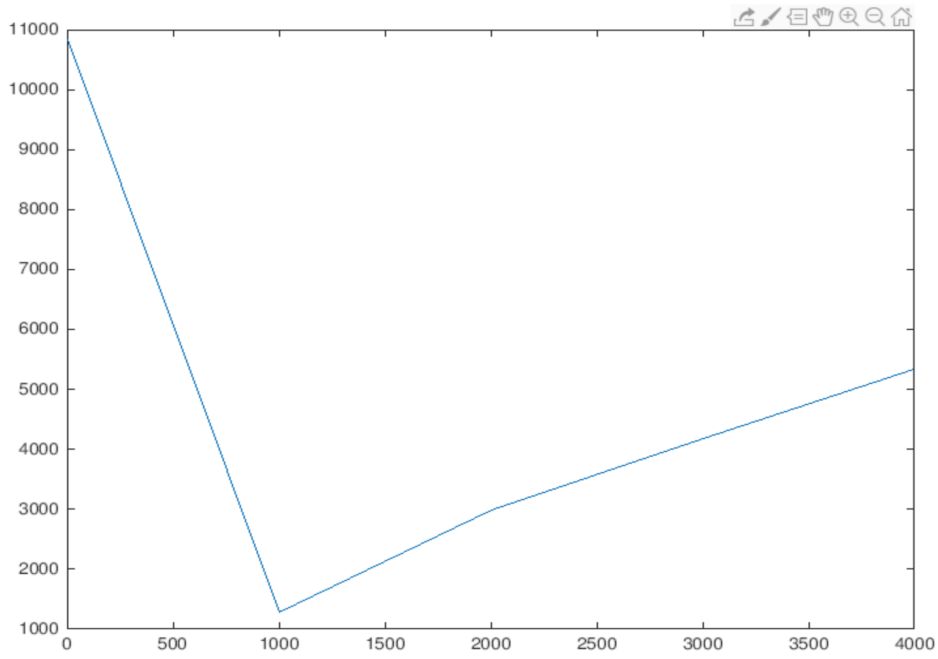
CMPE362: Introduction to Signals for Computer Engineers

Homework-3

Yağmur Ceren DARDAĞAN
2014400063

Question 1

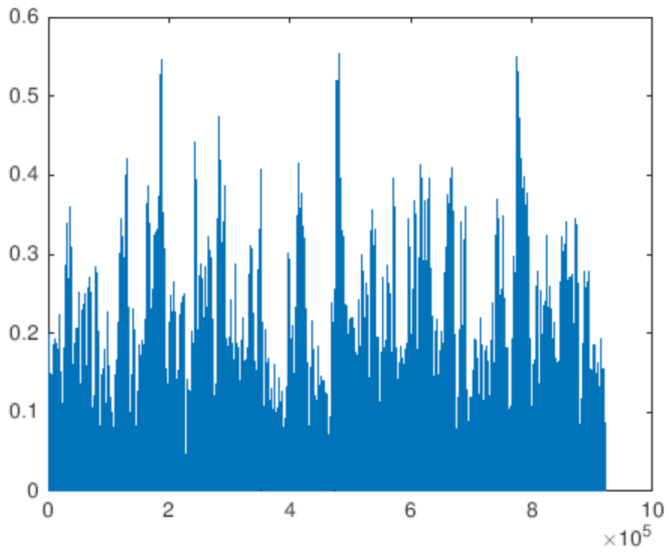


In this question, I designed a low-pass filter with fdesign library in matlab and applied this filter to data from csv file with different cut-off frequencies(1000Hz, 2000Hz, 3000Hz, 4000Hz).

```
clear;

data = csvread('exampleSignal.csv');
peakArray = zeros(1,5);
peakArray(1) = numel(findpeaks(data));

for freq = 1:4
    %designing lowpass filter
    d = fdesign.lowpass('N,Fc',90,1000*freq,22050);
    designmethods(d)
    Hd = design(d);
    %applying lowpass filter
    y = filter(Hd,data);

   peakArray(freq+1) = numel(findpeaks(y));

end
%plotting lowpass filter
x = 0:1000:4000;
plot(x,peakArray);
```

Question 2

In this question, we are asked to produce sound from an image's non black pixels. To achieve this goal, I first turn all the colored parts to gray with rgb2gray() method. Then I İnvestigate each pixel with 2 for loops. If the value of the pixel in image array higher than 40, it means this pixel corresponds to a nonblack pixel in the image. Then I use getAmplitude() function that I wrote to get pixel's amplitude with respect to it's row number. I am doing this for each pixel in the image and put each pixel into the signal_array. After each column examined, I am using ifft function for switching to time domain, even though I didn't use fft funtion to switch to frequency domain, all the values has already been in frequency domain since it has no imaginary value. After examining each pixel, I reshaped the array to produce audio that corresponds the image. Then I produced the audio with audiowrite() function with 900 sampling rate.




```
clear;
RGB = imread('Hubble-Massive-Panorama.png');
gray = rgb2gray(RGB);

signal_array = zeros(1024,900);
fs=900 ; % sampling frequency

%investigating each pixels with 2 for-loops
for Columns=1:1024
    for Rows= 1:900
        %if the value of the pixel higher than 40, it means pixel is not
        %black
        if RGB(Rows,Columns)>40
            %get amplitude of the pixel from the function
            Amplitude = getAmplitude(Rows) ;
            %make this pixel's amplitude different from zero to produce
            %signal at the and

            signal_array(Columns,Rows) = Amplitude;
        end


    end
        %turn respected column to time domain from freq domain
        %even if the column is not turned into freq domain before
        %actually it is in the freq domain because it doesn't have any
        %imaginary part
        signal_array(Columns,:) = abs(ifft(signal_array(Columns,:)));
end
%reshaping array as 1 column to produce audio
rSound = reshape(signal_array',1,[]);
%plotting the audio as a signal
plot(rSound)
%creating the audio signal with 900 sampling rate
audiowrite('SonifiedDeepSpace_sound.wav',rSound,fs);
```

```

%getAmplitude function returns the amplitude with respect to the decription
%it divides 900 rows into 10 groups and gives specified amplitude after
%examining it's row number
function amp = getAmplitude(row_number)
    if row_number>=0 && row_number<90
        amp=10;
    elseif row_number>=90 && row_number<180
        amp=9;
    elseif row_number>=180 && row_number<270
        amp=8;
    elseif row_number>=270 && row_number<360
        amp=7;
    elseif row_number>=360 && row_number<450
        amp=6;
    elseif row_number>=450 && row_number<540
        amp=5;
    elseif row_number>=540 && row_number<630
        amp=4;
    elseif row_number>=630 && row_number<720
        amp=3;
    elseif row_number>=720 && row_number<810
        amp=2;
    elseif row_number>=810 && row_number<=900
        amp=1;
    else
        disp('Value is above maximum value.')
    end

end
```