ELEC 221 Project

Let’s Make it Sound Better

Tabitha Lee

31079122

April 7th, 2016

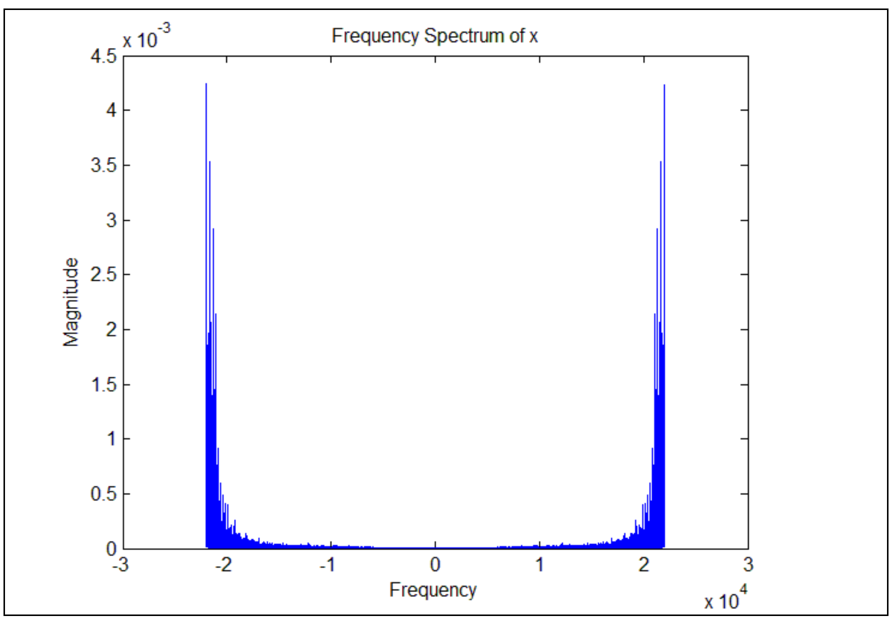
Task 1

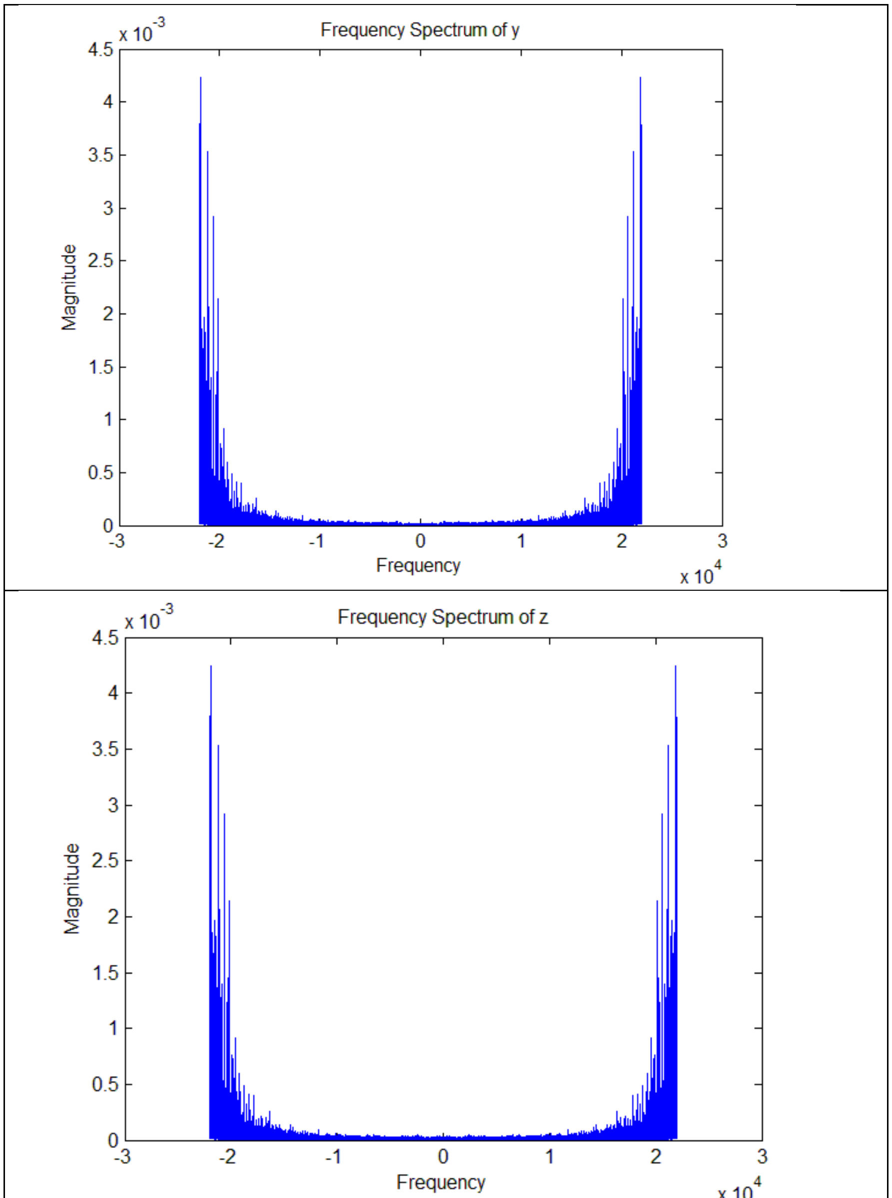
```
>> [x,fs]=wavread('audio_signal_1.wav');
>> sound(x);
```

Q0

The sampling frequency of x is fs=44100Hz, at 16 bits per sample

```
>> N = length(x);
>> y=x;
>> y=resample(y, 1, 2);
>> z=x;
>> z=downsample(x,2);
>> a = fft(x);
>> b =fft(y);
>> c = fft(z);
>> fr=(-N/2:N/2-1)*fs/N;
>> plot(fr, abs(a/N));;
>> N = length(b);
>> fr=(-N/2:N/2-1)*fs/N;
>> plot(fr, abs(b/N));
>> N = length(c);
>> fr=(-N/2:N/2-1)*fs/N;
>> plot(fr, abs(c/N));
```





P0 frequency spectra of x, y, and z

Q1

Since the signals are real, the magnitude of the signal after taking the fast fourier transform are symmetric across the y-axis, which means x, y, and z all have conjugate symmetry

Q2

The frequency spectra of y and z are different because downsample deletes the values in the sample between each Nth sequence whereas resample applies an antialiasing FIR lowpass filter to x then compensates for the delay.

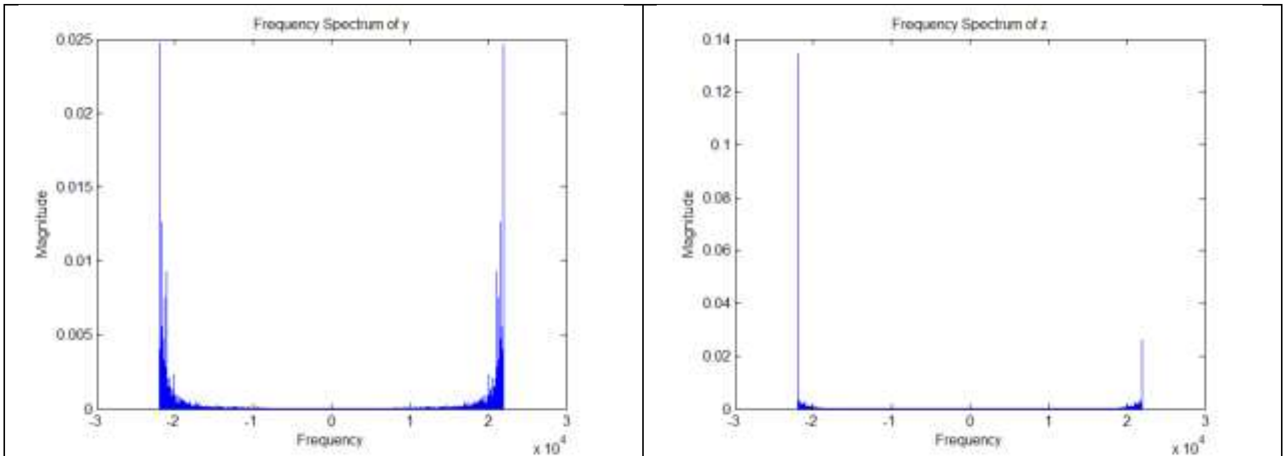
Task 2

```
>>x = wavread('audio_signal_1.wav');
>>h = wavread('room_impulse.wav');
>>y = conv(x,h);
>>z = MyConv(x,h);
>>a = fft(y); N=length(a); fr=(-N/2:N/2-1)*fs/(N); plot(fr,abs(a/N));
>>b = fft(z); N=length(b); fr=(-N/2:N/2-1)*fs/(N); plot(fr,abs(b/N));
>>audiowrite('1room_y.wav', y, fs);
>>audiowrite('1room_z.wav', z, fs);

>>h = wavread('stadium_impulse.wav');
>>y = conv(x,h);
>>z = MyConv(x,h);
>>a = fft(y); N=length(a); fr=(-N/2:N/2-1)*fs/(N); plot(fr,abs(a/N));
>>b = fft(z); N=length(b); fr=(-N/2:N/2-1)*fs/(N); plot(fr,abs(b/N));
>>audiowrite('1stadium_y.wav', y, fs);
>>audiowrite('1stadium_z.wav', z, fs);
```

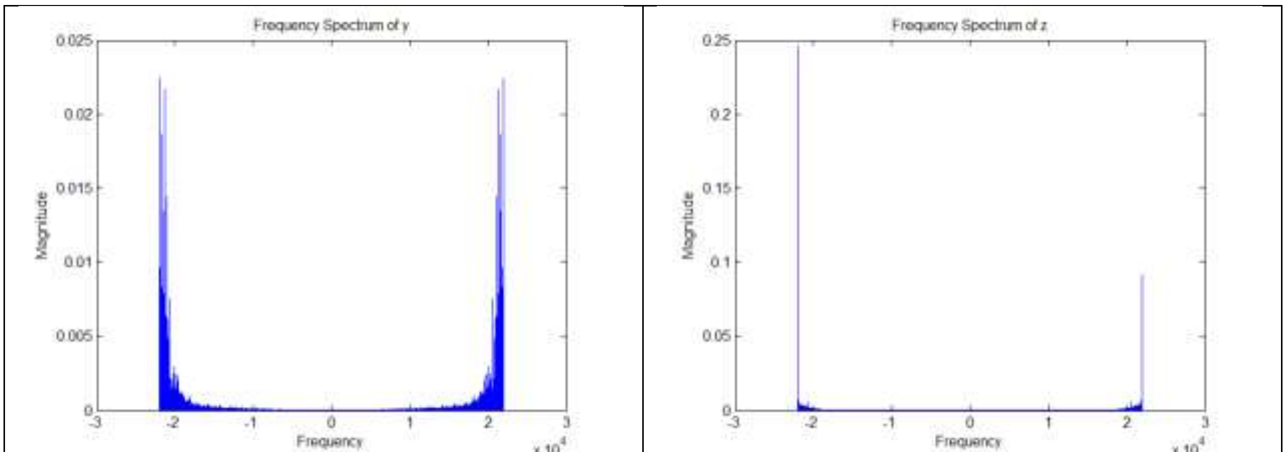
Q3

```
function f = MyConv(x,h)
%MYCONV Summary of this function goes here
%   Detailed explanation goes here

lx=length(x);
lh=length(h);
x2=[x,zeros(1,lh-1)];
h2=[h,zeros(1,lx-1)];
a=fft(x2);
b=fft(h2);
c=a.*b;
d=ifft(c);
f=abs(d);
end
```

```
y = conv(x,h);
z = MyConv(x,h);
```



P1 audio_signal_1 convoluted with room_impulse



P1 audio_signal_1 convoluted with stadium impulse

Q3

Even if the sampling frequencies were different, as long as the frequencies are low enough and there is no aliasing, the two signals can still be convoluted. However, if aliasing were to occur, I would approach it with a low-pass filter to filter out the higher frequencies.

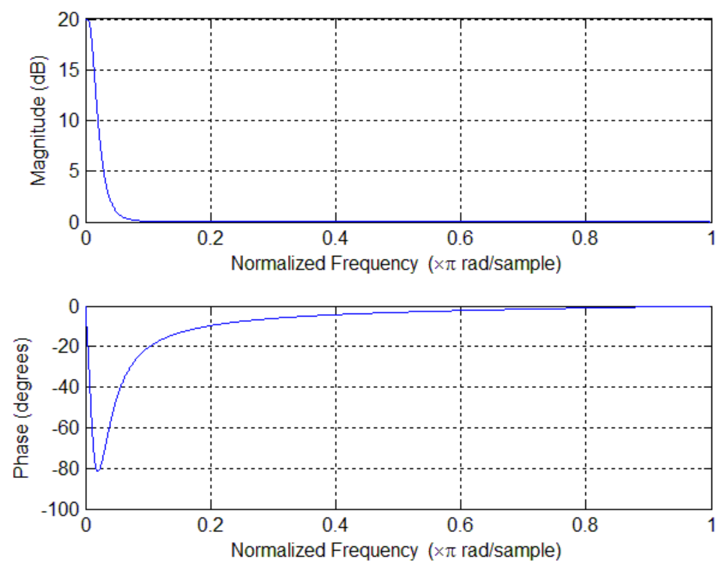
Task 3

Q4

The order of both the boosting filters is 2.

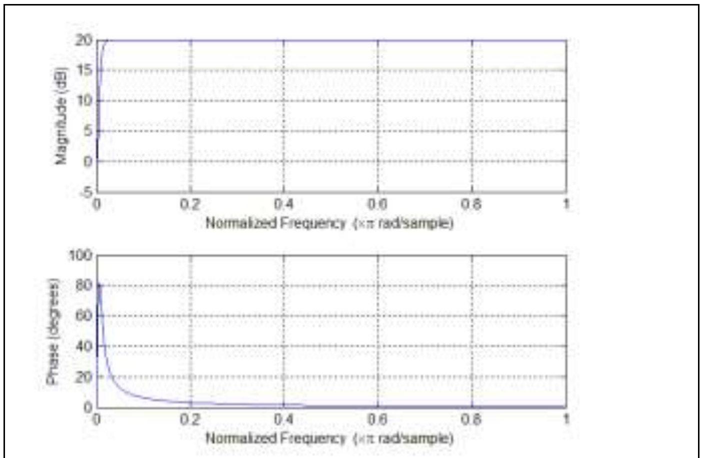
```
>> C= tan(pi*250/44100); Gl = 10; r = sqrt(2);
>>  b0=(1+sqrt(Gl)*r*C+Gl*C^2)/(1+r*C+C^2);
>> b1=(2*(Gl*C^2-1))/(1+r*C+C^2);
>> b2 = (1-sqrt(Gl)*r*C+Gl*C^2)/(1+r*C+C^2);
```

```
>> a0 =1;
>> a1 = (2*(C^2-1))/(1+r*C+C^2);
>> a2 = (1-r*C+C^2)/(1+r*C+C^2);
>> a=[a0,a1,a2];
>> bb = [b0,b1,b2];
>> freqz(b,a,441000)
```



P2 Frequency response of bass boost

```
>> C= tan(pi*4000/44100);
>> b0=(Gl+sqrt(Gl)*r*C+C^2)/(1+r*C+C^2);
>> b1=(2*(C^2-Gl))/(1+r*C+C^2);
>> b2=(Gl-sqrt(Gl)*r*C+C^2)/(1+r*C+C^2);
>> bt=[b0,b1,b2];
>> freqz(b,a,441000);
```



P3 Frequency response of treble boost

Q5

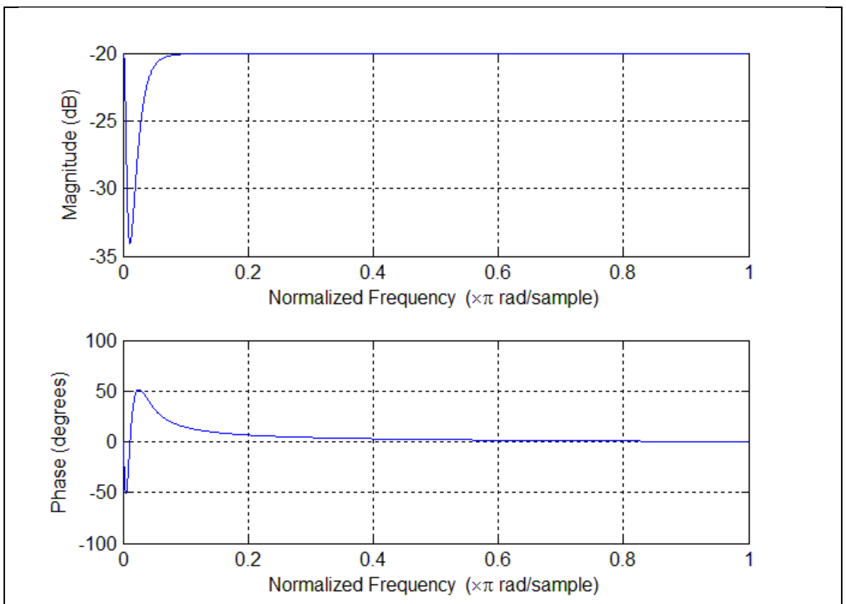
The bass boost is a low-pass filter, while the treble boost is a high-pass filter. This is evident in the graphs above. The bass filter cuts off at higher frequencies whereas the treble boost filter cuts off at lower frequencies.

Q6

$a_{new} = a_b * a_t = [1, -3.8993, 5.7028, -3.7077, 0.9042]$

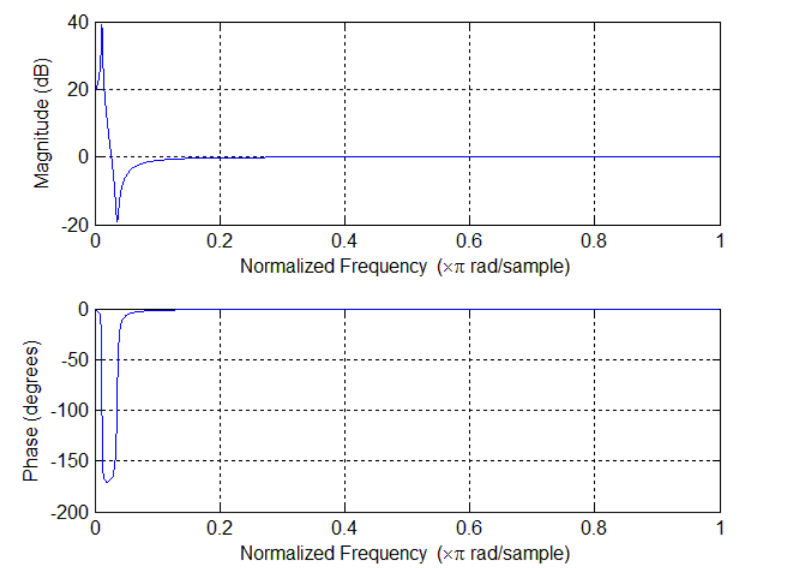
$b_{new} = b_b * b_t = [10.3787, -39.7008, 56.9795, -36.3692, 8.7118]$

The bass and treble filter is a band-stop filter. The frequencies below the cut-off frequency of the treble boost and above the cut-off frequency of the bass boost are attenuated and every other frequency is amplified.

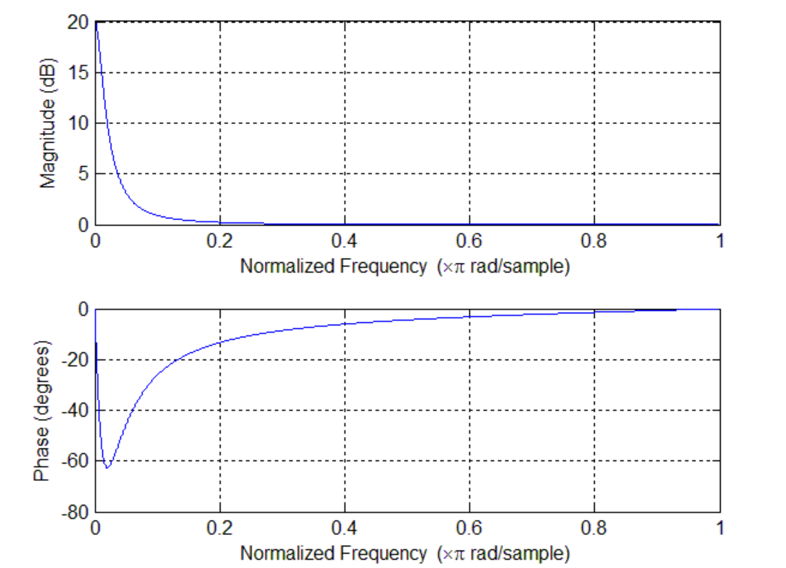


P4 Frequency response of bass then treble boost

Q7

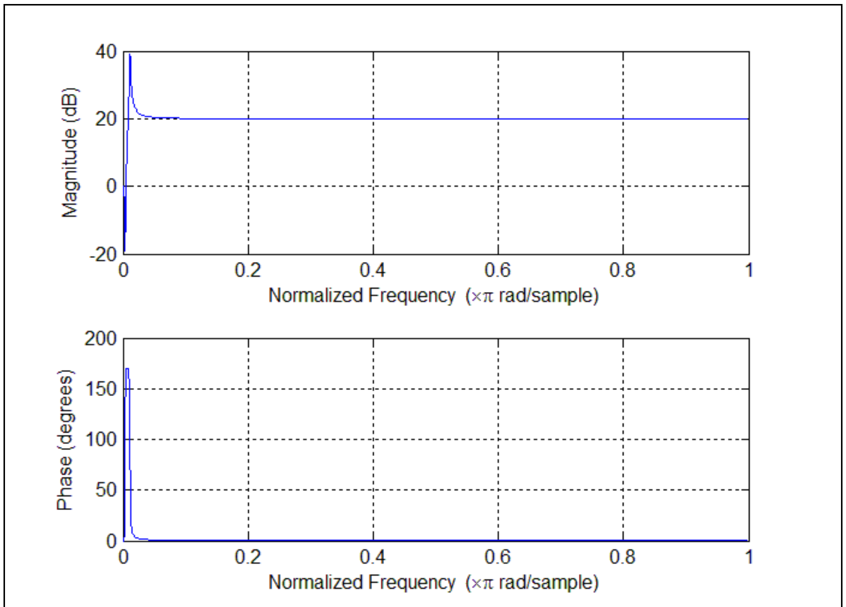


r=0.1, bass boost

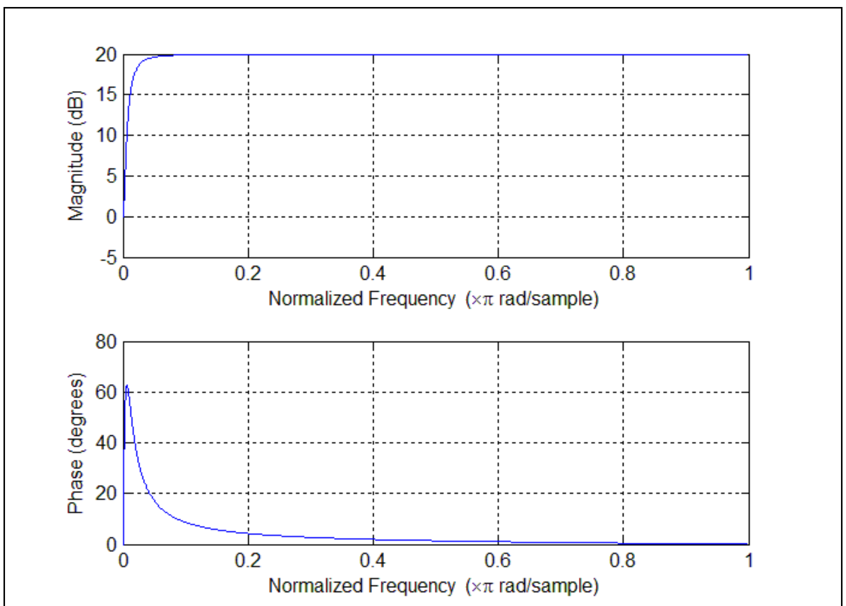


P4 r=2, bass boost

Bass boost: Decreasing the r value below 0.5 introduces a notch filter and doubles the magnitude of the frequency response for low frequencies. Increasing the r value smoothens the signal.



r=0.1, treble boost



r=2, treble boost

Similarly, decreasing the r value to below 0.1 introduces a notch filter and doubles the magnitude of the frequency response for low frequencies and increasing the r value smoothens the signal.

It is observed that better results for smoother signals can be attained by increasing the r value.