School of Engineering and Applied Science (SEAS)
Ahmedabad University

BTech(ICT) Digital Signal Processing (Section 1)

Laboratory Assignment-4 Question 4

Enrollment No: AU1841145 **Name: Samarth Shah**

AIM : Understand different concepts of Fourier transform along with its applications.

1. Solution Problem-1

   (a) Approach: 1) In this sub-question, audio was imported using audioread fuction. The output of the function was sampling frequency and a variable who has the same audio. Then, length function was used to get the full length of the audio signal. Then the frequency range of the audio was calculated ((0:(length of audio -1))*sampling-freq/length(audio-File)). Normalized frequency was calculated using this equation frequency/sampling frequency. In order to take the FFT of the signal, "fft" function was used and later the output was used as an input of absolute "fft" function and also for the "fftshit" function. This will give the magnitude spectrum of the audio signal. Then normalized frequency vs. Audio file and Normalized frequency vs. Absolute fft plots were plotted.
   2) In this sub question, the procedure till finding the normalized frequency was same. After that signal noise and it's range were created. Both signals were added. Noise signal should have transpose before adding to audio signal. Butterworth filter coefficients were generated and passed through filter along with final noise signal. Then both the original audio and filtered audio plots were plotted.

   (b) Matlab Script for sub-question 1:

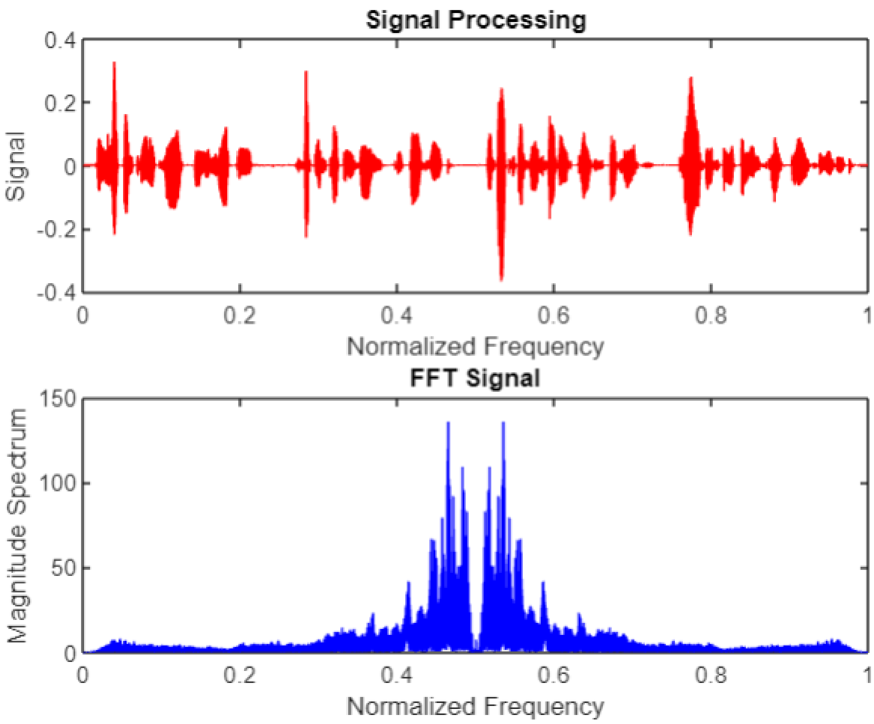
```
% Name : Samarth Shah
% Roll No: AU1841145
% Lab4 (Question_4) Take an audio file "sample_sound.wav" from shared folder.
    Apply different commends which are mentioned above on given audio file to find
     sampling frequency, frequency range of audio input file and normalized
    frequency. Plot the normalized audio signal. Take Fourier transform to plot
    and analyze magnitude spectrum of the same.
clc;
close all;
clear all;

%importing the audio file
[aud_File, sampling_freq] = audioread('sample_sound.wav');
%Finding the length of the audio
length_audio = length(aud_File);
%defining the range of the frequency
freq= (0:(length_audio-1))*sampling_freq/length(aud_File);
norm_freq = freq.*(length_audio)/((length_audio-1)*sampling_freq); %normalized
    frequency= frequency/sampling frequency
%FFT of the signal
fft_signal = fft(aud_File,length_audio);
abs_fft = abs(fftshift(fft_signal)); %taking the absolute value of the shifted
    zero-frequency component to center of spectrum

%plotting the normalized signal
figure;
```

```
subplot(211);
plot(norm_freq, aud_File);
title('Signal Processing');
xlabel('Normalized Frequency');%X-axis
ylabel('Signal');%Y-Axis

%plotting the magnitude spectrum of the normalized signal
subplot(212);
plot(norm_freq, abs_fft);
title('FFT Signal');
xlabel('Normalized Frequency');%X-axis
ylabel('Magnitude Spectrum');%Y-Axis
```

(c) Matlab Script for sub-question 2:

```
clc;
close all;
clear all;

%importing the audio file
[aud_File, sampling_freq] = audioread('sample_sound.wav');
%Finding the length of the audio
length_audio = length(aud_File);

%defining the range of the frequency
freq= (0:(length_audio-1))*sampling_freq/length(aud_File);
norm_freq = freq.*(length_audio)/((length_audio-1)*sampling_freq); %normalized frequency= frequency/sampling frequency
%generating a sinosudal noise
range_noise = 0:1/sampling_freq:((length_audio-1)/sampling_freq);
signal_noise = sin(2*pi*50*range_noise);

%Adding noice with the Audio file
noise_final= aud_File + signal_noise.';
%New noise file
audiowrite('Audio_Noise.wav',noise_final,sampling_freq);
%Applying butterworth filter
[b,a] = butter(3, [0.3 0.7],'bandpass');

%Filtering the noise signal
filtered_signal = filter(b,a,noise_final);

%Writing the filtered signal into a new file
audiowrite('Audio_Filtered.wav',filtered_signal,sampling_freq);

%plotting the normalized signal
figure;
subplot(211);
plot(norm_freq, aud_File);
title('Original Audio');
xlabel('Normalized Frequency');%X-axis
ylabel('Signal');%Y-Axis

%plotting the magnitude spectrum of the normalized signal
subplot(212);
plot(norm_freq, filtered_signal,'r');
title('Audio Filtered');
xlabel('Normalized Frequency');%X-axis
ylabel('Filtered Signal');%Y-Axis
```

(d) Simulation Output for sub-question 1:



(e) Simulation Output for sub-question 2: