Project Part-2

1 Noise Reduction

Noise reduction we make 2 different methods for noise reduction.

1.1 SOI

For our method in finding signal of interest we send a specific frequency which is 15kHz signal in transmitter part. In receiver part we design a filter for specific frequency given in figure 1. In that method we obtained easily start and end of the signal. Other parts of the signal are neglected. Therefore, we reduce the noise.

In our approach to identifying the signal of interest, a specific frequency of 15 kHz is transmitted from the transmitter section at the beginning and the at the end of the signal. At the receiver end, a filter is designed to isolate this specific frequency, which can be seen in Figure 1. Difference between input and output of this filter are shown in figure 2. Blue signal is our total signal, orange signal is filtered signal.

Later we need to find the start index and end index of filtered signal. We simply took absolute of filtered signal and smooth it with moving average filter. Last of all we determine a threshold according to maximum level of signal and mean of signal. We obtain the first and last index which amplitude is higher that threshold. Output of this method can be seen in figure 3.

This method effectively determines the start and end points of the signal while disregarding other portions of the received signal. Consequently, this approach significantly reduces noise interference.

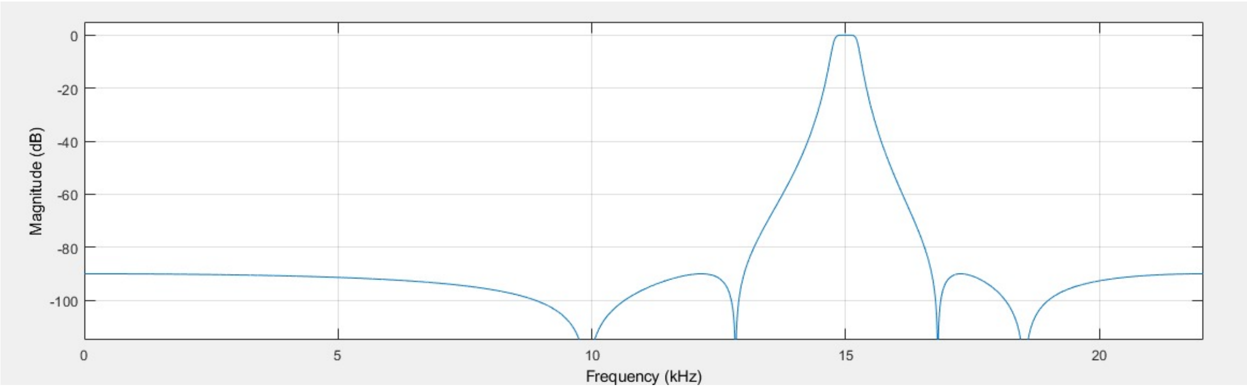


Figure 1

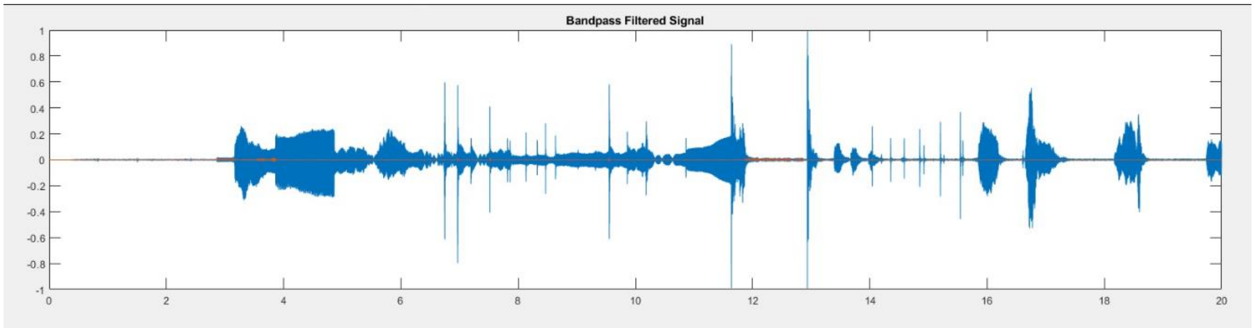


Figure 2

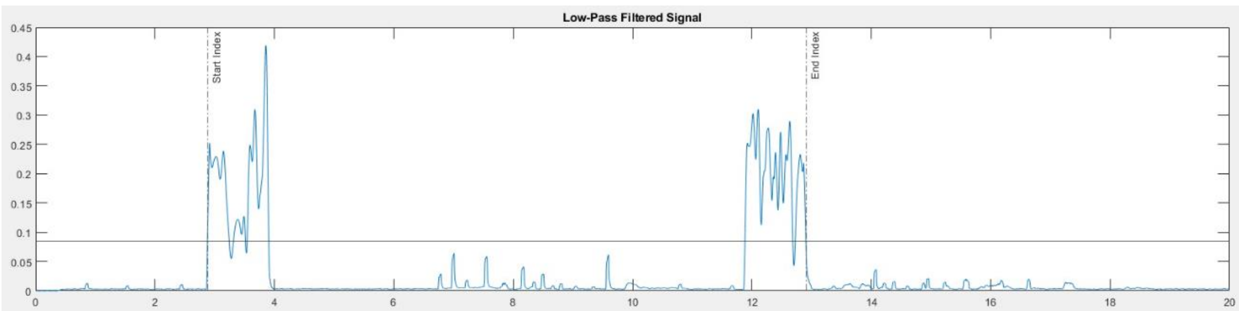


Figure 3

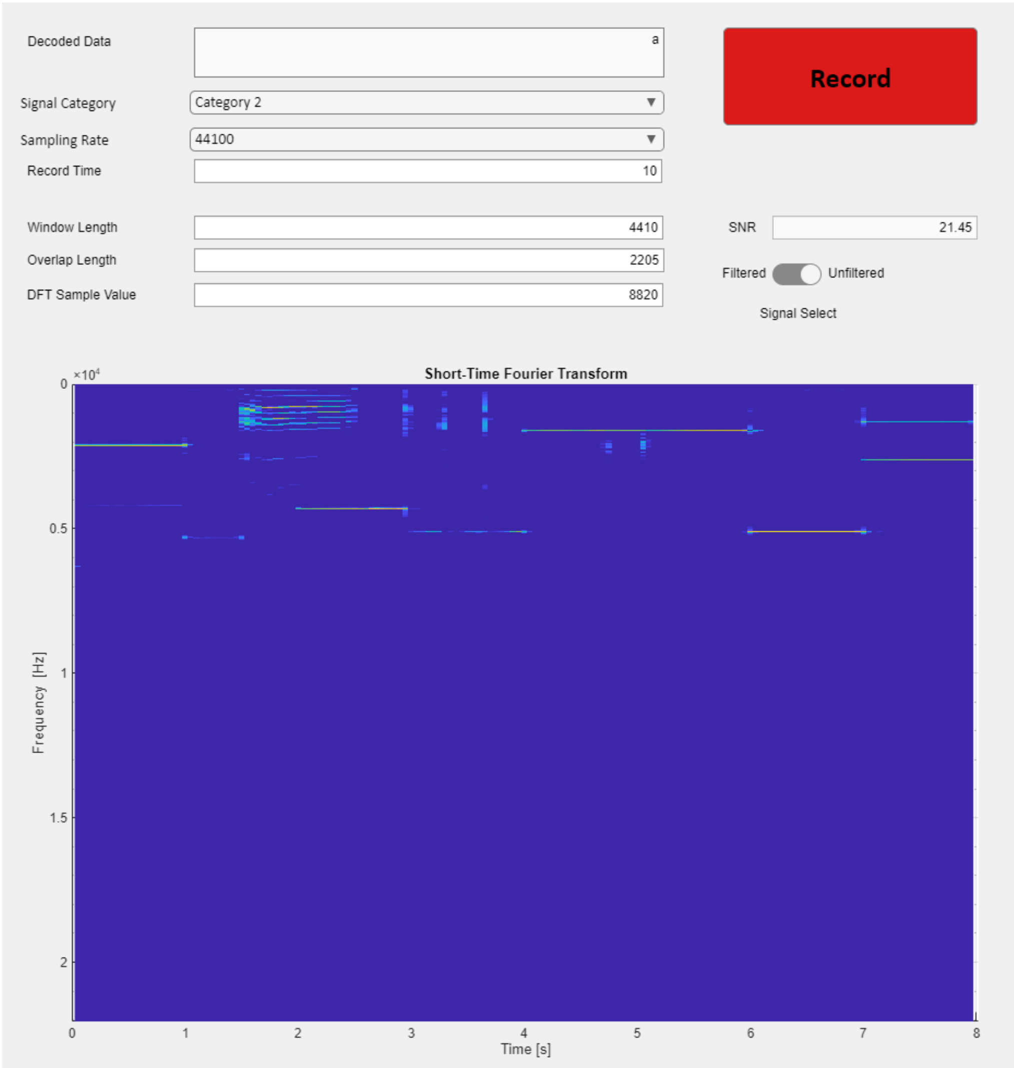
1.2 Dynamic Bandpass Filtering

After we obtain SOI. We obtain the length of the signal using the sampling rate and size of the signal array. We use this signal length to generate random numbers using RNG functions which are the same as transmitter side generated random numbers. Therefore, we can obtain expected frequencies. We use those frequencies and filter 150Hz much and 150Hz less. In that way we reduce the noise in frequencies our receiver does not expect.

After determining the SOI, the length of the signal is determined using the sampling rate and the size of the signal array. This method is implemented at the "findSignalLength" function. This signal length is used to generate random numbers using RNG. In that way we are ensuring consistency with the random numbers generated on the transmitter side because both rng functions have the same seed value. This method is implemented at the "selectFreqsfromTable2" function.

Using the generated random numbers, the expected frequencies are derived. These frequencies are then used to design bandpass filters with a tolerance of ±150 Hz around each expected frequency. This filtering process effectively attenuates noise in frequency bands outside the range of expected frequencies, thereby enhancing signal quality and reducing interference.

In figure 4 we see that unfiltered signal. In an unfiltered signal spectrogram we can see that there is a lot of noise between 1 to 4 seconds of SOI. But in the filtered signal, which is demonstrated in figure 5, we can see that those noises are filtered.



Decoded Data
a
Record
Signal Category
Category 2
Sampling Rate
44100
Record Time
10
Window Length
4410
Overlap Length
2205
DFT Sample Value
8820
SNR
21.45
Filtered
Unfiltered
Signal Select
Short-Time Fourier Transform
×10^4
Frequency [Hz]
Time [s]

Figure 4

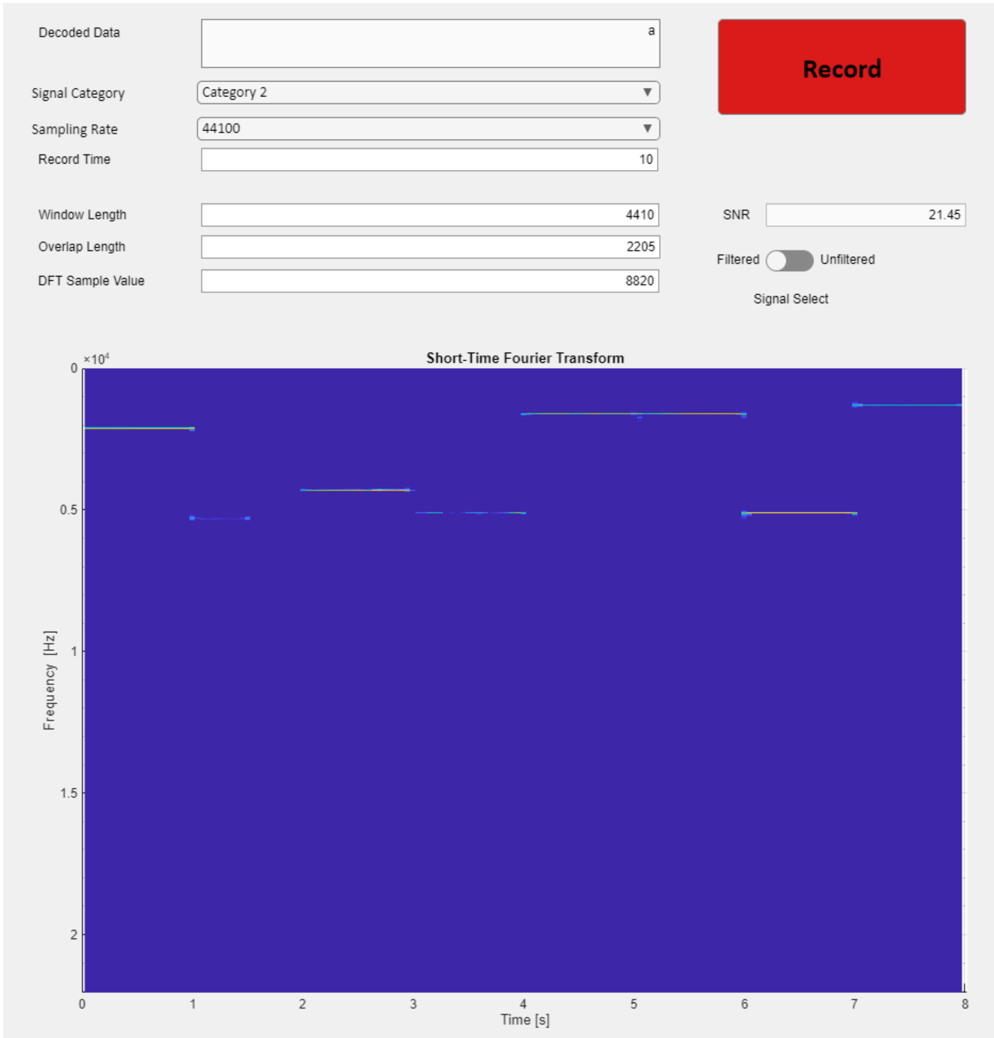


Figure 5

2 Tracking the Hop Frequencies

For category 1 signals it is known that there is one hop-frequency in each 1 second window. Knowing these and using the sampling rate, length of a signal can be found which corresponds to how many hop frequencies does the signal contain. Finding what these frequencies are is done by the “find_freqs()” function which is explained in the next section.

3 Demodulating Received Signal

After the filtered SOI is obtained its frequency spectrum is obtained. In this spectrum, for each hop period, the frequency that has the maximum amplitude is recorded. Since the recorded data is filtered with a bandpass filter centered around the expected frequencies, now the frequencies that have the maximum amplitude are the modulated frequencies from the transmitter. This functionality is implemented in the "find_freqs()" function.

(For Part I. find_freqs() is only implemented for Category 1 signals. But it can also easily be used for other categories if the parameter interval_length is changed for another category e.g. 0.5 for Category 2)

Now that each modulated frequency is obtained for each hop period, the difference between these obtained frequencies and the expected hop frequencies (obtained from the RNG of receiver side) gives the product: Δf * mk.

After this is done, the signal can be decoded by using "demodulate()" and "binarytoText()" functions.

4 Demonstration

4.1 Category 1

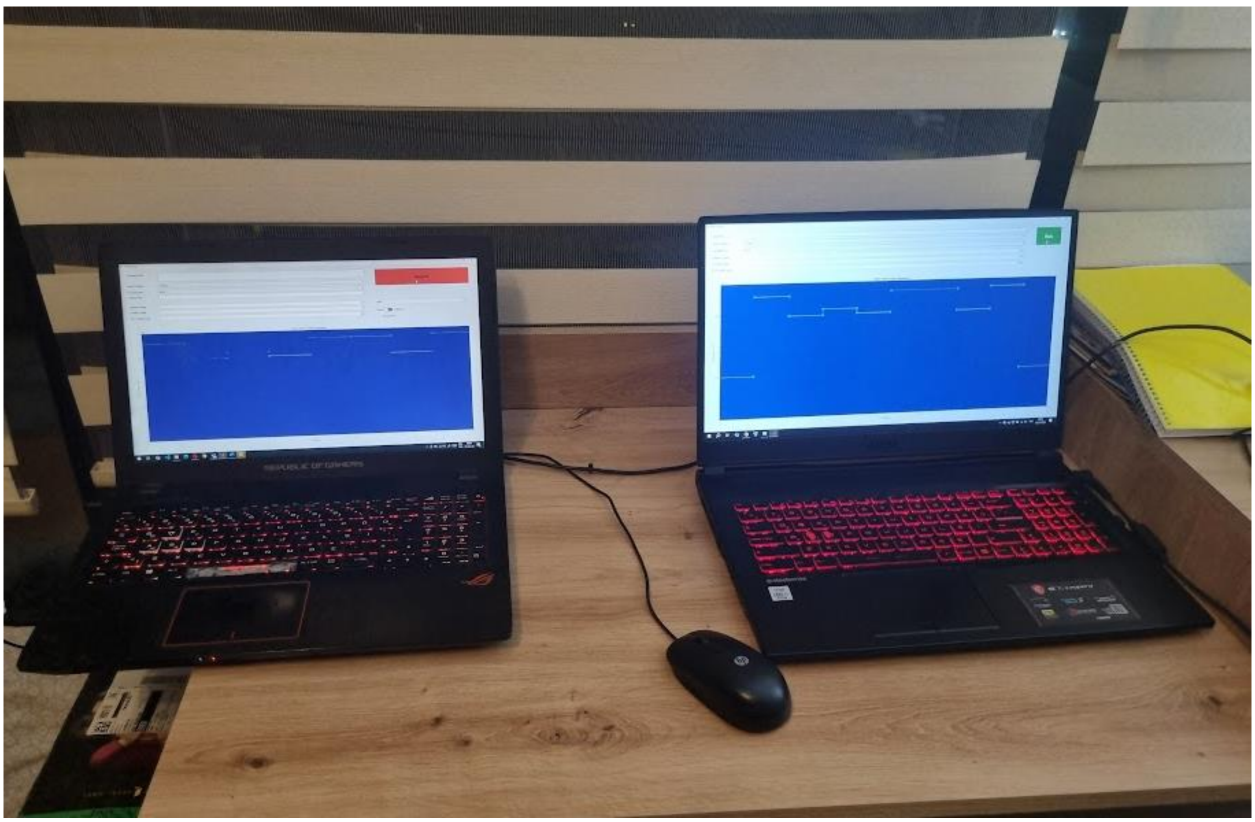


Figure 6

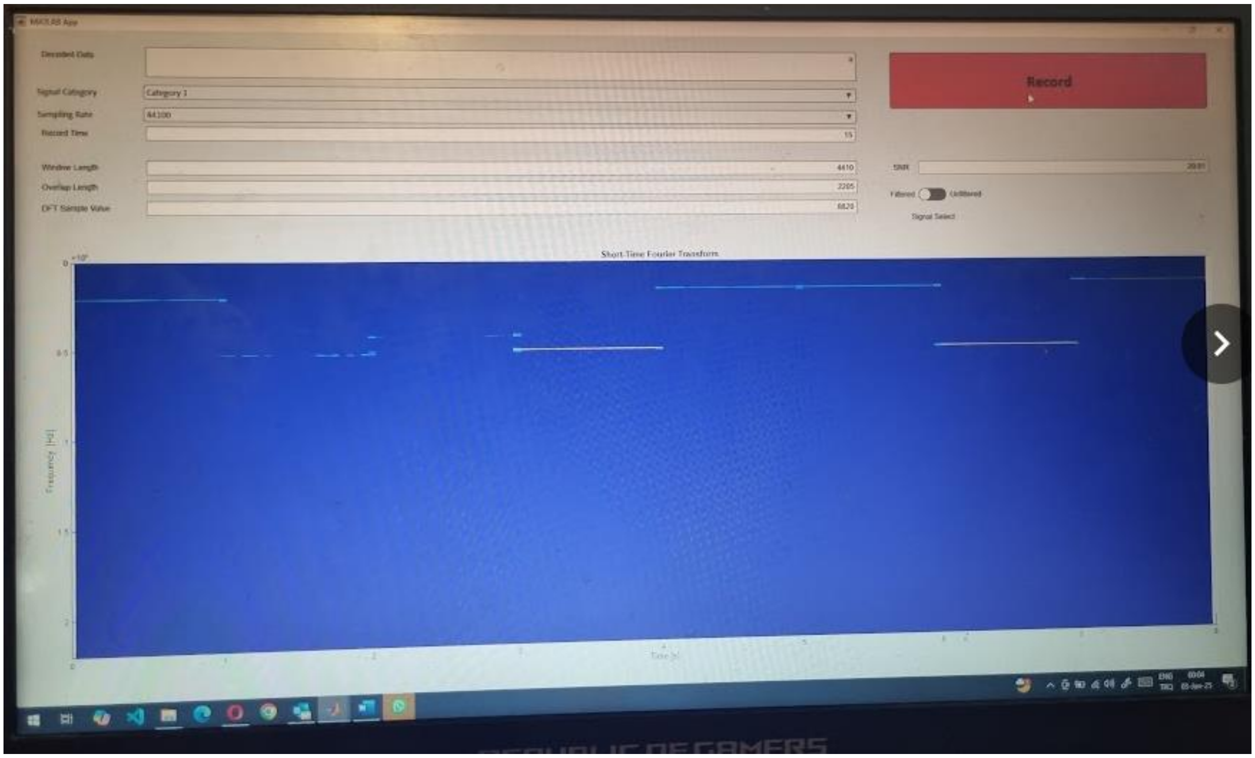


Figure 7

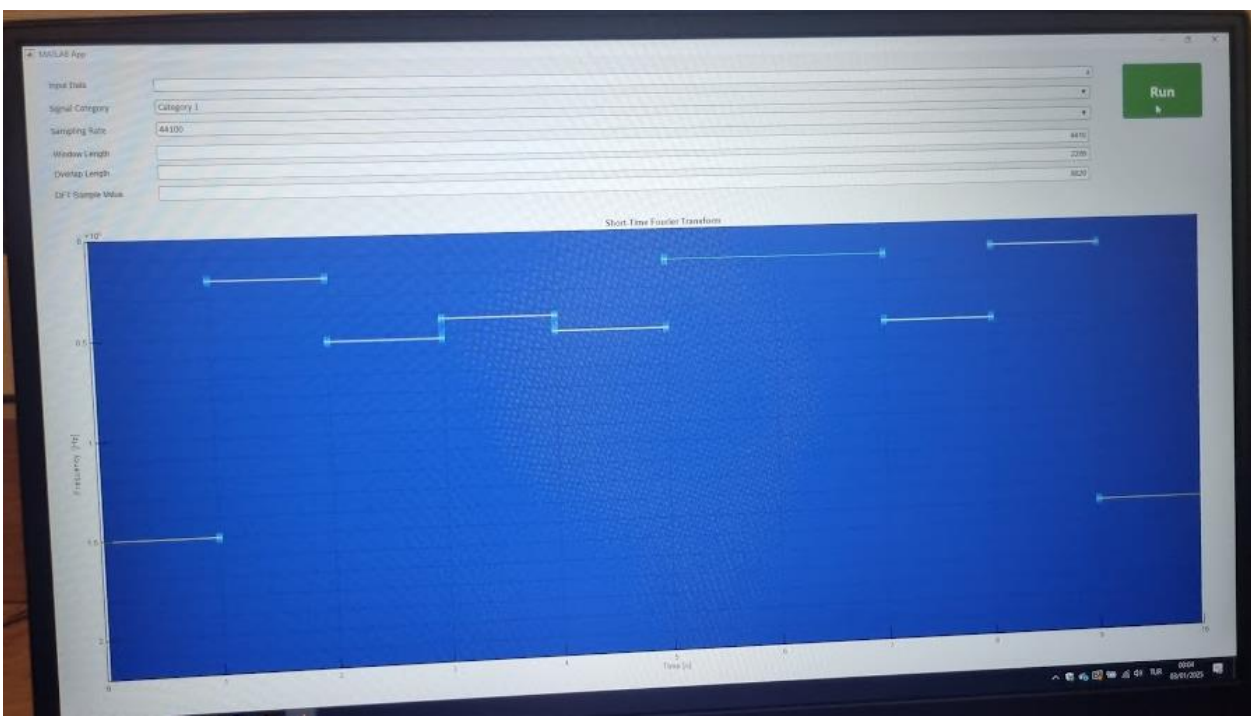


Figure 8

4.2 Category 2

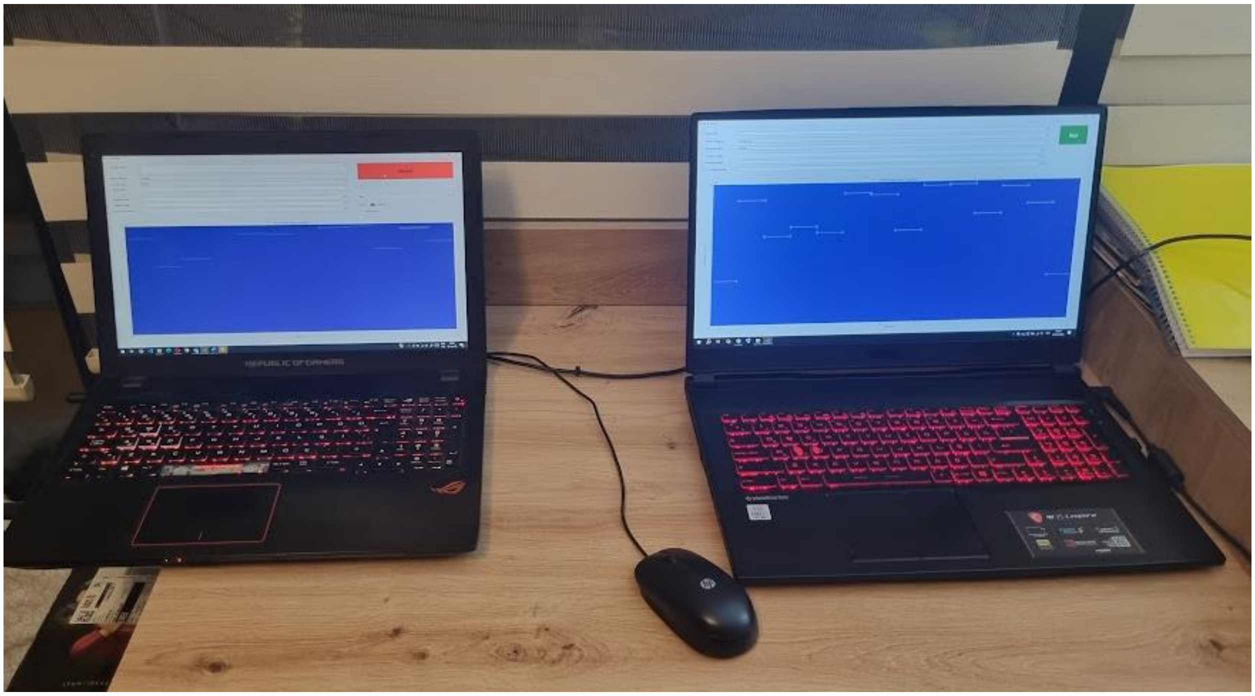


Figure 9

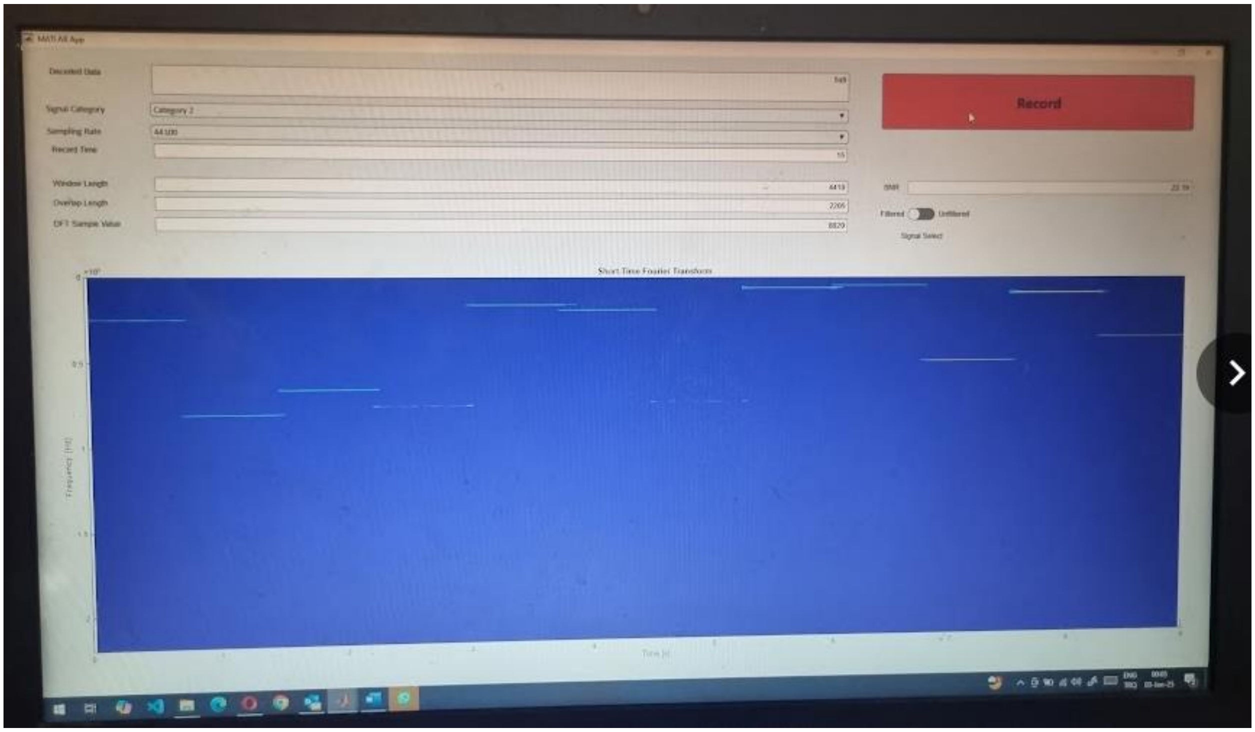


Figure 10

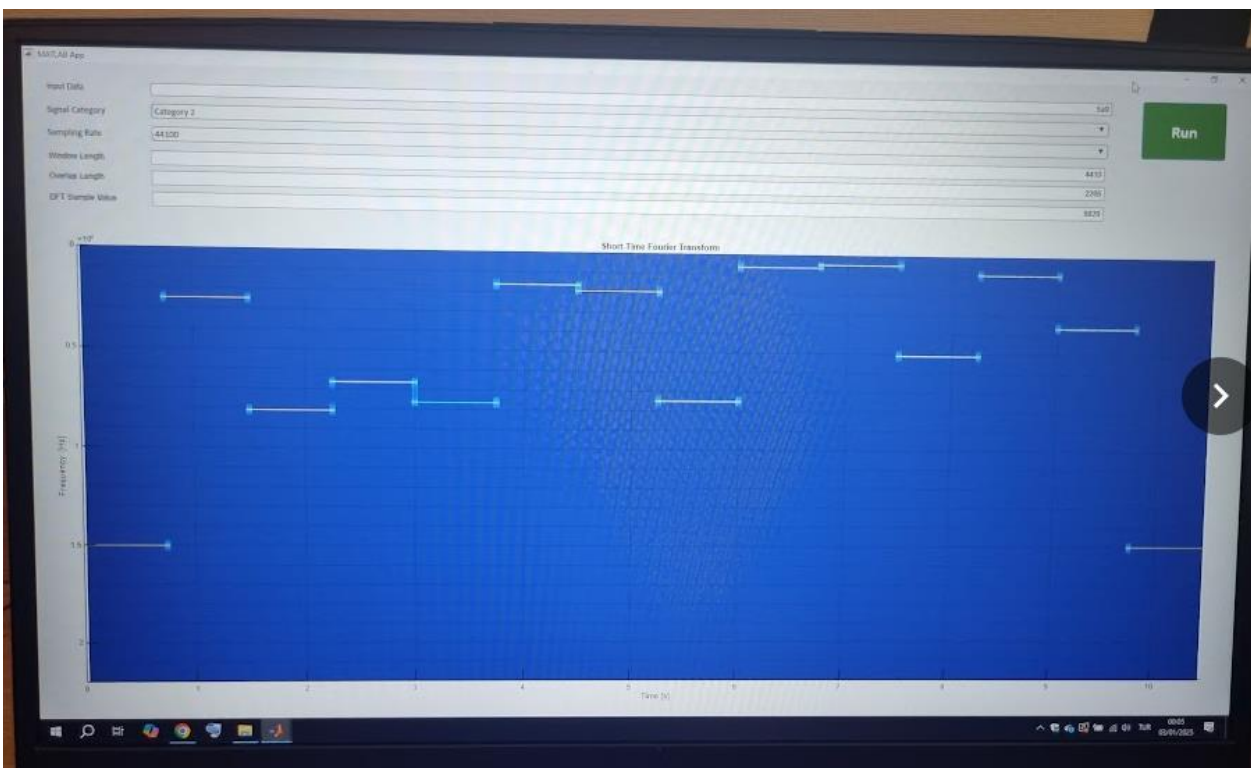


Figure 11

4.3 CATEGORY 3

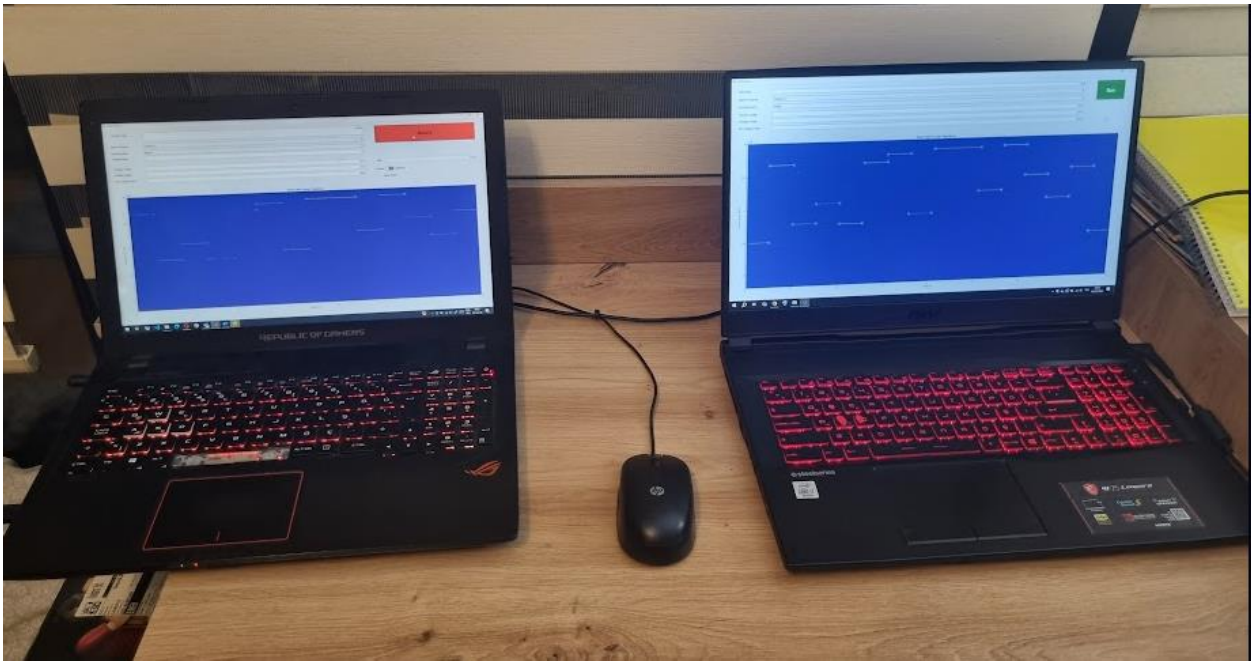


Figure 12

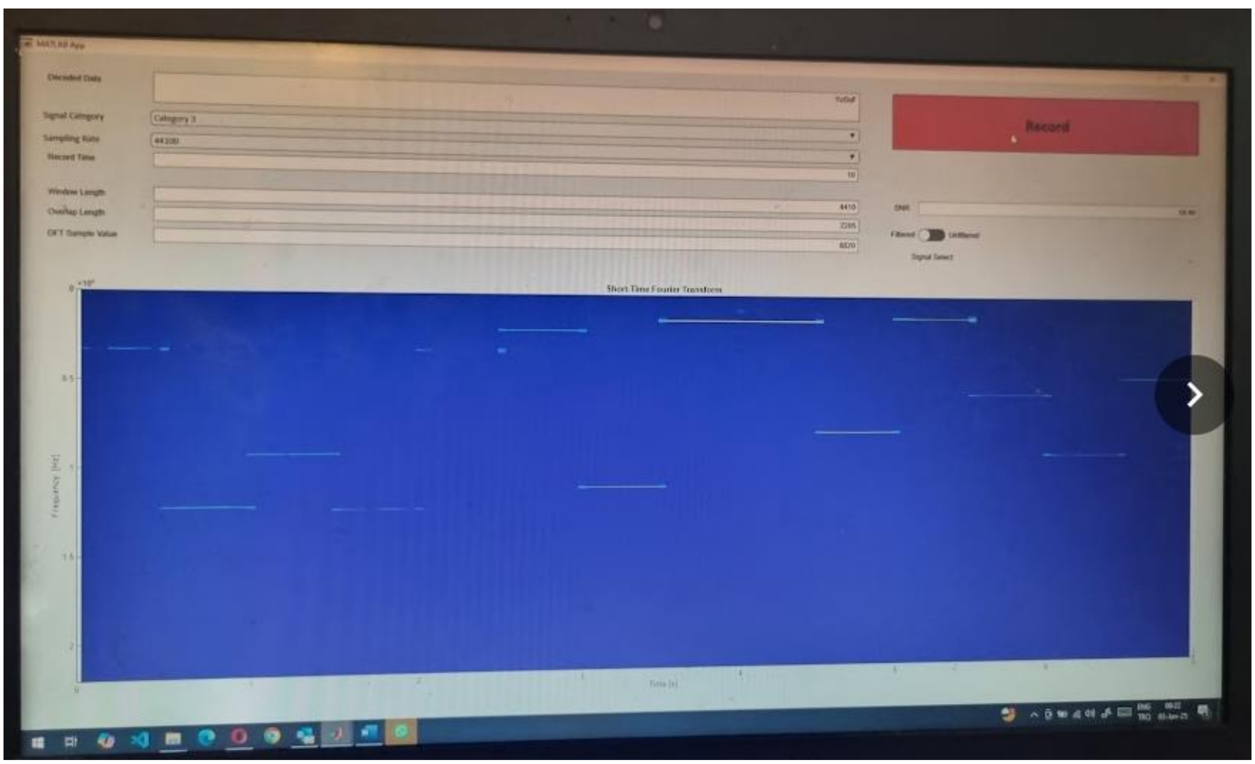


Figure 13

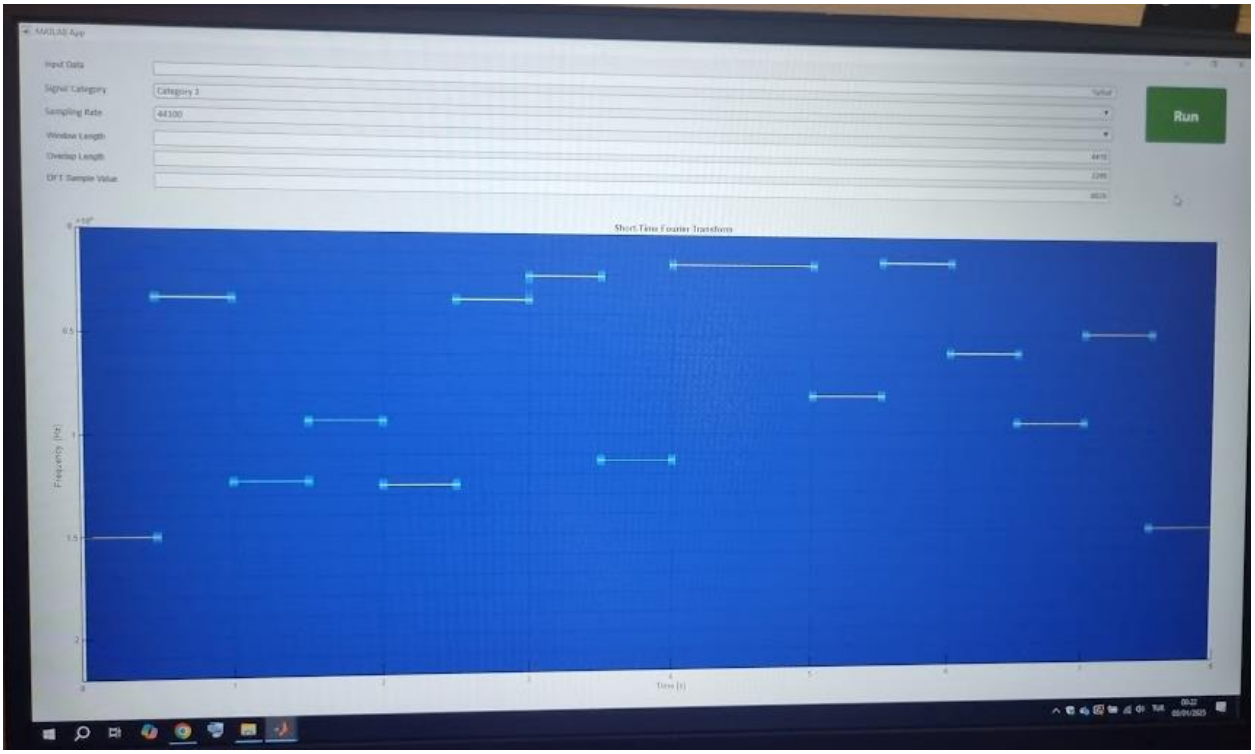

MATLAB App
Input Data
Signal Category
Category 2
Sampling Rate
44100
Window Length
Overlap Length
DFT Sample Value
Run
Short-Time Fourier Transform
Frequency (Hz)
Time (s)

Figure 14