



**DEPARTMENT OF COMPUTER & SOFTWARE
ENGINEERING
COLLEGE OF E&ME, NUST, RAWALPINDI**



Signals and Systems

Project Report

SUBMITTED TO:

LE Mudassir

SUBMITTED BY:

Muhammad Taimoor (355134)

Uzair Sultan (331638)

M. Abdullah bin Nisar (359079)

CE-42 B

Submission Date: 16-01-2023

VOICE CHANGER APPLICATION IN MATLAB

Objective:

As part of our lab requirements for the course of “Signals and Systems”, we were required to develop a project for it. We decided to go with a “Voice Converter” application in MATLAB. This report will describe its features, as well as source code.

Algorithm:

We performed the following steps to accomplish this task:

1. Take a voice sample (recorded at run-time or prerecorded).
2. Identify the gender of the voice sample by calculating its mean pitch frequencies.
3. If the mean is lesser than 165, the voice is male. (Male characteristic frequency is 85-155 Hz)
4. If it is more than 165, it is female. (Female characteristic frequency is 165-255 Hz)
5. We then shift the pitch of the given voice sample so that the frequency falls in the male or female characteristic frequency range.

Procedure to accomplish it in MATLAB:

1. We allowed the user to be able to record his voice on run-time for 5 seconds or upload a voice sample using a .wav file in MATLAB.
2. Once we have read the data from the audio sample, we will apply the pitch() function to our read data from the audio file.
3. The pitch function will return an array of frequencies. We will take the mean of this frequency array.
4. If the mean is less than 165, the voice is considered male. Else, it is considered female.
5. Once the gender has been identified, we will first use the Short Time Fourier Transform (STFT) on our voice sample.
6. We will then shift our voice using the shiftPitch() function. We will specify the number of semitones to shift depending on the result of our mean frequency of voice sample.
7. Now we will apply pitch() function on the shifted data. This will return an array frequency.
8. We will take the mean of this array. If the range falls between the characteristic male or female frequency range (depending on the gender of the voice sample), this means our shifting was successful.
9. This data will always be saved to a file by the name of **‘ShiftedVoice.wav’**

Note:

1. In the case, if one decides to record their own voice, the data will be saved to a file by the name of **‘OriginalVoice.wav’**.
2. For one to be able to upload files to this application, they must make sure that their files are present in the same directory as their project.

Code:

```
function varargout = VoiceChanger(varargin)
% VOICECHANGER MATLAB code for VoiceChanger.fig
%     VOICECHANGER, by itself, creates a new VOICECHANGER or raises the existing
%     singleton*.
%
%     H = VOICECHANGER returns the handle to a new VOICECHANGER or the handle to
%     the existing singleton*.
%
%     VOICECHANGER('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in VOICECHANGER.M with the given input arguments.
%
%     VOICECHANGER('Property','Value',...) creates a new VOICECHANGER or raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before VoiceChanger_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to VoiceChanger_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help VoiceChanger

% Last Modified by GUIDE v2.5 15-Jan-2023 01:11:00

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @VoiceChanger_OpeningFcn, ...
                  'gui_OutputFcn',  @VoiceChanger_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before VoiceChanger is made visible.
function VoiceChanger_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to VoiceChanger (see VARARGIN)

% Choose default command line output for VoiceChanger
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
```

```

% UIWAIT makes VoiceChanger wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = VoiceChanger_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
recorder=audiorecorder(8000,16,1); %Had to change Fs to 8000
% because it didn't work otherwise
filename='OriginalVoice.wav';
pause(2);
set(handles.prompt,'String','Started recording...');
recordingblocking(recorder,5);
set(handles.prompt,'String','Stopped recording...');
myrec=getaudiodata(recorder);
audiowrite(filename, myrec,8000); %Writes our audio data in .wav file
clear y Fs;
[y,Fs] = audioread(filename);
f0=pitch(y,Fs); %Get the frequency array of the sound signal.
fa=mean(f0,'all'); %Take mean of all values in the frequency array
set(handles.edit1,'String',num2str(fa)); %Displaying original frequency values in a text box
if fa>165
    set(handles.text9,'String','This is a female voice');
else
    set(handles.text9,'String','This is a male voice');
end
sound(y,Fs); %Play the data
info=audioinfo(filename); %Provides us info about the .wav file

%Creates a vector of the same length
%as y, this represents the time elapsed
t = 0:seconds(1/Fs):seconds(info.Duration);
t = t(1:end-1);

%Plot the signal
plot(t,y);
legend('Original Voice Graph');

handles.info=info; %Information necessary to plot a graph of our original sound signal
handles.y=y; %Audio data saved into GUI handles
handles.Fs=Fs; %Sample frequency saved into GUI handles.
handles.myrec=myrec; %Storing the audiorecorder object in GUI handles
handles.filename=filename; %Storing filename in GUI handles.
handles.fa=fa; %Storing mean frequency in GUI handles
handles.t=t; %Storing time in handles GUI
guidata(hObject,handles); %Changing the state of our GUI handles.

```

```

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
filename=uigetfile('*.wav'); %Filtering by .wav files
if isequal(filename,0)
    set(handles.prompt,'String','No file selected...');
else
    [y,Fs] = audioread(filename); %Reading data from our selected .wav file
    f0=pitch(y,Fs); %Get the frequency array of the sound signal.
    fa=mean(f0,'all'); %Take mean of all values in the frequency array
    set(handles.edit1,'String',num2str(fa)); %Displaying original frequency values in a text box
    if fa>165
        set(handles.text9,'String','This is a female voice'); %Display if mean is greater than 165
    else
        set(handles.text9,'String','This is a male voice');
    end
    sound(y,Fs); %Play the data
    info=audioinfo(filename); %Provides us info about the .wav file

%Creates a vector of the same length
%as y, this represents the time elapsed
t = 0:seconds(1/Fs):seconds(info.Duration);
t = t(1:end-1);

%Plot the signal
plot(t,y)
legend('Original Voice Graph');

handles.info=info; %Information necessary to plot a graph of our original sound signal
handles.y=y; %Audio data saved into GUI handles
handles.Fs=Fs; %Sample frequency saved into GUI handles.
handles.filename=filename;
handles.fa=fa; %Storing frequency in GUI handles.
guidata(hObject,handles); %Changing the state of our GUI handles.
end

function edit1_Callback(hObject, eventdata, handles)
% hObject      handle to edit1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

end

```
function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a double

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
y=handles.y;
Fs=handles.Fs;
fa=handles.fa;
win = kbdwin(512);
overlapLength = 0.75*numel(win);
lockPhase=false;
S = stft(y, ...
    "Window",win, ...
    "OverlapLength",overlapLength, ...
    "Centered",false);
if (fa(1)>=85 && fa(1)<=100)
    semitones=10;
else if (fa>=100 && fa<=120)
    semitones=8;
else if (fa>=120 && fa<=140)
    semitones=7;
else if (fa>=140 && fa<=165)
    semitones=6;
else if (fa>=165 && fa<=200)
    semitones=-6;
else if (fa>=200 && fa<=220)
    semitones=-12;
else if (fa>=220 && fa<=255)
    semitones=-12;
end
end
end
end
end
end
end
```

```

end
audioOut = shiftPitch(S,semitones, ...
    "Window",win, ...
    "OverlapLength",overlapLength, ...
    "LockPhase",lockPhase);
f0_shifted=pitch(audioOut,Fs);
f0_shifted=mean(f0_shifted);
set(handles.edit2,'String',f0_shifted(1));
audiowrite("ShiftedVoice.wav",audioOut,Fs);

info2=audioinfo("ShiftedVoice.wav");
t = 0:seconds(1/Fs):seconds(info2.Duration);
t = t(1:end-1);
plot(t,audioOut);
legend('Shifted Voice Graph');

%pause(5);
soundsc(audioOut,Fs)
handles.audioOut=audioOut;
guidata(hObject,handles);

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject     handle to popupmenu1 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1 contents as cell array
%         contents{get(hObject,'Value')} returns selected item from popupmenu1

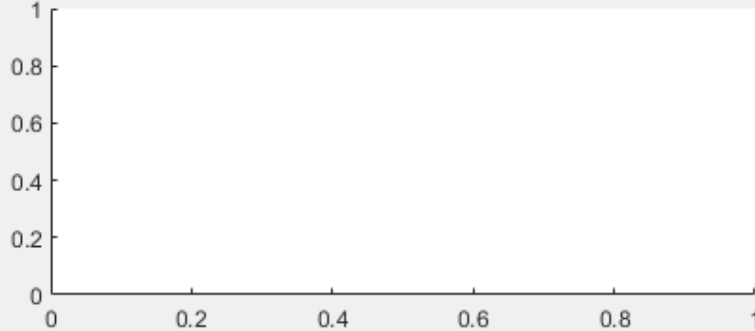
% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject     handle to popupmenu1 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called
audioOut=handles.audioOut;
y=handles.y;
val = get(hObject,'Value');
string_list = get(hObject,'String');
selected_string = string_list{val}; % convert from cell array to string
if (strcmp(selected_string,"Original Plot")==1)
    plot(y,t);
else if (strcmp(selected_string,"Shifted Plot")==1)
    plot(audioOut,t);
end
end

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

GUI:

OR

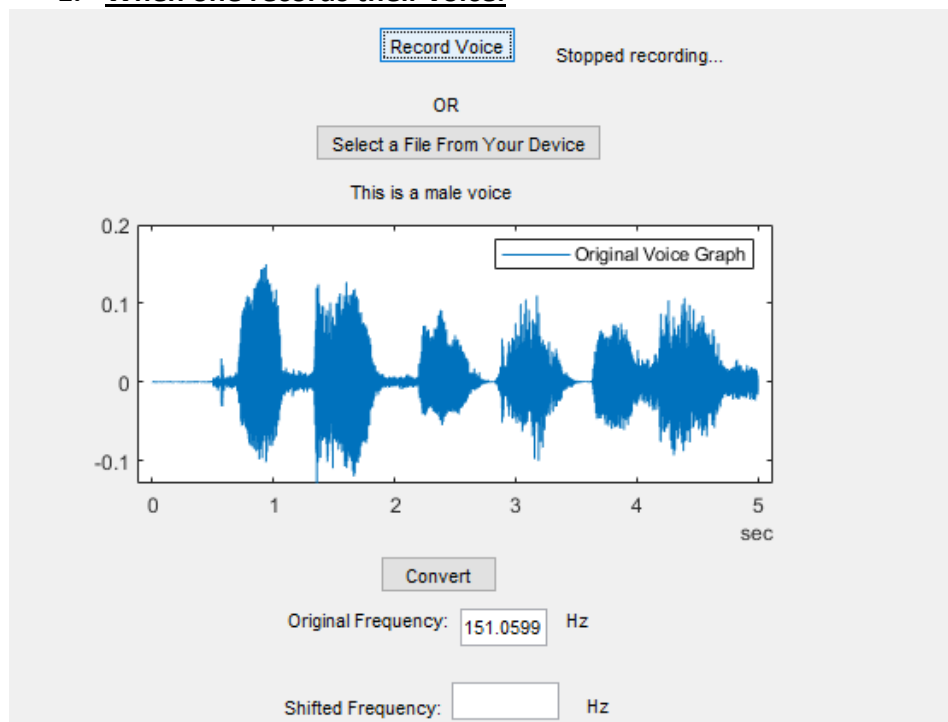


Original Frequency: Hz

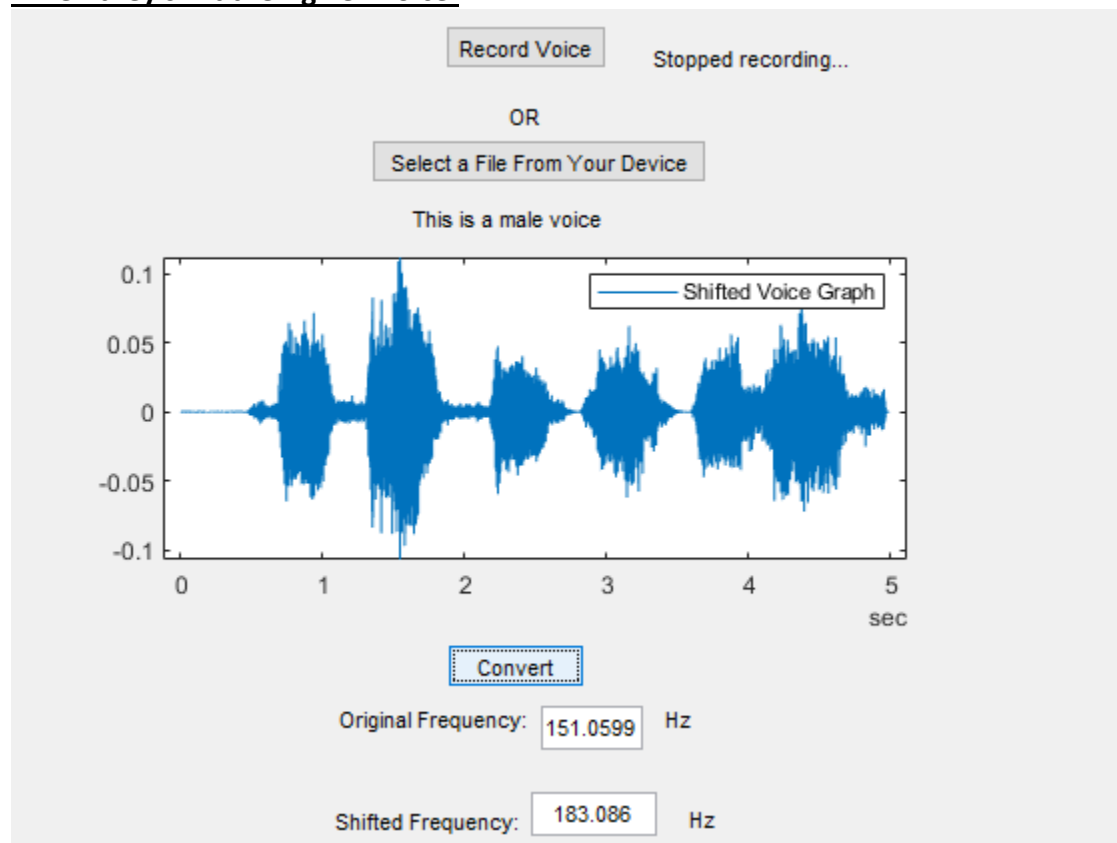
Shifted Frequency: Hz

Output when "Record Voice" is pressed:

1. When one records their voice.

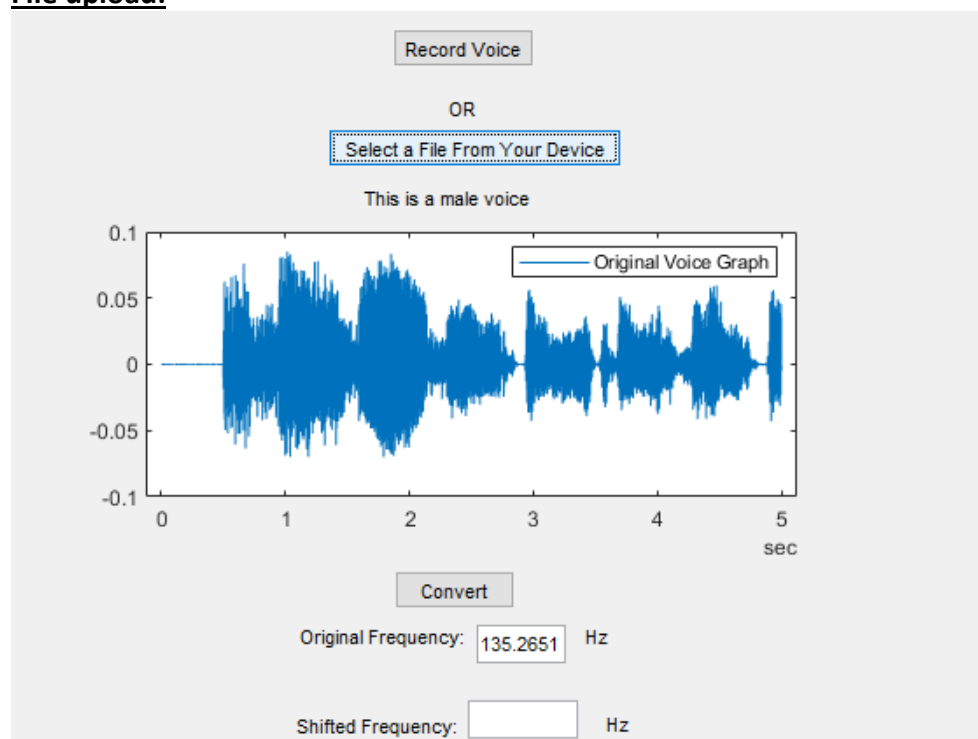


2. When they shift their given voice:



When a user decides to upload a prerecorded file:

1. File upload:



After shifting the voice:

