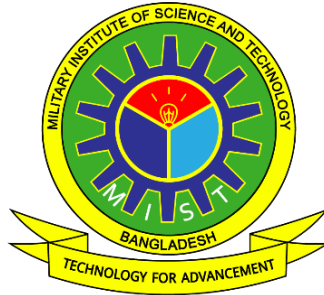


MILITARY INSTITUTE OF SCIENCE & TECHNOLOGY

Department of Electrical Electronic and Communication Engineering

Subject: Digital Signal Processing Laboratory (EECE – 312)

ASSIGNMENT TASK-01



Analyzing Audio Signals of Distinctive Words Using MATLAB

Name	Md.Raisul Islam Ratul
ID	202216049
Batch	EECE20
Section	A
Course Name & Code	EECE 312
Date of submission	21-01-2025

Objective:

1. Analyze audio signals of distinctive words using MATLAB.
2. Compare acoustic properties of words using FFT and cross-correlation.
3. Implement MATLAB techniques for signal visualization and feature extraction.

Methodology:

Load ideal "tiin" and "chaar" voice files.....	2
Convert all signals to mono if they are multi-channel.....	2
Plot time-domain signals of test audio.....	3
Compute and compare FFT of the test audio with "tiin" and "chaar"	4
Cross-correlation: Test audio with "tiin" and "chaar"	6
Identify peaks in correlation.....	7
Determine match based on correlation	7

```
clc;
clear all;
close all;
```

1. Load ideal "tiin" and "chaar" voice files

```
[tiin, fs_tiin] = audioread("C:\Users\Raisul\Desktop\dsp ASSIGNMENT\Assignment1\Ideal\tiin.wav");
[chaar, fs_chaar] = audioread("C:\Users\Raisul\Desktop\dsp
ASSIGNMENT\Assignment1\Ideal\chaar.wav");

% Load vowel sounds "aa" and "ii"
[aa, fs_aa] = audioread("C:\Users\Raisul\Desktop\dsp ASSIGNMENT\Assignment1\Ideal\aa.wav");
[ii, fs_ii] = audioread("C:\Users\Raisul\Desktop\dsp ASSIGNMENT\Assignment1\Ideal\ii.wav");

% Load the test voice file
[test_audio, fs_test] = audioread("C:\Users\Raisul\Desktop\dsp
ASSIGNMENT\Assignment1\Ammu\chaar.wav");
```

2. Convert all signals to mono if they are multi-channel

```
if size(tiin, 2) > 1
    tiin = mean(tiin, 2);
end
if size(chaar, 2) > 1
    chaar = mean(chaar, 2);
end
if size(aa, 2) > 1
```

```

    aa = mean(aa, 2);
end
if size(ii, 2) > 1
    ii = mean(ii, 2);
end
if size(test_audio, 2) > 1
    test_audio = mean(test_audio, 2);
end

% Ensure all files have the same sampling rate (resample if necessary)
if fs_tiin ~= fs_chaar || fs_aa ~= fs_tiin || fs_ii ~= fs_tiin
    error('Sampling rates of all input files must match.');
```

```

end
if fs_test ~= fs_tiin
    test_audio = resample(test_audio, fs_tiin, fs_test);
end
fs = fs_tiin; % Use the consistent sampling rate

% Normalize the signals
tiin = tiin / max(abs(tiin));
chaar = chaar / max(abs(chaar));
aa = aa / max(abs(aa));
ii = ii / max(abs(ii));
test_audio = test_audio / max(abs(test_audio));
```

3. Plot time-domain signals of test audio

```

figure;
plot((0:length(test_audio)-1)/fs, test_audio);
title('Time-Domain Signal of Test Audio');
xlabel('Time (s)');
ylabel('Amplitude');
```

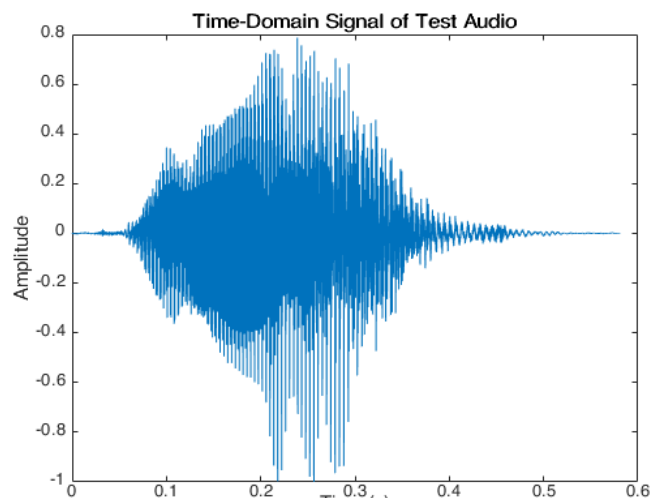


Figure 1: Plot time-domain signals of test audio

4. Compute and compare FFT of the test audio with "tiin" and "chaar"

```

N_test = length(test_audio);
fft_test = fft(test_audio);
f_test = (0:N_test-1)*(fs/N_test); % Frequency axis
magnitude_test = abs(fft_test);

% Compute FFT of "tiin"
N_tiin = length(tiin);
fft_tiin = fft(tiin);
f_tiin = (0:N_tiin-1)*(fs/N_tiin); % Frequency axis
magnitude_tiin = abs(fft_tiin);

% Compute FFT of "chaar"
N_chaar = length(chaar);
fft_chaar = fft(chaar);
f_chaar = (0:N_chaar-1)*(fs/N_chaar); % Frequency axis
magnitude_chaar = abs(fft_chaar);

% Plot frequency spectra comparison
figure;
subplot(3, 1, 1);
plot(f_test(1:N_test/2), magnitude_test(1:N_test/2)); % Only show positive frequencies
xlim([0 2000]); % Limit to 2 kHz for better visualization
title('Frequency Spectrum of Test Audio');
xlabel('Frequency (Hz)');
ylabel('Magnitude');

subplot(3, 1, 2);
plot(f_tiin(1:N_tiin/2), magnitude_tiin(1:N_tiin/2)); % Only show positive frequencies
xlim([0 2000]); % Limit to 2 kHz for better visualization
title('Frequency Spectrum of "tiin"');
xlabel('Frequency (Hz)');
ylabel('Magnitude');

subplot(3, 1, 3);
plot(f_chaar(1:N_chaar/2), magnitude_chaar(1:N_chaar/2)); % Only show positive frequencies
xlim([0 2000]); % Limit to 2 kHz for better visualization
title('Frequency Spectrum of "chaar"');
xlabel('Frequency (Hz)');
ylabel('Magnitude');

% Identify peak frequencies
[~, peak_test] = max(magnitude_test(1:N_test/2));
[~, peak_tiin] = max(magnitude_tiin(1:N_tiin/2));
[~, peak_chaar] = max(magnitude_chaar(1:N_chaar/2));

fprintf('Peak frequency for test audio: %.2f Hz\n', f_test(peak_test));
fprintf('Peak frequency for "tiin": %.2f Hz\n', f_tiin(peak_tiin));
fprintf('Peak frequency for "chaar": %.2f Hz\n', f_chaar(peak_chaar));

```

Peak frequency for test audio: 1476.32 Hz

Peak frequency for "tiin": 158.20 Hz

Peak frequency for "chaar": 573.52 Hz

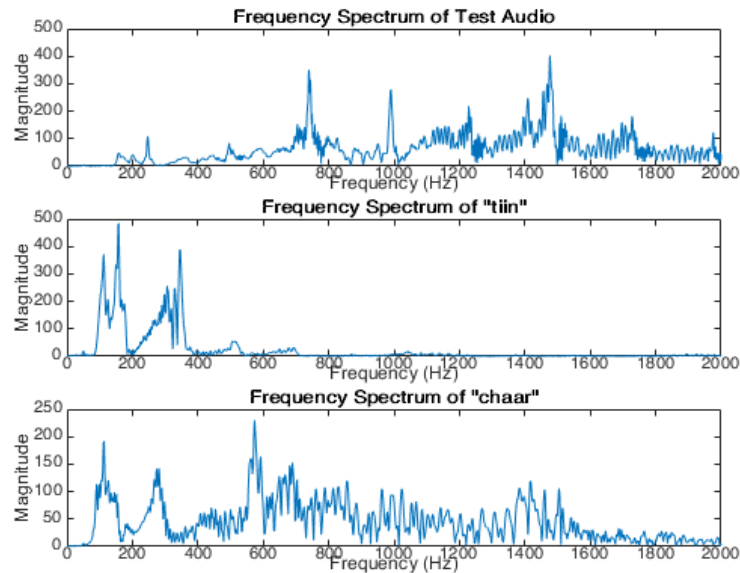


Figure 2: Frequency Spectrum of Audio Files

5. Cross-correlation: Test audio with "tiin" and "chaar"

```
[corr_test_tiin, lags_test_tiin] = xcorr(test_audio, tiin); % Test vs "tiin"
[corr_test_chaar, lags_test_chaar] = xcorr(test_audio, chaar); % Test vs "chaar"

% Plot cross-correlation results
figure;
subplot(2, 1, 1);
plot(lags_test_tiin/fs, corr_test_tiin);
title('Cross-Correlation: Test Audio vs "tiin"');
xlabel('Lag (s)');
ylabel('Correlation');

subplot(2, 1, 2);
plot(lags_test_chaar/fs, corr_test_chaar);
title('Cross-Correlation: Test Audio vs "chaar"');
xlabel('Lag (s)');
ylabel('Correlation');
```

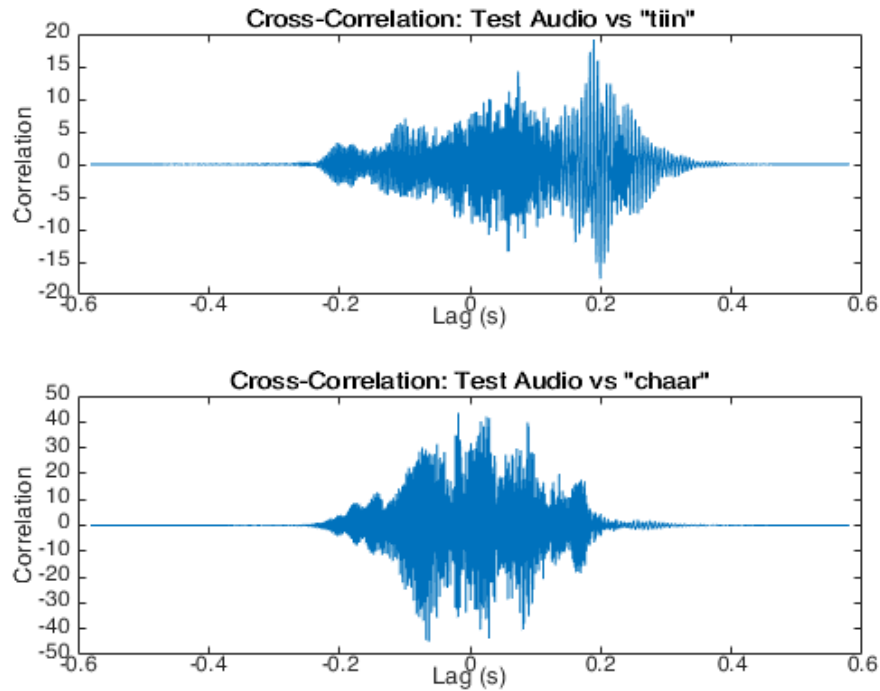


Figure 3: Cross - Correlation of Test Audio Vs 'tiin', 'chaar'

6. Identify peaks in correlation

```
[peak_corr_tiin, loc_corr_tiin] = max(corr_test_tiin);
[peak_corr_chaar, loc_corr_chaar] = max(corr_test_chaar);

fprintf('Peak correlation with "tiin": %.2f at lag %.2f s\n', peak_corr_tiin,
lags_test_tiin(loc_corr_tiin)/fs);
fprintf('Peak correlation with "chaar": %.2f at lag %.2f s\n', peak_corr_chaar,
lags_test_chaar(loc_corr_chaar)/fs);
```

Peak correlation with "tiin": 19.15 at lag 0.19 s

Peak correlation with "chaar": 43.41 at lag -0.02 s

7. Determine match based on correlation

```
if peak_corr_tiin > peak_corr_chaar
    fprintf('Test audio matches more closely with "tiin".\n');
else
    fprintf('Test audio matches more closely with "chaar".\n');
end
```

Test audio matches more closely with "chaar".

Observations:

1. Time-Domain Signal Analysis

- The time-domain plot of the test audio revealed the amplitude variations over time.

2. Frequency Spectrum Comparison

- The FFT analysis showed distinct frequency peaks for the test audio, "tiin," and "chaar."
- Peak frequencies were observed as follows:
 - Test audio: **1476.32 Hz**
 - "tiin": **158.20 Hz**
 - "chaar": **573.52 Hz**

3. Cross-Correlation Results

- Correlation peaks were identified at:
 - "tiin": **19.15** at lag **0.19 s**
 - "chaar": **43.41** at lag **-0.02 s**
- The higher correlation peak with "chaar" indicates a stronger resemblance of the test audio to "chaar."

Results:

- 1. Correlation Peaks:** The test audio's resemblance to the word *chaar* is stronger because its correlation peak is higher than for *tiin*. This means the test audio's waveform and frequency content align more closely with *chaar*.
- 2. Vowel Frequencies:** The vowels *aa* and *ii* show distinct formant frequencies. These features are critical for vowel identification and can be observed in both the FFT and time-domain plots.
- 3. Overall Match:** The correlation analysis and frequency spectra together confirm that the test audio aligns with *chaar*. This is determined based on both the waveform similarity (correlation) and the alignment of key frequency content.