

CMPE 362

**INTRO.TO SIGNAL PROC. FOR
COMPUTER ENG.**

MatLab

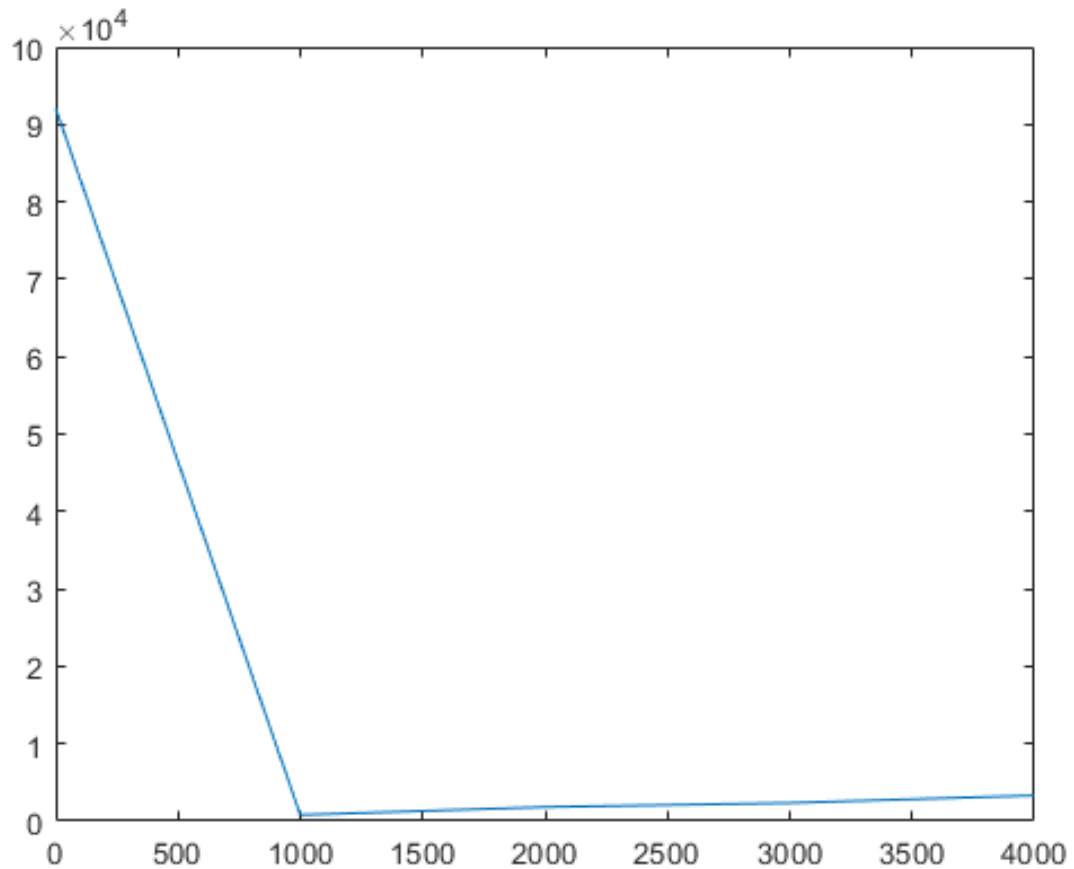
III. Project

BATURALP YÖRÜK

2015400036

26/05/2019

Problem 1



In this question, the number of peaks are too many at the zeroth element which is with no filter. It is due to the fact that there is no lowpass filter or any filter. However, the lowpass filter with 1000 cutoff frequency has very small number of peaks, since its cutoff is very small when compared to others. As long as cutoff frequency gets higher and higher, the number of peaks are increasing since they are less filtered.

Code1:

```
clc;
clear all;

[y,Fs] = audioread("PinkPanther30.wav");
cutoffs = [0, 1000, 2000, 3000, 4000]; % cutoff frequencies
results = zeros(1,5); % initialize the results array
results(1) = numel(findpeaks(y)); % the one with no cutoff

for i = 2:numel(results)
    myfilter = designfilt('lowpassiir','FilterOrder',8,
    'PassbandFrequency',cutoffs(i), ...
    'PassbandRipple',0.2,'SampleRate',22050); %filter with cutoff
    frequencies
    temp = filter(myfilter, y); % filtered data
    results(i) = numel(findpeaks(temp)); % assign the result
end

figure;
plot(cutoffs,results); %plot cutoff frequencies versus results
```

Problem 2

At this question, I first read the image and converted it to black and white version. After that, I converted it to binary values to understand whether it is black or not. Then, for every column, I assigned the necessary value to an array if that index has a non-black cell. After that, I convert it to time domain by using the ifft method. Then, I appended the 1024 arrays to have a 1024 seconds of sound. Lastly, I play it at the end of the file.

Code2:

```
clc;
clear all;

photo = imread('Hubble-Massive-Panorama.png'); % Read the image
grayphototemp = rgb2gray(photo);
grayphoto = imbinarize(photo); % The binary version of photo
timedomain = []; % Initialize the timedomain array
for j=1:1024
    temp = zeros(900,1); % For every column
    for i=900:-1:1
        if(grayphoto(i,j)==1)
            amp= fix(i/90)+1; % Amplitude
            temp(i) = amp*exp(-1i*2*pi*i); % Assign the value " amp *
e^(j*2*pi*i) "
        end
    end
    temp2 = ifft(temp,900); % Inverse Fourier Transform
    timedomain = [timedomain ; temp2] ; % Append them all to have 1024
seconds
end

filename = 'space_sound.wav'; % Create a new wav file
audiowrite(filename,timedomain,900); % Write it to file
[y,Fs] = audioread(filename); % Read the file
sound(y,2000); % Play the sound file
```