Digital Signal Processing (Simulation Assignment 2)

Group Members -

1. Anish Mondal (18EC30047)
2. Himadri Saha (18EC30016)
3. Santanu Kundu (18EC10052)

Problem Statement

Part A

1. Get speech recording of combination lock number “two-four-three-nine-one-seven-five” (8 KHz sampling rate) (i)uttered by two boys and two girls registered in this course at (ii) two different sessions -early morning immediately after waking up and during lunch break and in each session (iii) 3 utterances are to be recorded from each. That is, you will have 2 x 2 x 2 x 3 = 24 speech files. The class representative may share these speech files to all workgroups.

2. Use MATLAB’s audioreador wavereadcommand to read the speech data and note the spectrum of these speech files. Is any similarity or dissimilarity among different groups / subgroups visually identifiable?

3. Pass the signal through equi-spaced filters covering the speech spectrum i.e. if 10 filters are used then 0-400 Hz (LPF), 400-800 Hz (BPF), 800-1200 Hz (BPF) … etc. can be considered and calculate the energy of each filter output. For 10 filters, it will have 10 values. Comment on their similarity / dissimilarity? In this study, you can play with window type of filter and for similarity / dissimilarity calculation, you may use Euclidean distance.

4. Increase the number of filter banks in and see itseffect on similarity / dissimilarity.

5. Make half the filters equally spaced between 0-1000 Hz and the rest between 1000-4000 Hz. Repeat the exercise to see its effect on similarity / dissimilarity.

Part B

B. What is pitch of speech signal? Can you find from published literature a pitch extraction algorithm that makes sense to you? Can you use it in previous study?

Part A

First we try to obtain the length of the largest audio file so that it can be used as a fixed length and all other audio files are padded with zeros so that they can be stored in a single matrix.

```
clearvars;
%setting the path to the audio files
addpath('D:\6th Sem\DSP\DSP_Recordings\');
path = 'D:\6th Sem\DSP\DSP_Recordings\';

number = 4;
fname = "rudra-afternoon (4).wav";
inputFileName = char(strcat(path,fname));
[audio_file, Fs] = audioread(inputFileName,'native');
```

```
max_length = length(audio_file)
```

```
max_length = 56138
```

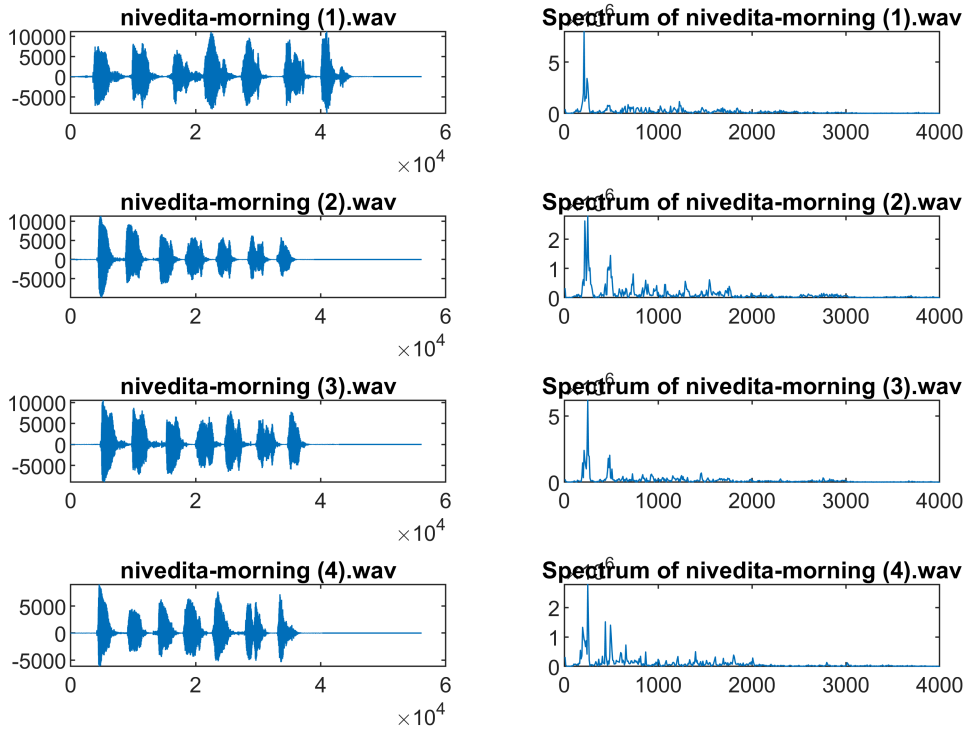
Here we extract the voice samples provided, pad zeros and then enter them into a matrix X.

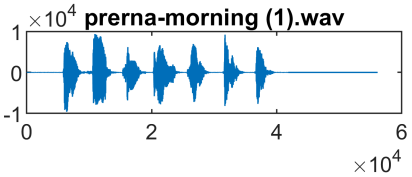
```
% extracting the sounds of all 4 people
element = 1;
name_array = ["nivedita","prerna","rudra","swarnava"];
time_array = ["morning","afternoon"];

NAME = [];

for time = 1:length(time_array)
    for name = 1:length(name_array)
        for i = 1:number
            %generating the name of the file using a for loop and then
            %obtaining it using 'audioread' function
            fname = strcat(name_array(name),'-',time_array(time),' (',num2str(i),').wav');
            inputFileName = char(strcat(path,fname));
            [audio_file, Fs] = audioread(inputFileName,'native');

            audio_file = audio_file';
            audio_file = double(audio_file);
            %audio_file = audio_file./max(audio_file);

            %zero padding code
            if(length(audio_file)<max_length)
                audio_file = padarray(audio_file,[0 (max_length-length(audio_file))],0,'post');
            else
                audio_file = audio_file(1:max_length);
            end

            %entering the sampled audio file in matrix X as well as the
            %name of the file in matrix NAME.
            X(element,:) = audio_file;
            NAME = [NAME fname];

            element = element + 1;
        end
    end
end
```

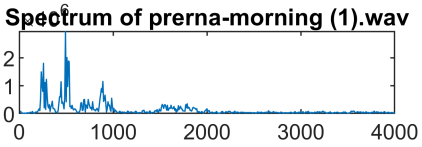
In this part we veiw the sound files as well as their frequency spectrum.

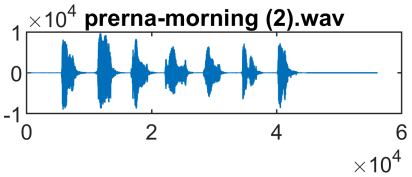
```
Nf = 512; %for 512-point spectrum
Y = zeros(32, Nf); %stores all spectrum
freq_array = (4000/Nf)*linspace(1,Nf,Nf);

for i = 1:8
```

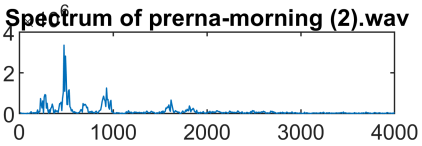
```
    figure
    for j = 1:4
        subplot(4,2,2*j-1)
        plot(X((i-1)*4+j, :));
        title(NAME((i-1)*4+j));

        Y((i-1)*4+j, :) = freqz(double(X((i-1)*4+j,:)),1);
        subplot(4, 2, 2*j)
        plot(freq_array,abs(Y((i-1)*4+j, :)));
        title(strcat("Spectrum of ", NAME((i-1)*4+j)));
    end
end
```

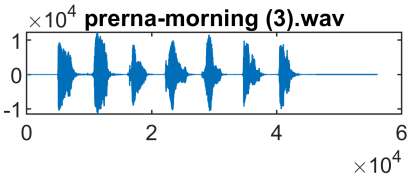


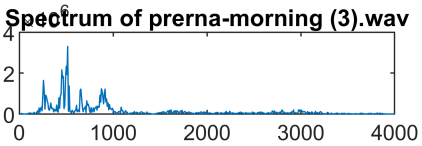

prerna-morning (1).wav

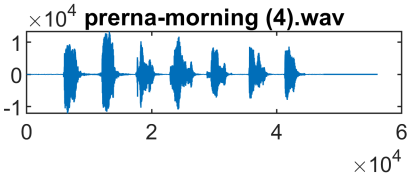

Spectrum of prerna-morning (1).wav

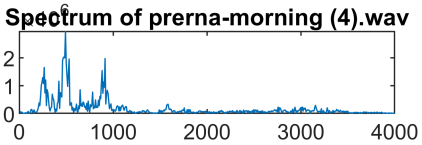

prerna-morning (2).wav

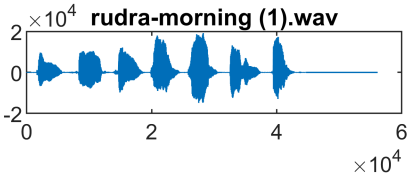

Spectrum of prerna-morning (2).wav

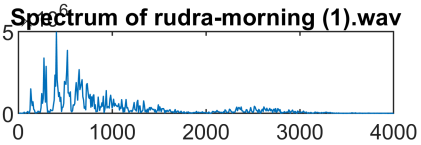

prerna-morning (3).wav

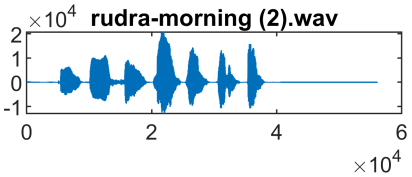

Spectrum of prerna-morning (3).wav

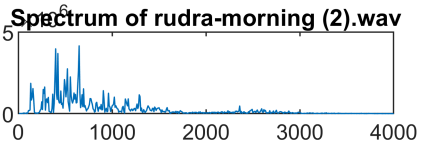

prerna-morning (4).wav

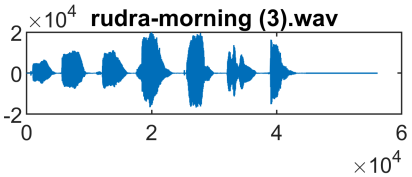

Spectrum of prerna-morning (4).wav

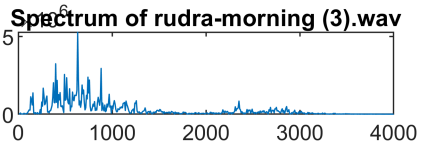

rudra-morning (1).wav

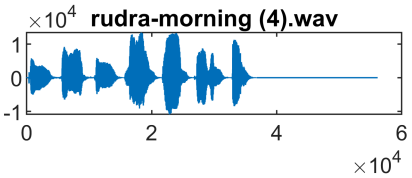

Spectrum of rudra-morning (1).wav

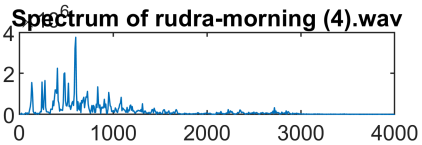

rudra-morning (2).wav

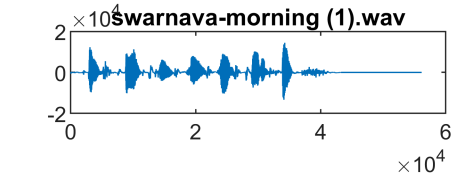

Spectrum of rudra-morning (2).wav

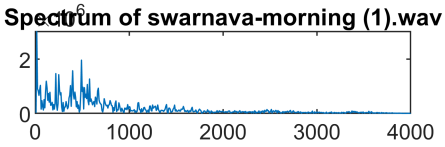

rudra-morning (3).wav

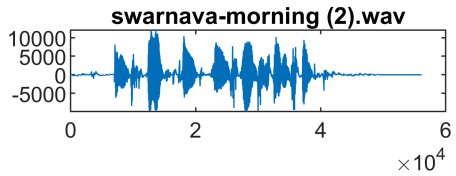

Spectrum of rudra-morning (3).wav

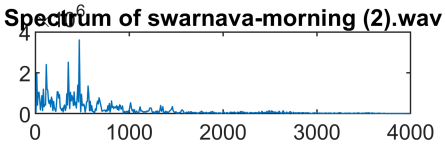

rudra-morning (4).wav

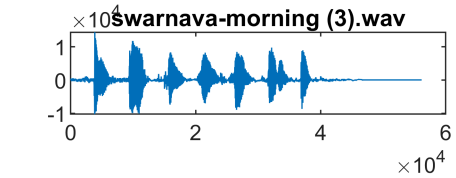

Spectrum of rudra-morning (4).wav

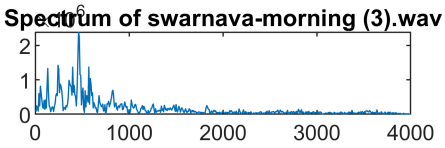

swarnava-morning (1).wav
×10^4
2
0
-2
0
2
4
6
×10^4

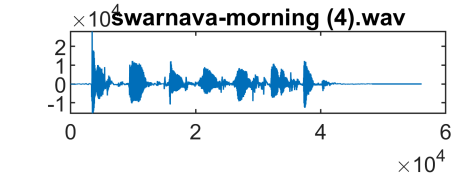

Spectrum of swarnava-morning (1).wav
×10^6
2
0
0
1000
2000
3000
4000

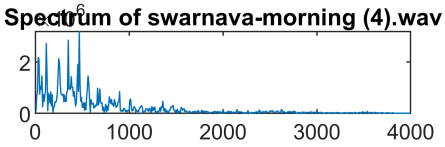

swarnava-morning (2).wav
10000
5000
0
-5000
0
2
4
6
×10^4

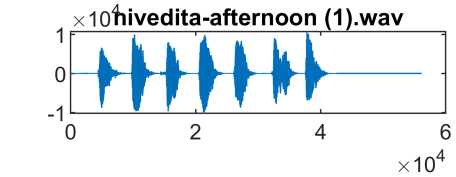

Spectrum of swarnava-morning (2).wav
×10^6
4
2
0
0
1000
2000
3000
4000

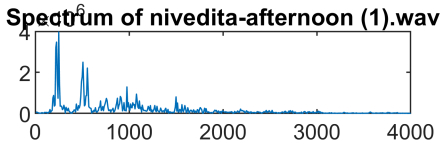

swarnava-morning (3).wav
×10^4
1
0
-1
0
2
4
6
×10^4

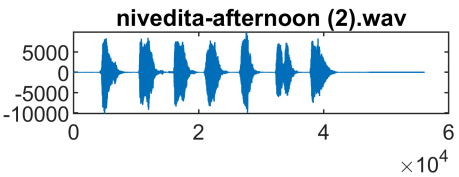

Spectrum of swarnava-morning (3).wav
×10^6
2
1
0
0
1000
2000
3000
4000

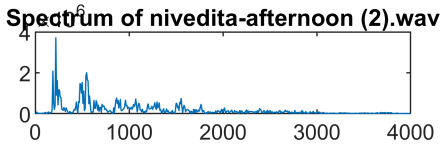

swarnava-morning (4).wav
×10^4
2
1
0
-1
0
2
4
6
×10^4

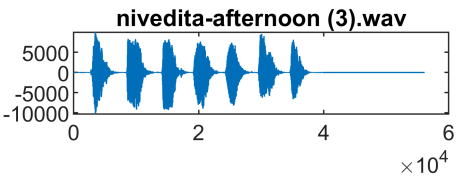

Spectrum of swarnava-morning (4).wav
×10^6
2
0
0
1000
2000
3000
4000

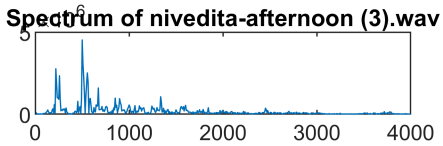

nivedita-afternoon (1).wav
×10^4
1
0
-1
0
2
4
6
×10^4

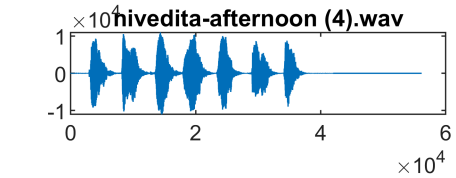

Spectrum of nivedita-afternoon (1).wav
×10^6
4
2
0
0
1000
2000
3000
4000

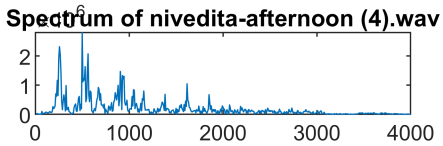

nivedita-afternoon (2).wav
5000
0
-5000
-10000
0
2
4
6
×10^4

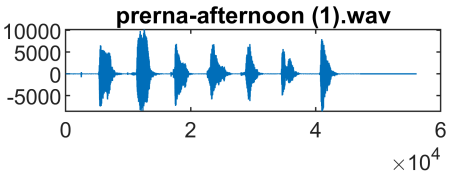

Spectrum of nivedita-afternoon (2).wav
×10^6
4
2
0
0
1000
2000
3000
4000

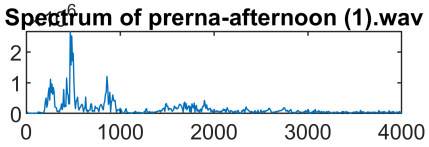

nivedita-afternoon (3).wav
5000
0
-5000
-10000
0
2
4
6
×10^4

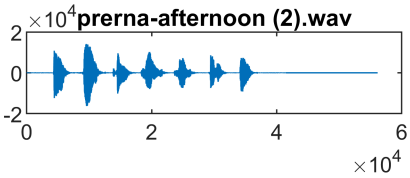

Spectrum of nivedita-afternoon (3).wav
×10^6
5
0
0
1000
2000
3000
4000

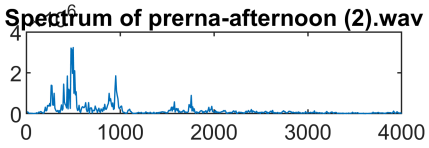

nivedita-afternoon (4).wav
×10^4
1
0
-1
0
2
4
6
×10^4

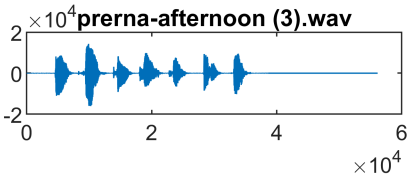

Spectrum of nivedita-afternoon (4).wav
×10^6
2
1
0
0
1000
2000
3000
4000

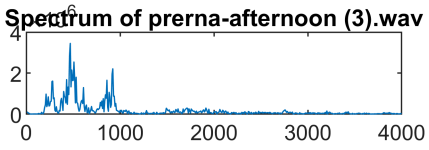

prerna-afternoon (1).wav
10000
5000
0
-5000
0
2
4
6
×10^4

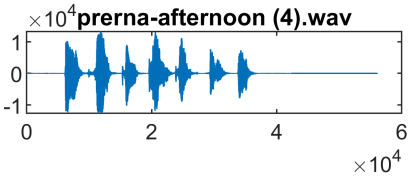

Spectrum of prerna-afternoon (1).wav
×10^6
2
1
0
0
1000
2000
3000
4000

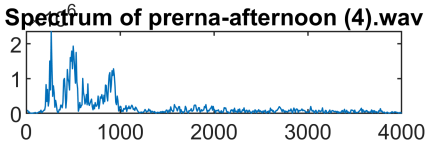

×10^4 prerna-afternoon (2).wav
2
0
-2
0
2
4
6
×10^4

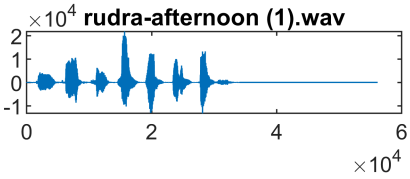

Spectrum of prerna-afternoon (2).wav
×10^6
4
2
0
0
1000
2000
3000
4000

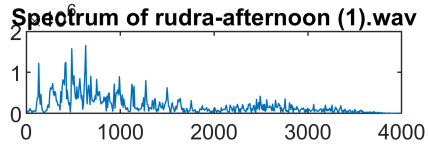

×10^4 prerna-afternoon (3).wav
2
0
-2
0
2
4
6
×10^4

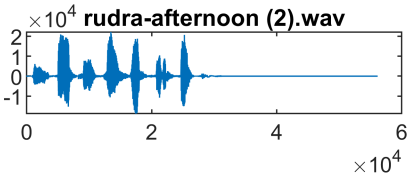

Spectrum of prerna-afternoon (3).wav
×10^6
4
2
0
0
1000
2000
3000
4000

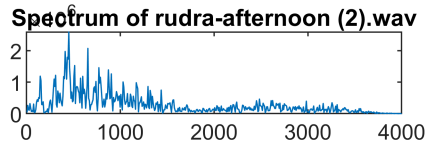

×10^4 prerna-afternoon (4).wav
1
0
-1
0
2
4
6
×10^4

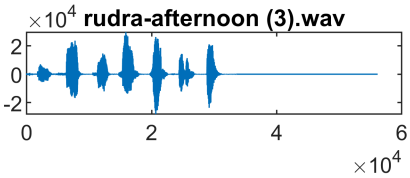

Spectrum of prerna-afternoon (4).wav
×10^6
2
1
0
0
1000
2000
3000
4000

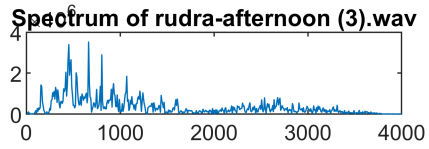

×10^4 rudra-afternoon (1).wav
2
1
0
-1
0
2
4
6
×10^4

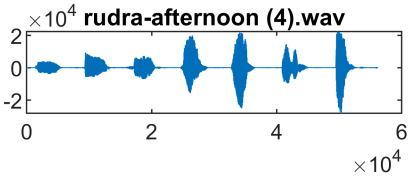

Spectrum of rudra-afternoon (1).wav
×10^6
2
1
0
0
1000
2000
3000
4000

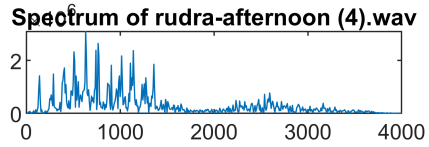

×10^4 rudra-afternoon (2).wav
2
1
0
-1
0
2
4
6
×10^4

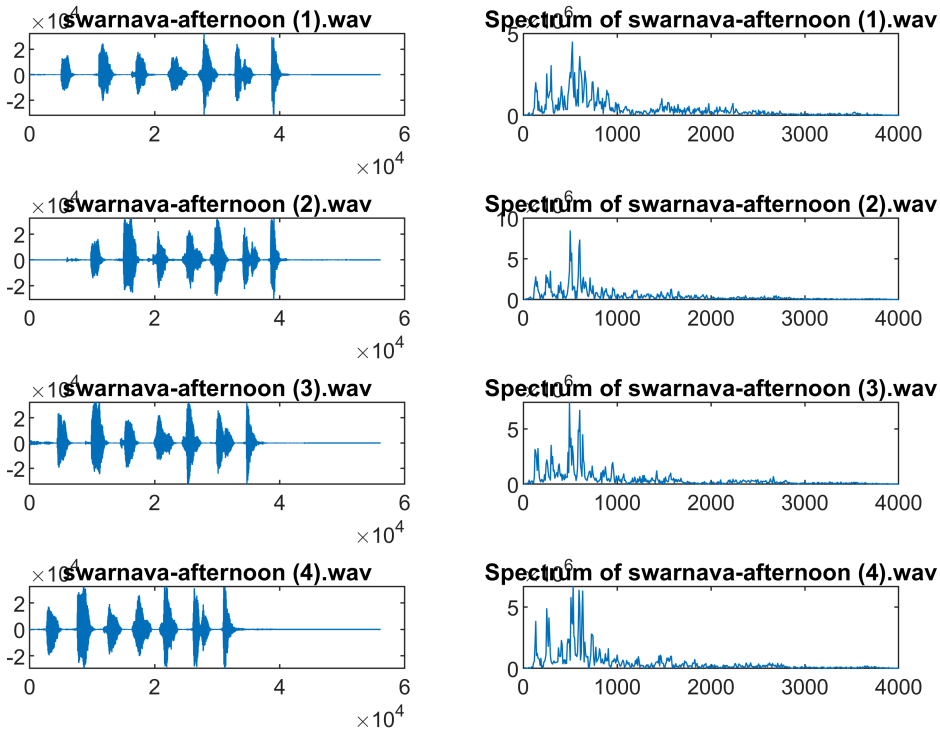

Spectrum of rudra-afternoon (2).wav
×10^6
2
1
0
0
1000
2000
3000
4000


×10^4 rudra-afternoon (3).wav
2
0
-2
0
2
4
6
×10^4


Spectrum of rudra-afternoon (3).wav
×10^6
4
2
0
0
1000
2000
3000
4000


×10^4 rudra-afternoon (4).wav
2
0
-2
0
2
4
6
×10^4

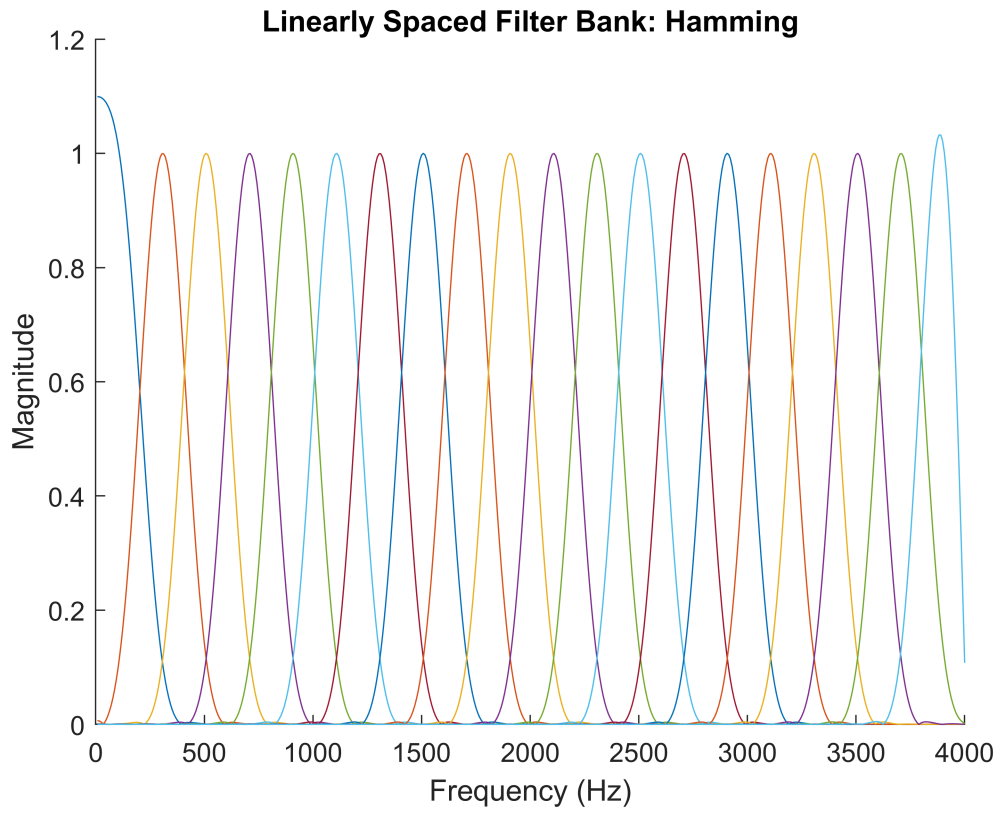

Spectrum of rudra-afternoon (4).wav
×10^6
2
0
0
1000
2000
3000
4000



Linearly Spaced Filter Bank: Hamming Window (Default)

In this part we create the filter bank using a for loop. This filter bank is linearly spaced between 0 and 4000Hz.

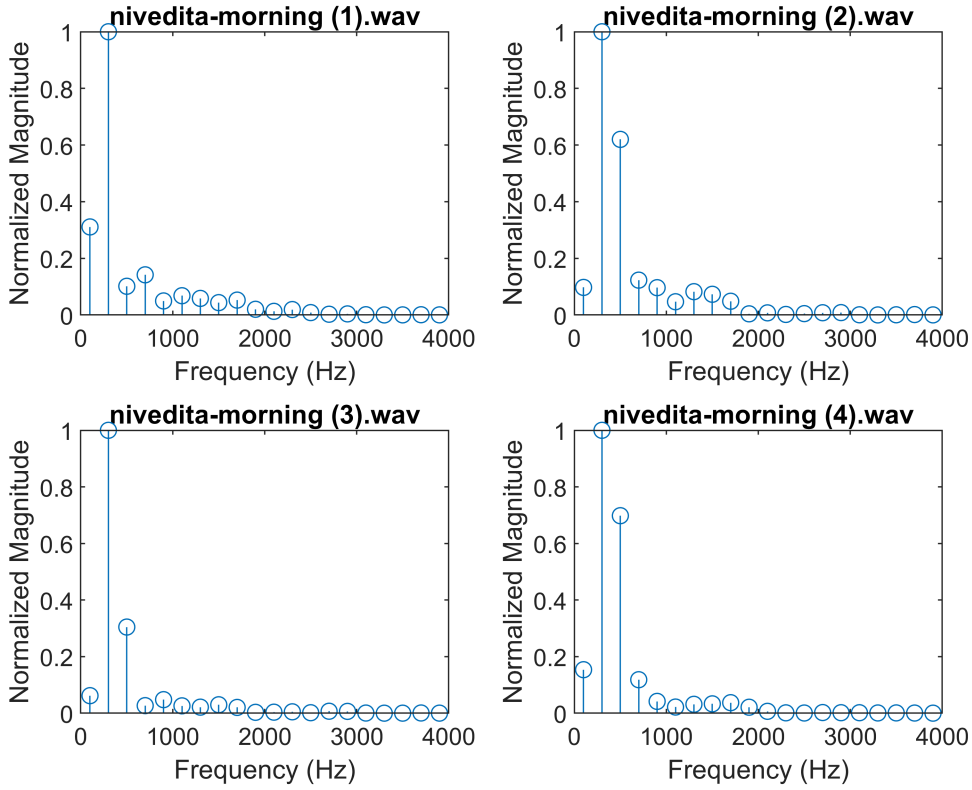
```
%making the filter bank

num_filters = 20;
fl = 1;
fh = 3999;
fs = Fs;
filter_center_freq = zeros(1,num_filters);

%factor = power(fh/fl,1/num_filters);
difference = ((fh)-(fl))/num_filters;

%fac = factor.^(0:num_filters);
diff = difference.*(0:num_filters);
figure
hold on
for i = 1:num_filters
    [b(i,:),a(i,:)] = fir1(75,[((fl + diff(i))/(fs/2)) ((fl + diff(i+1))/(fs/2))],'bandpass');
    H = freqz(b(i,:),a(i,:)); plot(freq_array,abs(H));
    filter_center_freq(i) = fl + (diff(i)+diff(i+1))/2;
end
title('Linearly Spaced Filter Bank: Hamming')
xlabel('Frequency (Hz)')
ylabel('Magnitude')
hold off
```

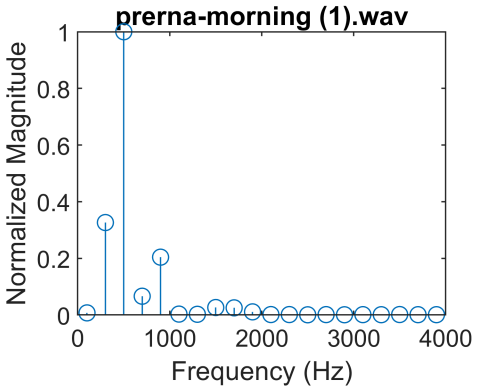


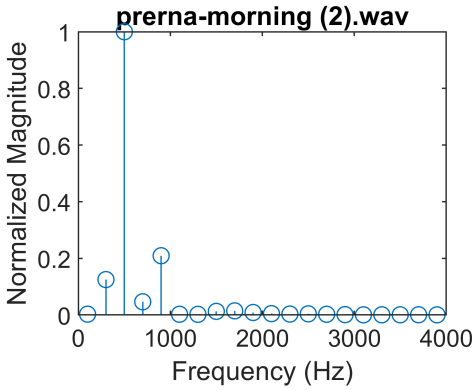
We pass the sound signals through the filter bank and then take the power of the output of each filter. The filter outputs of each sound signal is normalized by the power of the filter bank with maximum value and is then plotted so that the spectral distribution can be uniformly compared.

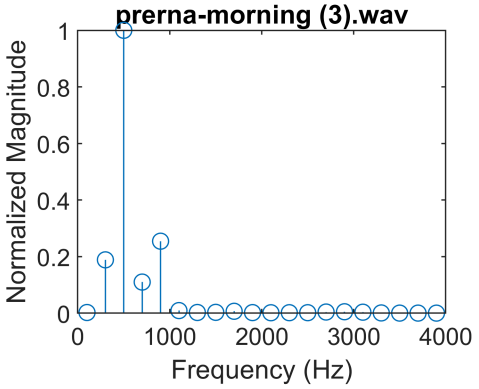
```
X = double(X);
psd = zeros(size(X,1),num_filters);

for j = 1:size(X,1)
    for i = 1:num_filters
        X_bandpassed(i,:) = filtfilt(b(i,:),a(i,:), X(j,:));
    %     subplot(211)
    %     plot(X_bandpassed(i,:))

        H = freqz(double(X_bandpassed(i,:)),1);

        %subplot(212)
        %plot(freq_array,abs(H))
        psd(j,i) = sum(abs(H).^2);
    end
    figure
%     stem(psd(j,:));
%     title(NAME(j))
end

%normalization code
psd_max = max(psd,[],2);
```

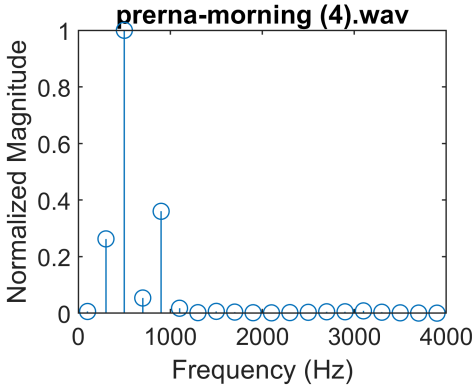
```
psd_normalized = psd./psd_max;

%plotting the calculated PSD
for i = 1:8
    figure
    for j = 1:4
        subplot(2,2,j)

        %stem(filter_center_freq,psd(4*(i-1)+j,:)/psd_max(4*(i-1)+j));
        stem(filter_center_freq,psd_normalized(4*(i-1)+j,:));
        title(NAME(4*(i-1)+j))
        xlabel("Frequency (Hz)")
        ylabel("Normalized Magnitude")
    end
end
```

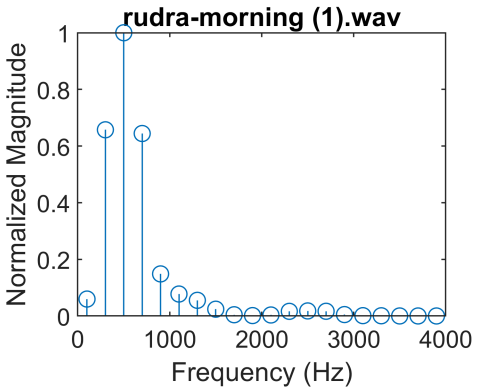


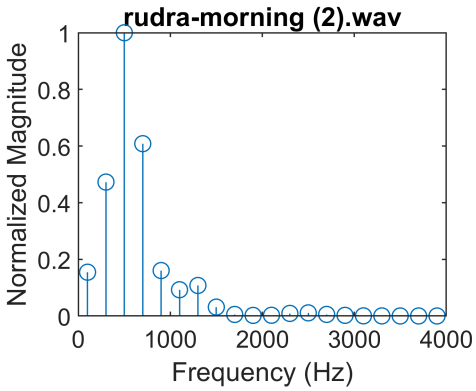

prerna-morning (1).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

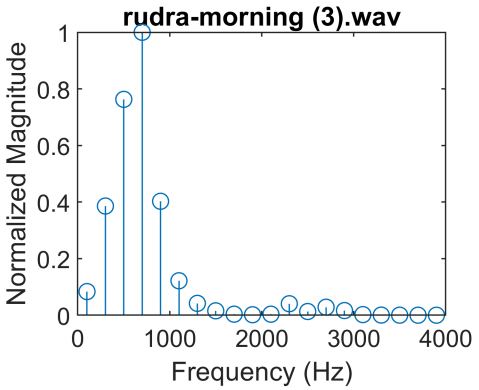

prerna-morning (2).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

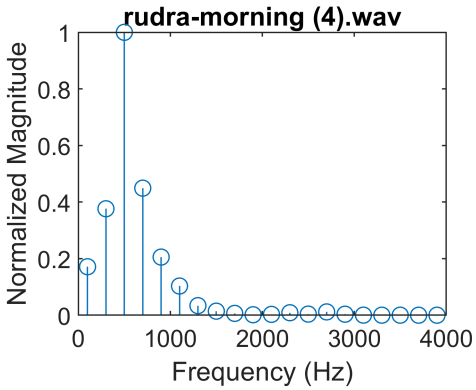

prerna-morning (3).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

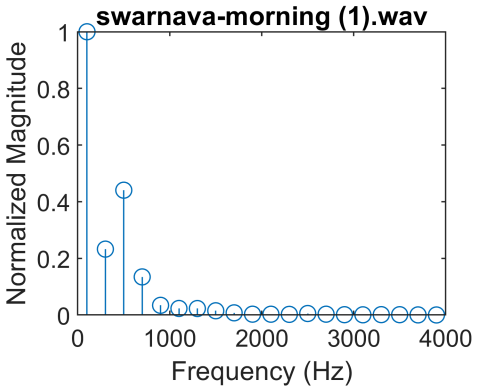

prerna-morning (4).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

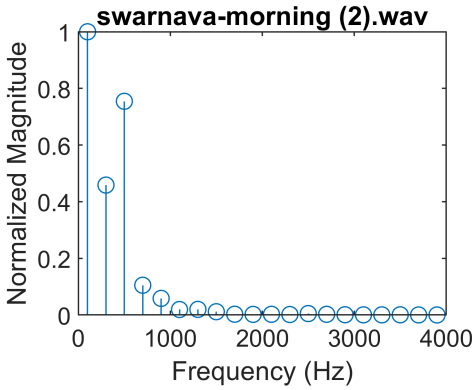

rudra-morning (1).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

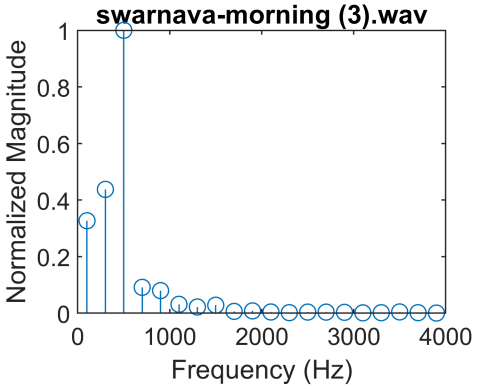

rudra-morning (2).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

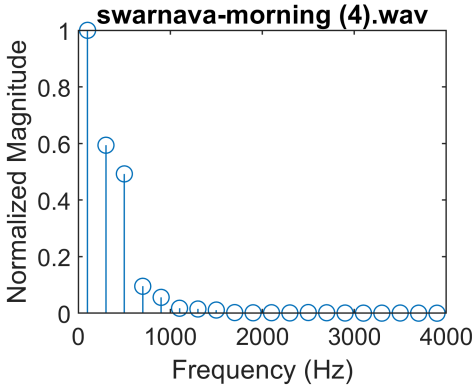

rudra-morning (3).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

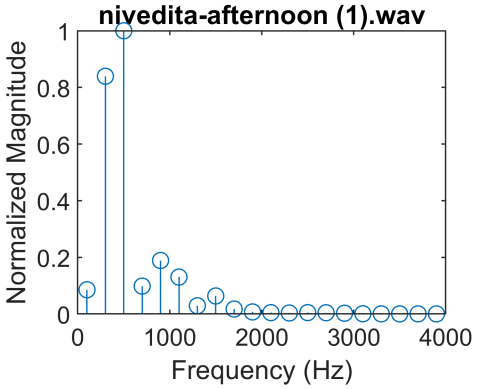

rudra-morning (4).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

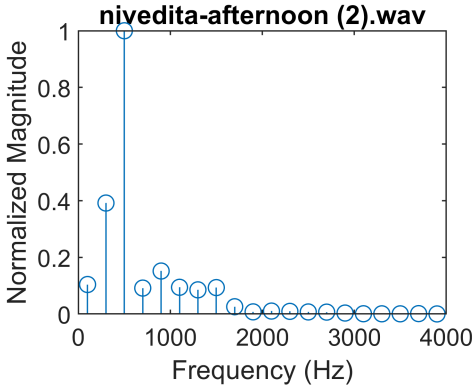

swarnava-morning (1).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

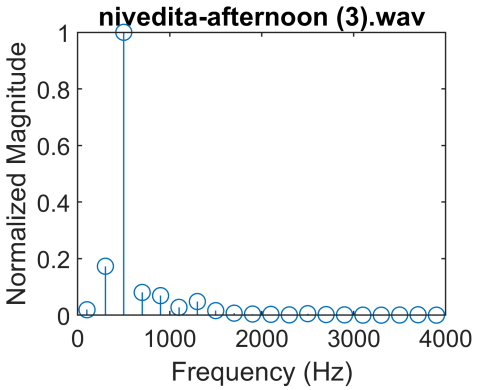

swarnava-morning (2).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

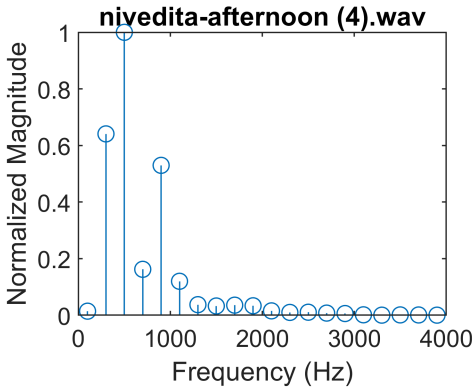

swarnava-morning (3).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

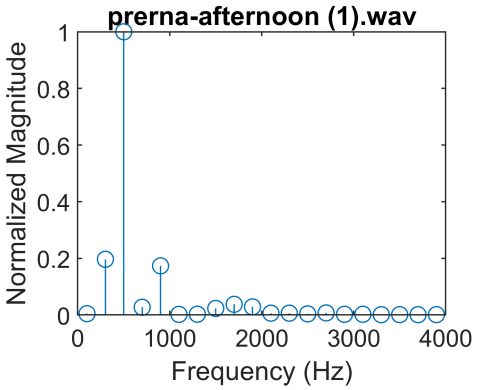

swarnava-morning (4).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

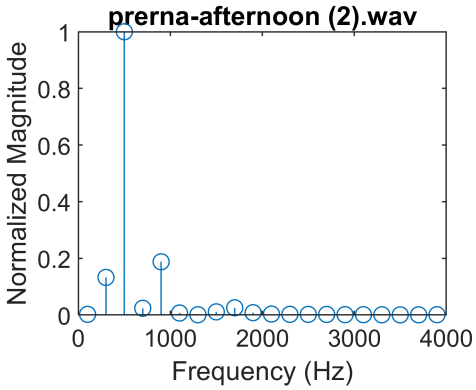

nivedita-afternoon (1).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

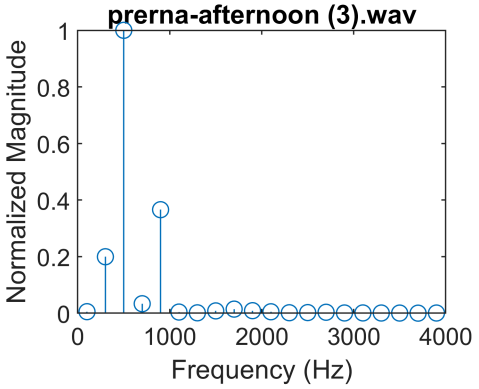

nivedita-afternoon (2).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

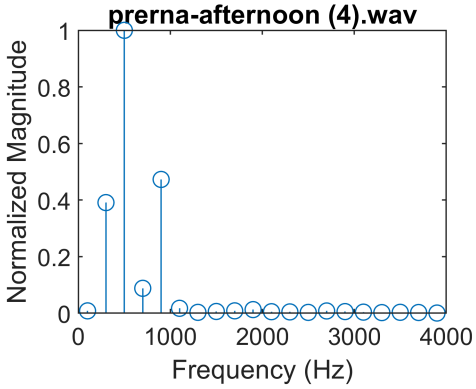

nivedita-afternoon (3).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

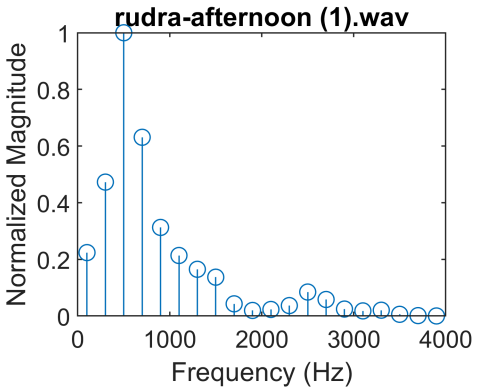

nivedita-afternoon (4).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

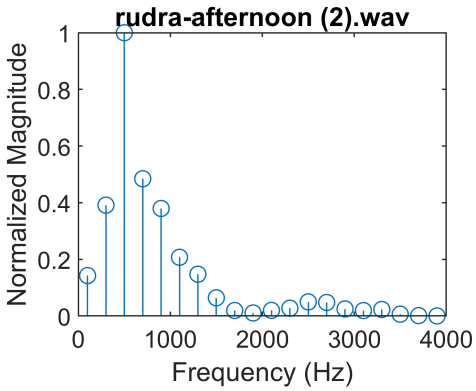

prerna-afternoon (1).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

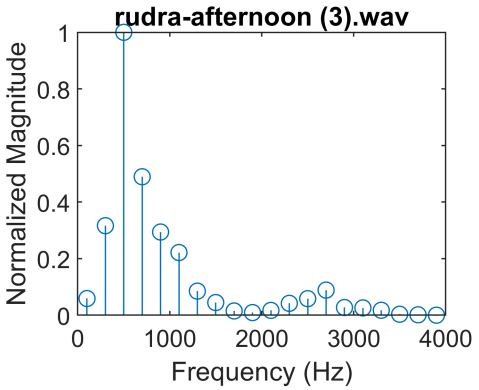

prerna-afternoon (2).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

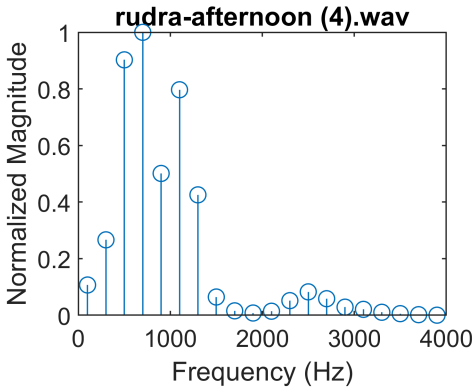

prerna-afternoon (3).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

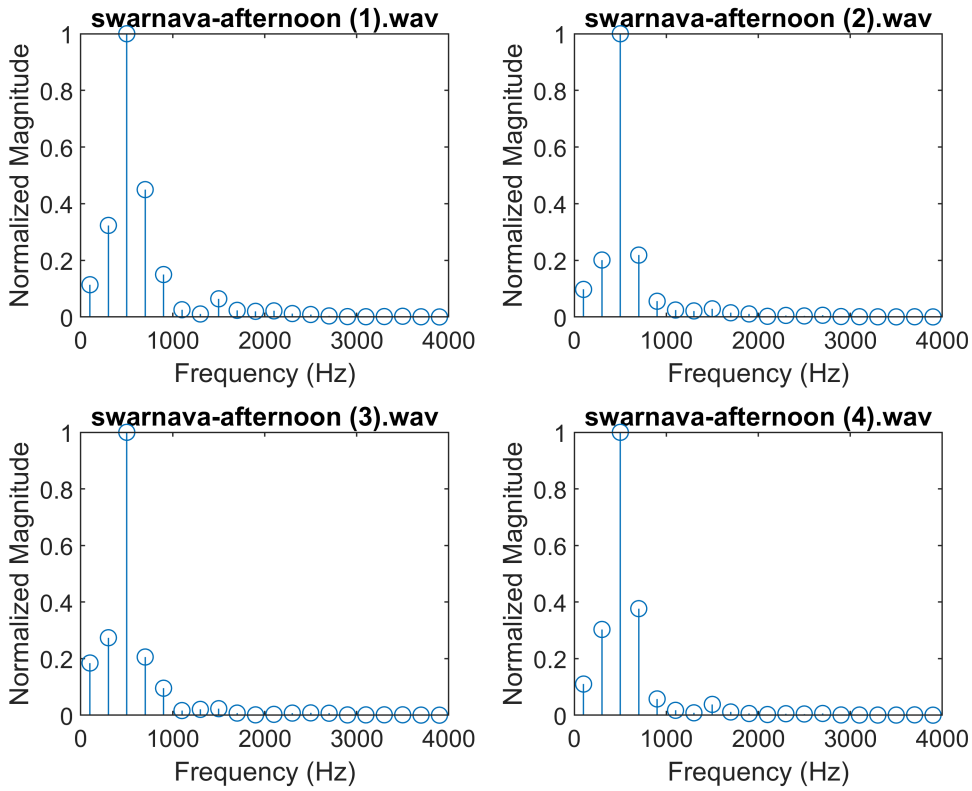

prerna-afternoon (4).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000


rudra-afternoon (1).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000


rudra-afternoon (2).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000


rudra-afternoon (3).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

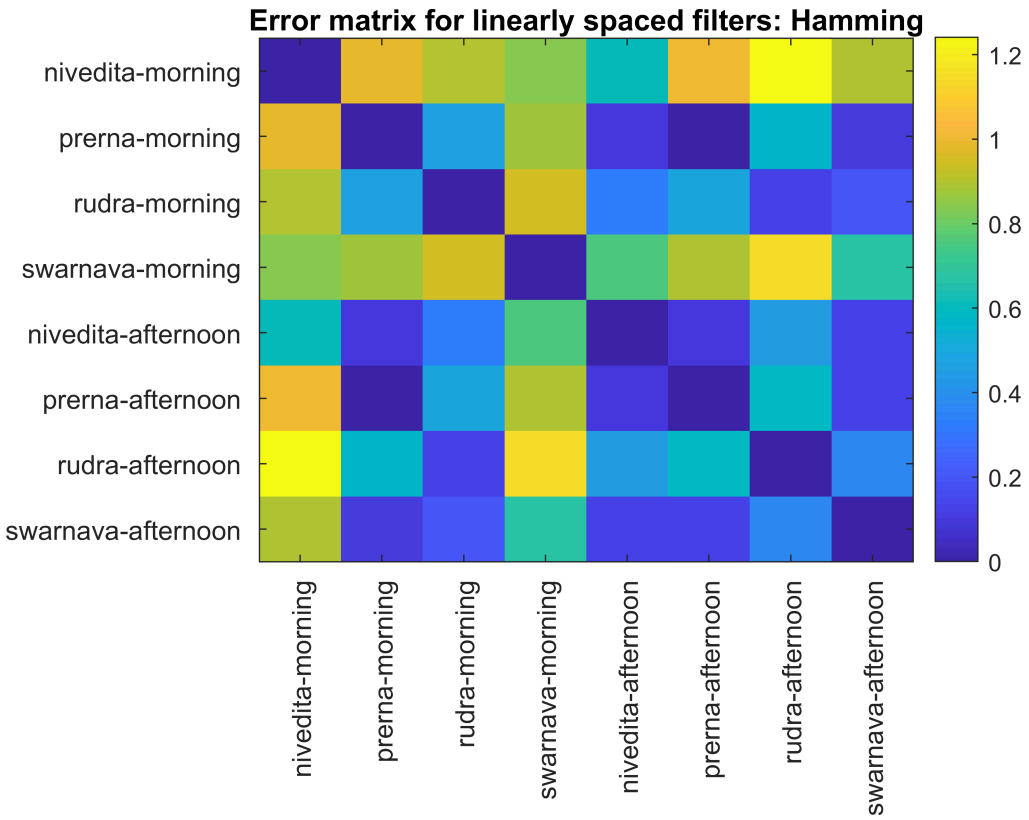

rudra-afternoon (4).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000



In this part we measure the similarity / dissimilarity between the voices of our 4 volunteers voices, in both morning and afternoon. In order to to simplify the plot we have taken the average spectrum of the 4 samples (which can be considered as a characteristic PSD for a particular person at a particular time.

```
m_s_error = zeros(size(X,1)/4);
psd_average = zeros(size(psd,1)/4,size(psd,2));
for i = 1:8
    sum1 = zeros(1,size(psd,2));
    for j = 1:4
        sum1 = sum1 + psd_normalized(4*(i-1)+j,:)/4;
    end
    psd_average(i,:) = sum1;
end

for i = 1:8
    for j = 1:8

        m_s_error(i,j) = sum((psd_average(i,:)-psd_average(j,:)).^2);

    end
end
figure
%imshow(m_s_error)
imagesc(m_s_error)
% xlabel(NAME(1:4:length(NAME)))
% ylabel(NAME(1:4:length(NAME)))
```

```
title("Error matrix for linearly spaced filters: Hamming")

xticks([1:8])
xticklabels(extractBetween(NAME(1:4:length(NAME)),1,strlength(NAME(1:4:length(NAME)))-7));
%xticklabels(NAME(1:4:length(NAME)))
xtickangle(90)
yticks([1:8])
%yticklabels(NAME(1:4:length(NAME)))
yticklabels(extractBetween(NAME(1:4:length(NAME)),1,strlength(NAME(1:4:length(NAME)))-7));
ytickangle(0)
colorbar
```



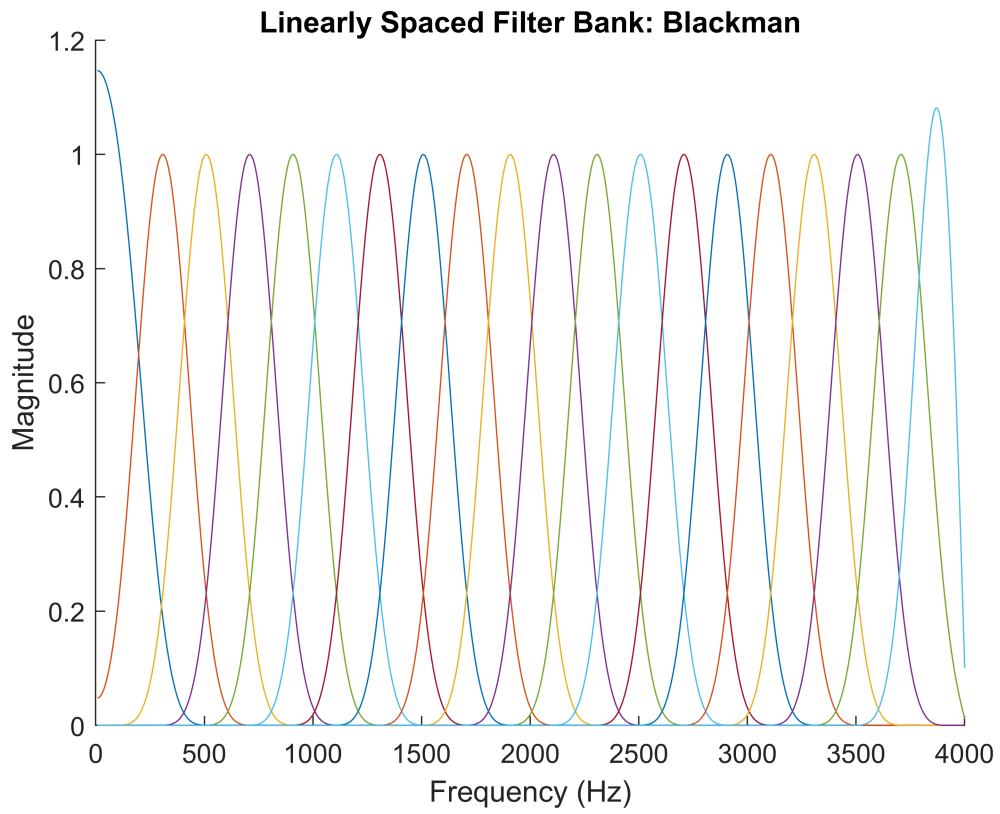
To get the amount of error between two signals we see the sqare corresponding to the row and column of the two signals. The diagonal are 0 as they represent the same signal.

Linearly Spaced Filter Bank with a Different Window: Blackman Window

In this part we create the filter bank using a for loop. This filter bank is linearly spaced between 0 and 4000Hz.

```
%making the filter bank

num_filters = 20;
fl = 1;
fh = 3999;
fs = Fs;
filter_center_freq = zeros(1,num_filters);

%factor = power(fh/fl,1/num_filters);
```

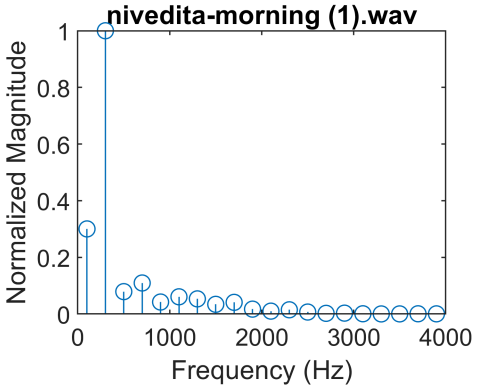
```
difference = ((fh)-(fl))/num_filters;

%fac = factor.^(0:num_filters);
diff = difference.*(0:num_filters);
figure
hold on
for i = 1:num_filters
    [b(i,:),a(i,:)] = fir1(75,[((fl + diff(i))/(fs/2)) ((fl + diff(i+1))/(fs/2))],'bandpass',bl
    H = freqz(b(i,:),a(i,:)); plot(freq_array, abs(H));
    filter_center_freq(i) = fl + (diff(i)+diff(i+1))/2;
end
title('Linearly Spaced Filter Bank: Blackman')
xlabel('Frequency (Hz)')
ylabel('Magnitude')
hold off
```

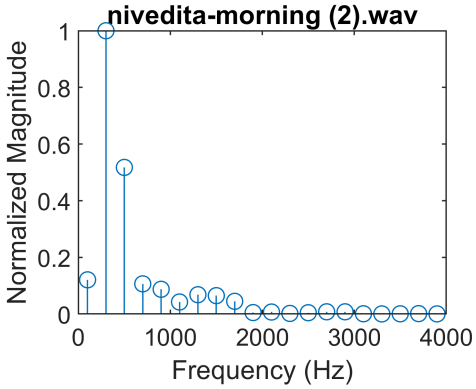


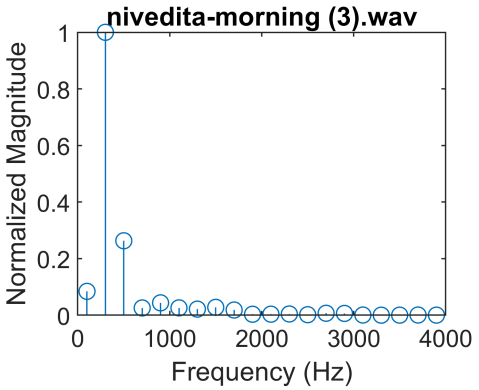
We pass the sound signals through the filter bank and then take the power of the output of each filter. The filter outputs of each sound signal is normalized by the power of the filter bank with maximum value and is then plotted so that the spectral distribution can be uniformly compared.

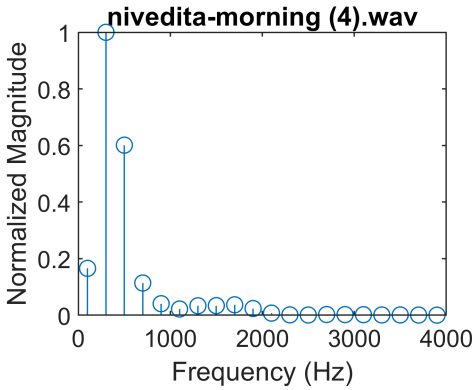
```
X = double(X);
psd = zeros(size(X,1),num_filters);

for j = 1:size(X,1)
    for i = 1:num_filters
        X_bandpassed(i,:) = filtfilt(b(i,:),a(i,:), X(j,:));
    %     subplot(211)
```

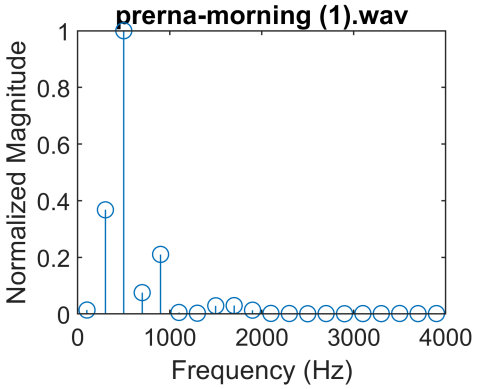
```
    %     plot(X_bandpassed(i,:))

        H = freqz(double(X_bandpassed(i,:)),1);

        %subplot(212)
        %plot(freq_array,abs(H))
        psd(j,i) = sum(abs(H).^2);
    end
    figure
%     stem(psd(j,:));
%     title(NAME(j))
end

%normalization code
psd_max = max(psd,[],2);
psd_normalized = psd./psd_max;

%plotting the calculated PSD
for i = 1:8
    figure
    for j = 1:4
        subplot(2,2,j)

        %stem(filter_center_freq,psd(4*(i-1)+j,:)/psd_max(4*(i-1)+j));
        stem(filter_center_freq,psd_normalized(4*(i-1)+j,:));
        title(NAME(4*(i-1)+j))
        xlabel("Frequency (Hz)")
        ylabel("Normalized Magnitude")
    end
end
```

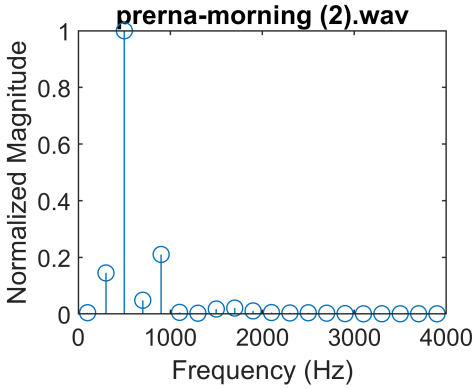

nivedita-morning (1).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

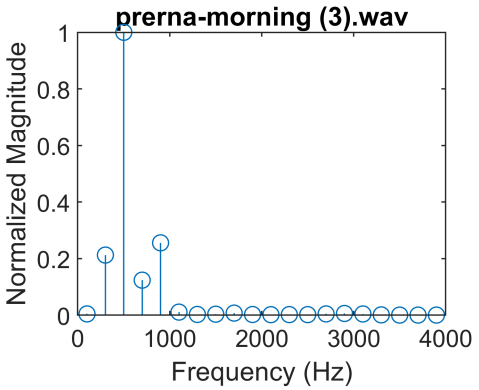

nivedita-morning (2).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

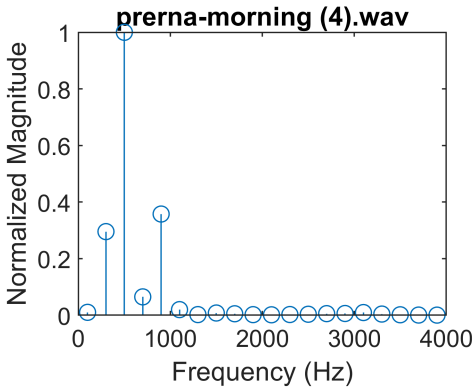

nivedita-morning (3).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

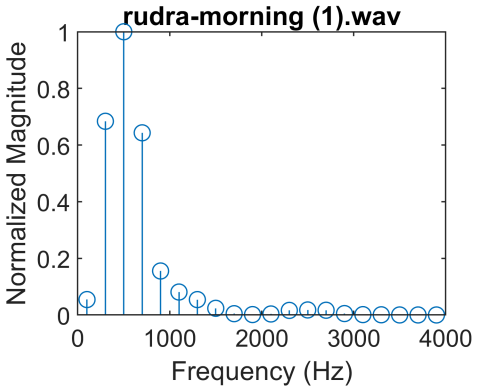

nivedita-morning (4).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

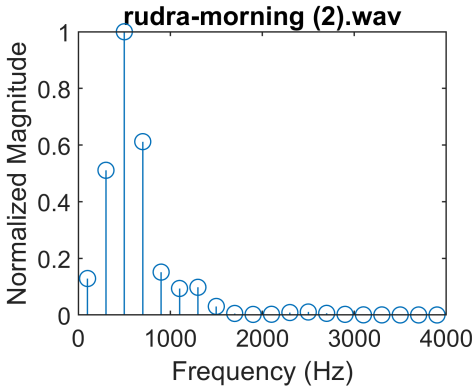

prerna-morning (1).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

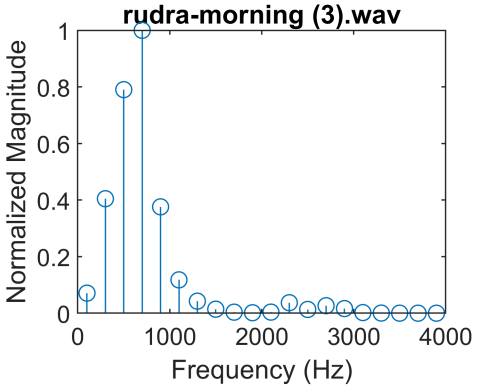

prerna-morning (2).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

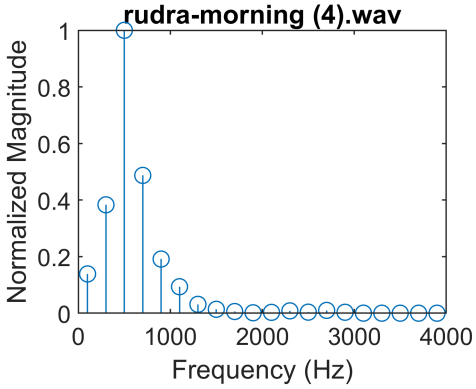

prerna-morning (3).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

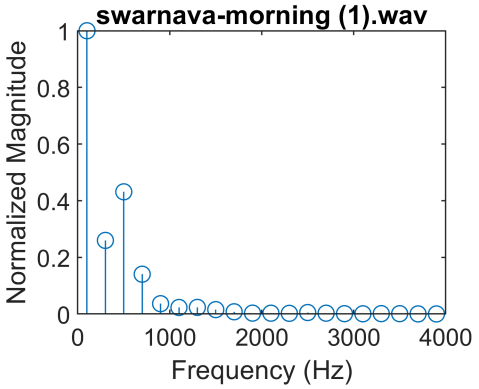

prerna-morning (4).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

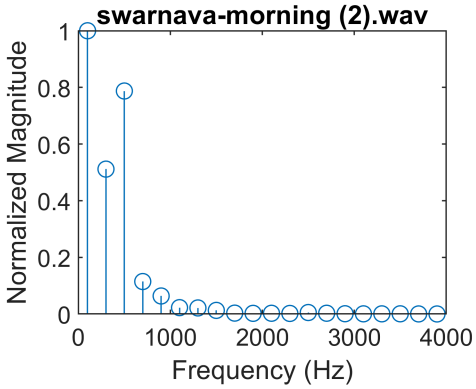

rudra-morning (1).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

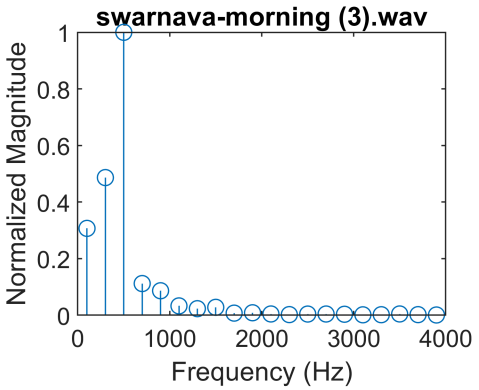

rudra-morning (2).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

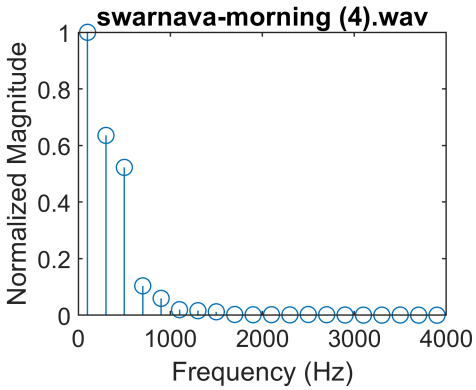

rudra-morning (3).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

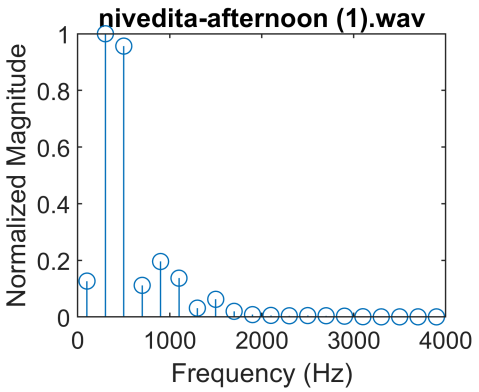

rudra-morning (4).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

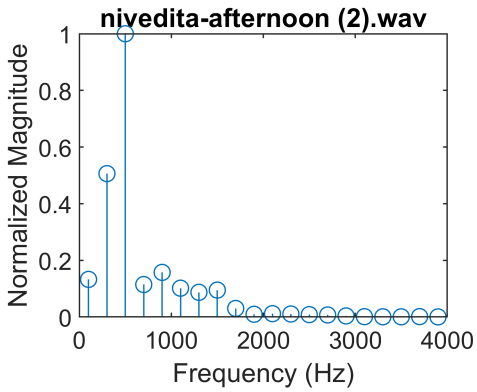

swarnava-morning (1).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

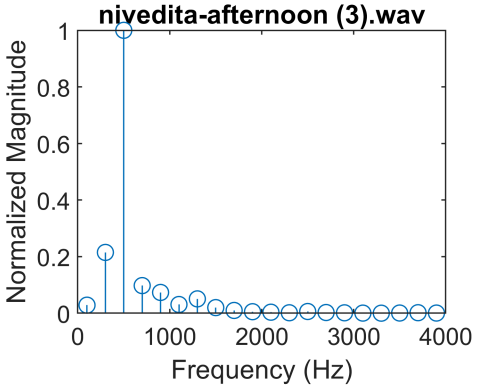

swarnava-morning (2).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

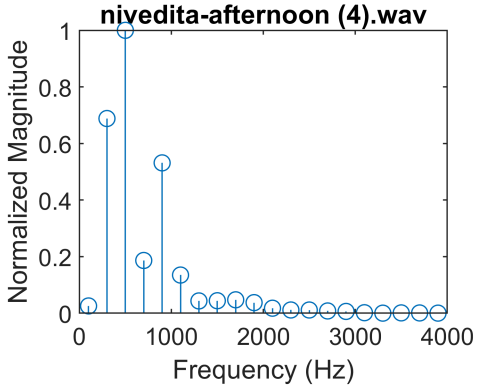

swarnava-morning (3).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

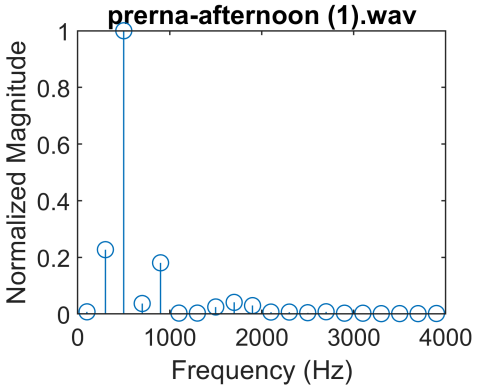

swarnava-morning (4).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

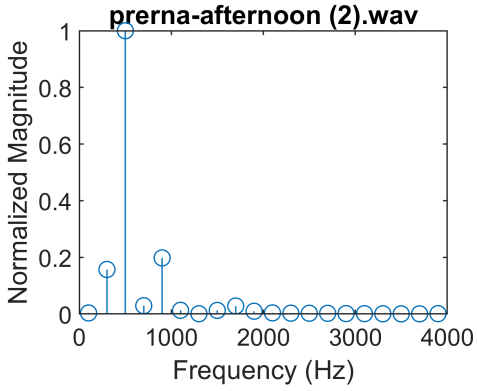

nivedita-afternoon (1).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

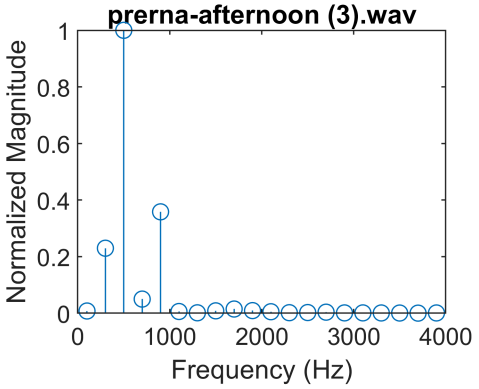

nivedita-afternoon (2).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

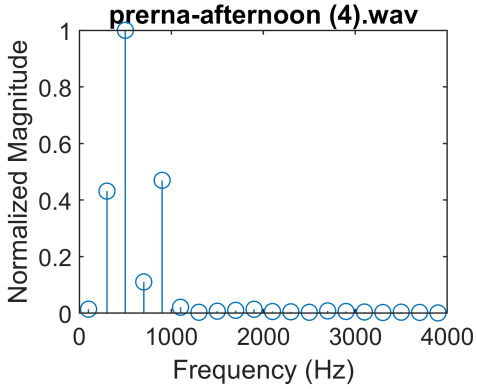

nivedita-afternoon (3).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

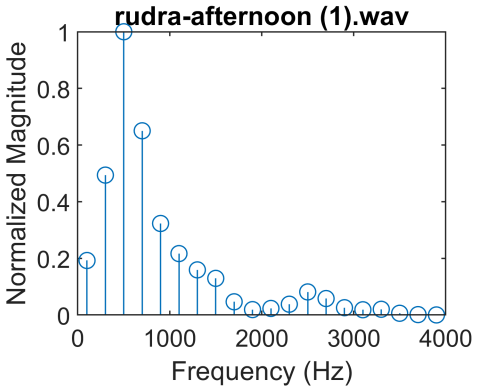

nivedita-afternoon (4).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

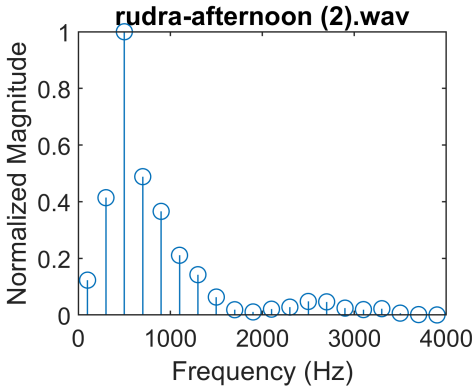

prerna-afternoon (1).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

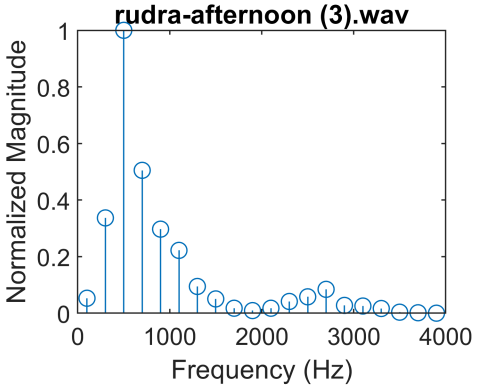

prerna-afternoon (2).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

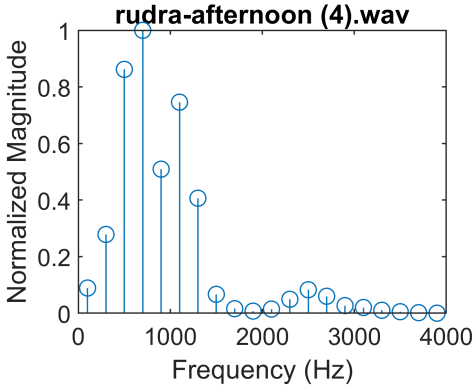

prerna-afternoon (3).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

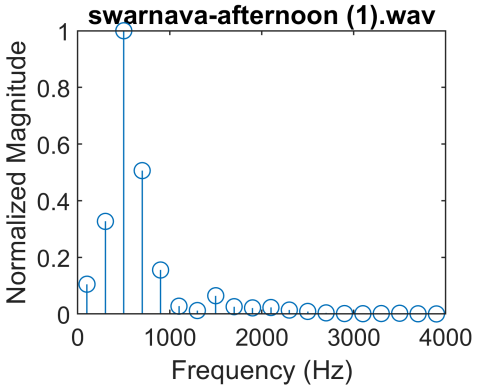

prerna-afternoon (4).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

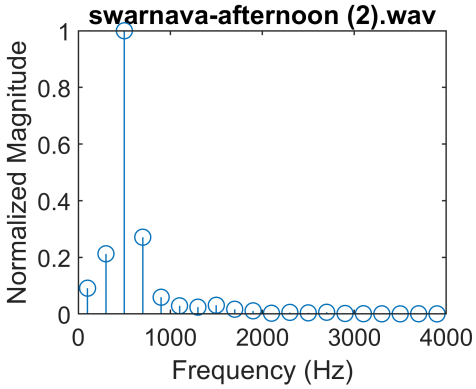

rudra-afternoon (1).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

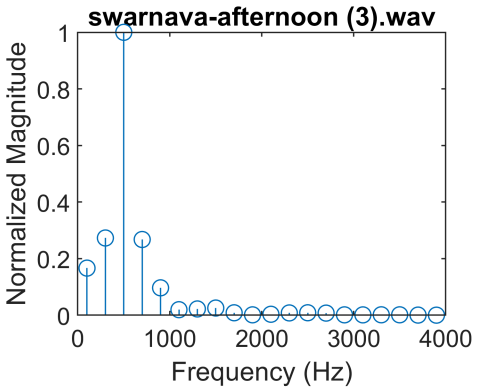

rudra-afternoon (2).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

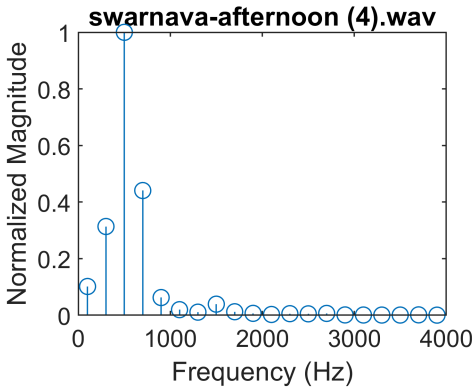

rudra-afternoon (3).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000


rudra-afternoon (4).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000


swarnava-afternoon (1).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

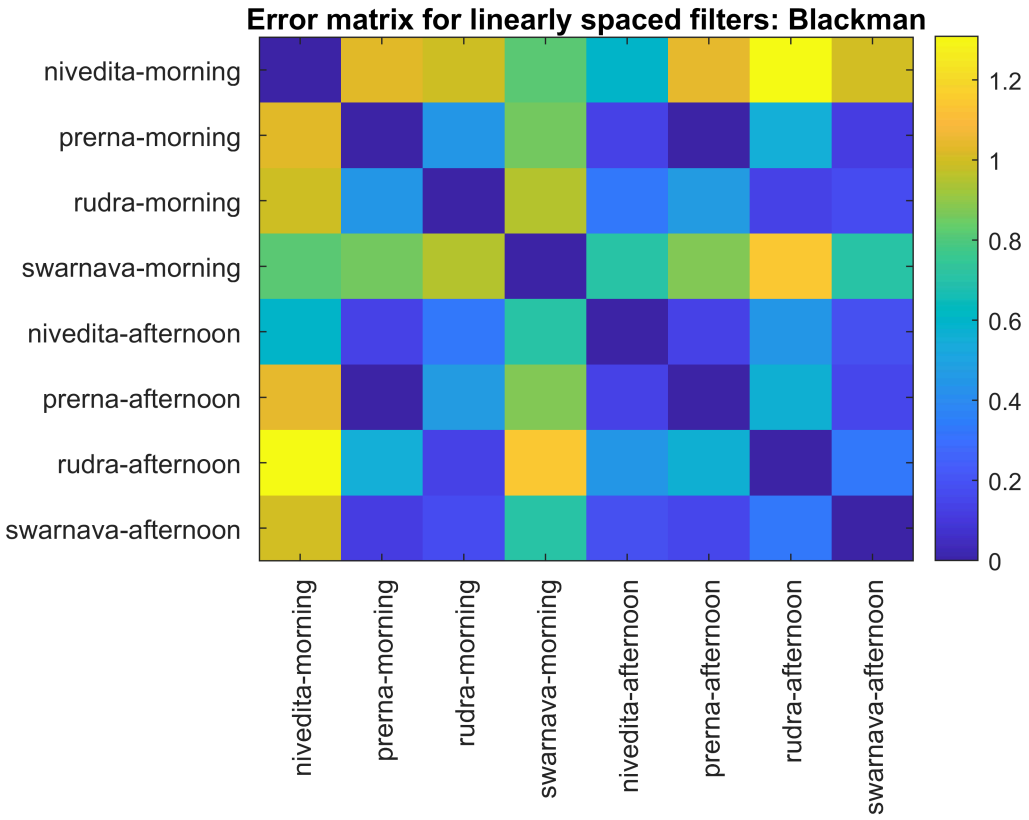

swarnava-afternoon (2).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000


swarnava-afternoon (3).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000


swarnava-afternoon (4).wav
Normalized Magnitude
Frequency (Hz)
0
0.2
0.4
0.6
0.8
1
0
1000
2000
3000
4000

In this part we measure the similarity / dissimilarity between the voices of our 4 volunteers voices, in both morning and afternoon. In order to to simplify the plot we have taken the average spectrum of the 4 samples (which can be considered as a characteristic PSD for a particular person at a particular time.

```
m_s_error = zeros(size(X,1)/4);
psd_average = zeros(size(psd,1)/4,size(psd,2));
for i = 1:8
    sum1 = zeros(1,size(psd,2));
    for j = 1:4
        sum1 = sum1 + psd_normalized(4*(i-1)+j,:)/4;
    end
    psd_average(i,:) = sum1;
end

for i = 1:8
    for j = 1:8

        m_s_error(i,j) = sum((psd_average(i,:)-psd_average(j,:)).^2);

    end
end
figure
%imshow(m_s_error)
imagesc(m_s_error)
% xlabel(NAME(1:4:length(NAME)))
% ylabel(NAME(1:4:length(NAME)))
title("Error matrix for linearly spaced filters: Blackman")

xticks([1:8])
xticklabels(extractBetween(NAME(1:4:length(NAME)),1,strlength(NAME(1:4:length(NAME)))-7));
%xticklabels(NAME(1:4:length(NAME)))
xtickangle(90)
yticks([1:8])
%yticklabels(NAME(1:4:length(NAME)))
yticklabels(extractBetween(NAME(1:4:length(NAME)),1,strlength(NAME(1:4:length(NAME)))-7));
ytickangle(0)
colorbar
```



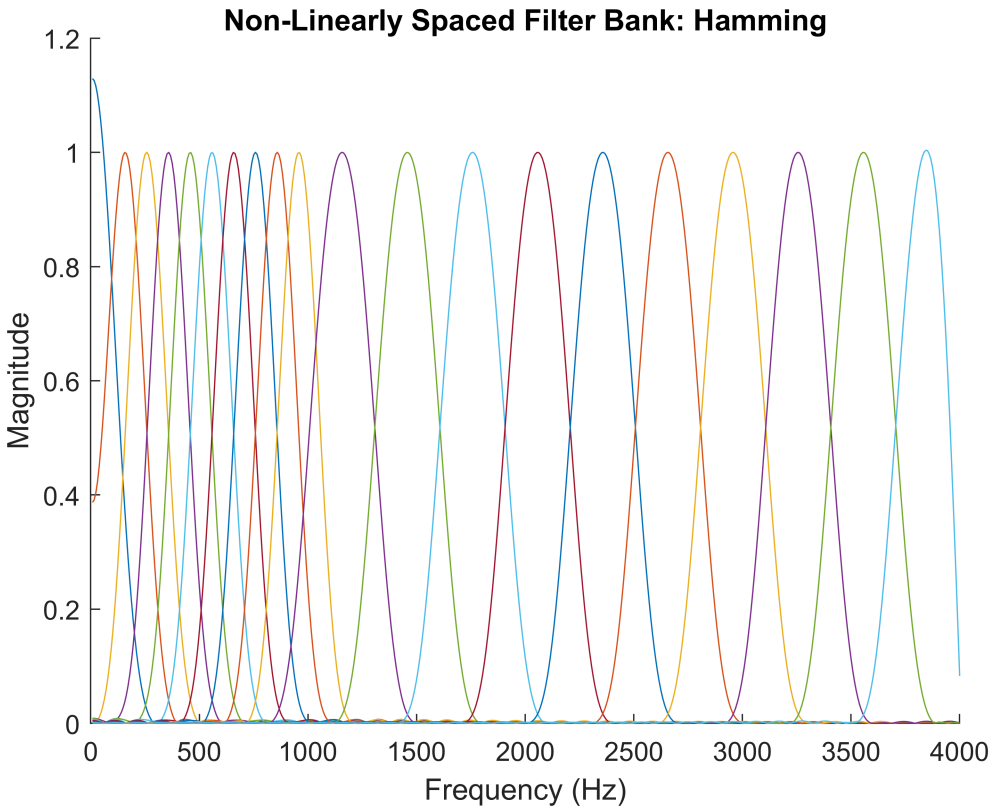
To get the amount of error between two signals we see the sqare corresponding to the row and column of the two signals.

Non-Linearly Spaced Filter Bank: Hamming Window (Default)

Creating non uniform filter bank which has half of the filters in 0 to 1000Hz and the other half in the range 2KHz to 4KHz.

```
%making the filter bank with non linear spacing

num_filters = 20;
fl = 1;
fh = 3999;
fm = 1000;
fs = Fs;

filter_center_freq2 = zeros(1,num_filters);


difference1 = ((fm)-(fl))/(num_filters/2);
difference2 = ((fh)-(fm))/(num_filters/2);


diff1 = difference1.*(0:num_filters/2);
diff2 = difference2.*(0:num_filters/2);

figure
```

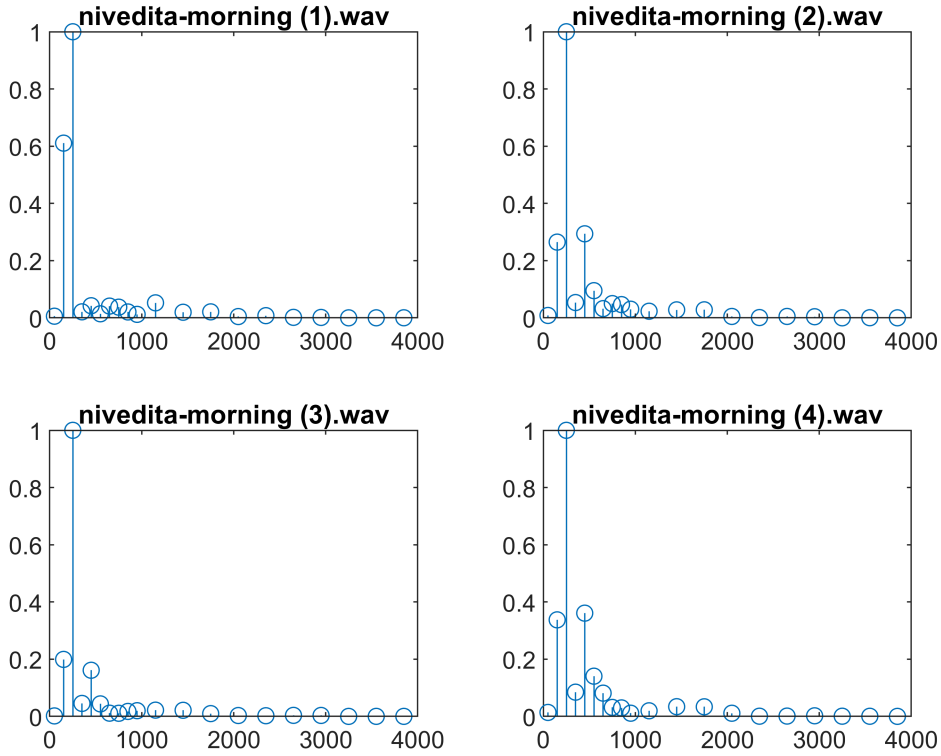
```
hold on

for i = 1:num_filters/2
    [b(i,:),a(i,:)] = fir1(75,[((fl + diff1(i))/(fs/2)) ((fl + diff1(i+1))/(fs/2))],'bandpass')
    H = freqz(b(i,:),a(i,:)); plot(freq_array, abs(H));
    filter_center_freq2(i) = fl + (diff1(i)+diff1(i+1))/2;
end

for i = 1 :num_filters/2
    [b(i+num_filters/2,:),a(i+num_filters/2,:)] = fir1(75,[((fm + diff2(i))/(fs/2)) ((fm + diff
    H = freqz(b(i+num_filters/2,:),a(i+num_filters/2,:)); plot(freq_array, abs(H));
    filter_center_freq2(i + num_filters/2) = fm + (diff2(i)+diff2(i+1))/2;
end
title('Non-Linearly Spaced Filter Bank: Hamming')
xlabel('Frequency (Hz)')
ylabel('Magnitude')
hold off
```

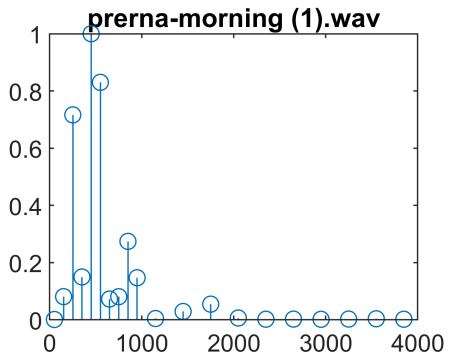


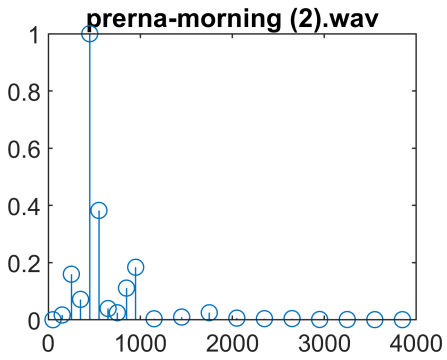
Again creating the PSD using the filter bank and normalizing using the maximum value as done before with the linearly spaced filter bank.

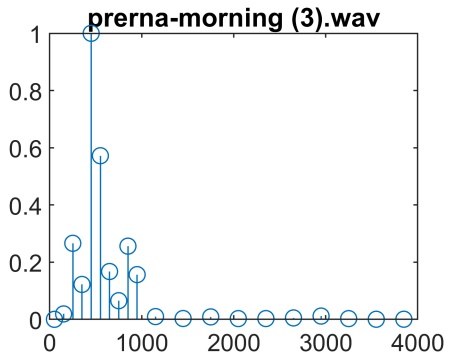
```
psd2 = zeros(size(X,1),num_filters);

for j = 1:size(X,1)
    for i = 1:num_filters
        X_bandpassed2(i,:) = filtfilt(b(i,:),a(i,:), X(j,:));
```

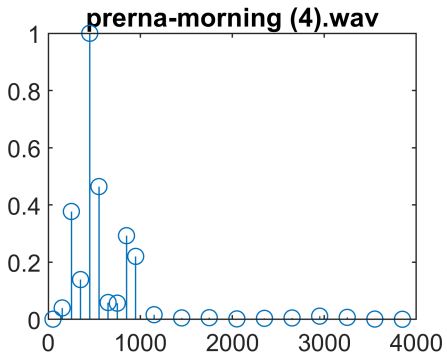
```
        H = freqz(double(X_bandpassed2(i,:)),1);

        psd2(j,i) = sum(abs(H).^2);
    end
    figure

end

psd2_max = max(psd2,[],2);
psd_normalized2 = psd2./psd2_max;
for i = 1:8
    figure
    for j = 1:4
        subplot(2,2,j)
        %stem(filter_center_freq2,psd2(4*(i-1)+j,:)/psd2_max(4*(i-1)+j));
        stem(filter_center_freq2,psd_normalized2(4*(i-1)+j,:))
        title(NAME(4*(i-1)+j))
    end
end
```

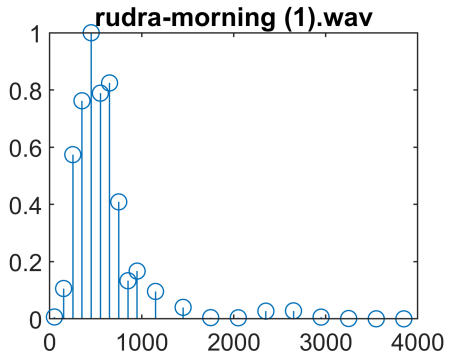


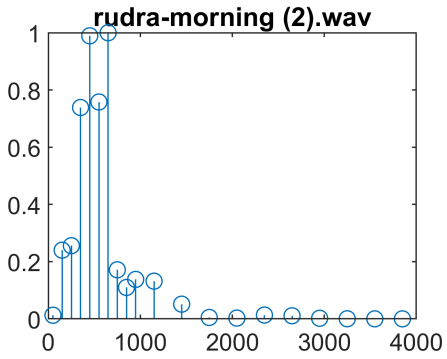

prerna-morning (1).wav
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000

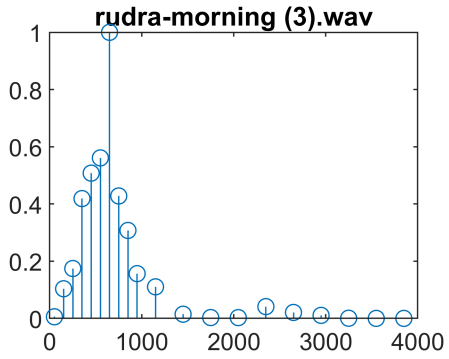

prerna-morning (2).wav
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000

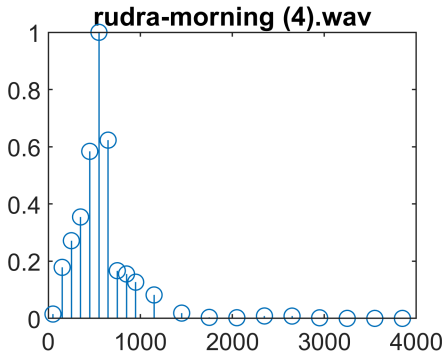

prerna-morning (3).wav
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000

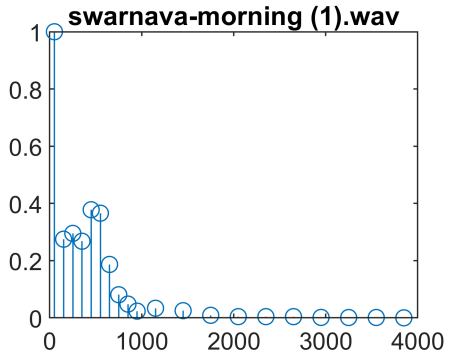

prerna-morning (4).wav
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000

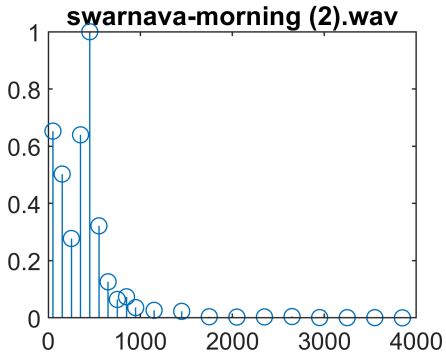

rudra-morning (1).wav
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000

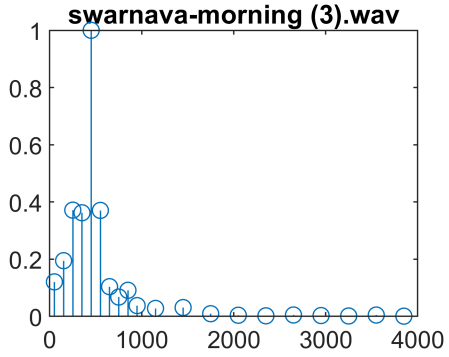

rudra-morning (2).wav
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000

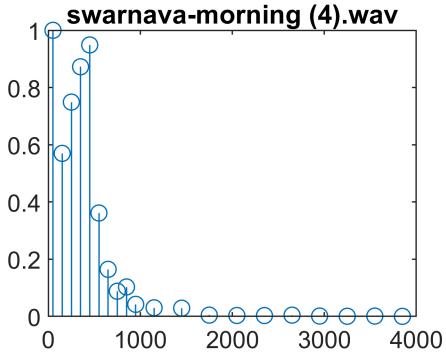

rudra-morning (3).wav
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000

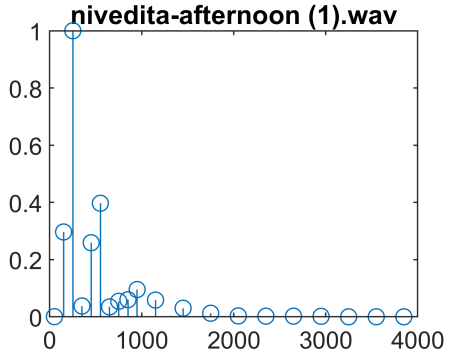

rudra-morning (4).wav
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000

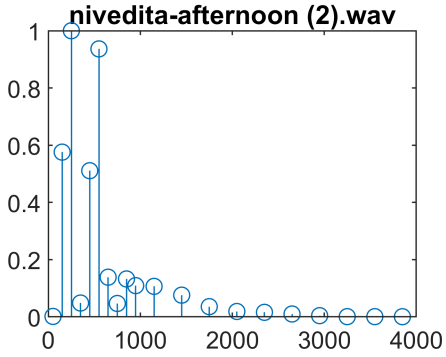

swarnava-morning (1).wav
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000

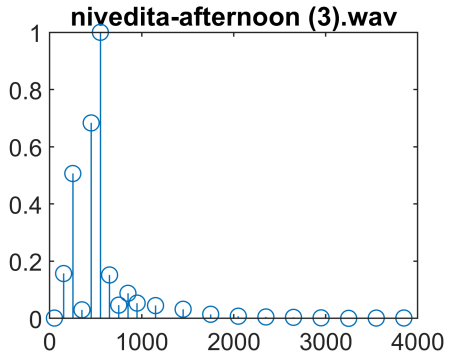

swarnava-morning (2).wav
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000

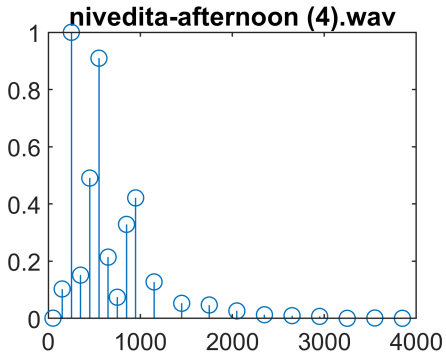

swarnava-morning (3).wav
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000

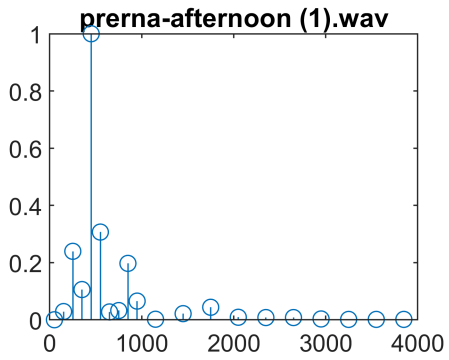

swarnava-morning (4).wav
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000

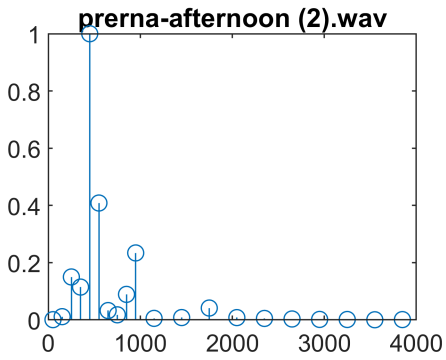

nivedita-afternoon (1).wav
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000

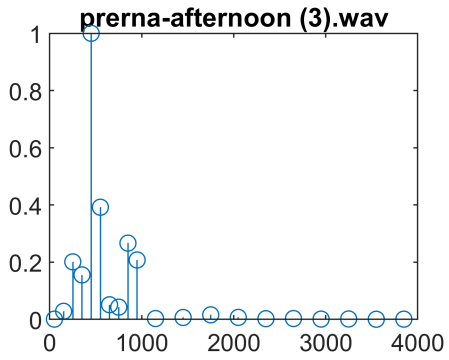

nivedita-afternoon (2).wav
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000

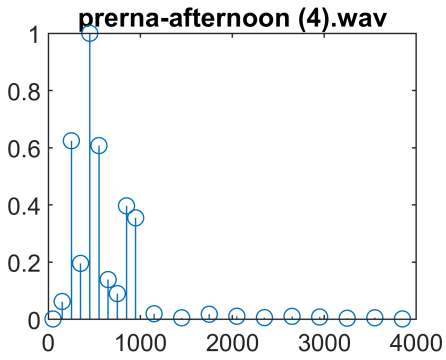

nivedita-afternoon (3).wav
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000

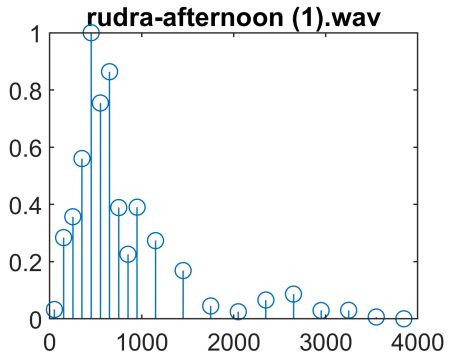

nivedita-afternoon (4).wav
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000

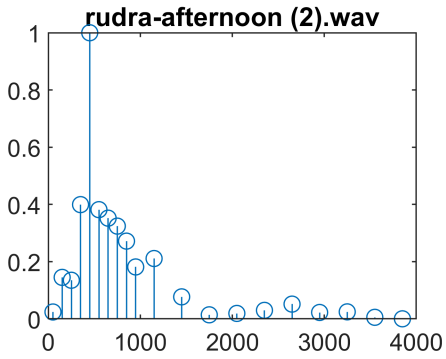

prerna-afternoon (1).wav
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000

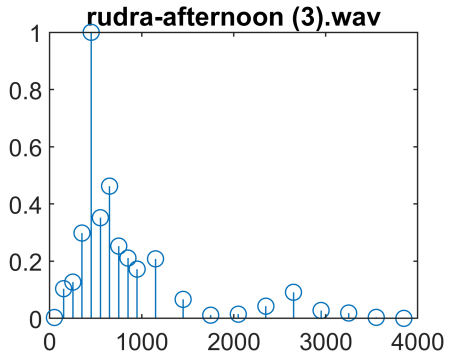

prerna-afternoon (2).wav
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000

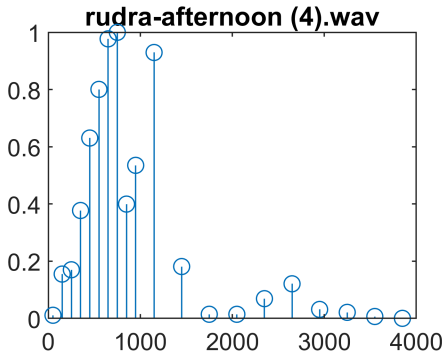

prerna-afternoon (3).wav
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000

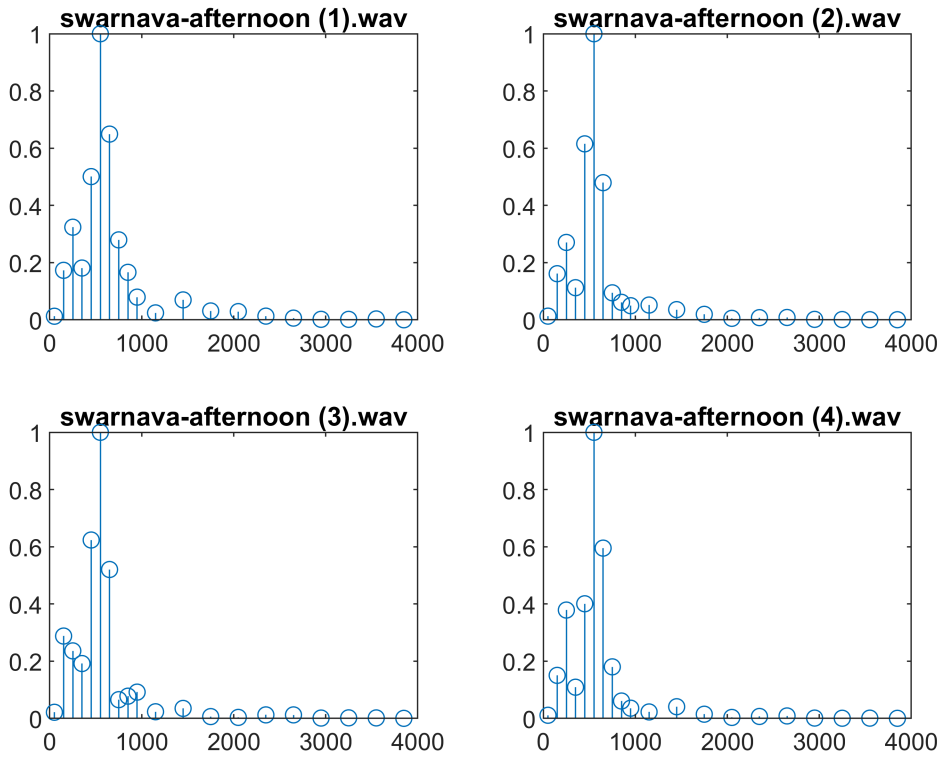

prerna-afternoon (4).wav
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000


rudra-afternoon (1).wav
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000


rudra-afternoon (2).wav
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000

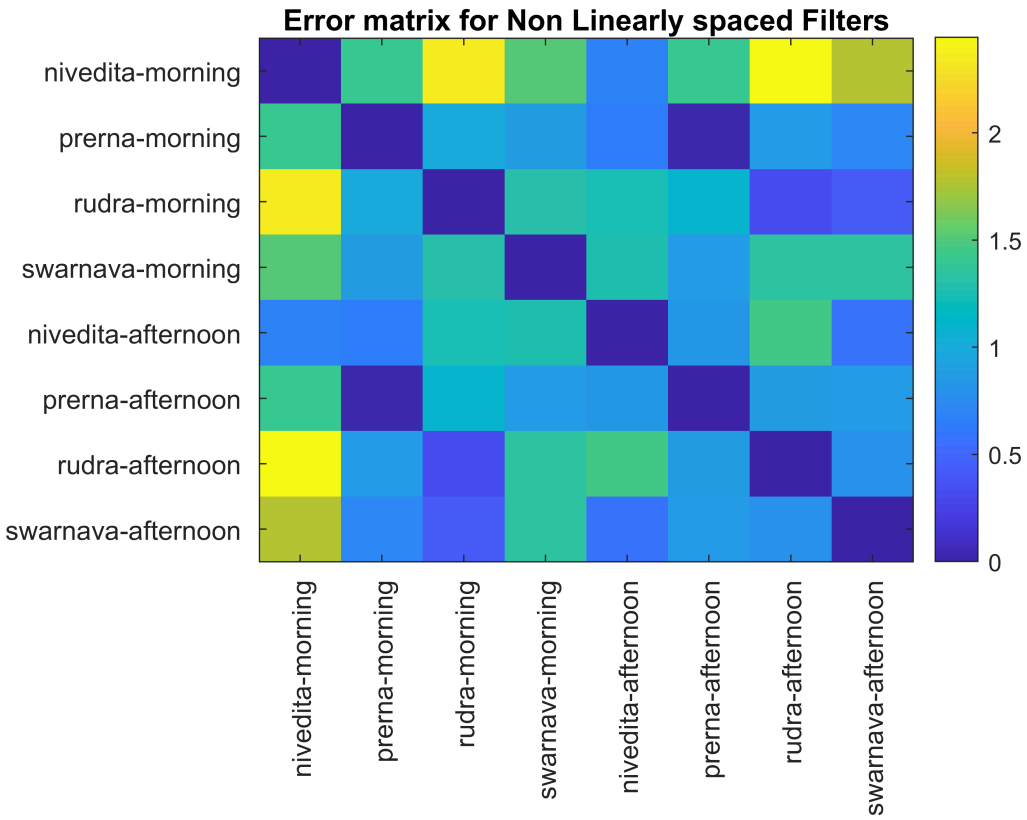

rudra-afternoon (3).wav
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000


rudra-afternoon (4).wav
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000



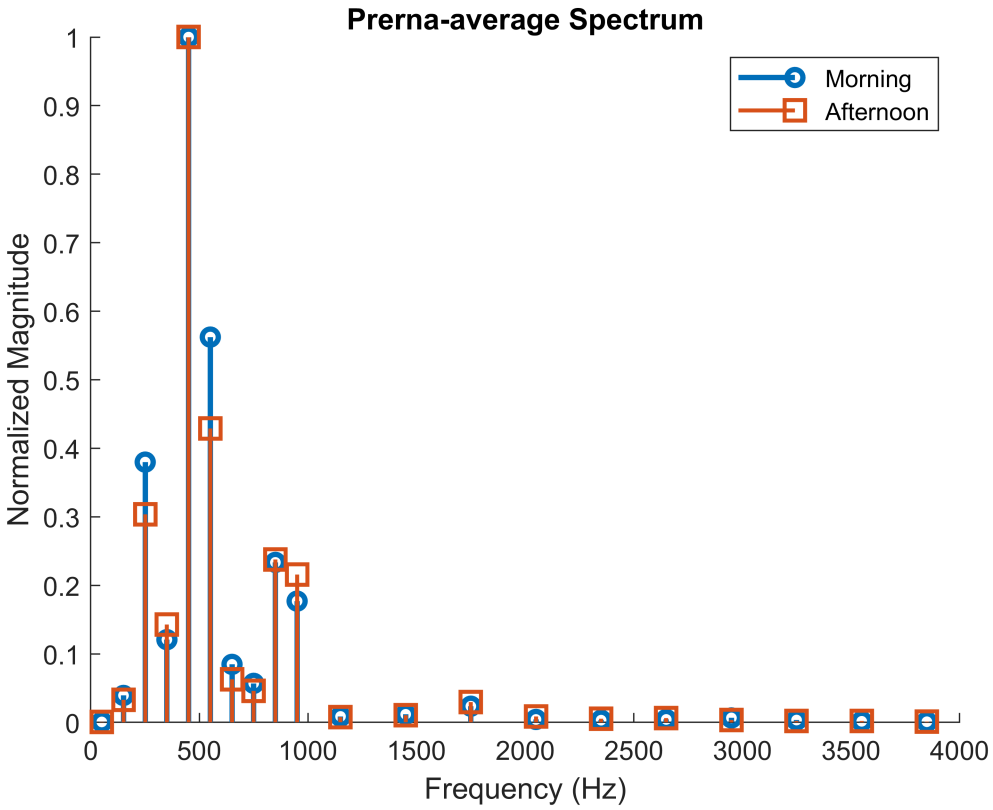

```
m_s_error2 = zeros(size(X,1)/4);
psd_average2 = zeros(size(psd,1)/4,size(psd,2));
for i = 1:8
    sum1 = zeros(1,size(psd,2));
    for j = 1:4
        sum1 = sum1 + psd_normalized2(4*(i-1)+j,:)/4;
    end
    psd_average2(i,:) = sum1;
end

for i = 1:8
    for j = 1:8

        m_s_error2(i,j) = sum((psd_average2(i,:)-psd_average2(j,:)).^2);

    end
end
figure

imagesc(m_s_error2)
title("Error matrix for Non Linearly spaced Filters")

xticks([1:8])

xticklabels(extractBetween(NAME(1:4:length(NAME)),1,strlength(NAME(1:4:length(NAME)))-7));
xtickangle(90)
yticks([1:8])
```

```
yticklabels(extractBetween(NAME(1:4:length(NAME)),1,strlength(NAME(1:4:length(NAME)))-7));
ytickangle(0)
colorbar
```

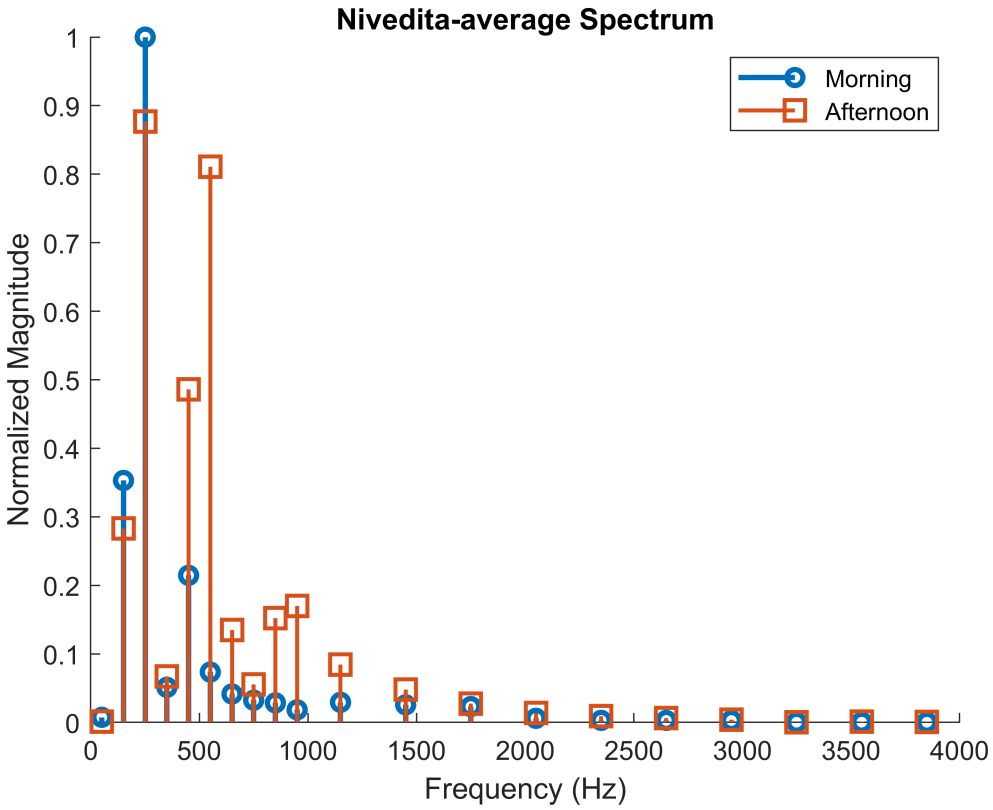


Comparison betweeen morning and afternooon Spectrum:

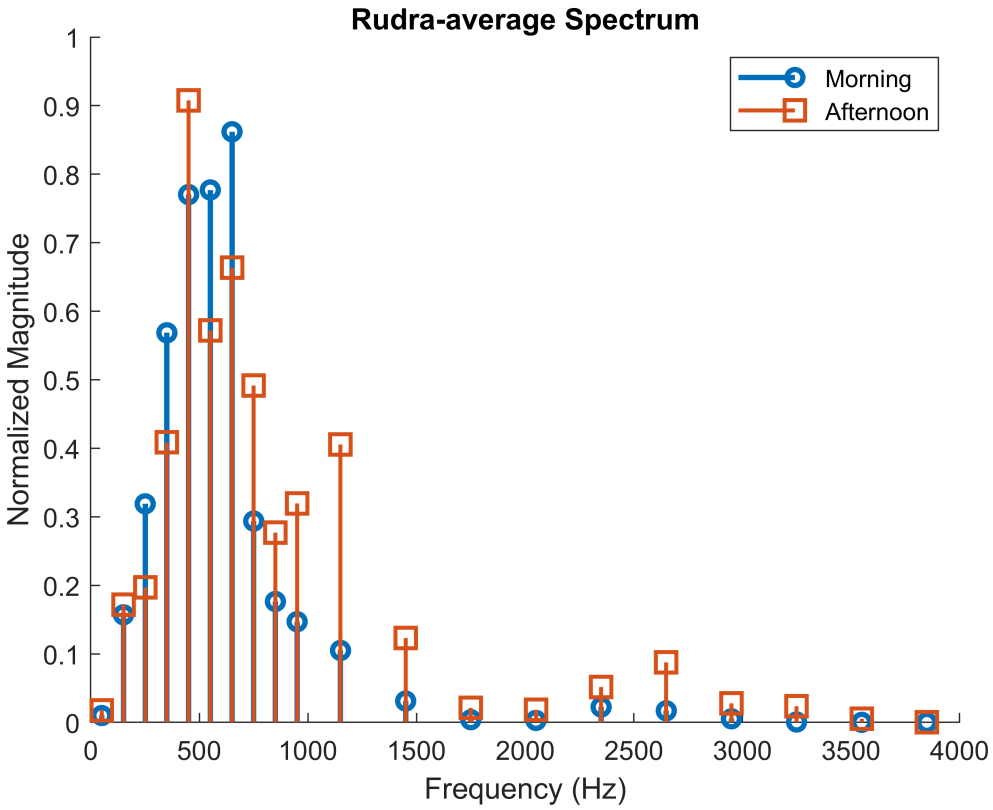
```
figure
hold on
stem(filter_center_freq2,psd_average2(2,:),"LineWidth",2)
stem(filter_center_freq2,psd_average2(6,:),'Marker',"square","MarkerSize",10,"LineWidth",1.5)
hold off
legend('Morning','Afternoon')
title("Prerna-average Spectrum")
ylabel("Normalized Magnitude")
xlabel("Frequency (Hz)")
```



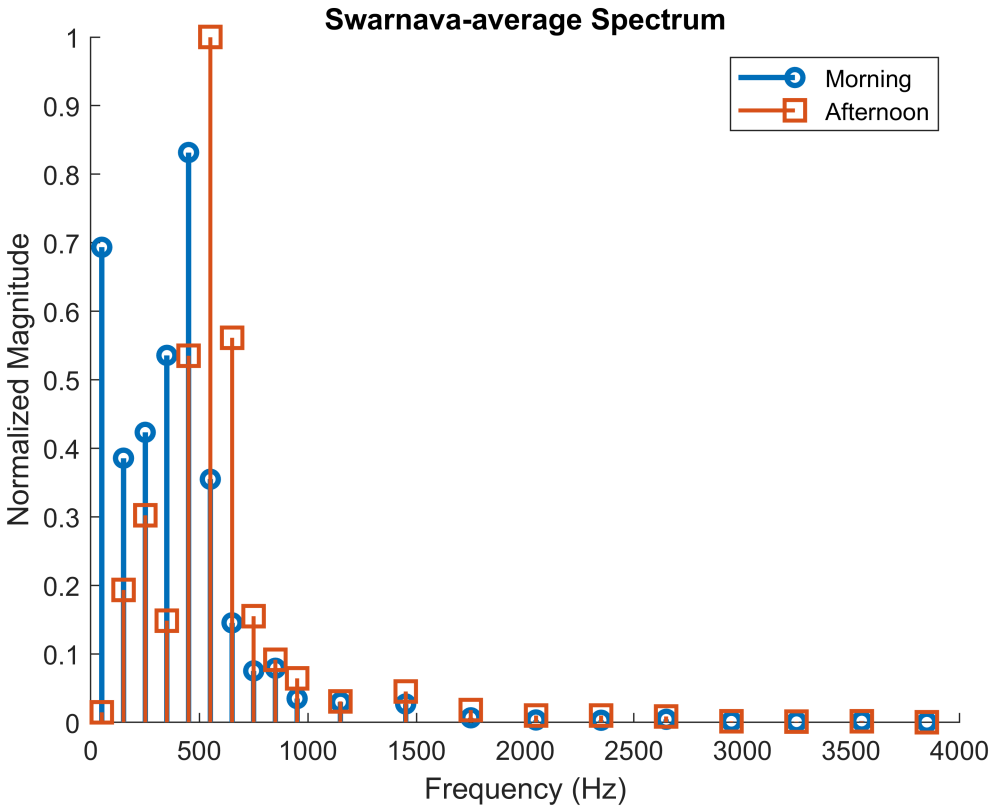

```
figure
hold on
stem(filter_center_freq2,psd_average2(1,:),"LineWidth",2)
stem(filter_center_freq2,psd_average2(5,:),'Marker',"square","MarkerSize",10,"LineWidth",1.5)
hold off
legend('Morning','Afternoon')
title("Nivedita-average Spectrum")
ylabel("Normalized Magnitude")
xlabel("Frequency (Hz)")
```




```
figure
hold on
stem(filter_center_freq2,psd_average2(3,:),"LineWidth",2)
stem(filter_center_freq2,psd_average2(7,:),'Marker',"square","MarkerSize",10,"LineWidth",1.5)
hold off
legend('Morning','Afternoon')
title("Rudra-average Spectrum")
ylabel("Normalized Magnitude")
xlabel("Frequency (Hz)")
```




```
figure
hold on
stem(filter_center_freq2,psd_average2(4,:),"LineWidth",2)
stem(filter_center_freq2,psd_average2(8,:),'Marker',"square","MarkerSize",10,"LineWidth",1.5)
hold off
legend('Morning','Afternoon')
title("Swarnava-average Spectrum")
ylabel("Normalized Magnitude")
xlabel("Frequency (Hz)")
```



Rudimentary Detection Algorithm

It can be seen from the speech signals of the volunteers that there speech can be characterized by the variance in the normalized frequency spectrum. It is observed that the speech of "Rudra" has peaks in more frequency bands when compared to "Nivedita" or "Prerna". This helps us distinguish between the voice signals by taking a summation of the magnitudes of each band pass filter output.

```
count_correct = 0;
for i = 1:32
    disp(NAME(i))
    sum_final = sum(psd_normalized2(i,1:10));
    %Hence we characterize the voices by total energy in the discretized PSD
        if(sum_final>3.6)
        name_predict = "rudra"
        elseif (sum_final> 2.8)
        name_predict = "swarnava"
        elseif (sum_final>2.2)
        name_predict = "prerna"
        else
        name_predict = "nivedita"
        end
    disp(" ")
    if(contains(NAME(i),name_predict))
        count_correct = count_correct+1;
    end
end
```

```
nivedita-morning (1).wav
name_predict =
"nivedita"

nivedita-morning (2).wav
name_predict =
"nivedita"

nivedita-morning (3).wav
name_predict =
"nivedita"

nivedita-morning (4).wav
name_predict =
"nivedita"

prerna-morning (1).wav
name_predict =
"swarnava"

prerna-morning (2).wav
name_predict =
"nivedita"

prerna-morning (3).wav
name_predict =
"prerna"

prerna-morning (4).wav
name_predict =
"prerna"

rudra-morning (1).wav
name_predict =
"rudra"

rudra-morning (2).wav
name_predict =
"rudra"

rudra-morning (3).wav
name_predict =
"rudra"

rudra-morning (4).wav
name_predict =
"swarnava"

swarnava-morning (1).wav
name_predict =
"swarnava"

swarnava-morning (2).wav
name_predict =
"rudra"

swarnava-morning (3).wav
name_predict =
"prerna"

swarnava-morning (4).wav
name_predict =
"rudra"
```

```
nivedita-afternoon (1).wav
name_predict =
"prerna"

nivedita-afternoon (2).wav
name_predict =
"swarnava"

nivedita-afternoon (3).wav
name_predict =
"prerna"

nivedita-afternoon (4).wav
name_predict =
"rudra"

prerna-afternoon (1).wav
name_predict =
"nivedita"

prerna-afternoon (2).wav
name_predict =
"nivedita"

prerna-afternoon (3).wav
name_predict =
"prerna"

prerna-afternoon (4).wav
name_predict =
"swarnava"

rudra-afternoon (1).wav
name_predict =
"rudra"

rudra-afternoon (2).wav
name_predict =
"swarnava"

rudra-afternoon (3).wav
name_predict =
"swarnava"

rudra-afternoon (4).wav
name_predict =
"rudra"

swarnava-afternoon (1).wav
name_predict =
"swarnava"

swarnava-afternoon (2).wav
name_predict =
"swarnava"

swarnava-afternoon (3).wav
name_predict =
"swarnava"

swarnava-afternoon (4).wav
name_predict =
"swarnava"
```

```
disp(strcat("The algorithm has an accuracy of ",num2str(count_correct/32*100),"%"))
```

```
The algorithm has an accuracy of 53.125%
```

Part-B

Pitch detection

What is pitch of speech signal? Can you find from published literature a pitch extraction algorithm that makes sense to you? Can you use it in previous study?

For any person, his/her voice signal can be considered a resultant sum of multiple harmonics, that arises due to the vocal cord structure. Due to these, certain frequency components would be present more than others. Out of these harmonics, the fundamental frequency (which is the lowest harmonic) would have the highest energy and will have the most prominent peak in the spectrum, and this frequency is defined as the pitch of the voice. Every person has an unique pitch, that varies slightly based on situations, etc that essentially helps us in identifying the voice.

For a human adult male, voice pitch value is typically 85-180 Hz and for human adult female, it is about 165-255 Hz. This is the main reason why adult males have a deep, groany voice while females have a shrilly speech.

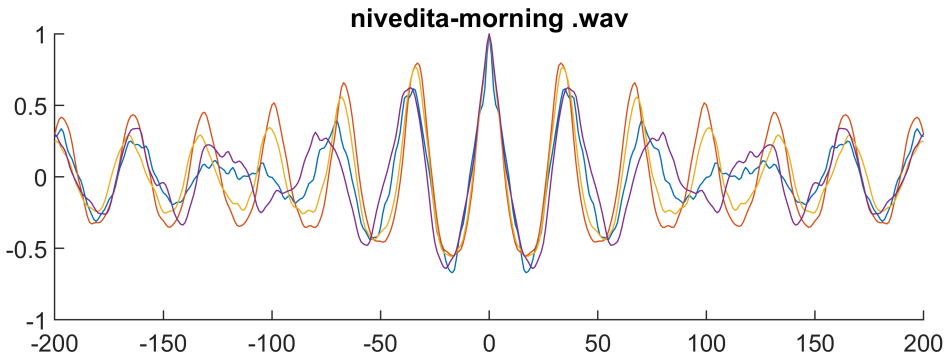
There are many pitch detecting algorithms, out of these we will apply the method based on auto-correlation of the speech signal (time-domain based method). In this method, we will take the audio signal, and find its auto-correlation sequence in a given lag range and see its plot. As the pitch frequency defines the slowest changes in audio signal, the auto-correlation peaks will directly correspond to the period of the pitch.

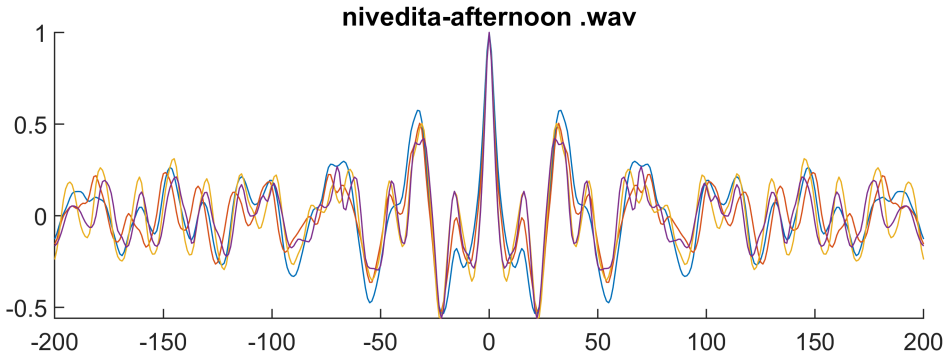
We first find the normalised autocorrelation sequence of each audio signal, and try to obtain the separation between two consecutive peaks (primary one at zero lag and secondary one at some finite lag). If this separation say occurs at some index n, then th pitch is given by **f0 = Fs/n**.

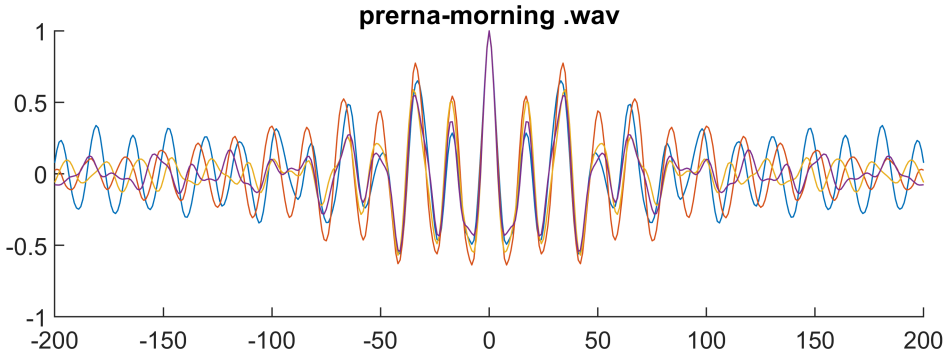
```
element = 1; number = 4;
max_length = 56138; %max permissible filesize
name_array = ["nivedita","prerna","rudra","swarnava"];
time_array = ["morning","afternoon"];

NAME = [];  %initate array of names

for name = 1:length(name_array)
    for time = 1:length(time_array)
        for i = 1:number
            %generating the name of the file using a for loop and then
            %obtaining it using 'audioread' function
            fname = strcat(name_array(name),'-',time_array(time),' (',num2str(i),').wav');
            inputFileName = char(strcat(path,fname));
            [audio_file, Fs] = audioread(inputFileName,'native');

            audio_file = audio_file';
            audio_file = double(audio_file);
%             %audio_file = audio_file./max(audio_file);
```

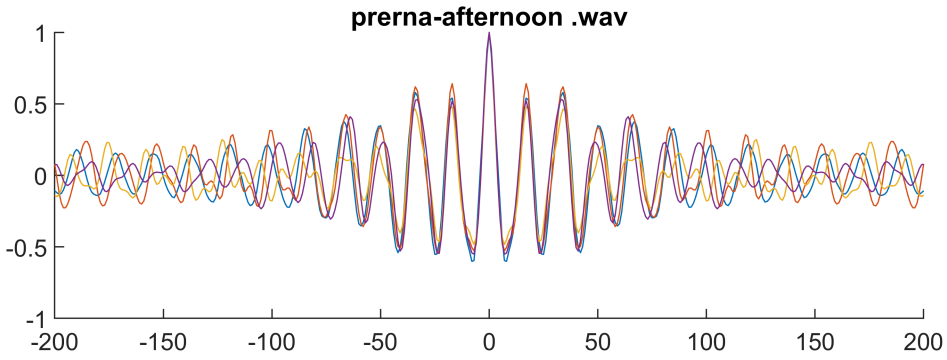
```
            %zero padding code
            if(length(audio_file)<max_length)
                audio_file = padarray(audio_file,[0 (max_length-length(audio_file))],0,'post');
            else
                audio_file = audio_file(1:max_length);
            end

            %entering the sampled audio file in matrix X as well as the
            %name of the file in matrix NAME.
            X(element,:) = audio_file;
            NAME = [NAME fname];

            element = element + 1;
        end
    end
end
```

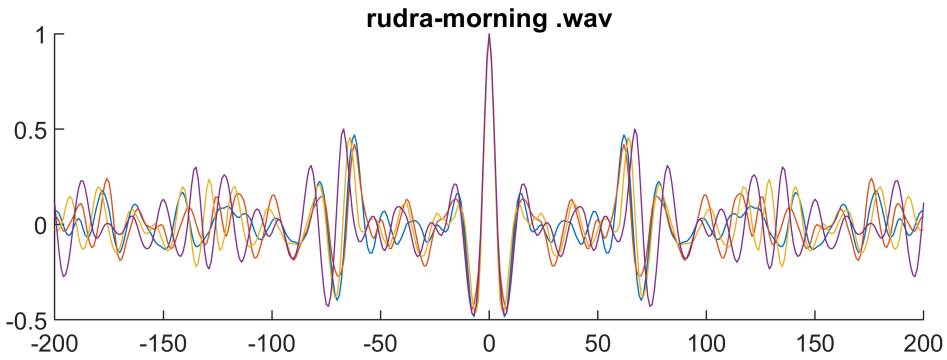
```
%find the correlation sequence
lag_lim = 200; %xlim lag value for auto-corr range
Rxx = zeros(32, 2*lag_lim+1);
lags = zeros(32, 2*lag_lim+1);
sorted_corr_val = zeros(2, 2*lag_lim+1);
f0 = zeros(1, 32);   %will store the calculated pitch frequency

%compute auto-corr of all samples
for a = 1:32
    [Rxx(a, :) ,lags(a, :)] = xcorr(X(a, :), X(a, :), lag_lim, 'normalized');
end

for a = 0:3
    figure
    %auto-corr plots
    subplot(211)
    hold on
    for b = 1:4
        plot(lags(8*a+b, :), Rxx(8*a+b, :));
    end
    title(erase(NAME(8*a+1),"(1)"));
    hold off

    subplot(212)
    hold on
    for b = 5:8
        plot(lags(8*a+b, :), Rxx(8*a+b, :));
    end
    title(erase(NAME(8*a+5),"(1)"));
    hold off


end
```

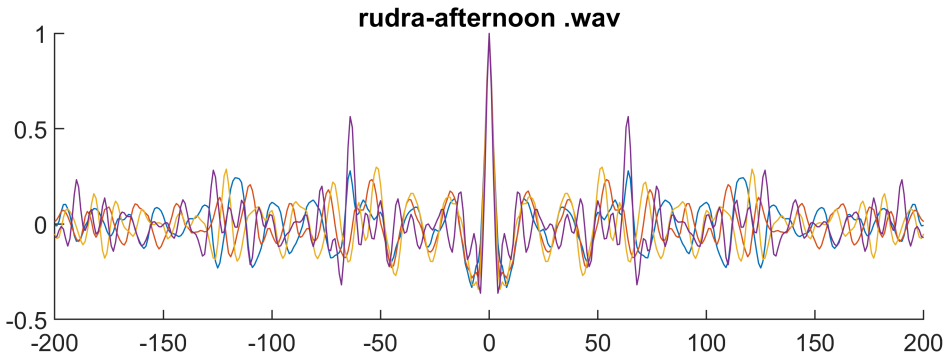

nivedita-morning .wav
1
0.5
0
-0.5
-1
-200
-150
-100
-50
0
50
100
150
200

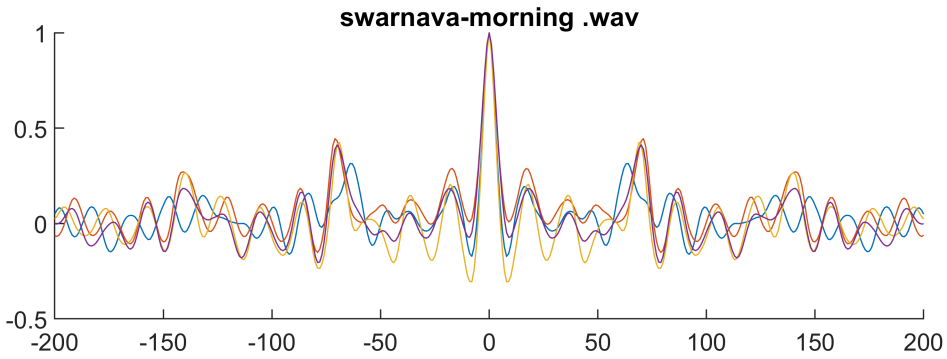

nivedita-afternoon .wav
1
0.5
0
-0.5
-200
-150
-100
-50
0
50
100
150
200

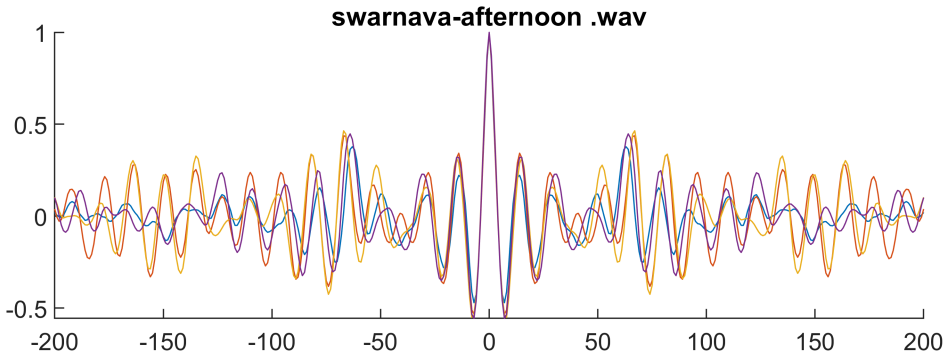

prerna-morning .wav
1
0.5
0
-0.5
-1
-200
-150
-100
-50
0
50
100
150
200


prerna-afternoon .wav
1
0.5
0
-0.5
-1
-200
-150
-100
-50
0
50
100
150
200





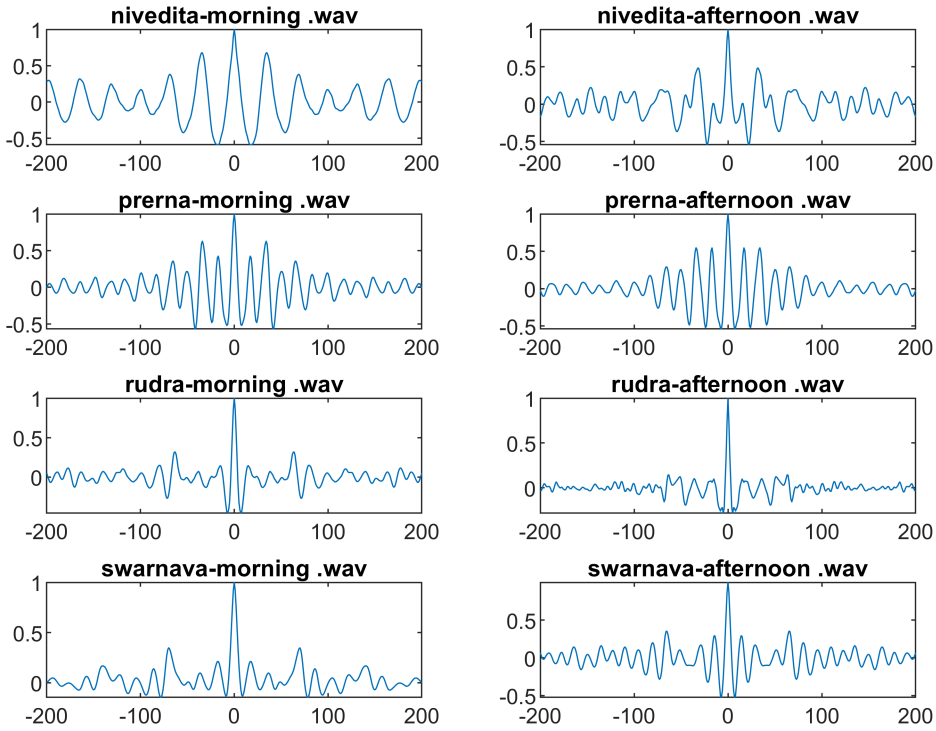




Now we will try to get the average auto-corr sequence then plot it consecutively against each person at each time. In this way, we can see how the pitch varies for a person across the daytime. The secondary peak is

found by sorting the array, finding the lag corresponding to that maxima. This peak is observed to occur after some threshold index, say 5 to eliminate detection of primary lobe.

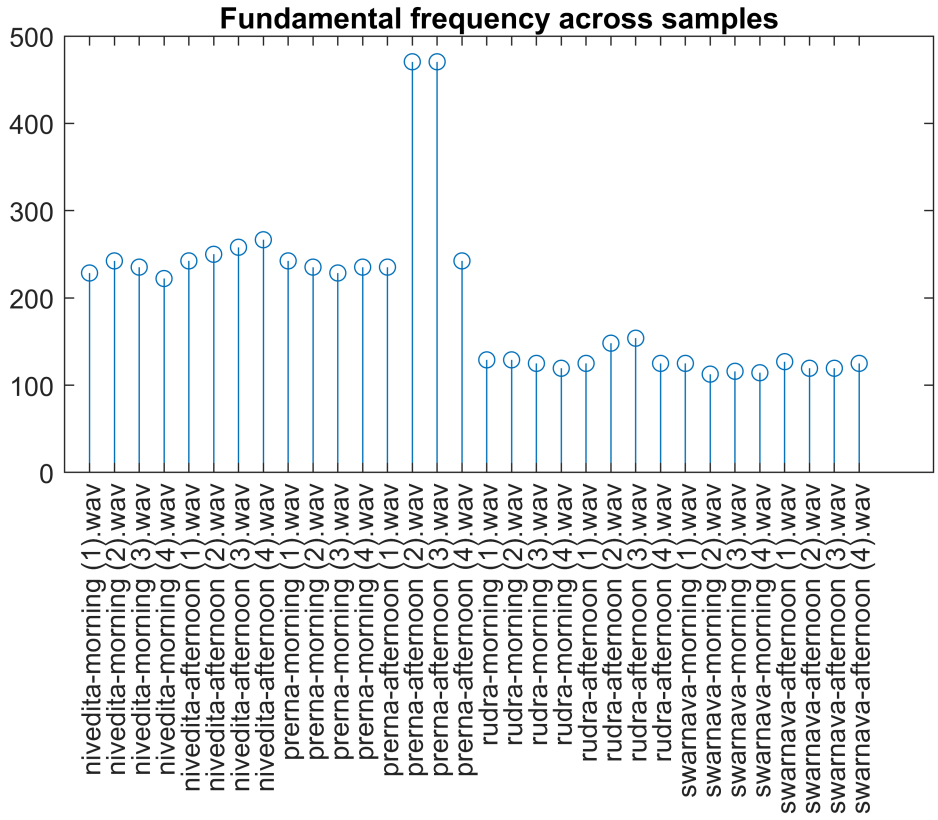
```
Rxx_avg = zeros(8, 2*lag_lim+1);
figure
for a = 1:8
    Rxx_avg(a, :) = mean([Rxx(4*a-3, :); Rxx(4*a-2, :); Rxx(4*a-1, :); Rxx(4*a, :)], 1);
    subplot(4,2,a)
    plot(lags(1, :), Rxx_avg(a, :));
    title(erase(NAME(4*a-3),"(1)"));
end
```




```
ith = 8; %index threshold for secondary peak

%Finding the index of secondary peak
for k = 1:32
    sorted_corr_val = [Rxx(k, :); lags(k, :)];
    sorted_corr_val = sortrows(sorted_corr_val.', 1, 'descend')';
    for b = 1:16
        if(sorted_corr_val(2, b)>ith)
            f0(k) = sorted_corr_val(2, b); break;
        end
    end
end
```

```
f0 = Fs./f0;
```

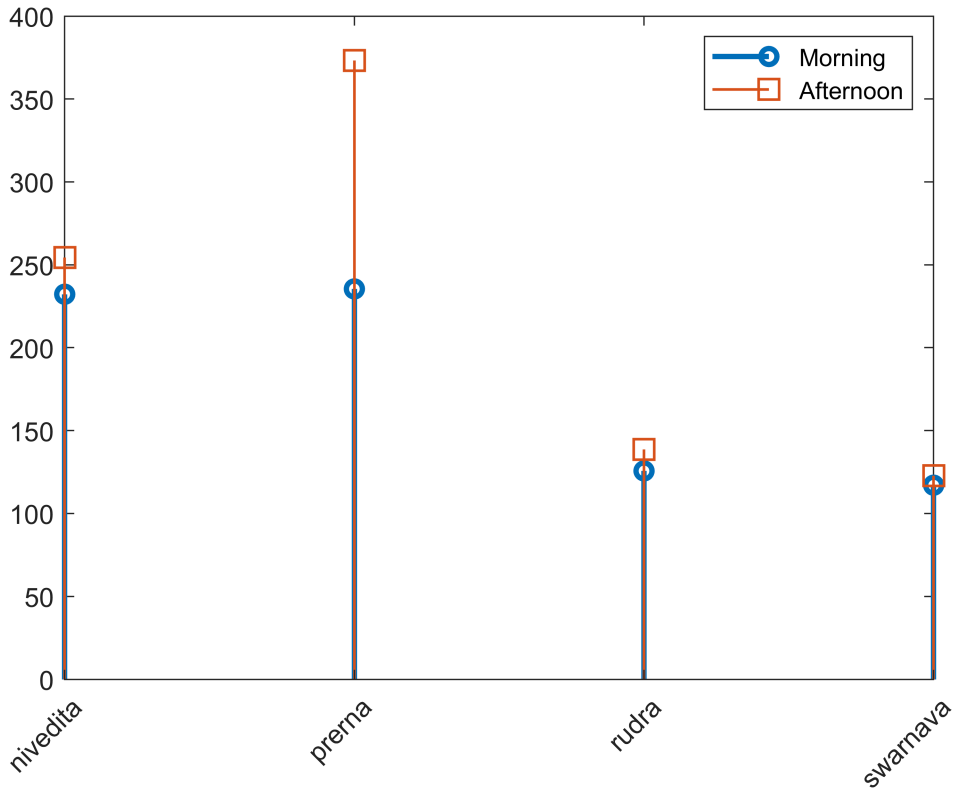
```
figure(111)
stem(f0);
title("Fundamental frequency across samples");
xticks(linspace(1, 32, 32));
xticklabels(NAME);
xtickangle(90);
```



We can clearly see for a given person at a given time, the pitch is almost the same. Girls have a higher pitch than boys. but person wise the distinction is not clear for the given gender. The presence of 2 outliers also mean there will be some error in estimate. Now we try to determine the RMS value of pitch for a given person for a given time of the day.

```
%Get average f0 from this graph
f0_avg = zeros(1, 8);
for a = 1:8
    f0_avg(a) = rms([f0(4*a-3), f0(4*a-2), f0(4*a-1), f0(4*a)]);
end
%store the pitch values
pitch_val = [f0_avg(1), f0_avg(3), f0_avg(5), f0_avg(7)];
pitch_gen = [mean([f0_avg(1), f0_avg(3)]), mean([f0_avg(5), f0_avg(7)])];

figure
stem([f0_avg(1), f0_avg(3), f0_avg(5), f0_avg(7)], "LineWidth", 2);
hold on
stem([f0_avg(2), f0_avg(4), f0_avg(6), f0_avg(8)], 'Marker',"square","MarkerSize",10,"LineWidth
hold off
legend('Morning', 'Afternoon');
```

```
xticks(1:1:4); xticklabels(name_array); xtickangle(45);
```



This graph clearly shows we have estimated the pitch with some variation across morning/afternoon. Those two outliers in the dataset cause Prerna's pitch to be significantly higher in afternoon. From here if we take about some error margin for variation, we can map the obtained pitch to the person speaking as we will do follows.

Person identification from Pitch detection algorithm

In the previous part we have implemented the pitch detection algorithm from voice samples. Thus for each person we have an approximate value of their voice pitch. Now we will try to apply it in identifying the person and the gender, ie. by taking a voice sample, finding its pitch using the same auto-correlation method and then map it to our obtained set of values. We run the same sample set of audio files for this.

```
conf_level1 = 0; conf_level2 = 1;
gender = ["female", "male"];
for a = 1:length(NAME)
    [audio_file, Fs] = audioread(NAME(a), 'native');
    audio_file = transpose(audio_file);
    [audio_auto_corr, lag] = xcorr(audio_file, audio_file, lag_lim, 'normalized');
    sorted_corr_val = [audio_auto_corr; lag];
    sorted_corr_val = sortrows(sorted_corr_val.', 1, 'descend')';
    for b = 1:16
        if(sorted_corr_val(2, b)>ith)
            fp = sorted_corr_val(2, b); break;
        end
    end
    fp = Fs/fp;  %get pitch freq
```

```
    %person-based identification
    for b = 1:number
        if abs(fp - pitch_val(b)) < 10
            display(strcat(NAME(a),"  pitch = ", num2str(fp), " person = ", name_array(b)))
            if contains(NAME(a), name_array(b))
                conf_level1 = conf_level1 + 1;
            end
            break;
        end
    end
    %gender-based identification
    for b = 1:2
        if abs(fp - pitch_gen(b)) < 40
            display(strcat(NAME(a),"  pitch = ", num2str(fp), "gender = ", gender(b)))
            if (a < 17 && b == 1) || (a > 16 && b == 2)
                conf_level2 = conf_level2 + 1;
            end
            break;
        end
    end
end
```

```
"nivedita-morning (1).wav  pitch = 228.5714 person = nivedita"
"nivedita-morning (1).wav  pitch = 228.5714gender = female"
"nivedita-morning (2).wav  pitch = 242.4242 person = prerna"
"nivedita-morning (2).wav  pitch = 242.4242gender = female"
"nivedita-morning (3).wav  pitch = 235.2941 person = nivedita"
"nivedita-morning (3).wav  pitch = 235.2941gender = female"

"nivedita-morning (4).wav  pitch = 222.2222gender = female"
"nivedita-afternoon (1).wav  pitch = 242.4242 person = prerna"
"nivedita-afternoon (1).wav  pitch = 242.4242gender = female"

"nivedita-afternoon (2).wav  pitch = 250gender = female"

"nivedita-afternoon (3).wav  pitch = 258.0645gender = female"

"nivedita-afternoon (4).wav  pitch = 266.6667gender = female"
"prerna-morning (1).wav  pitch = 242.4242 person = prerna"
"prerna-morning (1).wav  pitch = 242.4242gender = female"
"prerna-morning (2).wav  pitch = 235.2941 person = nivedita"
"prerna-morning (2).wav  pitch = 235.2941gender = female"
"prerna-morning (3).wav  pitch = 228.5714 person = nivedita"
"prerna-morning (3).wav  pitch = 228.5714gender = female"
"prerna-morning (4).wav  pitch = 235.2941 person = nivedita"
"prerna-morning (4).wav  pitch = 235.2941gender = female"
"prerna-afternoon (1).wav  pitch = 235.2941 person = nivedita"
"prerna-afternoon (1).wav  pitch = 235.2941gender = female"
"prerna-afternoon (4).wav  pitch = 242.4242 person = prerna"
"prerna-afternoon (4).wav  pitch = 242.4242gender = female"
"rudra-morning (1).wav  pitch = 129.0323 person = rudra"
"rudra-morning (1).wav  pitch = 129.0323gender = male"
"rudra-morning (2).wav  pitch = 129.0323 person = rudra"
"rudra-morning (2).wav  pitch = 129.0323gender = male"
"rudra-morning (3).wav  pitch = 125 person = rudra"
"rudra-morning (3).wav  pitch = 125gender = male"
"rudra-morning (4).wav  pitch = 119.403 person = rudra"
"rudra-morning (4).wav  pitch = 119.403gender = male"
"rudra-afternoon (1).wav  pitch = 125 person = rudra"
"rudra-afternoon (1).wav  pitch = 125gender = male"
```

```
"rudra-afternoon (2).wav  pitch = 148.1481gender = male"

"rudra-afternoon (3).wav  pitch = 153.8462gender = male"
"rudra-afternoon (4).wav  pitch = 125 person = rudra"
"rudra-afternoon (4).wav  pitch = 125gender = male"
"swarnava-morning (1).wav  pitch = 125 person = rudra"
"swarnava-morning (1).wav  pitch = 125gender = male"
"swarnava-morning (2).wav  pitch = 112.6761 person = swarnava"
"swarnava-morning (2).wav  pitch = 112.6761gender = male"
"swarnava-morning (3).wav  pitch = 115.942 person = rudra"
"swarnava-morning (3).wav  pitch = 115.942gender = male"
"swarnava-morning (4).wav  pitch = 114.2857 person = swarnava"
"swarnava-morning (4).wav  pitch = 114.2857gender = male"
"swarnava-afternoon (1).wav  pitch = 126.9841 person = rudra"
"swarnava-afternoon (1).wav  pitch = 126.9841gender = male"
"swarnava-afternoon (2).wav  pitch = 119.403 person = rudra"
"swarnava-afternoon (2).wav  pitch = 119.403gender = male"
"swarnava-afternoon (3).wav  pitch = 119.403 person = rudra"
"swarnava-afternoon (3).wav  pitch = 119.403gender = male"
"swarnava-afternoon (4).wav  pitch = 125 person = rudra"
"swarnava-afternoon (4).wav  pitch = 125gender = male"
```

```
disp(strcat("True detections: Person-wise = ", num2str(conf_level1*100/32), "% " + ...
    "   Gender-wise = ", num2str(conf_level2*100/32), "%"))
```

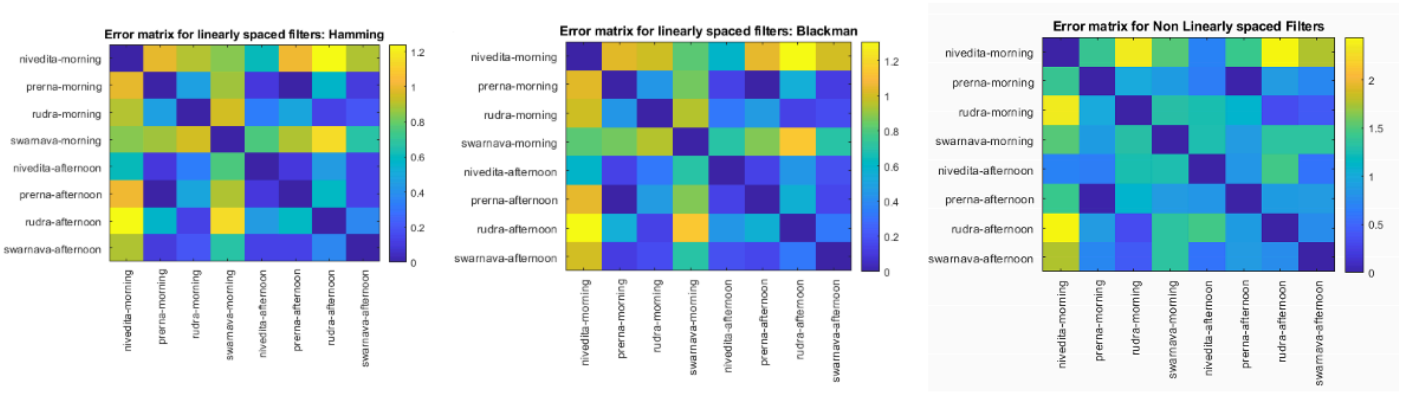
```
True detections: Person-wise = 37.5%    Gender-wise = 96.875%
```

Result

Thus this pitch detection algorithm gives correct person identification (true positives) as 37.5%, but gender wise it is much higher at 97%. This is basically due to the fact that both the boys and the girls had comparable pitch when taken the RMS value. The variation, that can be as low as 5-10 Hz is higher than the difference between them, and thus having a good thershold difference is impractial for the given data set. This threshold is higher gender wise, thus higher true positives. As a result, we got only 37.5% correct identification of person, which is not that much depended on the gender.

Discussion:

- From the initial spectrum we see that for the sampling rate of 8000Hz, most of the energy in the speech singal is in the lower frequencies, than the higher frequencies.
- It can be seen that in the morning, all volunteers are seen to have more energy in lower frequencies in their speech, and as the day progresses, there speech signal spectrum seems to have shifted slightly to the right side (higher frequency).
- As most of there energy is in the range of 0 to 1000Hz, the filter bank using the non linear filter distribution is able to convey more characteristic information about the speech signal of each volunteer.
- We see the error matrices of the 3 filter banks side by side.



Contrary to our intuition, we can see a much more varied coloring in the cases for equally spaced filters. The more varied coloring means that the differenciation of the speech signals is better in that case.

- The upper and lower parts of the left diagonal are symmetric due to the symmetric axis of the error matrix.
- From the spectrum, we also notice that volunteers who were girls had more reletive energy in the higher frequency as compared to boy volunteers, which is also established with the pitch detection algorithm.
- The pitch detection algorithm is based on the fundamental frequency and hence it has a very high precision in detecting gender due to the natural differnece in pitch, but is unable figure out the difference between two boys or two girls.
- The rudimentary algorithm is based on the shape of the reletive frequency spectrum and hence does a better job is relating a spectrum to the correct person.

References

[1] D. Gerhard. Pitch Extraction and Fundamental Frequency: History and Current Techniques, technical report, Dept. of Computer Science, University of Regina, 2003.