## AIN SHAMS UNIVERSITY FACULTY OF ENGINEERING

# ECE251: Signals and Systems Fundamentals

## Fall 2024 Project Report

| Student Name | ID |
|---|---|
| Mostafa Ahmed Abdelsattar | 2201092 |
| Tarek hazem Salah Abdelhamed | 2200680 |
| Omar Mohamed Fathi | 2300781 |
| Ahmed Muhammad Mohey | 1802942 |
| Mohamed Abdel Nasser Abouelkhier | 2300778 |
| Ahmed Mahmoud Elmorsy | 2200725 |
| Ahmed mohamed mohyeldin | 2200955 |
| Mahmoud Ibrahim Mohamed | 2101110 |
| Mohamed Samy Abdelfattah | 2200685 |

# Table of Contents

# Introduction

Signal processing is a fundamental aspect of modern engineering, with applications spanning various fields such as telecommunications, audio engineering, biomedical signal analysis, and more. This project, undertaken as part of the course *Signals and Systems Fundamentals*, aims to deepen our understanding of key concepts by applying them to a real-world scenario: the analysis and processing of audio signals.

The primary goal of this project is to develop a MATLAB-based code that performs comprehensive analysis and filtering of an audio signal. Through this activity, we aim to reinforce the theoretical concepts covered in the course, such as time-domain and frequency-domain representations, energy calculations, and system responses to filtering.

# Methodology

As a team, we wanted to divide the project into smaller tasks that each of us could do, so the whole project was done in the end. We divided the project into those specific tasks:

- Making the audio file of us mentioning our name
- Plotting the audio signal in the time domain
- Calculating the signal energy from the time domain
- Finding the frequency domain representation of the signal and plotting it
- Calculating the signal energy from the frequency domain
- Comparing it with the calculated value in the time domain
- Applying a Butterworth filter of 10th order to change the original frequency components of the audio signal and plotting the filtered signal in the frequency domain
- Finding the corresponding signal in the time domain for the filtered signal and plotting it.
- Saving the filtered audio file in time domain on the hard disk
- Collecting the distributed code in only one file
- Verifying its functionality

# Implementation and Plots

In the coming headings, I will provide each task with the corresponding code and plot, knowing that the whole code is provided with the other deliverables.

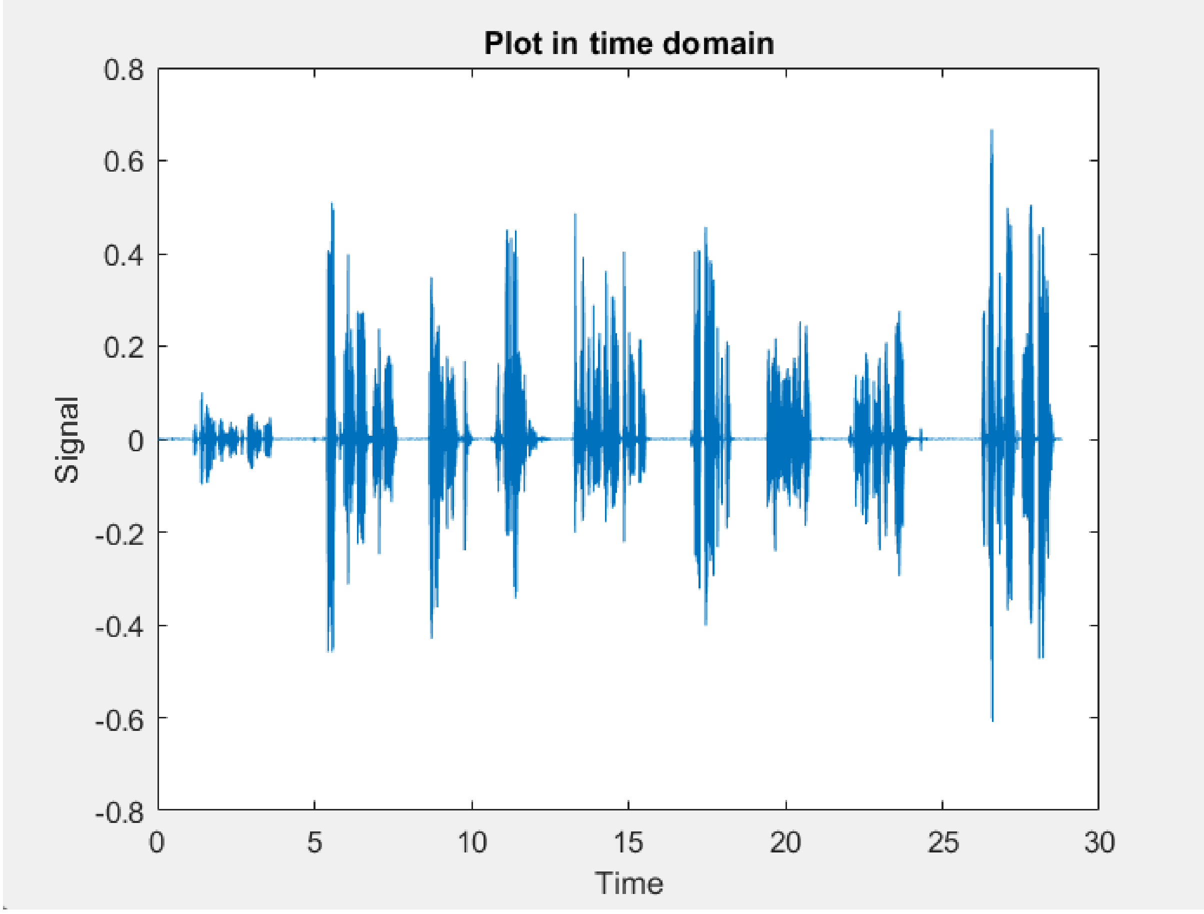
## Reading the audio file and plotting it in the time domain

### Code

PR_FI.m *

```
%reading the audio file

[signal_ft,fs] = audioread("ST.ogg");

%plotting in time domain


x = length(signal_ft);

t = (0:x-1)/fs;
figure('Name', 'Time Domain', 'NumberTitle', 'off');

plot(t,signal_ft)
xlabel('Time')
ylabel('signal')
title("Plot in time domain")

```

## Plot




# Calculating the signal energy from the time domain

## Code

```
%calculating energy in time domain


Energy_in_time = sum(signal_ft.^2);

```

### Output

Energy_in_time

1x1 double

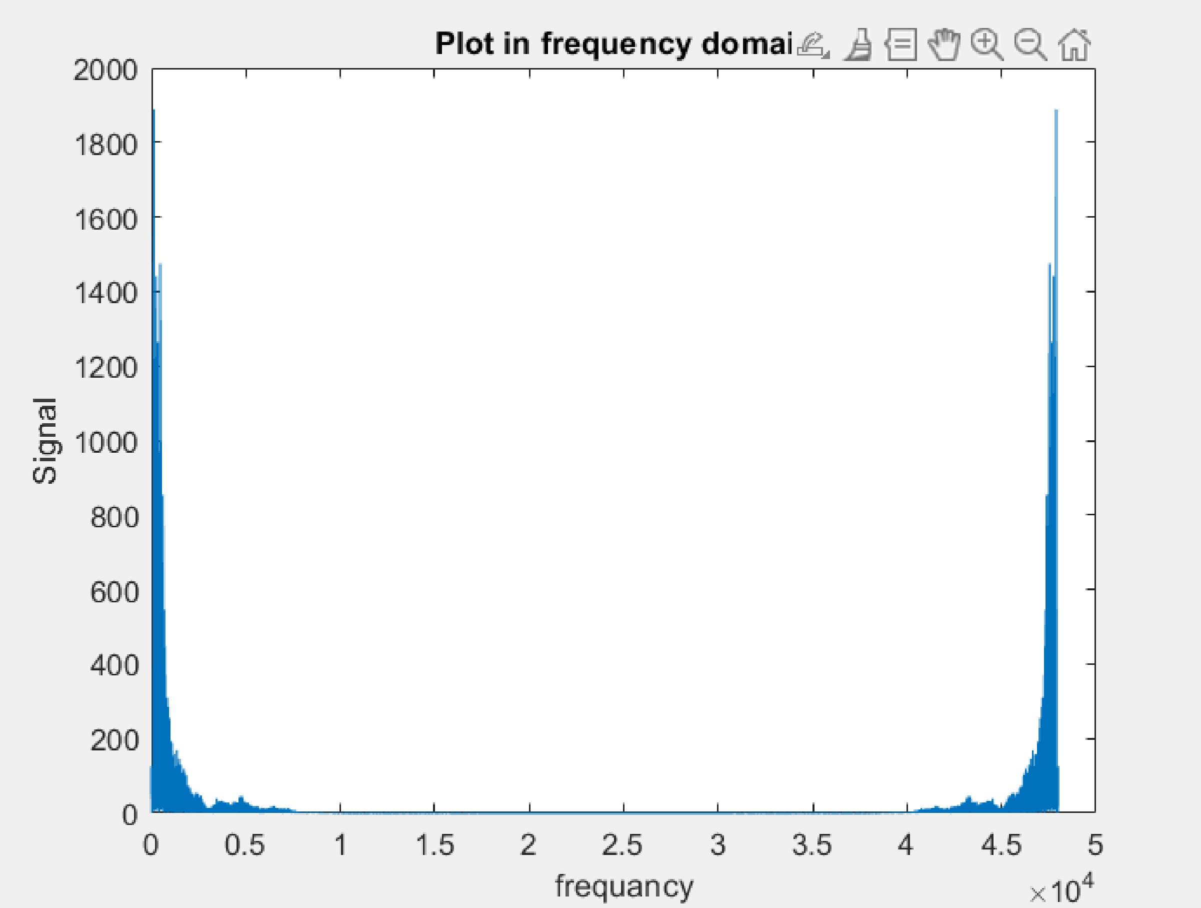
| | 1 |
|---|---|
| 1 | 4.3121e+03 |

## Finding the frequency domain representation of the signal and plotting it

### Code

```
figure('Name', 'Frequency Domain', 'NumberTitle', 'off');
freq_domain = fft(signal_ft);
l=length(freq_domain);
f=(0:l-1)*(fs/l);
plot(f,abs(freq_domain))
xlabel('frequancy')
ylabel('Signal')
title('Plot in frequency domain')
```

## Plot

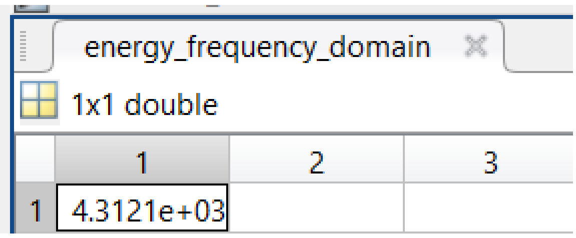



# Calculating the signal energy from the frequency domain

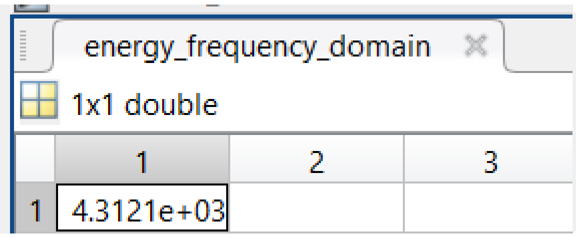
## Code

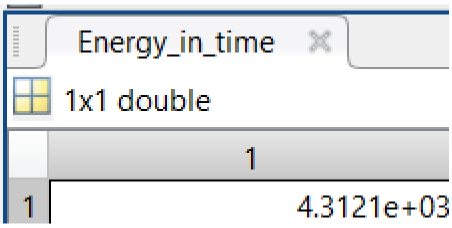
```
energy_frequency_domain = sum(abs(freq_domain).^2) / l;
```

## Output



## Comparing it with the calculated value in the time domain



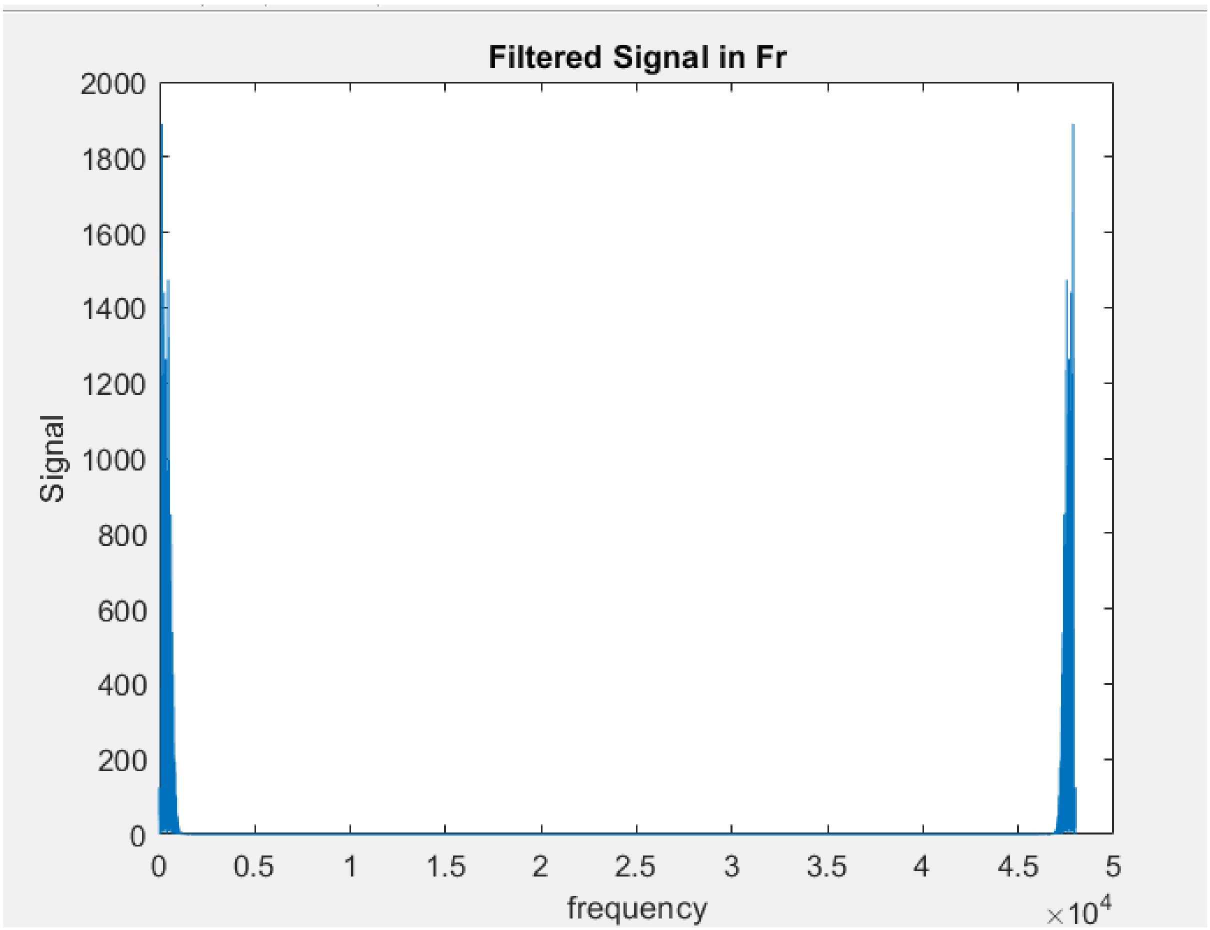

We see from both outputs that they have the value of $4.321 * 10^3\ joule$ , which verifies Persevel's relationship.

## Applying a Butterworth filter of 10th order to and plotting the filtered signal in the frequency domain

### Code

```
fc=800; %cut off frequency
[b,a] = butter(10,fc/(fs/2),'low'); %low pass filter of 10th order

filtered_signal=fft(filter(b,a,signal_ft));

f=(0:N-1)*(fs/N);
figure('Name', 'Signal in frequency after applying a filter', 'NumberTitle', 'off');
plot(f,abs(filtered_signal))
xlabel("frequency");
ylabel("Signal");
title("Filtered Signal in Fr")
```

## Plot

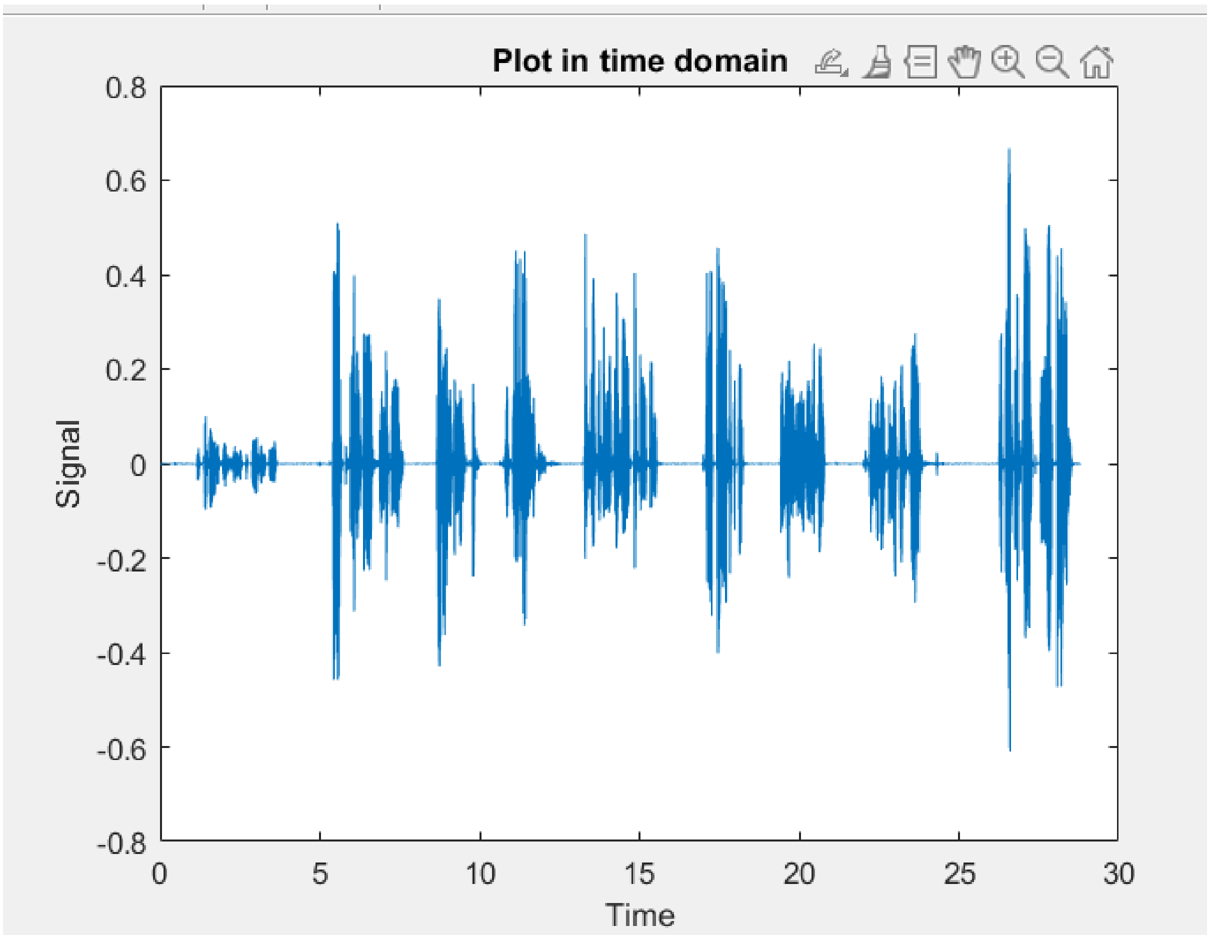



# Finding the corresponding signal in the time domain for the filtered signal and plotting it.

## Code

```
final_audio=ifft(filtered_signal);
figure('Name', 'Signal in time after applying a filter', 'NumberTitle', 'off');
plot(t,final_audio)
xlabel('Time')
ylabel('Signal')
title("Plot in time domain")
```

Plot




## Saving the filtered audio file in the time domain on the hard disk

Code

```
audiowrite("final audio.ogg",final_audio,fs)
```

# Team Contributions

- ***Mostafa Ahmed Abdelsattar***
  *Mostafa was responsible for integrating all individual code components into a single cohesive file. He verified the functionality of the complete program and provided assistance to teammates in troubleshooting issues.*
- ***Tarek Hazem***
  *Tarek developed the code to read the audio file and plot the signal in the time domain, forming the foundation for subsequent analysis.*
- ***Mohamed Samy***
  *Mohamed implemented the code for calculating the signal's energy in the time domain, ensuring accurate energy analysis.*
- ***Mohamed Abdlennaser***
  *Mohamed worked on the code to find the frequency domain representation of the signal, enabling further insights into its frequency components.*
- ***Omar Fathi***
  *Omar reviewed and verified the accuracy of the frequency domain representation code. He also developed the code for plotting the frequency domain signal.*
- ***Ahmed Mohamed Mohyeldin***
  *Ahmed wrote the code to calculate the signal's energy in the frequency domain and assisted in comparing it with the time-domain energy.*
- ***Ahmed Muhammad Mohey***
  *Ahmed developed the code for applying a 10th-order Butterworth filter (LPF or HPF) to the signal. He also implemented the code to plot the filtered signal.*
- ***Mahmoud Ibrahim***
  *Mahmoud was responsible for converting the filtered signal back into the time domain and writing the corresponding code to plot it.*
- ***Ahmed Mahmoud Elmorsy***
  *Ahmed wrote the code to save the filtered signal in the same format as the original audio file onto the hard disk.*

# Conclusion

In contrast, by analyzing and processing an audio signal, we gained valuable insights into the key concepts of time-domain and frequency-domain representations, signal energy calculations, and the application of filtering techniques. This project not only reinforced our theoretical understanding of signal processing but also enhanced our practical programming skills and our ability to work collaboratively in a team. By breaking down complex problems into manageable tasks, we were able to apply learned concepts effectively and gain experience with real-world signal processing challenges.

# Discussion Video Link

Video Link

# References

*Textbook*

*Signals and Systems*

*Alan V. Oppenheim, Alan S. Willsky , S. Hamid Nawab*

*MATLAB documentations*

*https://www.mathworks.com/help*