SIGNALS AND SYSTEMS

MATLAB – SHELF FILTERS

BY: AHAD AL HASSAN

Introduction

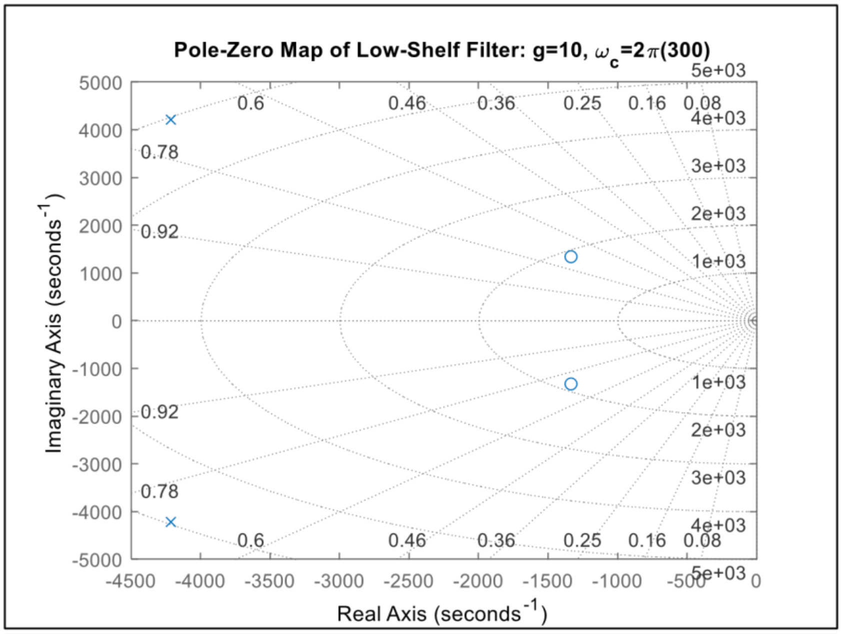
The purpose of this lab is to work with a special kind of low-pass & high-pass filers called shelf filters. A shelf filter is used to amplify or attenuate the signals over a defined frequency while leaving other frequencies in the signal unchanged. The purpose of this lab is to design and work with low-shelf and high-shelf filters.

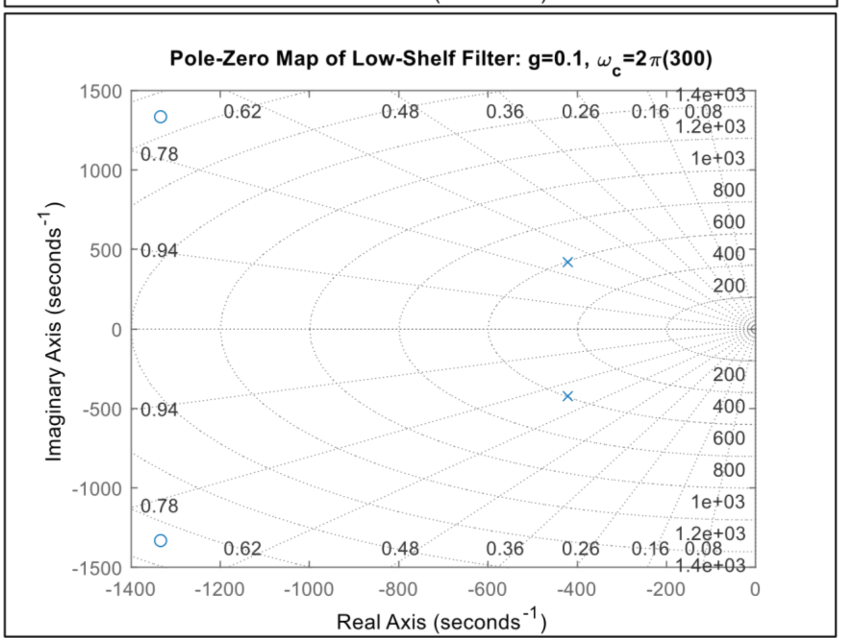
2. Design of a Low-Shelf Filter

(a) Designing a shelf-filter

Refer to MATLAB reference page in lab report to see Low-Shelf filter details (b)

Obtaining pz-maps from respective transfer functions



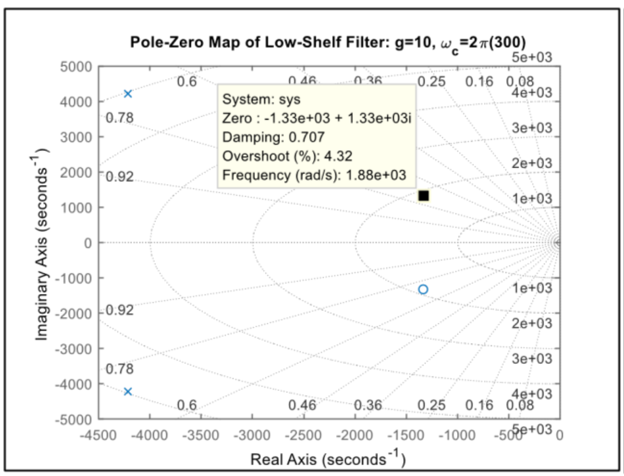


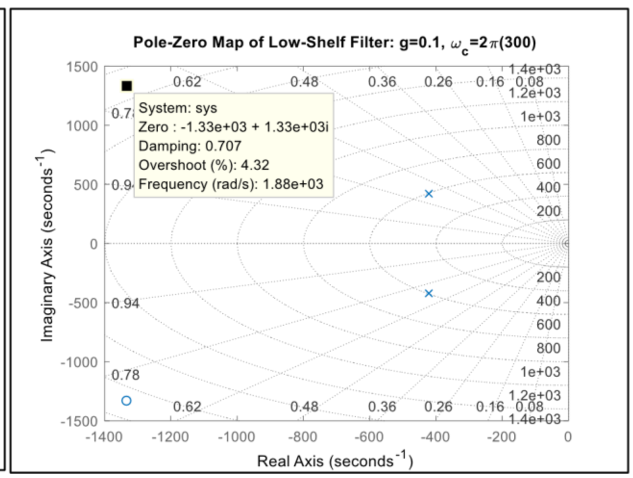
Where the transfer function of the low-shelf filter is

$$H_{ls}(s) = \frac{(s + \omega_c\sqrt{g}e^{j\pi/4})(s + \omega_c\sqrt{g}e^{-j\pi/4})}{(s + \omega_c e^{j\pi/4})(s + \omega_c e^{-j\pi/4})}$$

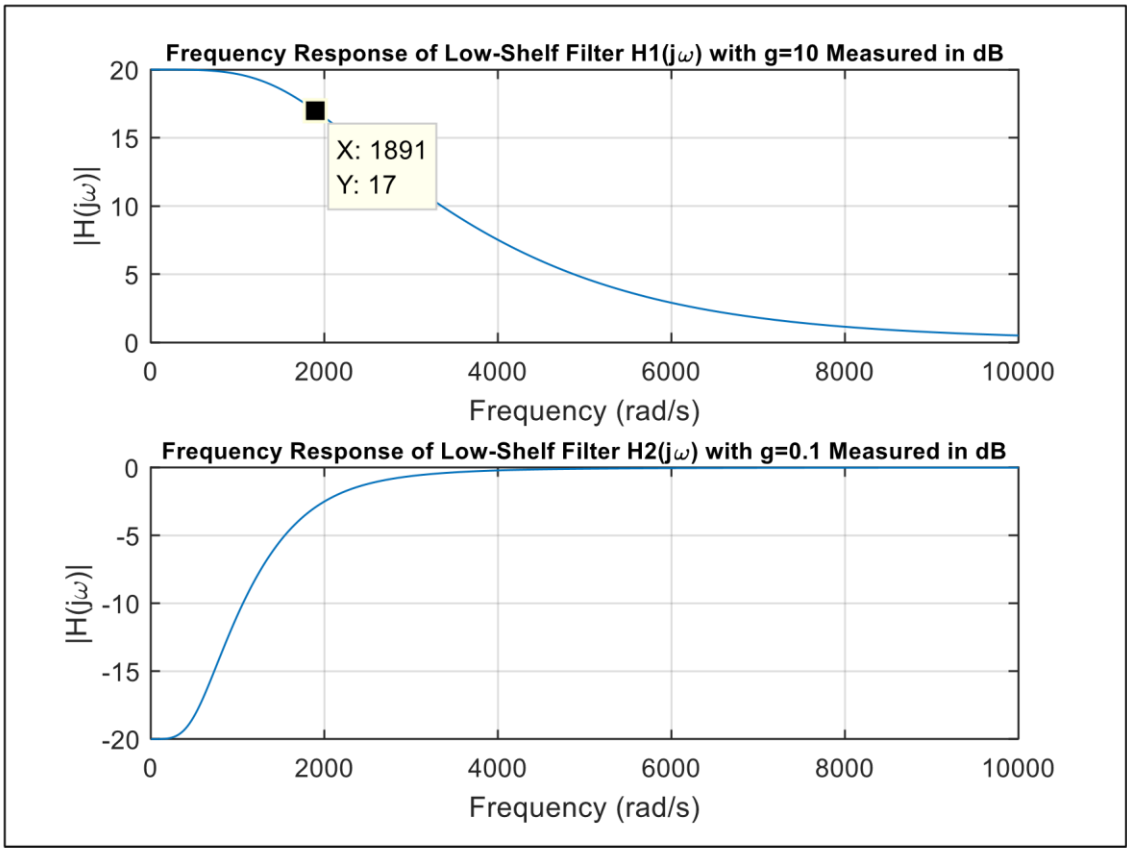
If we take a moment to analyze the transfer function of the low-shelf filter we can see that the gain factor g only impacts the numerator; the zeros of the transfer function. Thus, any changes made to the gain only impacts the zeros of the pz map in MATLAB while leaving the poles unchanged. Thus, any modifications made to the gain value only changes the zero's location/distance relative to the $j\omega$ axis. The same can be said of the high shelf filter.
This is evident from the pz-map as shown above. For the first pz-map, the poles exist between -1500 and -1000 on the x-axis while on the second graph, we can see more clearly that the poles are located around -1400 on the x-axis. So, the only thing that changes during the modification of the gain value is the location of the zeros.

We can see more clearly below that the actual points themselves for the poles don't change, only the zeros.





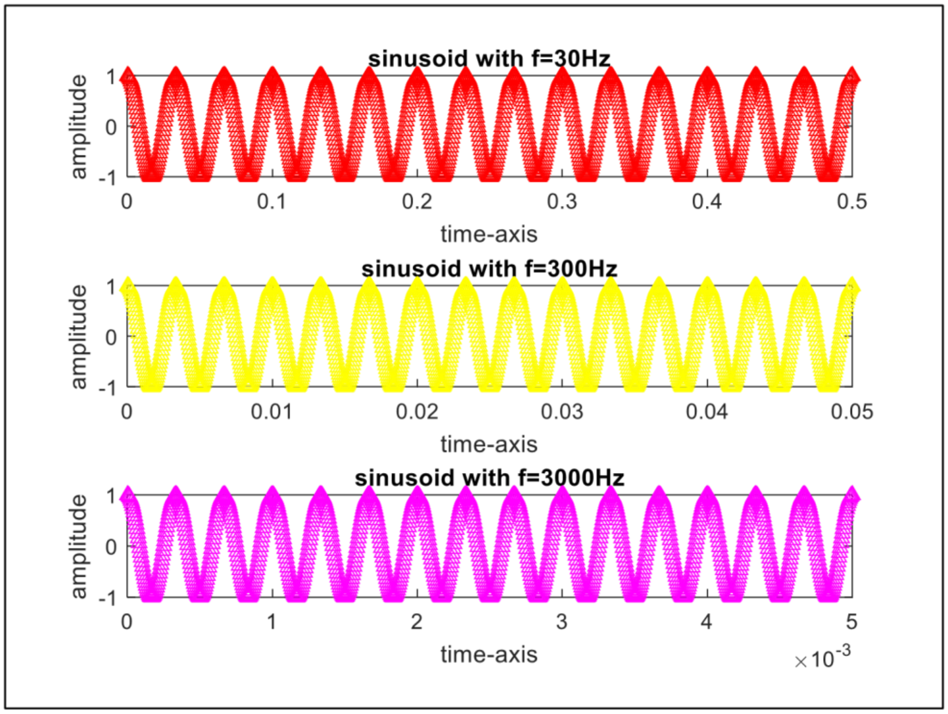
(c) Frequency Response of Low-Shelf Filter

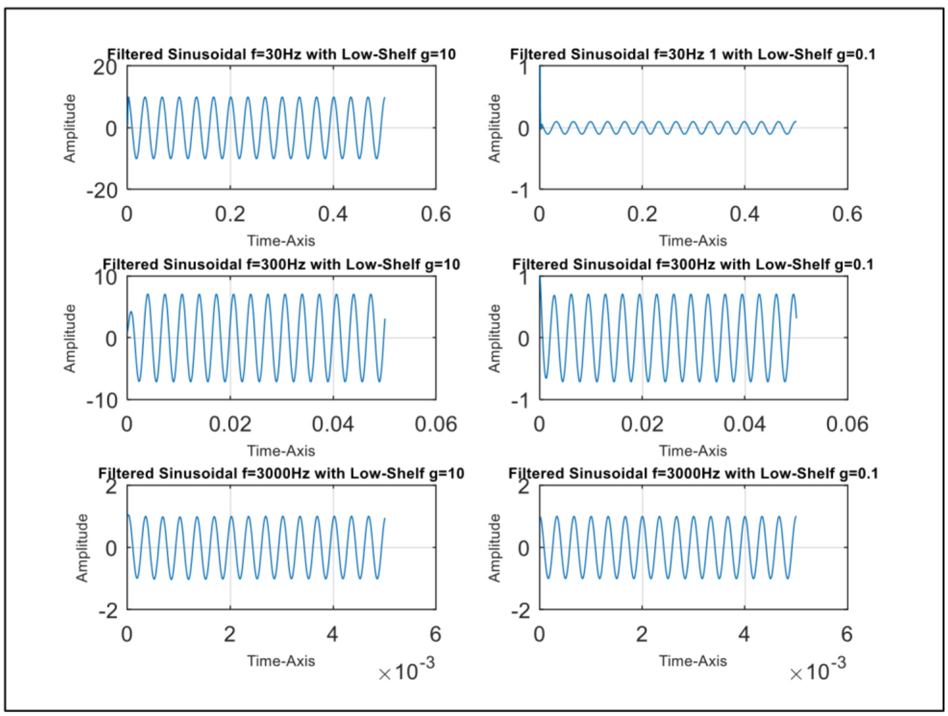


As we can see from the above plot, the point for our cut off frequency is 3dB below the maximum point on the graph, which is what we expect.

Everything to the left of $2\pi(300Hz)$ for the top plot is amplified with a gain of 10 then decays to zero. For the bottom plot, everything to the left of our designated cut off frequency is attenuated by 0.1, until it decays to zero. This signifies that for anything below our cut off frequency, we're modifying those signals with a gain value, while everything to the right of the cut off frequency is being left alone.

(d) Creating 3 sinusoids and filtering them with the low-shelf filter



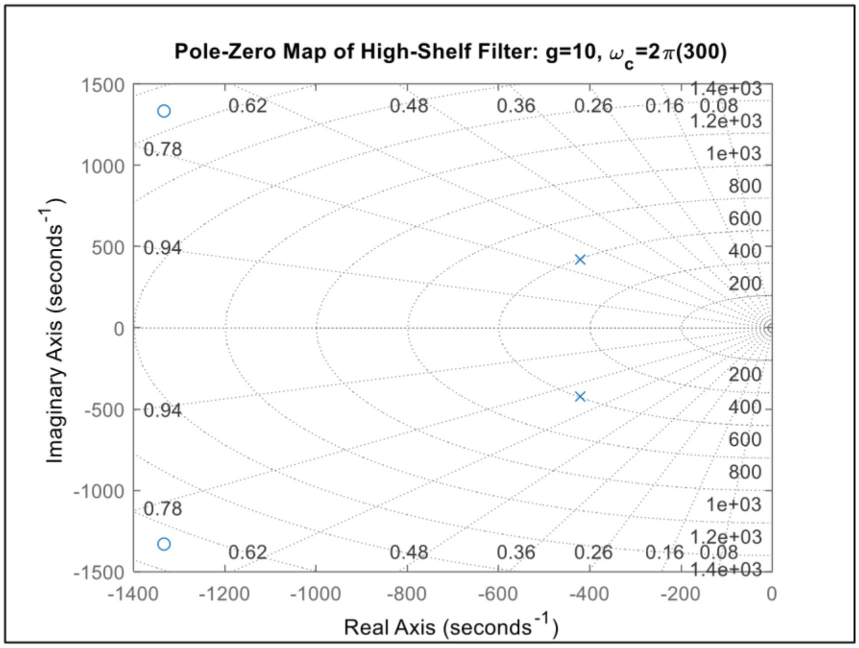


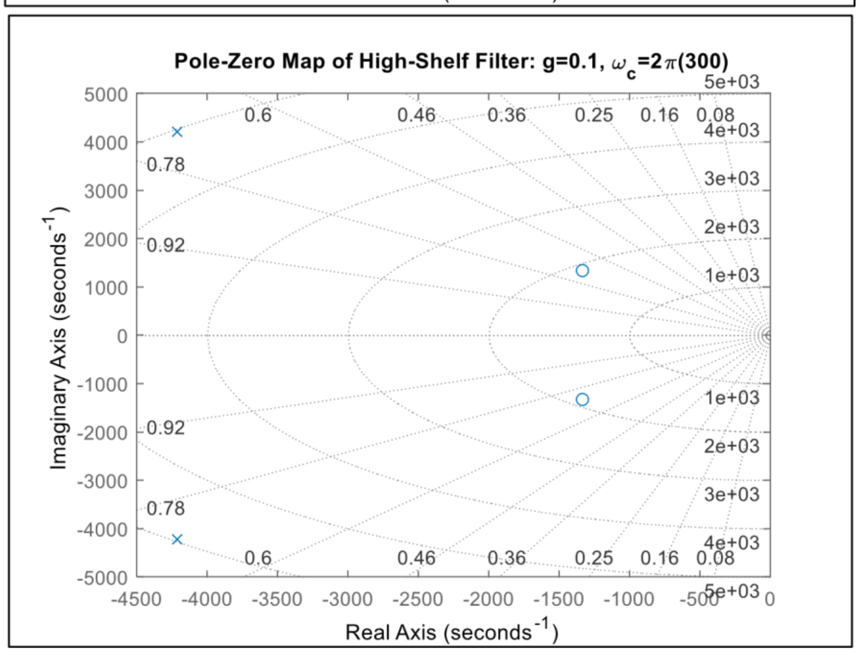
As we can see from the above plot, the sinusoid of 30Hz is impacted by the low shelf filter because it's frequency is below the cut off frequency of the low pass filter. The sinusoidal wave that has a frequency of 300Hz is not as impacted as the previous sinusoid above it due to the fact that it's exactly 300Hz, while the sinusoid that's 3000Hz is not impacted at all since it's above the cut off frequency of the low shelf filter.

3. Design of a High-Shelf Filter

(a) *Refer to MATLAB reference page in lab report to see High-Shelf filter details*

(b)pz-map of high shelf filter with gains 10 & 0.1





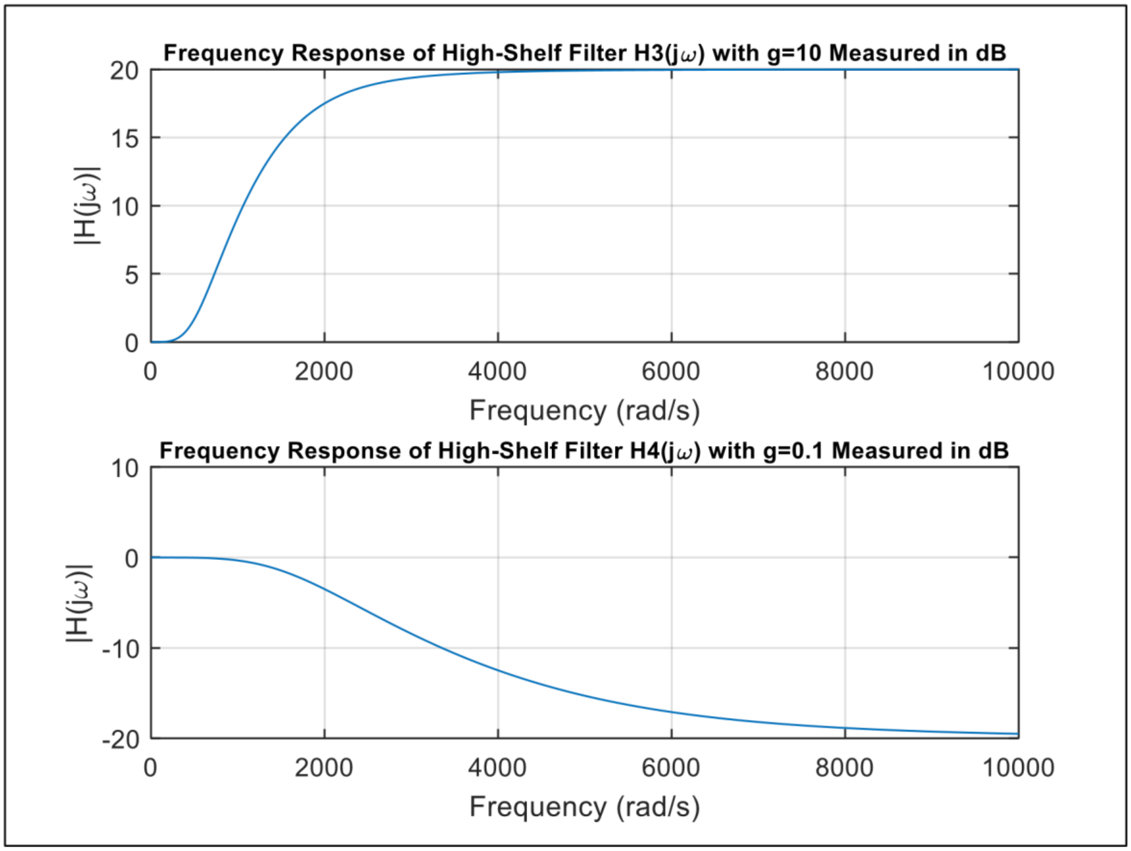
Where the transfer function of the high-shelf filter is

$$H_{hs}(s) = \frac{g(s + \frac{\omega_c}{\sqrt{g}} e^{-j\pi/4})(s + \frac{\omega_c}{\sqrt{g}} e^{\pi/4})}{(s + \omega_c e^{j\pi/4})(s + \omega_c e^{-j\pi/4})}$$

As explained earlier in the low shelf filter, the only thing that should change for pz-map of the high shelf transfer function is the relative distance of the zeros from the $j\omega$ axis because the zeros depend on the gain while the pole do not.

This page is intentionally left blank

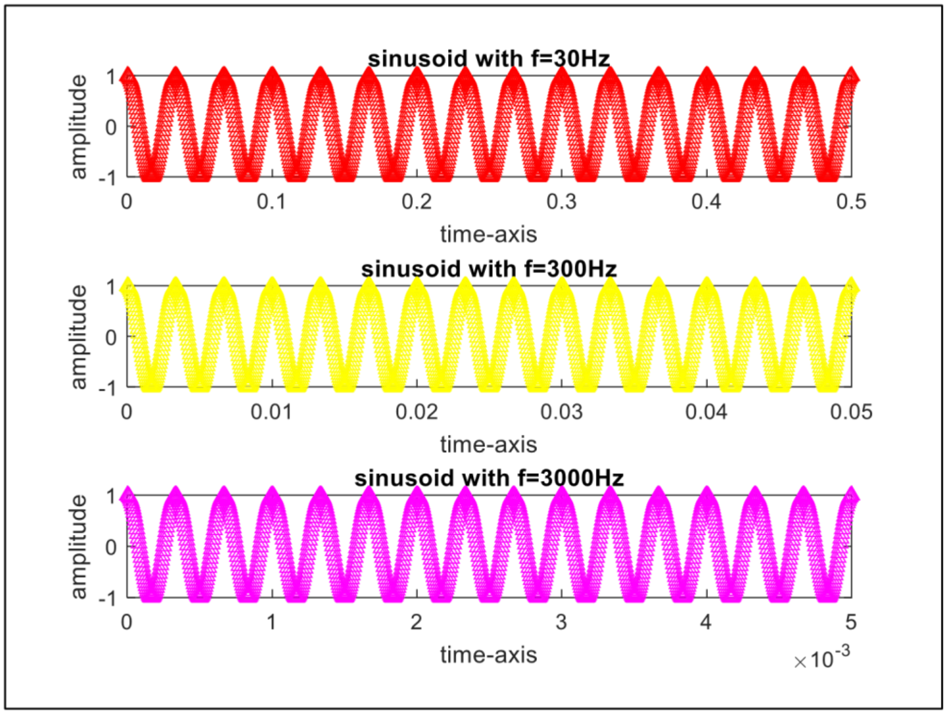
(c) Frequency response of High-Shelf filter

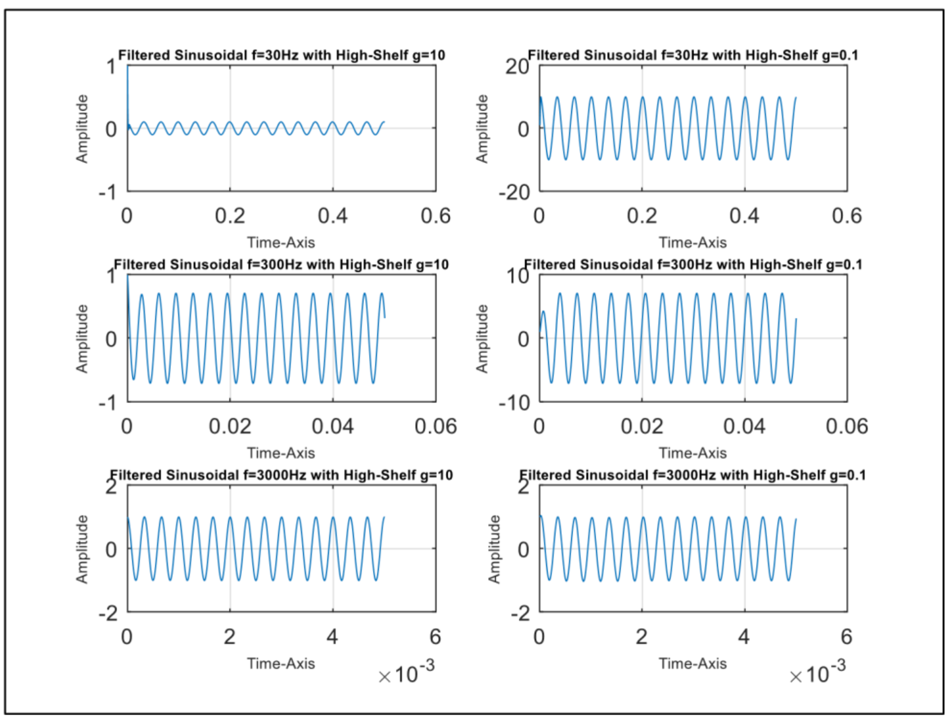


As we can see from the plot above, everything to the left of our cut off signal is being left alone, signified by the fact that our frequency response starts off at zero. Everything to the right of our cut off frequency is reaching a value of 20dB. This signifies a positive gain.

In the frequency response below, we can see that everything to the left of our cut off frequency is being left alone designated by the fact that our frequency response starts off at 0. Then, as we hit our cut off frequency, we can see that the frequency response

(d) Creating 3 sinusoids and Filtering them with the High-Shelf Filter



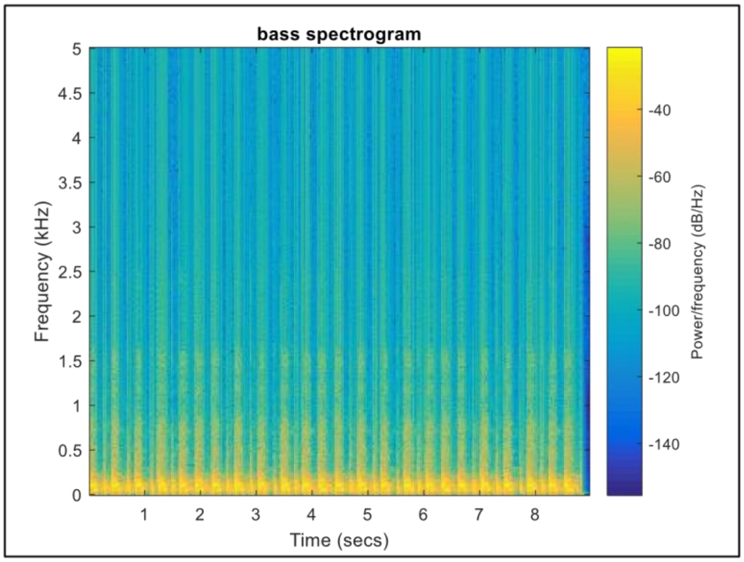


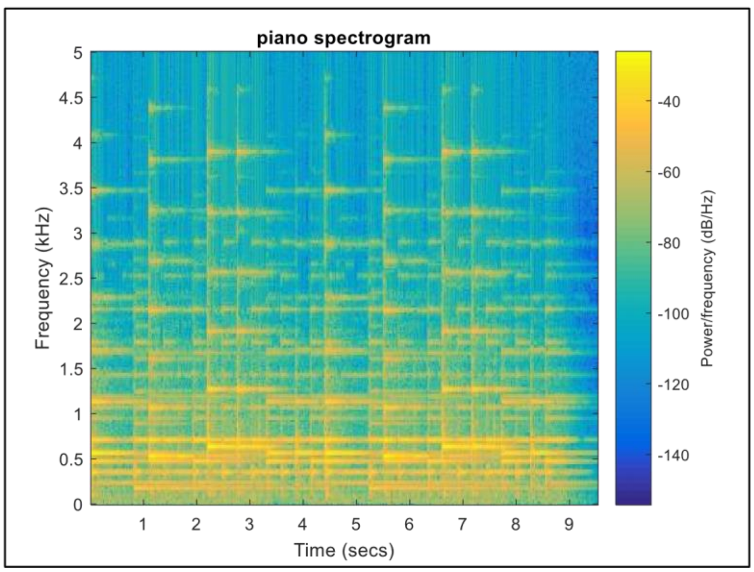
As explained in the low-shelf filter, the same idea applies here.

4. Music Equalization – Bass/Treble Boost and Cut

(a) *This part of the lab was simply calling the bass, piano, and mixture2 signals and thus requires analysis*

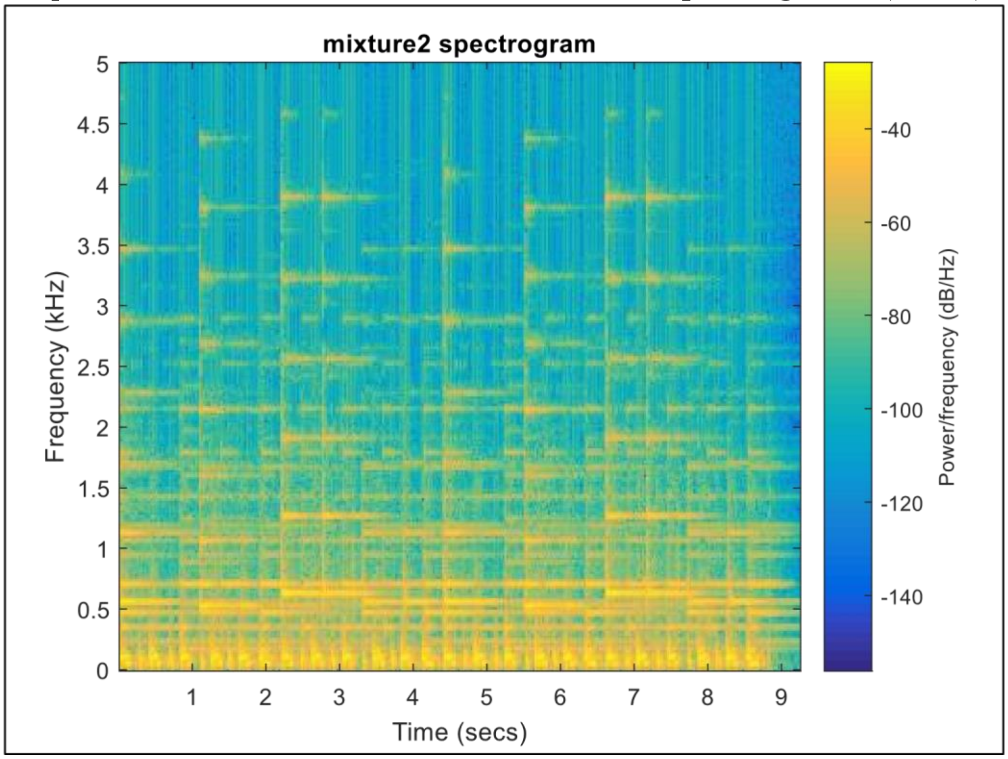
(b)Spectrogram plots of the following sound signals; bass, piano, and mixture2 (used to discussion of the signals)





From the above spectrograms, we can see the bass exists for lower frequencies at a power/frequency scale ranging from -140 to -40 dB/Hz, just like the piano does.

4. Music Equalization – Bass/Treble Boost and Cut: *Spectrograms (Cont.)*

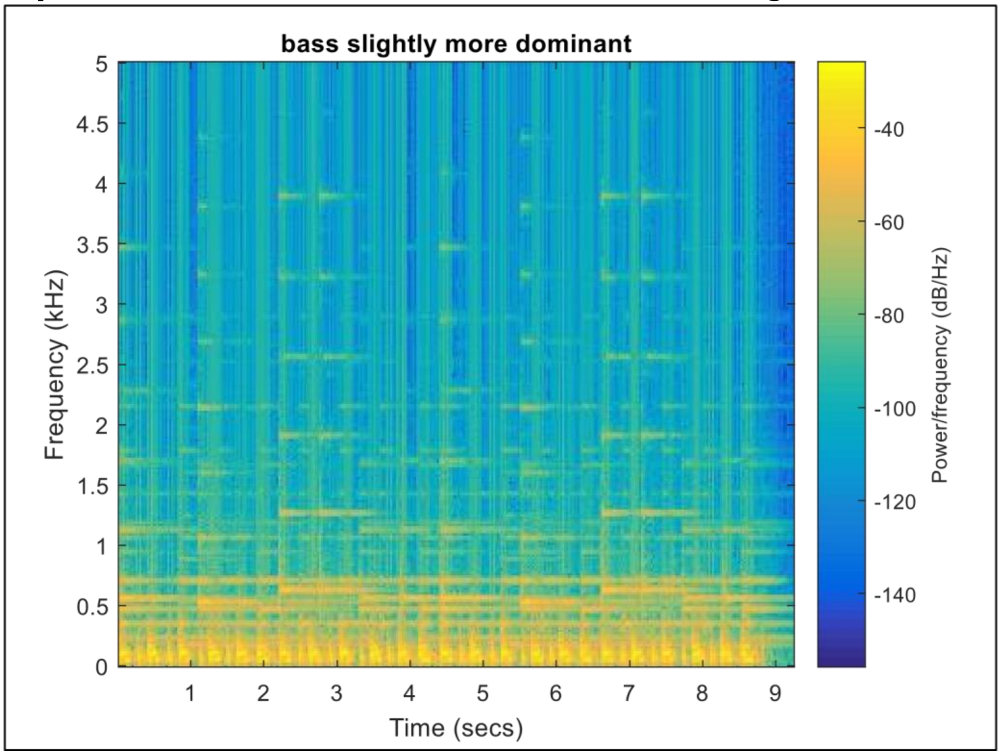


Listening to the sound signals by themselves; the bass is very deep and has high power evident from its spectrogram. From the spectrograms, we can see that when mixed, the colorbar scale is still ranges from -140 to -40 dB/Hz like individual piano and bass spectrograms. The piano has a wide diversity to frequencies again apparent from its spectrogram above.
From a spectrogram view, the sound looks diverse. Listening to the audio, it's far more interesting due to the diversity in tones.

Listening to both the piano and the bass, however, it's somewhat unbalanced. The piano overpowers the bass, despite the bass having a yellower color on the colorbar around -50dB. Despite this, the piano still *sounds* overpowering compared to the base (at least from my perspective and taste in music).

If I were to filter the mixture2, I would apply a low shelf filter of around 500Hz. I would do this and use a gain of roughly 10 to boost the power/intensity of the low frequency bass signal so it's more balanced with the high frequency piano keys. Personally, for my taste in music, I prefer deeper tones that are more powerful than other instruments. I may even use a high shelf filter to attenuate the piano keys, but that depends on how the amplified bass sounds afterwards. I may even apply multiple filters, depending on how the filtering process goes.

Music Equalization – Bass/Treble Boost and Cut: *Filtering mixture2 Weak Piano*

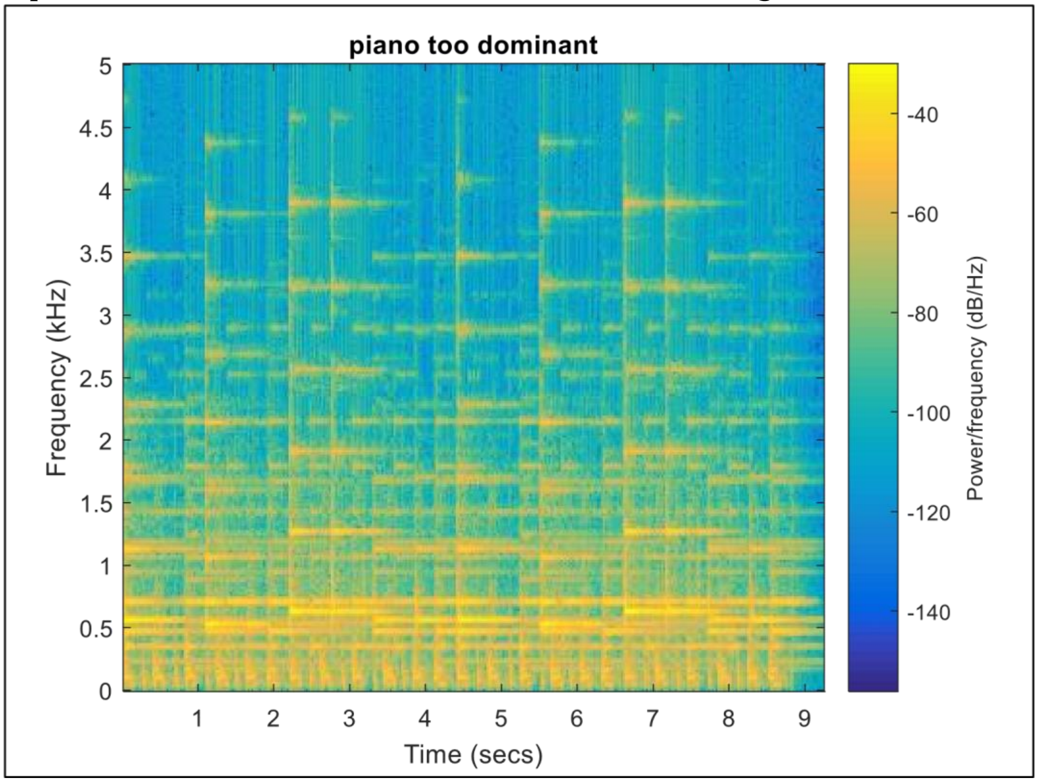


From my MATLAB code, I used the *lsim* command to simulate *sys_4* with *mixture2* and called it *example_1*. *sys_4* is a high-shelf filter with a cut off frequency of 300Hz with a gain of 0.1. The following result is this *figure(13)* spectrogram. The result is all frequencies above 300Hz are attenuated with a gain of 0.1.

From my analysis of the unfiltered *mixture2* spectrogram, I stated the sound file sounded as if the piano overpowered the bass and stated I would prefer a heavier bass. Applying a highshelf filter that attenuates the higher frequencies of the piano accomplishes this feat. By attenuating the higher frequencies of the piano (thus lowering their intensity and power evident from the above spectrogram as the higher frequency piano keys are faded out), the sound more "at balance" with the bass. I say "at balance" in quotes because for my taste in music I prefer a heavier bass.

In conclusion, *example_1* is a sound file with weaker higher frequency intensity/power compared *mixture2*. This results in a more "balanced" tone with the bass that meets the objective parameters of this lab.

Music Equalization – Bass/Treble Boost and Cut: *Filtering mixture2 Piano*



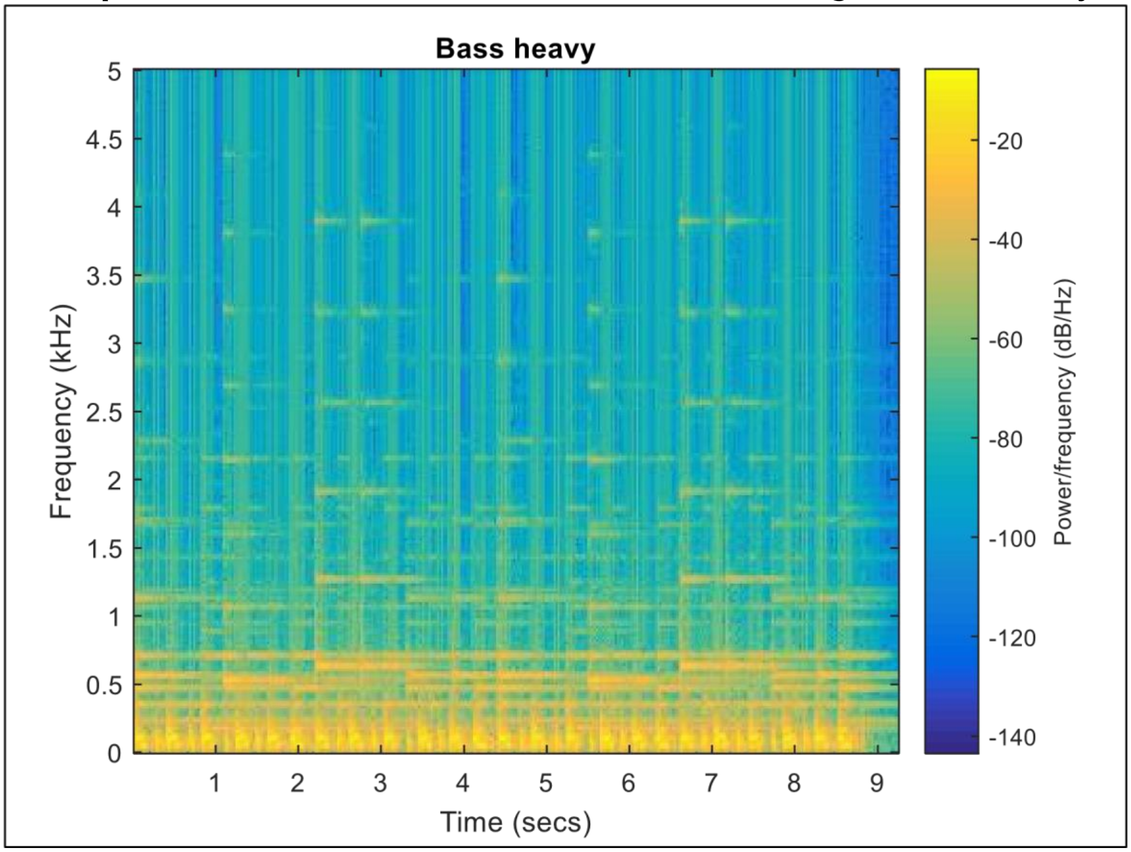
From my MATLAB code, I used the *lsim* command to simulate *sys_3* with *mixture2* and called it *example_2*. *sys_3* is a high shelf filter with a cut off frequency of 300Hz with a gain of 10. The following result is this *figure(14)* spectrogram. The result is all frequencies above 300Hz are amplified with a gain of 10.

When compared to *figure(13) example_1's* spectrogram, is clear that the spectrogram for *example_2* has higher intensity/power for frequencies above 300Hz. The overall yellowness is much more bright and intense. From the colorbar on the side of the spectrogram, higher yellow frequencies correspond to a power/frequency of -40dB/Hz or higher. The reason for the higher dB/Hz is because we're amplifying *mixture2* with a gain of 10. So, the above results make intuitive sense.

Listening to this signal, however, it's absolutely horrible. If I though the piano was overpowering before, it's borderline painful now. On full volume with headphones on, the amplified piano frequencies hurt my ear drums listening to the signal.

In conclusion, *figure(14) example_2's* audio is very off putting and unattractive to listen to. There's a lot to be learned from its corresponding spectrogram. However, from a music perspective and the objective parameters of this lab, *example_2's* filtering was a failure.

Music Equalization – Bass/Treble Boost and Cut: *Filtering mixture2 Heavy Bass*



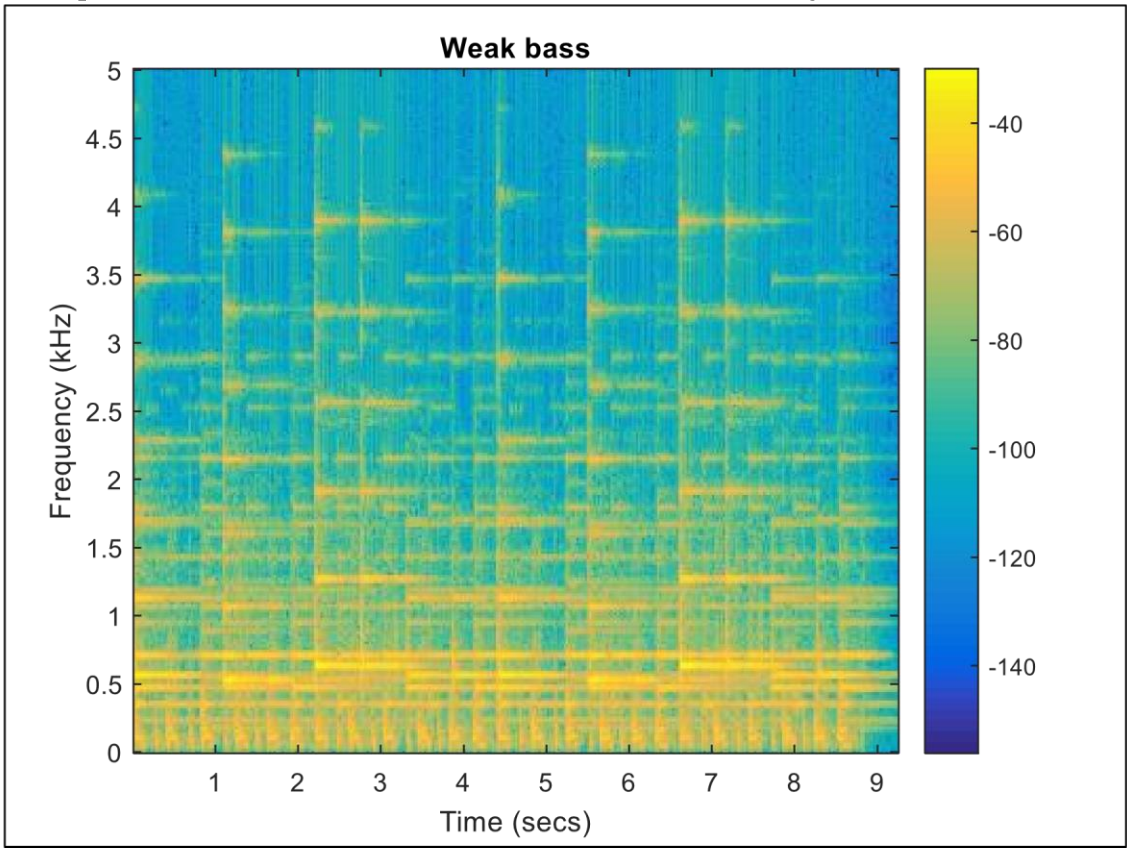
From my MATLAB code, I used the *lsim* command to simulate *sys_1* with *mixture2* and called it *example_3*. *sys_1* is a low-shelf filter with a cut-off frequency of 300Hz with a gain of 10. The following result is this *figure(15)* spectrogram. The result is that frequencies below 300Hz are amplified with a gain of 10.

When viewing the spectrogram, we can see that the lower frequency bass has a high power/intensity of 20dB/Hz compared to the faded yellows of the higher frequency piano keys. This suggests that the bass was amplified and the resulting spectrogram makes the higher frequency piano keys looked more faded out compared to the newly amplified bass frequencies. This makes *figure(15) example_3's* spectrogram look like a filtered high-shelf filter attenuating the higher frequency piano keys.

Listening to the audio file, it's probably the most pleasant out of all the filtered *mixture2* sound signals. The heavier bass signals is very satisfying to listen to compared to the high frequency piano keys. While someone else may think that the heavier bass overpowers the piano keys, I personally love it.

In conclusion, *figure(15) example_3's* filter results in an audio file with a much heavier bass when compared to *mixture2's* audio. The spectrogram looks as if the higher frequency piano keys are faded out – most likely due to the fact the amplified bass frequencies have an increased power/intensity that make the higher frequencies *appear* to have less power compared to the original *mixture2* spectrogram. For the objective parameters of this lab, this filtering, in my opinion, was a success.

Music Equalization – Bass/Treble Boost and Cut: *Filtering mixture2 – Weak Bass*

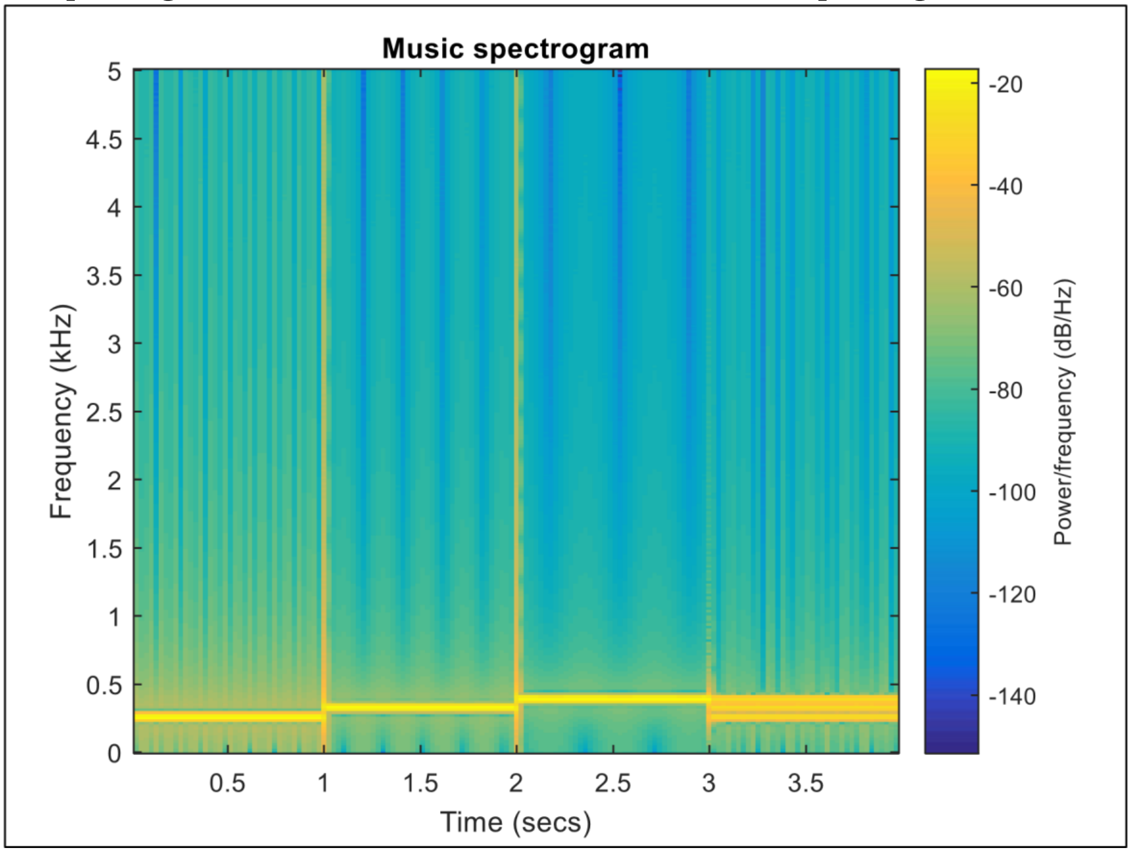


From the MATLAB code, I used *lsim* command to simulate *sys_2* with *mixture2* and called it *example_4. sys_2* is a low-shelf filter with a gain of 0.1 and a cut off frequency of 300Hz. The following result is this *figure(16)* spectrogram. The result is that all frequencies below 300Hz is attenuated with a gain of 0.1

When viewing the spectrogram, we can see that the lower frequency bass signals are attenuated when compared to the original *mixture2* spectrogram. The higher intensity piano keys are left alone from the original *mixture2. figure(16)'s* spectrogram strongly resembles the *piano* spectrogram with the attenuated bass frequencies.

When listening to this audio signal, it has a very weak, but still noticeable, bass (despite the attenuated bass frequencies and the strong resemblance to *figure(16)'s* spectrogram to the *piano* spectrogram). Personally for my taste of music, I don't like weak bass signals. However, it's interesting to listen to a weakened bass audio signal.

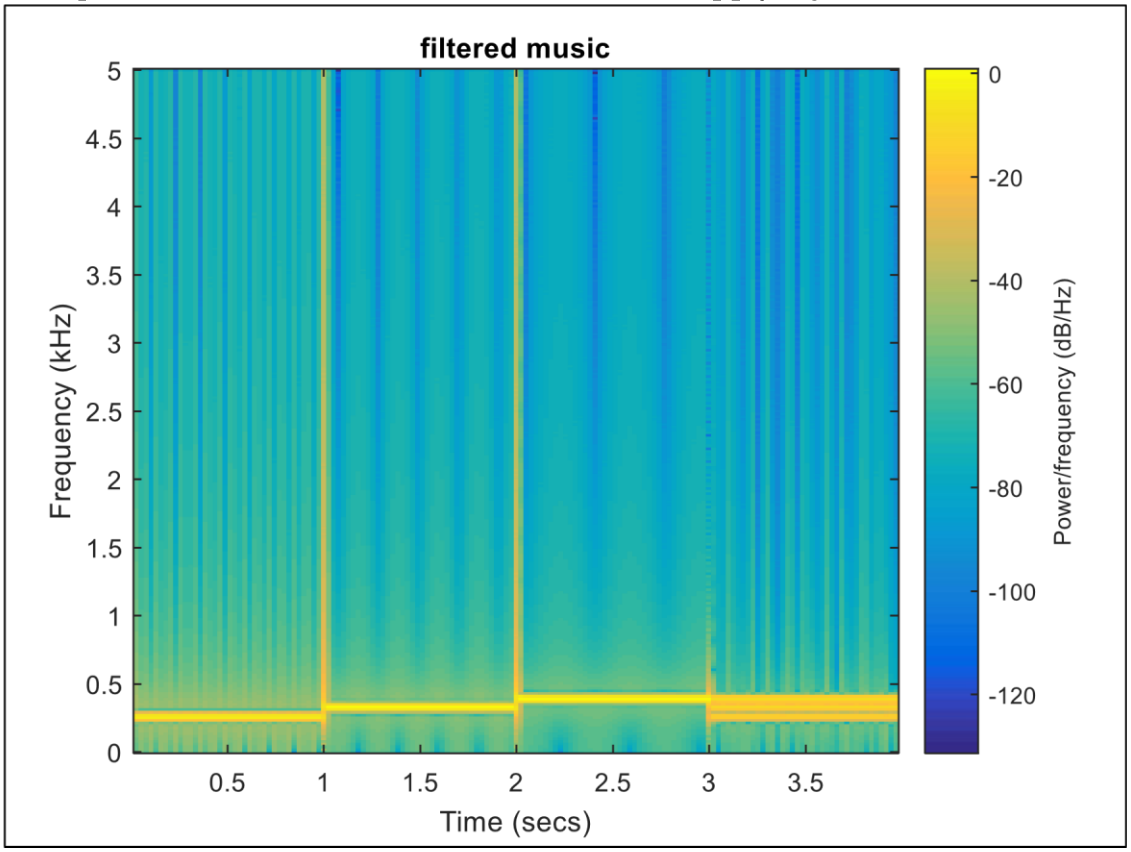
Music Spectrogram – Bass/Treble Boost and Cut: Music Spectrogram



From the above plot, the following spectrogram is a music tone composed of 3 tones, then a fourth term composed of all three tones played together simultaneously. The objective parameter of this part of the lab is to attenuate the first two tones and fourth terms while amplifying the third tone.

Based on the objective parameter of this portion of the lab, we will need to use multiple filters to accomplish this feat by; applying a shelf filter to boost the third tone, and then another shelf filter to attenuate the first, second, and fourth tones. This will accomplish the requirements for this portion of the lab.

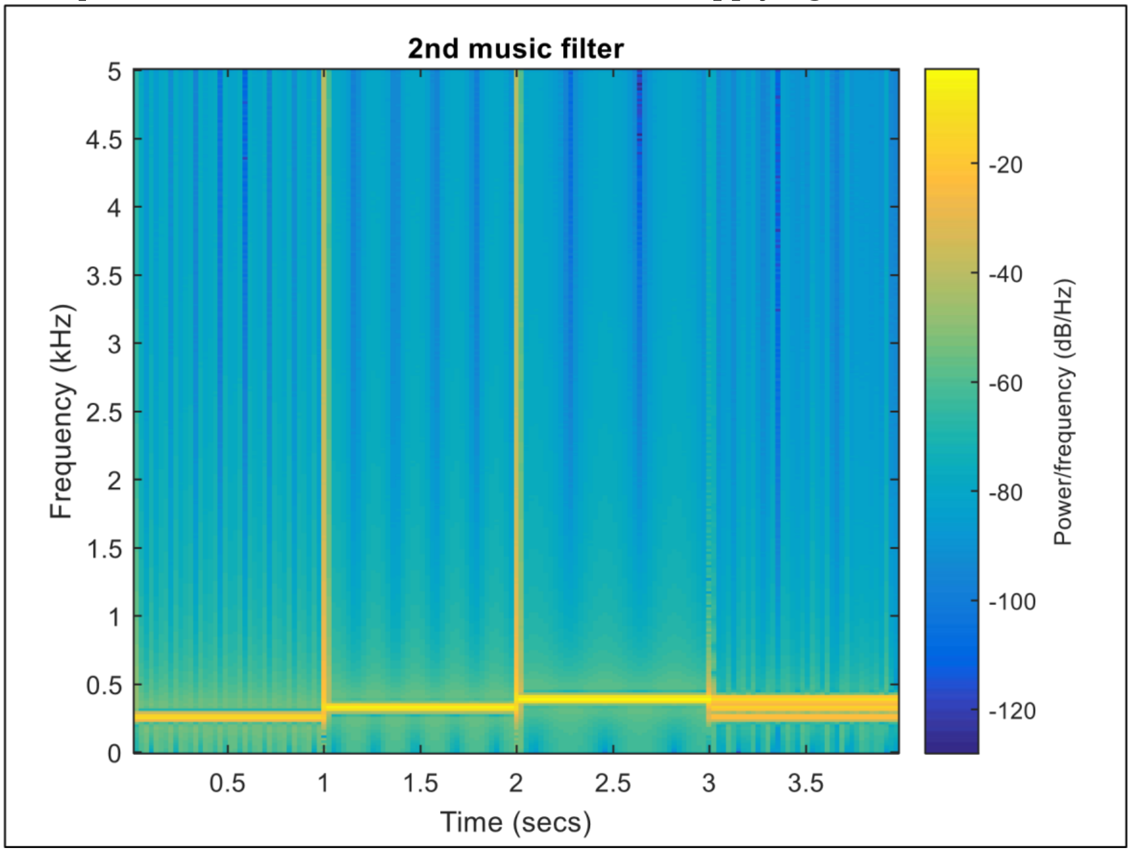
Music Equalization – Bass/Treble Boost and Cut: *Applying the 1st Filter on Music*



From my MATLAB code, I used a high shelf filter to enhance the third tone. I used a gain factor of 10 and a cut off frequency of 400Hz and called it *y_music*. Thus, the third tone is amplified with a gain of 10 while all other frequencies are left alone. This is the first of two filters to enhance the third tone while attenuating the other three tones.
Strangely, the only thing that was affected during this *y_music* was the background and the colorbar on the right of the spectrogram. I was expecting the color intensity of the tones to change colors like when I filtered the *mixture2* spectrogram. However, this was not the case.

Listening to the sound signal, I was even more confused. There wasn't a significant change to the audio of the signal compared to the original. However, I know something must be changing since the colorbar for the filtered spectrogram changed and the background for the filtered spectrogram also changed (despite the audio file of the filtered *Music* tone suggesting otherwise).

Music Equalization – Bass/Treble Boost and Cut: *Applying the 2nd Filter*



From my MATLAB code, I used a low shelf filter on the previously filtered *music* signal *y_music* with a cut off frequency of 420Hz with a gain of 0.1 to attenuate the first, second, and fourth tones and called it *y_music_2*. This is the second filter applied to the original spectrogram music signal.

Just like before, the only thing that changes during this filter is the background of the spectrogram, and not the actual intensity of the tones themselves. Applying a second filter did somewhat distort the sound; keyword being "distort." The purpose of this portion of the lab was to enhance the third tone while attenuating the other tones. This objective wasn't met like I had originally intended and I inadvertently attenuated all 4 of the tones. However, based on the corresponding spectrograms, along with listening to the filtered sound files, there is a change happening.

In conclusion, applying multiple filters to the *Music* sound file did not result in the filtered sound file I had originally hypothesized. The filters are not having a significant impact on the power/intensity of the four tones. However, the spectrograms are yielding subtle changes in their backgrounds with each additional filter applied to the *Music* file. So, the objective parameters for this portion of the lab are, to an extent, met; just not as I had originally intended to meet them.

MATLAB Appendices/References

```
%-------------------------------------------------------------------------
%2: DESIGN OF A SECOND ORDER LOW-SHELF FILTER
%-------------------------------------------------------------------------
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%PART A START ///////////////////////////////////////////////////////////
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%The purpose of this part of the lab was to design the low-shelf filter. To
%see the specifics of this portion of the lab, refer to the design_shelf
%function associated with this lab.
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%PART A END /////////////////////////////////////////////////////////////
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%PART B START ///////////////////////////////////////////////////////////
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%gain = 10, omega_c = 2*pi*(300);
omega_c = 2*pi*300;  gain_1 = 10;

[B1,A1] = design_shelf(gain_1,omega_c,0);
figure(1);  pzmap(A1,B1);
title('Pole-Zero Map of Low-Shelf Filter: g=10, \omega_c=2\pi(300)');
grid on;

%gain = 1, omega_c = 2*pi*(300);
omega_c = omega_c;  gain_2 =
0.1;

[B2,A2] = design_shelf(gain_2,omega_c,0);
figure(2);  pzmap(A2,B2);
title('Pole-Zero Map of Low-Shelf Filter: g=0.1, \omega_c=2\pi(300)');
grid on;

% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%PART B END ///////////////////////////////////////////////////////////////
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%PART C START /////////////////////////////////////////////////////////////
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

omega = linspace(0,10000,10000);
H1 = freqs(B1,A1,omega);
H2 = freqs(B2,A2,omega);
  figure(3)
subplot(2,1,1), plot(omega,20*log10(abs(H1)));
xlabel('Frequency (rad/s)'); ylabel('|H(j\omega)|');
title('Frequency Response of Low-Shelf Filter H1(j\omega) with g=10 Measured in dB',
'FontSize',9);
grid on;
subplot(2,1,2), plot(omega,20*log10(abs(H2)));
xlabel('Frequency (rad/s)'); ylabel('|H(j\omega)|');
title('Frequency Response of Low-Shelf Filter H2(j\omega) with g=0.1 Measured in dB',
'FontSize',9);  grid
on;

% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
%PART C END ///////////////////////////////////////////////////////////////
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%PART D START /////////////////////////////////////////////////////////////
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%Creating 3 sinusoids   t1 =
linspace(0,.50,1000);
sinusoid_1 = cos(2*pi*30*t1);
figure(4);
subplot(3,1,1), plot(t1, sinusoid_1, 'r^');
title('sinusoid with f=30Hz');
xlabel('time-axis');  ylabel('amplitude');

t2 = linspace(0,0.050, 1000);  sinusoid_2
= cos(2*pi*300*t2);  subplot(3,1,2),
plot(t2,sinusoid_2,'y^');
title('sinusoid with f=300Hz');
xlabel('time-axis');
ylabel('amplitude');

t3 = linspace(0,0.0050,1000);  sinusoid_3
= cos(2*pi*3000*t3);  subplot(3,1,3),
plot(t3,sinusoid_3, 'm^');
title('sinusoid with f=3000Hz');
xlabel('time-axis');  ylabel('amplitude');

%delcaring the figure to plot the system responses  figure(5);

%gain of 10, low-shelf filter sys_1
= tf(B1,A1);

%gain of 0.1, low-shelf filter   sys_2
= tf(B2,A2);

%Simulating multiple systems with different inputs and different shelf
%design parameters
%-------------------------------------------------------------------------
%Sinusoid 1 f=30Hz
%System reponse with a gain of 10, low-shelf filter
y_1 = lsim(sys_1,sinusoid_1,t1);
subplot(3,2,1),plot(t1,y_1); grid on;
title('Filtered Sinusoidal f=30Hz with Low-Shelf g=10','FontSize',7);
xlabel('Time-Axis', 'FontSize',7);  ylabel('Amplitude','FontSize',7);

%Sinusoid 1 f=30Hz
%System response with a gain of 0.1, low-shelf filter
y_2 = lsim(sys_2,sinusoid_1,t1);
subplot(3,2,2),plot(t1,y_2); grid on;
title('Filtered Sinusoidal f=30Hz 1 with Low-Shelf g=0.1','FontSize',7);
xlabel('Time-Axis','FontSize',7);
ylabel('Amplitude','FontSize',7);
%-------------------------------------------------------------------------
%Sinusoid 2 f=300Hz
%System response with a gain of 10, low-shelf filter
y_3 = lsim(sys_1,sinusoid_2,t2);
subplot(3,2,3),plot(t2,y_3); grid on;
title('Filtered Sinusoidal f=300Hz with Low-Shelf g=10','FontSize',7);
xlabel('Time-Axis', 'FontSize',7);  ylabel('Amplitude','FontSize',7);
```

```
%Sinusoid 2 f=300Hz
%System response with a gain of 0.1, low-shelf filter
y_4 = lsim(sys_2,sinusoid_2,t2);
subplot(3,2,4),plot(t2,y_4); grid on;
title('Filtered Sinusoidal f=300Hz with Low-Shelf g=0.1','FontSize',7);
xlabel('Time-Axis', 'FontSize',7);  ylabel('Amplitude','FontSize',7);

%-----------------------------------------------------------------------
%Sinusoid 3 f = 300Hz
%System response with a gain of g=10, low-shelf filter
y_5 = lsim(sys_1,sinusoid_3,t3);
subplot(3,2,5),plot(t3,y_5); grid on;
title('Filtered Sinusoidal f=3000Hz with Low-Shelf g=10','FontSize',7);
xlabel('Time-Axis', 'FontSize',7);  ylabel('Amplitude','FontSize',7);

%Sinusoid 3 f=300Hz
%System response with a gain of g=0.1, low-shelf filter
y_6 = lsim(sys_2,sinusoid_3,t3);
subplot(3,2,6),plot(t3,y_6); grid on;
title('Filtered Sinusoidal f=3000Hz with Low-Shelf g=0.1','FontSize',7);
xlabel('Time-Axis', 'FontSize',7);  ylabel('Amplitude','FontSize',7);
%-----------------------------------------------------------------------
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%PART D END ////////////////////////////////////////////////////////////
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%---------------------------------------------------------------------
%3: DESIGN OF A SECOND ORDER-SHELF FILTER
%---------------------------------------------------------------------
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%PART A START //////////////////////////////////////////////////////////
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%This part of the lab was to design the high-shelf filter inside of the
%design shelf function; refer to that .m file to see the code/and comments
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%PART A END ////////////////////////////////////////////////////////////
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%PART B START //////////////////////////////////////////////////////////
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
omega_c = 2*pi*300;  gain_1 = 10;  gain_2 = 0.1;

[B3,A3] = design_shelf(gain_1,omega_c,1);
[B4,A4] = design_shelf(gain_2,omega_c,1);
  figure(6);
pzmap(A3,B3);
title('Pole-Zero Map of High-Shelf Filter: g=10, \omega_c=2\pi(300)');
grid on;    figure(7);  pzmap(A4,B4);
title('Pole-Zero Map of High-Shelf Filter: g=0.1, \omega_c=2\pi(300)');
grid on;

% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%PART B END //////////////////////////////////////////////////////////// %
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%PART C START ////////////////////////////////////////////////////////// %
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
omega = linspace(0,10000,10000);
```

```
H3 = freqs(B3,A3,omega);
H4 = freqs(B4,A4,omega);
  figure(8);
subplot(2,1,1), plot(omega,20*log10(abs(H3)));
xlabel('Frequency (rad/s)'); ylabel('|H(j\omega)|');
title('Frequency Response of High-Shelf Filter H3(j\omega) with g=10 Measured in dB',
'FontSize',9);
grid on;
subplot(2,1,2), plot(omega,20*log10(abs(H4)));
xlabel('Frequency (rad/s)'); ylabel('|H(j\omega)|');
title('Frequency Response of High-Shelf Filter H4(j\omega) with g=0.1 Measured in
dB','FontSize',9);  grid on;
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%PART C END //////////////////////////////////////////////////////////// %
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%PART D START //////////////////////////////////////////////////////////
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%Systems built for high-self filter
sys_3 = tf(B3,A3);  sys_4 =
tf(B4,A4);
  figure(9);

%Simulating multiple systems with different inputs and different shelf
%design parameters
%-----------------------------------------------------------------------
%Sinusoid 1
%System response for high-shelf filter with gain g=10
y_7 = lsim(sys_3,sinusoid_1,t1);
subplot(3,2,1),plot(t1,y_7);  grid on;
title('Filtered Sinusoidal f=30Hz with High-Shelf g=10','FontSize',6.5);
xlabel('Time-Axis', 'FontSize',7);  ylabel('Amplitude','FontSize',7);

%Sinusoid 1
%System response for high-shelf filter with gain g=0.1
y_8 = lsim(sys_4,sinusoid_1,t1);
subplot(3,2,2),plot(t1,y_8);  grid on;
title('Filtered Sinusoidal f=30Hz with High-Shelf g=0.1','FontSize',6.5);
xlabel('Time-Axis', 'FontSize',7);  ylabel('Amplitude','FontSize',7);
%-----------------------------------------------------------------------
%Sinusoid 2
%System response for high-shelf filter with gain g=10;
y_9 = lsim(sys_3,sinusoid_2,t2);
subplot(3,2,3),plot(t2,y_9);  grid on;
title('Filtered Sinusoidal f=300Hz with High-Shelf g=10','FontSize',6.5);
xlabel('Time-Axis', 'FontSize',7);  ylabel('Amplitude','FontSize',7);

%Sinusoid 2
%System response for high-shelf filter with gain g=0.1;
y_10 = lsim(sys_4, sinusoid_2,t2);
subplot(3,2,4),plot(t2,y_10);  grid on;
title('Filtered Sinusoidal f=300Hz with High-Shelf g=0.1','FontSize',6.5);
xlabel('Time-Axis', 'FontSize',7);  ylabel('Amplitude','FontSize',7);
%-----------------------------------------------------------------------
%Sinusoid 3
%System response for high-shelf filter with gain g=10;
y_11 = lsim(sys_3,sinusoid_3,t3);
subplot(3,2,5),plot(t3,y_11);  grid on;
```

```
title('Filtered Sinusoidal f=3000Hz with High-Shelf g=10','FontSize',6.5);
xlabel('Time-Axis', 'FontSize',7);  ylabel('Amplitude','FontSize',7);

%Sinusoid 3
%System response for high-shelf filter with gain g=0.1;
y_12 = lsim(sys_4,sinusoid_3,t3);
subplot(3,2,6),plot(t3,y_12);  grid on;
title('Filtered Sinusoidal f=3000Hz with High-Shelf g=0.1','FontSize',6.5);
xlabel('Time-Axis', 'FontSize',7);  ylabel('Amplitude','FontSize',7);
%-------------------------------------------------------------------------
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%PART D END ////////////////////////////////////////////////////////////
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%-------------------------------------------------------------------------
%4: MUSIC EQUALIZATION - BASS/TREBLE BOOST AND CUT
%-------------------------------------------------------------------------
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%PART A START///////////////////////////////////////////////////////////
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
load('lab6_2016.mat');  soundsc(bass,fs);  soundsc(piano,fs);
soundsc(mixture2,fs);
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%PART A END //////////////////////////////////////////////////////////// %
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%PART B START/////////////////////////////////////////////////////////// %
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -  %{
Discussion question: compare the sound of the bass with the piano and the
bass and how they sound mixed together.
%}
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%PART B END //////////////////////////////////////////////////////////// %
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%PART C START///////////////////////////////////////////////////////////
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
win = 512; noverlap = 256; nfft = 512; figure(10);
spectrogram(bass,win,noverlap,nfft,fs,'yaxis');
title('bass spectrogram');
%colorbar;
figure(11);
spectrogram(piano,win,noverlap,nfft,fs,'yaxis');
title('piano spectrogram');
  figure(12);
spectrogram(mixture2,win,noverlap,nfft,fs,'yaxis');
title('mixture2 spectrogram');

% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%PART C END //////////////////////////////////////////////////////////// %
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%PART D START/////////////////////////////////////////////////////////// %
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

time = 1:length(mixture2); time=
time*(1/fs);
```

```
%This seems to be just right, more bass dominant
example_1 = lsim(sys_4,mixture2,time);
soundsc(example_1,fs);  figure(13);
spectrogram(example_1,win,noverlap,nfft,fs,'yaxis');
title('bass slightly more dominant');

%{
    for the code above, we used a high shelf filter with a frequency of
    300Hz; so anything above 300Hz was attenuated by a gain factor of 0.10
%}

%For this, the piano is far too dominant
example_2 = lsim(sys_3,mixture2,time);
soundsc(example_2,fs); figure(14);
spectrogram(example_2,win,noverlap,nfft,fs,'yaxis');
title('piano too dominant');

%{
    for the code above; we applied a high-shelf filter with a frequency of     300Hz;
so this time, the gain was 10 and amplified making the piano     very loud with intense
power.
%}

%Bass heavy
example_3 = lsim(sys_1,mixture2,time);
soundsc(example_3,fs); figure(15);
spectrogram(example_3,win,noverlap,nfft,fs,'yaxis');
title('Bass heavy');
  %weak
bass
example_4 = lsim(sys_2,mixture2,time);
soundsc(example_4,fs); figure(16);
spectrogram(example_4,win,noverlap,nfft,fs,'yaxis');
title('Weak bass');
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%PART D END ////////////////////////////////////////////////////////////
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%PART E START/////////////////////////////////////////////////////////// %
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%PART E END //////////////////////////////////////////////////////////// %
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%PART F START///////////////////////////////////////////////////////////
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%soundsc(music,fs);  figure(17);
spectrogram(music,win,noverlap,nfft,fs,'yaxis');
title('Music spectrogram');
%colormap bone

%-------------------------------------------------------------------------
%First filter, a high pass filter to enhance the third tone
time_music = 1:length(music); time_music = time_music
*(1/fs);

omega_music = 2*pi*(400);   gain_music
= 10;
```

```
[B_music,A_music] = design_shelf(gain_music,omega_music,1); sys_music
= tf(B_music,A_music);

y_music = lsim(sys_music,music,time_music);
%soundsc(y_music,fs);  figure(18);
spectrogram(y_music,win,noverlap,nfft,fs,'yaxis');
%colormap bone;  title('filtered
music');
%------------------------------------------------------------------------
%Second filter, a low pass filter to attenuate the other tones
omega_music_2 = 2*pi*(420);  gain_music_2 = 0.1;

[B_music_2,A_music_2] = design_shelf(gain_music_2,omega_music_2,0);
sys_music_2 = tf(B_music_2,A_music_2);
y_music_2 = lsim(sys_music_2,y_music,time_music);
%soundsc(y_music_2); figure(19);
spectrogram(y_music_2,win,noverlap,nfft,fs,'yaxis');
title('2nd music filter');
%------------------------------------------------------------------------
% %Third filter, another low pass filter
% omega_3 = 2*pi*(400);
% gain_3 = 10;
%
% [B_3,A_3] = design_shelf(gain_3,omega_3,1);
% sys_music_3 = tf(B_3,A_3);
%
% y_music_3 = lsim(sys_music_3,sys_music_2,time_music);
% soundsc(y_music_3);
% figure(20);
% spectrogram(y_music_3,win,noverlap,nfft,fs,'yaxis');
% title('3rd music filter');

% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%PART F END ///////////////////////////////////////////////////////////// %
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```