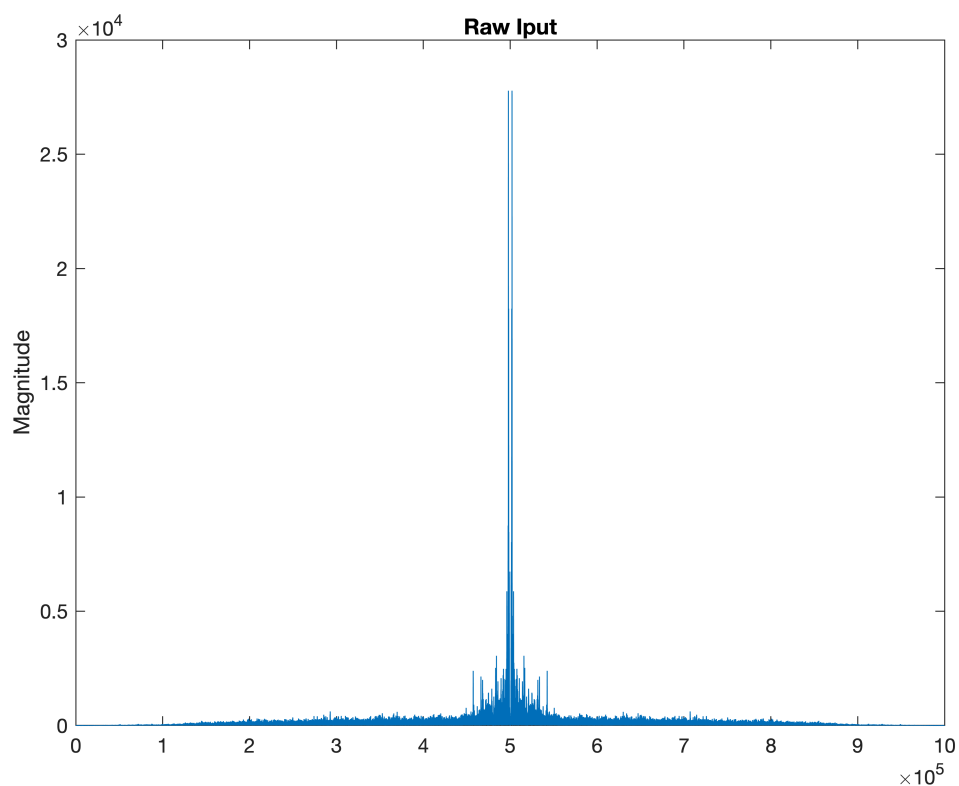
```
a = 0.8 - 0.001j;
c = (1-a)*(1-conj(a));
% Apply h[n] to x[n] (music signal)
% h => conv()
n = 0:100;
% impulse response h[n]
h1 = ((c/(1-(a/conj(a)))).*a.^(n)) + ((c/(1-(conj(a)/a)))).*conj(a).^(n));

% audio_sample(:,1) => first channel
Y_1(:,1) = conv(audio_sample(:,1), h1, 'same');
% the second output
Y_1(:,2) = conv(audio_sample(:,2), h1, 'same');
p2 = audioplayer(Y_1, fs);
%play(p2)
```

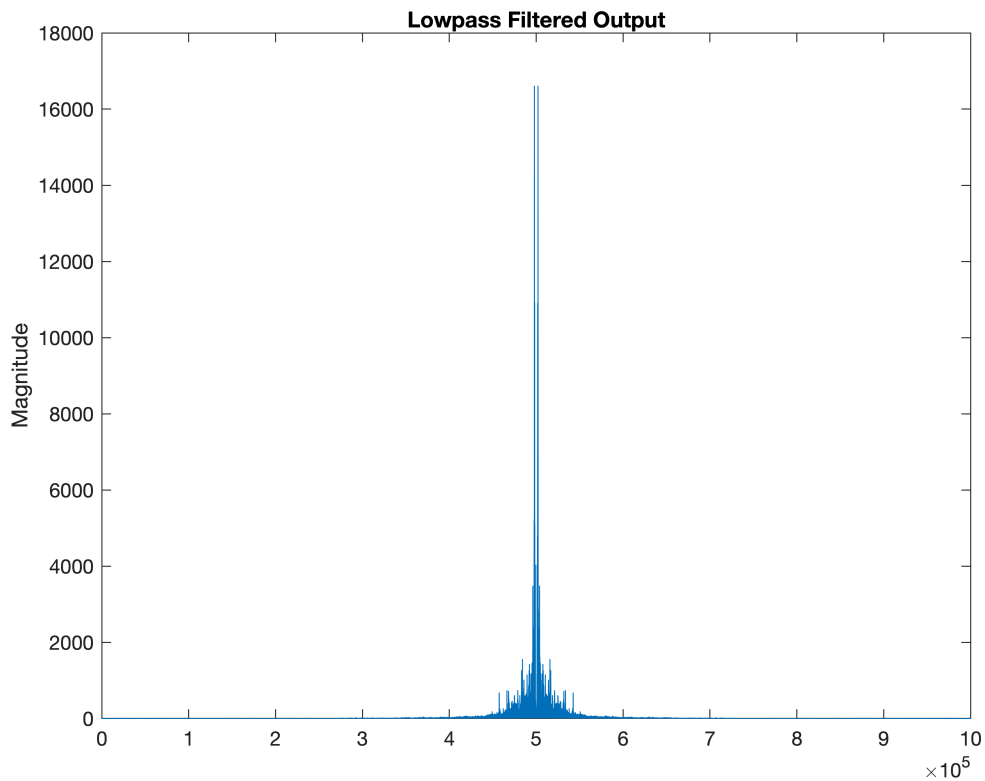
Yes, I do hear a muffled version of the original audio file.

c) Using the fft command, take the DTFT of both the input and the output of one of the channels. Plot their magnitudes side by side.

```
figure
plot(fftshift(abs(fft(audio_sample(:,1)))))
title('Raw Input')
ylabel('Magnitude')
hold on;
```



```
figure
plot(fftshift(abs(fft(Y_1(:,1)))))
title('Lowpass Filtered Output')
ylabel('Magnitude')
```



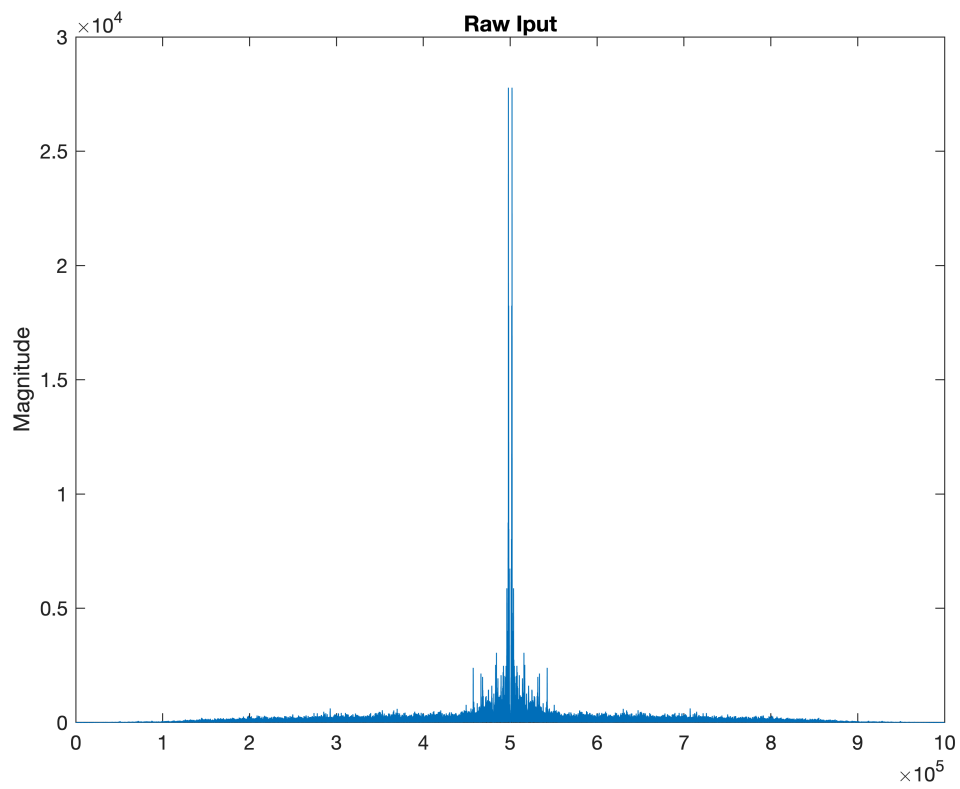
Yes I see the high frequencies suppressed at the tail end of the plot. The delay, after passing the signal through the low-pass filter is seen on the x-axis. This is why the signal starts at 5.

d) Repeat b and c with a highpass filter. '

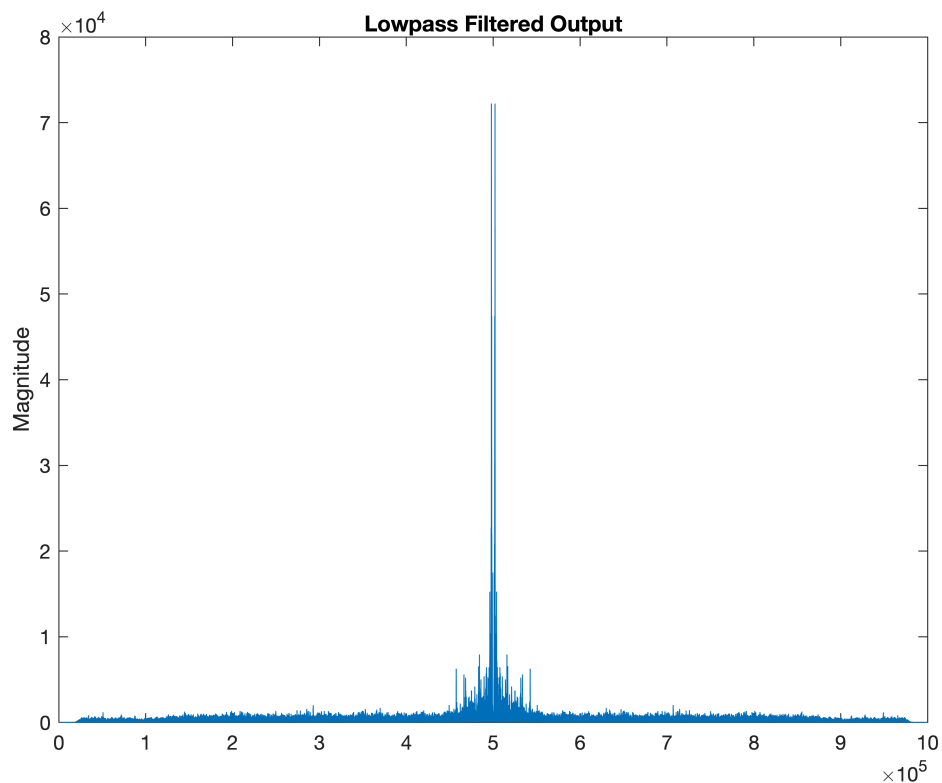
```
a = -0.8 - 0.001j;
c = (1-a)*(1-conj(a));
% Apply h[n] to x[n] (music signal)
% h => conv()
n = 0:100;
% impulse response h[n]
h1 = ((c/(1-(a/conj(a)))).*a.^(n)) + ((c/(1-(conj(a)/a)))).*conj(a).^(n));

% audio_sample(:,1) => first channel
Y_1(:,1) = conv(audio_sample(:,1), h1, 'same');
% the second output
Y_1(:,2) = conv(audio_sample(:,2), h1, 'same');
p2 = audioplayer(Y_1, fs);
%play(p2)

figure
plot(fftshift(abs(fft(audio_sample(:,1)))))
title('Raw Input')
ylabel('Magnitude')
hold on;
```



```
figure
plot(fftshift(abs(fft(Y_1(:,1)))))
title('Highpass Filtered Output')
ylabel('Magnitude')
```



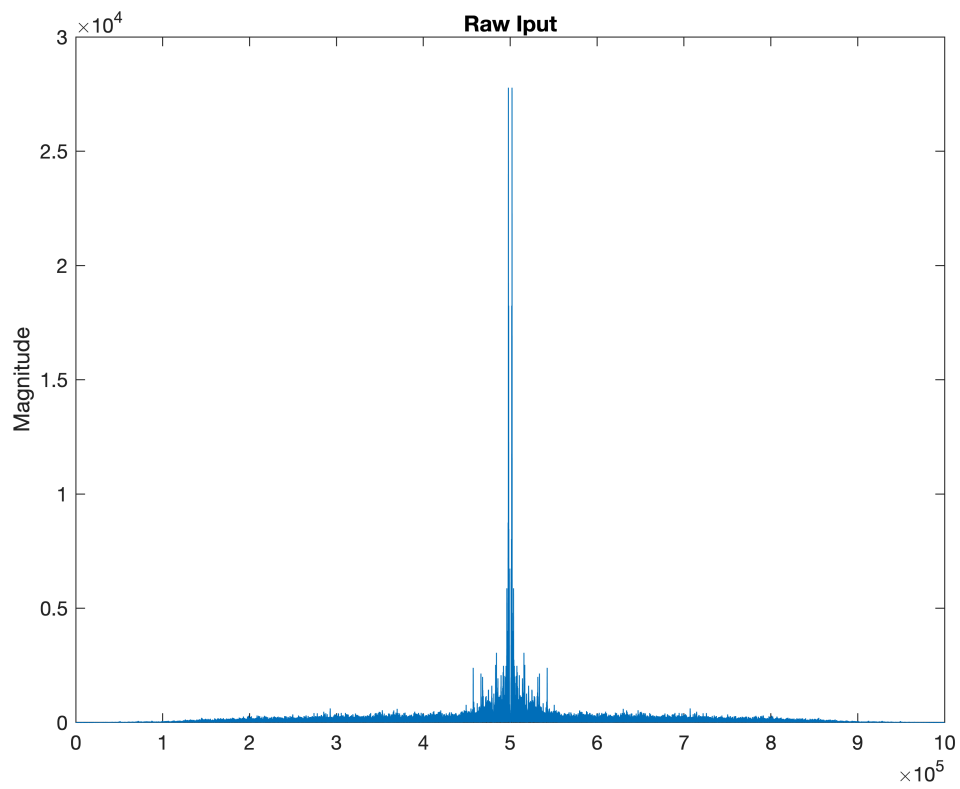
Yes, I do see that the high frequencies are more prominent.

e) Repeat b and c with a bandpass filter.

```
a = 0 - 0.001j;
c = (1-a)*(1-conj(a));
% Apply h[n] to x[n] (music signal)
% h => conv(1j
n = 0:100;
% impulse response h[n]
h1 = ((c/(1-(a/conj(a)))).*a.^(n))+ ((c/(1-(conj(a)/a)))).*conj(a).^(n));

% audio_sample(:,1) => first channel
Y_1(:,1) = conv(audio_sample(:,1), h1, 'same');
% the second output
Y_1(:,2) = conv(audio_sample(:,2), h1, 'same');
p2 = audioplayer(Y_1, fs);
play(p2)

figure
plot(fftshift(abs(fft(audio_sample(:,1)))))
title('Raw Input')
ylabel('Magnitude')
hold on;
```



```
figure
plot(fftshift(abs(fft(Y_1(:,1)))))
title('Bandpass Filtered Output')
ylabel('Magnitude')
```

