BACHELORS OF TECHNOLOGY
(Electronics & Communication)

Netaji Subhas University of Technology, East Campus
(Formerly Ambedkar Institute of Advanced Communication Technologies & Research , Geeta Colony)
Delhi-110031

CONTROL SYSTEMS
MINI PROJECT

Submitted By :

SUMIT KUMAR SHAKYA
03410102819
5th Semester, ECE1

EXPERIMENT – 10 : MINI PROJECT

AIM :- *Speech Recognition Using Correlation Method*

SOFTWARE USED : *MATLAB R2021a.*

Video where I am demonstrating our project : ***drive link***
Source code : ***GitHub link***

THEORY :

Speech Recognition in MATLAB using Correlation

Speech recognition is used in almost every security project where you need to speak and tell your password to a computer and is also used for automation.

For example, I want to turn my AC on or off using voice commands then I have to use Speech Recognition. I have to make the system recognize that whether I am saying ON or OFF. In short, speech recognition plays a vital role in voice control projects.
We have used in this project is known as cross correlation.
Correlation is normally used in signal processing, where you need to compare two signals and need to find the similarity between them. It is also known as the dot product of those two signals. Correlation has many uses. Correlation is also used for pattern recognition like you want to find some pattern in the signal then you can use Correlation.
In our project, we are using correlation to find similarities between our stored signals and the testing signal. So, let's get started with Speech Recognition in MATLAB using Correlation.

Functions Used :

1. Cross-correlation
xcorr(x, y)

syntax:

```
r = xcorr(x, y)
r = xcorr(x)
r = xcorr(___, maxlag)
r = xcorr(___, scaleopt)
[r, lags] = xcorr(___)
```

Description

r = xcorr(x, y) returns the cross-correlation of two discrete-time sequences. Cross-correlation measures the similarity between a vector x and shifted (lagged) copies of a vector y as a function of the lag. If x and y have different lengths, the function appends zeros to the end of the shorter vector so it has the same length as the other.

2. Scale data and play as sound : soundsc(y, Fs)

syntax:

```
soundsc(y)
soundsc(y, Fs)
soundsc(y, Fs, nBits)
soundsc(___, yRange)
```

Description

soundsc(y) scales the values of audio signal y to fit in the range from –1.0 to 1.0, and then sends the data to the speaker at the default sample rate of 8192 hertz. By first scaling the data, soundsc plays the audio as loudly as possible without clipping. The mean of the dynamic range of the data is set to zero.

soundsc(y,Fs) sends audio signal y to the speaker at sample rate Fs.

3. Read audio file : audioread()

syntax:

```
[y, Fs] = audioread(filename)
[y, Fs] = audioread(filename, samples)
[y, Fs] = audioread(___, dataType)
```

Description

[y, Fs] = audioread(filename) reads data from the file named filename, and returns sampled data, y, and a sample rate for that data, Fs.
[y, Fs] = audioread(filename, samples) reads the selected range of audio samples in the file, where samples is a vector of the form [start, finish].
[y, Fs] = audioread(___, dataType) returns sampled data in the data range corresponding to the dataType of 'native' or 'double', and can include any of the input arguments in previous syntaxes.

CODE :

```
%Speech Recognition Using Correlation Method
clc;
close all;
Speech_Recognition = 'test.wav'
voice=audioread(Speech_Recognition);
x=voice;
x=x';
x=x(1,:);
x=x';

%Following is the code for the database of .wav file samples

%Test Sample
yt=audioread('test.wav');
yt=yt';
yt=yt(1,:);
yt=yt';
zt=xcorr(x,yt);
mt=max(zt);
lt=length(zt);
tt=-((lt-1)/2):1:((lt-1)/2);
tt=tt';
subplot(3,2,6);
plot(tt,zt,'r');
title(Speech_Recognition);

%First Sample
y1=audioread('one.wav');
y1=y1';
y1=y1(1,:);
y1=y1';
z1=xcorr(x,y1);
m1=max(z1);
l1=length(z1);
t1=-((l1-1)/2):1:((l1-1)/2);
t1=t1';
subplot(3,2,1);


plot(t1,z1);
title('Sample-1');


%Second Sample
y2=audioread('two.wav');
y2=y2';
y2=y2(1,:);
y2=y2';
z2=xcorr(x,y2);
m2=max(z2);
l2=length(z2);
t2=-((l2-1)/2):1:((l2-1)/2);
t2=t2';
subplot(3,2,2);
```

```
plot(t2,z2);
title('Sample-2');

%Third Sample
y3=audioread('three.wav');
y3=y3';
y3=y3(1,:);
y3=y3';
z3=xcorr(x,y3);
m3=max(z3);
l3=length(z3);
t3=-((l3-1)/2):1:((l3-1)/2);
t3=t3';
subplot(3,2,3);
plot(t3,z3);
title('Sample-3');

%Fourth Sample
y4=audioread('four.wav');
y4=y4';
y4=y4(1,:);
y4=y4';
z4=xcorr(x,y4);
m4=max(z4);
l4=length(z4);
t4=-((l4-1)/2):1:((l4-1)/2);
t4=t4';
subplot(3,2,4);
plot(t4,z4);
title('Sample-4');

%Fifth Sample
y5=audioread('five.wav');
y5=y5';
y5=y5(1,:);
y5=y5';
z5=xcorr(x,y5);
m5=max(z5);
l5=length(z5);


t5=-((l5-1)/2):1:((l5-1)/2);
t5=t5';
subplot(3,2,5);
plot(t5,z5);


title('Sample-5');

m6=300;
a=[m1 m2 m3 m4 m5 m6];
m=max(a);

h=audioread('allow.wav');
if m<=m1
    soundsc(audioread('one.wav'),50000)
```

```
        soundsc(h,50000)
elseif m<=m2
    soundsc(audioread('two.wav'),50000)
        soundsc(h,50000)
elseif m<=m3
    soundsc(audioread('three.wav'),50000)
        soundsc(h,50000)
elseif m<=m4
    soundsc(audioread('four.wav'),50000)
        soundsc(h,50000)
elseif m<m5
    soundsc(audioread('five.wav'),50000)
        soundsc(h,50000)
else
   soundsc(audioread('denied.wav'),50000)

end
```

OUTPUT :