

APPENDIX A (The Main Program)

speaker.m

```

%=====
%                NEAR EAST UNIVERSITY SPEAKER RECOGNITION SYSTEM
%                =====
%
% This is the MATLAB speaker recognition program. First of all a number
% of sounds are stored in a database file. Then the program is used to
% recognise a particular speaker.
%
% The program allows the user to read ".wav" sound files, to play them,
% and also to plot their waveforms.
%
% The files are stored in a file called "SOUNDS.DAT" in the following
% format:
% (assuming there are two files in the database)
%
% File:s1.wav
% Location:c:\.....
% Sound ID: 1
% -
% File:s2.wav
% Location:c:\.....
% Sound ID: 2
% -
%
% The program is MENU based for simplicity and easy user interface.
% There are 10 options in the MENU. The MENU options are:
%
% OPTION 1 - Load a new sound file from disk
% OPTION 2 - Play a sound filr from disk
% OPTION 3 - Display a sound waveform from disk
% OPTION 4 - Display a sound waveform from database
% OPTION 5 - Display all sound waveforms in database at the same
%            time
% OPTION 6 - Speaker recognition
% OPTION 7 - Display Power Spectrum (Linear and Logarithmic)
% OPTION 8 - Display sound with and without windowing
% OPTION 9 - Sound database information
% OPTION 10 - Display information of a sound file in database
% OPTION 11 - Delete sound database
% OPTION 12 - Help
% OPTION 13 - Exit
%
%
% Program Developed By: Maysa Radwan
% Date                : April, 2010
% File                : speaker.m
% Version             : 1.0
% Modifications      :
%
%=====

```

```

options = 13;
sel = 0;
%
%
%           THE MAIN MENU
%          =====
%
%
while sel ~= options,
    sel = menu('NEAR EAST UNIVERSITY SPEAKER RECOGNITION SYSTEM',...
              'Load a new sound file from disk',...
              'Play a sound file from disk',...
              'Display a sound waveform from disk',...
              'Display a sound waveform from database',...
              'Display all sound waveforms in database at the same
time',...
              'Speaker recognition',...
              'Display sound power spectrum',...
              'Display sound with and without windowing',...
              'Sound database information',...
              'Display information of a sound file in the database',...
              'Delete sound database',...
              'Help',...
              'Exit');

%
%
% MENU OPTION 1 - LOAD A NEW SOUND FILE FROM DISK
% =====
%
%
    if sel == 1
        [filename,pathname] = uigetfile('*.wav','Select a new sound
file');
        [y, Fs, nbits] = wavread(strcat(pathname,filename));

        ID = input('Enter an ID that will be used for recognition:');

if(exist('C:\Users\MR\Documents\MATLAB\attachments_2010_04_19\SOUNDS.DAT')
== 2)

load('C:\Users\MR\Documents\MATLAB\attachments_2010_04_19\SOUNDS.DAT','-
mat');

        sound_no = sound_no + 1;
        data{sound_no,1} = y;
        data{sound_no,2} = ID;
        data{sound_no,3} = pathname;
        data{sound_no,4} = filename;

save('C:\Users\MR\Documents\MATLAB\attachments_2010_04_19\SOUNDS.DAT','data
','sound_no','-append');

```

```

        sound(y, Fs);
        msgbox('New sound added to database...');
    else
        sampling_freq = Fs;
        sampling_bits = nbits;
        sound_no = 1;
        data{sound_no,1} = y;
        data{sound_no,2} = ID;
        data{sound_no,3} = pathname;
        data{sound_no,4} = filename;

save('C:\Users\MR\Documents\MATLAB\attachments_2010_04_19\SOUNDS.DAT', 'data
', 'sound_no', 'sampling_freq', 'sampling_bits');
        sound(y, Fs);
        msgbox('New Sound added to database...');
    end
end

%
%
%
% MENU OPTION 2 - PLAY A SOUND FILE FROM DISK
% =====
%
%
%
%
if sel == 2
    [filename,pathname] = uigetfile('*.wav');
    [y, Fs, nbits] = wavread(strcat(pathname,filename));
    wavplay(y,Fs);
end

%
%
%
% MENU OPTION 3 - DISPLAY A SOUND WAVEFORM FROM DISK
% =====
%
%
%
%
if sel == 3
    [filename,pathname] = uigetfile('*.wav');
    [y, Fs, nbits] = wavread(strcat(pathname,filename));
    t = 0:1/Fs:length(y)/Fs-1/Fs;
    plot(t,y)
    xlabel('Time (secs)');
    ylabel('Amplitude');
    title (filename);
end

%

```

```

%
%
% MENU OPTION 4 - DISPLAY A SOUND WAVEFORM FROM DATABASE
% =====
%
%
%
%
if sel == 4
    IDD = input('Enter the ID no of the sound: ');

load('C:\Users\MR\Documents\MATLAB\attachments_2010_04_19\SOUNDS.DAT','-
mat');

    for ii=1:sound_no
        id = data{ii,2};
        if (IDD == id)
            filename = data{ii,4};
            pathname = data{ii,3};
            [y, Fs, nbits] = wavread(strcat(pathname,filename));
            wavplay(y,Fs);
            t = 0:1/Fs:length(y)/Fs-1/Fs;
            plot(t,y);
        end
    end
end

%
%
% MENU OPTION 5 - DISPLAY ALL SOUND WAVEFORMS IN DATABASE AT THE SAME
% TIME
% =====
%
%
%
%
if(sel == 5)
    for ii=1:sound_no
        id = data{ii,2};
        filename = data{ii,4};
        pathname = data{ii,3};
        [y, Fs, nbits] = wavread(strcat(pathname,filename));
        t = 0:1/Fs:length(y)/Fs-1/Fs;
        subplot(4,3,ii); plot(t,y); xlabel('Time (secs)');
ylabel('Amplitude'); title (filename);
    end
end

%
%
%

```

```

% MENU OPTION 6 - SPEAKER RECOGNITION
% =====
%
%
%
%
if sel == 6,
    clc;
    [filename,pathname] = uigetfile('*.wav','Select a new sound file');
    [y, Fs, nbits] = wavread(strcat(pathname,filename));

    disp('Sound file selected for recognition: ');
    msg = strcat('File:',filename);
    disp(msg);
    msg=strcat('Location:',pathname);
    disp(msg);

if(exist('C:\Users\MR\Documents\MATLAB\attachments_2010_04_19\SOUNDS.DAT')
== 2)

load('C:\Users\MR\Documents\MATLAB\attachments_2010_04_19\SOUNDS.DAT','-
mat');

    disp(' ');
    disp('NEAR EAST UNIVERSITY-SPEAKER RECOGNITION SYSTEM');
    disp('=====');
    disp(' ');
    disp('START OF SPEAKER RECOGNITION');
    disp('Compute MFCC coefficients for each sound in the
Database...');

    k = 16;
    for ii=1:sound_no
        %
        % Create a matrix M containing all the frames
        %
        disp(' ');
        message = strcat('CREATE MATRIX CONTAINING ALL THE FRAMES
- Sound ',num2str(ii));
        disp(message);
        m = Cmatrix(data{ii,1}, Fs);

        %
        % Training code
        %
        code{ii} = vq1bg(m, k);
    end
    disp('Database part completed...');
    %
    % Now compute MFCC coefficients for the input sound
    %
    m = Cmatrix(y, Fs);
    %
    % Distance and sound initialization
    %
    mindist = Inf;

```

```

k1 = 0;
for ii=1:sound_no
    d = disteu(m, code{ii});

    dist = sum(min(d, [], 2)) / size(d, 1);
    if dist < mindist
        mindist = dist;
        k1 = ii;
    end
end
min_index = k1;

%
% Match is found
%
speech_id = data{min_index, 2};

disp(' ');
disp('A MATCHING speaker is found...');
disp(' ');
message = strcat('Filename: ', data{min_index, 4});
disp(message);
message = strcat('Location: ', data{min_index, 3});
disp(message);
message = strcat('Recognised speaker ID is: ',
num2str(speech_id));
disp(message);
end
end

%
%
%
% MENU OPTION 7 - DISPLAY SOUND POWER SPECTRUM (Linear and Logarithmic)
% =====
%
%
%
if sel == 7,
    clc;

load('C:\Users\MR\Documents\MATLAB\attachments_2010_04_19\SOUNDS.DAT', '-
mat');
[filename,pathname] = uigetfile('*.wav','Select a new sound file');
[y, Fs, nbits] = wavread(strcat(pathname,filename));

m = pspectrum(y, Fs);
end

%

```

```

%
%
% MENU OPTION 8 - DISPLAY SOUND WITH AND WITHOUT WINDOWING
% =====
%
%
%
if sel == 8,
    clc;

load('C:\Users\MR\Documents\MATLAB\attachments_2010_04_19\SOUNDS.DAT','-
mat');
    [filename,pathname] = uigetfile('*.wav','Select a new sound file');
    [y, Fs, nbits] = wavread(strcat(pathname,filename));

    m = compwind(y, Fs);
end

%
%
% MENU OPTION 9 - SOUND DATABASE INFORMATION
% =====
%
%
%
if sel == 9,
    if
    (exist('C:\Users\MR\Documents\MATLAB\attachments_2010_04_19\SOUNDS.DAT') ==
2)

load('C:\Users\MR\Documents\MATLAB\attachments_2010_04_19\SOUNDS.DAT','-
mat');
        clc;
        message = strcat('Database SOUNDS.DAT has # ',num2str(sound_no),'
sounds:');
        disp(message);
        disp(' ');
        for ii=1:sound_no
            message = strcat('File:',data{ii,4});
            disp(message);
            message = strcat('Location:', data{ii,3});
            disp(message);
            message = strcat('Sound ID:',num2str(data{ii,2}));
            disp(message);
            disp('-');
        end
    else
        warndlg('Database SOUNDS.DAT is empty',' Warning ')
    end
end

%

```

```

%
%
% MENU OPTION 10 - DISPLAY INFORMATION OF A SOUND FILE IN DATABASE
% =====
%
%
%
%
%
%
if sel == 10
    clc;
    IDD = input('Enter ID no of the sound file to be displayed: ');
    disp(' ');

    for ii=1:sound_no
        id = data{ii,2};
        if(IDD == id)
            filename = data{ii,4};
            pathname = data{ii,3};
            message = strcat('Sound ID = ',num2str(id));
            disp(message);
            message = strcat('File: ',filename);
            disp(message);
            message = strcat('Location: ',pathname);
            disp(message);
        end
    end
end

%
%
% MENU OPTION 11 - DELETE SOUND DATABASE
% =====
%
%
%
%
%
%
if sel == 11
    clc;
    close all;

if(exist('C:\Users\MR\Documents\MATLAB\attachments_2010_04_19\SOUNDS.DAT')
== 2)
    button = questdlg('Do you want to delete the SOUNDS.DAT Database
?');
    if strcmp(button,'Yes')

delete('C:\Users\MR\Documents\MATLAB\attachments_2010_04_19\SOUNDS.DAT');
    msgbox('SOUNDS.DAT database deleted...');
    end
else
    warndlg('Database is already empty.', ' Warning')
end
end
end

```



```

%
%
%
% MENU OPTION 12 - HELP DISPLAY
% =====
%
%
%
if sel == 12
    clc;
    fprintf('\nThis is the MATLAB Speaker Recognition Program
help.\n\n');
    fprintf('The program has been developed as part of an MSc Thesis in
Computer Engineering at the Near East University.\n');
    fprintf('The program recognises a particular speaker from a number of
speakers. The program is MENU based with the following options:\n\n');
    fprintf('OPTION 1: This option loads a new sound file from disk. A
dialog box is displayed where the user is expected\n');
    fprintf('                to select the directory and name of
the .wav sound file to be loadd into the sound database\n');
    fprintf('OPTION 2: This option plays a sound file from the disk. A
dialog box is diplayed where the user can select\n');
    fprintf('                the directory and the filename of the
sound file to be played\n');
    fprintf('OPTION 3: This option displays a sound waveform from the
disk. The filename of the file to be displayed is selected\n');
    fprintf('                by the user from a dialogue box\n');
    fprintf('OPTION 4: This option displays a sound waveform from the
database. The filename of the file to be displayed is selected\n');
    fprintf('                by the user from a dialogue box\n');
    fprintf('OPTION 5: This option displays all of the sound files at
the same time in the same display. This option is very useful when it\n');
    fprintf('                required to compare various sound file
waveforms\n');
    fprintf('OPTION 6: This option performs the speaker recognition
function where the speaker is identified among several sound\n');
    fprintf('                files. This is the main part of this
program and also the main topic of this Thesis\n');
    fprintf('OPTION 7: This option displays the power spectrum (linear
and logarithmic of a sound file\n');
    fprintf('OPTION 8: This option displays sound with and without
windowing\n');
    fprintf('OPTION 9: This option displays information about all of
the sound files in the sound database. There are no options for\n');
    fprintf('                the user to select\n');
    fprintf('OPTION 10: This option displays information about a single
sound file in the sound database\n');
    fprintf('OPTION 11: This option is used to delete the sound file
database. When this option is selected the user is given the choice\n');
    fprintf('                of accepting or rejecting the
deletion. If accepted, all of the sound files will be deleted from the
sound database\n');

```

```
        fprintf('OPTION 12:  This option just displays this HELP text\n');
        fprintf('OPTION 13:  This option is used to terminate the
program\n');
    end

%
%
%
% MENU OPTION 13 - EXIT OF PROGRAM
% =====
%
%
%
%
    if sel == 13
        fprintf('End of program...\n');
    end

end

%
%
% ===== END OF PROGRAM =====
%
```