Evaluation of various techniques, and execution times, of different filtering systems against audio files

Alexander O’Callaghan

Abstract

Acknowledgements

# Contents Page

Contents

[Contents Page 4](#_Toc40440884)

[1.0 Introduction 5](#_Toc40440885)

[1.1 Background 5](#_Toc40440886)

[1.2 Focus 5](#_Toc40440887)

[1.3 Value 5](#_Toc40440888)

[2.0 Literature review 6](#_Toc40440889)

[2.1 High Level Overview 6](#_Toc40440890)

[2.2 Filtering Techniques 6](#_Toc40440891)

[3.0 Methodology 7](#_Toc40440892)

[3.1 Introduction 7](#_Toc40440893)

[3.2 Code Breakdown 7](#_Toc40440894)

[4.0 Findings/Results 9](#_Toc40440895)

[5.0 Discussion 10](#_Toc40440896)

[5.1 Introduction 10](#_Toc40440897)

[6.0 Conclusion 11](#_Toc40440898)

[6.1 11](#_Toc40440899)

[6.2 Future work 11](#_Toc40440900)

[References 12](#_Toc40440901)

[Appendix 13](#_Toc40440902)

# 1.0 Introduction

## 1.1 Background

Digital signal processing (DSP) is a way of using computer code to make a series of operations on signal data with the intention of creating data that is more useful in some manner, to the user. This can range from <<CITE SOMETHING>>. The main advantage of using software to do this is that designs can be created very easily, cheaply, and more effectively than creating similar designs in hardware. <<compare hardware to software reference>>. This is because modifying design in software can be done very quickly and tested on the fly. Hardware components must be physically altered to change processing aspects and rely on each part working correctly. This introduces more points of failure to the system.

## 1.2 Focus

The focus of this project is to make comparisons between different filtering techniques that can be used to increase correlation between audio samples and their <<dirty>> counterparts. The <<Dirty>> samples will be filtered using different methods and the performance will be examined. While it may be difficult to directly compare the results should indicate certain advantages and disadvantages in certain aspects.

Research Questions

1. Which filtering technique can consistently perform the best removal of unwanted noise from an audio signal?
2. Which filtering technique is the quickest in its execution?
3. Does a filtering technique contain a good mixture of both functionality and speed?

## 1.3 Value

The scope of this project focuses on examples where audio files are used. Though that alone could provide useful, the scope for filtering techniques can extend to a large range of different signal mediums. The results found here can contribute to the wider field of Digital Signal Processing in which the benefits of which can have far reaching effects.

# 2.0 Literature review

## 2.1 High Level Overview

This dissertation is about the effectiveness and performance of different filtering types. While many different media can have filters applied to them, this report will be specifically looking at digital filters acting on audio signals.

To describe a filter on a signal, Steven W. Smith, (1997) puts it well in his book that filters primarily are used to either separate “…signals that have been combined, and (2) restoration of signals that have been distorted in some way.” (p261). To expand on that point, an example could made of a voice recording that contains background noise. A filter, designed to reduce the frequencies that a human voice does not contain, could be used to clear the signal and make hearing the voice easier. The author later goes on to make a comparison of analogue and digital and while “digital is slow; analogue is fast. For example, a personal computer can only filter data at about 10,000 samples per second, using FFT convolution. Even simple op amps can operate at 100 kHz to 1 MHz, 1 to 100 times as fast as the digital system!” (p344), the digital bests analogue in almost all other areas.

## 2.2 Filtering Techniques

Signal filters can either be analogue or digital and while both have their merits, digital versions are what this report will focus on.

Two “families” of digital filters exist, and they are finite impulse response (FIR) and infinite impulse response (IIR). Aiming for the same result they both differ in their application.

FIR filters use a method called “convolution” when applying their filtering to signals. This is fundamentally different to IIR filters as though they employ this method too, they also take the output of the expression for further functions. The two different style produce different characteristics for example, (Burgess, 2014) weighs up some of the advantages and disadvantages of the two, commenting that while FIR has a lower efficiency than that of IIR, it is much more stable. This is due to FIR having no feedback, not matter how it is implemented.

A lot of research has been with the implementation of different filters being used as a tool rather than a focus whereas this report will seek to directly compare them in like for like cases.

# 3.0 Methodology

## 

## 3.1 Introduction

#### Recap of questions

The following methodology was designed with the research questions mentioned earlier in mind. To recap this project aims to assign values of functionality, speed, and an evaluation based on both things.

#### Design

Matlab was the software chosen to perform all the development of the filtering because of its functionality to easily implement the filters and output the results in a clear and concise way. The speed of development is very similar to prototyping as the language’s syntax lends itself to quick implementation without class structure. The version of Matlab used was R2019b and was running on a laptop with <<insert specs>>.

<<source control>>

Five audio files were picked from https://www.bensound.com/royalty-free-music to be used as the media in which the filters will be applied to.

## 3.2 Code Breakdown

The work of the project was completed entirely in MatLab and the following is an overview of the code itself.

The original audio files were cut down to a length of 30 seconds to reduce computational times while in the prototyping phase. The files were then played through the computer’s speakers while a microphone was used to capture the sounds. Both the cut down, and recorded files, were imported into the program in a for loop and placed in separate matrixes. During the import, the two audio channels we combined into a singular value of left and right to make an average. The sampling frequencies were also changed to make both sets the same.

Next a loop ran through each recorded file and found the correlation of it against each of the original files. The method used found at which point the two files were most similar and compared the recorded file against the original at that point. On some files that did not match, this point was at a moment in time that did not allow for a full comparison. As this only happened on files that did not match, marking these comparisons as Null was acceptable and could be dismissed in the analysis.

The matrix was then converted into a table for easier viewing and the file names were added in the columns to further aid analysis. The resulting five by five grid is from then on considered the baseline and these figures could then be used when comparing against the filtered results.

#### Scenarios

With the baseline set, the recorded versions of the files were combined with other signals to create “scenarios” in which an audio signal could realistically exist in real world applications. The scenarios were adding a 1Khz sine wave, adding a 60Hz, and adding the sound of a vacuum cleaner. With each scenario, the filters would have to be set up with different parameters.

For each track that has had a signal added to it, a multitude of filtering techniques will be applied to it, with the output of each being correlated against the original file, along with the recorded file it originated from and its execution time.

Once complete, the output data will be imported into Microsoft Excel to take advantage of the programs ease of use and ability to apply conditional formatting.

For the filtering, the Matlab application “Filter Designer” was a useful resource in making the filters.

# 4.0 Findings/Results

### 4.1 Introduction

In this chapter we will look at some of the key result created from the experiments and examine any observations that can be reached at a glance. The deeper look into the results will be discussed in the next chapter and all of the data can be seen in appendix WHATEVER.

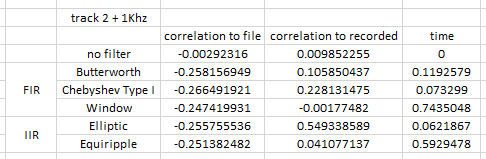


Figure 1. A typical table of results.

Above, figure 1, is the output of the experiments for one of the tests ran. It shows that the recorded version of track 2, with an added frequency of 1KHz, and the results from correlation with the original file, the recorded file, the time it took to perform the filter. To the left, we see that each row represents the type of filter, FIR and IIR, along with the different filter names. The no filter row was included as a reference point and because there was no process being completed, the time is zero.

After taking all the absolute values and averaging, figure 2 shows a graph of the results in a more presentable manner.

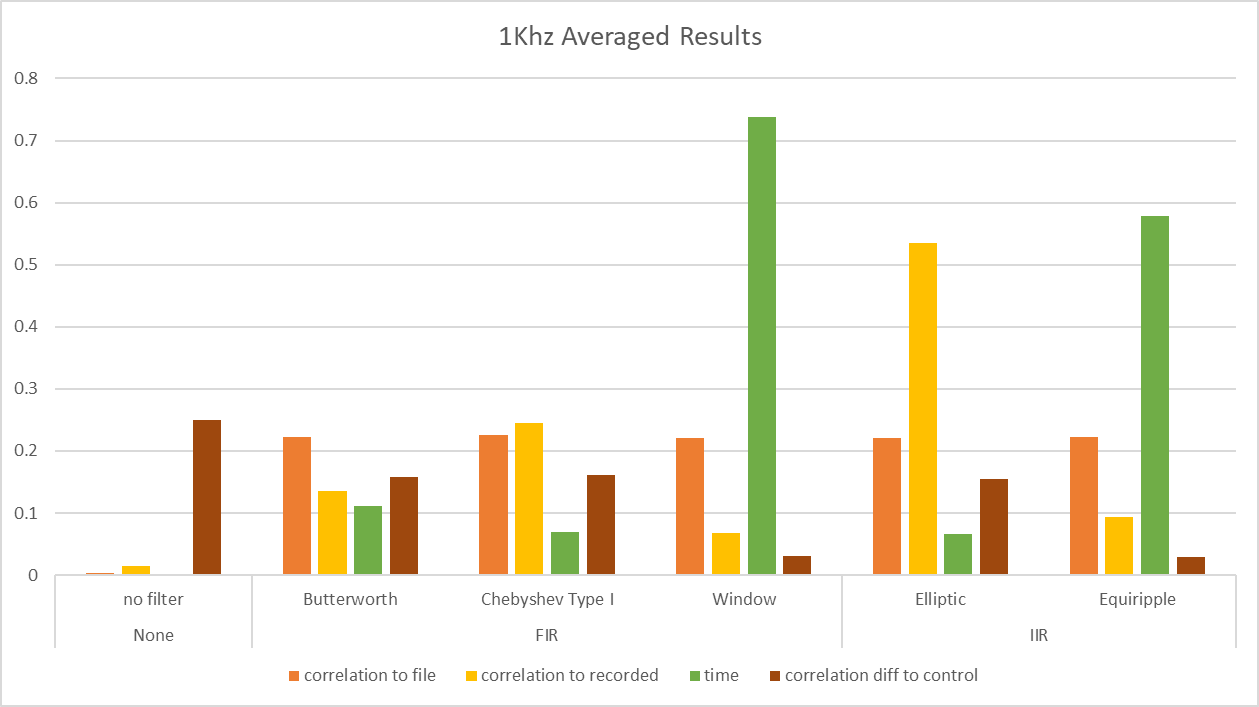


Figure 2.

The results are banded in either no filter applied, a FIR filer, or an IIR filter. We can see, on the whole, very little deviance between filters in correlation to the original file, which we would look for a higher value being a better correlation and therefore a more favourable result.

However, correlation to the recorded file is much more varied.

As acknowledged above, figure one is a typical results table from the experiment which share similar trends as most the other also do. Again, a higher value represents a higher correlation and is favourable. We can see that elliptical is at least double the correlation value compared to any of the other filter types.

The time taken for each filter has perhaps the biggest range with values starting below 0.1 up to heights of over 0.7. This section the favourable value would be the lower number, showing a quicker execution time. The window and equiripple techniques are shown to be many times higher than that of any of the others.

The last value represented on this graph is the correlation difference to the control which in this case is the value that the correlation between the original file and the recorded output. While it is not inherently true that a lower value is more favourable, as we already know that the first value in figure 2 (the correlation between the filtered results and the original file) does not vary much, a lower number does indicate a more favourable result because it shows that filter has been more successful at removing noise from the recorded file.

This chapter has been a brief look at how the results of the experiment can be displayed in a suitable manner. In the next chapter the results will be more closely investigated.

# 5.0 Discussion

## 

## 5.1 Introduction

For the project to use a “base line” of correlation from the signals that have had noise added, there was need to establish a control of the recorded tracks correlated against the original files. Figure 3 shows the results of this.

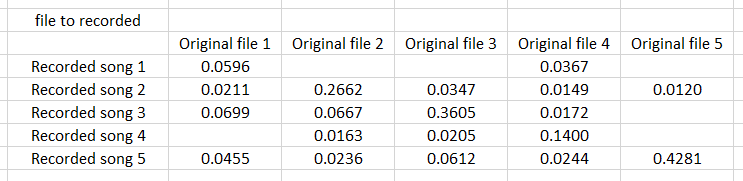


Figure 3. Cells without any values represent very poor correlation. All values are absolute values of results

The values of this table represent the correlation of each track based on a coefficient from 0 to 1, where 0 is least similar and 1 is most similar. In each case, the point at which a song is compared to its similar version, it is the highest scoring correlation coefficient in the column with the exception of the first column WHY THOUGH. This shows that the comparison method shown in APPENDOX WHATEVER works. While we cannot be sure to what extent, we can measure against this table to make comparisons later. Further demonstrations of this can be seen in APPENDIX WHATEBER where files are compared against themselves. In some cases, the highest scoring value is much higher than the others in the row/column, but in other cases it is only higher by a small amount. WHY THOUGH

With the data we can now aim to apply it to the original 3 research questions.

## 5.2 Which filtering technique resulted in the best removal of unwanted noise

In figure 2 we saw that the correlation from the filtered files were so similar that it hardly warranted discussion as to which did better. With that in mind, the next best metric became the correlation to the recorded file that the noisy signals came from. In that case it was the elliptical method that outperformed others, effectively doubling from the next highest scorer.

## 5.3 Which had the quickest execution times

For all the different execution times, the type of filter remained fairly consistent throughout the different applications of it.

One of the most apparent things in the results data was the high end of the execution times. Both the FIR window and IIR equiripple techniques consistently proved to be the longest to execute. At best they were 6 times the next longest timed filter and at the worst, up to 70!

## 5.4 Best overall

# 6.0 Conclusion

## 6.1

As is the case in much work, the accuracy of these results can be improved by increasing the number of samples tested against.

## 6.2 Future work

Future work and extensions of this project should look to change the type of data being used. A suitable example would be the medical field of study. Implementing filtering techniques to data such as MRI, ultrasound or other similar signal-based areas could create beneficial tools to medical professionals. By processing MRI data, for example, with filters that specifically determine tumour sizes in a patient could aid a doctor’s decision making prosses.

# References

Burgess, R.C. (2014). Filters, Analog/Digital. *Encyclopedia of the Neurological Sciences*, pp.299–307.

Smith, S. (1997). The Scientist and Engineer’s Guide to Digital Signal Processing.

‌

# Appendix