

EE430 Project Report

Part-1

Ahmet Alp Güneş 2443190

Nebile Şevval Karahan 2443299

Introduction

This project focuses on developing a MATLAB program that computes and displays a signal's Short-Time Fourier Transform (STFT), also called a Time-Dependent Fourier Transform, which allows temporal frequency content analysis. The program has an easy-to-use interface that makes use of the MATLAB App Designer, allowing users to process audio files, create synthetic signals with configurable parameters, and gather sound data from a microphone.

Data Acquisition

I. Recording

GUI EXPLANATION

The app we created allows the user to record and listen to audio in this part of the project. They can also see the audio signal that they recorded. The user is able to choose the length of the audio that will be recorded by typing the duration in seconds into the "Audio Length" edit field, and they can adjust the sampling frequency by dragging the "Fs" slide in Figure 1.

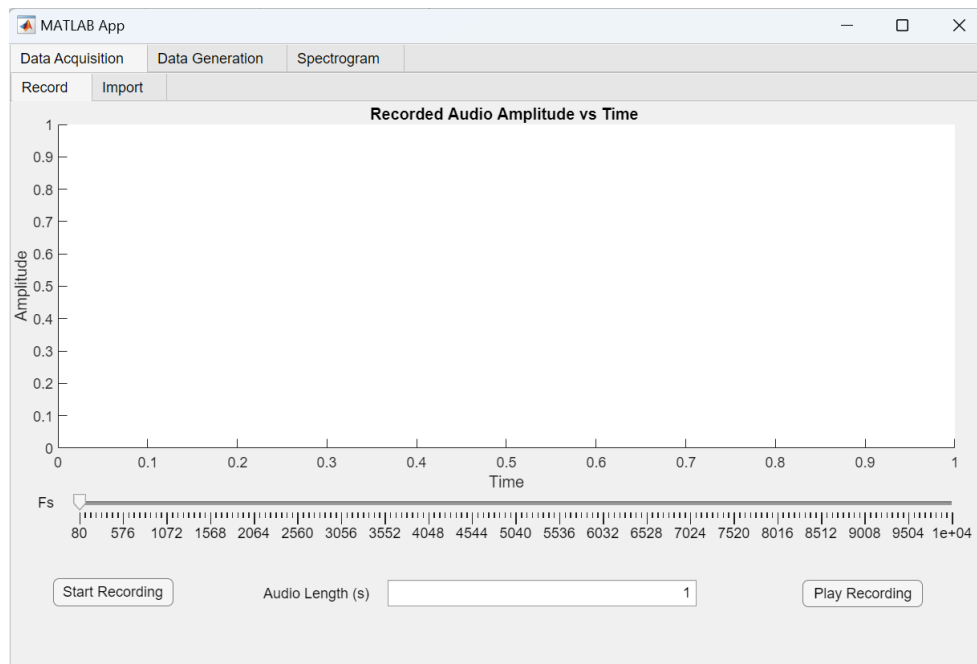


Figure-1: GUI for Recording and Playing Audio Signal

As can be seen in the Figure-2, after recording is finished, the time domain signal will be plotted on the axes on the GUI under the tab Data Acquisition/Record.

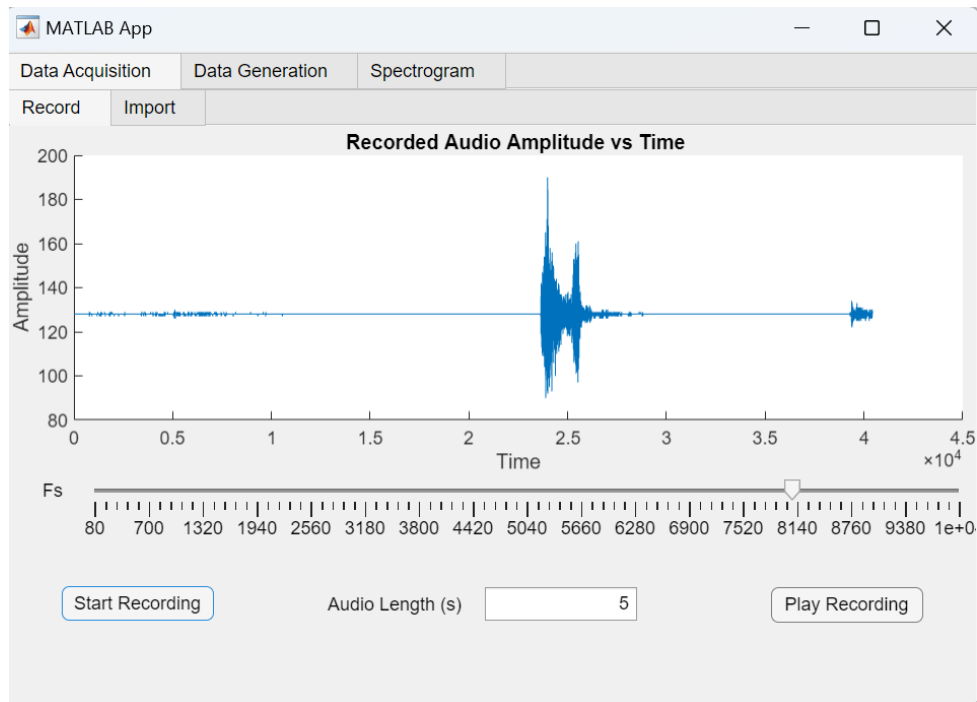


Figure-2: Time Domain Signal Plotting

MATLAB CODE EXPLANATION

The MATLAB code provided in the Figure-3 (also provided in appendix as text) is an App Designer code for implementing the GUI in the previous part of the report. There are two buttons, one slider and lastly one integer edit field in the Record tab of the GUI.

```

74 % Callbacks that handle component events
75 methods (Access = private)
76
77 % Value changing function: FsSlider
78 function FsSliderValueChanged(app, event)
79     app.Fs = event.Value;
80 end
81
82 % Button pushed function: StartRecordingButton
83 function StartRecordingButtonPushed(app, event)
84     app.Recorder = audiorecorder(app.Fs,24,1);
85     recordblocking(app.Recorder,app.Length);
86     x=getaudiodata(app.Recorder,'uint8');
87     plot(app.UIAxes,x);
88 end
89
90 % Value changed function: AudioLengthsEditField
91 function AudioLengthsEditFieldValueChanged(app, event)
92     app.Length = app.AudioLengthsEditField.Value;
93 end
94
95 % Button pushed function: PlayRecordingButton
96 function PlayRecordingButtonPushed(app, event)
97     play(app.Recorder)
98 end

```

Figure 3: MATLAB Code for Record Tab of the GUI

In order to use the sample frequency that the user selects on the slider when plotting the time domain signal, we store this value in a property at line 79. This code implements the 'Start Recording' button between lines 84 and 87. Here we use the 'audiorecorder' function of the MATLAB that can collect sound data from the device's microphone. Next, we use the built-in "recordblocking" feature to end the recording. This function terminates the recording process after recording the sound in whatever duration the user selected on the audio length edit field. Following that, we get the sound data using 'getaudiodata' function from the record and plot it on the axes we placed on the GUI. To be able to use the value entered by the user in the audio length edit field in the "recordblocking" function, line 92 stores it in a property called Length. Finally, pressing the play button on the app allows the user to listen to the previously recorded data.

II. Importing an Audio File

In order to import and play a '.wav' file, the user should open the Import tab, in Figure 4, under the Data Acquisition tab. There are some restrictions about the file to be imported. First, as we indicated before, the file must be a '.wav' file. Also, this file must be in the same folder as the app files if not, the file name must be written with the file path. After these conditions are satisfied, one can play the audio and plot display the time domain signal on the axes by entering the file name into the edit field.

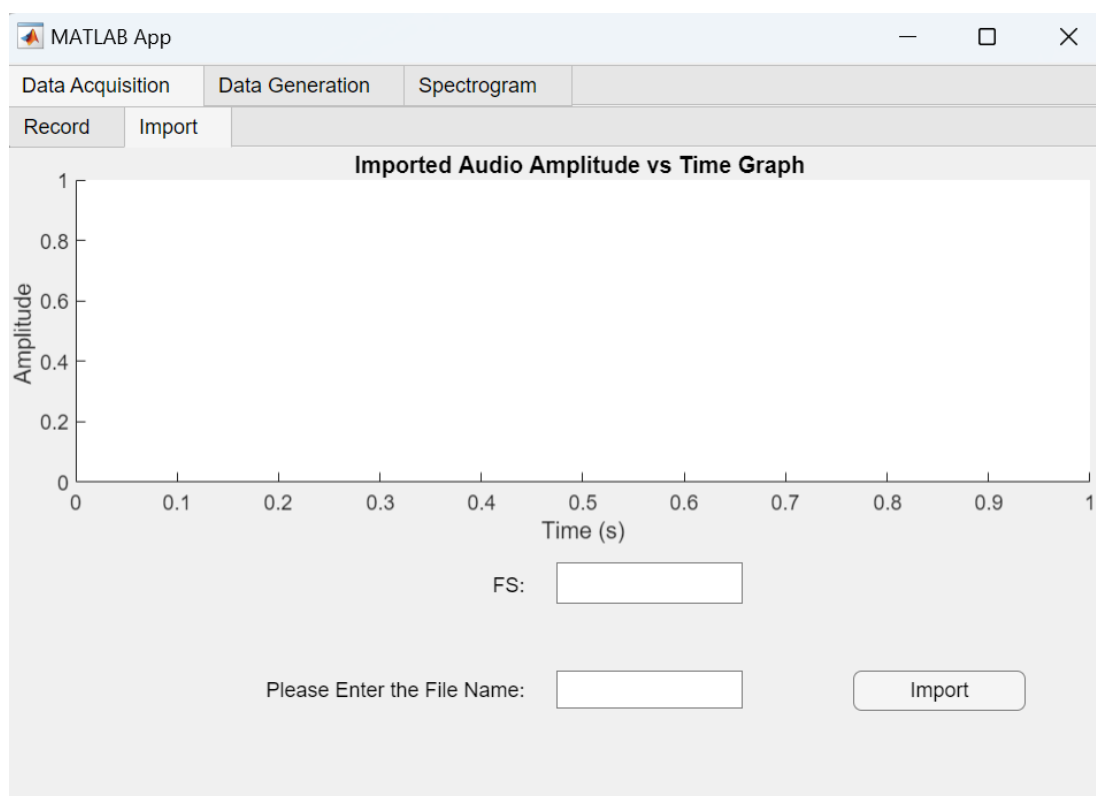


Figure 4: File Importing GUI

After playing the audio is finished time domain signal will be visible on the axes as in the Figure 5.

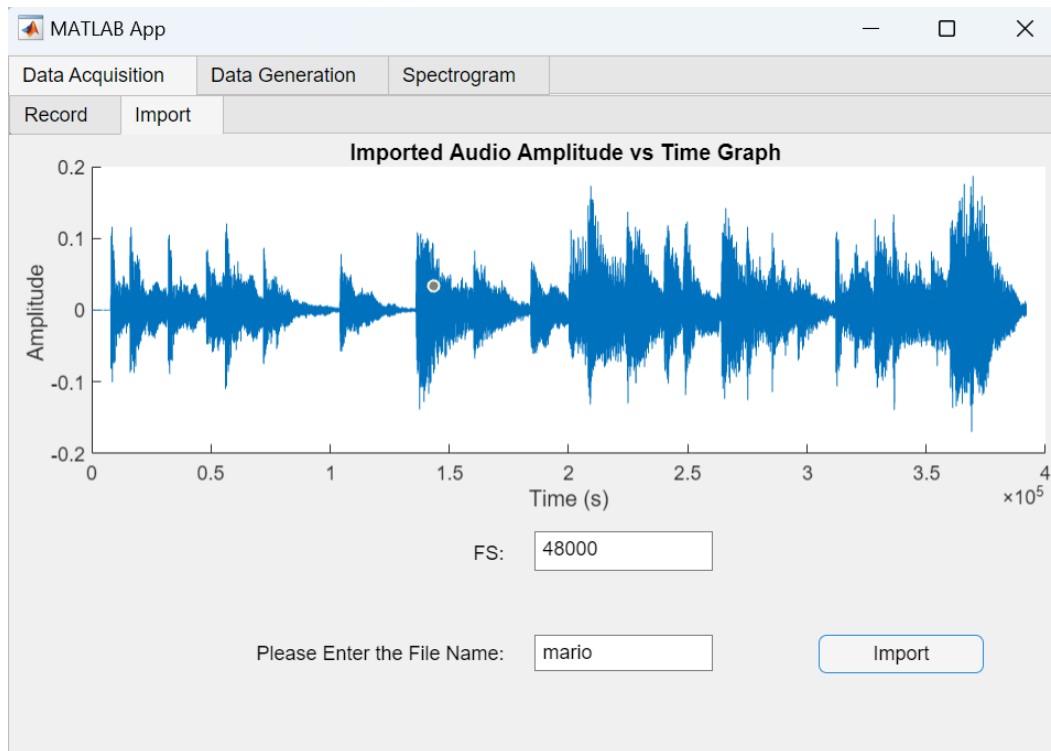


Figure 5: Plotting the Imported File

MATLAB CODE EXPLANATION

In this part of the project, as can be seen from the Figure 6, we simply read the file name from the text edit field and read this file by using the 'audioread' function of the MATLAB. File name is stored in a property named File Name to be able to use the name in the 'audioread' function. After playing the sound, time domain signal will be plotted on the axes.

```
322 % Value changed function: PleaseEntertheFileNameEditField
323 function PleaseEntertheFileNameEditFieldValueChanged(app, event)
324     app.FileName = app.PleaseEntertheFileNameEditField.Value;
325 end
326
327 % Button pushed function: ImportButton
328 function ImportButtonPushed(app, event)
329     [y,app.Fs] = audioread(app.FileName + ".wav");
330     sound(y, app.Fs)
331     app.FSTextArea.Value = sprintf('%2.f',app.Fs);
332     plot(app.UIAxes_2, y);
333 end
```

Figure 6: MATLAB Code for File Importing Tab

Data Generation

I. Sinusoid Signal

In this section, we are asked to generate a sinusoid signal that has changeable amplitude, frequency, and phase by the user. In this GUI one generates a sine signal by entering the parameters indicated in the Figure 7.

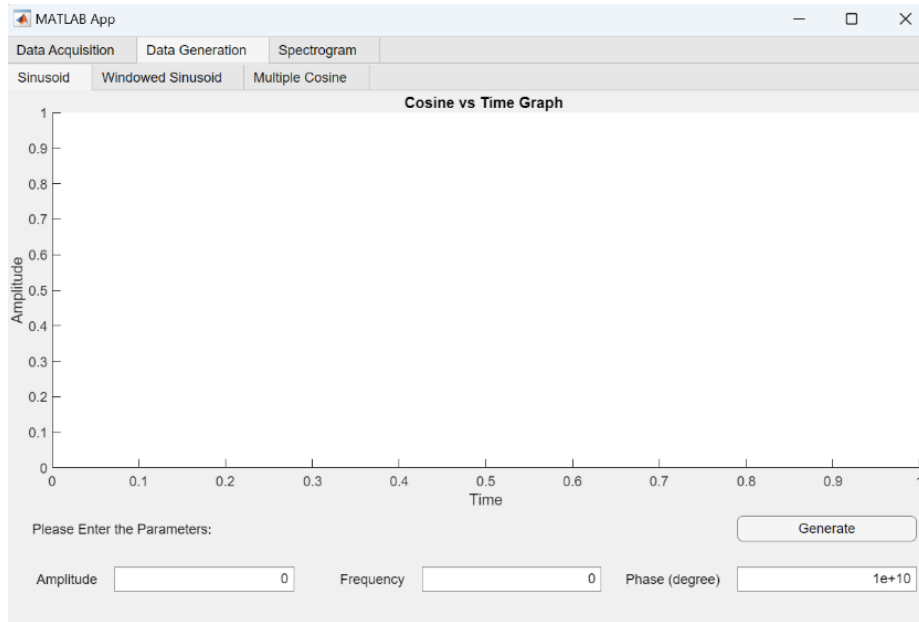


Figure 7: Sinusoid Generation GUI

After pushing the generate button, one can see the generated sine signal plot in the axes, as can be seen in the Figure 8.

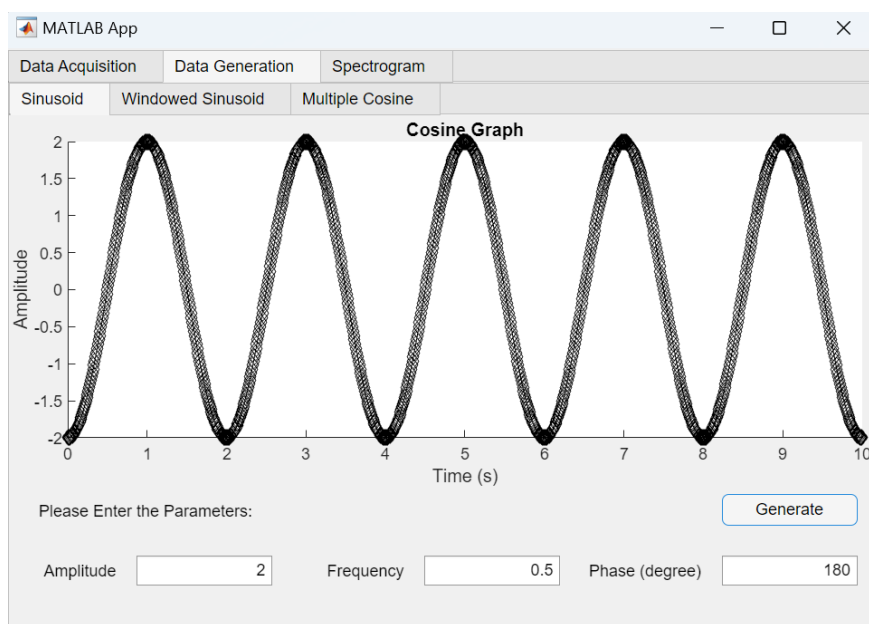


Figure 8: Generated Sine Signal Plot

MATLAB CODE EXPLANATION

In this part, we created a cosine signal. To create that signal, we took phase, frequency, and amplitude. Then, we simply created a cosine signal using built-in cos function.

```
145     t = 0:(1/100):10;  
146     phase_rad = app.Phase * pi /180;  
147     angle=2*pi*app.Frequency*t ;  
148     x = app.Amplitude * cos(angle+phase_rad);  
149     plot(app.UIAxes2,t, x, '-kd');  
150     grid on
```

Figure 9: MATLAB Code for Sinusoidal Signal Generation

II. Windowed Sinusoid Signal

By using our GUI, a windowed cosine signal can be plotted. As parameter, user must give amplitude of signal, phase of signal, frequency of signal and starting time of signal. User also must give a window length, then select a window from given choices as can be seen in Figure 10. Then, windowed sinusoidal signal will be plotted as in the Figure 11.

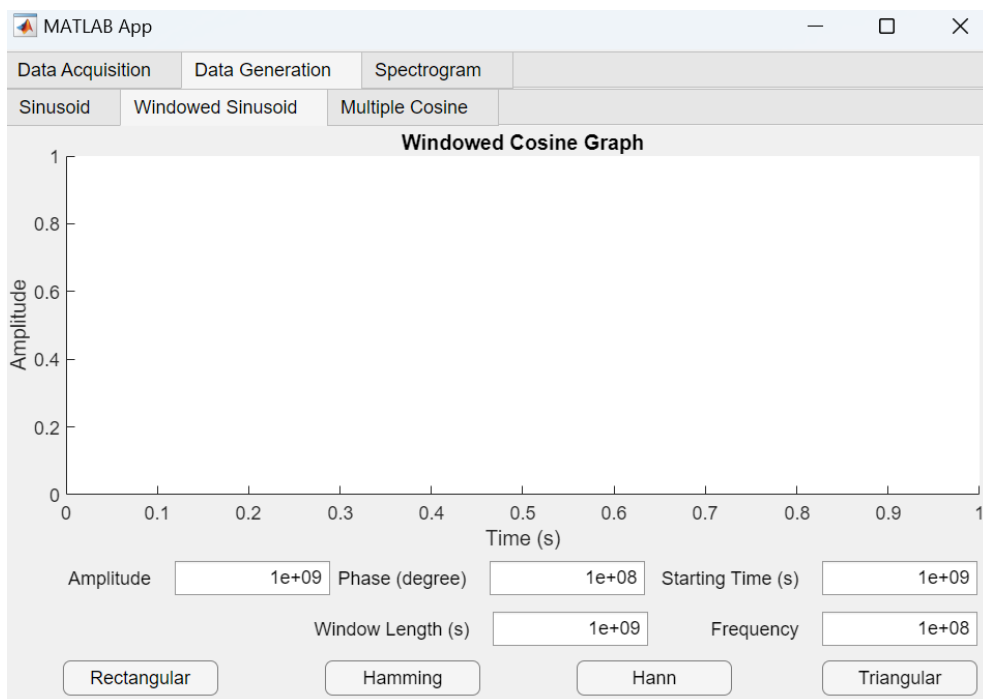


Figure 10: Generation of Windowed Cosine Signal GUI

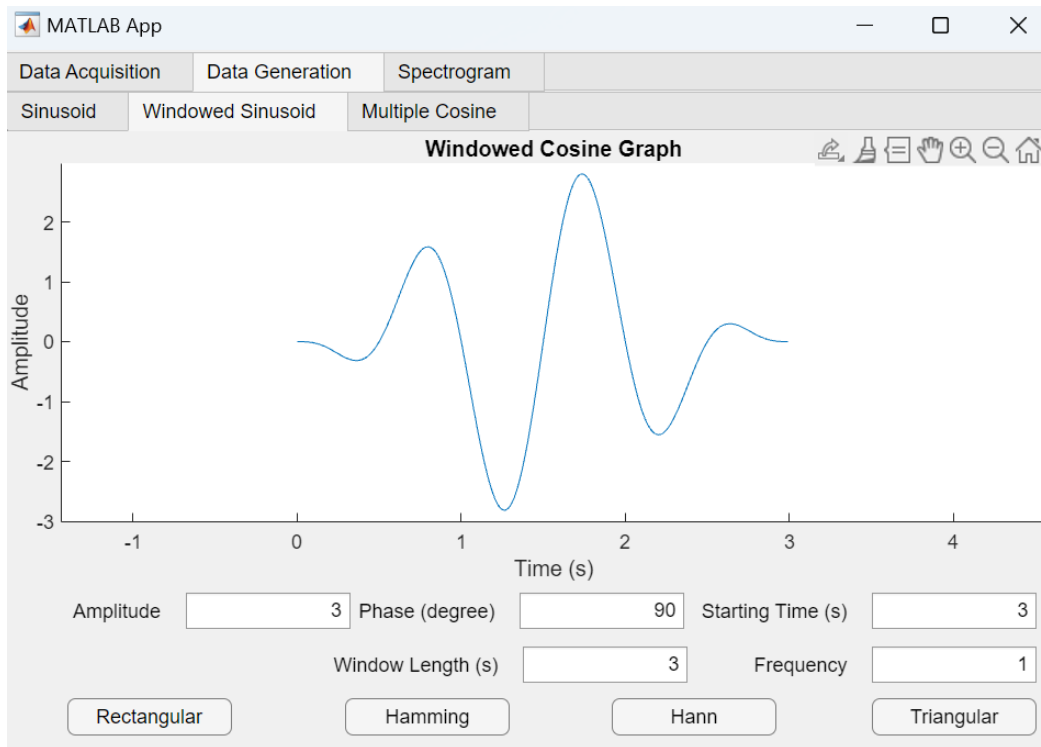


Figure 11: Generated Windowed Cosine Signal with Hann Window

MATLAB CODE EXPLANATION

In this part, we created a cosine signal with a shift. Then, we multiplied that signal with a window to filter it. Finally, we plotted the output signal.

```

219 function HannButtonPushed(app, event)
220     t = 0:(1/100):10;
221     phase_rad = app.Phase * pi / 180;
222
223     shifted_sine = app.Amplitude * cos(2*pi*app.Frequency*(t-app.StartingTime)+phase_rad);
224     win_length_samples = round(app.WindowLength * 100);
225     hann_window = hann(win_length_samples)';
226     windowed_signal = shifted_sine(1:win_length_samples) .* hann_window;
227
228     plot(app.UIAxes2_2,t(1:win_length_samples), windowed_signal);
229

```

Figure 12: MATLAB Code for Generating Cosine Signal with Hann Window

In the other three window functions, we only changed the line 225 in Figure 12 to change the window function, i.e. `rectangular_window = rectwin(win_length_samples)'`; for the rectangular window.

III. Signal Involving Multiple Components

In this part, we created multiple cosine signals, then summed them up. User firstly selects the number of cosine signals to be summed up. User can select number of cosine signals up to 4 in our

design. After that, user can select phase, amplitude, and frequency for each signal as in Figure 13. Finally, each cosine signal will be created and summed up. By pressing generate button, user can plot the summed signal as in Figure 14.

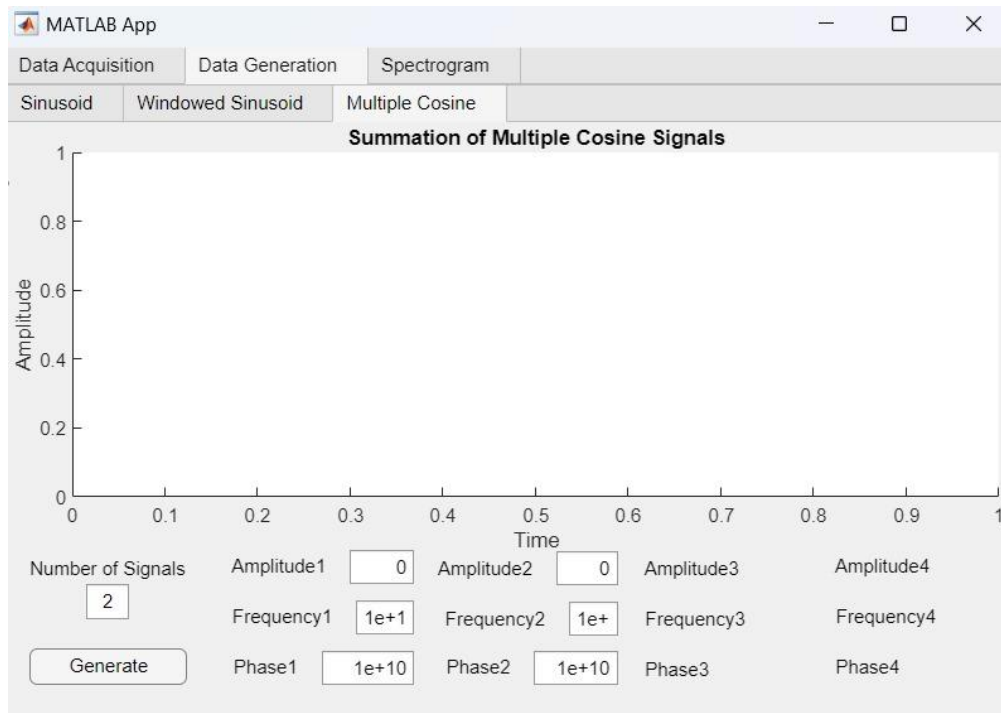


Figure 13: Generation of Multiple Cosine Signal GUI with 2 Components

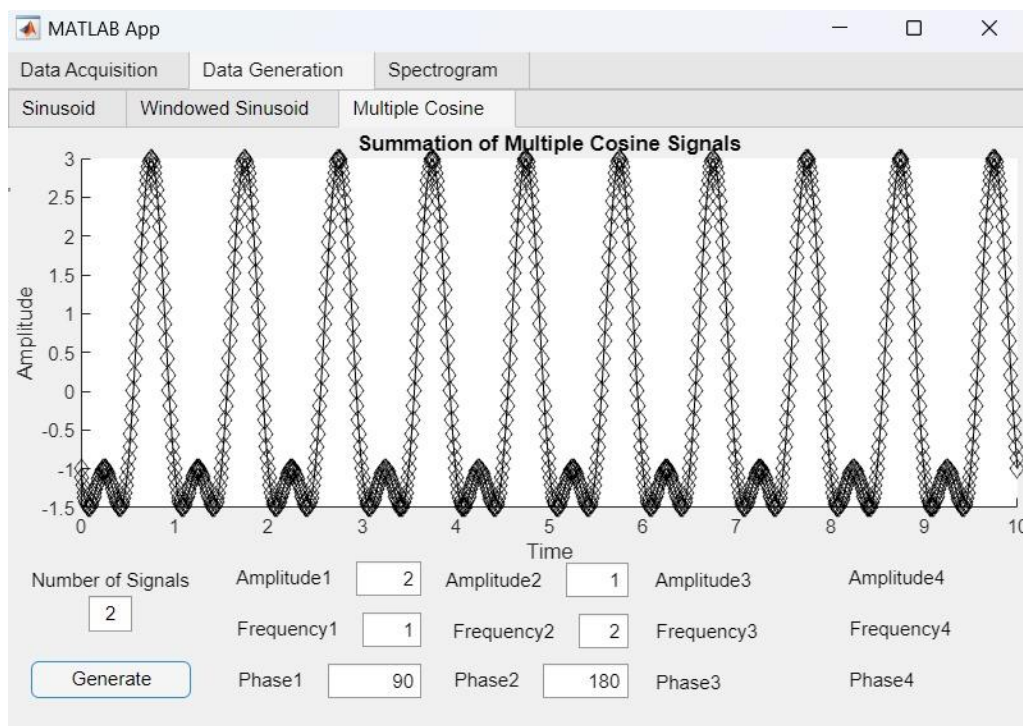


Figure 14: Generated Multiple Cosine Signal Plot

MATLAB CODE EXPLANATION

In this part, we created different cosine signals with given input parameters. User selects the number of components, then signals are created according to that case. Finally, they are summed up and plotted on axes on GUI. In Figure 15, one can see the code for the process when component number is two.

```
411         case 2
412             phase_rad1 = app.Phase1 * pi /180;
413             angle1=2*pi*app.Frequency1*t ;
414             sig1 = app.Amplitude1 * cos(angle1+phase_rad1);
415
416             phase_rad2 = app.Phase2 * pi /180;
417             angle2=2*pi*app.Frequency2*t ;
418             sig2 = app.Amplitude2 * cos(angle2+phase_rad2);
419
420             x = sig1 + sig2;
421             plot(app.UIAxes2_3,t, x, '-kd');
```

Figure 15: MATLAB Code for Multiple Cosine Signal Generation with 2 Components

APPENDIX

```
classdef app1 < matlab.apps.AppBase
```

```
    % Properties that correspond to app components
```

```
    properties (Access = public)
```

UIFigure	matlab.ui.Figure
TabGroup	matlab.ui.container.TabGroup
DataAcquisitionTab	matlab.ui.container.Tab
TabGroup2	matlab.ui.container.TabGroup
RecordTab	matlab.ui.container.Tab
PlayRecordingButton	matlab.ui.control.Button
AudioLengthsEditField	matlab.ui.control.NumericEditField
AudioLengthsEditFieldLabel	matlab.ui.control.Label
StartRecordingButton	matlab.ui.control.Button
FsSlider	matlab.ui.control.Slider
FsSliderLabel	matlab.ui.control.Label
UIAxes	matlab.ui.control.UIAxes
ImportTab	matlab.ui.container.Tab
ImportButton	matlab.ui.control.Button
PleaseEntertheFileNameEditField	matlab.ui.control.EditField
PleaseEntertheFileNameLabel	matlab.ui.control.Label
FSTextArea	matlab.ui.control.TextArea
FSLabel	matlab.ui.control.Label
UIAxes_2	matlab.ui.control.UIAxes
DataGenerationTab	matlab.ui.container.Tab
TabGroup3	matlab.ui.container.TabGroup
SinusoidTab	matlab.ui.container.Tab
PleaseEntertheParametersLabel	matlab.ui.control.Label
GenerateButton	matlab.ui.control.Button
PhaseEditField	matlab.ui.control.NumericEditField
PhaseEditFieldLabel	matlab.ui.control.Label
FrequencyEditField	matlab.ui.control.NumericEditField
FrequencyEditFieldLabel	matlab.ui.control.Label
AmplitudeEditField	matlab.ui.control.NumericEditField
AmplitudeEditFieldLabel	matlab.ui.control.Label
UIAxes2	matlab.ui.control.UIAxes
WindowedSinusoidTab	matlab.ui.container.Tab
HammingButton	matlab.ui.control.Button
HannButton	matlab.ui.control.Button
RectangularButton	matlab.ui.control.Button
TriangularButton	matlab.ui.control.Button
WindowLengthEditField	matlab.ui.control.NumericEditField
WindowLengthEditFieldLabel	matlab.ui.control.Label
StartingTimeEditField	matlab.ui.control.NumericEditField
StartingTimeEditFieldLabel	matlab.ui.control.Label
PhaseEditField_2	matlab.ui.control.NumericEditField
PhaseEditField_2Label	matlab.ui.control.Label
FrequencyEditField_2	matlab.ui.control.NumericEditField
FrequencyEditField_2Label	matlab.ui.control.Label
AmplitudeEditField_2	matlab.ui.control.NumericEditField
AmplitudeEditField_2Label	matlab.ui.control.Label
UIAxes2_2	matlab.ui.control.UIAxes
MultipleCosineTab	matlab.ui.container.Tab
NumberOfSignalsLabel	matlab.ui.control.Label
GenerateButton_2	matlab.ui.control.Button
CompEditField	matlab.ui.control.NumericEditField
Phase4EditField	matlab.ui.control.NumericEditField
Phase4EditFieldLabel	matlab.ui.control.Label

Frequency4EditField	matlab.ui.control.NumericEditField
Frequency4EditFieldLabel	matlab.ui.control.Label
Amplitude4EditField	matlab.ui.control.NumericEditField
Amplitude4EditFieldLabel	matlab.ui.control.Label
Phase3EditField	matlab.ui.control.NumericEditField
Phase3EditFieldLabel	matlab.ui.control.Label
Frequency3EditField	matlab.ui.control.NumericEditField
Frequency3EditFieldLabel	matlab.ui.control.Label
Amplitude3EditField	matlab.ui.control.NumericEditField
Amplitude3EditFieldLabel	matlab.ui.control.Label
Phase2EditField	matlab.ui.control.NumericEditField
Phase2EditFieldLabel	matlab.ui.control.Label
Frequency2EditField	matlab.ui.control.NumericEditField
Frequency2EditFieldLabel	matlab.ui.control.Label
Amplitude2EditField	matlab.ui.control.NumericEditField
Amplitude2EditFieldLabel	matlab.ui.control.Label
Phase1EditField	matlab.ui.control.NumericEditField
Phase1EditFieldLabel	matlab.ui.control.Label
Frequency1EditField	matlab.ui.control.NumericEditField
Frequency1EditFieldLabel	matlab.ui.control.Label
Amplitude1EditField	matlab.ui.control.NumericEditField
Amplitude1EditFieldLabel	matlab.ui.control.Label
Tree	matlab.ui.container.Tree
Node	matlab.ui.container.TreeNode
Node2	matlab.ui.container.TreeNode
Node3	matlab.ui.container.TreeNode
Node4	matlab.ui.container.TreeNode
UIAxes2_3	matlab.ui.control.UIAxes
SpectrogramTab	matlab.ui.container.Tab

end

properties (Access = private)

Length % Audio length
 Recorder % Audio
 Fs % Sampling frequency
 Amplitude
 Phase
 Frequency
 WindowLength
 StartingTime
 Phase1
 Frequency1
 Amplitude1
 Phase2
 Frequency2
 Amplitude2
 Phase3
 Frequency3
 Amplitude3
 Phase4
 Frequency4
 Amplitude4
 Number_of_sig
 FileName % Description

end

% Callbacks that handle component events

```

methods (Access = private)

% Value changing function: FsSlider
function FsSliderValueChanging(app, event)
    app.Fs = event.Value;
end

% Button pushed function: StartRecordingButton
function StartRecordingButtonPushed(app, event)
    app.Recorder = audiorecorder(app.Fs,24,1);
    recordblocking(app.Recorder,app.Length);
    x=getaudiodata(app.Recorder,'uint8');
    plot(app.UIAxes,x);
end

% Value changed function: AudioLengthsEditField
function AudioLengthsEditFieldValueChanged(app, event)
    app.Length = app.AudioLengthsEditField.Value;
end

% Button pushed function: PlayRecordingButton
function PlayRecordingButtonPushed(app, event)
    play(app.Recorder)
end

% Callback function: not associated with a component
function PaintItBlackButtonPushed(app, event)
    [y,Fs1] = audioread('paint_it_black.wav');
    %Fs1 = int32(Fs1);
    %app.Fs = int32(app.Fs);
    y = resample(y, 1, Fs1./app.Fs);
    sound(y, app.Fs);
    plot(app.UIAxes_2, y);
end

% Callback function: not associated with a component
function SweetChildButtonPushed(app, event)
    [y,Fs1] = audioread('sweet_child.wav');
    app.Fs = double(app.Fs);
    y = resample(y, app.Fs);
    sound(y, app.Fs);
    plot(app.UIAxes_2, y);
end

% Callback function: not associated with a component
function MarioButtonPushed(app, event)
    [y,Fs1] = audioread('mario.wav');
    app.Fs = double(app.Fs);
    y = resample(y, app.Fs, Fs1);
    sound(y, app.Fs);
    plot(app.UIAxes_2, y);
end

% Callback function: not associated with a component
function GodFatherButtonPushed(app, event)
    [y,app.Fs] = audioread('audio-samples/god_father.wav');
    sound(y, app.Fs)
    app.FSTextArea.Value = sprintf('%2.f',app.Fs);
    plot(app.UIAxes_2, y);
end

```

```

end

% Callback function: not associated with a component
function ZombieButtonPushed(app, event)
    [y,Fs1] = audioread('zombie.wav');
    app.Fs = double(app.Fs);
    y = resample(y, app.Fs);
    sound(y, app.Fs);
    plot(app.UIAxes_2, y);
end

% Value changed function: PhaseEditField
function PhaseEditFieldValueChanged(app, event)
    app.Phase = app.PhaseEditField.Value;
end

% Value changed function: FrequencyEditField
function FrequencyEditFieldValueChanged(app, event)
    app.Frequency = app.FrequencyEditField.Value;
end

% Value changed function: AmplitudeEditField
function AmplitudeEditFieldValueChanged(app, event)
    app.Amplitude = app.AmplitudeEditField.Value;
end

% Button pushed function: GenerateButton
function GenerateButtonPushed(app, event)
    t = 0:(1/100):10;
    phase_rad = app.Phase * pi /180;
    angle=2*pi*app.Frequency*t ;
    x = app.Amplitude * cos(angle+phase_rad);
    plot(app.UIAxes2,t, x, '-kd');
    grid on
end

% Value changed function: PhaseEditField_2
function PhaseEditField_2ValueChanged(app, event)
    app.Phase = app.PhaseEditField_2.Value;
end

% Button pushed function: RectangularButton
function RectangularButtonPushed(app, event)
    t = 0:(1/100):10;
    phase_rad = app.Phase * pi /180;

    shifted_sine = app.Amplitude * cos(2*pi*app.Frequency*(t-
app.StartingTime)+phase_rad);
    win_length_samples = round(app.WindowLength * 100);
    rectangular_window = rectwin(win_length_samples)';
    windowed_signal = shifted_sine(1:win_length_samples) .*
rectangular_window;

    plot(app.UIAxes2_2,t(1:win_length_samples), windowed_signal);
end

% Value changed function: FrequencyEditField_2

```

```

function FrequencyEditField_2ValueChanged(app, event)
    app.Frequency = app.FrequencyEditField_2.Value;
end

% Value changed function: AmplitudeEditField_2
function AmplitudeEditField_2ValueChanged(app, event)
    app.Amplitude = app.AmplitudeEditField_2.Value;

end

% Value changed function: StartingTimeEditField
function StartingTimeEditFieldValueChanged(app, event)
    app.StartingTime = app.StartingTimeEditField.Value;

end

% Value changed function: WindowLengthEditField
function WindowLengthEditFieldValueChanged(app, event)
    app.WindowLength = app.WindowLengthEditField.Value;

end

% Value changing function: FSTextArea
function FSTextAreaValueChanging(app, event)

end

% Value changed function: FSTextArea
function FSTextAreaValueChanged(app, event)
    app.FSTextArea.Value = app.Fs;

end

% Button pushed function: HammingButton
function HammingButtonPushed(app, event)
    t = 0:(1/100):10;
    phase_rad = app.Phase * pi /180;

    shifted_sine = app.Amplitude * cos(2*pi*app.Frequency*(t-
app.StartingTime)+phase_rad);
    win_length_samples = round(app.WindowLength * 100);
    hamming_window = hamming(win_length_samples)';
    windowed_signal = shifted_sine(1:win_length_samples) .*
hamming_window;

    plot(app.UIAxes2_2,t(1:win_length_samples), windowed_signal);

end

% Button pushed function: HannButton
function HannButtonPushed(app, event)
    t = 0:(1/100):10;
    phase_rad = app.Phase * pi /180;

    shifted_sine = app.Amplitude * cos(2*pi*app.Frequency*(t-
app.StartingTime)+phase_rad);
    win_length_samples = round(app.WindowLength * 100);
    hann_window = hann(win_length_samples)';
    windowed_signal = shifted_sine(1:win_length_samples) .* hann_window;

```

```

        plot(app.UIAxes2_2,t(1:win_length_samples), windowed_signal);

    end

    % Button pushed function: TriangularButton
    function TriangularButtonPushed(app, event)
        t = 0:(1/100):10;
        phase_rad = app.Phase * pi /180;

        shifted_sine = app.Amplitude * cos(2*pi*app.Frequency*(t-
app.StartingTime)+phase_rad);
        win_length_samples = round(app.WindowLength * 100);
        triangular_window = triang(win_length_samples)';
        windowed_signal = shifted_sine(1:win_length_samples) .*
triangular_window;

        plot(app.UIAxes2_2,t(1:win_length_samples), windowed_signal);

    end

    % Value changed function: CompEditField
    function CompEditFieldValueChanged(app, event)
        app.Number_of_sig = app.CompEditField.Value;

        if(app.Number_of_sig==1)
            app.Amplitude1EditField.Visible = 'on';
            app.Frequency1EditField.Visible = 'on';
            app.Phase1EditField.Visible = "on";
            app.Amplitude2EditField.Visible = 'on';
            app.Frequency2EditField.Visible = 'on';
            app.Phase2EditField.Visible = "on";
            app.Amplitude2EditField.Visible = 'off';
            app.Frequency2EditField.Visible = 'off';
            app.Phase2EditField.Visible = "off";
            app.Amplitude3EditField.Visible = 'off';
            app.Frequency3EditField.Visible = 'off';
            app.Phase3EditField.Visible = "off";
            app.Amplitude4EditField.Visible = 'off';
            app.Frequency4EditField.Visible = 'off';
            app.Phase4EditField.Visible = "off";

        elseif(app.Number_of_sig==2)
            app.Amplitude1EditField.Visible = 'on';
            app.Frequency1EditField.Visible = 'on';
            app.Phase1EditField.Visible = "on";
            app.Amplitude2EditField.Visible = 'on';
            app.Frequency2EditField.Visible = 'on';
            app.Phase2EditField.Visible = "on";
            app.Amplitude2EditField.Visible = 'on';
            app.Frequency2EditField.Visible = 'on';
            app.Phase2EditField.Visible = "on";
            app.Amplitude3EditField.Visible = 'off';
            app.Frequency3EditField.Visible = 'off';
            app.Phase3EditField.Visible = "off";
            app.Amplitude4EditField.Visible = 'off';
            app.Frequency4EditField.Visible = 'off';
            app.Phase4EditField.Visible = "off";
        end
    end
end

```

```

elseif(app.Number_of_sig==3)
    app.Amplitude1EditField.Visible = 'on';
    app.Frequency1EditField.Visible = 'on';
    app.Phase1EditField.Visible = "on";
    app.Amplitude2EditField.Visible = 'on';
    app.Frequency2EditField.Visible = 'on';
    app.Phase2EditField.Visible = "on";
    app.Amplitude2EditField.Visible = 'on';
    app.Frequency2EditField.Visible = 'on';
    app.Phase2EditField.Visible = "on";
    app.Amplitude3EditField.Visible = 'on';
    app.Frequency3EditField.Visible = 'on';
    app.Phase3EditField.Visible = "on";
    app.Amplitude4EditField.Visible = 'off';
    app.Frequency4EditField.Visible = 'off';
    app.Phase4EditField.Visible = "off";

```

```

elseif(app.Number_of_sig==4)
    app.Amplitude1EditField.Visible = 'on';
    app.Frequency1EditField.Visible = 'on';
    app.Phase1EditField.Visible = "on";
    app.Amplitude2EditField.Visible = 'on';
    app.Frequency2EditField.Visible = 'on';
    app.Phase2EditField.Visible = "on";
    app.Amplitude2EditField.Visible = 'on';
    app.Frequency2EditField.Visible = 'on';
    app.Phase2EditField.Visible = "on";
    app.Amplitude3EditField.Visible = 'on';
    app.Frequency3EditField.Visible = 'on';
    app.Phase3EditField.Visible = "on";
    app.Amplitude4EditField.Visible = 'on';
    app.Frequency4EditField.Visible = 'on';
    app.Phase4EditField.Visible = "on";

```

```

else
    app.Amplitude1EditField.Visible = 'off';
    app.Frequency1EditField.Visible = 'off';
    app.Phase1EditField.Visible = "off";
    app.Amplitude2EditField.Visible = 'off';
    app.Frequency2EditField.Visible = 'off';
    app.Phase2EditField.Visible = "off";
    app.Amplitude2EditField.Visible = 'off';
    app.Frequency2EditField.Visible = 'off';
    app.Phase2EditField.Visible = "off";
    app.Amplitude3EditField.Visible = 'off';
    app.Frequency3EditField.Visible = 'off';
    app.Phase3EditField.Visible = "off";
    app.Amplitude4EditField.Visible = 'off';
    app.Frequency4EditField.Visible = 'off';
    app.Phase4EditField.Visible = "off";

```

```

end

```

```

end

```

```

% Button pushed function: GenerateButton_2

```

```

function GenerateButton_2Pushed(app, event)

```

```

    t = 0:(1/100):10;

```

```

    switch app.Number_of_sig

```



```

case 1
    phase_rad1 = app.Phase1 * pi /180;
    angle1=2*pi*app.Frequency1*t ;
    x = app.Amplitude1 * cos(angle1+phase_rad1);
    plot(app.UIAxes2_3,t, x, '-kd');

case 2
    phase_rad1 = app.Phase1 * pi /180;
    angle1=2*pi*app.Frequency1*t ;
    sig1 = app.Amplitude1 * cos(angle1+phase_rad1);

    phase_rad2 = app.Phase2 * pi /180;
    angle2=2*pi*app.Frequency2*t ;
    sig2 = app.Amplitude2 * cos(angle2+phase_rad2);

    x = sig1 + sig2;
    plot(app.UIAxes2_3,t, x, '-kd');

case 3
    phase_rad1 = app.Phase1 * pi /180;
    angle1=2*pi*app.Frequency1*t ;
    sig1 = app.Amplitude1 * cos(angle1+phase_rad1);

    phase_rad2 = app.Phase2 * pi /180;
    angle2=2*pi*app.Frequency2*t ;
    sig2 = app.Amplitude2 * cos(angle2+phase_rad2);

    phase_rad3 = app.Phase3 * pi /180;
    angle3=2*pi*app.Frequency3*t ;
    sig3 = app.Amplitude3 * cos(angle3+phase_rad3);

    x = sig1 + sig2 + sig3;
    plot(app.UIAxes2_3,t, x, '-kd');

case 4
    phase_rad1 = app.Phase1 * pi /180;
    angle1=2*pi*app.Frequency1*t ;
    sig1 = app.Amplitude1 * cos(angle1+phase_rad1);

    phase_rad2 = app.Phase2 * pi /180;
    angle2=2*pi*app.Frequency2*t ;
    sig2 = app.Amplitude2 * cos(angle2+phase_rad2);

    phase_rad3 = app.Phase3 * pi /180;
    angle3=2*pi*app.Frequency3*t ;
    sig3 = app.Amplitude3 * cos(angle3+phase_rad3);

    phase_rad4 = app.Phase4 * pi /180;
    angle4=2*pi*app.Frequency4*t ;
    sig4 = app.Amplitude4 * cos(angle4+phase_rad4);

    x = sig1 + sig2 + sig3 + sig4;
    plot(app.UIAxes2_3,t, x, '-kd');

```

end

end

```

% Value changed function: Phase4EditField
function Phase4EditFieldValueChanged(app, event)
    app.Phase4 = app.Phase4EditField.Value;

end

% Value changed function: Frequency4EditField
function Frequency4EditFieldValueChanged(app, event)
    app.Frequency4 = app.Frequency4EditField.Value;

end

% Value changed function: Amplitude4EditField
function Amplitude4EditFieldValueChanged(app, event)
    app.Amplitude4 = app.Amplitude4EditField.Value;

end

% Value changed function: Amplitude3EditField
function Amplitude3EditFieldValueChanged(app, event)
    app.Amplitude3 = app.Amplitude3EditField.Value;

end

% Value changed function: Amplitude2EditField
function Amplitude2EditFieldValueChanged(app, event)
    app.Amplitude2 = app.Amplitude2EditField.Value;

end

% Value changed function: Amplitude1EditField
function Amplitude1EditFieldValueChanged(app, event)
    app.Amplitude1 = app.Amplitude1EditField.Value;

end

% Value changed function: Frequency3EditField
function Frequency3EditFieldValueChanged(app, event)
    app.Frequency3 = app.Frequency3EditField.Value;

end

% Value changed function: Frequency2EditField
function Frequency2EditFieldValueChanged(app, event)
    app.Frequency2 = app.Frequency2EditField.Value;

end

% Value changed function: Frequency1EditField
function Frequency1EditFieldValueChanged(app, event)
    app.Frequency1 = app.Frequency1EditField.Value;

end

% Value changed function: Phase1EditField
function Phase1EditFieldValueChanged(app, event)
    app.Phase1 = app.Phase1EditField.Value;

```

```

end

% Value changed function: Phase2EditField
function Phase2EditFieldValueChanged(app, event)
    app.Phase2 = app.Phase2EditField.Value;

end

% Value changed function: Phase3EditField
function Phase3EditFieldValueChanged(app, event)
    app.Phase3 = app.Phase3EditField.Value;

end

% Button pushed function: ImportButton
function ImportButtonPushed(app, event)
    [y,app.Fs] = audioread(app.FileName + ".wav");
    sound(y, app.Fs)
    app.FSTextArea.Value = sprintf('%2.f',app.Fs);
    plot(app.UIAxes_2, y);
end

% Value changed function: PleaseEntertheFileNameEditField
function PleaseEntertheFileNameEditFieldValueChanged(app, event)
    app.FileName = app.PleaseEntertheFileNameEditField.Value;

end
end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

    % Create UIFigure and hide until all components are created
    app UIFigure = uifigure('Visible', 'off');
    app UIFigure.Position = [100 100 640 429];
    app UIFigure.Name = 'MATLAB App';

    % Create TabGroup
    app.TabGroup = uitabgroup(app UIFigure);
    app.TabGroup.Position = [1 1 640 429];

    % Create DataAcquisitionTab
    app.DataAcquisitionTab = uitab(app.TabGroup);
    app.DataAcquisitionTab.Title = 'Data Acquisition';

    % Create TabGroup2
    app.TabGroup2 = uitabgroup(app.DataAcquisitionTab);
    app.TabGroup2.Position = [1 1 638 404];

    % Create RecordTab
    app.RecordTab = uitab(app.TabGroup2);
    app.RecordTab.Title = 'Record';

    % Create UIAxes
    app.UIAxes = uiaxes(app.RecordTab);
    title(app.UIAxes, 'Recorded Audio Amplitude vs Time Graph')

```

```

xlabel(app.UIAxes, 'Time')
ylabel(app.UIAxes, 'Amplitude')
zlabel(app.UIAxes, 'Z')
app.UIAxes.Position = [1 151 636 229];

% Create FsSliderLabel
app.FsSliderLabel = uilabel(app.RecordTab);
app.FsSliderLabel.HorizontalAlignment = 'right';
app.FsSliderLabel.Position = [12 130 25 22];
app.FsSliderLabel.Text = 'Fs';

% Create FsSlider
app.FsSlider = uislider(app.RecordTab);
app.FsSlider.Limits = [80 10000];
app.FsSlider.ValueChangingFcn = createCallbackFcn(app,
@FsSliderValueChanging, true);
app.FsSlider.Position = [58 139 568 3];
app.FsSlider.Value = 80;

% Create StartRecordingButton
app.StartRecordingButton = uibutton(app.RecordTab, 'push');
app.StartRecordingButton.ButtonPushedFcn = createCallbackFcn(app,
@StartRecordingButtonPushed, true);
app.StartRecordingButton.Position = [36 55 100 23];
app.StartRecordingButton.Text = 'Start Recording';

% Create AudioLengthsEditFieldLabel
app.AudioLengthsEditFieldLabel = uilabel(app.RecordTab);
app.AudioLengthsEditFieldLabel.HorizontalAlignment = 'right';
app.AudioLengthsEditFieldLabel.Position = [206 55 93 22];
app.AudioLengthsEditFieldLabel.Text = 'Audio Length (s)';

% Create AudioLengthsEditField
app.AudioLengthsEditField = uieditfield(app.RecordTab, 'numeric');
app.AudioLengthsEditField.Limits = [1 Inf];
app.AudioLengthsEditField.ValueChangedFcn = createCallbackFcn(app,
@AudioLengthsEditFieldValueChanged, true);
app.AudioLengthsEditField.Position = [314 55 100 22];
app.AudioLengthsEditField.Value = 1;

% Create PlayRecordingButton
app.PlayRecordingButton = uibutton(app.RecordTab, 'push');
app.PlayRecordingButton.ButtonPushedFcn = createCallbackFcn(app,
@PlayRecordingButtonPushed, true);
app.PlayRecordingButton.Position = [502 54 100 23];
app.PlayRecordingButton.Text = 'Play Recording';

% Create ImportTab
app.ImportTab = uitab(app.TabGroup2);
app.ImportTab.Title = 'Import';

% Create UIAxes_2
app.UIAxes_2 = uiaxes(app.ImportTab);
title(app.UIAxes_2, 'Imported Audio Amplitude vs Time Graph')
xlabel(app.UIAxes_2, 'Time')
ylabel(app.UIAxes_2, 'Amplitude')
zlabel(app.UIAxes_2, 'Z')
app.UIAxes_2.Position = [1 151 636 229];

```

```

% Create FSLabel
app.FSLabel = uilabel(app.ImportTab);
app.FSLabel.HorizontalAlignment = 'right';
app.FSLabel.Position = [235 116 27 22];
app.FSLabel.Text = 'FS: ';

% Create FSTextArea
app.FSTextArea = uitextarea(app.ImportTab);
app.FSTextArea.ValueChangedFcn = createCallbackFcn(app,
@FSTextAreaValueChanged, true);
app.FSTextArea.ValueChangingFcn = createCallbackFcn(app,
@FSTextAreaValueChanging, true);
app.FSTextArea.Position = [277 116 150 24];

% Create PleaseEntertheFileNameLabel
app.PleaseEntertheFileNameLabel = uilabel(app.ImportTab);
app.PleaseEntertheFileNameLabel.HorizontalAlignment = 'right';
app.PleaseEntertheFileNameLabel.Position = [151 45 155 22];
app.PleaseEntertheFileNameLabel.Text = 'Please Enter the File Name: ';

% Create PleaseEntertheFileNameEditField
app.PleaseEntertheFileNameEditField = uieditfield(app.ImportTab,
'text');
app.PleaseEntertheFileNameEditField.ValueChangedFcn =
createCallbackFcn(app, @PleaseEntertheFileNameEditFieldValueChanged, true);
app.PleaseEntertheFileNameEditField.Position = [321 45 100 22];

% Create ImportButton
app.ImportButton = uibutton(app.ImportTab, 'push');
app.ImportButton.ButtonPushedFcn = createCallbackFcn(app,
@ImportButtonPushed, true);
app.ImportButton.Position = [512 45 100 23];
app.ImportButton.Text = 'Import';

% Create DataGenerationTab
app.DataGenerationTab = uitab(app.TabGroup);
app.DataGenerationTab.Title = 'Data Generation';

% Create TabGroup3
app.TabGroup3 = uitabgroup(app.DataGenerationTab);
app.TabGroup3.Position = [1 0 639 405];

% Create SinusoidTab
app.SinusoidTab = uitab(app.TabGroup3);
app.SinusoidTab.Title = 'Sinusoid';

% Create UIAxes2
app.UIAxes2 = uiaxes(app.SinusoidTab);
title(app.UIAxes2, 'Cosine Graph')
xlabel(app.UIAxes2, 'Time')
ylabel(app.UIAxes2, 'Amplitude')
zlabel(app.UIAxes2, 'Z')
app.UIAxes2.Position = [1 108 636 273];

% Create AmplitudeEditFieldLabel
app.AmplitudeEditFieldLabel = uilabel(app.SinusoidTab);
app.AmplitudeEditFieldLabel.HorizontalAlignment = 'right';
app.AmplitudeEditFieldLabel.Position = [22 35 58 22];
app.AmplitudeEditFieldLabel.Text = 'Amplitude';

```

```

% Create AmplitudeEditField
app.AmplitudeEditField = uicontrol(app.SinusoidTab, 'numeric');
app.AmplitudeEditField.ValueChangedFcn = createCallbackFcn(app,
@AmplitudeEditFieldValueChanged, true);
app.AmplitudeEditField.Position = [95 35 100 22];

% Create FrequencyEditFieldLabel
app.FrequencyEditFieldLabel = uicontrol(app.SinusoidTab);
app.FrequencyEditFieldLabel.HorizontalAlignment = 'right';
app.FrequencyEditFieldLabel.Position = [230 35 62 22];
app.FrequencyEditFieldLabel.Text = 'Frequency';

% Create FrequencyEditField
app.FrequencyEditField = uicontrol(app.SinusoidTab, 'numeric');
app.FrequencyEditField.ValueChangedFcn = createCallbackFcn(app,
@FrequencyEditFieldValueChanged, true);
app.FrequencyEditField.Position = [307 35 100 22];

% Create PhaseEditFieldLabel
app.PhaseEditFieldLabel = uicontrol(app.SinusoidTab);
app.PhaseEditFieldLabel.HorizontalAlignment = 'right';
app.PhaseEditFieldLabel.Position = [472 35 39 22];
app.PhaseEditFieldLabel.Text = 'Phase';

% Create PhaseEditField
app.PhaseEditField = uicontrol(app.SinusoidTab, 'numeric');
app.PhaseEditField.ValueChangedFcn = createCallbackFcn(app,
@PhaseEditFieldValueChanged, true);
app.PhaseEditField.Position = [526 35 100 22];
app.PhaseEditField.Value = 9999999999;

% Create GenerateButton
app.GenerateButton = uicontrol(app.SinusoidTab, 'push');
app.GenerateButton.ButtonPushedFcn = createCallbackFcn(app,
@GenerateButtonPushed, true);
app.GenerateButton.Position = [526 79 100 23];
app.GenerateButton.Text = 'Generate';

% Create PleaseEntertheParametersLabel
app.PleaseEntertheParametersLabel = uicontrol(app.SinusoidTab);
app.PleaseEntertheParametersLabel.Position = [23 79 162 22];
app.PleaseEntertheParametersLabel.Text = 'Please Enter the
Parameters: ';

% Create WindowedSinusoidTab
app.WindowedSinusoidTab = uitab(app.TabGroup3);
app.WindowedSinusoidTab.Title = 'Windowed Sinusoid';

% Create UIAxes2_2
app.UIAxes2_2 = uiaxes(app.WindowedSinusoidTab);
title(app.UIAxes2_2, 'Windowed Cosine Graph')
xlabel(app.UIAxes2_2, 'Time')
ylabel(app.UIAxes2_2, 'Amplitude')
zlabel(app.UIAxes2_2, 'Z')
app.UIAxes2_2.Position = [1 108 636 273];

% Create AmplitudeEditField_2Label
app.AmplitudeEditField_2Label = uicontrol(app.WindowedSinusoidTab);

```

```

app.AmplitudeEditField_2Label.HorizontalAlignment = 'right';
app.AmplitudeEditField_2Label.Position = [37 79 58 22];
app.AmplitudeEditField_2Label.Text = 'Amplitude';

% Create AmplitudeEditField_2
app.AmplitudeEditField_2 = uicontrol(app.WindowedSinusoidTab,
'numeric');
app.AmplitudeEditField_2.ValueChangedFcn = createCallbackFcn(app,
@AmplitudeEditField_2ValueChanged, true);
app.AmplitudeEditField_2.Position = [110 79 100 22];

% Create FrequencyEditField_2Label
app.FrequencyEditField_2Label = uicontrol(app.WindowedSinusoidTab);
app.FrequencyEditField_2Label.HorizontalAlignment = 'right';
app.FrequencyEditField_2Label.Position = [450 46 62 22];
app.FrequencyEditField_2Label.Text = 'Frequency';

% Create FrequencyEditField_2
app.FrequencyEditField_2 = uicontrol(app.WindowedSinusoidTab,
'numeric');
app.FrequencyEditField_2.ValueChangedFcn = createCallbackFcn(app,
@FrequencyEditField_2ValueChanged, true);
app.FrequencyEditField_2.Position = [527 46 100 22];

% Create PhaseEditField_2Label
app.PhaseEditField_2Label = uicontrol(app.WindowedSinusoidTab);
app.PhaseEditField_2Label.HorizontalAlignment = 'right';
app.PhaseEditField_2Label.Position = [259 79 39 22];
app.PhaseEditField_2Label.Text = 'Phase';

% Create PhaseEditField_2
app.PhaseEditField_2 = uicontrol(app.WindowedSinusoidTab,
'numeric');
app.PhaseEditField_2.ValueChangedFcn = createCallbackFcn(app,
@PhaseEditField_2ValueChanged, true);
app.PhaseEditField_2.Position = [313 79 100 22];

% Create StartingTimeEditFieldLabel
app.StartingTimeEditFieldLabel = uicontrol(app.WindowedSinusoidTab);
app.StartingTimeEditFieldLabel.HorizontalAlignment = 'right';
app.StartingTimeEditFieldLabel.Position = [436 79 76 22];
app.StartingTimeEditFieldLabel.Text = 'Starting Time';

% Create StartingTimeEditField
app.StartingTimeEditField = uicontrol(app.WindowedSinusoidTab,
'numeric');
app.StartingTimeEditField.ValueChangedFcn = createCallbackFcn(app,
@StartingTimeEditFieldValueChanged, true);
app.StartingTimeEditField.Position = [527 79 100 22];

% Create WindowLengthEditFieldLabel
app.WindowLengthEditFieldLabel = uicontrol(app.WindowedSinusoidTab);
app.WindowLengthEditFieldLabel.HorizontalAlignment = 'right';
app.WindowLengthEditFieldLabel.Position = [212 46 88 22];
app.WindowLengthEditFieldLabel.Text = 'Window Length';

% Create WindowLengthEditField
app.WindowLengthEditField = uicontrol(app.WindowedSinusoidTab,
'numeric');

```

```

        app.WindowLengthEditField.ValueChangedFcn = createCallbackFcn(app,
@WindowLengthEditFieldValueChanged, true);
        app.WindowLengthEditField.Position = [315 46 100 22];

        % Create TriangularButton
        app.TriangularButton = uibutton(app.WindowedSinusoidTab, 'push');
        app.TriangularButton.ButtonPushedFcn = createCallbackFcn(app,
@TriangularButtonPushed, true);
        app.TriangularButton.Position = [527 14 100 23];
        app.TriangularButton.Text = 'Triangular';

        % Create RectangularButton
        app.RectangularButton = uibutton(app.WindowedSinusoidTab, 'push');
        app.RectangularButton.ButtonPushedFcn = createCallbackFcn(app,
@RectangularButtonPushed, true);
        app.RectangularButton.Position = [38 14 100 23];
        app.RectangularButton.Text = 'Rectangular';

        % Create HannButton
        app.HannButton = uibutton(app.WindowedSinusoidTab, 'push');
        app.HannButton.ButtonPushedFcn = createCallbackFcn(app,
@HannButtonPushed, true);
        app.HannButton.Position = [369 14 100 23];
        app.HannButton.Text = 'Hann';

        % Create HammingButton
        app.HammingButton = uibutton(app.WindowedSinusoidTab, 'push');
        app.HammingButton.ButtonPushedFcn = createCallbackFcn(app,
@HammingButtonPushed, true);
        app.HammingButton.Position = [207 14 100 23];
        app.HammingButton.Text = 'Hamming';

        % Create MultipleCosineTab
        app.MultipleCosineTab = uitab(app.TabGroup3);
        app.MultipleCosineTab.Title = 'Multiple Cosine';

        % Create UIAxes2_3
        app.UIAxes2_3 = uiaxes(app.MultipleCosineTab);
        title(app.UIAxes2_3, 'Summation of Multiple Cosine Signals')
        xlabel(app.UIAxes2_3, 'Time')
        ylabel(app.UIAxes2_3, 'Amplitude')
        zlabel(app.UIAxes2_3, 'Z')
        app.UIAxes2_3.Position = [3 108 636 273];

        % Create Tree
        app.Tree = uitree(app.MultipleCosineTab);
        app.Tree.Position = [1 341 2 2];

        % Create Node
        app.Node = uitreenode(app.Tree);
        app.Node.Text = 'Node';

        % Create Node2
        app.Node2 = uitreenode(app.Node);
        app.Node2.Text = 'Node2';

        % Create Node3
        app.Node3 = uitreenode(app.Node);
        app.Node3.Text = 'Node3';

```



```

% Create Node4
app.Node4 = uitreenode(app.Node);
app.Node4.Text = 'Node4';

% Create Amplitude1EditFieldLabel
app.Amplitude1EditFieldLabel = uilabel(app.MultipleCosineTab);
app.Amplitude1EditFieldLabel.HorizontalAlignment = 'right';
app.Amplitude1EditFieldLabel.Position = [139 88 65 22];
app.Amplitude1EditFieldLabel.Text = 'Amplitude1';

% Create Amplitude1EditField
app.Amplitude1EditField = uieditfield(app.MultipleCosineTab,
'numeric');
app.Amplitude1EditField.ValueChangedFcn = createCallbackFcn(app,
@Amplitude1EditFieldValueChanged, true);
app.Amplitude1EditField.Visible = 'off';
app.Amplitude1EditField.Position = [219 88 41 22];

% Create Frequency1EditFieldLabel
app.Frequency1EditFieldLabel = uilabel(app.MultipleCosineTab);
app.Frequency1EditFieldLabel.HorizontalAlignment = 'right';
app.Frequency1EditFieldLabel.Position = [140 56 68 22];
app.Frequency1EditFieldLabel.Text = 'Frequency1';

% Create Frequency1EditField
app.Frequency1EditField = uieditfield(app.MultipleCosineTab,
'numeric');
app.Frequency1EditField.ValueChangedFcn = createCallbackFcn(app,
@Frequency1EditFieldValueChanged, true);
app.Frequency1EditField.Visible = 'off';
app.Frequency1EditField.Position = [223 56 37 22];
app.Frequency1EditField.Value = 9999999999;

% Create Phase1EditFieldLabel
app.Phase1EditFieldLabel = uilabel(app.MultipleCosineTab);
app.Phase1EditFieldLabel.HorizontalAlignment = 'right';
app.Phase1EditFieldLabel.Position = [140 24 46 22];
app.Phase1EditFieldLabel.Text = 'Phase1';

% Create Phase1EditField
app.Phase1EditField = uieditfield(app.MultipleCosineTab, 'numeric');
app.Phase1EditField.ValueChangedFcn = createCallbackFcn(app,
@Phase1EditFieldValueChanged, true);
app.Phase1EditField.Visible = 'off';
app.Phase1EditField.Position = [201 24 59 22];
app.Phase1EditField.Value = 9999999999;

% Create Amplitude2EditFieldLabel
app.Amplitude2EditFieldLabel = uilabel(app.MultipleCosineTab);
app.Amplitude2EditFieldLabel.HorizontalAlignment = 'right';
app.Amplitude2EditFieldLabel.Position = [270 87 65 22];
app.Amplitude2EditFieldLabel.Text = 'Amplitude2';

% Create Amplitude2EditField
app.Amplitude2EditField = uieditfield(app.MultipleCosineTab,
'numeric');
app.Amplitude2EditField.ValueChangedFcn = createCallbackFcn(app,
@Amplitude2EditFieldValueChanged, true);

```

```

app.Amplitude2EditField.Visible = 'off';
app.Amplitude2EditField.Position = [350 87 39 22];

% Create Frequency2EditFieldLabel
app.Frequency2EditFieldLabel = uilabel(app.MultipleCosineTab);
app.Frequency2EditFieldLabel.HorizontalAlignment = 'right';
app.Frequency2EditFieldLabel.Position = [275 55 68 22];
app.Frequency2EditFieldLabel.Text = 'Frequency2';

% Create Frequency2EditField
app.Frequency2EditField = uieditfield(app.MultipleCosineTab,
'numeric');
app.Frequency2EditField.ValueChangedFcn = createCallbackFcn(app,
@Frequency2EditFieldValueChanged, true);
app.Frequency2EditField.Visible = 'off';
app.Frequency2EditField.Position = [358 55 31 22];
app.Frequency2EditField.Value = 9999999999;

% Create Phase2EditFieldLabel
app.Phase2EditFieldLabel = uilabel(app.MultipleCosineTab);
app.Phase2EditFieldLabel.HorizontalAlignment = 'right';
app.Phase2EditFieldLabel.Position = [275 24 46 22];
app.Phase2EditFieldLabel.Text = 'Phase2';

% Create Phase2EditField
app.Phase2EditField = uieditfield(app.MultipleCosineTab, 'numeric');
app.Phase2EditField.ValueChangedFcn = createCallbackFcn(app,
@Phase2EditFieldValueChanged, true);
app.Phase2EditField.Visible = 'off';
app.Phase2EditField.Position = [336 24 53 22];
app.Phase2EditField.Value = 9999999999;

% Create Amplitude3EditFieldLabel
app.Amplitude3EditFieldLabel = uilabel(app.MultipleCosineTab);
app.Amplitude3EditFieldLabel.HorizontalAlignment = 'right';
app.Amplitude3EditFieldLabel.Position = [401 87 65 22];
app.Amplitude3EditFieldLabel.Text = 'Amplitude3';

% Create Amplitude3EditField
app.Amplitude3EditField = uieditfield(app.MultipleCosineTab,
'numeric');
app.Amplitude3EditField.ValueChangedFcn = createCallbackFcn(app,
@Amplitude3EditFieldValueChanged, true);
app.Amplitude3EditField.Visible = 'off';
app.Amplitude3EditField.Position = [481 87 32 22];

% Create Frequency3EditFieldLabel
app.Frequency3EditFieldLabel = uilabel(app.MultipleCosineTab);
app.Frequency3EditFieldLabel.HorizontalAlignment = 'right';
app.Frequency3EditFieldLabel.Position = [402 55 68 22];
app.Frequency3EditFieldLabel.Text = 'Frequency3';

% Create Frequency3EditField
app.Frequency3EditField = uieditfield(app.MultipleCosineTab,
'numeric');
app.Frequency3EditField.ValueChangedFcn = createCallbackFcn(app,
@Frequency3EditFieldValueChanged, true);
app.Frequency3EditField.Visible = 'off';
app.Frequency3EditField.Position = [485 55 28 22];

```

```

app.Frequency3EditField.Value = 9999999999;

% Create Phase3EditFieldLabel
app.Phase3EditFieldLabel = uilabel(app.MultipleCosineTab);
app.Phase3EditFieldLabel.HorizontalAlignment = 'right';
app.Phase3EditFieldLabel.Position = [401 23 46 22];
app.Phase3EditFieldLabel.Text = 'Phase3';

% Create Phase3EditField
app.Phase3EditField = uieditfield(app.MultipleCosineTab, 'numeric');
app.Phase3EditField.ValueChangedFcn = createCallbackFcn(app,
@Phase3EditFieldValueChanged, true);
app.Phase3EditField.Visible = 'off';
app.Phase3EditField.Position = [462 23 51 22];
app.Phase3EditField.Value = 9999999999;

% Create Amplitude4EditFieldLabel
app.Amplitude4EditFieldLabel = uilabel(app.MultipleCosineTab);
app.Amplitude4EditFieldLabel.HorizontalAlignment = 'right';
app.Amplitude4EditFieldLabel.Position = [522 88 65 22];
app.Amplitude4EditFieldLabel.Text = 'Amplitude4';

% Create Amplitude4EditField
app.Amplitude4EditField = uieditfield(app.MultipleCosineTab,
'numeric');
app.Amplitude4EditField.ValueChangedFcn = createCallbackFcn(app,
@Amplitude4EditFieldValueChanged, true);
app.Amplitude4EditField.Visible = 'off';
app.Amplitude4EditField.Position = [602 88 35 22];

% Create Frequency4EditFieldLabel
app.Frequency4EditFieldLabel = uilabel(app.MultipleCosineTab);
app.Frequency4EditFieldLabel.HorizontalAlignment = 'right';
app.Frequency4EditFieldLabel.Position = [523 56 68 22];
app.Frequency4EditFieldLabel.Text = 'Frequency4';

% Create Frequency4EditField
app.Frequency4EditField = uieditfield(app.MultipleCosineTab,
'numeric');
app.Frequency4EditField.ValueChangedFcn = createCallbackFcn(app,
@Frequency4EditFieldValueChanged, true);
app.Frequency4EditField.Visible = 'off';
app.Frequency4EditField.Position = [606 56 31 22];
app.Frequency4EditField.Value = 9999999999;

% Create Phase4EditFieldLabel
app.Phase4EditFieldLabel = uilabel(app.MultipleCosineTab);
app.Phase4EditFieldLabel.HorizontalAlignment = 'right';
app.Phase4EditFieldLabel.Position = [522 24 46 22];
app.Phase4EditFieldLabel.Text = 'Phase4';

% Create Phase4EditField
app.Phase4EditField = uieditfield(app.MultipleCosineTab, 'numeric');
app.Phase4EditField.ValueChangedFcn = createCallbackFcn(app,
@Phase4EditFieldValueChanged, true);
app.Phase4EditField.Visible = 'off';
app.Phase4EditField.Position = [583 24 56 22];
app.Phase4EditField.Value = 9999999999;

```

```

        % Create CompEditField
        app.CompEditField = uieditfield(app.MultipleCosineTab, 'numeric');
        app.CompEditField.Limits = [0 4];
        app.CompEditField.ValueChangedFcn = createCallbackFcn(app,
@CompEditFieldValueChanged, true);
        app.CompEditField.HorizontalAlignment = 'center';
        app.CompEditField.Position = [52 66 27 22];

        % Create GenerateButton_2
        app.GenerateButton_2 = uibutton(app.MultipleCosineTab, 'push');
        app.GenerateButton_2.ButtonPushedFcn = createCallbackFcn(app,
@GenerateButton_2Pushed, true);
        app.GenerateButton_2.Position = [16 24 100 23];
        app.GenerateButton_2.Text = 'Generate';

        % Create NumberofSignalsLabel
        app.NumberofSignalsLabel = uilabel(app.MultipleCosineTab);
        app.NumberofSignalsLabel.Position = [16 87 104 22];
        app.NumberofSignalsLabel.Text = 'Number of Signals';

        % Create SpectrogramTab
        app.SpectrogramTab = uitab(app.TabGroup);
        app.SpectrogramTab.Title = 'Spectrogram';

        % Show the figure after all components are created
        app.UIFigure.Visible = 'on';
    end
end

% App creation and deletion
methods (Access = public)

    % Construct app
    function app = app1

        % Create UIFigure and components
        createComponents(app)

        % Register the app with App Designer
        registerApp(app, app.UIFigure)

        if nargin == 0
            clear app
        end
    end

    % Code that executes before app deletion
    function delete(app)

        % Delete UIFigure when app is deleted
        delete(app.UIFigure)
    end
end
end

```