AMATH 482 Homework 2

Tsz Wai Tsui

February 10, 2021

Abstract

Using the Gabor Transform, which is a windowed Fourier Transform, we aim to reproduce the music score for the guitar in the 14-second audio clip of the song *Sweet Child O' Mine* by Guns N' Roses and for the base and guitar in the 1-minute audio clip of the song *Comfortably Numb* by Pink Floyd. By performing a time-frequency analysis on the two audio clips, we are able to create spectrograms for visualizing the present frequencies during the music so that we can identify the corresponding notes. We also use the Shannon function to filter a certain frequency of *Comfortably Numb* out.

1 Introduction and Overview

Given two audio music clips, we are asked to reproduce the score of them. One contains 14 seconds of the song *Sweet Child O' Mine* by Guns N' Roses and is played by electric guitar only. Another one contains 60 seconds of the song *Comfortably Numb* by Pink Floyd and is played by the base, guitar, and drum. For the shorter clip, we directly apply Gabor transform on it. For the longer one, since the data consists of three instruments' frequencies, we need to apply Shannon filter to isolate the base out so that we can distinguish the base's notes from the guitar's and then perform Gabor transform to reproduce corresponding music scores.

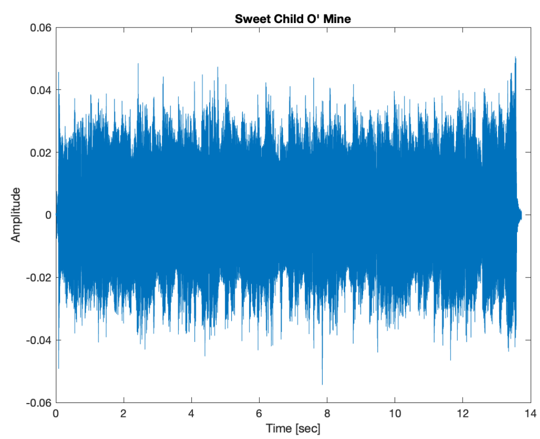


Figure 1: Visualization of the song, *Sweet Child O' Mine*

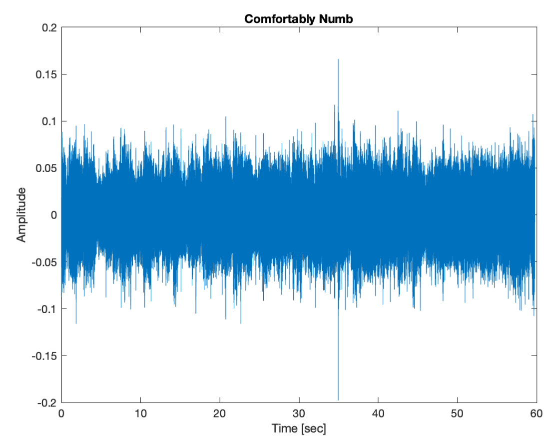


Figure 2: Visualization of the song, *Comfortably Numb*

2 Theoretical background

Fourier transform takes a function of space or time and converts it to a function to frequencies by the formula:

$$\hat{f}(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-ikx} dx. \tag{1}$$

The inverse of Fourier transform takes a function of frequencies and converts it back to a function of space of time by the formula:

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \hat{f}(x) e^{-ikx} dx. \tag{2}$$

Fourier transform is very useful when analyzing signals with a fixed frequency, but the drawback is that the shift invariance of the absolute value of the Fourier transform loses information about what is happening in the time domain. Particularly, we lost when certain frequencies occur or how the frequencies change over time. As the two audio clips we need to analyze are mixed of high and low frequencies, we need a different approach to get both time and frequency information from a signal.

The Gabor transform, also known as the short-time Fourier transform, can make up Fourier transform's shortcoming. It is defined as:

$$\tilde{f}_g(r,k) = \int_{-\infty}^{\infty} f(t) g(t-\tau) e^{-ikt} dt \tag{3}$$

$g(t)$ is a filter function, and the value of τ describes where the filter is centered. Therefore, for a fixed τ the function $\tilde{f}_g(r,k)$ gives you information about the frequency components near time τ. There are many choices for g, and some commonly used assumptions are:

1. The function g is real and symmetric.
2. $\|g\|_2 \coloneqq \left(\int_{-\infty}^{\infty} |g(t)|^2 dt\right)^{\frac{1}{2}} = 1$. That is, the L_2-norm of g is set to unity.

When Gabor developed this method, he used Gaussian as the filter function $g(t)$. Therefore, Gaussian is also our default choice. It is defined as:

$$g(t-\tau) = e^{-a(t-\tau)^2}. \tag{4}$$

The value of a describes the size of the window, so $a > 0$. If the window is too huge, we just recover the Fourier transform over the whole signal, which has all the frequency information, but no information about time. If the window is extremely small, then we are almost look at individual times along the signal, and so there would be no frequency information. Therefore, in order to perform time-frequency analysis, we need to strike a balance.

Beside from Gaussian, we use Shannon (rectangular) function, which is unity within a given interval and 0 otherwise, to filter a range of frequency around the center. Given an interval $[-x_1, x_1]$, its formula is:

$$S(t) = \begin{cases} 1, if |t - center| < x_1 \\ 0, otherwise \end{cases}. \tag{5}$$

3 Algorithm Implementation and development

For both of the audio clips, we perform the same first step. First, we load the audio file into MATLAB, convert it into a vector that represents the music, and calculate its record time in seconds. Note that we rescale the frequencies by $\frac{1}{L}$ because the scale used for the frequencies of the musical notes is Hertz. Then, we define the window size and the time vector with 0.1 second increment.

For *Sweet Child O' Mine*'s audio file. We then directly perform Gabor transform. In a for loop, we multiply Gaussian function which centers around each time step by the signal and take the `fft` of the filter signal. After shifting and taking the absolute value of the data, we plug it in `pcolor` to plot spectrograms outside the loop. Spectrogram is a visual representation of the frequencies of a signal against time, so it can show the changing of the Fourier transform as the window slides over the domain. Different values of the window size (a) are tested for the best resolution. At last, we plot lines of a few music notes' frequencies to identify what notes are played by the guitar.

For *Comfortably Numb*'s audio file, we trim the clip into 6 parts, 10 seconds each and apply Shannon filter before performing Gabor transform. First, we want to isolate the base from the audio. As shown in Figure 3, the frequencies of the notes played by the base lay between 0 Hz to 200Hz. We set the center of Shannon function as 0 and the width as 400, so frequencies of the signal in the interval $[-200, 200]$ can be filtered. Since we are filtering in the frequency domain, we need to `fft` the whole signal and use `ifft` to convert it back to the time domain. Then, we perform Gabor transform on it, plot spectrograms, and add lines of notes' frequencies, just like what we did for the *Sweet Child O' Mine*'s audio file.

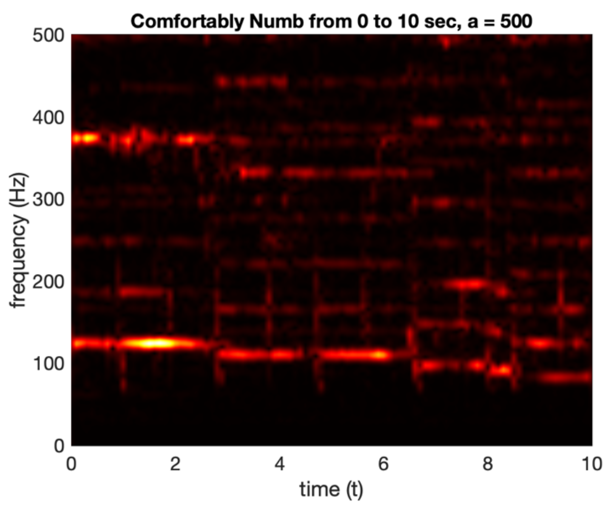


Figure 3: Spectrogram of Gabor transform on *Comfortably Numb* from 0-10 seconds

As we know the base's frequency range is below 200Hz, the guitar's frequency range will be above 200Hz. We increase the y-axis of the spectrogram and find out that the frequencies of the notes played by the guitar lay between 200Hz and 600Hz except for the second clip (10-20 seconds), which ranges from 150Hz to 600Hz. Therefore, the Shannon filter we use to filter guitar's frequency has 400 as center and 400 or 500 as the width. At last, we perform Gabor transform, plot spectrograms, and add lines of notes' frequencies.

4 Computational Results

4.1 *Sweet Child O' Mine*

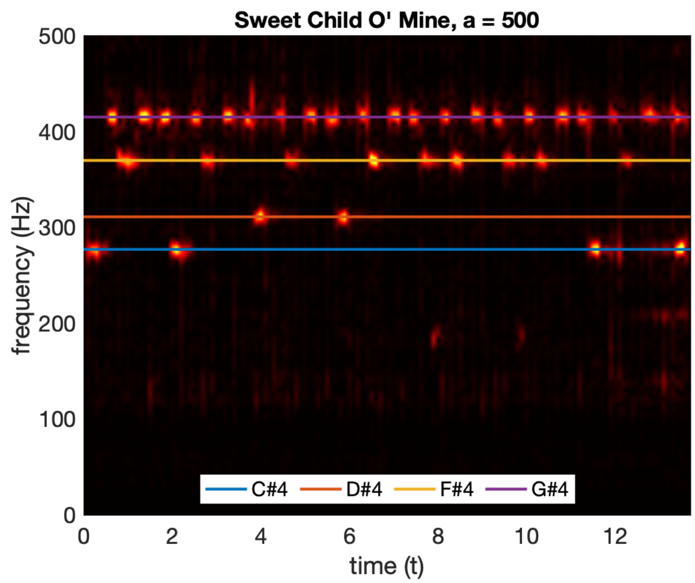
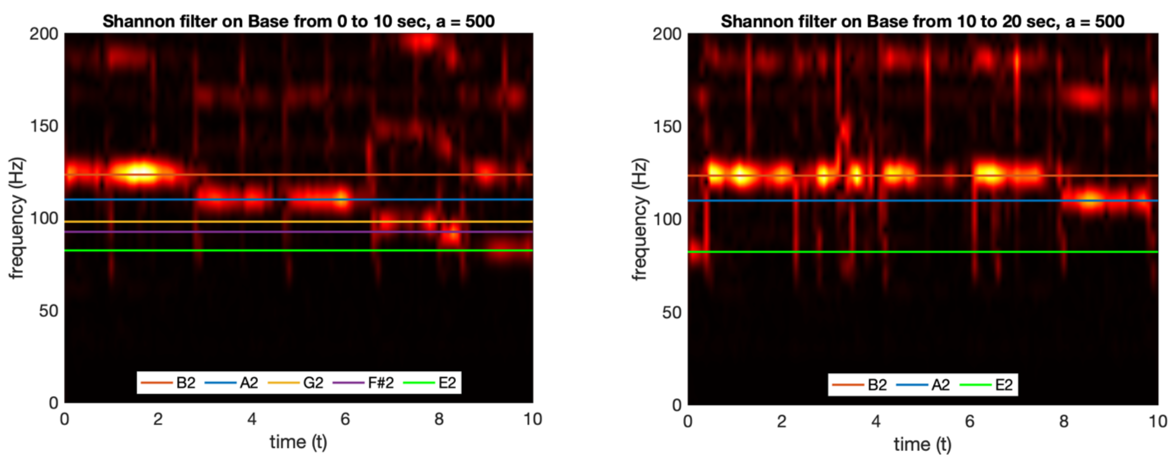


Figure 4: Spectrogram of Gabor transform on *Sweet Child O' Mine*

As shown in Figure 4, there are four notes played by the guitar in *Sweet Child O' Mine.* They are C#4, D#4, F#4, and G#4.

4.2 Base in *Comfortably Numb*

As shown in Figure 5, there are six notes played by the base in *Comfortably Numb.* They are B2, A2, G2, F#2, E2, D#2.



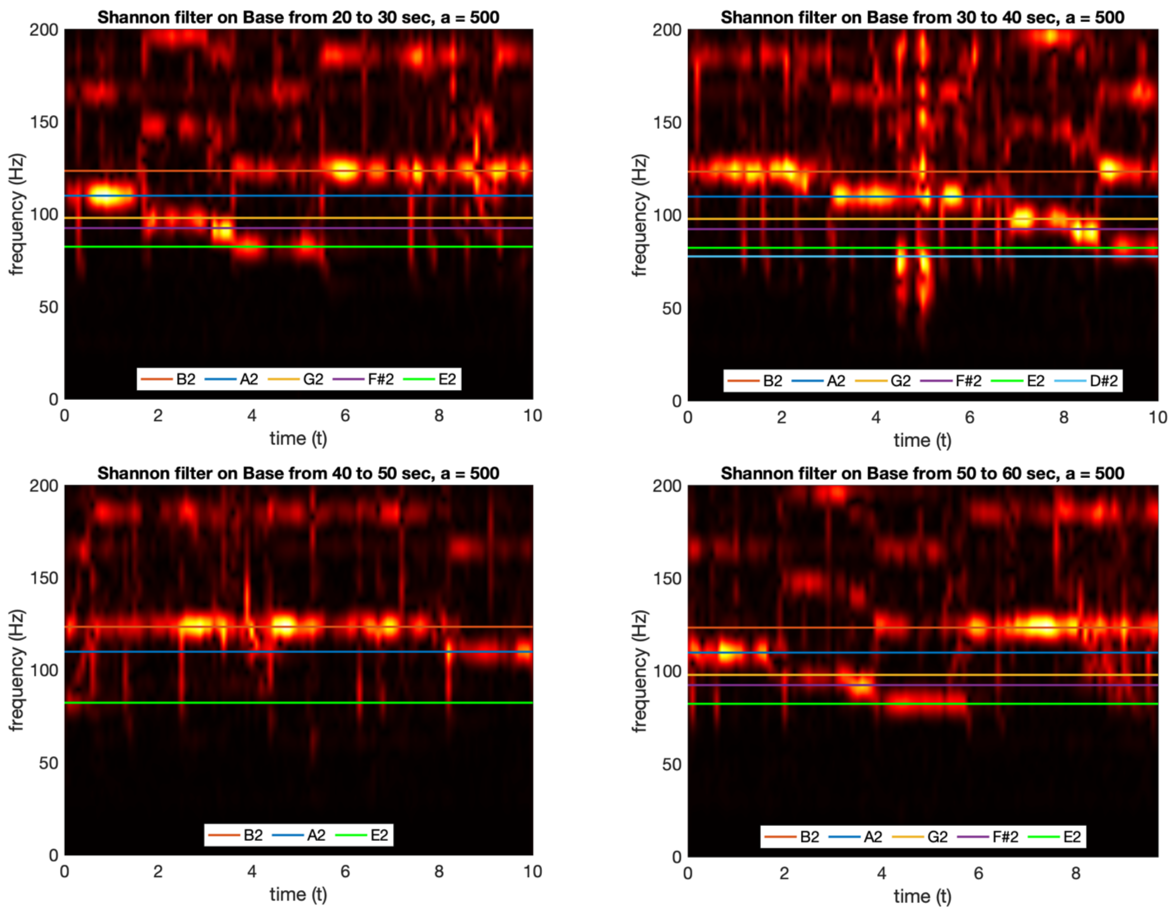
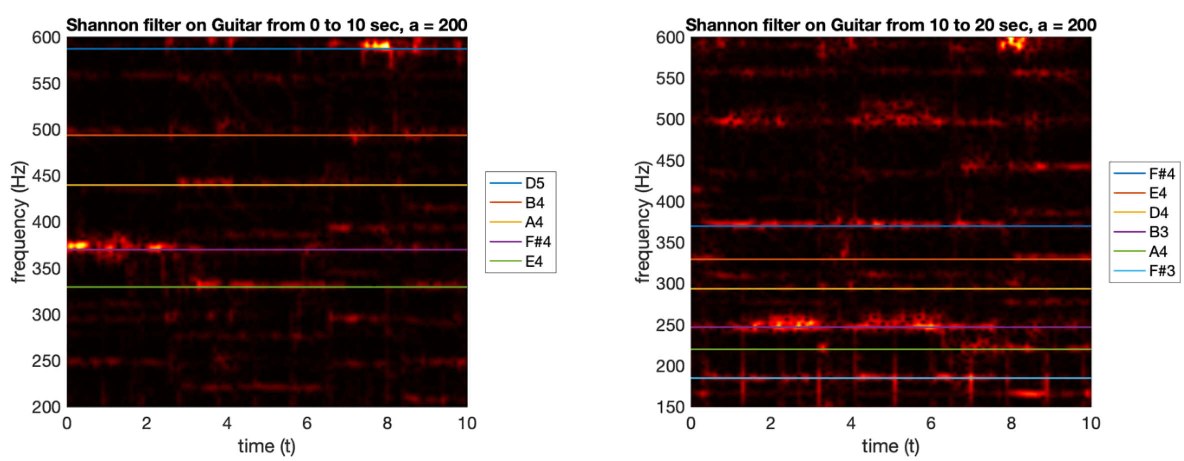


Figure 5: Spectrograms of the base in *Comfortably Numb* after the applications of Shannon filter and Gabor transform

4.3 Guitar in *Comfortably Numb*

The guitar solo consists of more notes comparing to the base. Vibrations and overtones are more significant. For example, when D3 is played, the spectrogram shows both the frequencies of D3 and D4, which is in a higher octave. Therefore, the spectrograms of the guitar notes are not as clean as the base one.



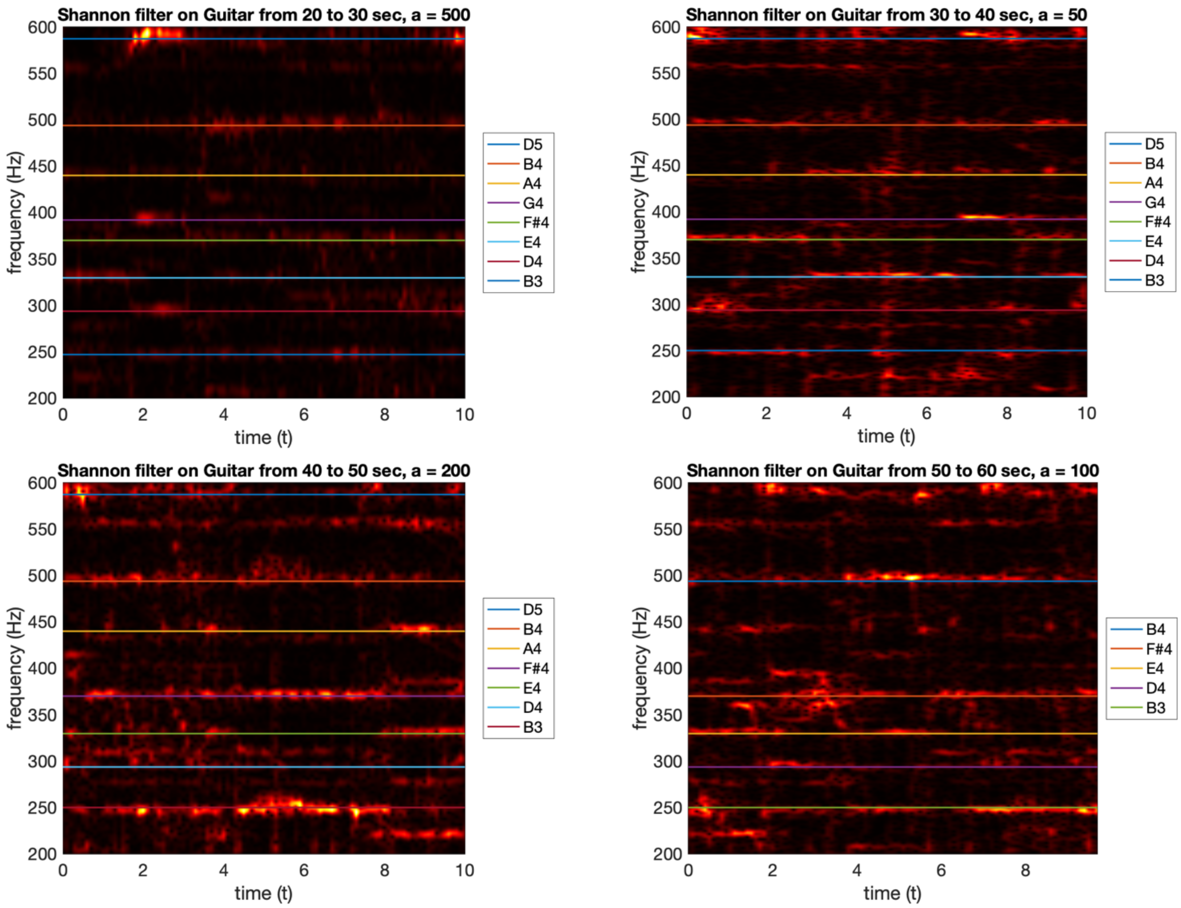


Figure 6: Spectrograms of the guitar in *Comfortably Numb* after the applications of Shannon filter and Gabor transform

5 Summary and Conclusions

Gabor transform allow us to perform a time-frequency analysis on the audio files of the song *Sweet Child O' Mine* and *Comfortably Numb* to reproduce their music score. By changing the window size, we are able to see the difference in resolution and strike a balance between frequency and time domain. Also, we use the Shannon function to filter out the frequencies in the interval $[0, 200]$ from the audio file of *Comfortably Numb* and isolate the base out. We then apply the same method on the frequency range of the guitar and plot the corresponding spectrograms. After adding lines of notes' frequencies, we successfully identify the music notes played in the two songs.

Appendix A. MATLAB functions

- `[y,Fs] = audioread(filename)` reads data from the file named filename, and returns sampled data, `y`, and a sample rate for that data, `Fs`.
- `Y = fft(X)` computes the discrete Fourier transform (DFT) of `X` using a fast Fourier transform (FFT) algorithm. If `X` is a vector, then `fft(X)` returns the Fourier transform of the vector. If `X` is a matrix, then `fft(X)` treats the columns of `X` as vectors and returns the Fourier transform of each column. If `X` is a multidimensional array, then `fft(X)` treats the values along the first array dimension whose size does not equal 1 as vectors and returns the Fourier transform of each vector.
- `Y = fftshift(X)` rearranges a Fourier transform `X` by shifting the zero-frequency component to the center of the array. If `X` is a vector, then `fftshift` swaps the left and right halves of `X`. If `X` is a matrix, then `fftshift` swaps the first quadrant of `X` with the third, and the second quadrant with the fourth. If X is a multidimensional array, then `fftshift` swaps half-spaces of X along each dimension.
- `pcolor(X,Y,C)` specifies the *x*- and *y*-coordinates for the vertices. The size of `C` must match the size of the *x*-*y* coordinate grid. For example, if `X` and `Y` define an *m*-by-*n* grid, then `C` must be an *m*-by-*n* matrix.
- `colormap(map)` sets the colormap for the current figure to the colormap specified by `map`

Appendix B. MATLAB codes

```
% Clean workspace
clear all; close all; clc
[y, Fs] = audioread('GNR.m4a');
tr_gnr = length(y)/Fs; % record time in seconds
plot((1:length(y))/Fs,y);
xlabel('Time [sec]'); ylabel('Amplitude');
title('Sweet Child O'' Mine');
print('gnr.png','-dpng')
%p8 = audioplayer(y,Fs); playblocking(p8);
L = tr_gnr;
n = length(y);
k = (1/L)*[0:n/2-1 -n/2:-1];
ks = fftshift(k);
a = 500;
tau = 0:0.1:tr_gnr;
y = y';
t = (1:length(y))/Fs;
for j = 1:length(tau)
    g = exp(-a*(t-tau(j)).^2);
    Yg = g.*y;
    Ygt = fft(Yg);
    Ygt_spec(j,:) = fftshift(abs(Ygt));
    subplot(3,1,1)
    plot(t,y,'k','Linewidth',2), axis([0 L -0.1 1])
    hold on
    plot(t,g,'m','Linewidth',2)
    set(gca,'Fontsize',16), xlabel('time (t)'), ylabel('y(t)')
    subplot(3,1,2) % Time domain
    plot(t,Yg,'k','Linewidth',2), axis([0 L -0.05 0.05])
    set(gca,'Fontsize',16), xlabel('time (t)'), ylabel('y(t)*g(t-\tau)')
```

```matlab
    subplot(3,1,3) % Fourier domain
    plot(ks,abs(fftshift(Ygt))/max(abs(Ygt)),'r','Linewidth',2);
    set(gca,'Fontsize',16),
    xlabel('frequency (Hz)'), ylabel('fft(y(t)*g(t-\tau))')
    drawnow
    print('gnr_plot.png','-dpng')
    pause(0.01); close all;
end

pcolor(tau,ks,Ygt_spec.')
hold on
cs = plot([0 14], [277.18 277.18], 'Linewidth', 2);
hold on
ds = plot([0 14], [311.13 311.13], 'Linewidth', 2);
hold on
fs = plot([0 14], [369.99 369.99], 'Linewidth', 2);
hold on
gs = plot([0 14], [415.3 415.3], 'Linewidth', 2);
shading interp
set(gca,'ylim',[0 500],'Fontsize',16)
colormap(hot)
xlabel('time (t)'), ylabel('frequency (Hz)')
title('Sweet Child O'' Mine, a = 500','Fontsize',16)
legend([cs,ds,fs,gs],'C#4','D#4','F#4','G#4','Location','south', ...
'Orientation','horizontal')
print('spectrogram_gnr.png','-dpng')

%% % plot floyd signal
clear all; close all; clc
[y, Fs] = audioread('Floyd.m4a');
tr_floyd = length(y)/Fs; % record time in seconds
plot((1:length(y))/Fs,y);
xlabel('Time [sec]'); ylabel('Amplitude');
title('Comfortably Numb');
print('floyd.png','-dpng')
%p8 = audioplayer(y,Fs); playblocking(p8);
L = tr_floyd;
n = length(y(1:2635920));
k = (1/L)*[0:n/2-1 -n/2:-1];
ks = fftshift(k);
a = 500;
tau = 0:0.1:tr_floyd;
y = y(1:2635920)';
t = (1:length(y))/Fs;
for j = 1:length(tau)
    g = exp(-a*(t-tau(j)).^2);
    Yg = g.*y;
    Ygt = fft(Yg);
    Ygt_spec(j,:) = fftshift(abs(Ygt));
    subplot(3,1,1)
    plot(t,y,'k','Linewidth',2), axis([0 L -0.5 1])
    hold on
    plot(t,g,'m','Linewidth',2)
    set(gca,'Fontsize',16), xlabel('time (t)'), ylabel('y(t)')
    subplot(3,1,2) % Time domain
    plot(t,Yg,'k','Linewidth',2), axis([0 L -0.1 0.1])
    set(gca,'Fontsize',16), xlabel('time (t)'), ylabel('y(t)*g(t-\tau)')
    subplot(3,1,3) % Fourier domain
```

```
    plot(ks,abs(fftshift(Ygt))/max(abs(Ygt)),'r','Linewidth',2);
    set(gca,'Fontsize',16),
    xlabel('frequency (Hz)'), ylabel('fft(y(t)*g(t-\tau))')
    drawnow
    print('floyd_plot.png','-dpng')
    pause(0.01); close all;
end

%% % part 1 Shannon filter Base Floyd 0-10sec
clear all; close all; clc
[y, Fs] = audioread('Floyd-1.m4a');
tr_floyd = length(y)/Fs; % record time in seconds
plot((1:length(y))/Fs,y);
xlabel('Time [sec]'); ylabel('Amplitude');
title('Comfortably Numb');
%p8 = audioplayer(y,Fs); playblocking(p8);
L = tr_floyd;
n = length(y);
k = (1/L)*[0:n/2-1 -n/2:-1];
ks = fftshift(k);
a = 500;
tau = 0:0.1:tr_floyd;
y = y';
t = (1:length(y))/Fs;
Yst_spec = [];
width = 400;
s = abs(ks-0) <= width/2;
Ys = fftshift(s).*fft(y);
Yst = ifft(Ys);
for j = 1:length(tau)
    g = exp(-a*(t-tau(j)).^2);
    Ysgt = Yst.*g;
    Ysg = fft(Ysgt);
    Yst_spec(j,:)= fftshift(abs(Ysg));
end
pcolor(tau,ks,Yst_spec.');
shading interp
set(gca,'ylim',[0 200],'Fontsize',16)
hold on
a = plot([0 10], [110.00 110.00], 'Linewidth', 2);
hold on
b = plot([0 10], [123.47 123.47], 'Linewidth', 2);
hold on
g = plot([0 10], [97.999 97.999], 'Linewidth', 2);
hold on
fs = plot([0 10], [92.499 92.499], 'Linewidth', 2);
hold on
e = plot([0 10], [82.407 82.407],'g', 'Linewidth', 2);
legend([b,a,g,fs,e],'B2','A2','G2','F#2','E2','Location','south', ...
'Orientation','horizontal')
colormap(hot)
xlabel('time (t)'), ylabel('frequency (Hz)')
title('Shannon filter on Base from 0 to 10 sec, a = 500','Fontsize',16)
print('spectrogram_floyd_base_1.png','-dpng')

%% % part 1 Floyd 10-20sec
clear all; close all; clc
[y, Fs] = audioread('Floyd-2.m4a');
```

```
tr_floyd = length(y)/Fs; % record time in seconds
plot((1:length(y))/Fs,y);
xlabel('Time [sec]'); ylabel('Amplitude');
title('Comfortably Numb');
%p8 = audioplayer(y,Fs); playblocking(p8);
L = tr_floyd;
n = length(y);
k = (1/L)*[0:n/2-1 -n/2:-1];
ks = fftshift(k);
a = 500;
tau = 0:0.1:tr_floyd;
y = y';
t = (1:length(y))/Fs;
Yst_spec = [];
width = 400;
s = abs(ks-0) <= width/2;
Ys = fftshift(s).*fft(y);
Yst = ifft(Ys);
for j = 1:length(tau)
    g = exp(-a*(t-tau(j)).^2);
    Ysgt = Yst.*g;
    Ysg = fft(Ysgt);
    Yst_spec(j,:)= fftshift(abs(Ysg));
end
pcolor(tau,ks,Yst_spec.');
shading interp
set(gca,'ylim',[0 200],'Fontsize',16)
hold on
a = plot([0 10], [110.00 110.00], 'Linewidth', 2);
hold on
b = plot([0 10], [123.47 123.47], 'Linewidth', 2);
colormap(hot)
hold on
e = plot([0 10], [82.407 82.407],'g', 'Linewidth', 2);
legend([b,a,e],'B2','A2','E2','Location','south', ...
'Orientation','horizontal')
xlabel('time (t)'), ylabel('frequency (Hz)')
title('Shannon filter on Base from 10 to 20 sec, a = 500','Fontsize',16)
print('spectrogram_floyd_base_2.png','-dpng')

%% %% % part 1 Floyd 20-30sec
clear all; close all; clc
[y, Fs] = audioread('Floyd-3.m4a');
tr_floyd = length(y)/Fs; % record time in seconds
% plot((1:length(y))/Fs,y);
% xlabel('Time [sec]'); ylabel('Amplitude');
% title('Comfortably Numb');
%p8 = audioplayer(y,Fs); playblocking(p8);
L = tr_floyd;
n = length(y);
k = (1/L)*[0:n/2-1 -n/2:-1];
ks = fftshift(k);
a = 500;
tau = 0:0.1:tr_floyd;
y = y';
t = (1:length(y))/Fs;
Yst_spec = [];
width = 400;
```

```
s = abs(ks-0) <= width/2;
Ys = fftshift(s).*fft(y);
Yst = ifft(Ys);
for j = 1:length(tau)
    g = exp(-a*(t-tau(j)).^2);
    Ysgt = Yst.*g;
    Ysg = fft(Ysgt);
    Yst_spec(j,:)= fftshift(abs(Ysg));
end
pcolor(tau,ks,Yst_spec.');
shading interp
set(gca,'ylim',[0 200],'Fontsize',16)
hold on
a = plot([0 10], [110.00 110.00], 'Linewidth', 2);
hold on
b = plot([0 10], [123.47 123.47], 'Linewidth', 2);
hold on
g = plot([0 10], [97.999 97.999], 'Linewidth', 2);
hold on
fs = plot([0 10], [92.499 92.499], 'Linewidth', 2);
colormap(hot)
hold on
e = plot([0 10], [82.407 82.407],'g', 'Linewidth', 2);
legend([b,a,g,fs,e],'B2','A2','G2','F#2','E2','Location','south', ...
'Orientation','horizontal')
xlabel('time (t)'), ylabel('frequency (Hz)')
title('Shannon filter on Base from 20 to 30 sec, a = 500','Fontsize',16)
print('spectrogram_floyd_base_3.png','-dpng')

%% part 1 Floyd 30-40sec
clear all; close all; clc
[y, Fs] = audioread('Floyd-4.m4a');
tr_floyd = length(y)/Fs; % record time in seconds
% plot((1:length(y))/Fs,y);
% xlabel('Time [sec]'); ylabel('Amplitude');
% title('Comfortably Numb');
%p8 = audioplayer(y,Fs); playblocking(p8);
L = tr_floyd;
n = length(y);
k = (1/L)*[0:n/2-1 -n/2:-1];
ks = fftshift(k);
a = 500;
tau = 0:0.1:tr_floyd;
y = y';
t = (1:length(y))/Fs;
Yst_spec = [];
width = 400;
s = abs(ks-0) <= width/2;
Ys = fftshift(s).*fft(y);
Yst = ifft(Ys);
for j = 1:length(tau)
    g = exp(-a*(t-tau(j)).^2);
    Ysgt = Yst.*g;
    Ysg = fft(Ysgt);
    Yst_spec(j,:)= fftshift(abs(Ysg));
end
pcolor(tau,ks,Yst_spec.');
shading interp
```

```
set(gca,'ylim',[0 200],'Fontsize',16)
hold on
a = plot([0 10], [110.00 110.00], 'Linewidth', 2);
hold on
b = plot([0 10], [123.47 123.47], 'Linewidth', 2);
hold on
g = plot([0 10], [97.999 97.999], 'Linewidth', 2);
hold on
fs = plot([0 10], [92.499 92.499], 'Linewidth', 2);
hold on
e = plot([0 10], [82.407 82.407],'g', 'Linewidth', 2);
hold on
ds = plot([0 10], [77.782 77.782], 'Linewidth', 2);
legend([b,a,g,fs,e,ds],'B2','A2','G2','F#2','E2','D#2','Location', ...
    'south','Orientation','horizontal')
colormap(hot)
xlabel('time (t)'), ylabel('frequency (Hz)')
title('Shannon filter on Base from 30 to 40 sec, a = 500','Fontsize',16)
print('spectrogram_floyd_base_4.png','-dpng')

%% part 1 Floyd 40-50sec
clear all; close all; clc
[y, Fs] = audioread('Floyd-5.m4a');
tr_floyd = length(y)/Fs; % record time in seconds
% plot((1:length(y))/Fs,y);
% xlabel('Time [sec]'); ylabel('Amplitude');
% title('Comfortably Numb');
%p8 = audioplayer(y,Fs); playblocking(p8);
L = tr_floyd;
n = length(y);
k = (1/L)*[0:n/2-1 -n/2:-1];
ks = fftshift(k);
a = 500;
tau = 0:0.1:tr_floyd;
y = y';
t = (1:length(y))/Fs;
Yst_spec = [];
width = 400;
s = abs(ks-0) <= width/2;
Ys = fftshift(s).*fft(y);
Yst = ifft(Ys);
for j = 1:length(tau)
    g = exp(-a*(t-tau(j)).^2);
    Ysgt = Yst.*g;
    Ysg = fft(Ysgt);
    Yst_spec(j,:)= fftshift(abs(Ysg));
end
pcolor(tau,ks,Yst_spec.');
shading interp
set(gca,'ylim',[0 200],'Fontsize',16)
hold on
a = plot([0 10], [110.00 110.00], 'Linewidth', 2);
hold on
b = plot([0 10], [123.47 123.47], 'Linewidth', 2);
hold on
e = plot([0 10], [82.407 82.407],'g', 'Linewidth', 2);
legend([b,a,e],'B2','A2','E2','Location','south', ...
'Orientation','horizontal')
```

```
colormap(hot)
xlabel('time (t)'), ylabel('frequency (Hz)')
title('Shannon filter on Base from 40 to 50 sec, a = 500','Fontsize',16)
print('spectrogram_floyd_base_5.png','-dpng')

%% %% part 1 Floyd 50-60sec
clear all; close all; clc
[y, Fs] = audioread('Floyd-6.m4a');
tr_floyd = length(y)/Fs; % record time in seconds
L = tr_floyd;
n = length(y);
k = (1/L)*[0:n/2-1 -n/2:-1];
ks = fftshift(k);
a = 500;
tau = 0:0.1:tr_floyd;
y = y(1:431120)';
t = (1:length(y))/Fs;
Yst_spec = [];
width = 400;
s = abs(ks-0) <= width/2;
Ys = fftshift(s).*fft(y);
Yst = ifft(Ys);
for j = 1:length(tau)
    g = exp(-a*(t-tau(j)).^2);
    Ysgt = Yst.*g;
    Ysg = fft(Ysgt);
    Yst_spec(j,:)= fftshift(abs(Ysg));
end
pcolor(tau,ks,Yst_spec.');
shading interp
set(gca,'ylim',[0 200],'Fontsize',16)
hold on
a = plot([0 10], [110.00 110.00], 'Linewidth', 2);
hold on
b = plot([0 10], [123.47 123.47], 'Linewidth', 2);
hold on
g = plot([0 10], [97.999 97.999], 'Linewidth', 2);
hold on
fs = plot([0 10], [92.499 92.499], 'Linewidth', 2);
hold on
e = plot([0 10], [82.407 82.407],'g', 'Linewidth', 2);
colormap(hot)
legend([b,a,g,fs,e],'B2','A2','G2','F#2','E2','Location','south', ...
'Orientation','horizontal')
xlabel('time (t)'), ylabel('frequency (Hz)')
title('Shannon filter on Base from 50 to 60 sec, a = 500','Fontsize',16)
print('spectrogram_floyd_base_6.png','-dpng')

%% % No Shannon filter Floyd 0-10sec
clear all; close all; clc
[y, Fs] = audioread('Floyd-1.m4a'); % parts of original audio file
tr_floyd = length(y)/Fs; % record time in seconds
L = tr_floyd;
n = length(y);
k = (1/L)*[0:n/2-1 -n/2:-1];
ks = fftshift(k);
a = 500;
tau = 0:0.1:tr_floyd;
```

```
y = y';
t = (1:length(y))/Fs;
for j = 1:length(tau)
    g = exp(-a*(t-tau(j)).^2);
    Yg = g.*y;
    Ygt = fft(Yg);
    Ygt_spec(j,:) = fftshift(abs(Ygt));
end
pcolor(tau,ks,Ygt_spec.');
set(gca,'ylim',[0 500],'Fontsize',16)
shading interp
colormap(hot)
xlabel('time (t)'), ylabel('frequency (Hz)')
title('Comfortably Numb from 0 to 10 sec, a = 500','Fontsize',16)
print('spectrogram_floyd_500.png','-dpng')

%% % Shannon filter Guitar Floyd 0-10sec
clear all; close all; clc
[y, Fs] = audioread('Floyd-1.m4a'); % parts of original audio file
tr_floyd = length(y)/Fs; % record time in seconds
L = tr_floyd;
n = length(y);
k = (1/L)*[0:n/2-1 -n/2:-1];
ks = fftshift(k);
a = 200;
tau = 0:0.1:tr_floyd;
y = y';
t = (1:length(y))/Fs;
Yst_spec = [];
width = 400;
s = abs(ks-400) <= width/2; % center at 400Hz
Ys = fftshift(s).*fft(y);
Yst = ifft(Ys);
for j = 1:length(tau)
    g = exp(-a*(t-tau(j)).^2);
    Ysgt = Yst.*g;
    Ysg = fft(Ysgt);
    Yst_spec(j,:)= fftshift(abs(Ysg));
end
pcolor(tau,ks,Yst_spec.');
hold on
d = plot([0 10], [587.33 587.33], 'Linewidth', 1.5);
hold on
b = plot([0 10], [493.88 493.88], 'Linewidth', 1.5);
hold on
a = plot([0 10], [440 440], 'Linewidth', 1.5);
hold on
fs = plot([0 10], [369.99 369.99],'Linewidth', 1.5);
hold on
e = plot([0 10], [329.63 329.63],'Linewidth', 1.5);
shading interp
legend([d,b,a,fs,e],'D5','B4','A4','F#4','E4','Location','eastoutside', ...
'Orientation','vertical')
set(gca,'ylim',[200 600],'Fontsize',16)
colormap(hot)
xlabel('time (t)'), ylabel('frequency (Hz)')
title('Shannon filter on Guitar from 0 to 10 sec, a = 200','Fontsize',16)
print('spectrogram_floyd_guitar_1.png','-dpng')
```

```
%% % Shannon filter Guitar Floyd 10-20sec
clear all; close all; clc
[y, Fs] = audioread('Floyd-2.m4a'); % parts of original audio file
tr_floyd = length(y)/Fs; % record time in seconds
L = tr_floyd;
n = length(y);
k = (1/L)*[0:n/2-1 -n/2:-1];
ks = fftshift(k);
a = 200;
tau = 0:0.1:tr_floyd;
y = y';
t = (1:length(y))/Fs;
Yst_spec = [];
width = 500;
s = abs(ks-400) <= width/2;
Ys = fftshift(s).*fft(y);
Yst = ifft(Ys);
for j = 1:length(tau)
    g = exp(-a*(t-tau(j)).^2);
    Ysgt = Yst.*g;
    Ysg = fft(Ysgt);
    Yst_spec(j,:)= fftshift(abs(Ysg));
end
pcolor(tau,ks,Yst_spec.');
set(gca,'ylim',[150 600],'Fontsize',16)
hold on
fs = plot([0 10], [369.99 369.99],'Linewidth', 1.5);
hold on
e = plot([0 10], [329.63 329.63],'Linewidth', 1.5);
hold on
d2 = plot([0 10], [293.66 293.66], 'Linewidth', 1.5);
hold on
b = plot([0 10], [246.94 246.94], 'Linewidth', 1.5);
hold on
a = plot([0 10], [220 220], 'Linewidth', 1.5);
hold on
fs2 = plot([0 10], [185 185], 'Linewidth', 1.5);
legend([fs,e,d2,b,a,fs2],'F#4','E4','D4','B3','A4','F#3','Location', ...
    'eastoutside','Orientation','vertical')
shading interp
colormap(hot)
xlabel('time (t)'), ylabel('frequency (Hz)')
title('Shannon filter on Guitar from 10 to 20 sec, a = 200','Fontsize',16)
print('spectrogram_floyd_guitar_2.png','-dpng')

%% % Shannon filter Guitar Floyd 20-30sec
clear all; close all; clc
[y, Fs] = audioread('Floyd-3.m4a'); % parts of original audio file
tr_floyd = length(y)/Fs; % record time in seconds
L = tr_floyd;
n = length(y);
k = (1/L)*[0:n/2-1 -n/2:-1];
ks = fftshift(k);
a = 500;
tau = 0:0.1:tr_floyd;
y = y';
t = (1:length(y))/Fs;
```

```
Yst_spec = [];
width = 400;
s = abs(ks-400) <= width/2;
Ys = fftshift(s).*fft(y);
Yst = ifft(Ys);
for j = 1:length(tau)
    g = exp(-a*(t-tau(j)).^2);
    Ysgt = Yst.*g;
    Ysg = fft(Ysgt);
    Yst_spec(j,:)= fftshift(abs(Ysg));
end
pcolor(tau,ks,Yst_spec.');
set(gca,'ylim',[200 600],'Fontsize',16)
hold on
d = plot([0 10], [587.33 587.33], 'Linewidth', 1.5);
hold on
b = plot([0 10], [493.88 493.88], 'Linewidth', 1.5);
hold on
a = plot([0 10], [440 440], 'Linewidth', 1.5);
hold on
g = plot([0 10], [392 392], 'Linewidth', 1.5);
hold on
fs = plot([0 10], [369.99 369.99],'Linewidth', 1.5);
hold on
e = plot([0 10], [329.63 329.63],'Linewidth', 1.5);
hold on
d2 = plot([0 10], [293.66 293.66], 'Linewidth', 1.5);
hold on
b2 = plot([0 10], [246.94 246.94], 'Linewidth', 1.5);
shading interp
legend([d,b,a,g,fs,e,d2,b2],'D5','B4','A4','G4','F#4','E4','D4','B3',...
    'Location', 'eastoutside','Orientation','vertical')
colormap(hot)
xlabel('time (t)'), ylabel('frequency (Hz)')
title('Shannon filter on Guitar from 20 to 30 sec, a = 500','Fontsize',16)
print('spectrogram_floyd_guitar_3.png','-dpng')

%% % Shannon filter Guitar Floyd 30-40sec
clear all; close all; clc
[y, Fs] = audioread('Floyd-4.m4a'); % parts of original audio file
tr_floyd = length(y)/Fs; % record time in seconds
L = tr_floyd;
n = length(y);
k = (1/L)*[0:n/2-1 -n/2:-1];
ks = fftshift(k);
a = 50;
tau = 0:0.1:tr_floyd;
y = y';
t = (1:length(y))/Fs;
Yst_spec = [];
width = 400;
s = abs(ks-400) <= width/2;
Ys = fftshift(s).*fft(y);
Yst = ifft(Ys);
for j = 1:length(tau)
    g = exp(-a*(t-tau(j)).^2);
    Ysgt = Yst.*g;
    Ysg = fft(Ysgt);
```

```
    Yst_spec(j,:)= fftshift(abs(Ysg));
end
pcolor(tau,ks,Yst_spec.');
set(gca,'ylim',[200 600],'Fontsize',16)
hold on
d = plot([0 10], [587.33 587.33], 'Linewidth', 1.5);
hold on
b2 = plot([0 10], [493.88 493.88], 'Linewidth', 1.5);
hold on
a = plot([0 10], [440 440], 'Linewidth', 1.5);
hold on
g = plot([0 10], [392 392], 'Linewidth', 1.5);
hold on
fs = plot([0 10], [369.99 369.99], 'Linewidth', 1.5);
hold on
e = plot([0 10], [329.63 329.63], 'Linewidth', 1.5);
hold on
d2 = plot([0 10], [293.66 293.66], 'Linewidth', 1.5);
hold on
b = plot([0 10], [249.94 249.94], 'Linewidth', 1.5);
legend([d,b2,a,g,fs,e,d2,b],'D5','B4','A4','G4','F#4','E4','D4','B3',...
    'Location', 'eastoutside','Orientation','vertical')
shading interp
colormap(hot)
xlabel('time (t)'), ylabel('frequency (Hz)')
title('Shannon filter on Guitar from 30 to 40 sec, a = 50','Fontsize',16)
print('spectrogram_floyd_guitar_4.png','-dpng')

%% % Shannon filter Guitar Floyd 40-50sec
clear all; close all; clc
[y, Fs] = audioread('Floyd-5.m4a'); % parts of original audio file
tr_floyd = length(y)/Fs; % record time in seconds
L = tr_floyd;
n = length(y);
k = (1/L)*[0:n/2-1 -n/2:-1];
ks = fftshift(k);
a = 200;
tau = 0:0.1:tr_floyd;
y = y';
t = (1:length(y))/Fs;
Yst_spec = [];
width = 400;
s = abs(ks-400) <= width/2;
Ys = fftshift(s).*fft(y);
Yst = ifft(Ys);
for j = 1:length(tau)
    g = exp(-a*(t-tau(j)).^2);
    Ysgt = Yst.*g;
    Ysg = fft(Ysgt);
    Yst_spec(j,:)= fftshift(abs(Ysg));
end
pcolor(tau,ks,Yst_spec.');
set(gca,'ylim',[200 600],'Fontsize',16)
hold on
d = plot([0 10], [587.33 587.33], 'Linewidth', 1.5);
hold on
b2 = plot([0 10], [493.88 493.88], 'Linewidth', 1.5);
hold on
```

```
a = plot([0 10], [440 440], 'Linewidth', 1.5);
hold on
fs = plot([0 10], [369.99 369.99], 'Linewidth', 1.5);
hold on
e = plot([0 10], [329.63 329.63], 'Linewidth', 1.5);
hold on
d2 = plot([0 10], [293.66 293.66], 'Linewidth', 1.5);
hold on
b = plot([0 10], [249.94 249.94], 'Linewidth', 1.5);
shading interp
legend([d,b2,a,fs,e,d2,b],'D5','B4','A4','F#4','E4','D4','B3',...
    'Location', 'eastoutside','Orientation','vertical')
colormap(hot)
xlabel('time (t)'), ylabel('frequency (Hz)')
title('Shannon filter on Guitar from 40 to 50 sec, a = 200','Fontsize',16)
print('spectrogram_floyd_guitar_5.png','-dpng')

%% % Shannon filter Guitar Floyd 50-60sec
clear all; close all; clc
[y, Fs] = audioread('Floyd-6.m4a'); % parts of original audio file
tr_floyd = length(y)/Fs; % record time in seconds
L = tr_floyd;
n = length(y);
k = (1/L)*[0:n/2-1 -n/2:-1];
ks = fftshift(k);
a = 100;
tau = 0:0.1:tr_floyd;
y = y(1:431120)';
t = (1:length(y))/Fs;
Yst_spec = [];
width = 400;
s = abs(ks-400) <= width/2;
Ys = fftshift(s).*fft(y);
Yst = ifft(Ys);
for j = 1:length(tau)
    g = exp(-a*(t-tau(j)).^2);
    Ysgt = Yst.*g;
    Ysg = fft(Ysgt);
    Yst_spec(j,:)= fftshift(abs(Ysg));
end
pcolor(tau,ks,Yst_spec.');
set(gca,'ylim',[200 600],'Fontsize',16)
hold on
b2 = plot([0 10], [493.88 493.88], 'Linewidth', 1.5);
hold on
fs = plot([0 10], [369.99 369.99], 'Linewidth', 1.5);
hold on
e = plot([0 10], [329.63 329.63], 'Linewidth', 1.5);
hold on
d2 = plot([0 10], [293.66 293.66], 'Linewidth', 1.5);
hold on
b = plot([0 10], [249.94 249.94], 'Linewidth', 1.5);
shading interp
legend([b2,fs,e,d2,b],'B4','F#4','E4','D4','B3',...
    'Location', 'eastoutside','Orientation','vertical')
colormap(hot)
xlabel('time (t)'), ylabel('frequency (Hz)')
title('Shannon filter on Guitar from 50 to 60 sec, a = 100','Fontsize',16)
```

```
print('spectrogram_floyd_guitar_6.png','-dpng')
```

Appendix C. Additional plots

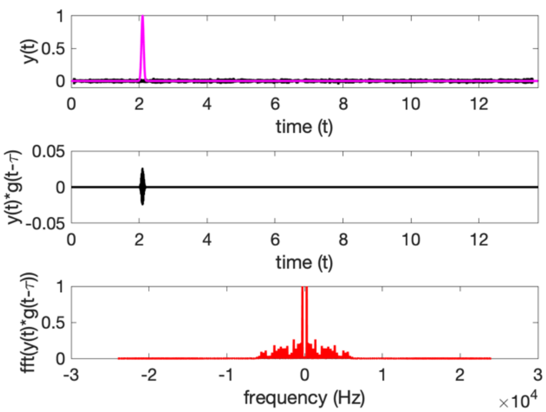


Figure 7: Top: *Sweet Child O' Mine*'s signal against time. Middle: Gaussian filter on the signal. Bottom: Fourier transform of the signal.

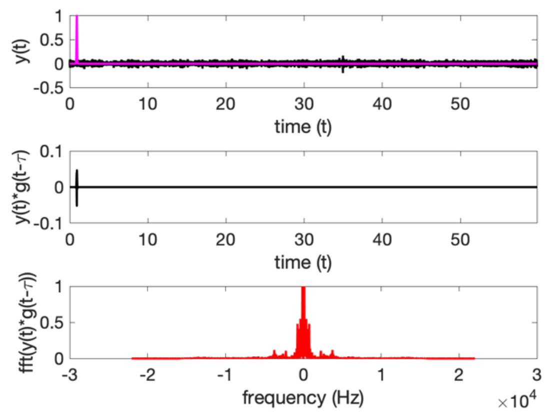


Figure 8: Top: *Comfortably Numb's* signal against time. Middle: Gaussian filter on the signal. Bottom: Fourier transform of the signal.