Analogue Signal Conditioning for Digital Voice Recorders: Design Challenges and Insights

Zackariya Mohammed Justin Taylor – n11592931

Submitted on 03/06/2024.

Executive Summary

This report documents the design process for an analogue signal conditioning system for a digital voice recorder (DVR). The DVR is designed to be interfaced with a pre-built microcontroller development board equipped with an ADC, memory card, pushbuttons, and LEDs. It will also interface with telephone lines, where the user can switch between an MCU-interfaced or a telephone-interfaced system.

The design encompasses three main stages: DC offset, anti-aliasing filter, and amplification. The design goals are to optimize passband ripple and attenuation while keeping the passband as wide as possible to retain audio information. The specifications were determined mathematically and qualitatively.

The circuit topology is the result of an iterative design process involving three cascaded Sallen-Key low-pass filters, creating a 6th order filter. The components for the filters were determined through characteristic equations and MATLAB tools. Once preferred values were chosen, LTspice was used to simulate the frequency response of the filters to determine their performance. This was followed by developing a prototype breadboard circuit, which was then tested using oscilloscope bode plots and transient analysis.

Regrettably, the filter did not meet specifications and started attenuating frequencies at lower frequencies than anticipated. The cause of this premature attenuation could be attributed to factors such as component tolerances, preferred value rounding errors, non-ideal behavior, or op-amp characteristics. The actual reason behind the design's flaw was not addressed because it was too technically advanced and outside of the project's scope. However, the margin of error was under 10% and was deemed acceptable due to the non-critical applications of the DVR. This flaw, though not critical, presents an opportunity for further investigation and potential improvements in future design iterations.

In conclusion, while the designed analogue signal conditioning system for the digital voice recorder encountered challenges in meeting all specifications, the process from design to prototyping provided valuable insights into the complexities of analogue circuitry. Despite the identified issues with filter performance, the margin of error remained within acceptable limits. This experience underscores the importance of thorough testing and iterative refinement in achieving desired outcomes in analogue signal processing systems. These lessons will inform future design endeavors, ensuring continued innovation and improvement in similar projects.

Table of Contents

1.0 Introduction

2.0 Design and Testing

- 2.1 Project Description
- 2.2 Scope
- 2.3 Design Goals and Specifications
  - 2.3.1 Filter Type and Passband Ripple
  - 2.3.2 Filter Frequency and Attenuation
  - 2.3.3 DC Offset and Amplification
- 2.4 Methodology
- 2.5 Technical Design
  - 2.5.1 Design Process and Details
  - 2.5.2 Principles of Operation and Schematics
- 2.6 Implementation and Testing Procedure
- 2.7 Experimental Design Review
- 2.8 Future Work

3.0 Conclusion

4.0 References

5.0 Appendix

List of Tables and Figures

Figure 2.1.1 General DVR circuit design – Page 6

Figure 2.1.2 Signal conditioning and MCU application diagram – Page 6

Figure 2.3.1 Bode plot showing filter specifications – Page 7

Figure 2.3.2 Type 1 vs Type 2 Chebyshev bode plot [22] – Page 8

Figure 2.3.3 Additive noise process (from EGB242 Assessment 1 task sheet) – Page 8

Table 2.3.1 Human hearing perception to sound changes [2] – Page 8

Figure 2.3.4 Butterworth frequency response order comparison [23] – Page 9

Figure 2.3.5 Oscilloscope capture of microphone before amplification – Page 11

Figure 2.5.1 Recommended microphone circuit topology – Page 13

Figure 2.5.2 DC offset and buffer circuit – Page 13

Figure 2.5.3 DC offset and buffer in LTspice – Page 14

Figure 2.5.4 Transient response of DC offset circuit – Page 14

Figure 2.5.5 Step response of anti-aliasing filter – Page 15

Figure 2.5.6 Poles plot of anti-aliasing filter – Page 16

Figure 2.5.7 Lowpass Sallen-Key filter topology – Page 16

Figure 2.5.8 MATLAB bode and phase plot of MCU-interfaced anti-aliasing filter – Page 18

Figure 2.5.9 MATLAB bode and phase plot of telephone-interfaced anti-aliasing filter – Page 18

Figure 2.5.10 LTspice simulation of bode plots – Page 19

Figure 2.5.11 Circuit topology of DC offset and anti-aliasing filters – Page 19

Figure 2.5.12 Inverting amplifier topology – Page 20

Figure 2.5.13 DVR amplifier circuit topology – Page 20

Figure 2.5.14 Block diagram representing DVR operation – Page 21

Figure 2.5.15 Internals of an operational amplifier – Page 21

Figure 2.5.16 Complete analogue conditioning circuit topology – Page 22

Figure 2.5.17 Transient analysis showing common-mode input voltage clipping – Page 22

Figure 2.6.1 Breadboard DVR prototype with centimeters scale – Page 23

Figure 2.6.2 Breadboard prototype with annotations – Page 23

Figure 2.6.3 Prototype being testing with oscilloscope probes connected – Page 24

Figure 2.7.1 Oscilloscope bode plot capture of MCU-interfaced filter – Page 26

Figure 2.7.2 Oscilloscope bode plot capture of telephone-interfaced filter – Page 26

Figure 2.7.3 Oscilloscope capture show DC offset and amplification – Page 27

1.0 Introduction

This report documents the design process behind a DVR analogue conditioning circuit. The design was developed in adherence to sound engineering principles, focusing on optimizing the design. The circuit processes an electret microphone's input. This report describes designing, implementing, and evaluating the circuit.

The circuit's output is received by a QUT pre-designed development board, a key component of the digital voice recorder. This board houses an analogue-to-digital converter (ADC), a memory card for data storage, interactive pushbuttons for user control, and LEDs for visual feedback. The proposed design solutions were rigorously evaluated through breadboard prototypes and LTspice simulations, ensuring the design met sound engineering expectations and principles.

2.0 Design and Testing

2.1 Project Description

This project aims to develop a digital voice recorder that can interface with telephone lines and the MCU, which have different sampling rates. The design can be broken down into three sections: DC offset, anti-aliasing filters (for MCU and telephone line usage), and amplification. Each section must meet requirements determined by research and testing conducted through LTspice and MATLAB. The justified component selection was then progressed to assemble a prototype, which was compared against theoretical and simulation results to determine the accuracy of the proposed solution. Figure 2.1.1 shows a general design of the circuit, which will be expanded upon.

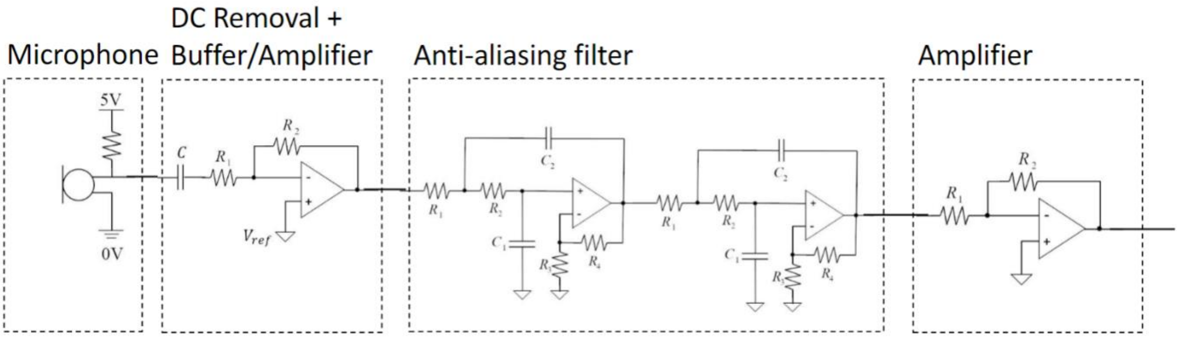


Figure 2.1.1 General analogue conditioning circuit design

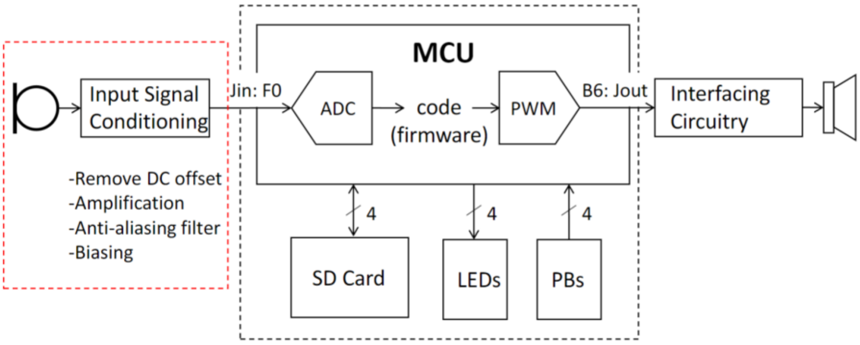


Figure 2.1.2 Signal conditioning and MCU application diagram

The DVR solution interfaces with an MCU provided with firmware and hardware (Figure 2.1.2). The DVR must be designed to abide by the MCU's signal input requirements and telephony sampling rate requirements.

2.2 Scope

The scope of this project outlines the design process of a digital voice recorder. The voice recorder will be evaluated through simulations in MATLAB and LTspice and practical measurements using oscilloscopes. The filter will be designed to meet specifications with an allowance for error due to the device's non-critical nature. Although the filter's specifications should not adversely impact performance.

The project's boundaries and limitations include time constraints, component availability, and technological difficulties. The project is scheduled to be completed within 7 weeks. Component values are limited to the E12 series, which are readily available; obtaining components from other series may be time-consuming and exacerbate time constraints. Addressing performance issues may be technically challenging and will be tolerated within a moderate margin of error.

The following assumptions were made to simplify and streamline the project's progress:

- The simulation tools (MATLAB and LTspice) accurately model real-world behaviour.
- The availability of E12 series components remains consistent throughout the project.
- Technical difficulties will not introduce significant delays beyond the 8-week timeline.
- The performance requirements of the digital voice recorder are flexible enough to allow for a 10% margin of error.
- The equipment (oscilloscopes and other measurement tools) functions correctly and provides accurate data.

2.3 Design goals & Specifications

The mandatory specifications of the system (Figure 2.3.1) to be determined are passband frequency, stopband frequency, maximum passband ripple, minimum attenuation at the stopband frequency, filter type, and filter order. For the DVR to be interfaced with a telephone line, the system must be able to switch between anti-aliasing filters to ensure the widest band of frequencies is retained while avoiding aliasing. These specifications will be determined to balance high sound quality with low complexity.

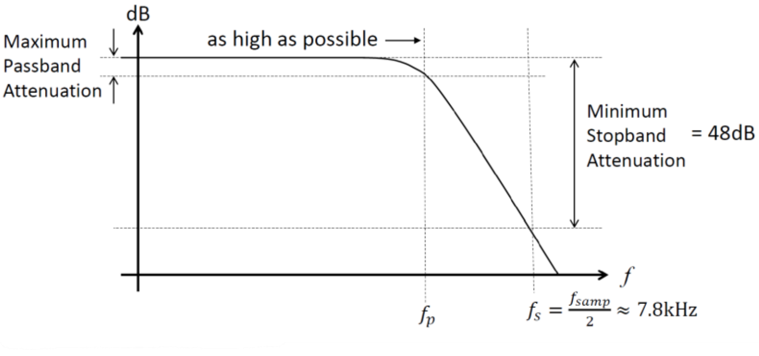

dB
Maximum Passband Attenuation
as high as possible ⟶
Minimum Stopband Attenuation
= 48dB
f
f_p
$f_s = \frac{f_{samp}}{2} \approx 7.8\text{kHz}$

Figure 2.3.1 Bode plot showing filter specifications

An optional design target is to minimize the number of op-amp ICs used to a reasonable extent. This will be done for cost-effectiveness and compactness.

2.3.1 Filter type and Passband Ripple

The two common filter topologies include the Chebyshev and Butterworth filters. Each one of these filters has distinct properties. Simplicity and steep roll-off response are essential for choosing a suitable filter topology.

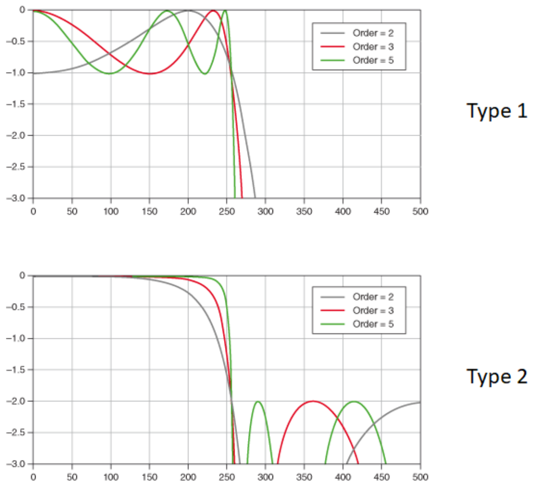


Figure 2.3.2 Type 1 vs Type 2 Chebyshev bode plot [22]

There are two types of Chebyshev filters, types 1 and 2 (Figure 2.3.2). Type 1 has a ripple response in its passband, whereas type 2 has a ripple response in its stopband [1]. Since filters aim to prevent aliasing, the stopband should be attenuated sufficiently; thus, a type 2 Chebyshev topology is deemed unsuitable due to its unwanted stopband ripple.

The roll-off steepness and filter order are proportional to the magnitude of the ripple in the passband. Passband ripple will cause additive noise to the audio signal, leading to audible distortion. Figure 2.3.3 depicts the additive noise process, where $n(t)$ is the noise signal caused by the passband ripple. By determining the maximum passband ripple, the order and roll-off steepness of the type 1 Chebyshev filter can be established.

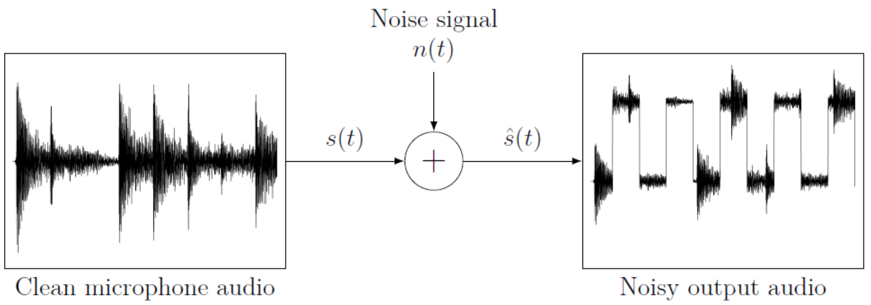


Figure 2.3.3 Additive noise process (from EGB242 Assessment 1 task sheet)

Difference in sound level (dB)	**Change in auditory perception**
1-2	Not perceptible
3-4	Barely perceptible
5-9	Audible difference
10+	Sensation is scaled significantly

Table 2.3.1 Human hearing perception to sound changes [2]

Table 2.3.1 shows the difference in sound levels compared to the change in perception. Therefore, $A_{max} = 4dB$ is the maximum allowable ripple in the passband that will not cause significant distortion to the microphone audio.

The alternative filter topology is a Butterworth filter. A Butterworth filter has no ripple in the passband, at the cost of a slower roll-off (Figure 2.3.4), in comparison to a Chebyshev filter. To reach a roll-off rate comparable to a Chebyshev filter, a higher-order Butterworth filter is required, which may not be suitable to the design requirements of compactness.

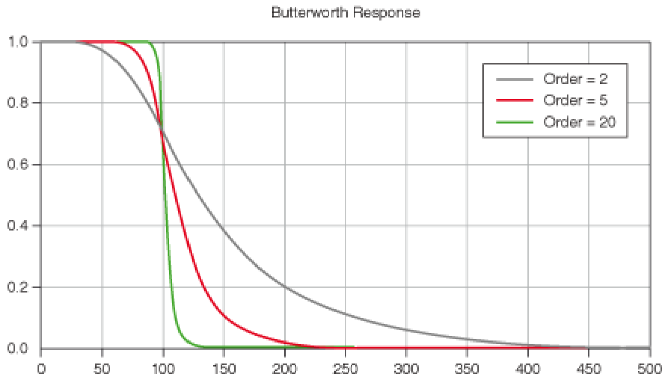


Figure 2.3.4 Butterworth frequency response order comparison [23]

With a ±4dB passband ripple tolerance, a Chebyshev filter will be used for the anti-aliasing as an nth order Butterworth filter will have a slower roll-off rate than an nth order Chebyshev filter. A Chebyshev topology will allow for the edge of the passband to be maximized, while meeting stopband attenuation requirements and minimizing the number of op amps used.

2.3.2 Filter Frequency and Attenuation

The purpose of filtering is so that the audio can be interfaced with the MCU and telephone lines, while avoiding aliasing. Two frequencies will need to be determined, the bandpass frequency f_p and the stopband frequency f_s.

The MCU's ADC reads 8-bit samples at a sample rate of 15.6 kHz [21]. According to Nyquist's theorem, a signal must be sampled at greater than double the highest frequency of the signal to avoid aliasing [26]. Therefore, the greatest frequency in the audio signal must be less than or equal to $\frac{f_{samp}}{2} = \frac{15.6\ kHz}{2} = 7.8\ kHz$. To retain higher frequencies and avoiding aliasing, the stopband frequency will be equal to the Nyquist rate, $f_s = 7.8\ kHz$.

The signal must be attenuated by at least 48dB once the Nyquist rate (7.8 kHz) has been met. The minimum attenuation is given by the ADC's sampling bit-size [21] $A_{min} = 20 \times \log\left(\frac{1}{2^8}\right) = -48.16dB$, as signals below the resolution of the ADC are negligible.

The bandpass frequency must be as high as possible to retain speech information, however this is restricted by the gain roll-off and must be compromised. The fundamental frequency range of a male and females voice is from 100 Hz to 3 kHz. Where frequencies up to 17 kHz are low energy harmonics [3]. The energy of vowels lies in the range of 250 Hz to 2,000 Hz, while voiced consonants (b, d, m, etc.) lie in the 250 Hz to 4,000 Hz range. Unvoiced consonants (f, s, t, etc.) are found in the 2,000 Hz to 8,000 Hz range [4]. Therefore, for audio to be understood clearly, frequencies up to 8kHz should not be attenuated, however, the sampling rate of the MCU's ADC requires the audio to be attenuated by 48 dB at 7.8 kHz. Based on these requirements and limitations, the passband frequency will need to be reverse engineered using the already derived values of A_{max}, A_{min}, and f_s.

As the design is limited to using 2 op amp ICs, the number of op amps available is 8 [20]. Two out of the eight op amps are allocated to DC offset and amplification, while the other 6 will be distributed between the two anti-aliasing filter subcircuits, leaving three op amps to each subcircuit.

Three op amps can be used to cascade three Sallen-Key topology filters in series, giving a 6th order lowpass filter. Through the filter order, and other specifications, the passband cutoff frequency can now be calculated, given the order of Chebyshev filter required formula:

$$n = \left(\frac{\cosh^{-1} \sqrt{\frac{10^{0.1A_{min}} - 1}{10^{0.1A_{max}} - 1}}}{\cosh^{-1} \frac{\omega_s}{\omega_p}} \right)$$

Where:

$$n = 6$$

$$A_{min} = 48dB$$

$$A_{max} = 4dB$$

$$\omega_s = 2\pi \times f_s = 2\pi \times 7.8 \times 10^3 \; rad \times s^{-1}$$

We can solve for ω_p.

$$\omega_p = \frac{\omega_s}{\cosh\left(\frac{\cosh^{-1}\left(\sqrt{\frac{10^{0.1A_{min}} - 1}{10^{0.1A_{max}} - 1}} \right)}{n} \right)} = 31709.37 \; rad \times s^{-1}$$

This is equal to approximately 5040 Hz, which sufficiently captures key speech information, while losing some unvoiced consonants. Therefore, the complete specifications of the MCU filtering system are as follows:

$$A_{min} = 48 \; dB$$

$$A_{max} = 4 \; dB$$

$$f_p = 5040 \; Hz$$

$$f_s = 7800 \; Hz$$

The same process was repeated for the telephone-use anti-aliasing filter. The sample rate of telephone lines is 8kHz, therefore, according to the Nyquist rate, frequencies above 4kHz need to be attenuated. Using the formula to calculate the filter order, filter specifications were derived, as previously done.

$$A_{min} = 48 \; dB$$

$$A_{max} = 4 \; dB$$

$$f_p = 2580 \; Hz$$

$$f_s = 4000 \; Hz$$

Telephone lines are quantized with 8-bits, giving 2^8 quantization levels. This value was the same as that on the MCU's ADC, therefore, both filters can share A_{min} and A_{max}.

2.3.3 DC Offset and Amplification

Before the analogue signal can be inputted into the microcontroller's ADC, it requires amplification and an appropriate DC offset. This is desired as it will allow the analogue signal to make the most of the ADC's resolution by occupying the entire 0V to 5V range [21]. Audio signals have equal positive and negative voltages. Positive voltage pushes the speaker cone forward, moving air towards the listener, while negative voltage pulls it back, moving air away [5]. This creates sound waves, allowing the listener to hear the audio. Therefore, to allow the audio signal to have equal positive and negative headroom around its DC component, the input should be centered halfway between 0V and 5V, giving a DC offset of 2.5V.

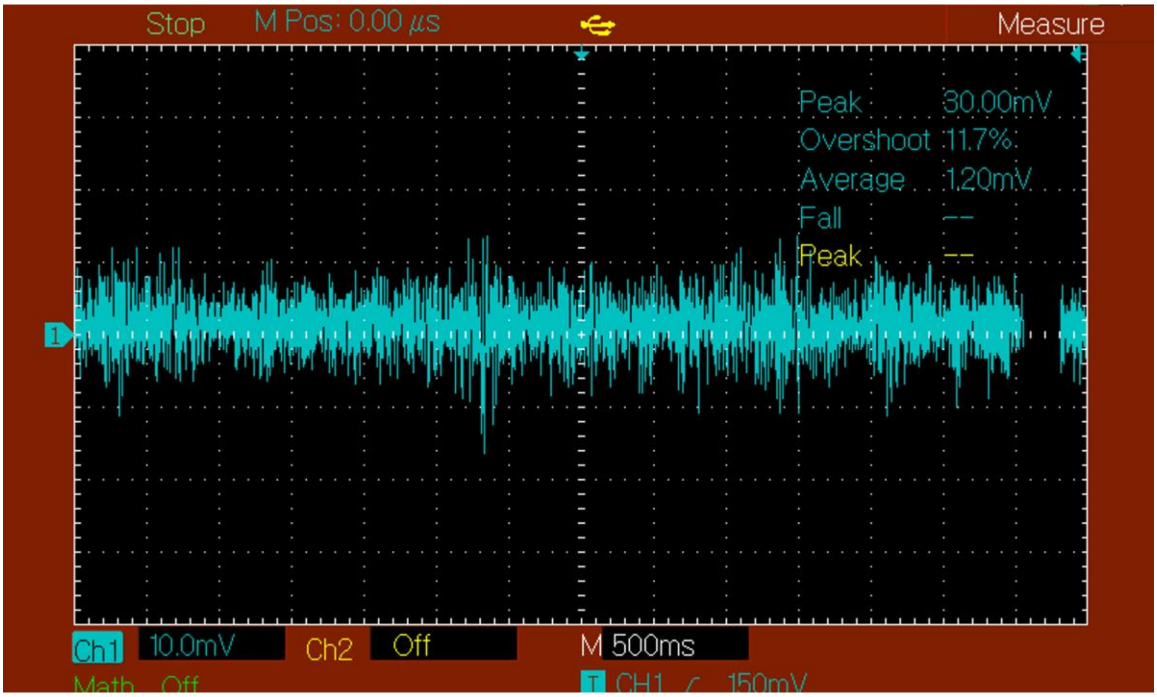


Figure 2.3.5 Oscilloscope capture of microphone before amplification

Using an oscilloscope, the voltage bandwidth of the un-amplified audio signal was measured in a loud conversation environment (Figure 2.3.5). The audio signal requires amplification as it only occupies a 30mV bandwidth and allowing it to occupy the full 5V range will maximize the quantization levels usage (2^8 levels in total). Using more quantization levels will provide a more accurate digital representation of the audio signal [6]. Also, when the signal utilizes more quantization levels, it will provide a better signal-to-quantization-noise ratio (SQNR). SQNR represents the noise caused by quantization, allowing us to quantify how much noise will be introduced during the analogue-to-digital conversion process. The formula for SQNR [7] is given by:

$$\sqrt{SQNR} = \frac{\frac{V}{\sqrt{2}}}{2^{N-1}\sqrt{12}} = 2^N\sqrt{\frac{3}{2}}$$

$$SQNR_{dB} = 20\log_{10}\left(2^N\sqrt{\frac{3}{2}}\right) \approx 6.02N + 1.7609$$

Where N is the number of quantization levels. Therefore as N increases, $SQNR_{dB}$ becomes larger, and the higher the $SQNR_{dB}$, the better the output [8]. Therefore, to maximize $SQNR_{dB}$, the signal should be amplified to utilize all the quantization levels in the 5V range, giving a gain of:

$$G = \frac{5V}{30mV} = 167$$

2.4 Methodology

The design process behind the digital voice recorder will be carried out chronologically, where the circuit is broken down into subsections and worked on one at a time. The subcircuits include DC offset + buffer, anti-aliasing filter, and amplification.

The specifications of each subcircuit were derived from section 2.3. The DC offset and buffer will be designed so that the signal can be centred around 2.5V and maintain this offset regardless of the load. This subcircuit will be validated through LTspice simulations.

The offset signal is then fed into the anti-aliasing filters. There are two filters in the design which the user can switch between using a SPDT switch. There are four specifications required to produce the filter design, A_{min}, A_{max}, f_p, and f_s. Using MATLAB, the transfer function and system's poles can be determined based on the specifications. Component values can be derived using the system's poles and will be rounded to the nearest preferred value, creating a new transfer function. The bode plot of the transfer function using the preferred values will then be compared against the initial parameters to determine if the component selection meets specifications. The step response of the filter system will be plotted and analysed to determine the characteristics of the system [9] and whether it is appropriate for the DVR application.

Afterwards, the filters can be experimentally assessed using an oscilloscope and a waveform generator to produce a bode plot of the practical circuit prototyped on a breadboard.

Finally, the amplification circuit will be designed with the topology of an inverting amplifier. An inverting amplifier was chosen as they are more stable and have lower distortion in comparison to non-inverting amps [10]. The audio signal will be amplified to the gain calculated in section 2.3.3.

2.5 Technical Design

2.5.1 Design Process and Details

The circuit's input comes from an electret microphone, as a voltage signal. This voltage signal is centered around an unknown DC bias and needs to be initially processed to maximize the ADC's range usage. To do this, the microphone was connected as depicted in Figure 2.5.1.

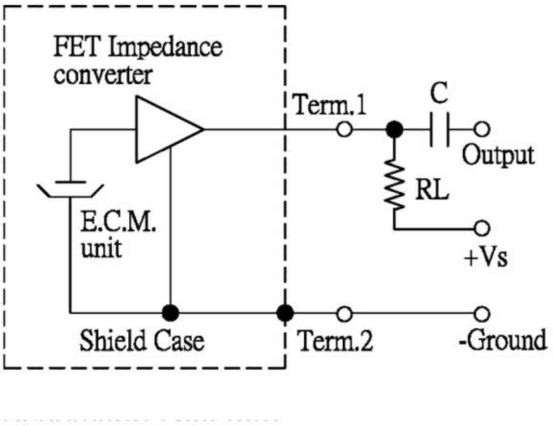


Figure 2.5.1 Recommended microphone circuit topology [25]

With the output fed into a unity gain buffer (Figure 2.5.2), using 2.5V as a reference voltage.

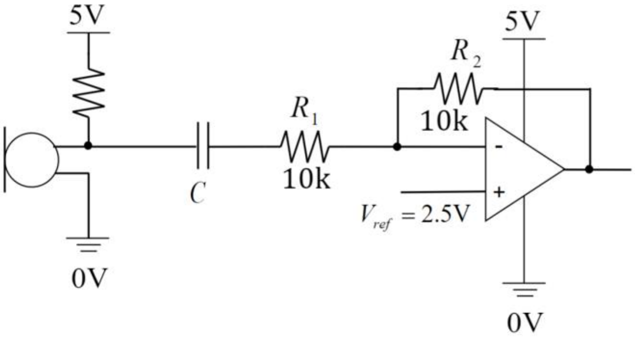
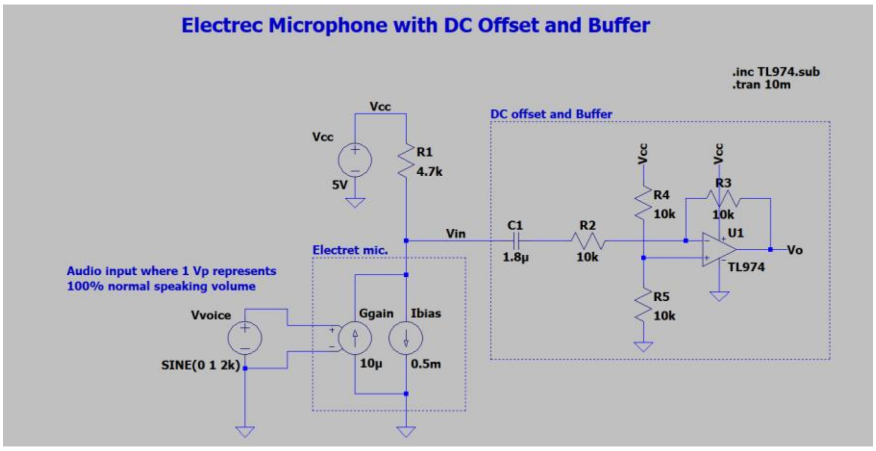


Figure 2.5.2 DC offset and buffer circuit

The purpose of the op-amp is to ensure that the reference voltage is stable at 2.5V. The capacitor, C (Figure 2.5.2), is used to remove the DC bias and center the signal around 0V, which is then recentered around 2.5V, optimizing it for the ADC. The capacitor and unity gain op-amp in combination creates a first order high pass filter, where frequencies under $f_C = \frac{1}{20\pi C \times 10^3}$ are cut-off. Since the frequency range of human speech starts at 90Hz [12], frequencies above this must be preserved for audio clarity. Therefore, the cut-off frequency was chosen to be a decade less than 90Hz, at 9Hz. This was done to ensure that frequencies at 90Hz+ remained intact. This gave a value for C of 1.77nF, rounded to the nearest preferred value in the E12 series gave 1.8nF.

This circuit can be validated in LTspice:



```
.inc TL974.sub
.tran 10m
```

Audio input where 1 Vp represents 100% normal speaking volume

Figure 2.5.3 DC offset and buffer in LTspice

The $10k\Omega$ resistor values are arbitrary and were used for a unity gain (R_2 and R_3), as well as a voltage divider (R_4 and R_5) to create a 2.5V reference voltage (Figure 2.5.3). Using a transient analysis, the performance of the circuit can be determined (Figure 2.5.4).

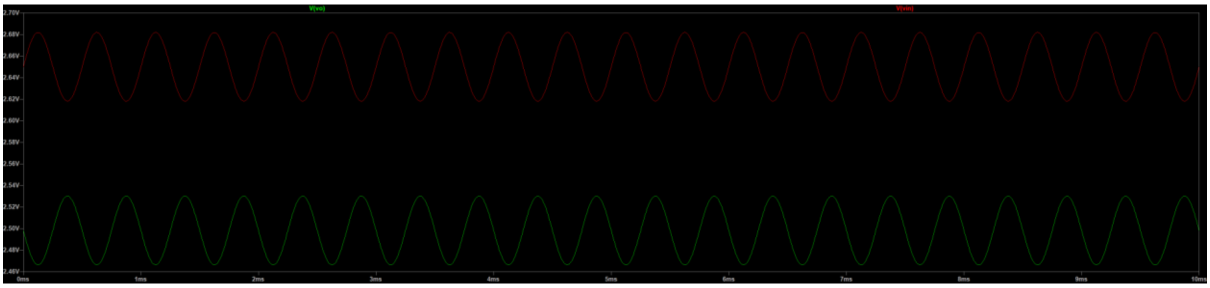


Figure 2.5.4 Transient response of DC offset circuit

As seen in Figure 2.5.4, the input signal (red) was originally centered around 2.65V, and once processed by the DC offset circuit, the output signal (green) was centered around 2.5V. Therefore, this subcircuit meets specifications as it successfully applied an appropriate amount of DC offset.

Now that the DC offset can be successfully applied, the signal is ready for filtering. Given the specifications of the MCU-interfaced filter:

$$A_{min} = 48\ dB$$

$$A_{max} = 4\ dB$$

$$f_p = 5040\ Hz$$

$$f_s = 7800\ Hz$$

Using MATLAB, the transfer function that meets these specifications can be determined.

```
[n_cheb, wn_cheb] = cheb1ord (wp, ws, Amax, Amin, 's');

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

Returning the transfer function:

$$H(s) = \frac{4.06 \times 10^{25}}{s^6 + 1.519 \times 10^4 s^5 + 1.62 \times 10^9 s^4 + 1.82 \times 10^{13} s^3 + 6.62 \times 10^{17} s^2 + 4.258 \times 10^{21} s + 4.06 \times 10^{25}}$$

The step response of a system can be used to measure the system's behavior when the input is suddenly changed [13], allowing us to determine the efficiency and effectiveness of the filter. Through the step response, rise time, settling time, and overshoot can be calculated [14].

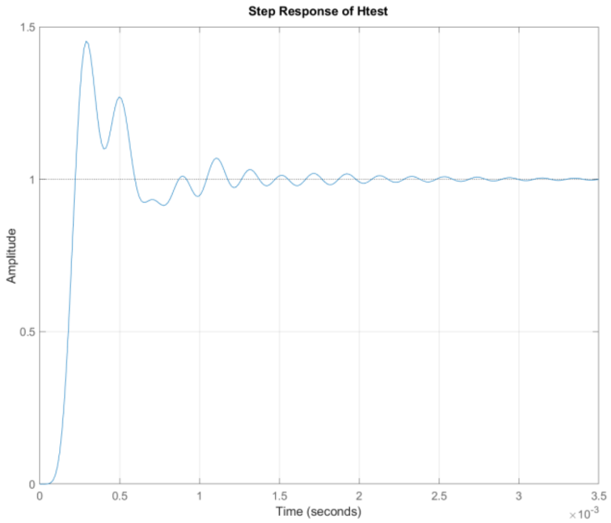


Figure 2.5.5 Step response of anti-aliasing filter

When input into the system switches from 0V to 1V, the system reacts accordingly in Figure 2.5.5. The system has a settling time of $0.0016s$ and overshoots by 45.23%.

```
sys = tf(numerator, denominator);
step_info = stepinfo(sys);

rise_time = step_info.RiseTime;
settling_time = step_info.SettlingTime;
percent_overshoot = step_info.Overshoot;
```

The system's step response converges to its gain factor $H(0) = 1$ within 0.0016 seconds, which is acceptable according to the devices non-high-precision critical requirements. The overshoot of 45% is undesirable and may cause unwanted sound amplitude fluctuations, however, solving this issue is outside the scope of the project. The telephone-interfaced anti-aliasing filter produced a similar step response.

The poles of the system can be determined and are stored in 'p' and can be analyzed to determine the systems stability characteristics.

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

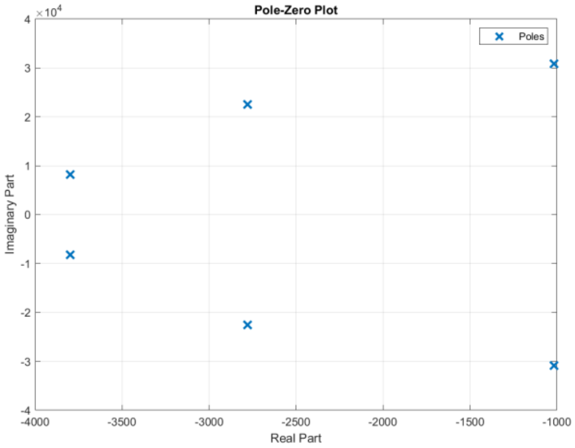


Figure 2.5.6 Poles plot of anti-aliasing filter

These are complex conjugate poles (Figure 2.5.6), meaning that the system is underdamped and stable [15].

The 6th order transfer function can be split into three 2nd order functions, being multiplied by each other. This demonstrates three cascading second order Sallen-Key lowpass active filters (Figure 2.5.7). To determine the component values in the filters, MATLAB was used to select components through the system's poles, where the constraints on the component values were equal value resistors and capacitor ratio.

The general form for a transfer function of a Sallen-Key low pass filter is given by:

$$\frac{v_o}{v_{in}} = \frac{\frac{K}{R_1R_2C_1C_2}}{s^2 + s\left(\frac{1}{R_1C_2} + \frac{1}{R_2C_2} + \frac{1-K}{R_2C_1}\right) + \frac{1}{R_1R_2C_1C_2}}$$

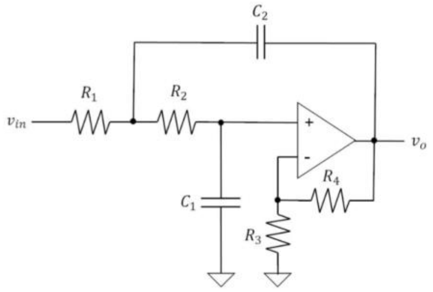


Figure 2.5.7 Lowpass Sallen-Key filter topology

Where $R_3 = \infty\Omega$ (open circuit), and $R_4 = 0\Omega$ (short circuit).

Equations for the natural frequency and quality factor can be derived from the general form.

$$\omega_n = \frac{1}{\sqrt{R_1R_2C_1C_2}} \quad \text{and} \quad Q = \frac{\sqrt{R_1R_2C_1C_2}}{R_2C_1+R_1C_1+(1-K)R_1C_2}$$

Where the constraints on component selection are as follows:

$$R_1 = R_2 = R$$

$$C_1 = C \rightarrow C_2 = nC_1$$

The equations for natural frequency and quality factor can be simplified down to:

$$\omega_n = \frac{1}{RC\sqrt{n}}$$

$$Q = \frac{\sqrt{n}}{2}$$

These exact values were then calculated in MATLAB for all three stages.

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

wnA = sqrt(a2A);
QA = wnA/a1A;


nA = (2*QA)^2;

C1A = (10e-6 / fp);
C2A = nA*C1A;

R1A = 1/(wnA * C1A * sqrt(nA));
R2A = R1A;
```

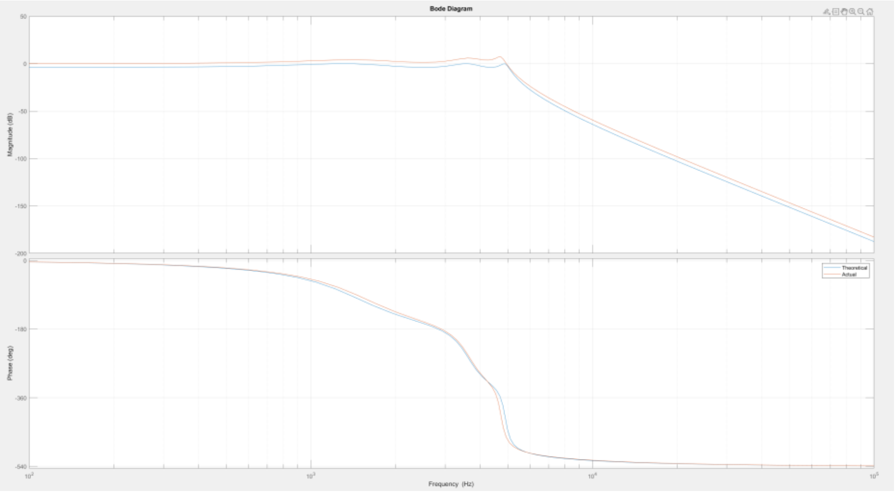
This was repeated for all three stages in the design. To ensure that low-cost goals are met, and that component values are moderately accurate, the E12 series [16] was chosen to derive component values from. Using the preferred values, transfer functions for each stage can be generated, the product of these gives the overall cascading transfer function, which can be compared to the original theoretical transfer function.

$$H_A(s) = \frac{8.858 \times 10^8}{s^2 + 1984s + 8.858 \times 10^8}$$

$$H_B(s) = \frac{5.276 \times 10^8}{s^2 + 5698s + 5.276 \times 10^8}$$

$$H_C(s) = \frac{9.391 \times 10^7}{s^2 + 8264s + 9.391 \times 10^7}$$

$$H_{actual}(s) = H_C(s) \times H_B(s) \times H_C(s)$$



2.5.8 MATLAB bode and phase plot of MCU-interfaced anti-aliasing filter

The bode plot generated (Figure 2.5.8) compares the response of the theoretical transfer functions (blue), against the actual transfer function using preferred values (orange). Both lines resemble each other, however, are different in gain. The gain difference is negligible as this can be corrected in the amplification stage.

The same process of component calculation was repeated for the telephone-interfaced anti-aliasing filter subcircuit. Using preferred values from the E12 series the following transfer functions in the Laplace domain were determined:

$$H_A(s) = \frac{2.096 \times 10^8}{s^2 + 915.8s + 2.096 \times 10^8}$$

$$H_B(s) = \frac{1.303 \times 10^8}{s^2 + 2743s + 1.303 \times 10^8}$$

$$H_C(s) = \frac{2.408 \times 10^7}{s^2 + 4132s + \ 2.408 \times 10^7}$$

$$H_{actual}(s) = H_A(s) + H_B(s) + H_C(s)$$

This transfer function produced the bode plot, Figure 2.5.9.

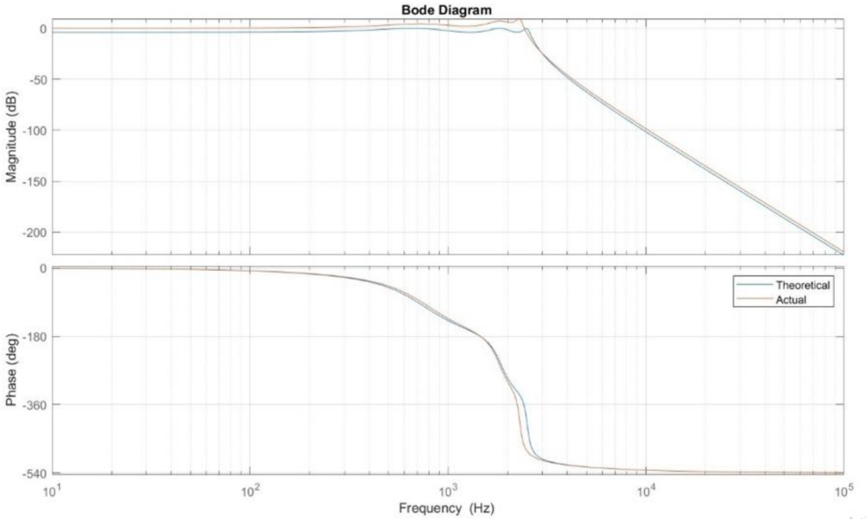


Figure 2.5.9 MATLAB bode and phase plot of telephone-interfaced anti-aliasing filter

These circuits can then be validated through an LTspice simulation.

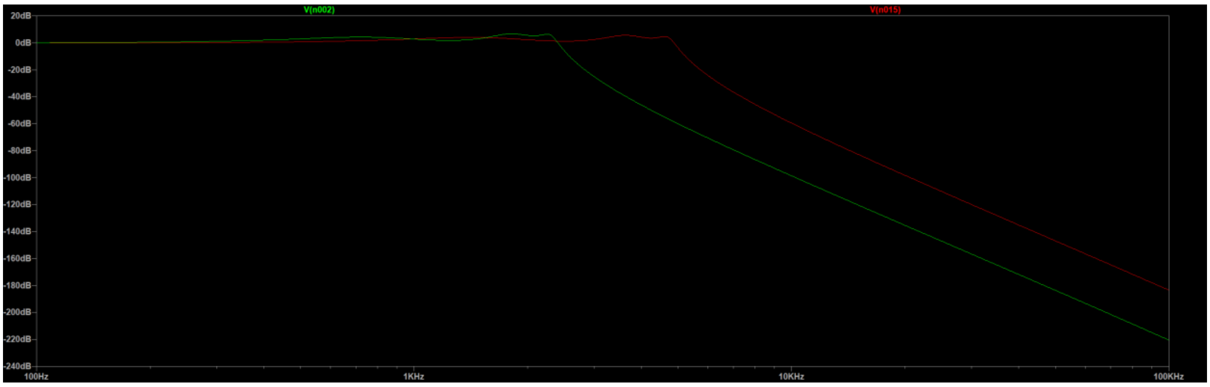
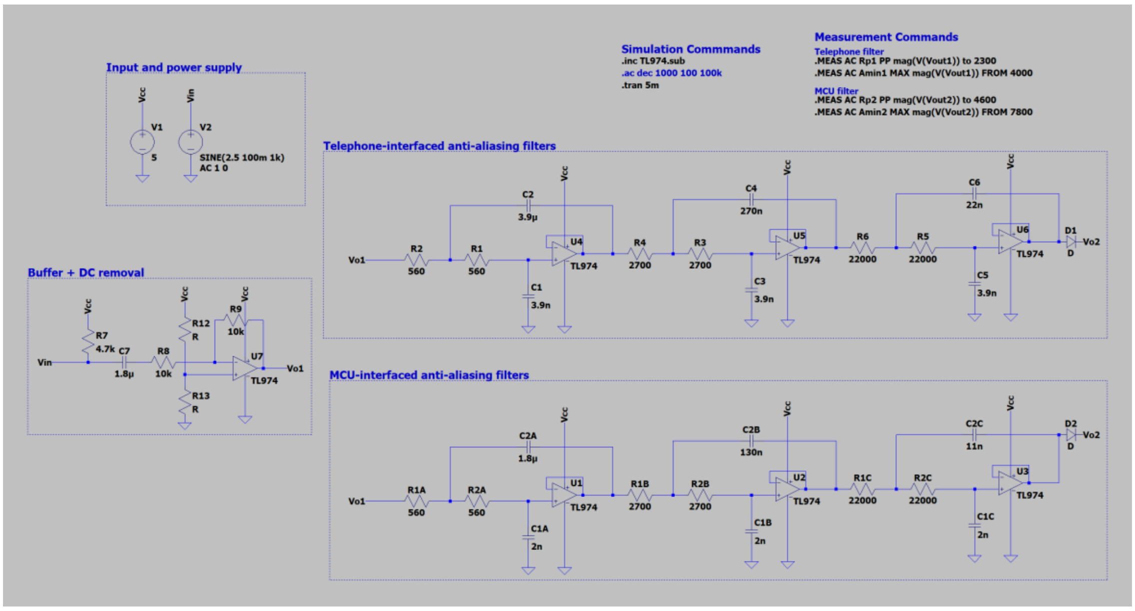


Figure 2.5.10 LTspice simulation of bode plots

The topology in Figure 2.5.11 was used to create this bode plot (Figure 2.5.10).



Simulation Commmands

```
.inc TL974.sub
.ac dec 1000 100 100k
.tran 5m
```

Measurement Commands

Telephone filter

```
.MEAS AC Rp1 PP mag(V(Vout1)) to 2300
.MEAS AC Amin1 MAX mag(V(Vout1)) FROM 4000
```

MCU filter

```
.MEAS AC Rp2 PP mag(V(Vout2)) to 4600
.MEAS AC Amin2 MAX mag(V(Vout2)) FROM 7800
```

2.5.11 Circuit topology of DC offset and anti-aliasing filters

The bode plot of the circuits using the preferred values introduced some error and the circuit does not meet the specifications for f_p as the roll-off starts at a lower frequency than expected. The roll-off for the MCU's filter started at 4.6 kHz, and the roll-off for the telephone's filter started at 2.3 kHz. The simulation's f_p is off by approximately 10%, for both filters. The error is within acceptable margins and will not require rework.

Using the new roll-off start frequencies, the other specifications were calculated using the measurement commands seen in Figure 2.5.11. From the SPICE output logs we can see that the recorded values were:

$$A_{\max\,(telephone)} = 6.7dB$$

$$A_{\min\,(telephone)} = -46.2dB$$

$$A_{\max\,(MCU)} = 5.7dB$$

$$A_{\min\,(MCU)} = -44dB$$

The bandpass ripple does not meet the specification of staying within 4 dB, however the signals are sufficiently attenuated as $A_{max} - A_{min} > 48dB$ for both cases.

Once filtered, the signal is now ready to be amplified before being input into the analogue to digital converter. The common topology of this amplifier type is shown by Figure 2.5.12.

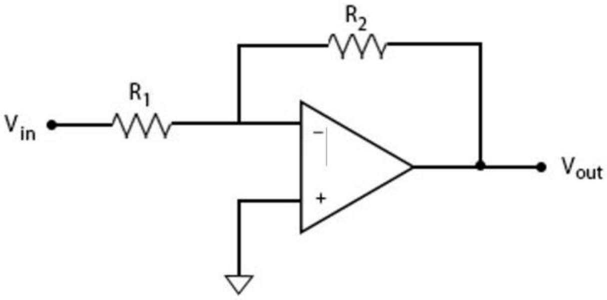


Figure 2.5.12 Inverting amplifier topology

Where the gain is given by $\frac{R_2}{R_1}$, the resistor values can be calculated knowing that the desired gain is 167. For simplicity, R_1 was chosen to be 1kΩ, and R_2, rounded down to the nearest E12 series value, gave 150kΩ (Figure 2.5.13). The resistor was rounded down to give headroom for the peaks, ensuring they do not get clipped by the ADC's 5V range. When the audio signal gets amplified by a gain of $\frac{R_2}{R_1} = 150$ its new bandwidth is expected to be $150 \times 30mV = 4.5V$, leaving 2.5V amplitude of head room on each side of the voltage bandwidth.

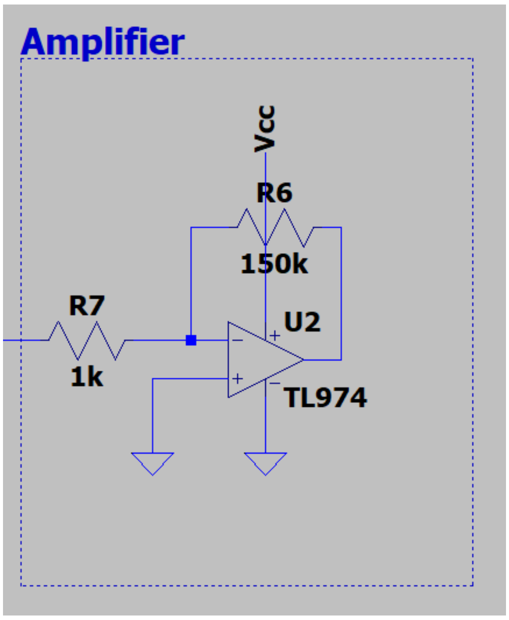


Figure 2.5.13 DVR amplifier circuit topology

2.5.2 Principles of Operation and Schematics

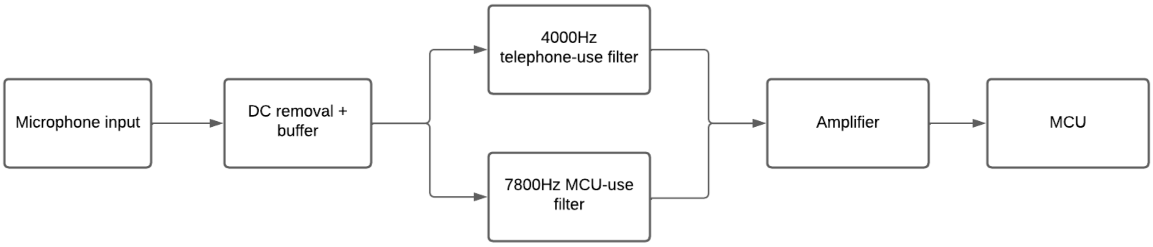


Figure 2.5.14 Block diagram representing DVR operation

Figure 2.5.14 depicts a flowchart representing the circuit's operation. Input is taken from the microphone, where DC bias is removed, and an offset at 2.5V is applied. The offset ensures that the analogue signal is centered within the ADC's input range. The user then has the option between two anti-aliasing subcircuits, one designed to interface with the microcontroller's ADC and the other designed to interface with telephone lines. Once filtered, the analogue signal can be amplified and input into the ADC for digitization.

Operational amplifiers (also known as op-amps) are a high gain voltage amplifier with two differential inputs and a single output (Figure 2.5.15). Op-amps are widely used in electronics due to their versatile applications, such as voltage amplifiers, inverters, buffers, and active filters [17]. In this project op-amps were used for voltage amplification and for active low-pass filters.

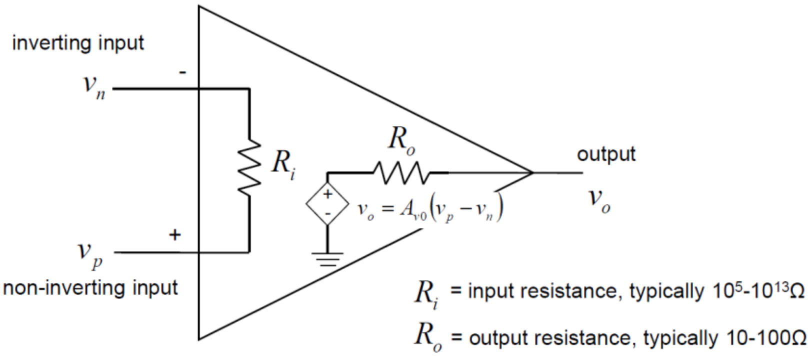
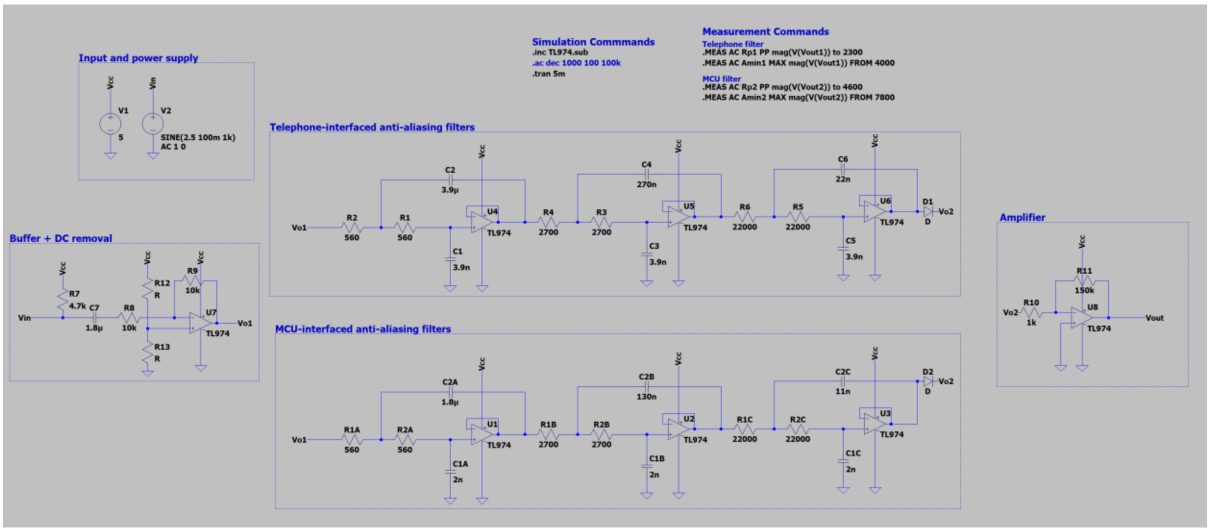

inverting input
v_n
-
R_i
R_o
output
$v_o = A_{v0}(v_p - v_n)$
v_o
+
v_p
non-inverting input
R_i = input resistance, typically 10^5-$10^{13}\Omega$
R_o = output resistance, typically 10-100Ω

Figure 2.5.15 Internals of an operational amplifier [24]

Op-amps work with external feedback components, such as resistors and capacitors, between input and output terminals [18]. The topology of these components depicts the function of the op-amp, whether it is an inverting amplifier, a non-inverting amplifier, or a unity gain follower. Op-amps operate on two golden rules: the output will attempt to make the voltage difference between the inputs zero, and the inputs draw no current [18].

Op-amps were used in a low-pass Sallen-Key topology, which was cascaded to produce higher-order filters for steeper stopband attenuation. The benefit of using an op-amp in this topology is that the design demonstrates the minor dependency of filter performance on the performance of the op-amp [19]. This allows higher frequency performance than other topologies since the op-amp's gain bandwidth does not limit the filter's performance.

The filters and amplifiers in combination give the final circuit topology, Figure 2.5.16.



```
Simulation Commmands
.inc TL974.sub
.ac dec 1000 100 100k
.tran 5m
```

```
Measurement Commands
Telephone filter
.MEAS AC Rp1 PP mag(V(Vout1)) to 2300
.MEAS AC Amin1 MAX mag(V(Vout1)) FROM 4000
MCU filter
.MEAS AC Rp2 PP mag(V(Vout2)) to 4600
.MEAS AC Amin2 MAX mag(V(Vout2)) FROM 7800
```

Figure 2.5.16 Complete analogue conditioning circuit topology

In between V_{o_1} will be a single pole, double throw switch. This was unavailable on the LTspice software and cannot be shown. The switch allows the user to switch between filters, depending on their application.

The purpose of placing the amplifier last after filtering was a crucial design choice to maintain signal quality. The TL974 operational amplifier has a common-mode input voltage range of $\pm 1.35V$ [20], voltages exceeding this range will be clipped off, causing audio data loss. If the signal was amplified to range $\pm 2.5V$ before filtering, the op amps in the anti-aliasing filter subcircuit would clip the peaks (Figure 2.5.17).

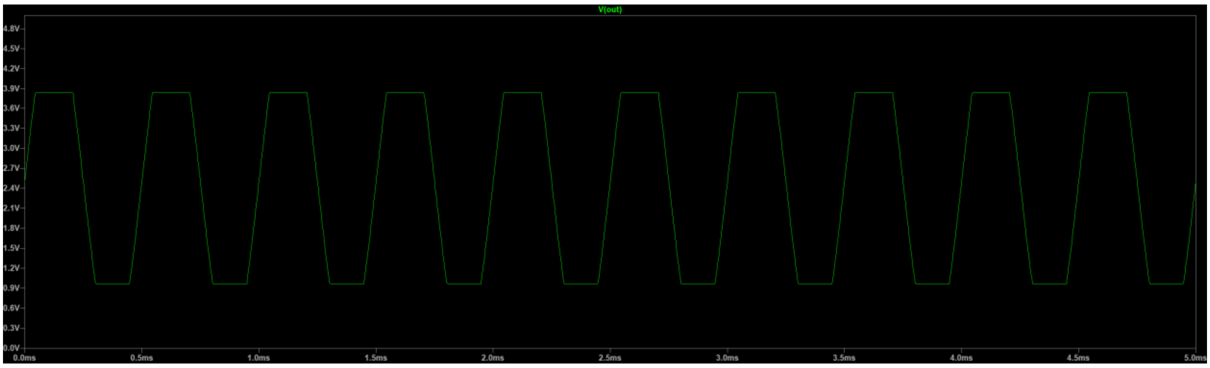


Figure 2.5.17 Transient analysis showing common-mode input voltage clipping

Before amplification, the voltage signal produced by the electret microphone ranged $\pm 31mV$, well within the common-mode input range and will not be clipped.

2.6 Implementation and Testing Procedure

Figure 2.6.1 shows the complete prototype built on a breadboard. Along with annotations of the breadboard assembly (Figure 2.6.2).

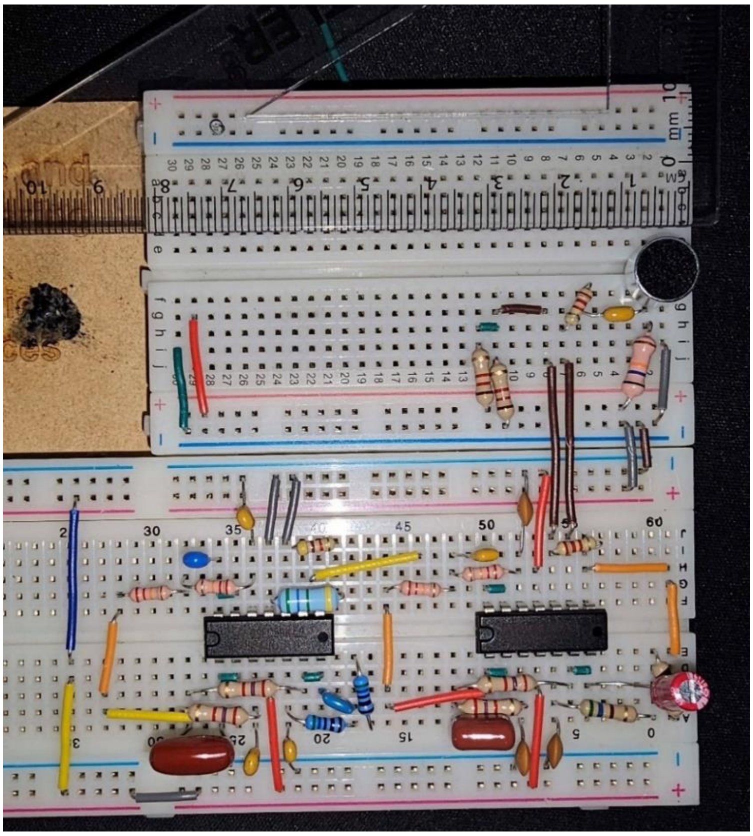


Figure 2.6.1 Breadboard DVR prototype with centimeters scale

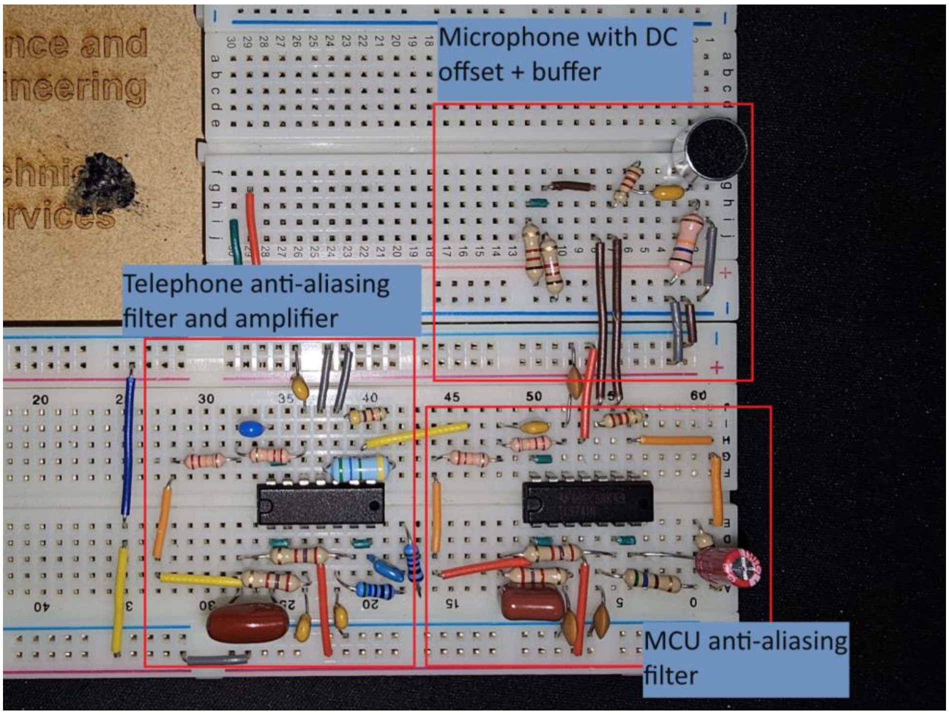

Microphone with DC offset + buffer
Telephone anti-aliasing filter and amplifier
MCU anti-aliasing filter

Figure 2.6.2 Breadboard prototype with annotations

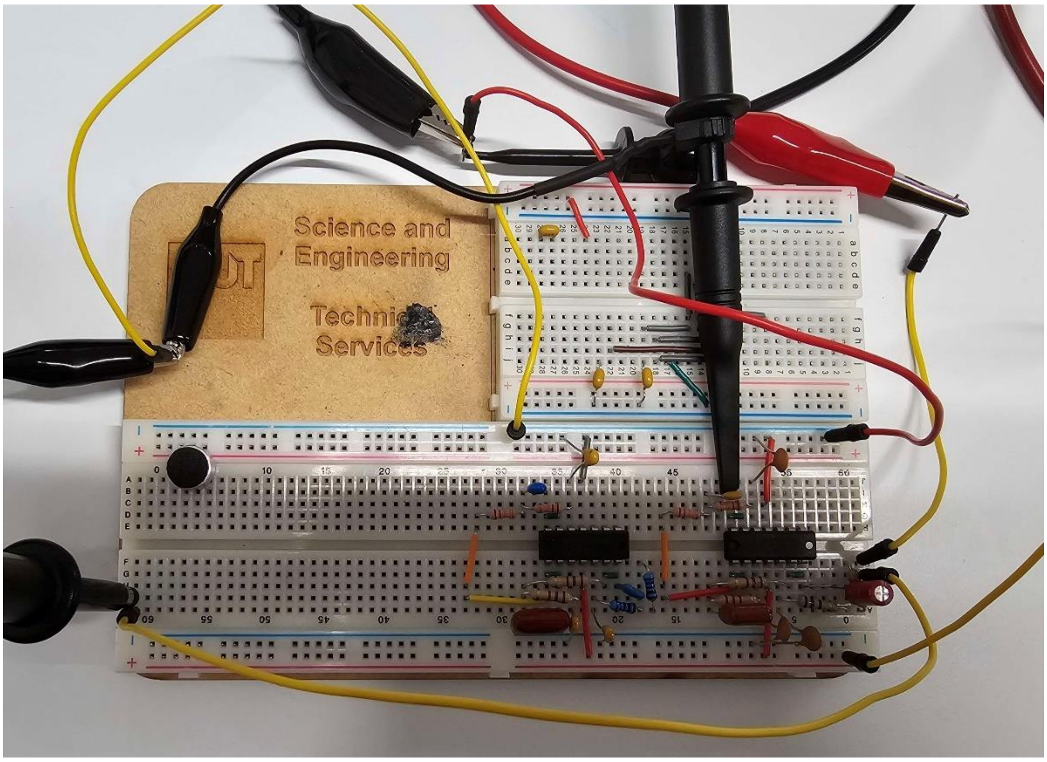


Figure 2.6.3 Prototype being testing with oscilloscope probes connected

Using an oscilloscope, a waveform generator, and a benchtop power supply (Figure 2.6.3), bode plots for the prototype can be developed:

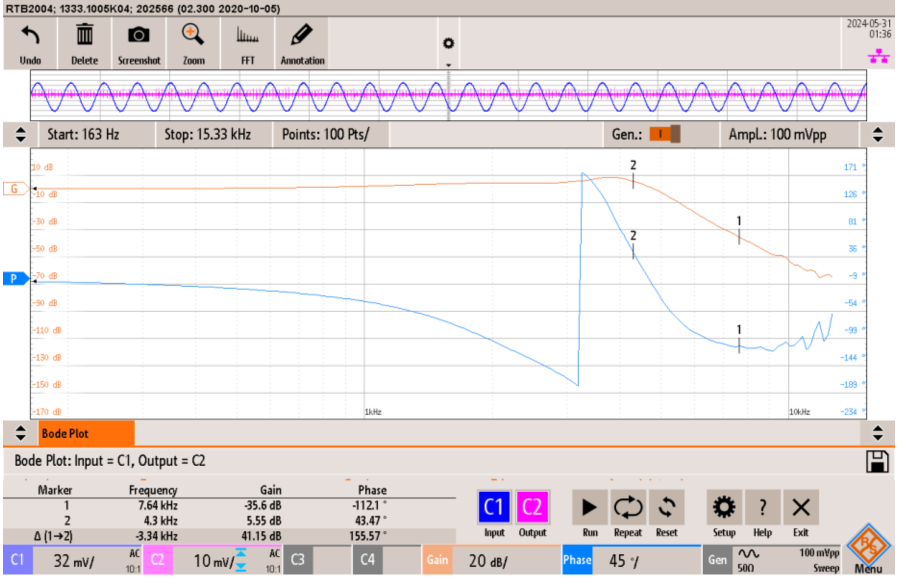
1. Connect the waveform generator to the filter's input; the generator's ground will lead to the power supply's ground.
2. Connect channel 1 of the oscilloscope to the input of the filter (where the waveform generator is connected) and channel 2 to the output of the filter. Then, connect the oscilloscope's channel 1 and 2 ground leads to the power supply's ground terminal.
3. Power the circuit through the op-amps with a 5V supply.

The bode plot feature can be accessed through the oscilloscope's menu. Ensure that input and output are allocated appropriately to channels 1 or 2. Run the plot with 100 iterations.
The source code for the MATLAB simulations and component derivations can be found under section 5.0.

The LTspice netlist for the complete simulation circuit (Figure 2.5.16) is as follows:

```
R2A N022 N018 560
C1A N022 0 2n
C2A N019 N018 1.8µ
R1A N018 Vo1 560
R2B N021 N016 2700
C1B N021 0 2n
C2B N017 N016 130n
R1B N016 N019 2700
R2C N020 N014 22000
C1C N020 0 2n
C2C N015 N014 11n
R1C N014 N017 22000
V1 Vcc 0 5
V2 Vin 0 SINE(2.5 100m 1k) AC 1 0
XU1 N022 N019 Vcc 0 N019 TL974
XU2 N021 N017 Vcc 0 N017 TL974
XU3 N020 N015 Vcc 0 N015 TL974
R7 Vcc Vin 4.7k
C7 N012 Vin 1.8µ
XU7 N013 N010 Vcc 0 Vo1 TL974
R8 N010 N012 10k
R9 Vo1 N010 10k
XU8 0 N011 Vcc 0 Vout TL974
R10 N011 Vo2 1k
R11 Vout N011 68k
R1 N009 N005 560
C1 N009 0 3.9n
C2 N006 N005 3.9µ
R2 N005 Vo1 560
R3 N008 N003 2700
C3 N008 0 3.9n
C4 N004 N003 270n
R4 N003 N006 2700
R5 N007 N001 22000
C5 N007 0 3.9n
C6 N002 N001 22n
R6 N001 N004 22000
XU4 N009 N006 Vcc 0 N006 TL974
XU5 N008 N004 Vcc 0 N004 TL974
XU6 N007 N002 Vcc 0 N002 TL974
D1 N002 Vo2 D
D2 N015 Vo2 D
R12 Vcc N013 R
R13 N013 0 R
.model D D
.lib C:\Users\zacka\AppData\Local\LTspice\lib\cmp\standard.dio
.tran 5m
* .ac dec 1000 100 100k
.inc TL974.sub
* Buffer + DC removal
* MCU-interfaced anti-aliasing filters
* Input and power supply
* Telephone-interfaced anti-aliasing filters
* Simulation Commmands
* Amplifier
* Measurement Commands
.MEAS AC Rp1 PP mag(V(Vout1)) to 2300
.MEAS AC Amin1 MAX mag(V(Vout1)) FROM 4000
.MEAS AC Rp2 PP mag(V(Vout2)) to 4600
.MEAS AC Amin2 MAX mag(V(Vout2)) FROM 7800
* MCU filter
* Telephone filter
.backanno
.end
```

2.7 Experimental design review



RTB2004; 1333.1005K04; 202566 (02.300 2020-10-05)

Start: 163 Hz | Stop: 15.33 kHz | Points: 100 Pts/ | Gen.: | Ampl.: 100 mVpp

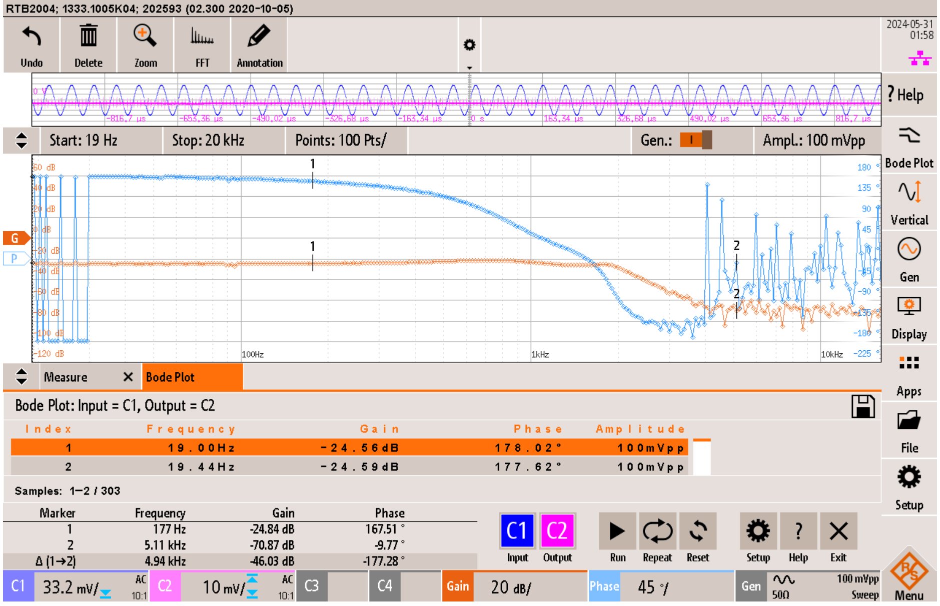
Bode Plot: Input = C1, Output = C2

Marker	Frequency	Gain	Phase
1	7.64 kHz	-35.6 dB	-112.1 °
2	4.3 kHz	5.55 dB	43.47 °
Δ (1→2)	-3.34 kHz	41.15 dB	155.57 °

C1 32 mV/ AC 10:1 | C2 10 mV/ AC 10:1 | Gain 20 dB/ | Phase 45 °/ | Gen 50Ω 100 mVpp Sweep

Figure 2.7.1 Oscilloscope bode plot capture of MCU-interfaced filter

As depicted in the bode plot (Figure 2.7.1), the roll-off starts too soon at before 4.3 kHz (marker 2). Although this does not meet the initial specifications, it does replicate the early roll-off seen in the LTspice simulation (Figure 2.5.10).



RTB2004; 1333.1005K04; 202593 (02.300 2020-10-05)

Start: 19 Hz | Stop: 20 kHz | Points: 100 Pts/ | Gen.: | Ampl.: 100 mVpp

Bode Plot: Input = C1, Output = C2

Index	Frequency	Gain	Phase	Amplitude
1	19.00Hz	-24.56dB	178.02°	100mVpp
2	19.44Hz	-24.59dB	177.62°	100mVpp

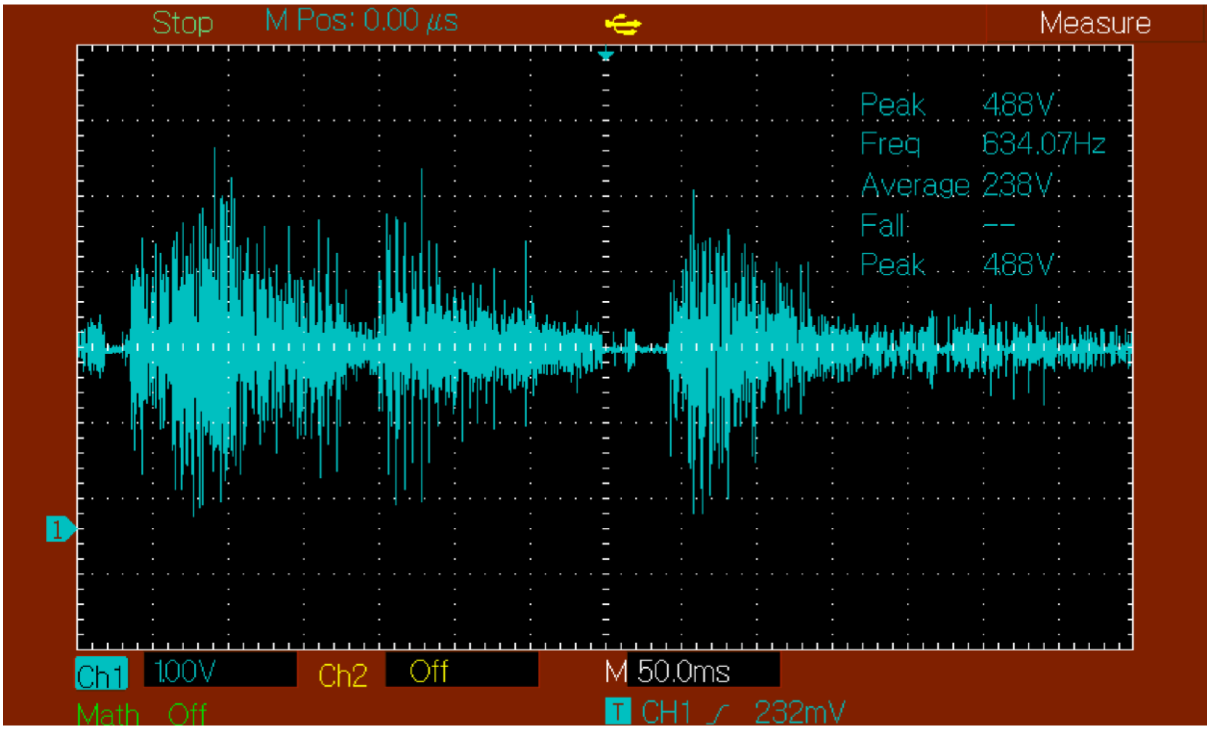
Samples: 1–2 / 303

Marker	Frequency	Gain	Phase
1	177 Hz	-24.84 dB	167.51 °
2	5.11 kHz	-70.87 dB	-9.77 °
Δ (1→2)	4.94 kHz	-46.03 dB	-177.28 °

C1 33.2 mV/ AC 10:1 | C2 10 mV/ AC 10:1 | Gain 20 dB/ | Phase 45 °/ | Gen 50Ω 100 mVpp Sweep

Figure 2.7.2 Oscilloscope bode plot capture of telephone-interfaced filter

For the telephone-interfaced anti-aliasing filter (Figure 2.7.2), the gain roll-off occurred prematurely, as seen from the LTspice simulation (Figure 2.5.10). However, in this case the gain and phase lines started produces erratic behavior at higher frequencies. Although the reason is not certain, possible influences for the erratic behavior may be due to noise interference, poor breadboard circuit connections, or incorrect oscilloscope parameter selection.



Stop M Pos: 0.00 µs

Measure

Peak 4.88V
Freq 634.07Hz
Average 2.38V
Fall --
Peak 4.88V

Ch1 1.00V Ch2 Off M 50.0ms
Math Off T CH1 232mV

Figure 2.7.3 Oscilloscope capture show DC offset and amplification

Figure 2.7.3 shows the DC offset, buffer, and amplification circuit under oscilloscope simulation. The peak-to-peak value was measured to be 4.88V, slightly over the expected $150 \times 30mV = 4.5V$ bandwidth (8% difference). The error could be due to the E12 tolerance of 10% [16].

The average measurement, 2.38V (Figure 2.7.3), shows the DC offset. Although the desired DC offset was 2.5V (5% difference), the error is expected, and acceptable, to due resistor tolerances.

Overall, the circuit meets expectations, with some unexpected behavior from the filter passband frequencies. The DC offset and amplification circuit works as expected with foreseeable error, due to resistor value tolerances.

Finally, the analogue conditioning circuit was connected to the development board to record music audio. The output of the digital speaker accurately represented the recorded audio, with some noise present. Overall, the speaker output was easily understood, deeming the designed circuit successful.

2.8 Future work

Unfortunately, several compromises were made during this project due to technical constraints. For example, the early roll-off encountered during the LTspice simulations and experimental captures persisted through thorough calculations. The reasons behind the design flaw still need to be clarified and may not be resolved due to technical difficulties.

While the troubleshooting of the issues encountered in this project was hindered by time and technical constraints, it presents a promising area for future development. An additional avenue for enhancement would be to increase the order of filter in the anti-aliasing filter subcircuits. This would broaden the passband, enhancing the clarity of unvoiced consonants (f, s, t, etc.) for the listener.

Future work for this project may include a potentiometer to control the input sound level, allowing the user to reduce the recording volume or mute themselves. This may be useful in public environments where a lot of background noise is present, and the user wants to reduce the background noise.

3.0 Conclusion

The project aimed to design an analogue signal conditioning system for a digital voice recorder (DVR) that interfaces with a microcontroller and telephone lines. The design process, which included stages of DC offset adjustment, anti-aliasing filtering, and signal amplification, utilized a 6th-order Sallen-Key lowpass filter. Despite thorough mathematical and qualitative specification determination, component selection, and simulation via LTspice, the prototype was still needed to fully meet the design specifications, as it showed evidence of premature attenuation.

Several factors likely contributed to this deviation, including component tolerances, rounding errors, non-ideal component behavior, and op-amp characteristics. These issues still need to be fully resolved due to the project's technical and scope limitations. However, the observed performance deviation was within a 10% margin of error, deemed acceptable for the DVR's non-critical applications.

The project's progression from design through prototyping provided significant insights into the intricacies of analogue signal processing. Despite not achieving the ideal performance, the experience highlighted the necessity for rigorous testing and iterative refinement. This project provided an insight to real-world electronic design and should be reflected upon for future projects.

4.0 References

[1] T. Agarwal, “Chebyshev filter - Diffferent types of Chebyshev filters,” ElProCus - Electronic Projects for Engineering Students, Jun. 16, 2015. https://www.elprocus.com/types-of-chebyshev-filters/

[2] “Sound level and auditory perception.” https://www.acousti-tech.com/en/blog/detail/sound-level-and-auditory-perception-4129

[3] “HUMAN VOICE FREQUENCY RANGE - SEA,” SEA. https://seaindia.in/blogs/human-voice-frequency-range/

[4] “Generating and understanding speech | Ecophon.” https://www.ecophon.com/en/about-ecophon/acoustic-knowledge/basic-acoustics/generating-and-understanding-speech/

[5] “Phase vs polarity explained.” https://www.simplymixing.com/blog/phase-vs-polarity-audio

[6] “Electronics simplified,” ScienceDirect. https://www.sciencedirect.com/book/9780080970639/electronics-simplified

[7] “Compute SQNR (Signal to Quantization Noise Ratio),” Signal Processing Stack Exchange. https://dsp.stackexchange.com/questions/43580/compute-sqnr-signal-to-quantization-noise-ratio

[8] P. Kumar and P. Kumar, “What is Signal-to-Noise Ratio (SNR)? Why is SNR important in embedded cameras?,” E-con Systems, May 21, 2024. https://www.e-consystems.com/blog/camera/technology/what-is-signal-to-noise-ratio-snr-why-is-snr-important-in-embedded-cameras/

[9] “Rise time, settling time, and other step-response characteristics - MATLAB stepinfo - MathWorks Australia.” https://au.mathworks.com/help/ident/ref/dynamicsystem.stepinfo.html

[10] “Inverting vs Non-Inverting Op-Amp: A Comparison,” Cadence, Mar. 19, 2024. https://resources.pcb.cadence.com/blog/2024-inverting-vs-non-inverting-op-amp-a-comparison

[11] SwellFox, “What are the Golden Rules of Op-Amps?,” CircuitBread, Mar. 20, 2024. https://www.circuitbread.com/ee-faq/what-are-the-golden-rules-of-op-amps

[12] “HUMAN VOICE FREQUENCY RANGE - SEA,” SEA. https://seaindia.in/blogs/human-voice-frequency-range/

[13] Khan Academy, “RC Step response,” Khan Academy. https://www.khanacademy.org/science/electrical-engineering/ee-circuit-analysis-topic/ee-natural-and-forced-response/a/ee-rc-step-response

[14] “Control Systems In Practice, Part 9: The step response.” https://au.mathworks.com/videos/control-systems-in-practice-part-9-the-step-response-1593067191882.html

[15] “2nd-order System dynamics.” https://controlsystemsacademy.com/0024/0024.html

[16] E. Notes, “Standard resistor values: E3 E6 E12 E24 E48 E96 » Electronics Notes.” https://www.electronics-notes.com/articles/electronic_components/resistors/standard-resistor-values-e-series-e3-e6-e12-e24-e48-e96.php

[17] “For what applications are op-amps used? | Toshiba Electronic Devices & Storage Corporation | Asia-English.” https://toshiba.semicon-storage.com/ap-en/semiconductor/knowledge/faq/linear_opamp/for-what-applications-are-op-amps-used.html

[18] “Operational Amplifier Basics,” ElectronicsTutorials. https://www.electronics-tutorials.ws/opamp/opamp_1.html

[19] “Linear Circuit Design Handbook,” ScienceDirect. https://www.sciencedirect.com/book/9780750687034/linear-circuit-design-handbook

[20] “TL97X Output Rail-To-Rail Very-Low-Noise operational amplifiers,” technical, Jan. 2015. [Online]. Available: https://www.ti.com/lit/ds/symlink/tl974.pdf?ts=1717212928770&ref_url=https%253A%252F%252Fwww.google.com%252F

[21] “ATmega16U4/32U4 [DATASHEET],” Apr. 2016. [Online]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7766-8-bit-AVR-ATmega16U4-32U4_Datasheet.pdf

[22] “About the Chebyshev filter,” Jan. 05, 2024. https://resources.pcb.cadence.com/blog/about-the-chebyshev-filter

[23] “Product Documentation - NI.” https://www.ni.com/docs/en-US/bundle/labwindows-cvi/page/advancedanalysisconcepts/lvac_butterworth_filters.html

[24] “Operational Amplifier Basics, Types and Uses| Article | MPS.” https://www.monolithicpower.com/operational-amplifiers

[25] CUI DEVICES, “ELECTRET CONDENSER MICROPHONE,” Aug. 2022. [Online]. Available: https://www.cuidevices.com/product/resource/cma-6542tf-k.pdf

[26] “Electrical Engineer’s Reference Book,” *ScienceDirect*. https://www.sciencedirect.com/book/9780750646376/electrical-engineers-reference-book

5.0 Appendices

```
%%% Filter specs
fsamp = 15.625e3;

fs = fsamp/2;    % stopband frequency, Hz
fp = 5.04e3;      % passband frequency, Hz

ws = 2*pi*fs;
wp = 2*pi*fp;

Amin = 20*log10(1/2^8);   % stopband attenuation, dB   48dB
Amax = 4;               % passband attenuation, dB


%Calculate required order of Butterworth and Chebyshev filters
[n_cheb, wn_cheb] = cheb1ord (wp, ws, Amax, Amin, 's');

%%
% Bode plot
[b a] = cheby1(n_cheb, Amax, wn_cheb,'low','s');
H = tf(b,a);

%figure (2);
h = bodeplot(H);                                       % display bode plot
setoptions(h,'FreqUnits','Hz');                        % change units to Hz
%setoptions(h,'FreqUnits','Hz','PhaseVisible','off');  % remove phase plot
grid on;

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

%% Calculate natural frequency and quality factor for each stage

%%%% Stage A
a1A = -p(1)-p(2);
a2A = p(1)*p(2);

wnA = sqrt(a2A);
QA = wnA/a1A;

fprintf(1,'Stage A: wn = %g rad/s, Q = %g\n',wnA,QA);

%%% Stage B
a1B = -p(3)-p(4);
a2B = p(3)*p(4);

wnB = sqrt(a2B);
QB = wnB/a1B;

fprintf(1,'Stage B: wn = %g rad/s, Q = %g\n',wnB,QB);

%%% Stage C
a1C = -p(5)-p(6);
a2C = p(5)*p(6);

wnC = sqrt(a2C);
QC = wnC/a1C;
```

```
fprintf(1,'Stage C: wn = %g rad/s, Q = %g\n',wnC,QC);

%%

%%% Stage A design
KA = 1;

nA = (2*QA)^2;

C1A = (10e-6 / fp);
C2A = nA*C1A;

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

fprintf(1,'R1A = %g, R2A = %g, C1A = %g, C2A = %g\n',R1A,R2A,C1A,C2A);


%%% Preferred Values

R1A = 560;
R2A = 560;
C1A = 2e-9;
C2A = 1.8e-6;

%%% Stage B design
KA = 1;

nB = (2*QB)^2;

C1B = (10e-6 / fp);
C2B = nB*C1B;

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

fprintf(1,'R1B = %g, R2B = %g, C1B = %g, C2B = %g\n',R1B,R2B,C1B,C2B);

%%% Preferred Values

R1B = 2700;
R2B = 2700;
C1B = 2e-9;
C2B = 130e-9;

%%% Stage C design
KA = 1;

nC = (2*QC)^2;

C1C = (10e-6 / fp);
C2C = nC*C1C;

R1C = 1/(wnC * C1C * sqrt(nC));
R2C = R1C;

fprintf(1,'R1C = %g, R2C = %g, C1C = %g, C2C = %g\n',R1C,R2C,C1C,C2C);

%%% Preferred Values
```

```
R1C = 22000;
R2C = 22000;
C1C = 2e-9;
C2C = 11e-9;

%%

%%% Compare with original function
K1test = 1;
numA = [0 0 K1test/(R1A*R2A*C1A*C2A)];
% numerator
denA = [1 (1/(R1A*C2A) + 1/(R2A*C2A) + (1-K1test)/(R2A*C1A)) 1/(R1A*R2A*C1A*C2A)];
% denominator
HtestA = tf(numA,denA)

K2test = 1;
numB = [0 0 K2test/(R1B*R2B*C1B*C2B)];
% numerator
denB = [1 (1/(R1B*C2B) + 1/(R2B*C2B) + (1-K2test)/(R2B*C1B)) 1/(R1B*R2B*C1B*C2B)];
% denominator
HtestB = tf(numB,denB)

K3test = 1;
numC = [0 0 K3test/(R1C*R2C*C1C*C2C)];
% numerator
denC = [1 (1/(R1C*C2C) + 1/(R2C*C2C) + (1-K3test)/(R2C*C1C)) 1/(R1C*R2C*C1C*C2C)];
% denominator
HtestC = tf(numC,denC)

%%

Htest = HtestA * HtestB * HtestC

%%% Theoretical
figure;
h1 = bodeplot(H);
setoptions(h1, 'FreqUnits', 'Hz');
%setoptions(h1, 'FreqUnits', 'Hz', 'PhaseVisible', 'off');
grid on;
hold on;

%%% Actual
h2 = bodeplot(Htest);
setoptions(h2, 'FreqUnits', 'Hz');
%setoptions(h2, 'FreqUnits', 'Hz', 'PhaseVisible', 'off');
grid on;

% Add legend
legend('Theoretical', 'Actual');

hold off;

%%

% Plot step response
figure;
step(Htest);
title('Step Response of Htest');
```

```
grid on;
```

Telephone Interfaced

```
%%% Filter specs
fsamp = 8e3;

fs = fsamp/2;    % stopband frequency, Hz
fp = 2580;      % passband frequency, Hz

ws = 2*pi*fs;
wp = 2*pi*fp;

Amin = 20*log10(1/2^8);   % stopband attenuation, dB   48dB
Amax = 4;               % passband attenuation, dB


%Calculate required order of Butterworth and Chebyshev filters
[n_cheb, wn_cheb] = cheb1ord (wp, ws, Amax, Amin, 's');

%%
% Bode plot
[b a] = cheby1(n_cheb, Amax, wn_cheb,'low','s');
H = tf(b,a);

%figure (2);
h = bodeplot(H);                                       % display bode plot
setoptions(h,'FreqUnits','Hz');                        % change units to Hz
%setoptions(h,'FreqUnits','Hz','PhaseVisible','off');  % remove phase plot
grid on;

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

%% Calculate natural frequency and quality factor for each stage

%%%% Stage A
a1A = -p(1)-p(2);
a2A = p(1)*p(2);

wnA = sqrt(a2A);
QA = wnA/a1A;

fprintf(1,'Stage A: wn = %g rad/s, Q = %g\n',wnA,QA);

%%% Stage B
a1B = -p(3)-p(4);
a2B = p(3)*p(4);

wnB = sqrt(a2B);
QB = wnB/a1B;

fprintf(1,'Stage B: wn = %g rad/s, Q = %g\n',wnB,QB);

%%% Stage C
a1C = -p(5)-p(6);
a2C = p(5)*p(6);

wnC = sqrt(a2C);
```

```
QC = wnC/a1C;

fprintf(1,'Stage C: wn = %g rad/s, Q = %g\n',wnC,QC);

%%

%%% Stage A design
KA = 1;

nA = (2*QA)^2;

C1A = (10e-6 / fp);
C2A = nA*C1A;

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

fprintf(1,'R1A = %g, R2A = %g, C1A = %g, C2A = %g\n',R1A,R2A,C1A,C2A);


%%% Preferred Values

R1A = 560;
R2A = 560;
C1A = 3.9e-9;
C2A = 3.9e-6;

%%% Stage B design
KA = 1;

nB = (2*QB)^2;

C1B = (10e-6 / fp);
C2B = nB*C1B;

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

fprintf(1,'R1B = %g, R2B = %g, C1B = %g, C2B = %g\n',R1B,R2B,C1B,C2B);

%%% Preferred Values


R1B = 2700;
R2B = 2700;
C1B = 3.9e-9;
C2B = 270e-9;

%%% Stage C design
KA = 1;

nC = (2*QC)^2;

C1C = (10e-6 / fp);
C2C = nC*C1C;

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

```
fprintf(1,'R1C = %g, R2C = %g, C1C = %g, C2C = %g\n',R1C,R2C,C1C,C2C);

%%% Preferred Values

R1C = 22000;
R2C = 22000;
C1C = 3.9e-9;
C2C = 22e-9;


%%

%%% Compare with original function
K1test = 1;
numA = [0 0 K1test/(R1A*R2A*C1A*C2A)];
% numerator
denA = [1 (1/(R1A*C2A) + 1/(R2A*C2A) + (1-K1test)/(R2A*C1A)) 1/(R1A*R2A*C1A*C2A)];
% denominator
HtestA = tf(numA,denA)

K2test = 1;
numB = [0 0 K2test/(R1B*R2B*C1B*C2B)];
% numerator
denB = [1 (1/(R1B*C2B) + 1/(R2B*C2B) + (1-K2test)/(R2B*C1B)) 1/(R1B*R2B*C1B*C2B)];
% denominator
HtestB = tf(numB,denB)

K3test = 1;
numC = [0 0 K3test/(R1C*R2C*C1C*C2C)];
% numerator
denC = [1 (1/(R1C*C2C) + 1/(R2C*C2C) + (1-K3test)/(R2C*C1C)) 1/(R1C*R2C*C1C*C2C)];
% denominator
HtestC = tf(numC,denC)

%%


Htest = HtestA * HtestB * HtestC

%%% Theoretical
figure;
h1 = bodeplot(H);
setoptions(h1, 'FreqUnits', 'Hz');
%setoptions(h1, 'FreqUnits', 'Hz', 'PhaseVisible', 'off');
grid on;
hold on;

%%% Actual
h2 = bodeplot(Htest);
setoptions(h2, 'FreqUnits', 'Hz');
%setoptions(h2, 'FreqUnits', 'Hz', 'PhaseVisible', 'off');
grid on;

% Add legend
legend('Theoretical', 'Actual');

hold off;

%%
```

```
% Plot step response
figure;
step(Htest);
title('Step Response of Htest');
grid on;
```