# Signal Processing Course Project

This presentation showcases our signal processing project, exploring three distinct tasks: bird detection, heart rate analysis from an electrocardiogram (ECG), and loudness segmentation.

**Srihari Padmanabhan**

(2023102021)

**MP Samartha**

(2023102038)

**Kiruba Selvakumar**

(2023112010)

# Task 1: Bird Detection

### 1 Frequency Characteristics

Identify these for each bird. Here we use Peak Frequencies and Freq. Spread.

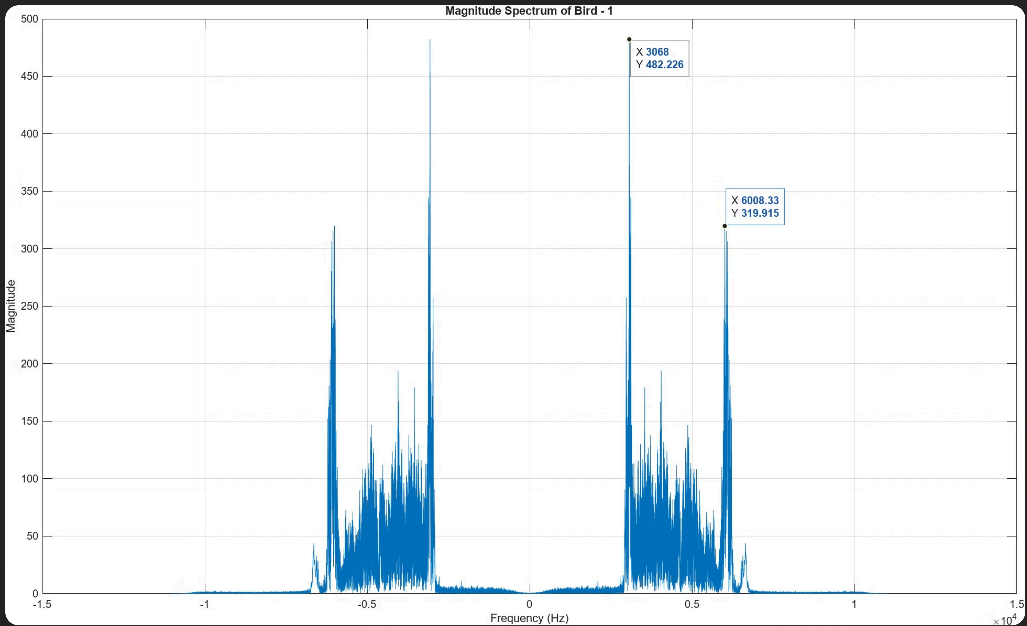
### 2 Signal Processing

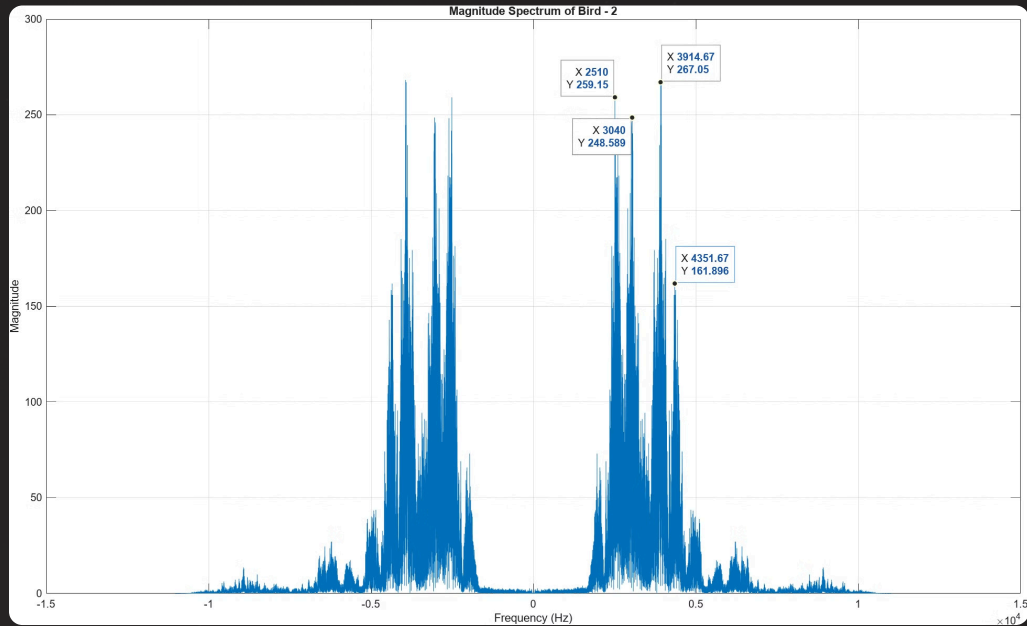
Next, we applied various signal processing techniques like filtering to eliminate unwanted frequencies.

### 3 Spectral Analysis

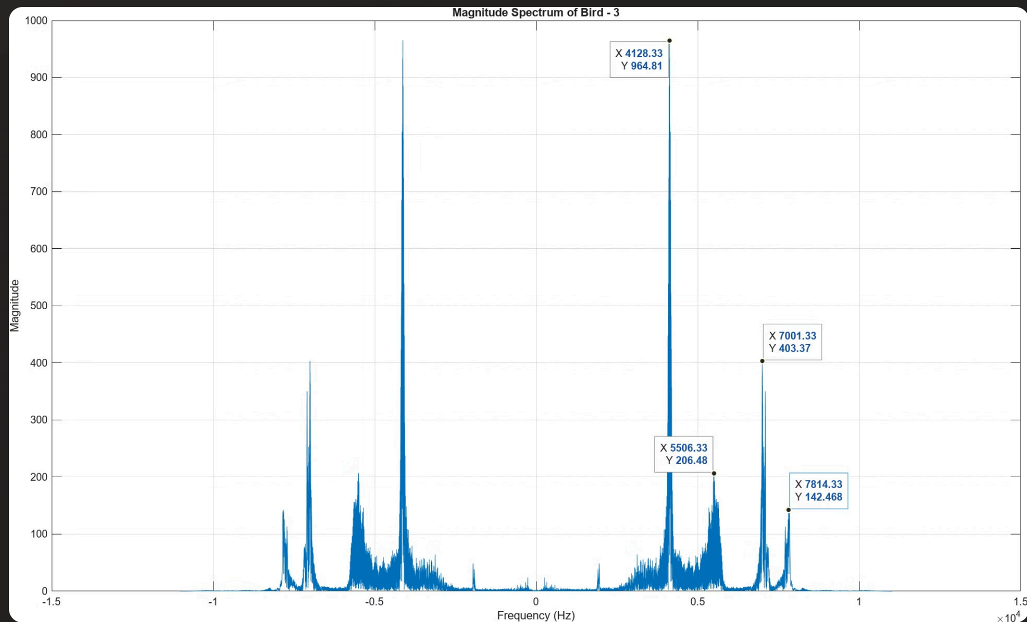
Finally, we analyzed the bird's call in the frequency domain, identifying characteristic peak frequencies and frequency spread. Using these we determine the bird species.

# Frequency Spectrum of Reference Bird Signals



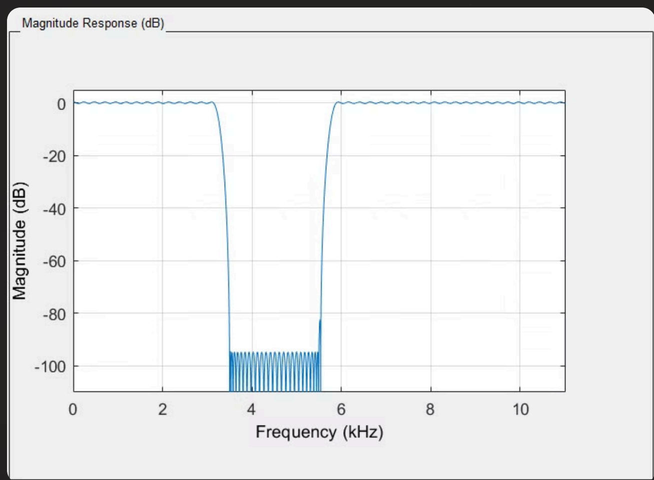




- Bird -1 :
  - Peak Frequency around 3k, 6k Hz
  - Frequency Spread of 4.67k Hz
- Bird-2 :
  - Peak Frequency around 2.5k, 3k, 4k, 4.3k Hz
  - Frequency Spread of 3.56k Hz
- Bird-3 :
  - Peak Frequency around 4.1k, 5.5k, 7k, 7.8k Hz
  - Frequency Spread of 6.64k Hz

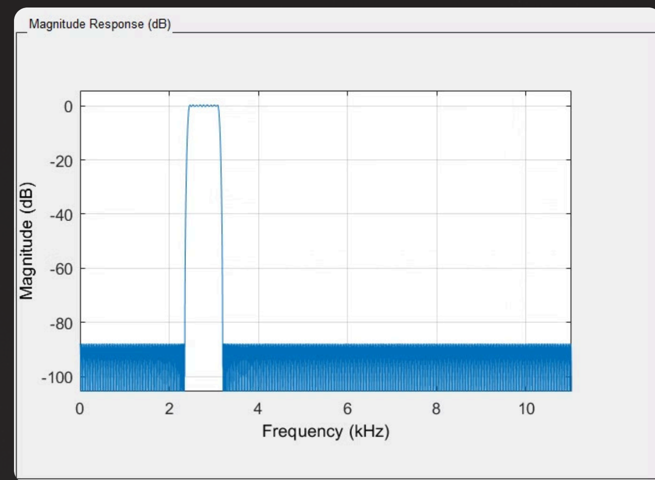


# Filter Design

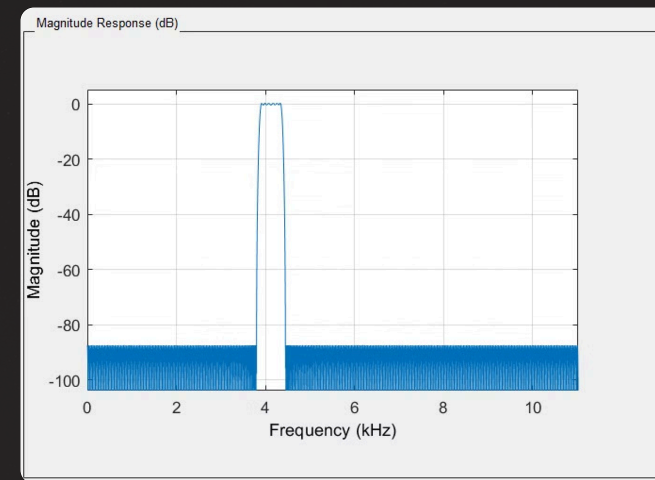
Bird-1 Bandstop



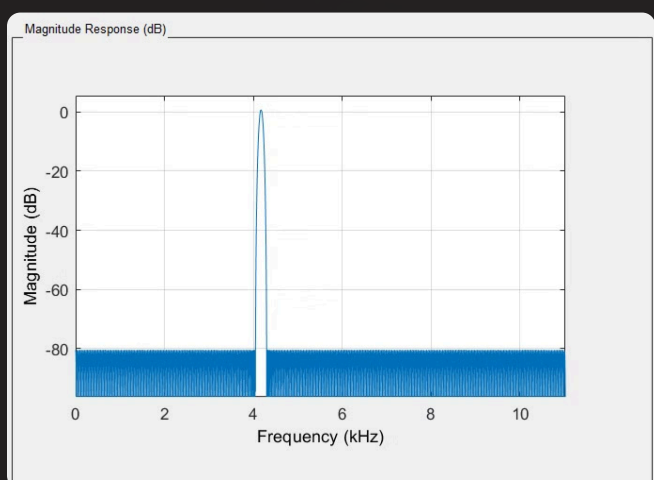

Bird-2 Bandpass1



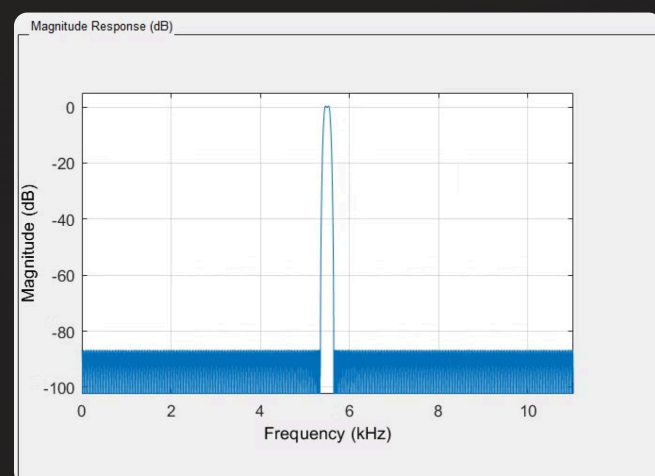

Bird-2 Bandpass2



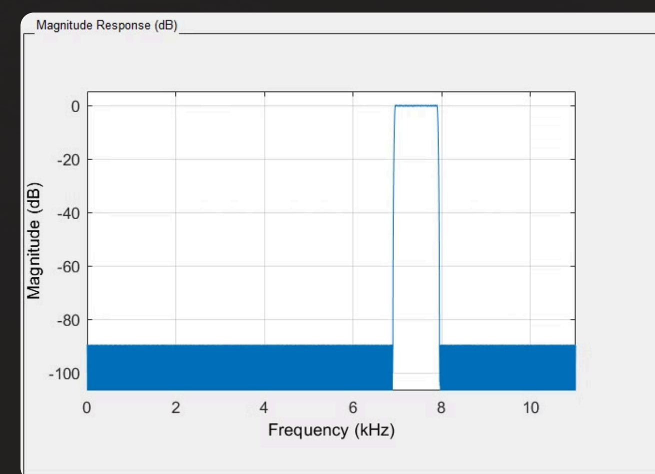

Bird-3 Bandpass1




Bird-3 Bandpass2

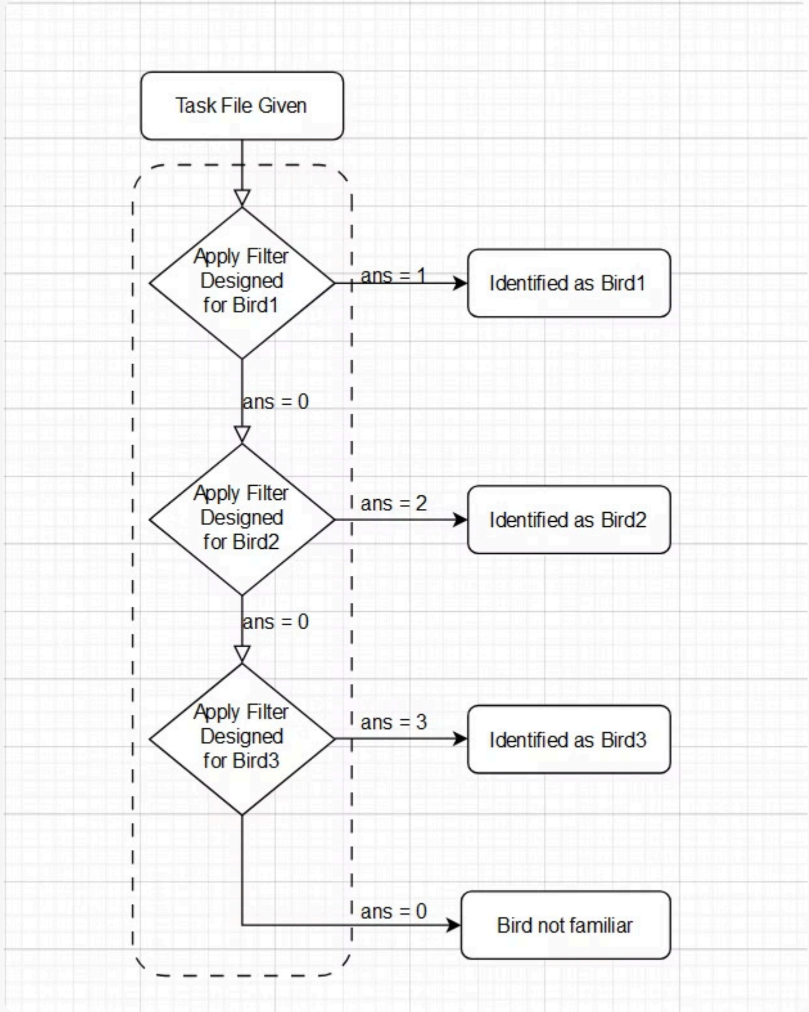



Bird-3 Bandpass3



# Logic Diagram

The logic diagram illustrates the flow of the signal processing pipeline, from input signals to final output. We apply corresponding filters and and identify the bird species based on the output variable, "ans".



# Command Window Output

The final command window output, detecting the birds in the given task files.

Command Window

```
Analysing File Task - 1
ans = 0
ans = 0
ans = 3
Task file F1.wav identified as Bird- 3

Analysing File Task - 2
ans = 1
Task file F2.wav identified as Bird- 1

Analysing File Task - 3
ans = 0
ans = 2
Task file F3.wav identified as Bird- 2

Analysing File Task - 4
ans = 0
ans = 0
ans = 3
Task file F4.wav identified as Bird- 3

Analysing File Task - 5
ans = 1
Task file F5.wav identified as Bird- 1

Analysing File Task - 6
ans = 0
ans = 0
ans = 3
Task file F6.wav identified as Bird- 3

Analysing File Task - 7
ans = 1
Task file F7.wav identified as Bird- 1

Analysing File Task - 8
ans = 0
ans = 2
Task file F8.wav identified as Bird- 2
```

# Task 2: Heart Rate from Electrocardiogram(ECG)

1. **Frequency Analysis and Noise Filtering**

   Using the frequency spectrum of clean signal E1, we located and filtered noisy frequencies present in the other two signals

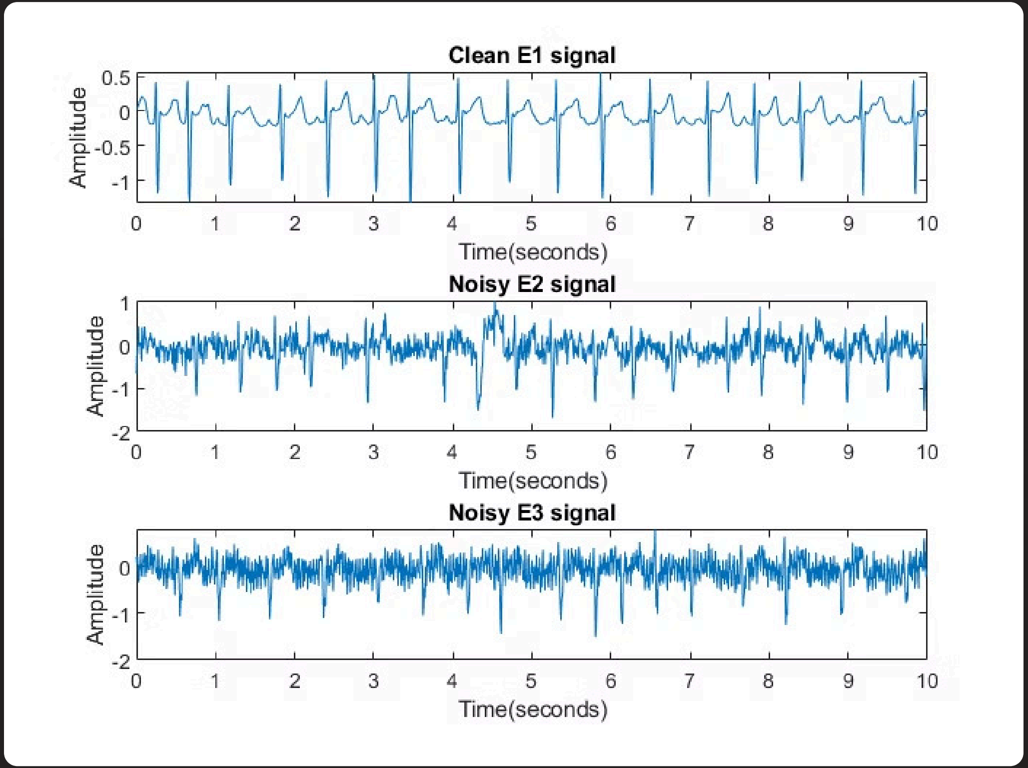
2. **Pan-Tompkins Algorithm**

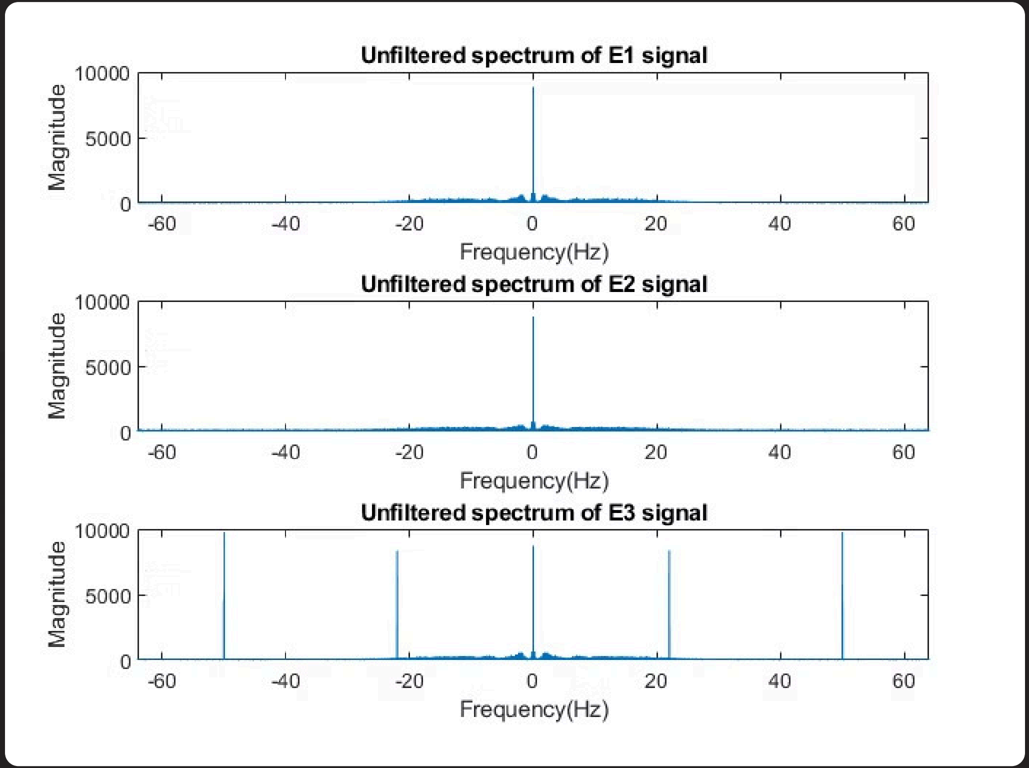
   After a preliminary noise filtering, we implemented the Pan-Tompkins algorithm to amplify QRS complexes.

3. **Heart Rate Estimation**

   We used a simple threshold based peak detector to calculate the heart rate (Beats Per Minute)

# Given signals and their frequency spectra

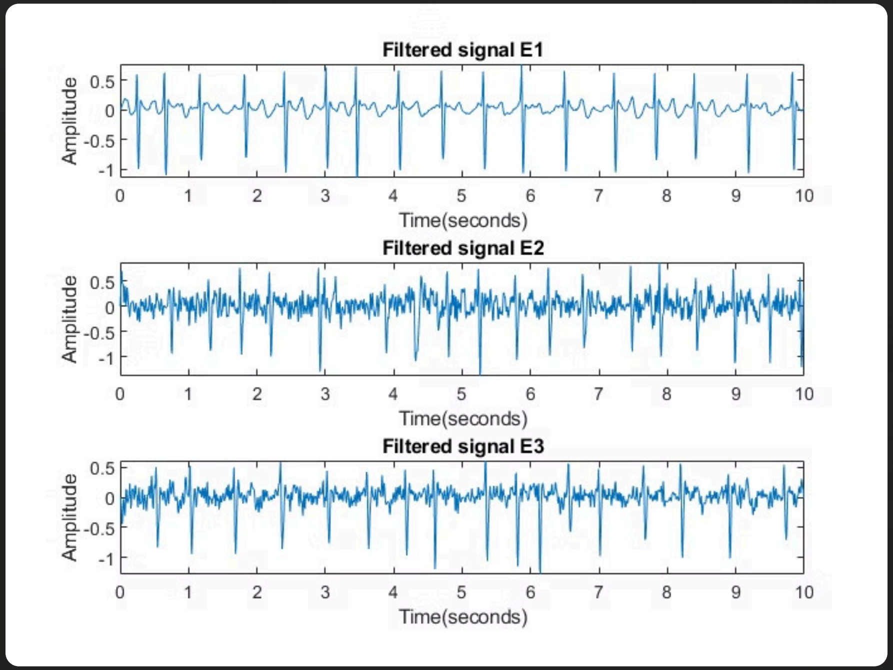


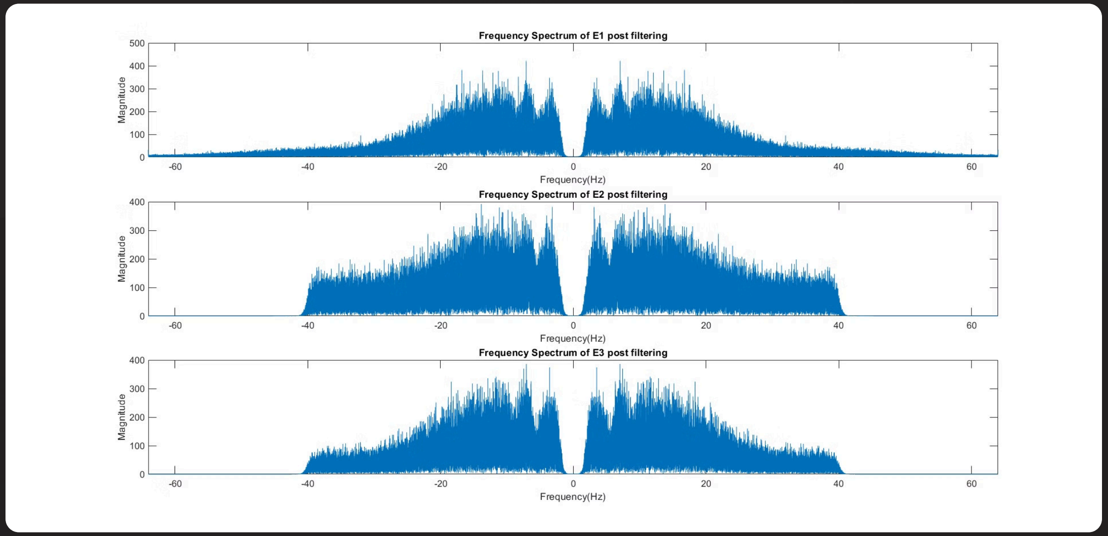



# Filtering in the frequency domain

- Bandpass filter applied on E2 and E3 to filter out high frequency noise missing in E1, and low frequency disturbances of the baseband due to corrupted signal measurement. For clear comparison, highpass filter is also applied on E1.
- Notch filter applied on E3 to remove noisy peak at 22Hz.

Post Filtering, we have the following signal profiles:



# Amplification of QRS complex

1

### Differentiation

The Pan-Tompkins algorithm suggests using a digital differentiator (We used the LCCDE $y[n] = 0.125(2x[n] + x[n-1] - x[n-3] - 2x[n-4])$ to characterize sharp R peaks.

2

### Squaring

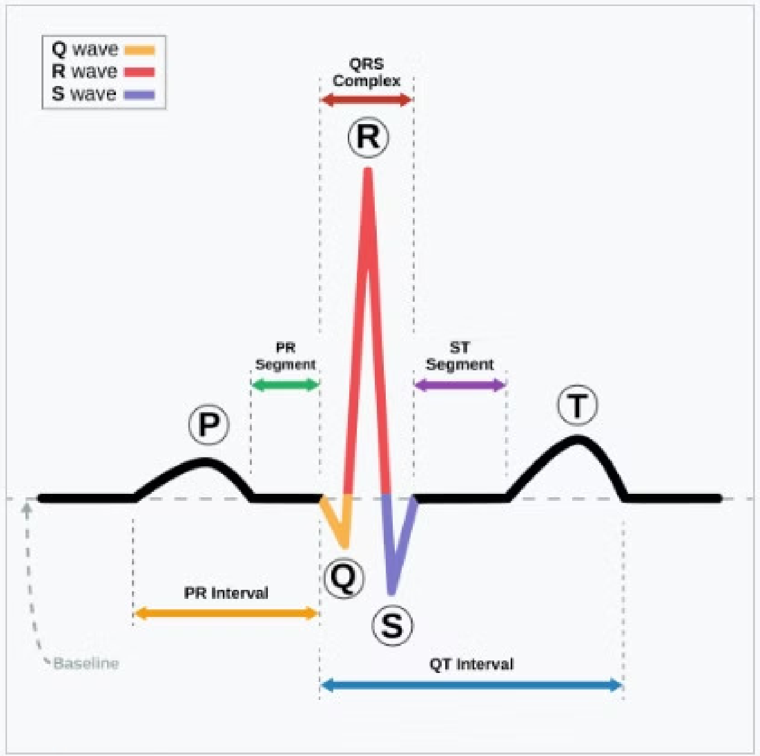
Squaring the differentiated signal amplifies the already enhanced R peaks further, but also introduces some noisy peaks.

3

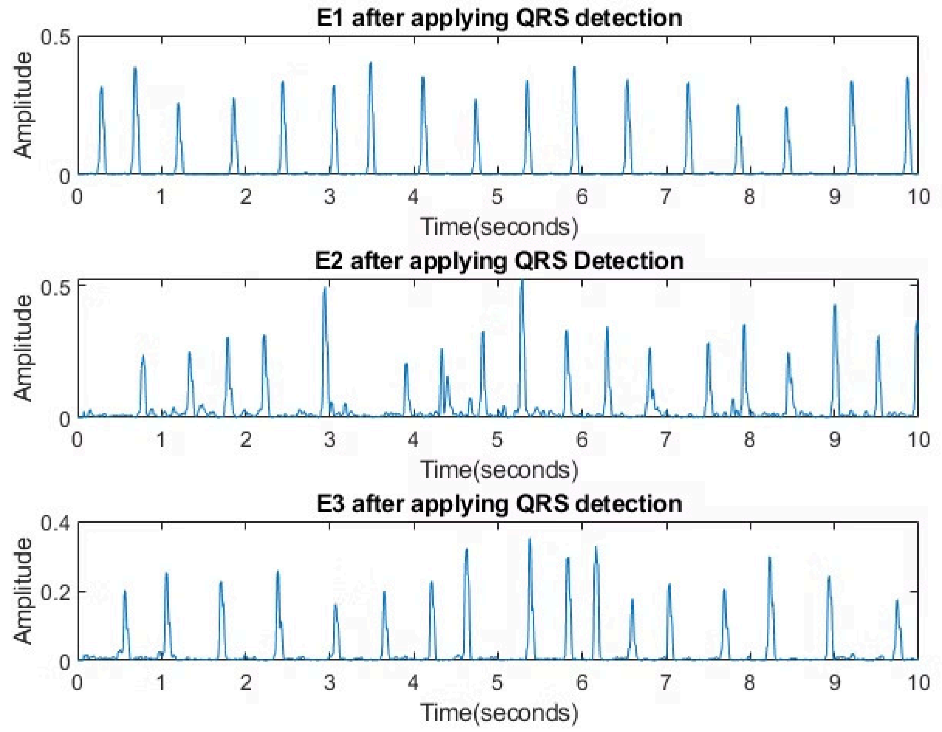
### Moving Average Integration

A moving average window (e.g., width 3) acts as a smoothing filter. This preserves the high-energy R peaks while suppressing remaining noise (P/T waves).

# Amplified R Peaks of the three signals

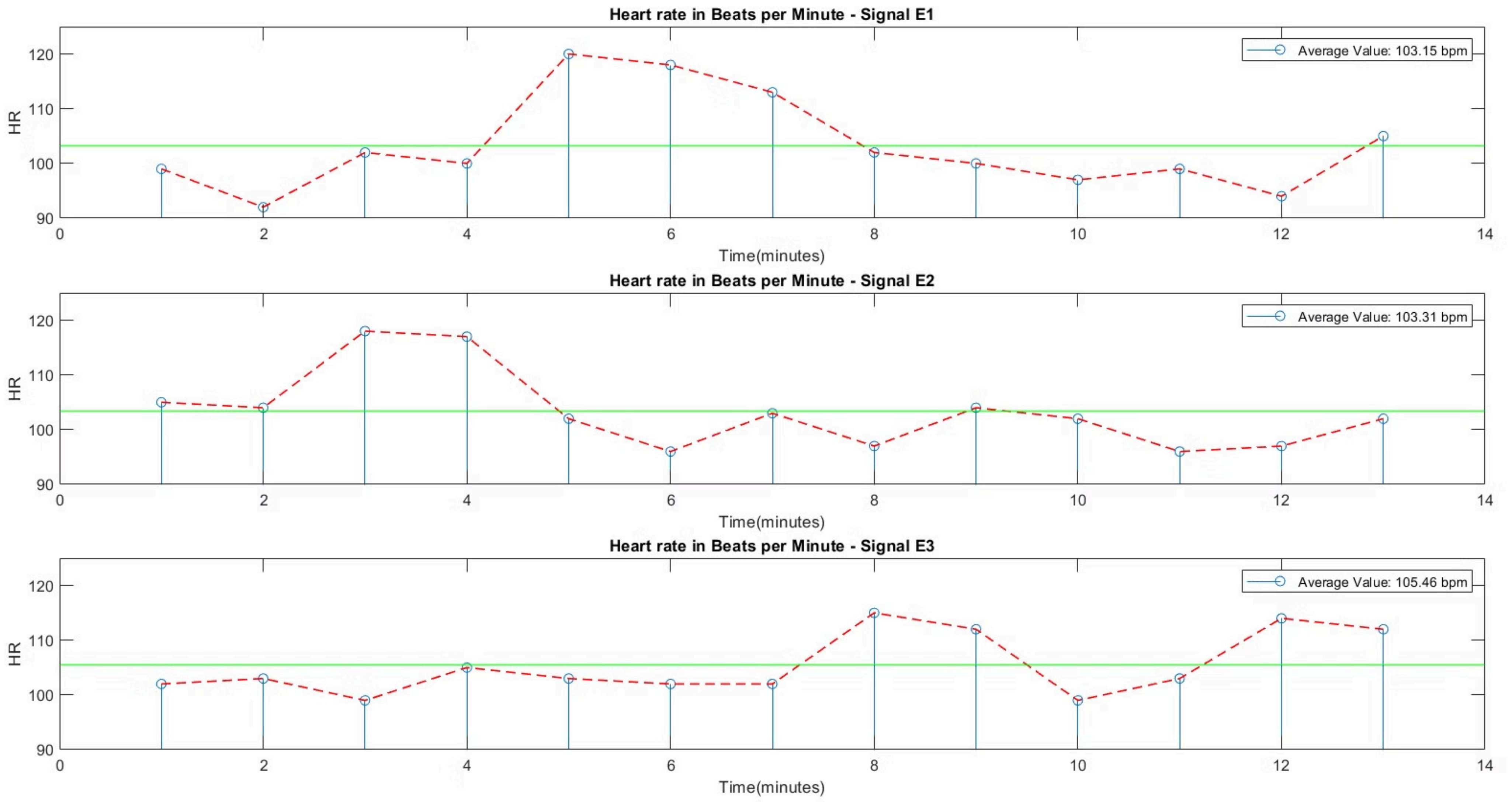



ECG beat




Made with Gamma

The peak detection algorithm uses an amplitude threshold of 0.1 and requires a minimum distance of 20 samples between peaks, since the minimum peak distance in the ECG does not go under 40 samples. After storing the number of peaks counted in each minute interval for each signal, we can summarize our BPM calculations:



# Task 3: Loudness Segmentation

1. **Speech Recognition (Word Identification)**

   We determine the start and end times of words spoken in the given speech sample.

2. **Loudness Estimation**

   We implemented a loudness estimation algorithm to calculate the *perceived* loudness of the audio signal.

3. **Segmentation**

   We then segmented the audio signal based on significant changes in loudness, resulting in distinct segments.

# Understanding "Loudness"

The energy content (i.e, sum of the square amplitude of the signal, a.k.a volume) is the term used in day-to-day life.

Loudness is a *perceived* quality of sound. The human ear isn't evenly sensitive to all frequencies, making the sound we perceive different than the actual noise. Loudness, a measure of sound pressure, is calculated in dBSPL units.

RMS is a commonly used measure to calculate the loudness, although is incomplete as it doesn't cover the frequency aspect of human perception. We need a new method.

### New Measurement Standard

*"A-weighting"* introduced based on Fletcher-Munson curves.

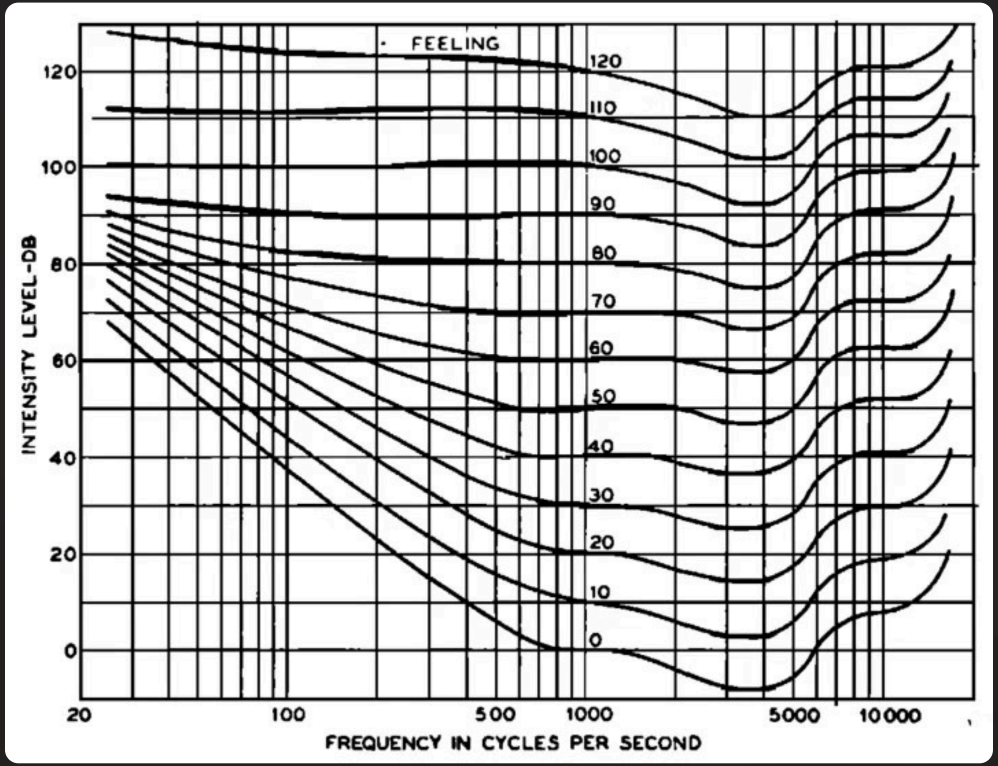
### Listener Consideration

Middle and inner ear structure, body of the listener now considered.

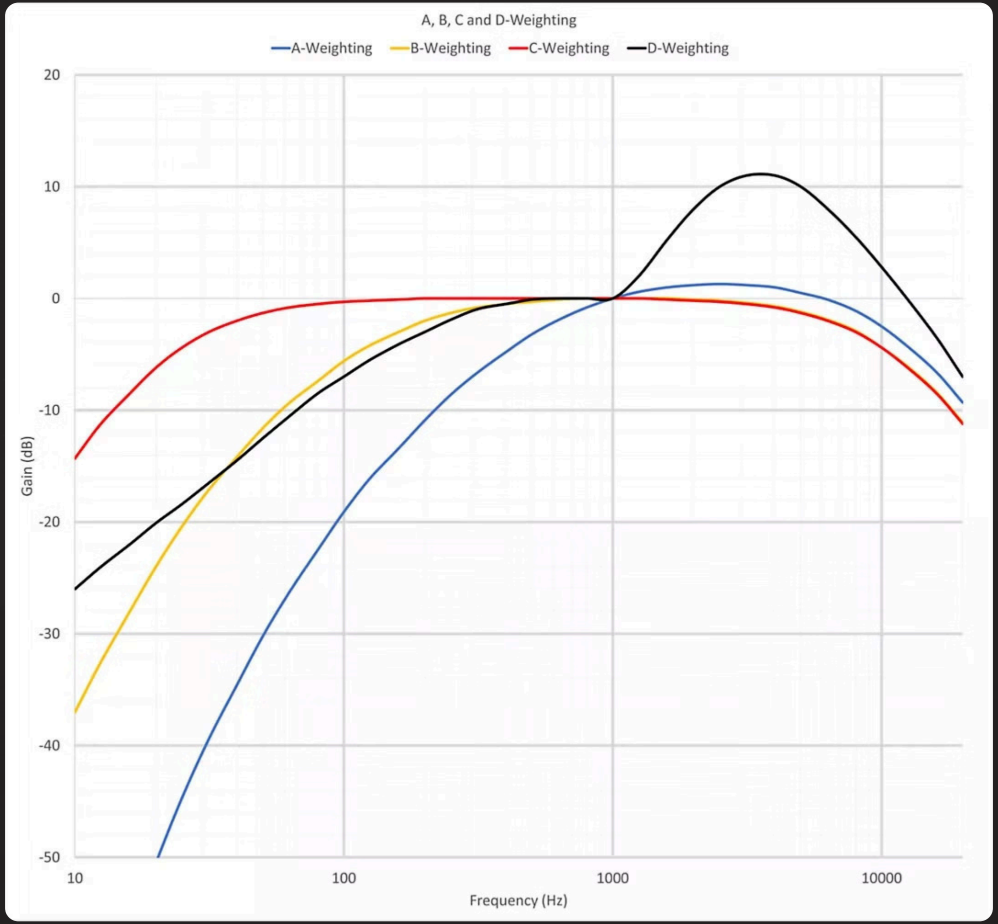
### Loudness Measurement

Loudness is now measured in dBA units.

# Intensity over different frequencies




Fletcher-Munson Curves




Weighting Curves

Made with Gamma

# Calculating Loudness

If signal is stereo, convert it to mono by averaging the signal values over all channels

1. Given start and end times of a word, extract segments (word sections in time).
2. A-weighting the segment's frequency content.
3. Finding RMS value of the segment. This is the loudness of the segment.
4. Order the loudness of different segments in descending order.
5. Set a cutoff based on the signal based on the properties of the segment. (We've used mean and standard deviation)
6. Signals louder than the cutoff have been categorised as "Loud" and vice-versa.

| Start Time | End Time | Loudness | Is it Loud |
|---|---|---|---|
| 0.6469 | 1.0036 | 0.1490 | 1.0000 |
| 2.7893 | 3.3600 | 0.1238 | 1.0000 |
| 1.5423 | 2.0883 | 0.1196 | 1.0000 |
| 1.4266 | 1.5423 | 0.0714 | 0 |
| 0.4526 | 0.6469 | 0.0574 | 0 |
| 2.0883 | 2.5409 | 0.0482 | 0 |
| 1.3406 | 1.4266 | 0.0357 | 0 |
| 1.0036 | 1.3406 | 0.0357 | 0 |
| 2.5409 | 2.7893 | 0.0170 | 0 |

Our readings for file 3

# Speech Recognition (Word Identification)

## Make segments

Choose a segment length. We've chosen segments of length 187.

Pad 186 zeros at the end of each audio file to allow partial overlap.

## Energy per segment

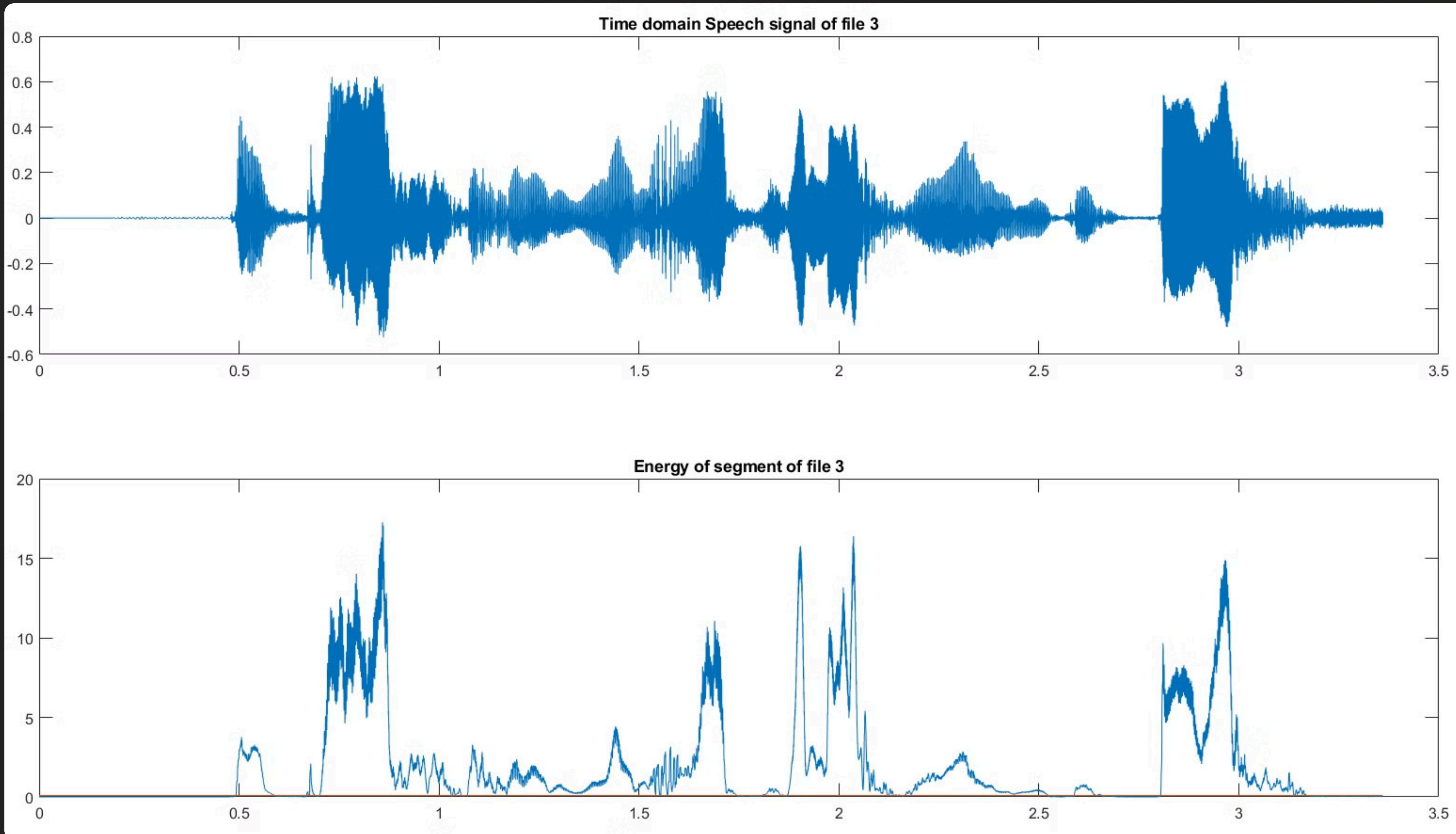
It is the sum of square amplitudes in the segment.

Energy per segment allows us to calculate the start and end time of  a word

## Compare with cutoff

If the segment's energy is greater than the cutoff, it is to be considered as a word.

I've chosen cutoff as a function of average segment energy




Observe the relation between the start and end times of the speech signal and the energy of each segment

# Algorithm analysis

Parameters (Minimum duration, Minimum gap and threshold) have been used

```
% Logic to segregate noise and speech (segmentation)
for ix = 2:N
    if (seg_energy(ix) > threshold && seg_energy(ix-1) <= threshold && abs(ix/fs - endtimes(length(endtimes)))>mingap)
        if (length(starttimes)==length(endtimes))
            starttimes = [starttimes; ix/fs];
            disp(num2str(ix/fs));
        end
    elseif (seg_energy(ix) < threshold && seg_energy(ix-1) >= threshold && abs(ix/fs - endtimes(length(endtimes)))>mingap)
        if (ix/fs - starttimes(length(starttimes)) > mindur)
            if (length(endtimes)+1==length(starttimes))
                endtimes = [endtimes; ix/fs];
            end
        end
    end
end
```

- Minimum duration (mindur): Shortest duration of all words.
- Minimum gap (mingap): Shortest silence between two words. Equivalently, it must be larger than the gaps between syllables in any word.
- Threshold: Minimum energy of a segment for it to be considered a word.

Using our algorithm, we were able to approximately determine the start and end times of the words.

File 1

| Start Time | End Time | Loudness | Is it Loud |
|---|---|---|---|
| 0.4751 | 1.0932 | 0.1778 | 1.0000 |
| 1.2950 | 1.5513 | 0.0771 | 0 |
| 1.0932 | 1.2950 | 0.0444 | 0 |
| 1.5513 | 1.9720 | 0.0249 | 0 |

**Expected**

File 1

| | | | |
|---|---|---|---|
| 0.4837 | 0.9314 | 0.2295 | 1.0000 |
| 1.3176 | 1.6378 | 0.0924 | 0 |
| 0.9321 | 1.3172 | 0.0483 | 0 |
| 1.6385 | 1.8840 | 0.0249 | 0 |

**Obtained**

# Modulation Analysis and Conclusions

### Modulation Analysis

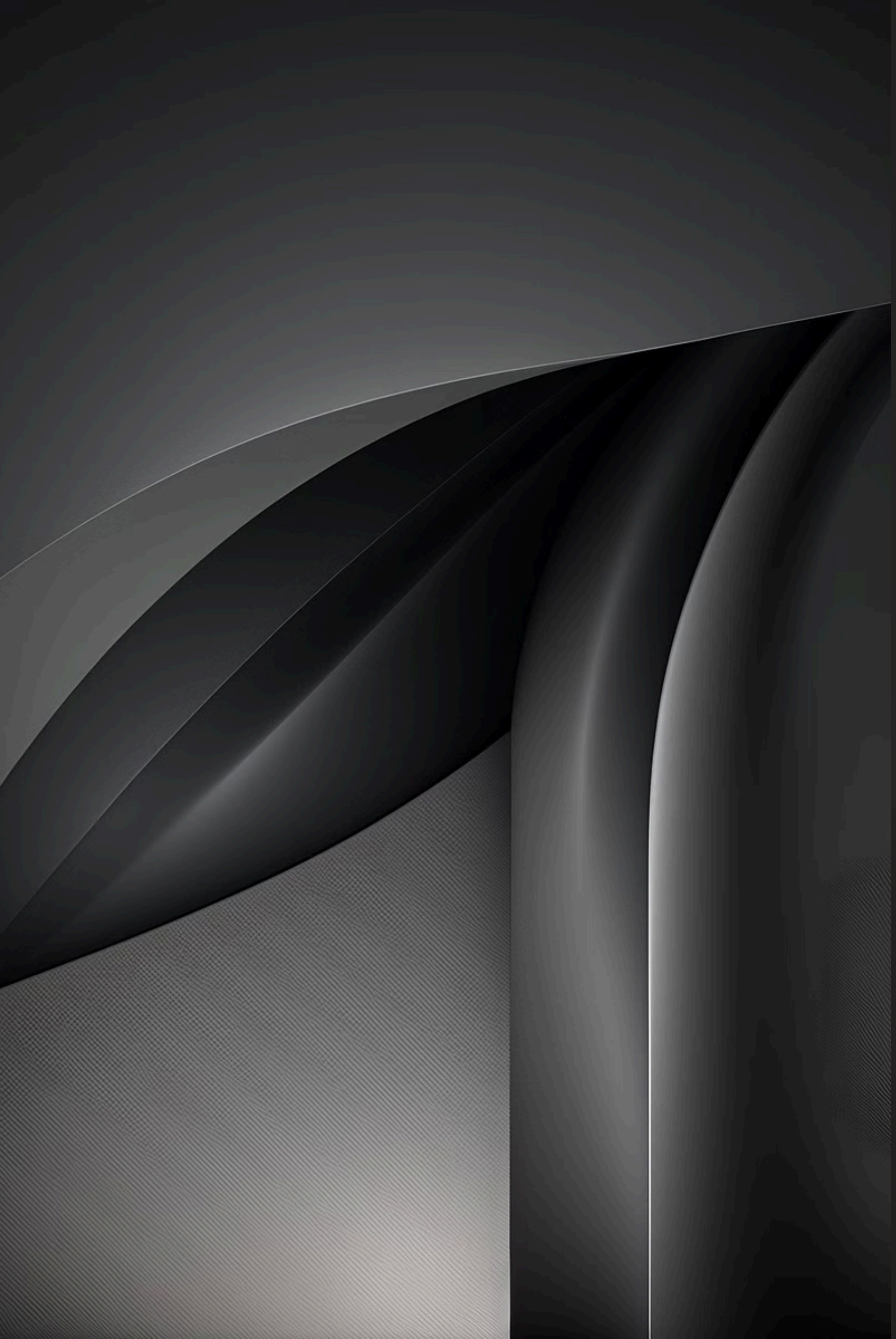
We investigated the modulation characteristics of the audio signal to understand the underlying patterns of loudness variations.

### Key Findings

Our analysis revealed that the audio signal exhibited significant modulation at specific frequencies, indicating changes in loudness over time and frequency.

### Conclusions

The loudness segmentation algorithm effectively separated the speech signal into distinct segments (words) based on variations in loudness.



# Closing Remarks and Next Steps

This project provided a practical application of signal processing concepts. We successfully implemented various techniques to analyze and extract meaningful information from different types of signals. Future work could include exploring more advanced signal processing techniques and expanding the scope of our analysis to encompass additional signal types.