Project Report

Sound Recognition and Visualization

Omar Mohammad Zaineh 1088546

Mohammad Fares Al jamous 1088672

Supervised By: Dr. Rabah Al Abdi

جامعة أبوظبي
ABU DHABI UNIVERSITY

Submitted: November 25, 2023

Contents

List of Figures

List of Tables

Abstract

In this project we were asked to create a program that can take an audio input from the user's device, visualize, and analyze the input audio signal. This program was created on the platform MATLAB which can be considered a programming language with an extensive library of built in functions. This audio visualizer and analyzer program can have many applications and its features can be extended way beyond what we have made here, as this is just a basic program that has basic features and functionality, that can be used by anyone easily. The program has very little computational cost as it uses fast algorithms to do the tasks it is supposed to do. The program shows the recorded audio signal itself, what frequencies make up the input audio signal and other properties of the audio signal.

1 Introduction

In this project we were required to create a sound visualizer and analyzer in MATLAB, which required us to write MATLAB code to record audio from the computer system's audio input device. There are many methods that we could have used but we used base MATLAB's built in functions to create the program, since MATLAB has a massive library of build in functions. We were also required to create a graphical user interface (GUI) which any user can use at any time with little difficulty, to record and analyze their audio.

1.1 Motivation

Audio analysis is a very important part of our modern world, which can be used everywhere, and allows us to create very powerful tools that have a wide range of uses, such as in music, speech recognition, healthcare, security and surveillance, education, entertainment (movies, video games, etc..), and most importantly telecommunications which is the backbone of our modern world. Without audio analysis that allowed us to develop telecommunication technology most industries would be unable to function, and the world would not have been able to advance as much as it had. So to understand audio analysis, it is important to have a visual image of what an audio signal looks like.

1.2 Problem Statement

Audio analysis and visualizer programs can have many uses such as finding irregularities in the sound signal in physical systems such as in car engines where a human might not be able to pick up that there is an issue just from the sound of the engine, they can also be used to find noise and filter it out of an audio signal, and they can be used to detect many other problems by finding certain frequencies in an audio signal which would not be present if there were no issues.

1.3 Literature Review

There are many advanced applications and programs that can take an audio signal analyze it and then allow a user to edit it, that exist in the market today, these programs were made using different methods, but achieve the same goal, examples of these applications include:

- Sonic visualizer: which is an open source program that can visualize and analyze audio files.
- SoundSpectrum: which can visualize any audio signal you give in real time
- After Effects: Which is an incredibly complex applictaion that can be used for audio and video editing

2 Design

MATLAB was used to create the audio visualizer and analyzer program, using built in MATLAB functions of the base version of MATLAB without using any toolboxes. The functions we used included methods that define an audio object in MATLAB, and record an audio using the computer system's input audio device which is stored in the object, a method to extract an audio vector from the audio object was used, and many other functions that are used in analyzing and visualizing discrete time signals. The graphical user interface was also created using MATLAB's app feature.

2.1 Requirements Constraints, and Considerations

There were many objectives, requirements, and objectives that we were required to fulfill in this project, and they are:

- Objectives:
  1. Use MATLAB software to simulate a sound visualizer.
  2. Find the Fourier transform of various recorded audio signals.
  3. plot the recorded audio signals.
  4. Obtain the statistical features of the recorded audio signals.
- Requirements:
  - Record an audio message.

  - Display the recorded audio signal.
  - Obtain and plot the Fourier transform of the recorded audio signal.
  - Plot the Power Spectrum of the recorded audio signal.
  - Calculate the Energy of the recorded audio signal.
  - Calculate the power spectral density of the recorded audio signal.
  - Find the maximum frequency component of the recorded audio signal.
  - Plot the histogram of the recorded audio signal.
  - Obtain the statistical features of the recorded audio signal, including the mean, standard deviation, variance, etc..

- Constraints:

  1. The system should be implemented using MATLAB.
  2. The system needs to visualize the audio signals recorded by the user.
  3. The system needs to analyze the recorded audio signals

2.2 Design Process

In total there were eight steps that we took to create the audio visualizer and analyzer program. In the first step recorded the audio signal and found the audio vector in MATLAB and its time vector. In the second step we found the discrete time Fourier transform of the time domain audio signal and its frequency vector. In the third step we found the power spectrum vector of the audio signal. In the fourth step we found the plots for each of the previous vectors (time domain audio signal, frequency domain audio signal, and the power spectrum of the audio signal). In the fifth step we found the energy and power spectral density of the audio signal. In the sixth step we found the maximum frequency component of the audio signal. In the seventh step we found the statistical features of the audio signal and the plotted a histogram for the audio signal. Lastly in the eighth step we displayed all the values we got (energy, statistical features, etc..), and then we made a GUI which any user can use.

2.2.1 Recording and Finding the Audio Signal

1. Step one, we defined the object (audio) in the built in MATLAB class audiorecorder, using three parameters which are sampling frequency (44.1 kHz), bits per sample (16 bits), and the number of channels (1 channel), these parameters were used as input to the method audiorecorder(). Then we took the duration of the audio signal.

```
% Defining And Giving Parameters To The Audio Object
% aduiorecorderObj
f = 44100;
audio = audiorecorder(f,16,1);
duration = input('Input The Duration of The Audio: ');
```

2. Step two, we recorded the audio using the method recordblocking() which takes the duration of the desired recording and the audiorecorder object (audio) that we defined as input.

```
% Recording The Audio
disp('start speaking')
recordblocking(audio, duration);
disp('End of recording')
```

3. Step three, we extracted the audio vector from the object (audio) using the method getaudiodata(), and we gave the audio vector the name "signalt". Then we defined the sampling time (t) using the period of the sampling time and the duration of the audio signal. And we defined the number of samples (n).

```
% Finding The Aduio Signal
signalt = getaudiodata(audio);
t = 0:1/f:duration-1/f;
n = length(t);
```

2.2.2 Finding The Discrete Time Fourier Transform of The Audio Signal

1. Step one, we simply used the function fft() (fast fourier transform) which is an algrothim used to compute the fourier transform quickly,

to find the discrete time fourier transform (signalf) of the time domain audio signal (signalt).

```
% Finding The Fourier Transform And The Frequencies That Make
    The Audio Signal Up
signalf = fft(signalt);
```

2. Step two, we found the frequency axes for the frequency domain signal (fd)

```
f_d = -n/2:(n/2)-1;
w = length(f_d);
```

2.2.3 Finding The Power Spectrum of The Audio Signal

1. We used the power formula to find the power of the audio signal from the frequency domain signal.

$$P = \frac{|F(\omega)|^2}{n}$$

```
power = (abs(signalf).^2)/n;
```

2.2.4 Plotting The Time Domain Audio Signal, Frequency Domain Audio Signal, And The Power Spectrum of The Audio Signal

1. Step one, we plotted the time domain signal against time.

```
figure(1)
stem(t, signalt, "MarkerSize",0.1,"LineWidth",0.1)
xlabel('Time (s)')
ylabel('Amplitude')
title('Time Domain Plot')
```

2. Step two, we plotted the frequency domain signal against the frequency.

```
figure(2)
stem(f_d(n/2+1:n), abs(signalf(n/2+1:n)),
    "MarkerSize",0.1,"LineWidth",0.1)
xlabel('Frequency (Hz)')
ylabel('Amplitude')
title('Fourier Transfrom Plot')
```

3. Step three, we plotted the power spectrum against the frequency

```
figure(3)
stem(f_d(n/2+1:n), power(n/2+1:n),
    "MarkerSize",0.1,"LineWidth",0.1)
xlabel('Frequency (Hz)')
ylabel('Power')
title('Power Spectrum Plot')
```

2.2.5 Calculating The Energy of The Audio Signal

1. We used the energy formula to find the energy of the audio signal from the time domain signal.

$$E = \sum |f(t)|^2$$

```
% Calculating The Energy
energy = sum(abs(signalt).^2);
```

2.2.6 Finding The Maximum Frequency Component of The Aduio Signal

1. Step one, we find the index of the the maximum frequency component using the functions max() and find().

```
% Finding The Maximum Frequency Component
max_freq_comp_index = find(abs(signalf) == max(abs(signalf)));
```

2. Step two, we find the positive and negative maximum frequency component (the frequencies are mirrored) of the signal and using the index we found earlier, and then only take the positive frequency component.

```
max_freq_comp = f_d(max_freq_comp_index);
max_freq_comp_1 = max_freq_comp(2)
```

2.2.7 Finding The Statistical Features of The Audio Signal And Plotting a Histogram

1. Step one, we found the common statistical features of the audio signals using base MATLAB functions.

```
% Statistical Features of The Recorded Audio Signal
Average = mean(signalt);
Standard_Deviation = std(signalt);
Variance = Standard_Deviation^2;
Mode = mode(signalt);
Median = median(signalt);
```

2. Step two, plotting the histogram of the time domain audio signal.

```
figure(4)
histogram(signalt)
title('The Signal''s Histogram' )
```

2.2.8 Displaying The Calculated Values And Creating The GUI

1. Step one, we used the fprintf() function to print each of the values we got.

```
fprintf('Duration = %4.2f s \n',duration)
fprintf('Energy = %6.2f \n',energy)
fprintf('Maximum Frequency Component = %6.0f
    \n',max_freq_comp_1)
fprintf('Average = %1.6f \n',Average)
fprintf('Standard Deviation = %1.6f \n',Standard_Deviation)
fprintf('Variance = %1.6f \n',Variance)
fprintf('Mode = %1.6f \n',Mode)
```

```
fprintf('Median = %1.6f \n',Median)
```

2. Step two, we created a GUI by integrating all the previous code we wrote into the app maker in MATLAB, and the final result can be seen in Figure 1.

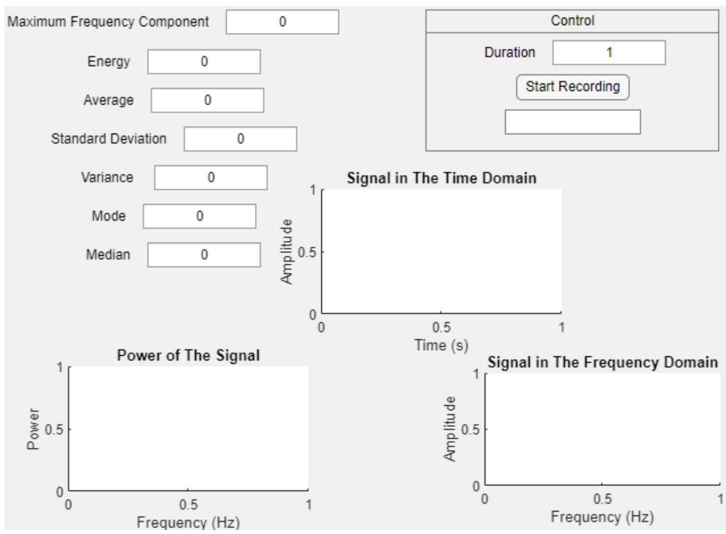

Figure 1: GUI.

2.3 System Overview

When all the code was written we had a simple system that records an audio signal and does a bunch of calculations with the signal, and then it shows the results to the user.

1. The user starts by running the script file Figure 2.
2. The user inputs the audio's duration Figure 3.
3. The user records the audio Figure 4.
4. The Program Outputs The Graphs Figure 5.
5. The Program Outputs The Values Figure 6.
6. The user can use the GUI similarly Figure 7.

2.4 Component Design

The system was completely made using built in MATLAB functions.

3 Experimental Testing and Results

3.1 Testing Plan and Acceptance Criteria

Testing the program is quite simple, since the user only has to give two inputs (duration, audio), and the program just does calculations and plotting based on the audio inputted. For the program to pass the testing it just needs to take the audio input and produce outputs based on the input signal.

3.2 Results

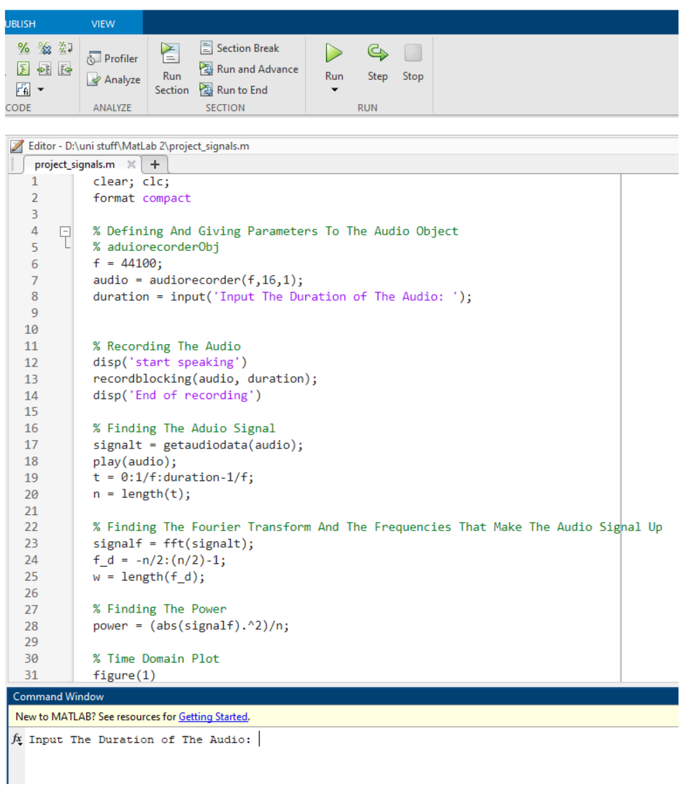

Figure 2: Running the script file.

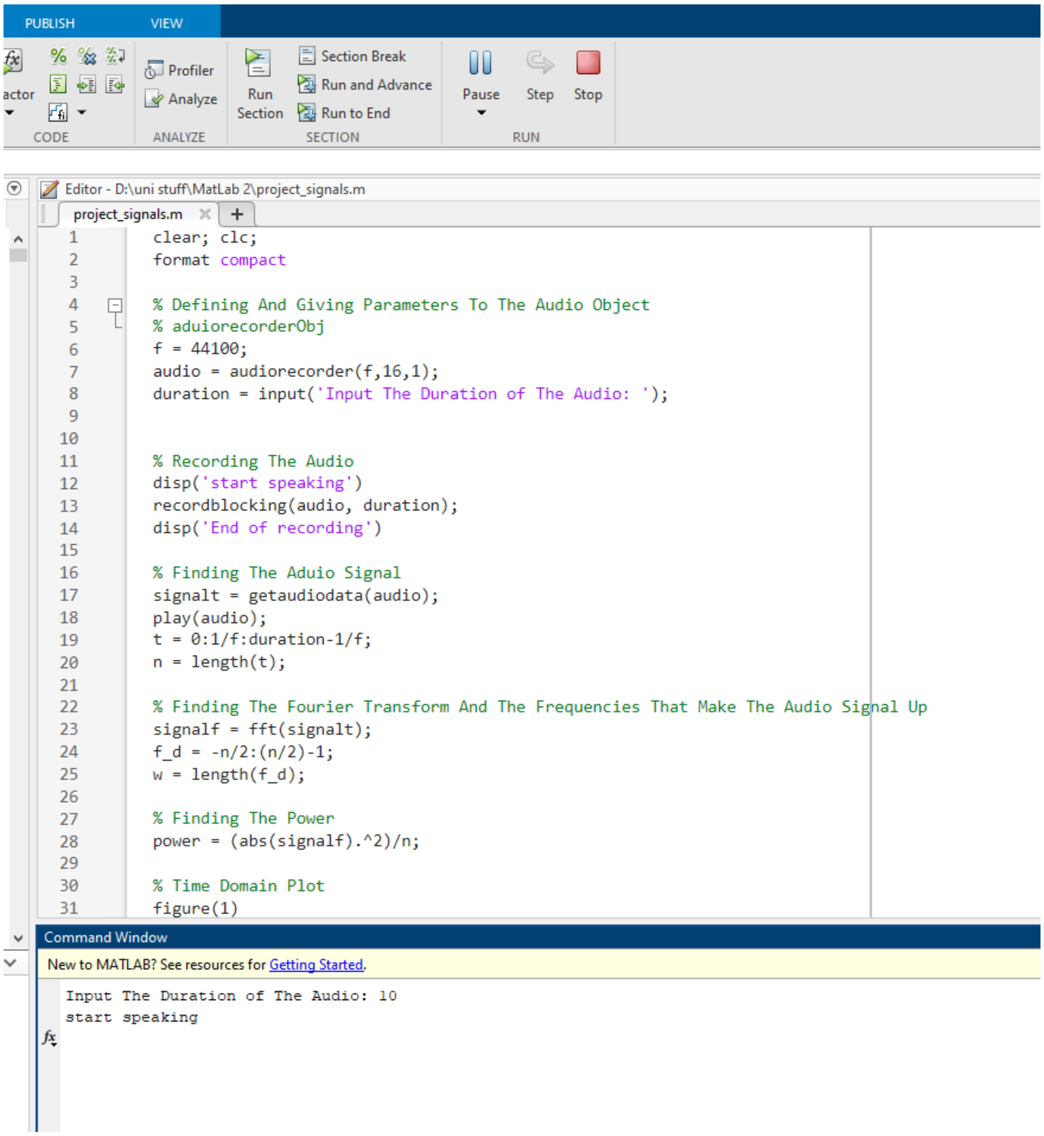

Figure 3: Inputting The duration.

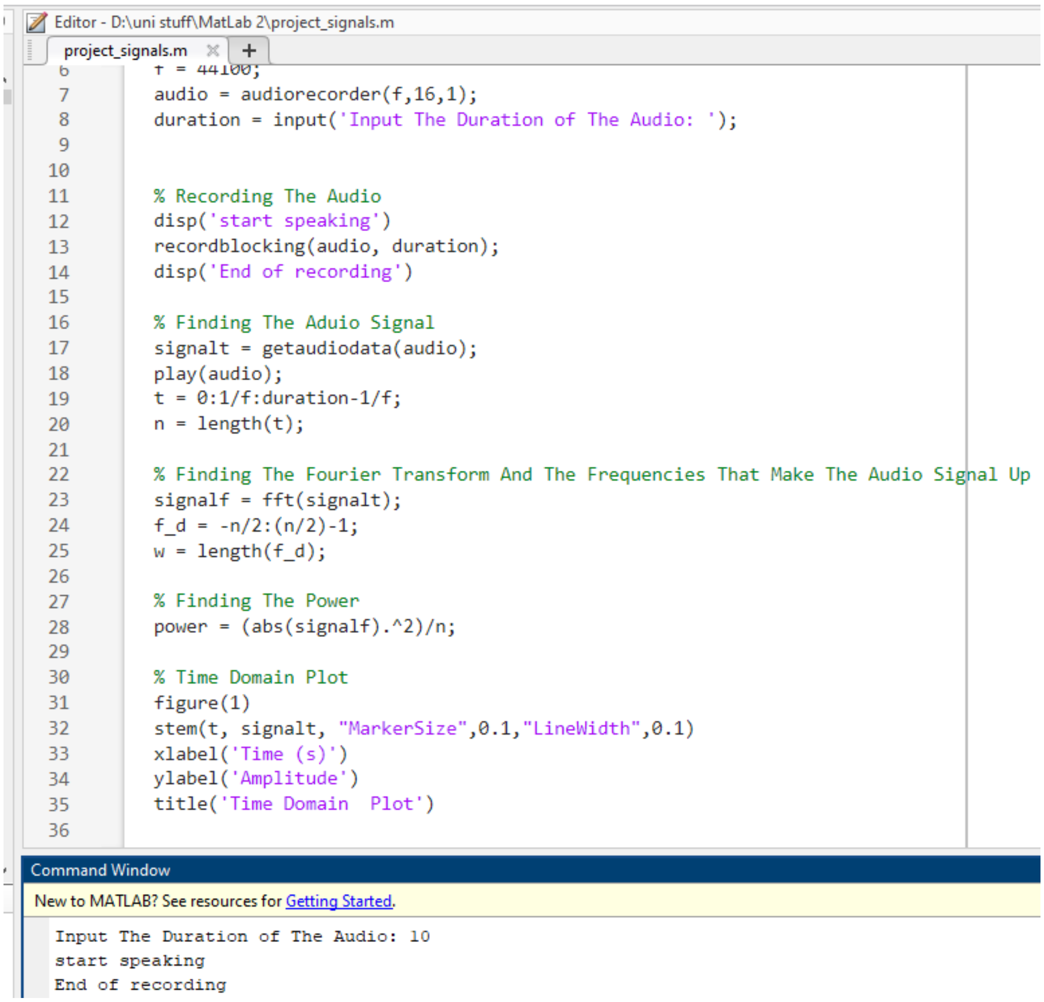

Figure 4: Recording The Audio.

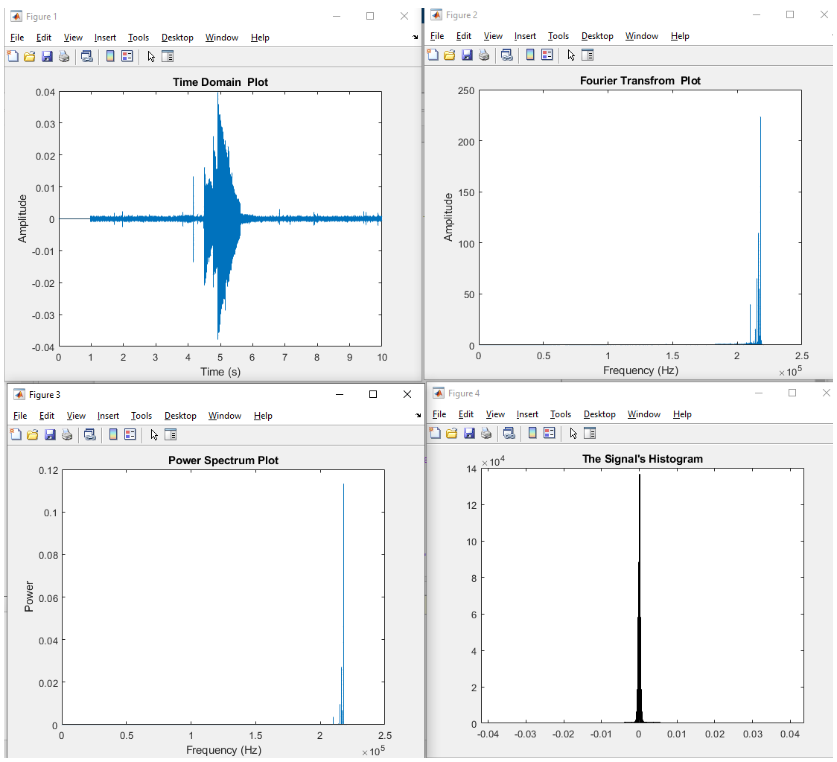

Figure 5: The Graphs.

```
max_freq_comp_1 =
      217883
Duration = 10.00 s
Energy =   5.33
Maximum Frequency Component = 217883
Average = 0.000000
Standard Deviation = 0.003478
Variance = 0.000012
Mode = 0.000000
Median = 0.000000
```

Figure 6: The Values.

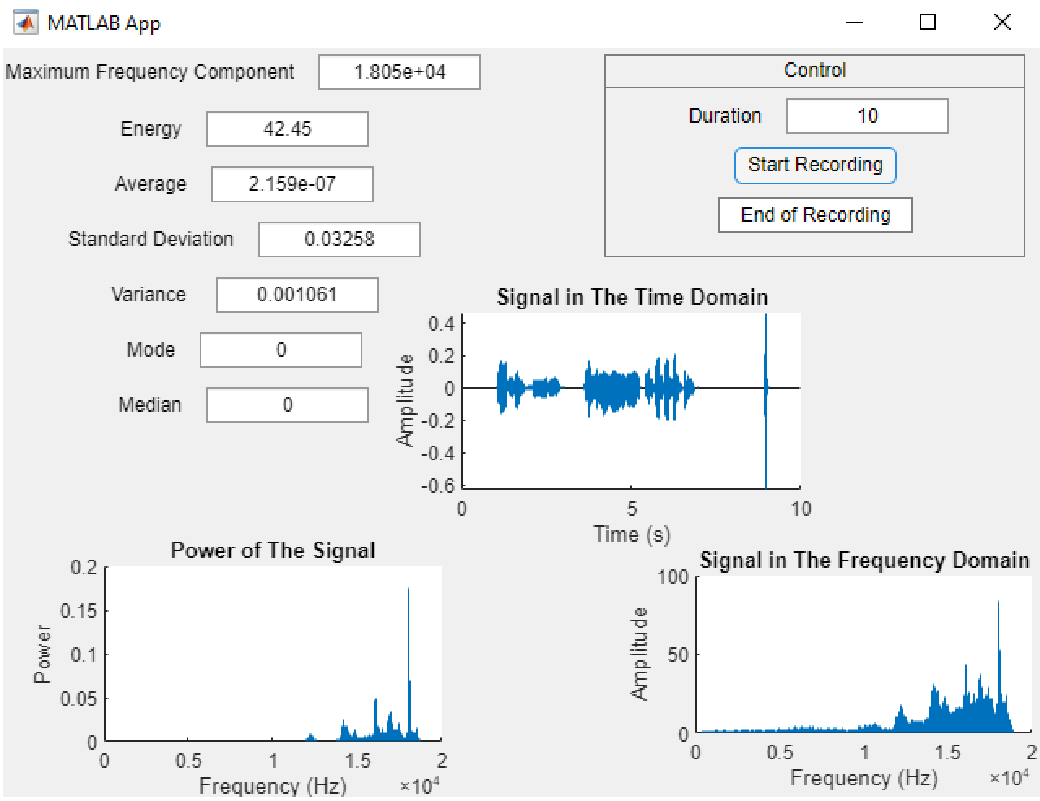

Figure 7: Using The GUI.

3.3 Analysis and Interpretation of Data

We can see from the testing that the program works as intended, where it is able to take an input from the computer system's audio input device, then do calculations on the input audio signal and plot the signal in the time domain alongside it's Fourier transform in the frequency domain, which completes all of the goals of the program.

4 Conclusion

4.1 Summary

In the end, the program we wrote in MATLAB was able to satisfy all the objectives, requirements and constraints we had put on it, and it was able to achieve its goal which is recording an audio signal and the visualize it and analyze it, and even though there was a ton of space for improvement we were satisfied with the results we got from the program even if it was not perfect.

4.2 Future Improvements and Takeaways

We learned a ton about audio signal visualization and analysis, and a basic idea how programs that specialize in these types of applications work. And we also learned more generally about digital signal processing which is a massive field with many applications and audio signal analysis is just one of these many fields.

4.3 Lessons Learned

Most of what was applied in the creation of the program was done after researching, watching video guides, and reading articles about the topic. For example, we were quite proficient with the basics of MATLAB, but we did not know how to create GUI's or apps with MATLAB, so we had to watch many video guides about designing GUI's, and more specifically, how to create GUI's in MATLAB, which after finishing the project we were comfortable with making.

4.4 Team Dynamics

- As our group was a duo, there was no team leader.
- Since we were a duo we were able to communicate personally without using any groups.
- Our goal was to create the program and learn how to do so along the way.
- In the end we were able to meet our objectives and goals, and we were able to create the program.

4.5 Impact Statement

What is the impact of your engineering solution on the economy, the environment, and the society?

	Environmental Impact Analysis							
Impact of your project	**Nature**	**Extent**	**Timing**	**Severity**	**Duration**	**Reversibility**	**Uncertainty**	**Significance**
The climate Example: *Does the project affect the emission of greenhouse gases into the atmosphere?*	Indirect Positive	Local	Immediate	Low	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: Does not affect the climate, as it is a simple matlab program.							
Use of Energy Example: *Does your project affect the energy consumption of the economy? How?*	Indirect Negative	Local	Immediate	Low	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: Has very low computitional cost							
Air quality Example: *Does the project have an effect on emissions of harmful air pollutants that might affect human health, damage crops or buildings or lead to deterioration in the environment (soil or rivers)?*	Indirect Positive	Local	Immediate	Low	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: Does not affect the air quality, as it is a simple matlab program.							
Biodiversity, flora, fauna and landscapes Example: *Does it affect endangered species, their habitats or ecologically-sensitive areas?*	Direct Positive	Local	Immediate	Low	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: Does not have an effect, as it is a simple matlab program.							

Figure 8: Impact 1.

Water quality and resources Example: *Does the project decrease or increase the quality or quantity of freshwater and groundwater?*	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: Does not have an effect, as it is a simple matlab program.							
Renewable or non-renewable resources Example: *Does the project reduce or increase use of non-renewable resources?*	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: Does not have an effect, as it is a simple matlab program.							
Sustainability Example: Does the option lead to more sustainable production and consumption? How?	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: Does not have an effect, as it is a simple matlab program.							
Waste production/generation/recycling Example: *Does the project affect waste production (solid, urban, agricultural, industrial, mining, radioactive or toxic waste) or*	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: Does not have an effect, as it is a simple matlab program.							

Figure 9: Impact 2.

how waste is treated, disposed of or recycled?								
	Economic Impact Analysis							
Impact of your project	**Nature**	**Extent**	**Timing**	**Severity**	**Duration**	**Reversibility**	**Uncertainty**	**Significance**
Economic Prosperity Example: *Does the project affect the GDP/capita, employment rate, household savings?*	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: Does not have an effect, as it is a simple matlab program.							
Investment Flows Example: *Does your project affect the flow of investment from outside the country? Does it encourage local investment in it?*	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: Does not have an effect, as it is a simple matlab program.							
Public Budgets or Services Example: *Does the project affect the budgets of hospitals, community services, older people services, transport services, service quality, schools, policing, municipality services..etc?*	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: Does not have an effect, as it is a simple matlab program.							
Market Mechanisms Example:	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: Does not have an effect, as it is a simple matlab program.							

Figure 10: Impact 3.

Does it affect the private sector business opportunities? Help companies reach more costumers? Change how business is done?								
Innovation, Research and Development Example: *Does the project have commercialization potential, lead to a potential patent? Does it allow others to innovate/research through it?*	Indirect Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: allows analysis of audio signals shich may or may not help research and development.							
Sustainable Consumption and Production Example: Does the project produce a sustainably consumed product or service? Can it be produced sustainably?	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: The program can be used in anyplace and anytime							

Figure 11: Impact 4.

	Social Impact Analysis							
Impact of your project	**Nature**	**Extent**	**Timing**	**Severity**	**Duration**	**Reversibility**	**Uncertainty**	**Significance**
Health and Longevity Example: *Does the project impact health and longevity? Does it affect physical activity, nutrition, chronic diseases, accidental injuries, independent living, mental wellbeing?*	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: Does not have an effect, as it is a simple matlab program.							
Safety Example: *Does your project affect safety of social environment, protection of older people against abuse, protection against risks, response to emergency cases, feelings of safety, physical safety?*	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: Does not have an effect, as it is a simple matlab program.							
Productive and Valued Activities Example: *Does the project increase leisure time, reduce stress, lead to positive behavior, increase productivity?*	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: Does not have an effect, as it is a simple matlab program.							

Figure 12: Impact 5.

Standard of Living Example: *Does it affect the quality of life? Make lives easier? Reduce poverty and deprivation? Increase life choices and opportunities?*	Direct Positive | Local | Immediate | High | Temporary | Reversible | Low Likelihood | Unimportant **Justification/Explanation:** Does not have an effect, as it is a simple matlab program.
Education/Life-long Learning Example: *Does the project affect literacy, use of ICT, chances of higher education, quality of education, life-long learning? Improve attainment of learning outcomes?*	Direct Positive | Local | Immediate | High | Temporary | Reversible | Low Likelihood | Unimportant **Justification/Explanation:** can help in understanding audio signals.
Quality of Social Interaction Example: *Does the project affect social connectedness, social participation, volunteering?*	Direct Positive | Local | Immediate | High | Temporary | Reversible | Low Likelihood | Unimportant **Justification/Explanation:** Does not have an effect, as it is a simple matlab program.

Figure 13: Impact 6.

Privacy and Personal Data Example: *Does the project reveal the user identities? Create potential private data leaks or identity theft?*	Direct Positive | Local | Immediate | High | Temporary | Reversible | Low Likelihood | Unimportant **Justification/Explanation:** Does not have an effect, as it is a simple matlab program.
Social Reasonability Example: *Does the project affect access to products and services for people of determination? Does it affect their integration into society? Does it affect their participation in the economy? Does it address their needs?*	Direct Positive | Local | Immediate | High | Temporary | Reversible | Low Likelihood | Unimportant **Justification/Explanation:** Does not have an effect, as it is a simple matlab program.

Figure 14: Impact 7.

5 Script File

```
clear; clc;
format compact

% Defining And Giving Parameters To The Audio Object
% aduiorecorderObj
f = 44100;
audio = audiorecorder(f,16,1);
duration = input('Input The Duration of The Audio: ');


% Recording The Audio
disp('start speaking')
recordblocking(audio, duration);
disp('End of recording')

% Finding The Aduio Signal
signalt = getaudiodata(audio);
play(audio);
t = 0:1/f:duration-1/f;
n = length(t);

% Finding The Fourier Transform And The Frequencies That Make The Audio Signal Up
signalf = fft(signalt);
f_d = -n/2:(n/2)-1;
w = length(f_d);

% Finding The Power
power = (abs(signalf).^2)/n;

% Time Domain Plot
figure(1)
stem(t, signalt, "MarkerSize",0.1,"LineWidth",0.1)
xlabel('Time (s)')
ylabel('Amplitude')
title('Time Domain Plot')

```

```
% Fourier Transfrom Plot
figure(2)
stem(f_d(n/2+1:n), abs(signalf(n/2+1:n)),
    "MarkerSize",0.1,"LineWidth",0.1)
xlabel('Frequency (Hz)')
ylabel('Amplitude')
title('Fourier Transfrom Plot')

% Power Spectrum Plot
figure(3)
stem(f_d(n/2+1:n), power(n/2+1:n),
    "MarkerSize",0.1,"LineWidth",0.1)
xlabel('Frequency (Hz)')
ylabel('Power')
title('Power Spectrum Plot')

% Calculating The Energy
energy = sum(abs(signalt).^2);

% Finding The Maximum Frequency Component
max_freq_comp_index = find(abs(signalf) == max(abs(signalf)));
max_freq_comp = f_d(max_freq_comp_index);
max_freq_comp_1 = max_freq_comp(2)

% Plotting The Histogram
figure(4)
histogram(signalt)
title('The Signal''s Histogram' )
% Statistical Features of The Recorded Audio Signal
Average = mean(signalt);
Standard_Deviation = std(signalt);
Variance = Standard_Deviation^2;
Mode = mode(signalt);
Median = median(signalt);

fprintf('Duration = %4.2f s \n',duration)
fprintf('Energy = %6.2f \n',energy)
fprintf('Maximum Frequency Component = %6.0f \n',max_freq_comp_1)
fprintf('Average = %1.6f \n',Average)
fprintf('Standard Deviation = %1.6f \n',Standard_Deviation)
```

```
fprintf('Variance = %1.6f \n',Variance)
fprintf('Mode = %1.6f \n',Mode)
fprintf('Median = %1.6f \n',Median)
```

6 GUI

```
classdef Project_GUI < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                    matlab.ui.Figure
        EnergyEditField             matlab.ui.control.NumericEditField
        EnergyEditFieldLabel        matlab.ui.control.Label
        MedianEditField             matlab.ui.control.NumericEditField
        MedianEditFieldLabel        matlab.ui.control.Label
        ModeEditField               matlab.ui.control.NumericEditField
        ModeEditFieldLabel          matlab.ui.control.Label
        VarianceEditField           matlab.ui.control.NumericEditField
        VarianceEditFieldLabel      matlab.ui.control.Label
        StandardDeviationEditField
            matlab.ui.control.NumericEditField
        StandardDeviationEditFieldLabel matlab.ui.control.Label
        AverageEditField            matlab.ui.control.NumericEditField
        AverageEditFieldLabel       matlab.ui.control.Label
        MaximumFrequencyComponentEditField
            matlab.ui.control.NumericEditField
        MaximumFrequencyComponentEditFieldLabel
            matlab.ui.control.Label
        ControlPanel                matlab.ui.container.Panel
        EditField                   matlab.ui.control.EditField
        StartRecordingButton        matlab.ui.control.Button
        DurationEditField           matlab.ui.control.NumericEditField
        DurationEditFieldLabel      matlab.ui.control.Label
        UIAxes3_4                   matlab.ui.control.UIAxes
        UIAxes3_2                   matlab.ui.control.UIAxes
        UIAxes3                     matlab.ui.control.UIAxes
    end

```

```
    % Callbacks that handle component events
    methods (Access = private)

        % Button pushed function: StartRecordingButton
        function StartRecordingButtonPushed(app, event)
            duration = app.DurationEditField.Value
            f = 4000;
            r = audiorecorder(f,16,1);
            app.EditField.Value = 'Start Speaking'
            recordblocking(r, duration);
            app.EditField.Value = 'End of Recording'
            signalt = getaudiodata(r);
            t = 0:1/f:duration-1/f;
            n = length(t);
            signalf = fft(signalt);
            f_d = -n/2:(n/2)-1;
            power = (abs(signalf).^2)/n;
            stem(app.UIAxes3,t, signalt,
                "MarkerSize",0.1,"LineWidth",0.1)
            stem(app.UIAxes3_2,f_d(n/2+1:n), abs(signalf(n/2+1:n)),
                "MarkerSize",0.1,"LineWidth",0.1)
            stem(app.UIAxes3_4,f_d(n/2+1:n), power(n/2+1:n),
                "MarkerSize",0.1,"LineWidth",0.1)
            energy = sum(abs(signalt).^2);
            max_freq_comp_index = find(abs(signalf) ==
                max(abs(signalf)));
            max_freq_comp = f_d(max_freq_comp_index);
            max_freq_comp_1 = max_freq_comp(2)
            Average = mean(signalt);
            Standard_Deviation = std(signalt);
            Variance = Standard_Deviation^2;
            Mode = mode(signalt);
            Median = median(signalt);
            app.MaximumFrequencyComponentEditField.Value =
                max_freq_comp_1
            app.EnergyEditField.Value = energy
            app.AverageEditField.Value = Average
            app.StandardDeviationEditField.Value = Standard_Deviation
            app.VarianceEditField.Value = Variance
            app.ModeEditField.Value = Mode
```

```
            app.MedianEditField.Value = Median
        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 = [100 100 640 463];
            app.UIFigure.Name = 'MATLAB App';

            % Create UIAxes3
            app.UIAxes3 = uiaxes(app.UIFigure);
            title(app.UIAxes3, 'Signal in The Time Domain')
            xlabel(app.UIAxes3, 'Time (s)')
            ylabel(app.UIAxes3, 'Amplitude')
            zlabel(app.UIAxes3, 'Z')
            app.UIAxes3.Position = [242 157 259 163];

            % Create UIAxes3_2
            app.UIAxes3_2 = uiaxes(app.UIFigure);
            title(app.UIAxes3_2, 'Signal in The Frequency Domain')
            xlabel(app.UIAxes3_2, 'Frequency (Hz)')
            ylabel(app.UIAxes3_2, 'Amplitude')
            zlabel(app.UIAxes3_2, 'Z')
            app.UIAxes3_2.Position = [386 6 255 152];

            % Create UIAxes3_4
            app.UIAxes3_4 = uiaxes(app.UIFigure);
            title(app.UIAxes3_4, 'Power of The Signal')
            xlabel(app.UIAxes3_4, 'Frequency (Hz)')
            ylabel(app.UIAxes3_4, 'Power')
            zlabel(app.UIAxes3_4, 'Z')
            app.UIAxes3_4.Position = [19 1 259 163];

```

```
            % Create ControlPanel
            app.ControlPanel = uipanel(app.UIFigure);
            app.ControlPanel.TitlePosition = 'centertop';
            app.ControlPanel.Title = 'Control';
            app.ControlPanel.Position = [373 335 259 125];

            % Create DurationEditFieldLabel
            app.DurationEditFieldLabel = uilabel(app.ControlPanel);
            app.DurationEditFieldLabel.HorizontalAlignment = 'right';
            app.DurationEditFieldLabel.Position = [47 76 50 22];
            app.DurationEditFieldLabel.Text = 'Duration';

            % Create DurationEditField
            app.DurationEditField = uieditfield(app.ControlPanel,
                'numeric');
            app.DurationEditField.HorizontalAlignment = 'center';
            app.DurationEditField.Position = [112 76 100 22];
            app.DurationEditField.Value = 1;

            % Create StartRecordingButton
            app.StartRecordingButton = uibutton(app.ControlPanel,
                'push');
            app.StartRecordingButton.ButtonPushedFcn =
                createCallbackFcn(app, @StartRecordingButtonPushed,
                true);
            app.StartRecordingButton.Position = [80 45 100 23];
            app.StartRecordingButton.Text = 'Start Recording';

            % Create EditField
            app.EditField = uieditfield(app.ControlPanel, 'text');
            app.EditField.HorizontalAlignment = 'center';
            app.EditField.Position = [70 15 120 22];

            % Create MaximumFrequencyComponentEditFieldLabel
            app.MaximumFrequencyComponentEditFieldLabel =
                uilabel(app.UIFigure);
            app.MaximumFrequencyComponentEditFieldLabel.HorizontalAlignment
                = 'right';
            app.MaximumFrequencyComponentEditFieldLabel.Position =
                [0 438 182 22];
```

```
            app.MaximumFrequencyComponentEditFieldLabel.Text = 'Maximum Frequency Component';

            % Create MaximumFrequencyComponentEditField
            app.MaximumFrequencyComponentEditField = uieditfield(app.UIFigure, 'numeric');
            app.MaximumFrequencyComponentEditField.HorizontalAlignment = 'center';
            app.MaximumFrequencyComponentEditField.Position = [197 438 100 22];

            % Create AverageEditFieldLabel
            app.AverageEditFieldLabel = uilabel(app.UIFigure);
            app.AverageEditFieldLabel.HorizontalAlignment = 'right';
            app.AverageEditFieldLabel.Position = [67 369 49 22];
            app.AverageEditFieldLabel.Text = 'Average';

            % Create AverageEditField
            app.AverageEditField = uieditfield(app.UIFigure, 'numeric');
            app.AverageEditField.HorizontalAlignment = 'center';
            app.AverageEditField.Position = [131 369 100 22];

            % Create StandardDeviationEditFieldLabel
            app.StandardDeviationEditFieldLabel = uilabel(app.UIFigure);
            app.StandardDeviationEditFieldLabel.HorizontalAlignment = 'right';
            app.StandardDeviationEditFieldLabel.Position = [38 335 107 22];
            app.StandardDeviationEditFieldLabel.Text = 'Standard Deviation';

            % Create StandardDeviationEditField
            app.StandardDeviationEditField = uieditfield(app.UIFigure, 'numeric');
            app.StandardDeviationEditField.HorizontalAlignment = 'center';
            app.StandardDeviationEditField.Position = [160 335 100 22];
```

```
            % Create VarianceEditFieldLabel
            app.VarianceEditFieldLabel = uilabel(app.UIFigure);
            app.VarianceEditFieldLabel.HorizontalAlignment = 'right';
            app.VarianceEditFieldLabel.Position = [64 301 55 22];
            app.VarianceEditFieldLabel.Text = 'Variance ';

            % Create VarianceEditField
            app.VarianceEditField = uieditfield(app.UIFigure,
                'numeric');
            app.VarianceEditField.HorizontalAlignment = 'center';
            app.VarianceEditField.Position = [134 301 100 22];

            % Create ModeEditFieldLabel
            app.ModeEditFieldLabel = uilabel(app.UIFigure);
            app.ModeEditFieldLabel.HorizontalAlignment = 'right';
            app.ModeEditFieldLabel.Position = [74 267 35 22];
            app.ModeEditFieldLabel.Text = 'Mode';

            % Create ModeEditField
            app.ModeEditField = uieditfield(app.UIFigure, 'numeric');
            app.ModeEditField.HorizontalAlignment = 'center';
            app.ModeEditField.Position = [124 267 100 22];

            % Create MedianEditFieldLabel
            app.MedianEditFieldLabel = uilabel(app.UIFigure);
            app.MedianEditFieldLabel.HorizontalAlignment = 'right';
            app.MedianEditFieldLabel.Position = [69 233 44 22];
            app.MedianEditFieldLabel.Text = 'Median';

            % Create MedianEditField
            app.MedianEditField = uieditfield(app.UIFigure,
                'numeric');
            app.MedianEditField.HorizontalAlignment = 'center';
            app.MedianEditField.Position = [128 233 100 22];

            % Create EnergyEditFieldLabel
            app.EnergyEditFieldLabel = uilabel(app.UIFigure);
            app.EnergyEditFieldLabel.HorizontalAlignment = 'right';
            app.EnergyEditFieldLabel.Position = [70 403 43 22];
```

```
            app.EnergyEditFieldLabel.Text = 'Energy';

            % Create EnergyEditField
            app.EnergyEditField = uieditfield(app.UIFigure,
                'numeric');
            app.EnergyEditField.HorizontalAlignment = 'center';
            app.EnergyEditField.Position = [128 403 100 22];

            % 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 = Project_GUI

            % Create UIFigure and components
            createComponents(app)

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

            if nargout == 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
```