EGB 240 Assessment 2

Author: Joseph Haddad

Student Number: n10535268

Submission Date: 29/05/2020

1.0 Executive Summary

The following report documents the design from concept to final design of a DC removal, Anti-Aliasing, and Gain filter for an Electret Microphone. This circuit should be designed to condition the input signal of a microphone to interface with a microcontroller which acts as an Analog to Digital Converter – (ADC). It is required to meet a few constraints as listed below:

- Run only on 5V provided by the microcontroller.
- Need to use the TL974 operational amplifier.

This design task is fairly open only providing only a few constraints allowing for the design to be targeted towards more effective performance in different areas and feature unique assumptions. The circuit presented by this report will target the range of human voice only. This range is from research found to be (300 – 3400hz) [1],[5],[3]. The design is greatly affected by components and their tolerances which could affect the performance of how well the passband ripple stays under 3dB. The passband at 3400hz should be less than 3dB which in accordance with Appendix F for a single case was 1.2dB for this design. Whilst the stopband was around 80dB down which well outperforms the 48dB. Therefore, the use of low tolerance components has been implemented to ensure that it says under 3dB in all cases.

Table of Contents

List of Figures

2.0 Introduction

This report entails the design process conducted to implement an analog input conditioning circuit. The design should be able to condition analog signals within the human voice frequency range that come from the electret microphone and then allow them to interface with the Arduino. The required frequency range in accordance with [5] ranges from (300hz - 3400hz). Since it is being designed to be interfaced by the Arduino it needs to be able to run off the 5V power supplied by the Arduino teensy board only. No external power should be supplied and it should be based upon the TL974 operational amplifier.

A background research section has been included to help showcase the specifications of a similar system like a telephone should perform in the real world. A telephone system has been selected as it is based on the same principles as this design. Telephones receive human voice frequency inputs by the user talking into its microphone. This is then sent in real-time to another users phone to generate an audible output of the frequencies via a speaker.

In terms of the design process of implementing the stages of the signal conditioning from concept to the final design has been documented. It shows the initial concept process on MATLAB and the LTSPICE simulations for each of the different stages of the filter and development. This helps highlight and evidence the design performance whilst portraying all the different choices that have been made in terms of the design, as it needs to meet the required specifications. The stages it has been split up to are DC Bias filter, Anti-Aliasing filter, and Gain filter. The DC Bias filter has two main jobs. It needs to drop the voltage down to 2.5V so that it does not exceed the common-mode input range of the op-amps. While also removing the DC voltage via AC coupling it also server a smaller third purpose as it also acts as a high pass filter which needs to at a minimum allow only frequencies above 300hz to pass through. The Anti-Aliasing filter is there to act as a low pass filter which will be designed for 3400hz. This has been implemented using the Chebyshev filter design and is aimed to provide a passband ripple that is less than 3dB as in accordance with [5] dB changes up to 3dB are not easily perceivable. However, the lower the ripple the better and less distorted the design will be. To comply with the specifications set by the Arduino it must also be able to drop at least 48dB by the time 7800hz is reached. If it drops more the design beats the spec allowing for a better filter. Lastly, the gain filter does the reverse of the DC offset filter. It brings the voltage back up to as close as possible to 5V to allow for interfacing with the Arduino. These are further discussed in section “4.4 Technical Design”.

To showcase these stages and their performance, simulation screenshots from LTSPICE and a full circuit diagram has been added these document the performance of the design and show how it performs under the set specifications. These draw comparisons to the background research to see if it performs appropriately or not. On the flip side, in the conclusion and future development sections, a summarising discussion of the overall results and where in the future this design could be improved are discussed. Representing the key findings and performance improvement methods.

3.0 Background Literature

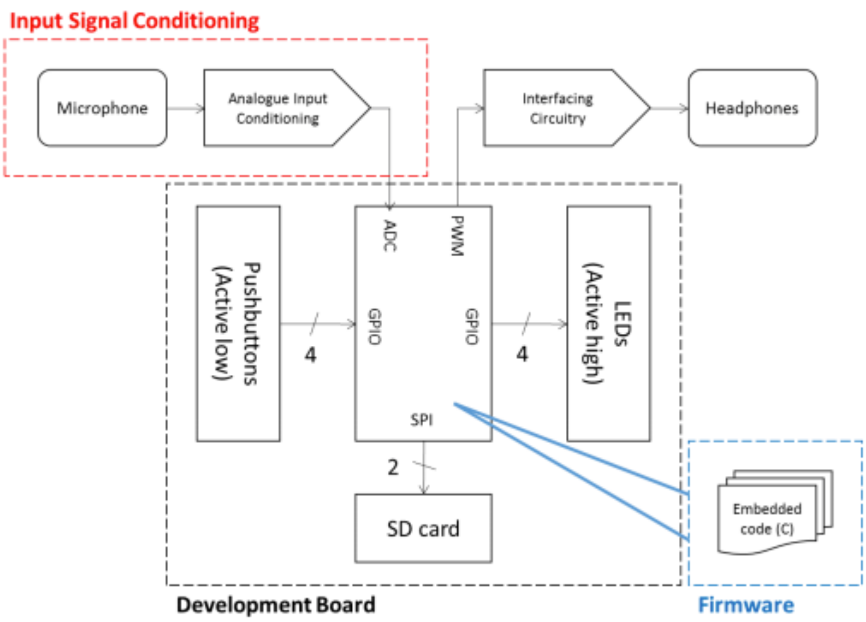
The frequency content of human speech and human hearing is key to the design being effective. The microphone needs to be able to pick up the frequencies a human can output by speaking into the microphone. It needs to take this and in real-time process the frequency so that it generates an audible input to the ADC in which further processes it so we can hear it. This system can be compared to that of a telephone, it too transmits a voice signal from one telephone to another and in real-time converts it so something the user on the other side can audibly hear. According to [5] the range of human voice varies depending on gender age and person to person, it's not just a set number of 300hz as it depends on many different factors such as age, gender, distance from the microphone and even language. For males, the typical values are around 120hz whilst females have a typical 210hz. Interestingly this can change with age and can decrease by up to 15hz. Certain letters or words could produce a higher frequency than what's mentioned above creating a range that is quite broad.

In terms of human hearing, not only does the frequency matter but also the change in dB can affect what's perceived. According to [1] An increase of 10dB to and audio level produces a perceived doubling of volume, whilst a change of 3dB is hardly perceived. The same source suggests that an effective range used by telephones to cover the frequencies of speech and hearing over a telephone line are (300hz - 3400hz). This provides enough room of speech frequency to ensure that all the needed signals are reached only keeping the system efficient. They also suggest that when converting to a digital signal. Sampling should be done at a rate double than the sampling rate of 3400hz. This information should assist with the development of the design allowing for the needed signals to be recorded in an undistorted or unattenuated form. It is essential to keep the passband ripple of the filter as low as possible and at the very lease must be under 3dB. Since as mentioned above by [5], a change of 3dB is hardly perceived by human hearing.

4.0 Design

4.1 Project Description

This project aims to design an analog input signal conditioning circuit. It is based on the frequencies of human speech which through research is (300 – 3400hz) [5]. These will be inputted via a microphone which will then go through a filter design that removes the offset voltage, then filters the signal and finally amplifies it to an audible level. This circuit is designed to that it can interface with a system as pictured below to allow a user to connect headphones to a microcontroller and listen or record audio.



4.2 Design Goals, Specifications & Scope

One of the main goals of this project is to allow for efficient methods of conditioning analog human speech frequencies. The use of minimal components to achieve the desired result without any drawbacks is one of the main goals of this design. This means a potential customer for this design can cost-effectively implement the signal conditioning circuit to their microphones or projects. Additionally, the design must be able to meet the same specifications as a telephone, as found in the background literature section. This frequency ranges from 300hz – 3400hz. Additionally, since a Chebyshev filter is going to be used for the design there will be some passband ripple. The goal is to keep this under 3dB as in the background research it has been found that humans cannot easily decipher between a sound that changes by only 3dB. So keeping the ripple as low as possible will reduce the amount perceivable noise/distortion outputted. Since many components such as resistors and capacitors are going to be used. They each have small tolerances which may make a design end up being over the 3dB ripple. Therefore, these should be accounted for in the calculations. Another requirement of the filter is that it must drop down 48dB as a minimum when it reaches 7800hz to meet the Arduino specification.

4.3 Methodology

To approach this problem, the separation of the different sections of the design has been made to allow for individual testing. These sections are DC Removal, Anti-aliasing, and an Amplifier. Starting with the DC Removal filter, this stage was entirely constructed and developed in LT SPICE. After this stage, the anti-aliasing filter was looked at. Through the use of initially, MATLAB and Excel refer to Appendix B, C, D, E. The order of the filter, components, and their tolerances were determined. Using this information it was then designed in LT Spice and tested to see if it meets the desired specifications. Lastly, the gain component was designed last. This is so that the desired gain could be achieved which is determined by looking at the transient graphs of the other stages.

4.4 Technical Design

DC Removal Filter:

This component is directly connected to the microphone which provides 5V. The filter aims to drop this down to 2.5V whilst keeping all the AC. Therefore, DC removal by AC coupling has been put in place to filter out the DC voltage and only leave the AC. The reason this is important to the circuit is due to the DC bias level of the microphone being unknown making it possible to vary from even the same brand of the microphone. This could generate the risk of exceeding the common-mode input range which in turn will mean the circuitry will not operate as it should. This acts as a buffer between the microphone and the rest of the circuitry and allows for a known bias level.

The designed circuit for this buffer is like an inverting amplifier, however, it has unity gain to ensure no clipping occurs due to exceeding the common-mode input range. This by itself does not remove DC. The addition, of a capacitor to the inverting input of the amplifier is needed as it blocks DC. This all together forms a circuit that looks similar to a high pass filter circuit. Allowing for blocking of low-frequency signals. In this case, it must only block frequency signals below 300hz. As 300hz is part of the 300hz to 3400hz desirable range as described in the (background literature 3.0). Using the formula below the calculation needs to be less than 300hz to allow for the desired frequencies to pass through. As calculated it is around 2.001hz which provides plenty of room for the 300hz frequencies to pass through.

$$\text{High Pass Frequency Cut-off} = \frac{1}{(2\pi RC)} = 2.00953211\text{hz} < 300\text{hz}$$

These numbers as mentioned in section 3.0 are based on the telephone frequencies as they perform a similar job to the proposed microphone circuit. An advantage of using this type of inverting amplifier configuration is that the 2.5V reference voltage can be provided straight from a voltage divider instead of having to use another op-amp to provide a buffered reference voltage like how a non-inverting amplifier system requires.

From LT SPICE the designed filter pk to pk is 31mV which is under the 50mV input range threshold. Here an assumption about R5 on the microphone has been made to leave it be 1k7. Which as shown in figure 1, leaves the input voltage to be around 4.17V. After having the dc removed from it, and having it dropped to 2.5V. The graph now is as desired with no clipping and is just about centred at 2.5V. As pictured in figure 2.

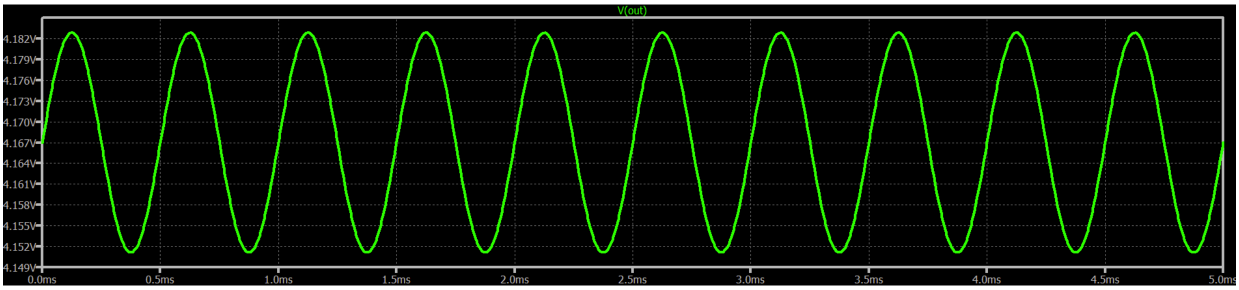


Figure 1

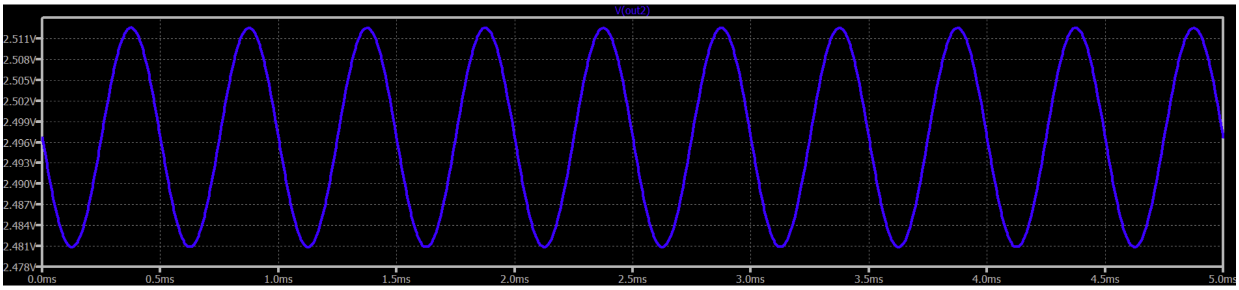


Figure 2

The graph in green (figure 1) shows the output from the microphone. Whilst the one in blue (figure 2) shows the output of the DC bias filter. There is no DC voltage in this signal, whilst being closely centred at 2.5V to avoid clipping. Refer to figure 10 and figure 11 to see an enlarged version of figures 1, 2 respectively.

Anti-Aliasing Filter:

Anti-Aliasing is needed as it acts as a low pass filter for the circuit. This limits the frequency to the researched 3400hz. This lowpass filter must drop 48dB before it reaches 7800hz as shown in the picture below.

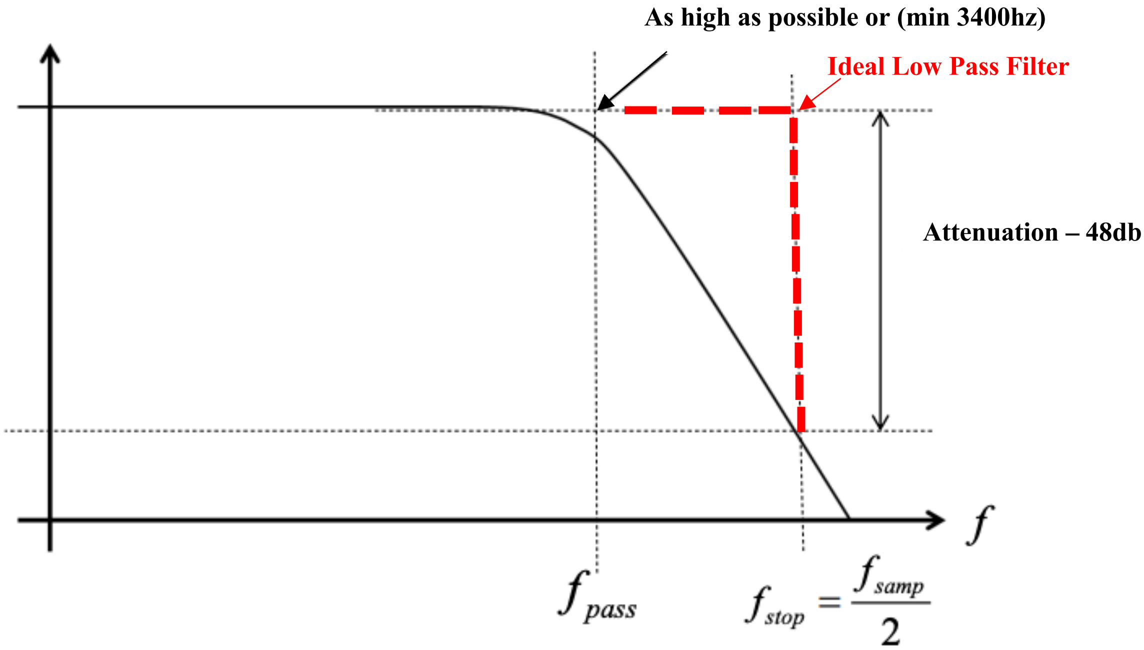


Figure 3

f_{stop} = can be max 7800hz. It is ideal to get a lower number at -48dB to beat the spec.

Figure 3 demonstrates it is ideal to get the graph as steep as possible to make it closer in performance to the ideal low pass filter. This is demonstrated by the dotted red lines in figure 3 which represent an ideal low pass filter. The black graph shows what an antialiasing filter looks like in a real scenario. It is more curved as in the real world you cannot get it as perfect as the ideal case. There are 2 different main strategies to approach increasing the steepness of the curve. Either increasing the filter order or changing the filter type. The design proposed in this solution features a 6^{th} order Chebyshev filter. The reason a 6^{th} order filter was selected is since it provides a steep curve, whilst not making it too difficult to tune it so that resistor tolerances won't have a strong effect on the design. To make the curve even steeper a Chebyshev filter was also used. It allows for the use of a lower order filter to achieve a high steepness. This comes at the cost of having a passband ripple, which is acceptable in this design if it is left to be under 3dB.

To design this filter the initial concept was developed on MATLAB refer to Appendix E. This allowed for component value approximations and the determination of the order of the filter required. Then alongside excel whose calculations are included in Appendix E. More specific component values were selected in preparation for designing the filter in LT Spice. In LT Spice these values alongside their tolerances were entered to see how the filter performed example of this is in figure 12. This process repeated with small tunings to perfect the filter to what is currently proposed in figure 6. Since this filter is based on the Chebyshev design the passband will have some ripple. As researched in section 3.0 humans can barely perceive the difference between sounds that change by 3dB. Therefore, if the passband is kept lower than 3dB it will be optimal. In this design, the passband is kept lower than 3dB and that is when accounting for component tolerances. When not accounting for component tolerances it records a ripple of around 1.2dB whilst when putting them into account they do vary a bit but are kept under 3dB. These specific values are shown in Appendix A and F.

Another tool used to help select low tolerance components is the circuit calculator referenced in [2]. This tool is highly recommended as it checks to see the closest standard component value to the desired one from calculations. In this case, it was used to find appropriate values from the excel sheet. This allowed all the resistors to feature a tolerance of 1%, and the capacitors are either 10% or 5%.

An aspect of the anti-aliasing filter design which hasn't been covered yet is the stopband. To meet the specifications of the ADC microcontroller, it needs to drop 48dB at 7800hz. The proposed design outdoes this by dropping approximately 80dB as outlined in Appendix A when taking account component tolerances and Appendix F when just looking at a single case. This is should easily allow for the desired range to be interfaced by the Arduino.

The organisation of the stages of this filter going from stage C to B and then A has to do with the q values calculated in MATLAB. Putting the smaller q stages first allows for less of a risk with there being clipping in the circuit.

Gain Filter:

All the gain has been left to the last stage to allow for a k value 1. Which is why there is a need for this gain stage. This stage aims to bring up the voltage from 2.5 volts to as close as possible to 5V to allow the full range to be used. The gain is developed through the use of an Inverting op-amp circuit. This circuit produces a gain of 170 and uses two 47k resistors in the reference voltage component to centre the wave at 2.5V which allows for maximum gain without distortion or clipping As seen below in figure 4. An enlarged version of this graph is located in figure 9.

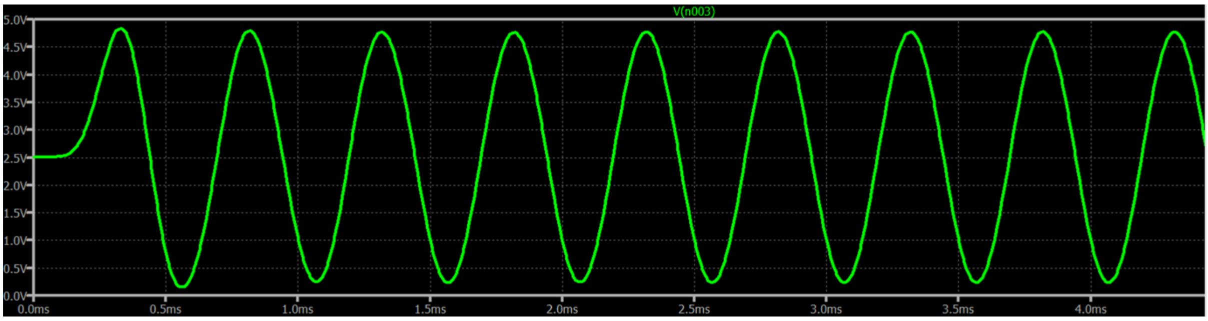


Figure 4

Design Assumptions:

Starting with the microphone simulations circuit. The resistance has been left to be modelled as a 1k7 resistor. In terms of op-amps each TL974 features 4. The design will require 5 to ensure that component tolerances are under 3dB and that the filter is steep. When designing the filter in MATLAB assumptions about the components were made so that a K of 1 is achieved. This is done by making R1 = R2 and leaving all the gain to the last stage. These in the final design were tweaked in the excel sheet to achieve optimal performance.

4.5 Implementation

This design only focuses on the simulation side of things. As shown below the proposed circuit to allow effective interfacing with the ADC is designed.

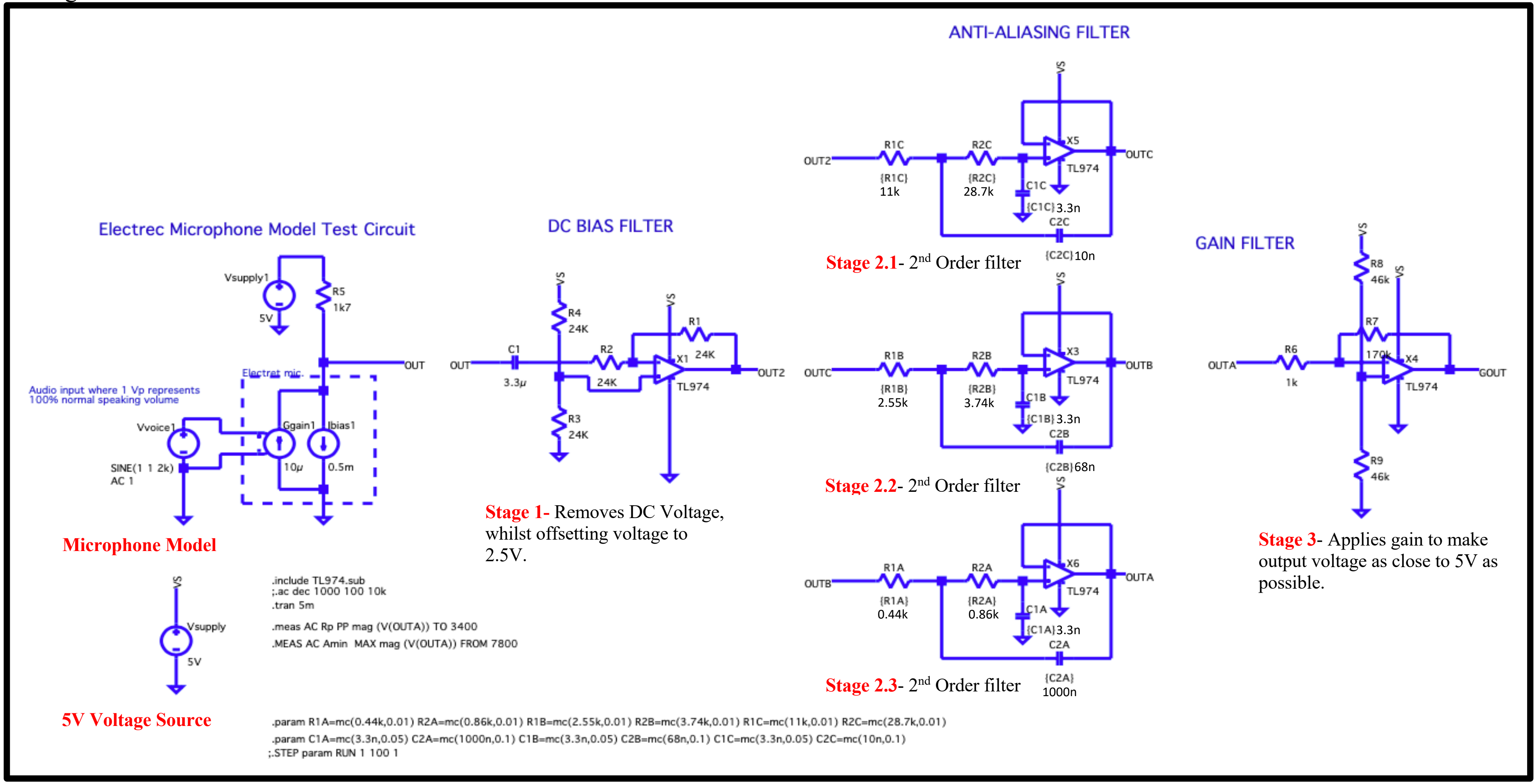


Figure 4 - Schematic Diagram

4.6 Simulation

OP Amp Model : .include TL947.sub
This relates to all the simulations as in all of them the same op-amp is used – (TL947)

AC Analysis Command: .ac dec 1000 100 10k
AC analysis command relates to simulations with bode plots: Simulations - (1, 2, 3, 7)

Transient Analysis Command: .tran 5m
This command relates to all the sin/cos wave plots: Simulation – (4, 5, 6)

Resistor Values (Anti-Aliasing): .param R1A=mc(0.44k,0.01) R2A=mc(0.86k,0.01) R1B=mc(2.55k,0.01) R2B=mc(3.74k,0.01) R1C=mc(11k,0.01) R2C=mc(28.7k,0.01)

Capacitor Values (Anti-Aliasing): .param C1A=mc(3.3n,0.05) C2A=mc(1000n,0.1) C1B=mc(3.3n,0.05) C2B=mc(68n,0.1) C1C=mc(3.3n,0.05) C2C=mc(10n,0.1)

Component Tolerance Simulation: .STEP param RUN 1 100 1

Command to check passband: .meas AC Rp PP mag (V(OUTA)) TO 3400

Command to check stopband: .MEAS AC Amin MAX mag (V(OUTA)) FROM 7800

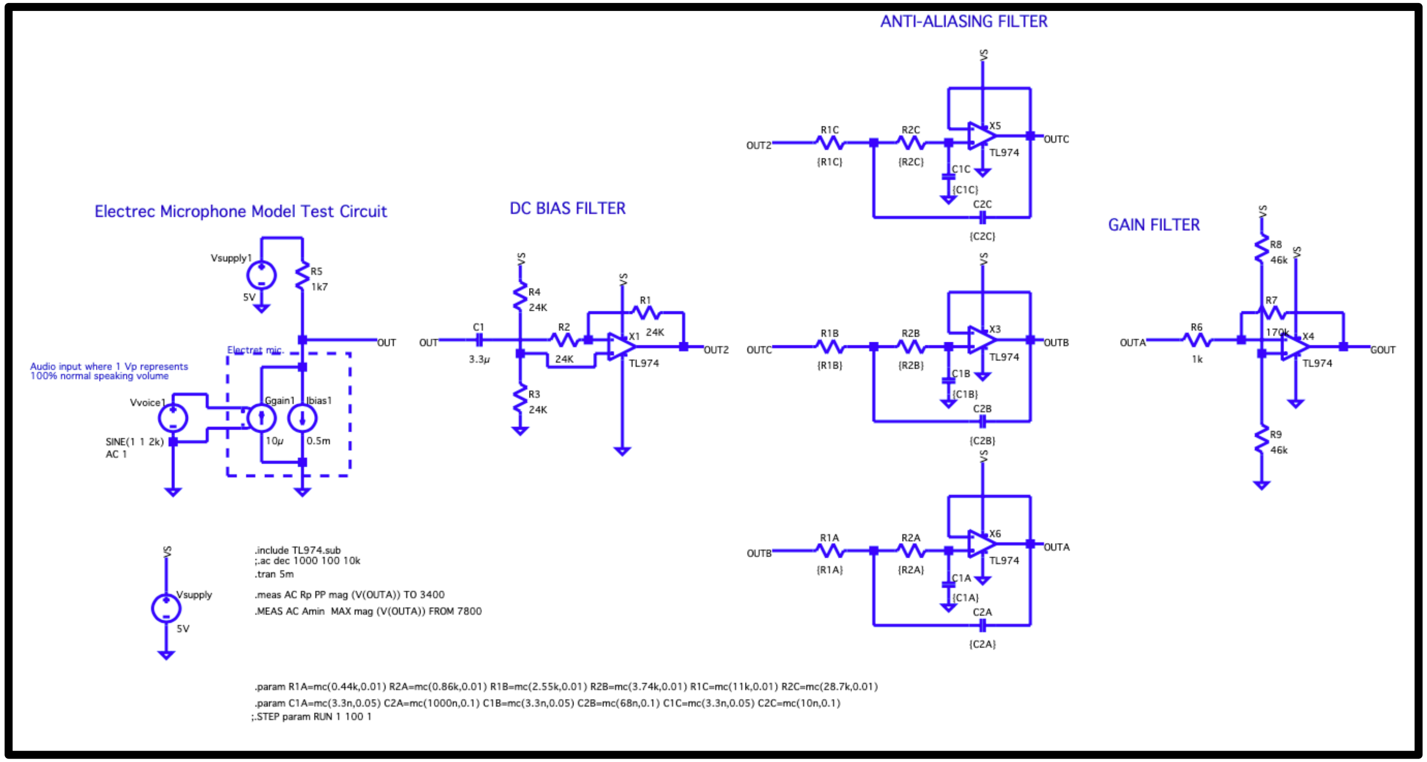


Figure 5.1 – Simulated Schematic Diagram

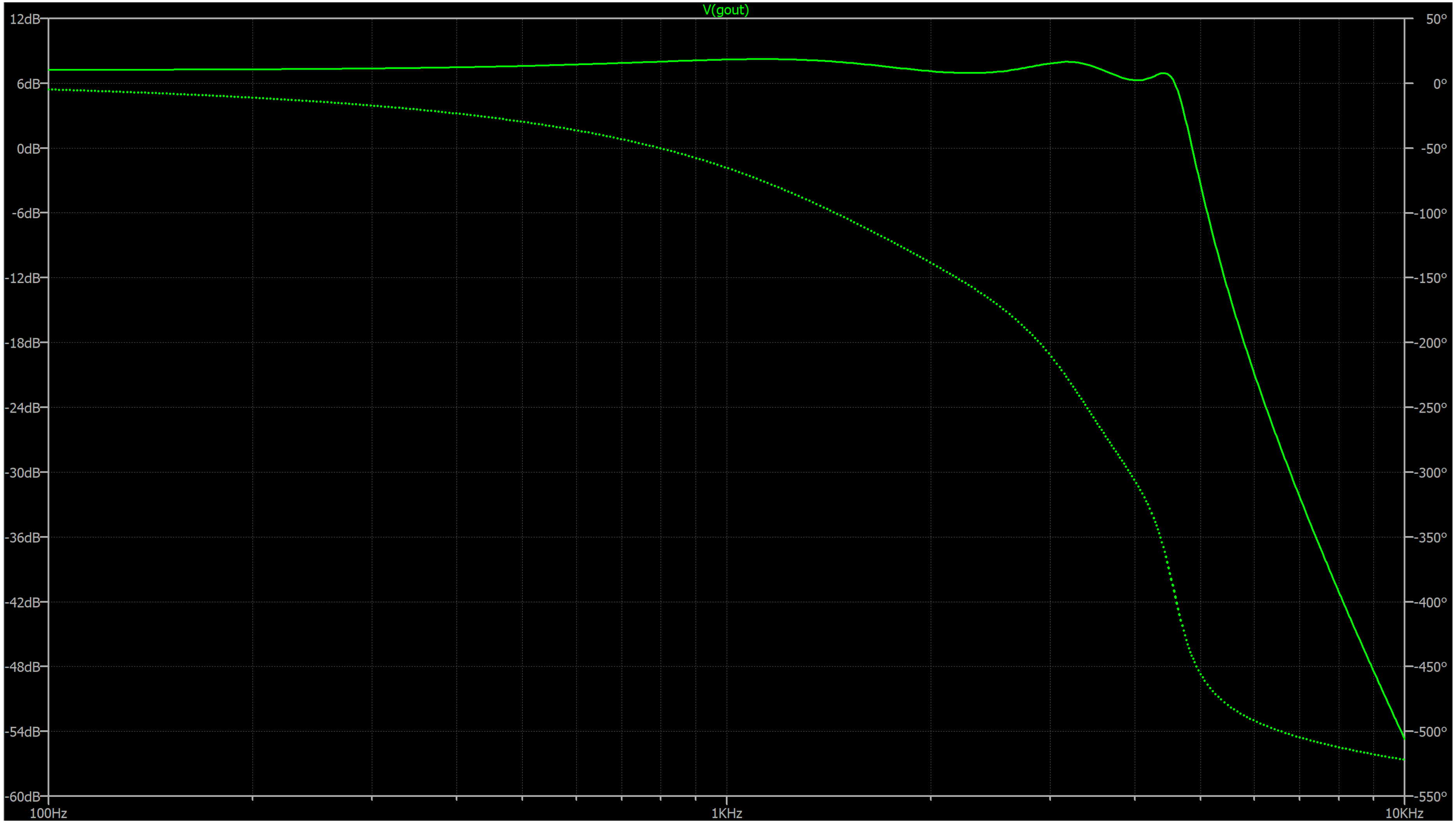


Figure 6 - Simulation 1 - Graph from (100hz – 10khz) of the gain output in AC.

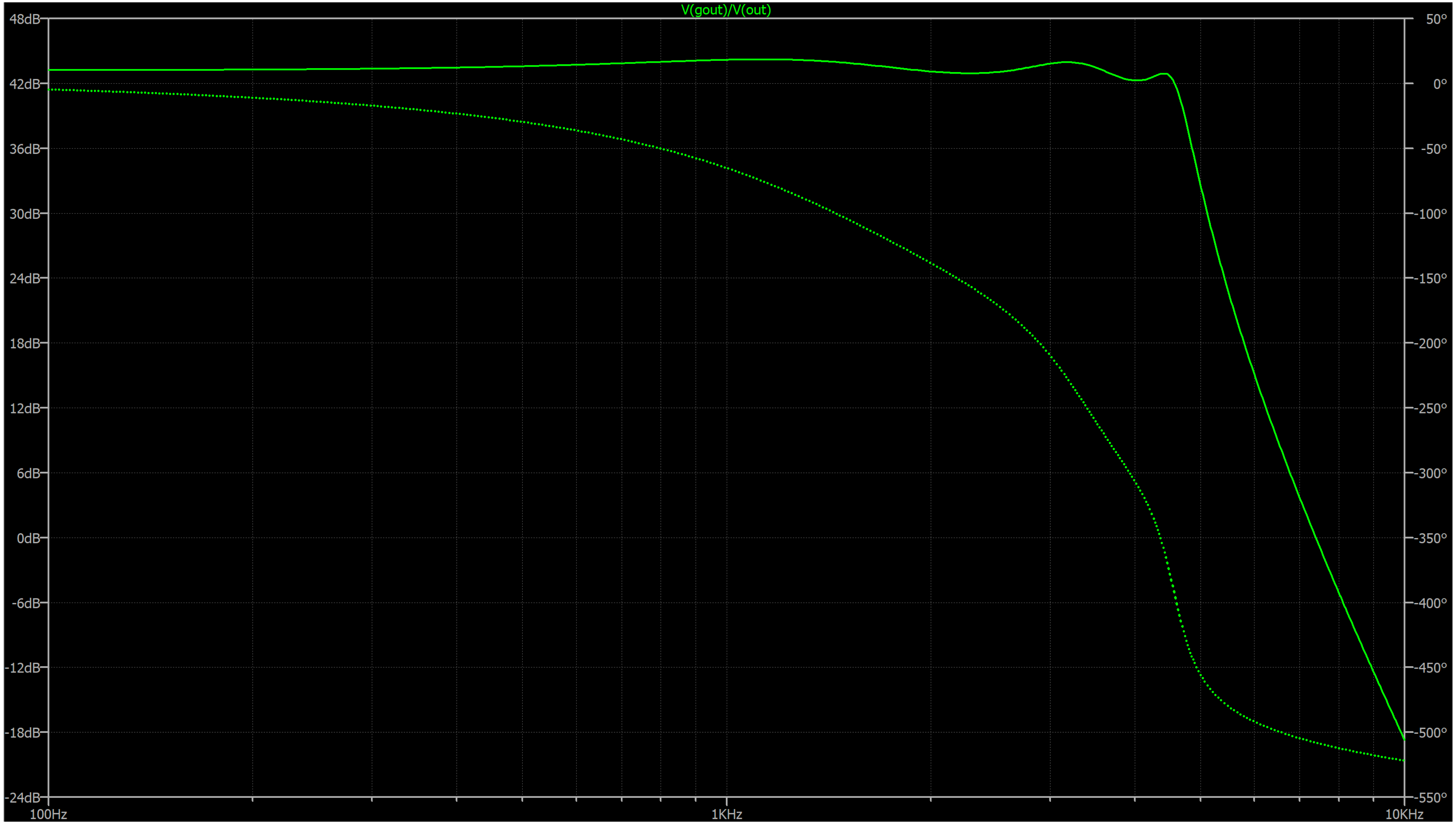


Figure 7 - Simulation 2 - Gain output/Microphone output, this graph is used to help scale the graph for any assumptions that have been made about the microphone AC source.

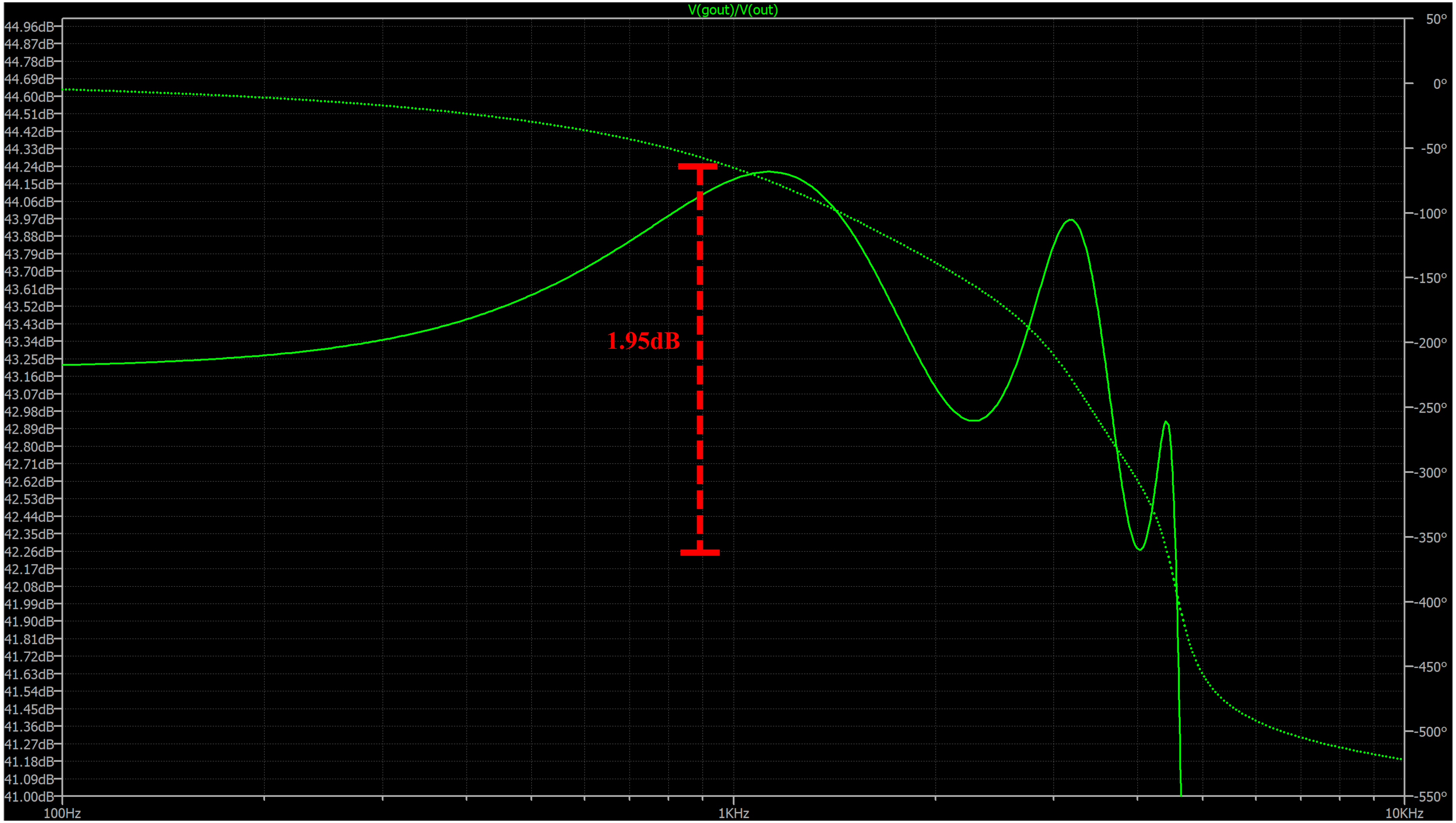


Figure 8 - Simulation 3 – As documented above the width of the passband is 1.9463185dB this is marked in red on the graph. The background researched stated that anything up to 3dB is barely perceivable by human hearing. Therefore, the goal was to keep this ripple as small as possible. In which case the value calculated does keep it under in turn allowing it to meet the specification.

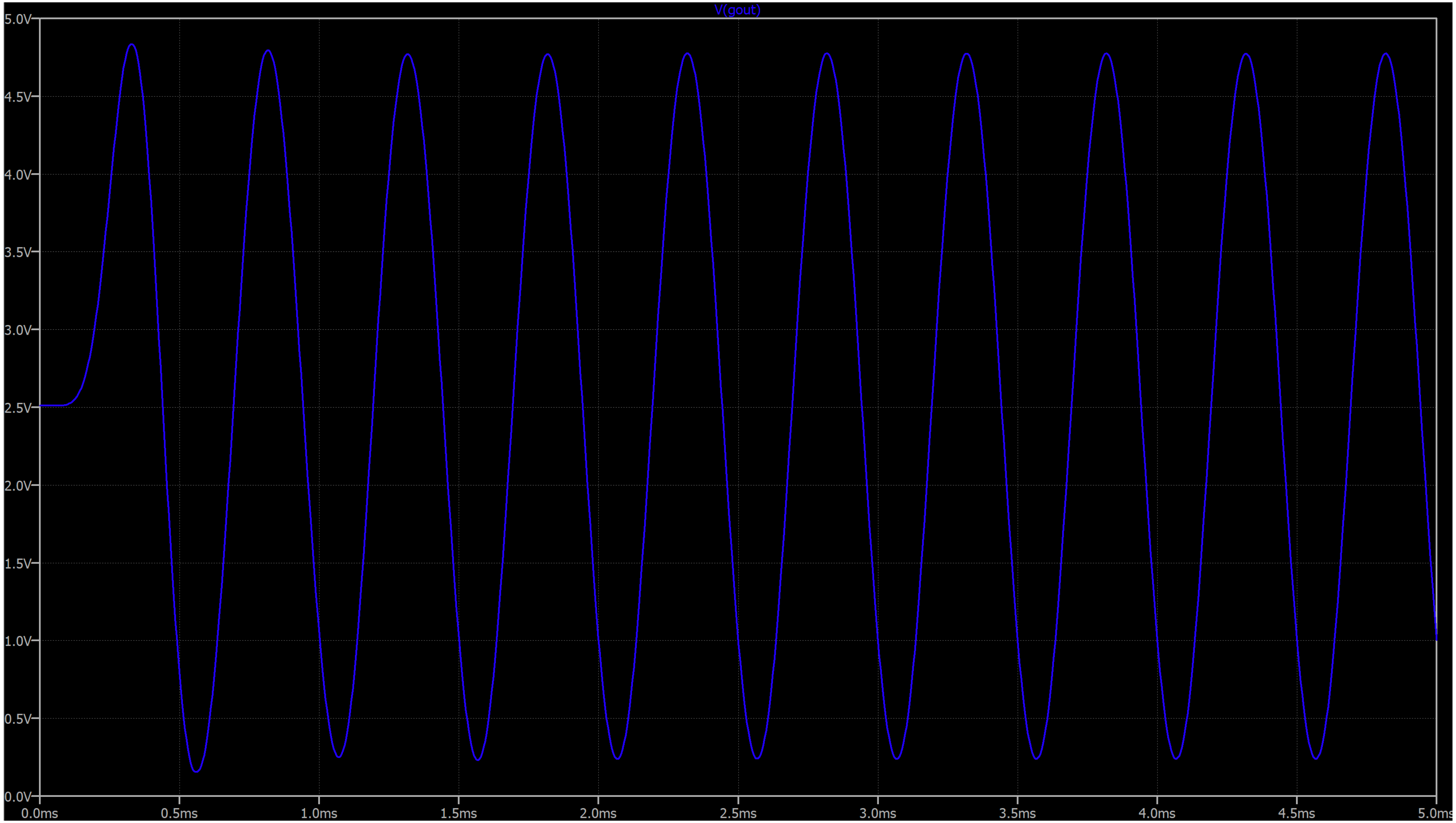


Figure 9 - Simulation 4 – This is the gain output of the circuit, it has been adjusted to be as close as possible to 5V whilst still using easily available resistors on the market. It's nearly perfectly centred at 2.5V allowing for maximum gain without clipping so that no distortion is introduced into the signal.

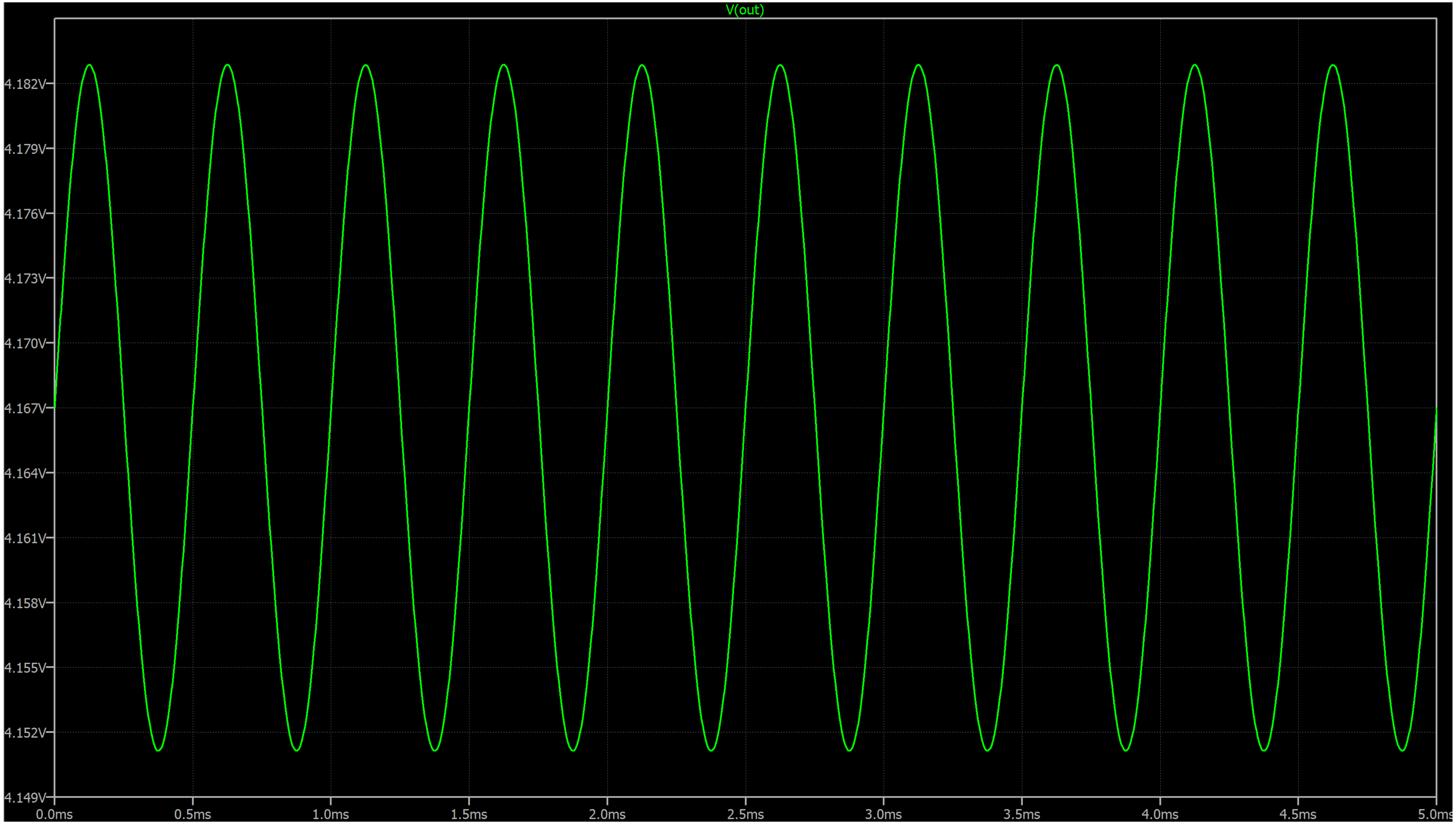


Figure 10 - Simulation 5 – Here is the initial input voltage from the microphone before it is offset. Using the assumed resistor (1k7) it is centred around the 4.167V mark.

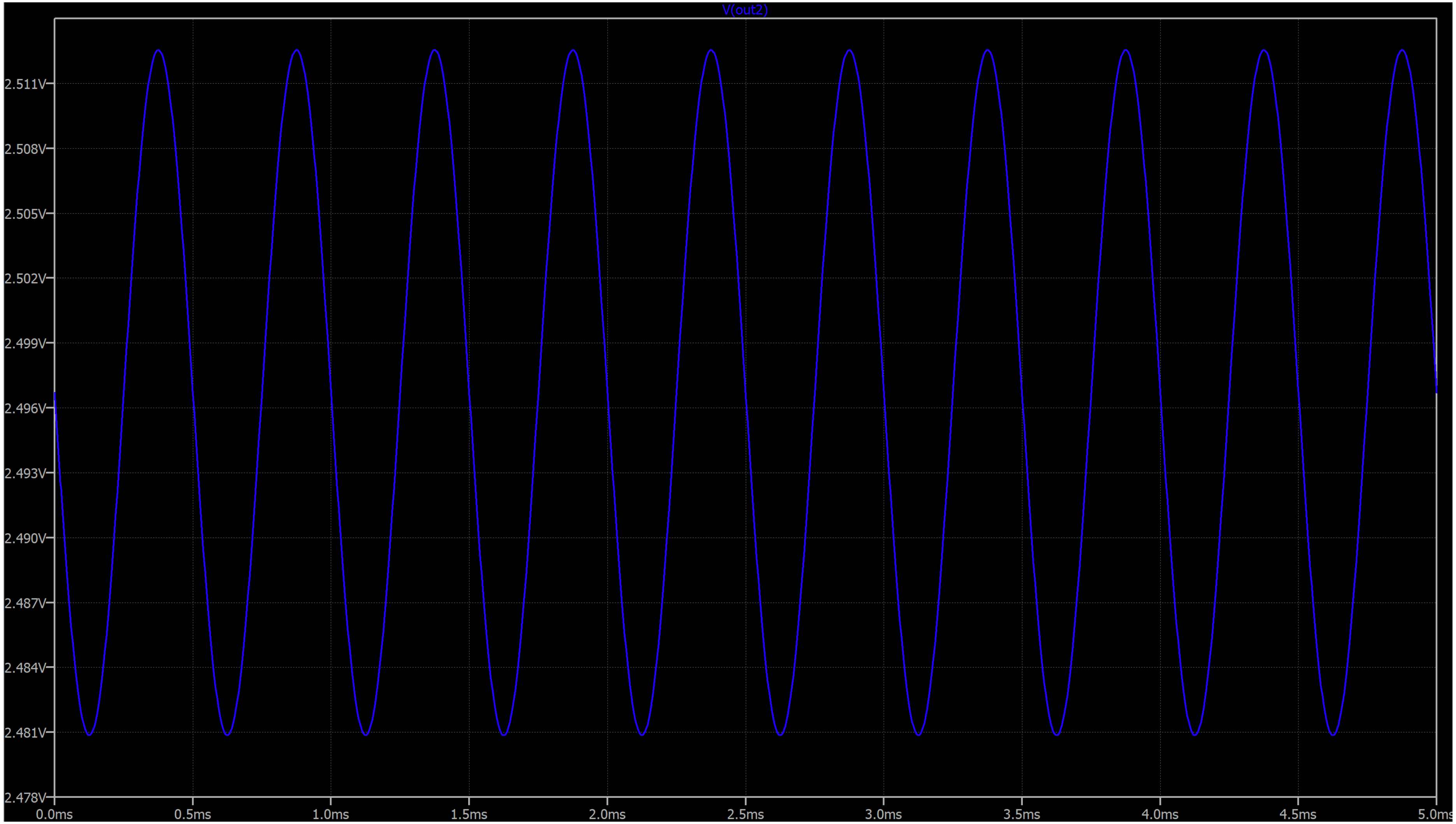


Figure 11 - Simulation 6 – Documented here is the output of the DC bias filter. No DC voltage is present here whilst featuring a wave that is centred at a voltage just under 2.5V. This is fine as it isn't too high or low for there to be clipping due to exceeding the common mode input range.

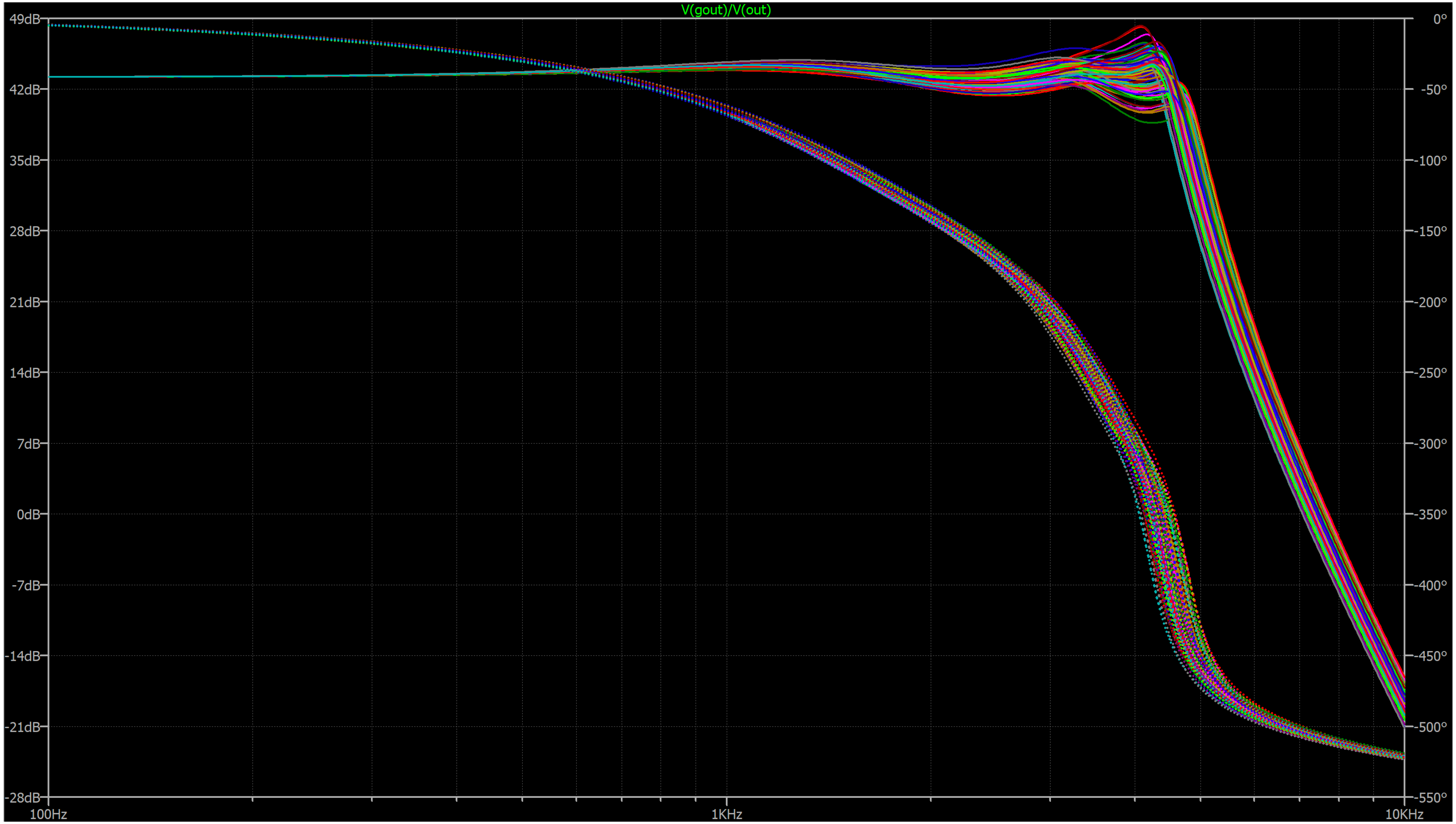


Figure 12 - Simulation 7 – The graph shown here is of the component tolerance variations. Exact values are listed in Appendix F for the change in performance due to tolerance. In comparison to the value when tolerances are not accounted for in Appendix F. It is clear that the components have a strong effect on the performance of the circuit.

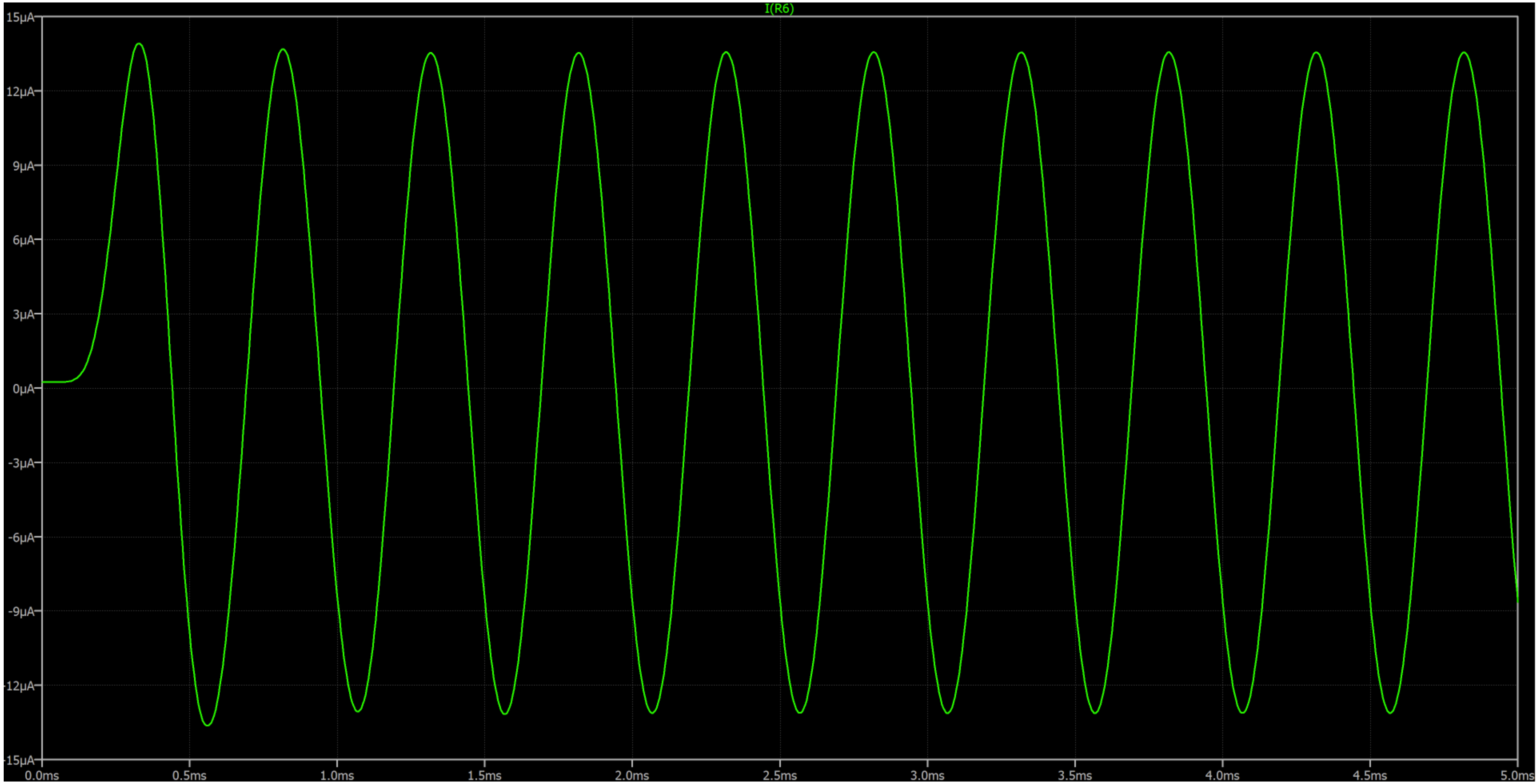


Figure 13 - Simulation 8 – This is the maximum current used in the circuit. It is well below the common mode input range and features no clipping as a result, This measurement is taken at R6 on the gain filter circuit.

4.7 Future Improvements

Having been through the design process of designing this filter, if you are looking to pick up this project or if it were to be further research and developed. Looking into a 4^{th} order filter design would be a cost-effective design choice. It would require more research to find out how component tolerances could be met and allow 3400hz. But if it can be done it will decrease the need for an extra op-amp allowing for the entire design to be made using only one TL974 unit. Additionally, an improvement to the ripple by making it less than 1dB would help reduce any possible signal distortion making for a better output signal. On the flip side, implementation in the real world and not only just simulation is recommended as it will ensure that you can make a comparison between the simulated theoretical results and the real-world results could differ. The design could require further work that has not been explored yet.

An aspect where the design could be largely improved is component tolerances. Without accounting for them the passband frequency is around 1.2dB. However, when it is accounted for the design varies largely. As highlighted in Appendix A the range goes from around 1.2 to 2.6. This could be largely improved by using a 4^{th} order filter as there are fewer components to provide an error. Using higher tolerance capable components is unlikely, as the current setup already uses the highest commonly available tolerances for both capacitors and resistors.

Another aspect that will definitely, provide better design results is prototyping a physical design. This will allow for validation and checking of the design and calculations in the real world. It would also mean hooking up the microphone and connecting it to the ADC so that it can be physically be tested to see how it realistically performs.

Looking towards the performance of the design it is possible to improve the passband ripple making it as low as possible to make it distortion-free. Since the stopband well exceeds the spec of needing to drop 48db as it drops 80. This fact means that in terms of the stopband the design is greatly overdesigned to an unnecessary level. Which means there is room for further optimisation with the filter. These values are highlighted in Appendix A for taking into account component tolerances and Appendix F for a single case.

5.0 Conclusion

To summarise the specifications set by research and the task to measure how these perform against them. The task requires the circuit to operate at 5V to interface with an Arduino ADC circuit. It should avoid exceeding the maximum current specified in the datasheet of the operational amplifier. When looking at the graphs shown below in the simulation section – (4.6 [figures 6 – 13]). It is apparent that these have been met. None of the transient graphs feature clipping meaning that the current and voltage have not exceeded the maximums set by the common-mode input ranges. In accordance with the background research, the 300 – 3400hz specifications have been met with a ripple that is less than 3dB. These are important so that not only all the frequencies can be herd. But also at minimum distortion due to keeping the ripple lower than 3dB. The design can well exceed the specification set by the microcontroller of it needing to drop at least 48dB by 7800hz. However, it well exceeds this spec which means it should be possible to further optimise the design to perform much better in the passband allowing for less distortion.

6.0 References

[1] Sound Engineering Academy. 2020. *Human Voice Frequency Range - Sound Engineering Academy*. [online] Available: https://www.seaindia.in/human-voice-frequency-range/ [Accessed 18 May 2020].

[2] Circuitcalculator.com. 2006. *The Circuitcalculator.Com Blog » Find Standard Resistor And Capacitor Values*. [online] Available: http://circuitcalculator.com/wordpress/2006/01/21/find-standard-resistor-and-capacitor-values/ [Accessed 19 May 2020].

[3] Mahbubur Rahman, S., 2001. *Multimedia Networking: Technology, Management And Applications*. [ebook] Google Books, p.271. Available: https://books.google.com.au/books?id=e3rAmuQSUXkC&pg=PA271&lpg=PA271&dq=limits+of+perception+of+analog+human+hearing+telephone&source=bl&ots=0i40I9Rt1R&sig=ACfU3U3omzyDZTr1uhas6dVwT6J4C5pzqg&hl=en&sa=X&ved=2ahUKEwjn__DstcjpAhVUmuYKHYbuD8sQ6AEwCXoECGUQAQ#v=onepage&q=limits%20of%20perception%20of%20analog%20human%20hearing%20telephone&f=false [Accessed 19 May 2020].

[4] Traunmüller, H. and Eriksson, A., 2020. *The Frequency Range Of The Voice Fundamental In The Speech Of Male And Female Adults*. [ebook] Institutionen för lingvistik, Stockholms universitet, S-106 91 Stockholm, Sweden. Available: https://www2.ling.su.se/staff/hartmut/f0_m&f.pdf [Accessed 19 May 2020].

[5] DPA. 2016. *FACTS ABOUT SPEECH INTELLIGIBILITY*. [online] Available: https://www.dpamicrophones.com/mic-university/facts-about-speech-intelligibility [Accessed 19 May 2020].

7.0 Appendix

Appendix A: Component Tolerance Evaluation

```
Circuit: * C:\Users\Joseph Haddad\Desktop\Moved from Mac\Assessment
2\assessment2circuit.asc

x1:c_cc3: both pins shorted together -- ignoring.
x3:c_cc3: both pins shorted together -- ignoring.
x4:c_cc3: both pins shorted together -- ignoring.
x5:c_cc3: both pins shorted together -- ignoring.
x6:c_cc3: both pins shorted together -- ignoring.
Direct Newton iteration for .op point succeeded.
.step run=1
.step run=2
.step run=3
.step run=4
.step run=5
.step run=6
.step run=7
.step run=8
.step run=9
.step run=10
.step run=11
.step run=12
.step run=13
.step run=14
.step run=15
.step run=16
.step run=17
.step run=18
.step run=19
.step run=20
.step run=21
.step run=22
.step run=23
.step run=24
.step run=25
.step run=26
.step run=27
.step run=28
.step run=29
.step run=30
.step run=31
.step run=32
.step run=33
.step run=34
.step run=35
.step run=36
.step run=37
.step run=38
.step run=39
.step run=40
.step run=41
.step run=42
.step run=43
.step run=44
.step run=45
```

```
.step run=46
.step run=47
.step run=48
.step run=49
.step run=50
.step run=51
.step run=52
.step run=53
.step run=54
.step run=55
.step run=56
.step run=57
.step run=58
.step run=59
.step run=60
.step run=61
.step run=62
.step run=63
.step run=64
.step run=65
.step run=66
.step run=67
.step run=68
.step run=69
.step run=70
.step run=71
.step run=72
.step run=73
.step run=74
.step run=75
.step run=76
.step run=77
.step run=78
.step run=79
.step run=80
.step run=81
.step run=82
.step run=83
.step run=84
.step run=85
.step run=86
.step run=87
.step run=88
.step run=89
.step run=90
.step run=91
.step run=92
.step run=93
.step run=94
.step run=95
.step run=96
.step run=97
.step run=98
.step run=99
.step run=100

Measurement: rp
  step       PP(mag (v(outa))) FROM   TO
      1      (1.27955dB,0∞)    100    3400
      2      (1.36436dB,0∞)    100    3400
```

3	(2.86348dB,0∞)	100	3400
4	(1.82671dB,0∞)	100	3400
5	(1.91369dB,0∞)	100	3400
6	(1.23618dB,0∞)	100	3400
7	(2.19841dB,0∞)	100	3400
8	(2.4433dB,0∞)	100	3400
9	(1.67791dB,0∞)	100	3400
10	(1.12721dB,0∞)	100	3400
11	(1.93806dB,0∞)	100	3400
12	(2.3173dB,0∞)	100	3400
13	(2.66717dB,0∞)	100	3400
14	(1.91569dB,0∞)	100	3400
15	(2.68121dB,0∞)	100	3400
16	(1.52823dB,0∞)	100	3400
17	(1.48365dB,0∞)	100	3400
18	(1.06044dB,0∞)	100	3400
19	(1.38841dB,0∞)	100	3400
20	(1.87562dB,0∞)	100	3400
21	(1.6175dB,0∞)	100	3400
22	(1.33499dB,0∞)	100	3400
23	(1.35654dB,0∞)	100	3400
24	(1.38723dB,0∞)	100	3400
25	(1.22288dB,0∞)	100	3400
26	(1.57005dB,0∞)	100	3400
27	(1.10444dB,0∞)	100	3400
28	(1.63496dB,0∞)	100	3400
29	(1.81546dB,0∞)	100	3400
30	(1.5401dB,0∞)	100	3400
31	(1.08158dB,0∞)	100	3400
32	(1.54293dB,0∞)	100	3400
33	(1.87594dB,0∞)	100	3400
34	(2.26084dB,0∞)	100	3400
35	(1.86982dB,0∞)	100	3400
36	(1.94408dB,0∞)	100	3400
37	(1.20508dB,0∞)	100	3400
38	(1.87619dB,0∞)	100	3400
39	(2.40798dB,0∞)	100	3400
40	(2.2182dB,0∞)	100	3400
41	(1.81097dB,0∞)	100	3400
42	(2.43564dB,0∞)	100	3400
43	(1.49996dB,0∞)	100	3400
44	(2.84026dB,0∞)	100	3400
45	(1.56216dB,0∞)	100	3400
46	(1.74417dB,0∞)	100	3400
47	(1.66135dB,0∞)	100	3400
48	(2.27559dB,0∞)	100	3400
49	(1.77301dB,0∞)	100	3400
50	(1.48908dB,0∞)	100	3400
51	(2.18029dB,0∞)	100	3400
52	(2.00425dB,0∞)	100	3400
53	(1.79313dB,0∞)	100	3400
54	(1.82977dB,0∞)	100	3400
55	(1.5398dB,0∞)	100	3400
56	(1.80496dB,0∞)	100	3400
57	(1.41993dB,0∞)	100	3400
58	(1.51754dB,0∞)	100	3400
59	(1.52036dB,0∞)	100	3400
60	(1.32236dB,0∞)	100	3400
61	(1.37216dB,0∞)	100	3400
62	(1.91953dB,0∞)	100	3400
63	(2.40302dB,0∞)	100	3400

64	(2.15843dB,0∞)	100	3400
65	(2.18245dB,0∞)	100	3400
66	(1.73667dB,0∞)	100	3400
67	(2.29017dB,0∞)	100	3400
68	(2.53548dB,0∞)	100	3400
69	(2.30735dB,0∞)	100	3400
70	(2.37097dB,0∞)	100	3400
71	(1.84227dB,0∞)	100	3400
72	(1.38882dB,0∞)	100	3400
73	(1.97019dB,0∞)	100	3400
74	(1.66343dB,0∞)	100	3400
75	(1.5391dB,0∞)	100	3400
76	(1.88309dB,0∞)	100	3400
77	(1.63412dB,0∞)	100	3400
78	(1.01423dB,0∞)	100	3400
79	(1.29303dB,0∞)	100	3400
80	(1.70551dB,0∞)	100	3400
81	(2.12518dB,0∞)	100	3400
82	(1.33014dB,0∞)	100	3400
83	(2.60177dB,0∞)	100	3400
84	(1.35988dB,0∞)	100	3400
85	(1.13267dB,0∞)	100	3400
86	(1.83347dB,0∞)	100	3400
87	(1.72687dB,0∞)	100	3400
88	(2.26226dB,0∞)	100	3400
89	(2.16569dB,0∞)	100	3400
90	(1.92498dB,0∞)	100	3400
91	(1.53919dB,0∞)	100	3400
92	(1.59553dB,0∞)	100	3400
93	(1.18995dB,0∞)	100	3400
94	(1.85018dB,0∞)	100	3400
95	(1.75968dB,0∞)	100	3400
96	(1.79139dB,0∞)	100	3400
97	(1.18805dB,0∞)	100	3400
98	(1.75334dB,0∞)	100	3400
99	(1.6376dB,0∞)	100	3400
100	(2.0337dB,0∞)	100	3400

Measurement: amin

step	MAX(mag (v(outa)))	FROM	TO
1	(-82.6984dB,0∞)	7800	10000
2	(-83.5411dB,0∞)	7800	10000
3	(-82.413dB,0∞)	7800	10000
4	(-85.4853dB,0∞)	7800	10000
5	(-82.3509dB,0∞)	7800	10000
6	(-81.1309dB,0∞)	7800	10000
7	(-82.4628dB,0∞)	7800	10000
8	(-82.6812dB,0∞)	7800	10000
9	(-84.0356dB,0∞)	7800	10000
10	(-83.8931dB,0∞)	7800	10000
11	(-81.5466dB,0∞)	7800	10000
12	(-83.0511dB,0∞)	7800	10000
13	(-82.6864dB,0∞)	7800	10000
14	(-84.0467dB,0∞)	7800	10000
15	(-83.3974dB,0∞)	7800	10000
16	(-82.1808dB,0∞)	7800	10000
17	(-81.1568dB,0∞)	7800	10000
18	(-82.8921dB,0∞)	7800	10000
19	(-83.8983dB,0∞)	7800	10000
20	(-83.0358dB,0∞)	7800	10000
21	(-83.527dB,0∞)	7800	10000

22	(-82.6877dB,0∞)	7800	10000
23	(-83.636dB,0∞)	7800	10000
24	(-82.513dB,0∞)	7800	10000
25	(-80.0465dB,0∞)	7800	10000
26	(-85.3285dB,0∞)	7800	10000
27	(-82.1489dB,0∞)	7800	10000
28	(-82.0168dB,0∞)	7800	10000
29	(-83.5405dB,0∞)	7800	10000
30	(-82.5762dB,0∞)	7800	10000
31	(-82.6611dB,0∞)	7800	10000
32	(-82.7603dB,0∞)	7800	10000
33	(-83.145dB,0∞)	7800	10000
34	(-83.2135dB,0∞)	7800	10000
35	(-83.1984dB,0∞)	7800	10000
36	(-80.8058dB,0∞)	7800	10000
37	(-82.8885dB,0∞)	7800	10000
38	(-83.5043dB,0∞)	7800	10000
39	(-80.8171dB,0∞)	7800	10000
40	(-83.4865dB,0∞)	7800	10000
41	(-80.0968dB,0∞)	7800	10000
42	(-83.6875dB,0∞)	7800	10000
43	(-82.7545dB,0∞)	7800	10000
44	(-83.3061dB,0∞)	7800	10000
45	(-83.1932dB,0∞)	7800	10000
46	(-82.3115dB,0∞)	7800	10000
47	(-82.5819dB,0∞)	7800	10000
48	(-80.7314dB,0∞)	7800	10000
49	(-81.8343dB,0∞)	7800	10000
50	(-81.1966dB,0∞)	7800	10000
51	(-79.8078dB,0∞)	7800	10000
52	(-83.485dB,0∞)	7800	10000
53	(-81.1716dB,0∞)	7800	10000
54	(-82.9727dB,0∞)	7800	10000
55	(-82.0245dB,0∞)	7800	10000
56	(-80.9479dB,0∞)	7800	10000
57	(-81.776dB,0∞)	7800	10000
58	(-85.038dB,0∞)	7800	10000
59	(-84.4408dB,0∞)	7800	10000
60	(-81.7725dB,0∞)	7800	10000
61	(-81.9435dB,0∞)	7800	10000
62	(-83.2902dB,0∞)	7800	10000
63	(-81.6424dB,0∞)	7800	10000
64	(-83.1185dB,0∞)	7800	10000
65	(-82.6493dB,0∞)	7800	10000
66	(-83.9321dB,0∞)	7800	10000
67	(-84.0424dB,0∞)	7800	10000
68	(-83.0748dB,0∞)	7800	10000
69	(-81.4308dB,0∞)	7800	10000
70	(-82.094dB,0∞)	7800	10000
71	(-82.953dB,0∞)	7800	10000
72	(-80.4272dB,0∞)	7800	10000
73	(-81.5207dB,0∞)	7800	10000
74	(-80.7796dB,0∞)	7800	10000
75	(-84.0131dB,0∞)	7800	10000
76	(-84.0065dB,0∞)	7800	10000
77	(-82.7168dB,0∞)	7800	10000
78	(-80.7164dB,0∞)	7800	10000
79	(-80.8477dB,0∞)	7800	10000
80	(-83.4706dB,0∞)	7800	10000
81	(-83.6427dB,0∞)	7800	10000
82	(-81.822dB,0∞)	7800	10000

83	(-83.4577dB,0∞)	7800	10000
84	(-84.2444dB,0∞)	7800	10000
85	(-81.5654dB,0∞)	7800	10000
86	(-83.405dB,0∞)	7800	10000
87	(-83.0503dB,0∞)	7800	10000
88	(-80.9536dB,0∞)	7800	10000
89	(-83.0596dB,0∞)	7800	10000
90	(-85.4969dB,0∞)	7800	10000
91	(-80.9794dB,0∞)	7800	10000
92	(-81.8436dB,0∞)	7800	10000
93	(-80.9699dB,0∞)	7800	10000
94	(-83.4022dB,0∞)	7800	10000
95	(-81.188dB,0∞)	7800	10000
96	(-82.8032dB,0∞)	7800	10000
97	(-84.5772dB,0∞)	7800	10000
98	(-82.4697dB,0∞)	7800	10000
99	(-83.6439dB,0∞)	7800	10000
100	(-83.9959dB,0∞)	7800	10000

```
Date: Sat May 23 14:32:48 2020
Total elapsed time: 26.045 seconds.

tnom = 27
temp = 27
method = trap
totiter = 19
traniter = 0
tranpoints = 0
accept = 0
rejected = 0
matrix size = 593
fillins = 645
Matrix Compiler1:    3222 opcodes
Matrix Compiler2: 63.28 KB object code size
```

Appendix B - (STAGE A – USING E24 SERIES)

	A	B	C	D	E	F
1	Q	8.27				
2	wn	28088				
3	C1	3.3	nF			
4						
5		m	n	C2 (nF)	R1 (kΩ)	R2 (kΩ)
6	0	1.00	273.46	902.42	0.65	0.65
7	1	1.10	274.09	904.49	0.62	0.68
8	2	1.21	275.98	910.75	0.59	0.71
9	3	1.33	279.16	921.23	0.56	0.75
10	4	1.47	283.65	936.05	0.53	0.78
11	5	1.62	289.50	955.33	0.50	0.81
12	6	1.78	296.75	979.26	0.47	0.84
13	7	1.96	305.47	1008.05	0.44	0.86
14	8	2.15	315.75	1041.97	0.41	0.89
15	9	2.37	327.68	1081.33	0.39	0.92
16	10	2.61	341.36	1126.50	0.36	0.94
17	11	2.87	356.94	1177.89	0.34	0.97
18	12	3.16	374.54	1235.97	0.31	0.99
19	13	3.48	394.33	1301.28	0.29	1.01
20	14	3.83	416.49	1374.42	0.27	1.03
21	15	4.22	441.23	1456.07	0.25	1.05
22	16	4.64	468.78	1546.97	0.23	1.07
23	17	5.11	499.38	1647.97	0.21	1.09
24	18	5.62	533.33	1759.99	0.20	1.11
25	19	6.19	570.93	1884.07	0.18	1.12
26	20	6.81	612.53	2021.34	0.17	1.14
27	21	7.50	658.51	2173.08	0.15	1.15
28	22	8.25	709.30	2340.68	0.14	1.16
29	23	9.09	765.36	2525.69	0.13	1.18
30	24	10.00	827.21	2729.81	0.12	1.19

Appendix C - (STAGE B – USING E24 SERIES)

	A	B	C	D	E	F
1	Q	2.27				
2	wn	2.10E+04				
3	C1	3.3	nF			
4						
5		m	n	C2 (nF)	R1 (kΩ)	R2 (kΩ)
6	0	1.00	20.56	67.86	3.18	3.18
7	1	1.10	20.61	68.01	3.02	3.33
8	2	1.21	20.75	68.48	2.87	3.48
9	3	1.33	20.99	69.27	2.72	3.63
10	4	1.47	21.33	70.39	2.57	3.78
11	5	1.62	21.77	71.84	2.43	3.92
12	6	1.78	22.31	73.63	2.29	4.07
13	7	1.96	22.97	75.80	2.15	4.20
14	8	2.15	23.74	78.35	2.01	4.34
15	9	2.37	24.64	81.31	1.88	4.47
16	10	2.61	25.67	84.71	1.76	4.59
17	11	2.87	26.84	88.57	1.64	4.71
18	12	3.16	28.16	92.94	1.53	4.83
19	13	3.48	29.65	97.85	1.42	4.93
20	14	3.83	31.32	103.35	1.31	5.04
21	15	4.22	33.18	109.49	1.22	5.13
22	16	4.64	35.25	116.32	1.13	5.23
23	17	5.11	37.55	123.92	1.04	5.31
24	18	5.62	40.10	132.34	0.96	5.39
25	19	6.19	42.93	141.67	0.88	5.47
26	20	6.81	46.06	151.99	0.81	5.54
27	21	7.50	49.52	163.40	0.75	5.60
28	22	8.25	53.34	176.01	0.69	5.67
29	23	9.09	57.55	189.92	0.63	5.72
30	24	10.00	62.20	205.27	0.58	5.77

Appendix D - (STAGE C – USING E24 SERIES)

	A	B	C	D	E	F
1	Q	0.78				
2	wn	9829.6				
3	C1	3.3	nF			
4						
5		m	n	C2 (nF)	R1 (kΩ)	R2 (kΩ)
6	0	1.00	2.40	7.93	19.88	19.88
7	1	1.10	2.41	7.95	18.93	20.83
8	2	1.21	2.43	8.01	17.98	21.78
9	3	1.33	2.45	8.10	17.04	22.72
10	4	1.47	2.49	8.23	16.11	23.65
11	5	1.62	2.55	8.40	15.20	24.56
12	6	1.78	2.61	8.61	14.31	25.45
13	7	1.96	2.69	8.86	13.45	26.32
14	8	2.15	2.78	9.16	12.61	27.16
15	9	2.37	2.88	9.51	11.79	27.97
16	10	2.61	3.00	9.90	11.01	28.75
17	11	2.87	3.14	10.36	10.27	29.50
18	12	3.16	3.29	10.87	9.55	30.21
19	13	3.48	3.47	11.44	8.87	30.89
20	14	3.83	3.66	12.08	8.23	31.53
21	15	4.22	3.88	12.80	7.62	32.14
22	16	4.64	4.12	13.60	7.05	32.71
23	17	5.11	4.39	14.49	6.51	33.25
24	18	5.62	4.69	15.47	6.00	33.76
25	19	6.19	5.02	16.57	5.53	34.23
26	20	6.81	5.39	17.77	5.09	34.67
27	21	7.50	5.79	19.11	4.68	35.08
28	22	8.25	6.24	20.58	4.30	35.47
29	23	9.09	6.73	22.21	3.94	35.82
30	24	10.00	7.27	24.00	3.61	36.15

Appendix E - (MATLAB)

```
fsamp = 15.625e3;
fs = fsamp / 2;
fp = 3.4e3;      % Frequency Pass @human analogue voice frequency range.
ws = 2 * pi * fs;
wp = 2 * pi * fp;
Amin = 20*log10(1/2^8);      % 48 db stopband attenuation
Amax = 0.3;                   % Passband Attenuation;
```

Calculate the Filter Order

```
% Butterworth
[n_butt, wn_butt] = buttord (wp,ws,Amax,Amin, 's')
% Chebyshev
[n_cheb, wn_cheb] = cheb1ord (wp,ws,Amax,Amin, 's')


% Will select Chebyshev filter as its a 6th order filter using less op
% amps then the Butterworth.
```

Transfer Function

```
[b,a] = cheby1(n_cheb, Amax, wn_cheb, 'low', 's');
H = tf(b,a)     % Transfer Function

% figure 1
h = bodeplot(H);
setoptions(h,'FreqUnits','Hz');
%setoptions(h, 'FreqUnits', 'Hz', 'PhaseVisible', 'off');
grid on;


[z, p, k] = tf2zpk (b,a)


% Since it is a 6th order filter it will have 3 filters
% 3 second order filters.


% Stage A Design 2nd Order Filter


a1A = -p(1) - p(2);
a2A = p(1) * p(2);


wnA = sqrt(a2A)
QA = wnA/a1A


nA = (2*QA).^2
```

```
% need to make these preferred values.

C1A = (10/fp) * 0.000001 % Capacitors
C2A = nA * C1A

R1A = 1 / (wnA * C1A * sqrt(nA))   % Resistors
R2A = R1A




%Stage B design 2nd Order Filter

a1B = -p(3) - p(4);
a2B = p(3) * p(4);

wnB = sqrt(a2B)
QB = wnB/a1B

nB = (2*QB).^2

% need to make these preferred values.
C1B = (10/fp) * 0.000001 % Capacitors
C2B = nB * C1B

R1B = 1 / (wnB * C1B * sqrt(nB))   % Resistors
R2B = R1B



%Stage C design 2nd Order Filter

a1C = -p(5) - p(6);
a2C = p(5) * p(6);

wnC = sqrt(a2C)
QC = wnC/a1C

nC = (2*QC).^2

% need to make these preferred values.
C1C = (10/fp) * 0.000001 % Capacitors
C2C = nC * C1C
```

```
R1C = 1 / (wnC * C1C * sqrt(nC))   % Resistors
R2C = R1C
```

Checking The Generated Transfer Function with Preferred Values

```
KA = 1;
numA = [0 0 KA/(R1A*R2A*C1A*C2A)];
denA = [1 (1/(R1A*C2A) + 1/(R2A*C2A) + (1-KA)/(R2A*C1A)) 1/(R1A*R2A*C1A*C2A)];
HtestA = tf(numA,denA);



KB = 1;
numB = [0 0 KB/(R1B*R2B*C1B*C2B)];
denB = [1 (1/(R1B*C2B) + 1/(R2B*C2B) + (1-KB)/(R2B*C1B)) 1/(R1B*R2B*C1B*C2B)];
HtestB = tf(numB,denB);


KC = 1;
numC = [0 0 KC/(R1C*R2C*C1C*C2C)];
denC = [1 (1/(R1C*C2C) + 1/(R2C*C2C) + (1-KC)/(R2C*C1C)) 1/(R1C*R2C*C1C*C2C)];
HtestC = tf(numC,denC);



Htest = HtestA * HtestB * HtestC


% figure 2
h = bodeplot(Htest);
setoptions(h,'FreqUnits','Hz');
%setoptions(h, 'FreqUnits', 'Hz', 'PhaseVisible', 'off');
grid on;
```

Appendix F – (Normal Passband and Stopband Analysis)

```
Circuit: * C:\Users\Joseph Haddad\Desktop\Moved from Mac\Assessment
2\assessment2circuit.asc

x1:c_cc3: both pins shorted together -- ignoring.
x3:c_cc3: both pins shorted together -- ignoring.
x4:c_cc3: both pins shorted together -- ignoring.
x5:c_cc3: both pins shorted together -- ignoring.
x6:c_cc3: both pins shorted together -- ignoring.
Direct Newton iteration for .op point succeeded.

rp: PP(mag (v(outa)))=(1.27955dB,0∞) FROM 100 TO 3400
amin: MAX(mag (v(outa)))=(-82.6984dB,0∞) FROM 7800 TO 10000


Date: Sun May 24 09:13:49 2020
Total elapsed time: 0.361 seconds.

tnom = 27
temp = 27
method = trap
totiter = 19
traniter = 0
tranpoints = 0
accept = 0
rejected = 0
matrix size = 593
fillins = 645
solver = Alternate
Matrix Compiler1:    3216 opcodes
Matrix Compiler2: 63.31 KB object code size
```