4C1: Integrated System Design

Practical 3: FIR Filter Implementation

Rahul Seth

17302557

Introduction

This reports outlines my approach to the problem given with the following objectives:

1. Remove noise from the given speech file using a FIR Filter,
2. Design a Full Precision FIR filter,
3. Design another Filter with quantisation,
4. Comparison and cost analysis between the two filters.

I was assigned the audio file “Speech_26.wav”. I started off by analysing the original audio signal by plotting it against the duration of the signal. Then to figure out the pure tone noise component the amplitude of the original audio signal was plotted against time. The parameters needed for the design of the FIR filter were figured out using these plots and other functions available in MATLAB. Then, the filter was designed using the fda tool and this filter was using in the main script to filter out the original signal. The filtered signal was plotted again and comparisons were done which are outlines in the sections below.

Design and implementation

To start off this practical I first analysed my speech file (26) to figure out the things like where the noise tone is, what is frequency of it etc. in order to figure out the extent of the noise in the signal and also to designed my FIR filter.

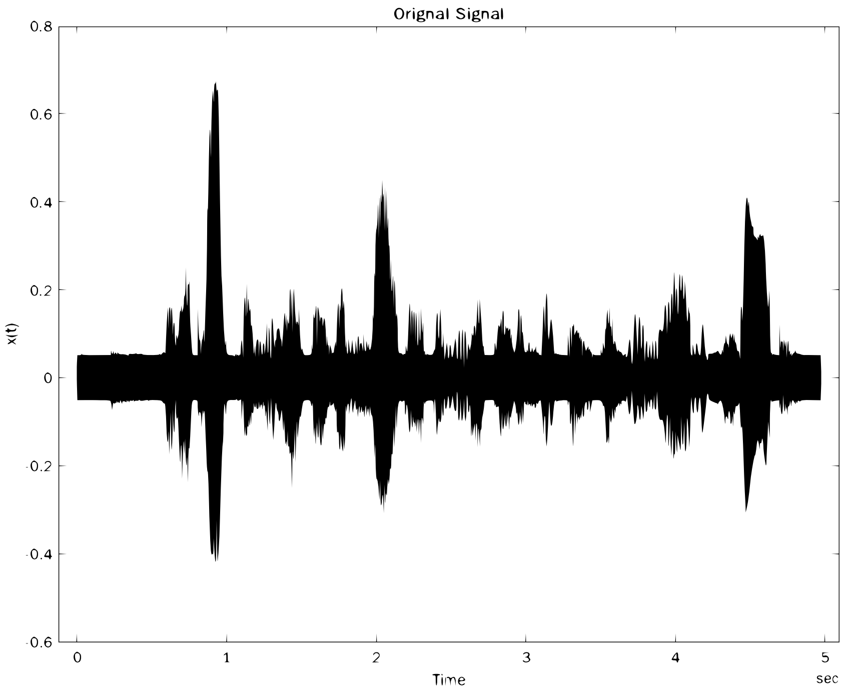


Fig 1: Original Signal vs Time

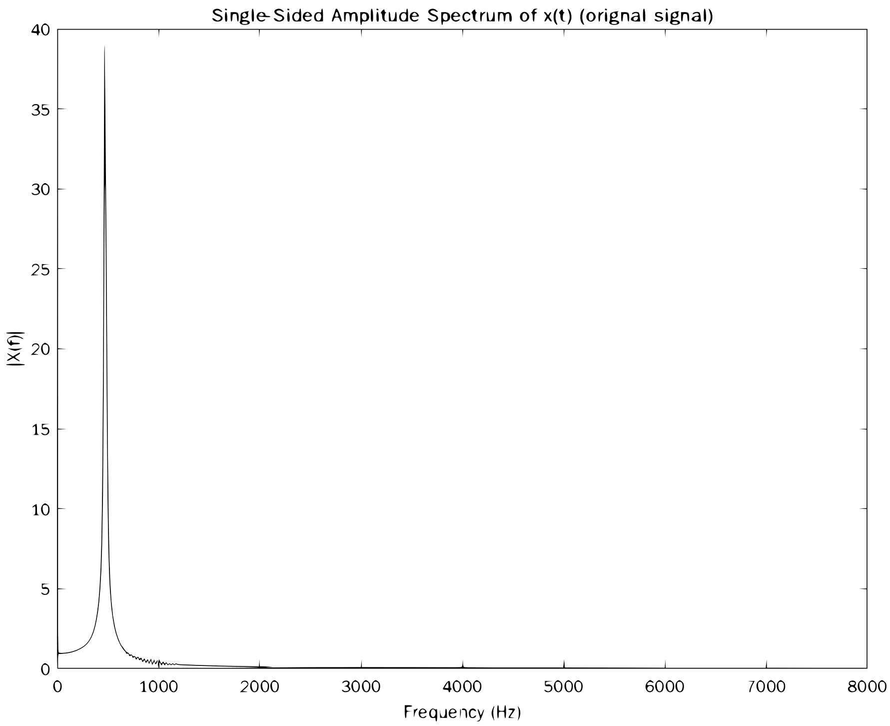


Fig 2: Frequency Content of Speech File

From the figure above we define our frequencies:

- Stopband: 218.75 Hz
- Passband: 1200 Hz

The above parameters were chosen because there is noise in this region that the filter should filter out.

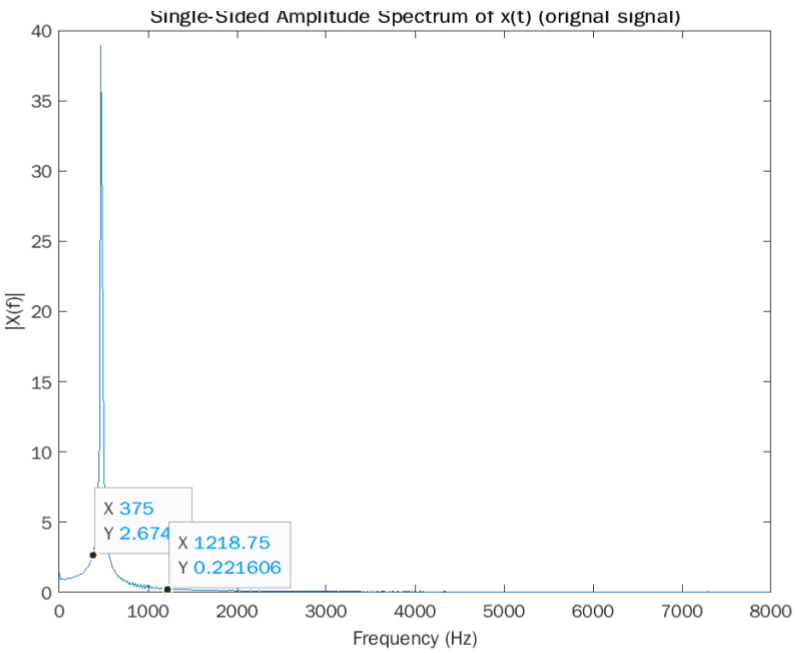


Fig 3: Chosen Frequencies for Filter

Filter Design

After analysis, I decided to design as full precision FIR high pass filter using the tutorial provided to us. I designed the filter using the fda tool available in MATLAB with the following specifications:

- Sampling frequency: 15000 Hz
- Stopband frequency: 380 Hz
- Passband frequency: 1200 Hz
- Stopband attenuation: 1 db
- Passband attenuation: 80 db

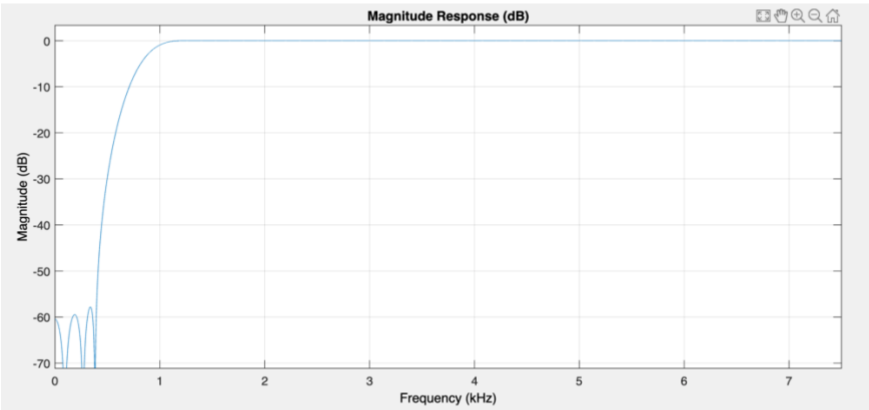


Fig 4: Magnitude Response of the Filter #1

Once, the filter was designed, the MATLAB code for it was generated and output script was called in the main script where we do our analysis. Our input signal was passed through the filter and the following graphs were generated:

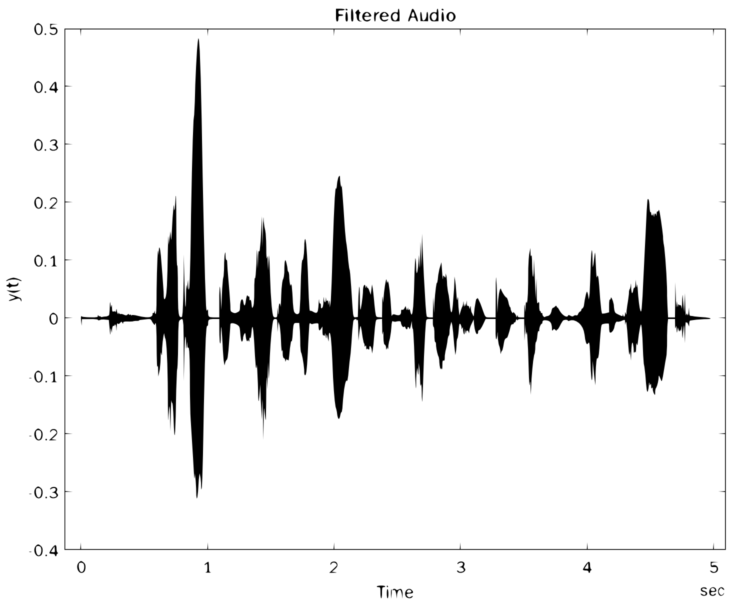


Fig 5: Filtered Signal vs Time

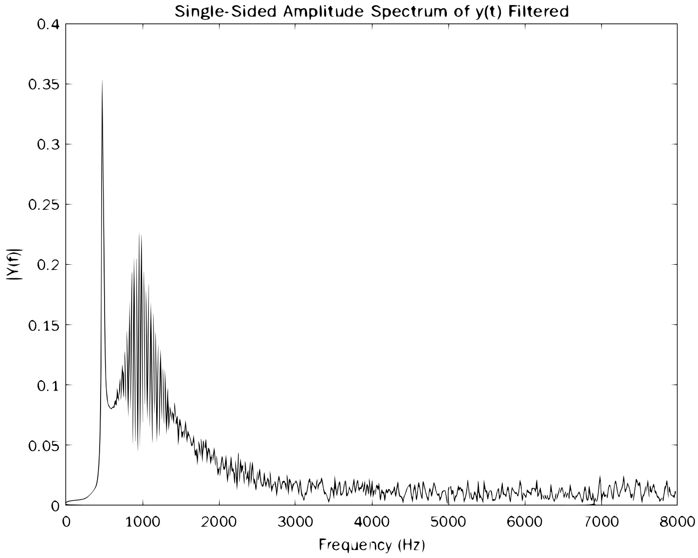


Fig 6: Frequency Content of Filtered Speech File

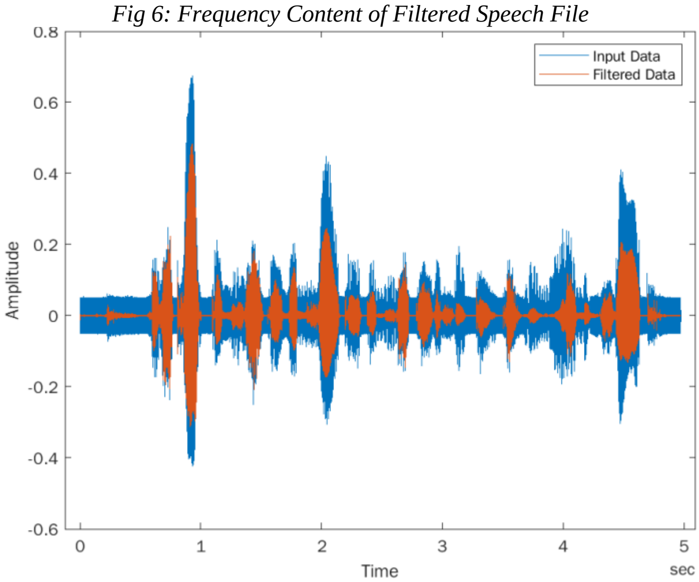
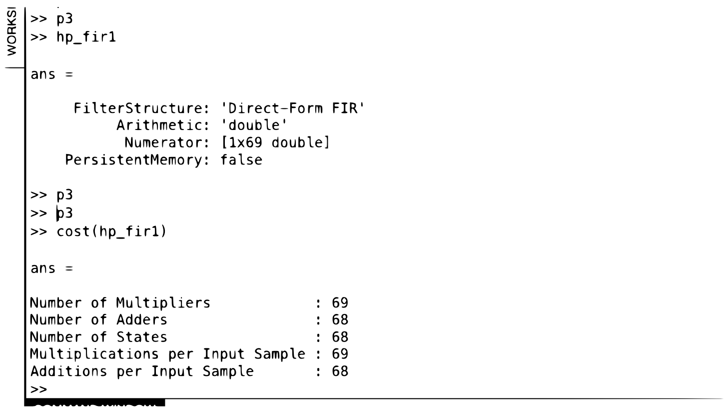


Fig 7: Comparison of Original Signal vs Filtered Signal

After filtering the speech file, the noise was certainly highly reduced. There was a buzzing tone that was observed initially which was reduced. Graphs were produced to verify this and Fig 4 and 5 indicate this. Fig 7 shows the comparison between the two signals and it is clear that there was significant noise reduction. The cost of the filter was calculated using the cost function available in MATLAB by passing the filter function through it. The output was:



```
>> p3
>> hp_fir1

ans =

     FilterStructure: 'Direct-Form FIR'
          Arithmetic: 'double'
           Numerator: [1x69 double]
    PersistentMemory: false

>> p3
>> p3
>> cost(hp_fir1)

ans =

Number of Multipliers            : 69
Number of Adders                 : 68
Number of States                 : 68
Multiplications per Input Sample : 69
Additions per Input Sample       : 68
>>
```

Fig 8: Cost Function output of Full Precision Filter

File name: "filtered_speech_26.wav"

Quantisation

As part of the practical we also had to quantise the filter to try and improve the function of the filter in clearing out the noise, compare the two in terms of performance and cost and justify the choice of filter selected.

The following specifications were used:

- Sampling frequency: 15000 Hz
- Stopband frequency: 380 Hz
- Passband frequency: 1200 Hz
- Stopband attenuation: 1 db
- Passband attenuation: 80 db

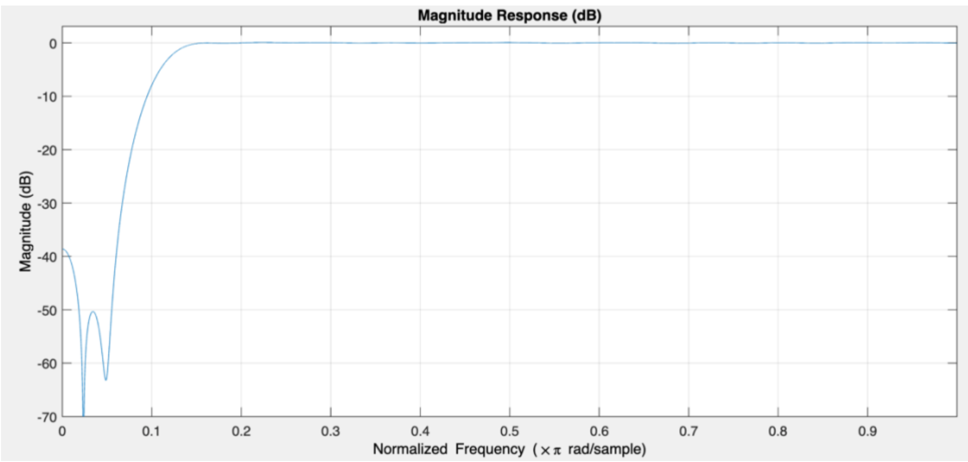


Fig 9: Magnitude Response of Filter #2

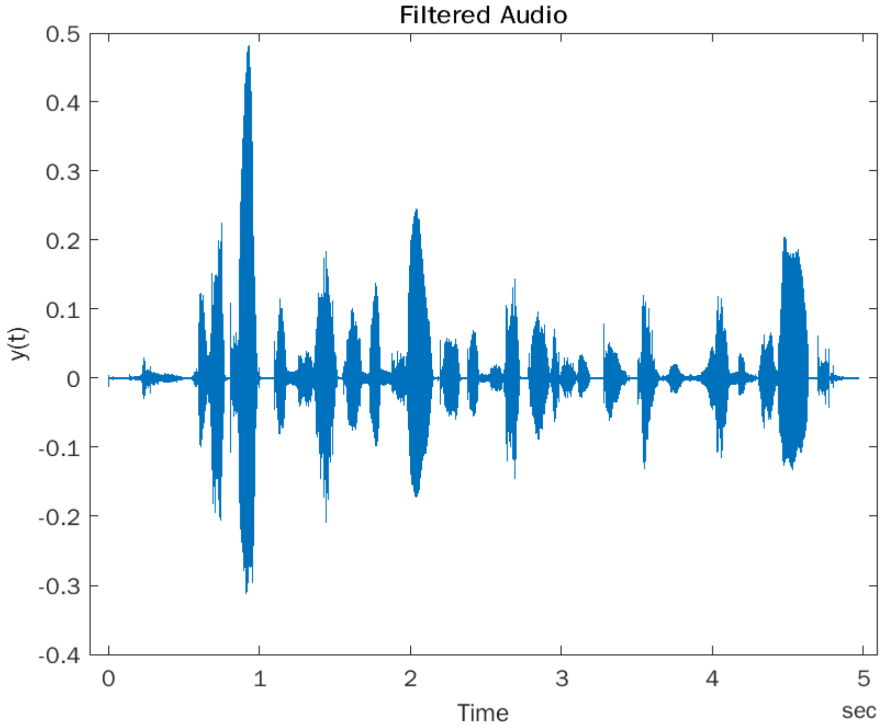


Fig 10: Filter signal (#2) vs time

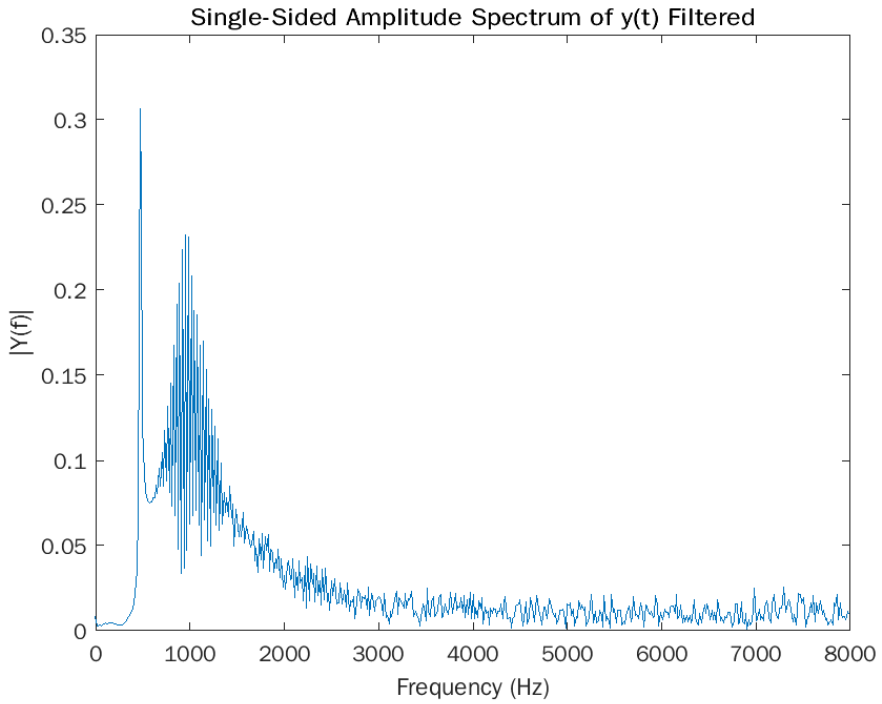


Fig 11: Frequency Content of Filtered (#2) Speech File

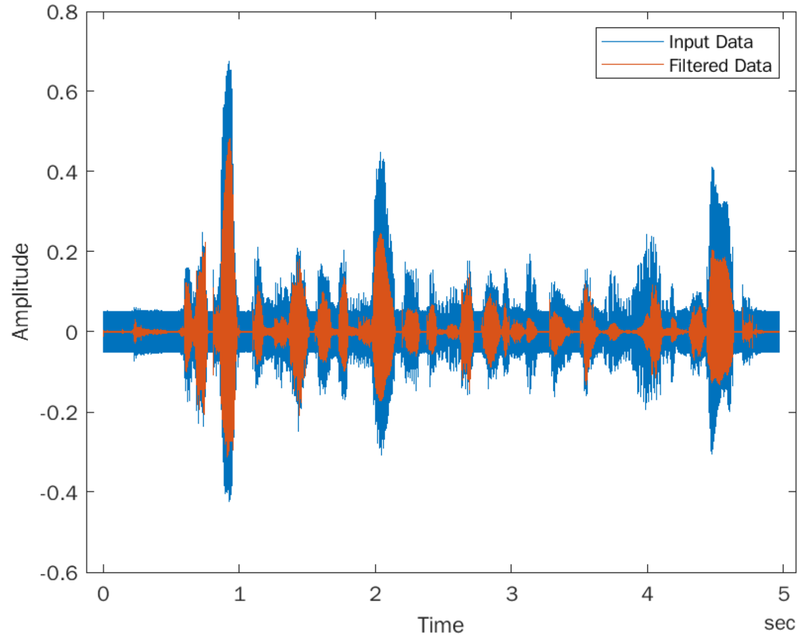


Fig 12: Comparison of Original Signal vs Filtered Signal (#2)

After multiple attempts at quantising it, I found the right parameters for the quantisation coefficients. It made the filter function better which was evident by further reduction of noise in the speech file. This was confirmed by hearing the audio file again and looking at the graphs above. Referring to Fig 11, we can see a reduction in the amplitude of noise as compared to the full precision filter (Fig 6). When comparing Fig 12 and Fig 7, we can see how noise has reduced using quantisation.

```
>> cost(hp_fir2)

ans =

Number of Multipliers            : 51
Number of Adders                 : 50
Number of States                 : 68
Multiplications per Input Sample : 51
Additions per Input Sample       : 50
>>
```
COMMAND WINDOW

Fig 13: Cost Function output of Quantised Precision Filter

File name: 'quant_filtered_speech_26.wav'

Conclusions

The following conclusions were made after analysing the original speech file and through the design process of the two filters:

- In this case, quantisation of filter helped reduced noise further.
- Frequency response of a full precision filter was quite different from quantised filter.
- Quantisation helped with reducing cost of the filter.
- If quantisation parameters are not chosen carefully, it can result in worse performance than the full precision filter.
- **Chosen Filter: Filter # 2 (Quantised Filter).**

****