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](#_Toc40641577)

[1.0 Introduction 5](#_Toc40641578)

[1.1 Background 5](#_Toc40641579)

[1.2 Focus 5](#_Toc40641580)

[1.3 Value 6](#_Toc40641581)

[2.0 Literature review 7](#_Toc40641582)

[2.1 High Level Overview 7](#_Toc40641583)

[2.2 Filtering Techniques 8](#_Toc40641584)

[3.0 Methodology 9](#_Toc40641585)

[3.1 Introduction 9](#_Toc40641586)

[3.2 Code Breakdown 11](#_Toc40641587)

[4.0 Findings/Results 13](#_Toc40641588)

[4.1 Introduction 13](#_Toc40641589)

[5.0 Discussion 16](#_Toc40641590)

[5.1 Introduction 16](#_Toc40641591)

[5.2 RQ1. Which filtering technique resulted in the best removal of unwanted noise 17](#_Toc40641592)

[5.3 Which had the quickest execution times 19](#_Toc40641593)

[5.4 Best overall 20](#_Toc40641594)

[6.0 Conclusion 21](#_Toc40641595)

[6.1 To conclude 21](#_Toc40641596)

[6.2 Future work 21](#_Toc40641597)

[References 22](#_Toc40641598)

[Appendix 23](#_Toc40641599)

# 1.0 Introduction

## 1.1 Background

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.

Digital signal processing (DSP) is a method of using computer code to make a series of operations on signal data with the intention of creating data that is more useful to the user (need ref for defining DSP). As (Salivahanan, Vallavaraj and Gnanapriya C., 2000) describe, this can range from “…speech, music, picture and video signals.” (p1)

One of the main advantages of using software to do this is that designs can be created very easily, cheaply, and more accurately (providing the designer has good programming knowledge) than it is creating similar designs in hardware. This is because the design of the software can be amended and modified very quickly, and tested multiple times, in different ways in quick succession. In contrast, hardware components, such as capacitors, inductors or resistors, must be physically altered to change processing aspects. Hardware components also rely on each individual part working correctly. This introduces more potential points of failure to the system than software.

One part of DSP is filtering. Filtering is a broad term used to explain a removal of a signal from another.

Is Digital Signal processing the same thing as filters? If not, introduce filters, basic explanation of what they are and what they are used for, ideally with a couple references. If it is the same thing, make reference to it also being called ‘filters’, so that your next paragraph is clear what you’re using

## 1.2 Focus

The focus of this project is to make comparisons between different filtering techniques. Specifically, this project will compare the ability of filters to increase correlation between audio samples and their counterparts which have had noise manually added to them. The noisy samples will be filtered using different methods and the performance of each filter will then be examined. It is hoped that this initial exploration should indicate certain advantages and disadvantages of the filters in certain situations and varying parameters.

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. Do any of the filtering techniques contain a good mixture of both functionality and speed?

## 1.3 Value

The main scope of this project is that ideally the results can be applied to a broad range of situations where audio signals are used. Though that alone could provide useful, it is also possible that the scope for filtering techniques could be extended to a large range of different signal mediums. For example, it could be applied to data in in medical fields where electronic interference causes inaccuracies in readings. Finally, the results found in this project can contribute to the wider field of Digital Signal Processing literature, in which the benefits of filtering can have far reaching effects.

# 2.0 Literature review

## 2.1 High Level Overview

This section will discuss some of the main literature relating to filters and ways in which this form of signal transforming can be applied to different media/mediums. This will show why the current project looking at the use of filters on audio files is necessary.

To describe a filter on a signal, Steven W. Smith, (1997) puts it well in his book that filters primarily are used either to separate “…signals that have been combined,” or for the “restoration of signals that have been distorted in some way.” (p261). To expand on that point, an example of the use of filters is where they are applied to 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 consequently make the voice clearer. Smith (1997: 344) later goes on to make a comparison of analogue and digital filters and concludes that while “digital is slow; analogue is fast.” As he explains in the extract below, the digital bests analogue in almost all other areas

*“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” Smith (1997: 344).*

(Salivahanan, Vallavaraj and Gnanapriya C., 2000) also comment on the difference between the two stating advantages of digital include storage ease, linear phase characteristics, and cost for processing cost per signal.

## 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. The main reason for this is that it is the programmatic development process is more suited for this type of report.

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

The first family, FIR filters, use a method called “convolution” when applying their filtering to signals (ref).

This is fundamentally different to IIR filters. Although IIR filters also employ this method of convolution, IIR filters also take the output of the expression for further functions (ref).

The two different styles of filtering produce different characteristics. (Burgess (2014: pp.299–307), in his comparison of the two filtering techniques explains some of the advantages and disadvantages of the two, comments that while FIR has a lower efficiency than that of IIR, it is much more stable.

The main reason for this is due to FIR having no feedback, no matter how it is implemented.

(Parks and Burrus, 1987) adds that “For narrow-band, sharp cut-off filters where phase is not important, IIR filters are likely to be superior to FIR filters.” (p14)

The way in which each filter is applied to the media file can also change. The two methods that this report will use are the “bandstop” and “lowpass” filter. Firstly, A bandstop is a filter that reduces the power of frequencies between two points (ref).

For example, a bandstop filter could have a range between 300Hz and 400Hz which would reduce the power of the signal between those frequencies, while leaving the other frequencies relatively unaltered. Secondly, a Lowpass is a filter that allows up to a certain frequency to pass through unaltered to a point. An example of this would be setting a lowpass filter at 80Hz to filter all frequencies after 80Hz, whilst leaving the rest of the frequencies to remain as per the original file.

A lot of research has been undertaken with the implementation of different filters being used as a tool rather than a focus. For example, the work of Balodi, Dewal and Rawat (2015), who used filters to do reduce the speckle count in ultrasound images and Berger, Demissie, Heckenbach, Willett and Zhou (2010) who used DSP in radar analysis. In contrast, this report will seek to directly compare the effectiveness of different filters when each are applied to the same audio files.

# 3.0 Methodology

## 3.1 Introduction

#### Recap of questions

To recap this project aims to assign values of functionality, speed, and an evaluation based on both things. The following methodology was designed to address the following research..

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. Do any of the filtering techniques contain a good mixture of both functionality and speed?

#### 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.

(Parks and Burrus, 1987) writes that “Higher level languages like *matlab* make it possible to write powerful programs much faster and more reliably.” (p25). The speed of development in MATLAB 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 .

As discussed previously, the media type that this project uses to compare the effectiveness of different filters is audio files. In order to select the audio files which would become the ‘samples’ for this study, a few considerations needed to be made. To avoid any copyright issues, Five audio files were picked from <https://www.bensound.com/royalty-free-music>, as this database contains audio files which are in the public domain and free to be used without infringing on copyright ~~.~~ In order to avoid any bias, the five audio files that were chosen were the first five that appeared on the database. Although more representative results could be obtained by applying filters to a large number of audio files, it was believed that using five files was an appropriate quantity for this initial exploration into the effectiveness of filters on audio files.

## 3.2 Code Breakdown

This project was completed entirely in MATLAB and the following is an overview of the code itself. For the filtering, the MATLAB application “Filter Designer” was a useful resource in making the filters. An extract of the code can be seen in Appendix 8, and the full version has been submitted alongside this report.

Firstly, the five 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 and an iPhone X microphone was used to capture the sounds. Both the 30 second files, and the recorded files were then imported into the program in a for loop and placed in separate matrixes. During the import, the two audio channels were 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.

Secondly, a loop ran through each recorded file and found the correlation of it against each of the original files. This found at which point the two files were most similar, and then 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”. These scenarios aimed to represent situations which could exist in real world applications. The three 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. The only difference in the setup is that the signal with the vacuum cleaner was filtered three times. This is because, unlike the two others, the vacuum cleaner noise spanned multiple frequencies and mostly where the original signal resided. This meant that a filter applied across the same range, that the noise spanned, would also cut out the track that we were trying to restore. So instead, two bandstop filters were applied and one lowpass.

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.

For each track that has had a signal added to it, 5 filtering techniques will be applied to it. The filters chosen for this investigation were as follows: Butterworth, Chebyshev Type I, Window, Elliptic, Equiripple.

These were chosen because .... (I don’t know if you had a choice of more than the ones you chose, but if you did then mention why you didn’t use the other, maybe they’re just not appropriate or maybe you think these are better.

Once complete, the output data will be imported into Microsoft Excel to take advantage of the programs ability to apply conditional formatting and present the data in a more accessible and informative format. Once there, trends will most likely begin to develop as different data points are manipulated using graphs to display groupings of THINGS

# 4.0 Findings/Results

### 4.1 Introduction

In this chapter we will present the findings and discuss some of the key results obtained from the experiments. In the following chapter these results will then be discussed with reference to what they mean. All the data obtained as a result of applying the filters can be seen in Appendices 1-5. An example extract is below.

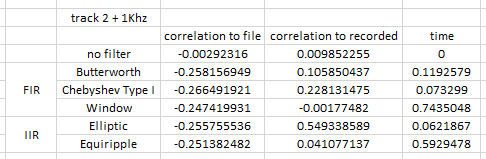


Figure 1. A typical table of results.

Above, figure 1, is the output of the experiments for one of the tests ran.

This specific test compared the recorded version of track 2, with an added frequency of 1KHz, against the original file, the recorded file and the time it took to perform the filter. It did this with no filter applied, and then the subsequent 5 filters.

To the left, we see that each row represents the type of filter, and whether that filter it is part of the FIR and IIR family.

The ‘no filter’ row has been included in the data as a reference point of how the results would look without any filter being applied. Because there was no process being completed, the time taken to process is zero for the no-filter option/variation. After taking all the absolute values and averaging them, figure 2 below shows a graph of the results in a more presentable manner.

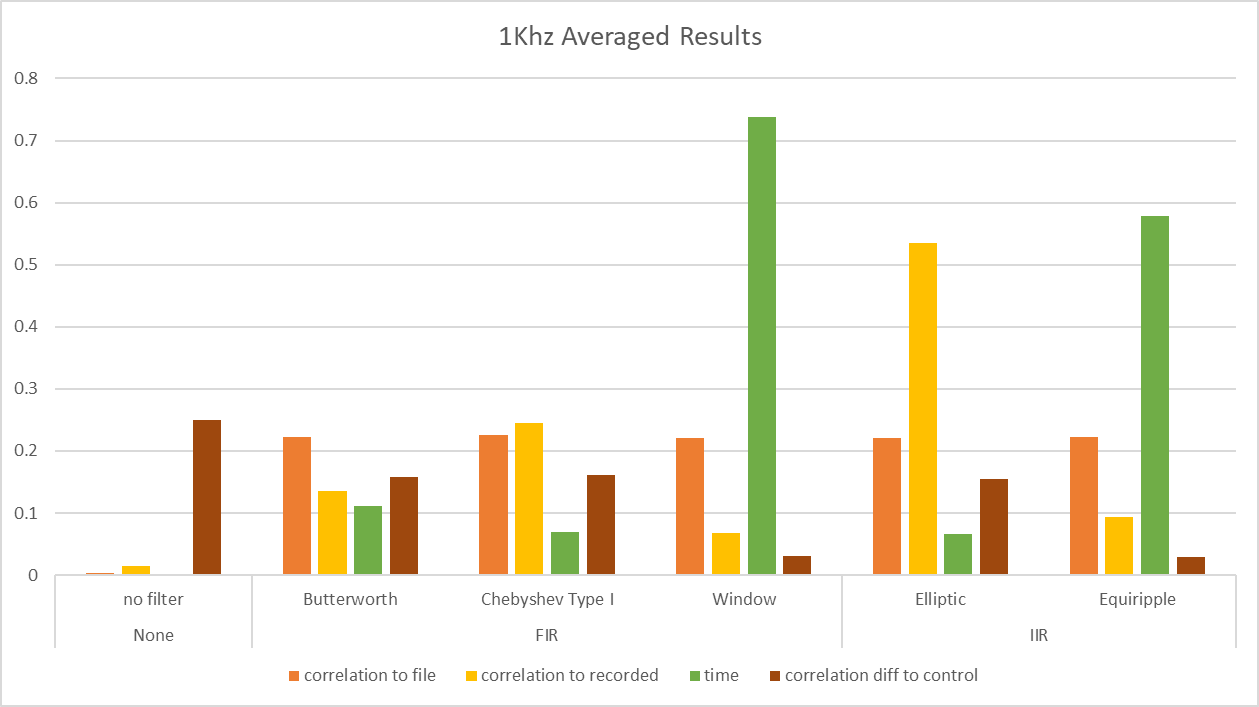


Figure 2.

The results in Figure 2 above have been categorised as having either no filter applied (left), a FIR filer (middle section), or an IIR filter (right). Looking firstly at the correlation between filtered signals and the original file we can see, on the whole, that there is very little deviance between the different filters. Ideally, we would look for a higher value for correlation to file as this could indicate a better correlation and therefore a more favourable result for the filter.

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. The other results can be seen in Appendices 1, 3 & 5. Again, a higher value represents a higher correlation and is favourable. When looking at correlation to recorded 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. For this section the favourable value would be the lower number, as this would indicate a quicker execution time. The window and equiripple filter techniques are shown to be many times higher than that of any of the others and therefore the least desirable, based purely from an execution perspective

The last value represented on this graph is the correlation difference to the control. In this case this is the correlation between the original file and the recorded output version. While it is not inherently true that a lower value is more favourable, we already know that the correlation between the filtered results and the original file does not vary greatly (see first value in Figure 2). Therefore, a lower number in this situation 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 introduced and presented the results of the experiment, displaying the effects of different filtering techniques when applied to noisy signal In the next chapter the results will be more closely investigated and discussed.

# 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 file compared to the original to ascertain what to expect from the filtered resultsof the recorded tracks correlated against the original files. Figure 3 shows the results of this. – I don’t completely understand what you’re trying to show here...why did you need to use a baseline? Also this section doesn’t really feel like an intro, I think its necessary just maybe it’s a different section

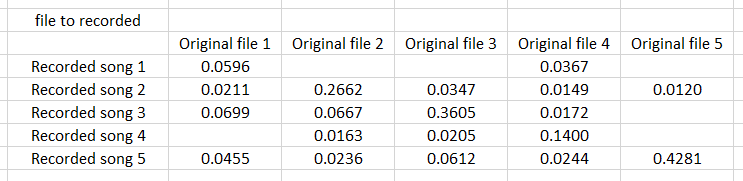


Figure 3. Cells without any values represent very poor correlation. All values are absolute values of results

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. This could be due to the two tracks having similar structures and frequency ranges and the degradation of the signal from being recorded exposes these similarities more. The table shows that the comparison function shown in appendix 7 works. In summery, the function assess at which point the two tracks share the highest correlation coefficient, arrange them at that point, and then output the similarity coefficient. This then gets stored in a matrix, much like the one seen above in figure 3. If, when comparing two tracks, the point at which they are most similar falls at a time where it would be impossible for the tracks to line up without the recorded version exceeding the end of the original track, then it is deemed to not be statistically similar enough. The cells that contain not values represent these outputs. While we cannot be sure to what extent, we can measure against this table to make comparisons later. In some cases in figure 3, 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 RQ1. Which filtering technique resulted in the best removal of unwanted noise

Figure 4 below encompasses all of the results obtained from the three different scenarios (adding 1Khz noise, 60Hz noise, and a vacuum cleaner noise). The coloured bars each represent one of the five filters that was applied (plus no filter condition), and the correlation coefficient value is the average result of that filter when applied to the five audio tracks (I don’t know if that even makes sense but just make it clearer what is in the graph). We are able to display the averages without any lose of context as the data shows, as shown in appendices 2, 4 and 6, that filters produce results that are consistent across the 5 tracks. It is worth noting that in all instances shown below in Figure 4, a higher result is more desirable.

Figure 4. Each section shows the results from the recorded track, with added noise, compared to the original recorded track.

One of the most obvious things from looking at the above chart is that the elliptic filter is much more effective than the other filters at returning the signal to its original state. In fact, the first two scenarios (1Khz and 60Hz respectively) for elliptic in figure 4 are the highest scoring correlation out of any of the tests. It is therefore interesting that this was not the case where the vacuum noise was added. Instead, with the vacuum scenario, we actually see that not applying a filter at all is more effective than any of the filters tested. However, the second highest correlation comes from using the elliptic filter, thus showing its strength again.

Moving on to Chebyshev and Butterworth. Chebyshev performs marginally better out of the two but not to the point where it could be considered statistically significant.

Equiripple and window are both very low scoring results on all three of these scenarios, in the context of the other filters. Even without the others as a comparison, consistently scoring lower than 0.1 correlation is a very poor result and would rarely be acceptable in any real world application.

This does not paint the full picture though. Figure 4 shows results when comparing the noisy tracks to the recorded file. What about comparisons to the original file?

Figure 5. Each section shows the results from the recorded track, with added noise, compared to the original file.

Figure 5 shows us that in this graph we see a much more uniform distribution of results, especially with the 1Khz noisy track. The results from the first section (recorded + 1KHz vs File) have very little variation, and so will not be discussed in detail. The same could also be said for the vacuum scenario though the average result of the filters in that scenario are much lower than any other result. A reason for this could be that due to the broader range of frequencies, it is harder to accurately remove the noise while maintaining the signal information that we want to keep. In contrast, however, the recorded + 60Hz results do show some variance. The equiripple and window filtering methods display much lower scores than the rest, whilst still managing to produce higher scores and therefore be more effective than using no filter. As for Butterworth, Chebyshev and Elliptic the 60Hz scenario also has all results so close that no single filter method could be said to be vastly superior.

We can say that one filter is better or worse than other filters based on our experiments but only within the scope of the parameters that were set. For example, though the two worst performing filters were poor performers in when comparing against the recorded file, they were both met the same average as the rest of the filters when filtering out the 1KHz noise and comparing it against the original file. This could be because the 1Khz signal was more isolated in its frequency range than the other scenarios. Both filters also struggled to filter the 60Hz. This could be because the filer parameters were so close to the start of the signal and the subsequent ripple effect the filter creates distorts when so close to the end of a signal.

BEACUASE CLOSE TO START!?

## 5.3 RQ2 Which had the quickest execution times

The second way in which this project assesses the effectiveness of different filtering techniques is by examining the execution times of each. When applied to the five different audio tracks, each of the five filters remained consistent in its execution time. Figure 6 shows the averaged results for each filter under each scenario where a longer time is less desirable. The data and further graphs can be seen in Appendices 5 and 6.

Figure 6. Average time taken for each filter to execute.

As noted previously, as no action was being performed for the no filter part, there was no time recorded so it was not included in figure 6. Again, the results are consistent through the three scenarios in that equiripple and window both took much longer to compute than any other method, by several orders of magnitude. This is not due to inherent flaws with the type of filter they are, as equiripple is IIR and window is a FIR.

why is the family type not a reason why they took longer? And also, if it’s not due to that, suggest what it could be due to?

Interestingly, in the vacuum scenario neither Chebyshev nor elliptic seem to increase much compared to their scores in the other two scenarios. This is especially significant because for the vacuum scenario the signal was filtered 3 times and the execution time was taken from before the first and after the third. This is particularly good because it had lots of work to do.

## 5.4 RQ3 Which showed a combination of GOOD and fast

The final research question that this project aimed to address was whether any of the five filters could be considered to have a good mixture of both functionality and speed. Using the findings from RQ1 RQ2 this section will provide an insight into which, if any, of the five filtering techniques could be considered the most effective.

Looking at the two previous sections some of the filters really stand out. We saw that equiripple and window both not only had some of the worst results in terms of not only the correlation coefficient they produced, but also the long execution times.

A long duration of a filtering method is not on its own a bad thing. If a technique takes a longer time to process but provides a much better result, than for some applications where time is not as important, it would be acceptable for it to take longer. This could also be flipped and be true in the sense of the correlation. A filter that works in a real time setting, such a noise cancelling device like headphones, the time it takes is much more important than a perfectly clear signal.

Finally, the filter that provided the best results for removing unwanted noise also happened to be the fastest in it’s execution. This is the elliptic filter. As shown in the results, the elliptic filter consistently produced either as good as, or better, than all the other correlation in every scenario and was the fastest, only beaten by not using a filter at all. From the results of this experiment it can therefore be concluded that, under the requirements that were set, elliptic is the best filter to use when looking for quick execution and effective removal of unwanted signals in audio files.

5.5 C­ontinuing You make your points about elliptic being good. Then you can say ‘in contrast these ones weren’t so good’..... However in these situations they actually did okay. Gives a balanced discussion of your results rather than making you look biased. You can still then say, overall, in your final section which you think is best despite the varied results that you got.

# 6.0 Conclusion

## 6.1 To Conclude

At the start of this report, we discussed that filters can be devided into two families, FIR and IIR types. Neither of the two

Based on the findings in this report, it is clear that the Elliptic filter is the best filter to use, from the ones tested, for use in removal of unwanted sound in an audio file.

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 Limitations and Future Work

With any project, limitations

#### Limitations

Future work and extensions of this project could look to change the type of data being used. Although we have used audio files, A suitable example would be .... find another area but less interesting/important than medical field.

Another way in which this work could be adapted to other data types is within 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, filters could be applied that specifically determine tumour sizes in a patient. This could be greatly beneficial to a doctor’s decision making process

In the report the filters success was measured in a programmatic way. Another way to have gained results would be to create a survey and have participants hear the files and rate what they consider to be the clearest, or best sounding. This kind of data would be beneficial as it would add some real world STUFF

# 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.

Salivahanan, S., Vallavaraj, A. and Gnanapriya C., 2000. *Digital Signal Processing*. New Delhi: Tata McGraw-Hill Pub.

Parks, T. and Burrus, C., 1987. *Digital Filter Design*. New York: John Wiley & Sons.

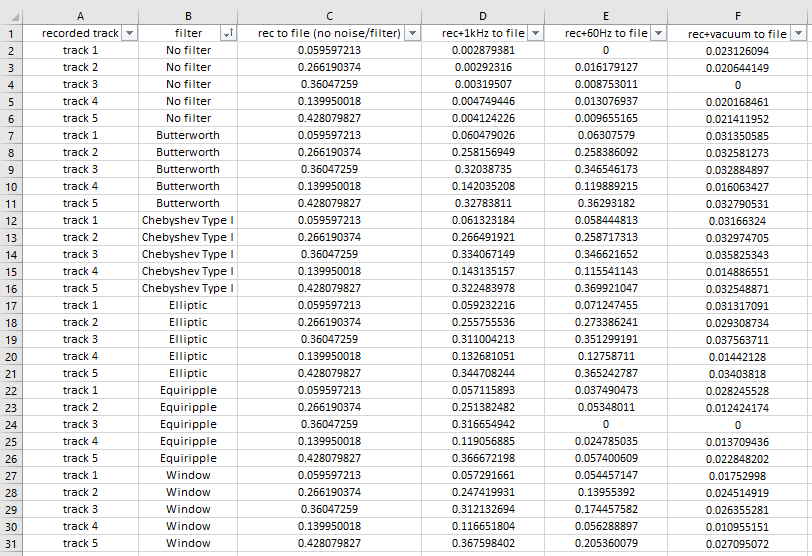
‌ Smith, J., 2008. *Introduction To Digital Filters*. Stanford, Calif.: Stanford Univ., p.25.

A. Balodi, M. L. Dewal and A. Rawat, "Comparison of despeckle filters for ultrasound images," 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, 2015, pp. 1919-1924.

C. R. Berger, B. Demissie, J. Heckenbach, P. Willett and S. Zhou, "Signal Processing for Passive Radar Using OFDM Waveforms," in IEEE Journal of Selected Topics in Signal Processing, vol. 4, no. 1, pp. 226-238, Feb. 2010, doi: 10.1109/JSTSP.2009.2038977.

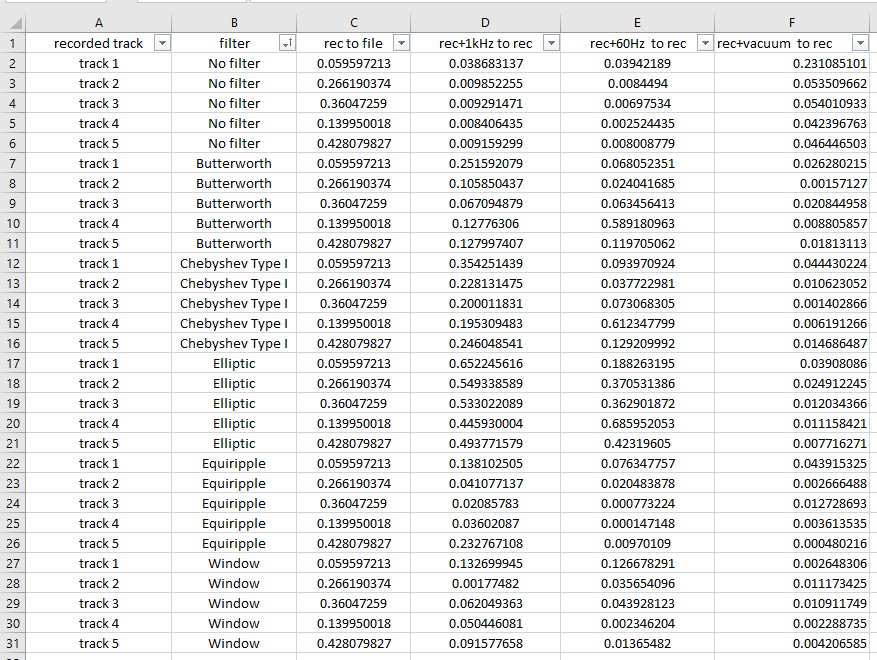
# Appendix

Appendix 1 – Recorded files, with noise, compared to original file



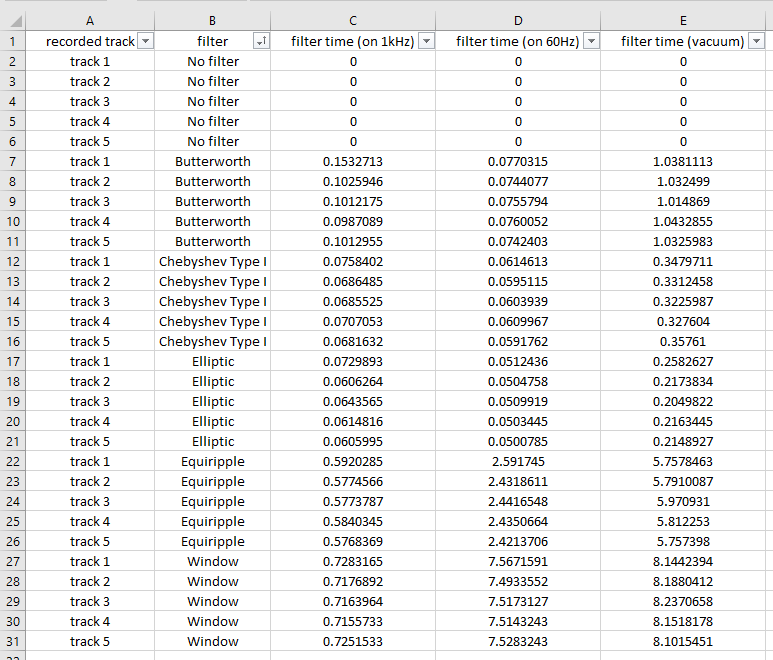
Appendix 2 – Graphs from the above data in appendix 1

Appendix 3 – Recorded files, with noise, compared to recorded file



Appendix 4 – Graphs from the above data in appendix 3

Appendix 5 – Data of execution times



Appendix 6 - Graphs from the above data in appendix 5

Appendix 7 – Code segment of the function used to compare two signal and output the similarity coefficient.



Appendix 8 – A code extract for the formulation of a butterworth filter used filter the different signals.

