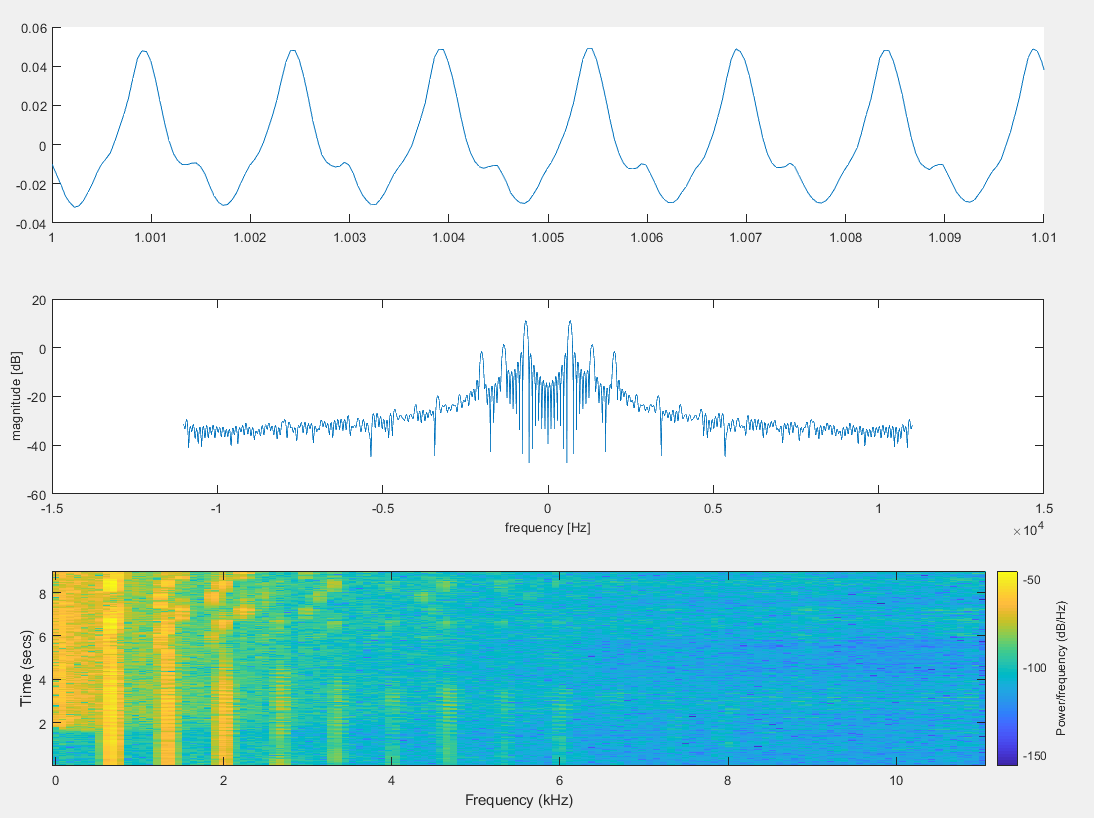
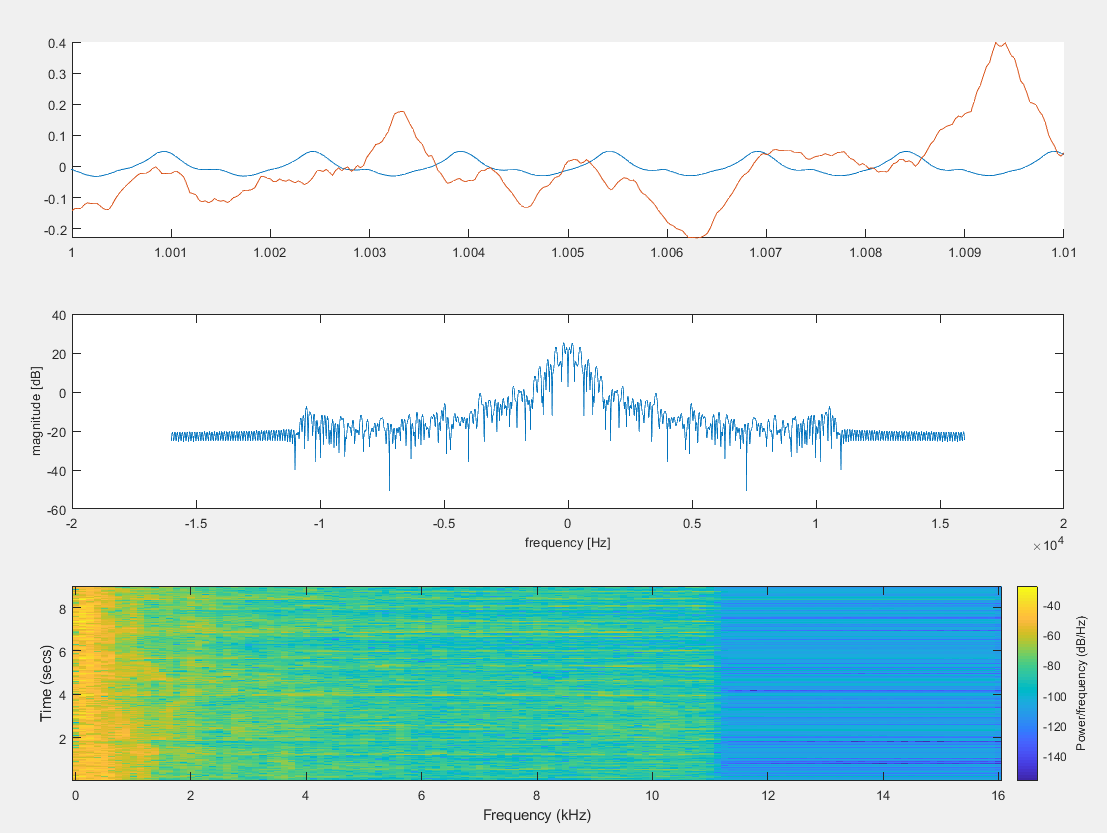
In this assignment I learned how to interface with files and convert between file formats. I also learned how to graph data in a way that is meaningful. During this assignment I was able to understand the data format and features of both audio and image files in MATLAB. The skill about which I learned the most is the plotting function. I am now able to place my data in a plot that meets the criteria necessary for the project.

Problem 1

Output from flute22.wav



Output from music.mp3



Code from problem 1:

%filename = 'flute22.wav' %put the file path inside these quotes

filename = 'music.mp3'

ai = audioinfo(filename) ;

disp(ai)% display audio header structure

[originalsoundData,fs] = audioread(filename,[1 10]\*ai.SampleRate); % read audio file

%part A

resultBinFile = audio2bin(filename, ai.SampleRate)

%part B

newWavFile = bin2audio(resultBinFile, ai.SampleRate)

%part C

ai = audioinfo(newWavFile);

[processedsoundData,fs] = audioread(newWavFile,[1 10]\*ai.SampleRate); % read audio file

soundsc(processedsoundData, fs)

%part D

subplot(3,1,1)

t = [0:length(processedsoundData)-1]/fs;

hold on

xlim([1 1.01]);

plot(t, processedsoundData)

hold off

%part E

subplot(3,1,2)

t1 = 1.0; % seconds

t2 = 1.01; % seconds

i1 = round(t1\*fs); % convert time to index

i2 = round(t2\*fs); % convert time to index

nfft = 2^12; % FFT size

freq = ([0: nfft-1]/nfft-0.5)\*fs; % frequency [Hz]

X = fft (processedsoundData(i1:i2),nfft); % compute the discrete - Fourier transform

plot(freq ,20\*log10(abs(fftshift(X))));

% plot with accurately scaled frequency axis

xlabel ('frequency [Hz]','FontSize' ,10) ;

ylabel ('magnitude [dB]','FontSize' ,10) ;

%part F

subplot(3,1,3)

nfft = 2^8; % FFT size

overlap = round (0.8\*nfft);

window = hamming(nfft) ;

spectrogram (processedsoundData,window ,overlap ,nfft ,fs ) ;

function y = audio2bin(filename, samplerate)

[soundData, fs] = audioread(filename);

% t = [0: length(soundData)-1]/fs;

% plot(t,soundData)

outputfilename = 'binaryout.bin'

fileID = fopen( outputfilename,'w' );

fwrite(fileID,soundData,'float');

fclose(fileID);

y =outputfilename;

end

function y = bin2audio(filename, samplerate)

outputfilename = 'outputAudio.wav';

binfile = fopen(filename,'rb');

if(binfile ==-1) fprintf('ERROR : Could not open file'); end

[s, cnt] = fread(binfile ,inf ,'float'); % read to the end of the file

fclose(binfile);

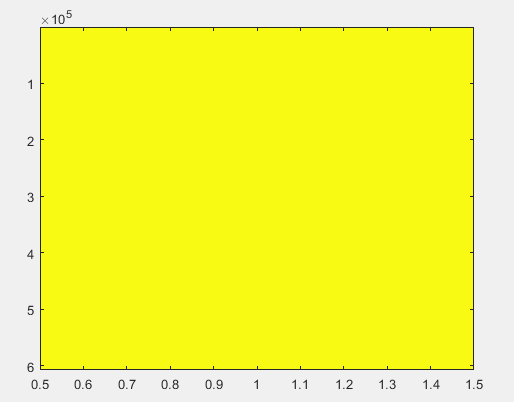
audiowrite(outputfilename, s, samplerate);

y = outputfilename;

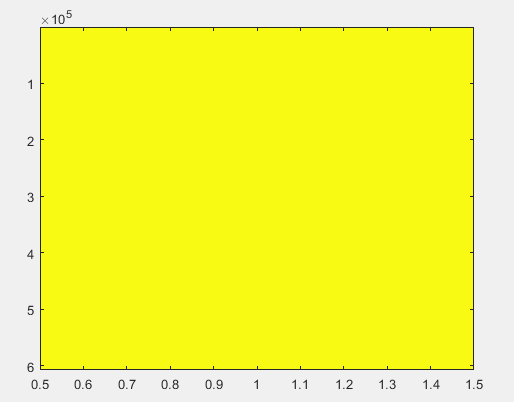
end

Problem 2

Output from coloredChips.png



Output from liftingbody.png



Code form problem 2

%inputImage = 'coloredChips.png'

inputImage = 'liftingbody.png'

%part A

imageBin = image2bin(inputImage)

%part B

newImage = bin2image(imageBin)

%part C

img = imread(newImage);

image(img)

function y = image2bin(filename)

outputfilename = 'binaryImage.bin'

imageData = imread(filename) ;

fileID = fopen(outputfilename,'w' );

fwrite(fileID,imageData,'uint8');

fclose(fileID);

y =outputfilename;

end

function y = bin2image(filename)

outputfilename = 'newImage.png';

binfile = fopen(filename,'rb');

if(binfile ==-1) fprintf('ERROR : Could not open file'); end

[rawImageData, cnt] = fread(binfile ,inf ,'uint8'); % read to the end of the file

fclose(binfile);

% r = abs ( real (x) ) ;

% g = abs ( imag (x) ) ;

% b = abs ( real (x) ) ;

% x = cat (3 ,r,g,b) ; % concatenate along the third dimension.

imagesc(rawImageData)

% imwrite(rawImageData,outputfilename,'PNG');

% image(rawImageData);

% axis image; % make the pixels square

% print -dpng dftmtx\_color\_matlab.png;

% [nrows ,ncols , nrgb ] = size (x) ;

y = outputfilename;

end