

Final Project Report

ECE 3750 Fall 2023

John Berberian, Jr. Paul Karhnak Lourdes Leung
Liam Timmins

January 31, 2024

Contents

| | | |
|----------|--------------------------------------------------|----------|
| 0.1 | Lab Responsibilities | 2 |
| 0.2 | Honor Pledge | 1 |
| 1 | Design Process | 2 |
| 1.1 | Background | 2 |
| 1.1.1 | Generic Components | 2 |
| 1.1.2 | Power Supply | 2 |
| 1.1.3 | Input Filters | 2 |
| 1.1.4 | Instrumentation Amplifier | 3 |
| 1.1.5 | Integrator | 3 |
| 1.1.6 | V_{mid} Supply | 3 |
| 1.1.7 | Butterworth Filter | 3 |
| 1.1.8 | Isolator | 3 |
| 1.2 | Schematics | 4 |
| 1.2.1 | Generic Components | 4 |
| 1.2.2 | Power Supply | 4 |
| 1.2.3 | Input Filters | 5 |
| 1.2.4 | Instrumentation Amplifier | 5 |
| 1.2.5 | Integrator | 7 |
| 1.2.6 | V_{mid} Supply | 8 |
| 1.2.7 | Butterworth Filter | 9 |
| 1.2.8 | Isolator | 10 |
| 1.3 | Multisim Simulations | 14 |
| 1.3.1 | Input Filters | 14 |
| 1.3.2 | Instrumentation Amplifier | 14 |
| 1.3.3 | Integrator | 14 |
| 1.3.4 | V_{mid} Supply | 16 |
| 1.3.5 | Butterworth Filter | 16 |
| 1.3.6 | Isolator | 21 |
| 1.3.7 | Individual section simulation overview | 21 |

| | | |
|----------|--------------------------------------------|-----------|
| 1.3.8 | Complete EKG Simulation | 21 |
| 1.4 | Layout | 24 |
| 2 | Experimental | 25 |
| 2.1 | Assembly and Testing | 25 |
| 2.1.1 | Preliminary Tests | 25 |
| 2.1.2 | Isolator | 25 |
| 2.1.3 | V_{mid} | 26 |
| 2.1.4 | Butterworth Filter | 26 |
| 2.1.5 | Input Filters | 26 |
| 2.1.6 | Integrator | 28 |
| 2.1.7 | Instrumentation Amplifier | 28 |
| 2.2 | Setup Images | 31 |
| 2.3 | Data Capture | 31 |
| 2.3.1 | Data Collection | 31 |
| 2.3.2 | FIR Filtering | 34 |
| 3 | Conclusion | 38 |
| 4 | Bibliography | 38 |
| 5 | Code | 39 |
| 5.1 | Input Filter Brute-Force | 39 |
| 5.2 | Input Signal Range Calculations | 40 |
| 5.3 | Integrator Value Selection | 40 |
| 5.4 | V_{mid} Properties Brute-Force | 41 |
| 5.5 | Filter Value Brute-Force | 42 |
| 5.6 | DSP Filter Calculations | 44 |
| 5.7 | DSP Filtering Code | 46 |

0.1 Lab Responsibilities

John Berberian, ccg3sr

This student performed most of the analysis to design the values for the project, made the board layout that we decided to use, and wrote the code for calculating design values. He also simulated the individual subsystems to establish expected behavior (along with some isolator failure modes) for the test plan, and designed most of the test plan. He helped Liam Timmins with experimental testing and wrote the digital signal processing code.

Paul Karhnak, zxz2hm

This student simulated the Multisim schematics for the input filters, integrator, and V_{mid} sections and verified the component values on the PCB for the resistors and capacitors before experimental testing. In addition, he contributed critical aesthetic suggestions for the board layout.

Lourdes Leung, mqw6nf

This student assisted in preliminary experimental testing of the board and made notes of the testing performed by John Berberian and Liam Timmins. They also formatted the final project report and completed the write up for the simulation and assembly and testing sections.

Liam Timmins, uhj6qw

This student verified the experimental capabilities of the board alongside John Berberian, adjusting the input signals required to test each component of the board. They additionally contributed to the analytical designs for the isolator and verified the calculations behind the other designs, as well as simulate the instrumentation amplifier.

They also wrote the setup images, layout, and conclusion sections and made miscellaneous edits throughout the report.

0.2 Honor Pledge

On my honor as a student, I have neither given nor accepted unauthorized aid on this assignment or exam.



John E. Berberian, Jr.



Paul D. Karhnak

12/5/2023

Date

December 5, 2023

Date



Lourdes Leung



Liam Timmins

12/05/2023

Date

December 5, 2023

Date

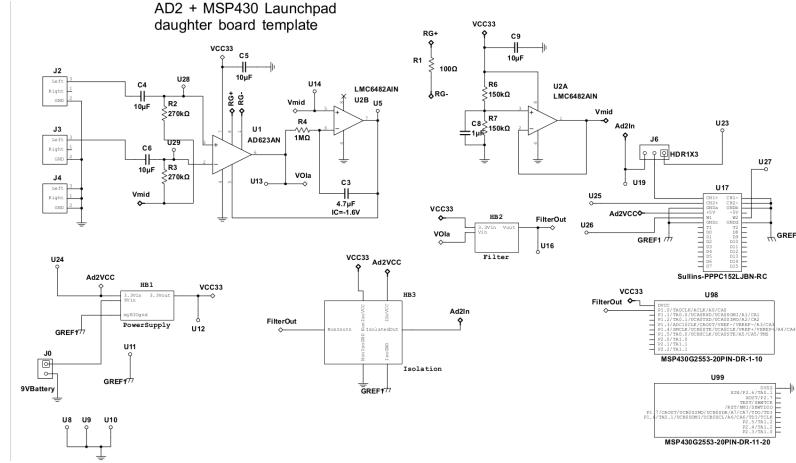


Figure 1: Full Project Schematic

1 Design Process

1.1 Background

We were given the basic layout for the project: an I-amp with an integrator to correct the DC offset, then a filter, and then an optocoupler. There's also a voltage regulator to act as the non-isolated power supply. The full schematic is shown in Fig. 1. The intent of the design is to amplify a small differential signal between the patient's wrists using the ankle as a ground reference, filter out any high frequencies to prevent aliasing during sampling, and couple the signal to a different ground reference for sampling.

1.1.1 Generic Components

To prevent damage to the ICs, we placed bypass capacitors next to the chips to buffer the power lines.

1.1.2 Power Supply

The power supply block is intended to supply a steady 3.3V power line to the rest of the board from a 9V battery. It uses a voltage regulator in the configuration recommended by the datasheet to achieve this (Fig. 3).

1.1.3 Input Filters

The input filters are intended to block the DC offset of the two wrists and re-reference the signals to V_{mid} . They do this with a simple RC high-pass.

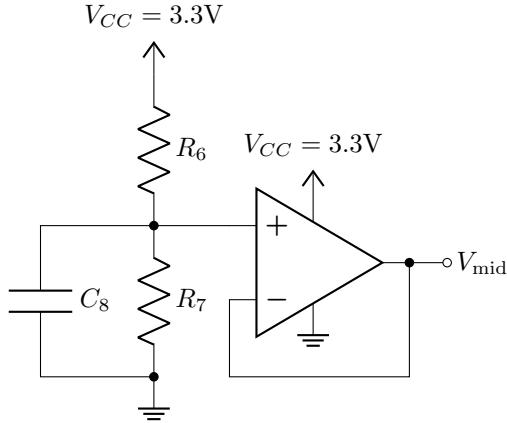


Figure 2: V_{mid} supply circuit. Bypass capacitor excluded from schematic.

1.1.4 Instrumentation Amplifier

We expect differential signals of up to 1mV in amplitude (though typically closer to $10 - 100\mu\text{V}$) so we need very high differential gain to amplify the signal to a readable level. The largest possible gain for the I-amp we were using is 1000, so we selected that.

1.1.5 Integrator

The integrator is intended to find the DC offset of the I-amp's output relative to V_{mid} and apply the necessary adjustment to center the signal on V_{mid} .

1.1.6 V_{mid} Supply

The V_{mid} supply is intended to be a DC reference around which the board's signals can be centered. Because we want to allow the signal to have as much space as possible both above and below the DC reference without hitting the rails, setting V_{mid} precisely in the middle of the rails seems like a good choice.

1.1.7 Butterworth Filter

The filter needs to attenuate higher frequencies to prevent aliasing during sampling without distorting the signal itself, so it should have a flat passband and a sharp drop-off. We were told that a fourth-order butterworth was required, and told that it should have around -72dB of attenuation at 500Hz.

1.1.8 Isolator

Almost all the subsystems use the common ground reference of the patient's ankle. This is a moving reference that is helpful for reducing noise in mea-

| RefDes | Value | Why |
|--------|-----------------|---------------|
| C5 | $10\mu\text{F}$ | U1 bypass cap |
| C7 | $10\mu\text{F}$ | U3 bypass cap |
| C9 | $10\mu\text{F}$ | U2 bypass cap |
| C14 | $10\mu\text{F}$ | U7 bypass cap |
| C16 | $10\mu\text{F}$ | U4 bypass cap |

Table 1: Values Designed in Section 1.1.1

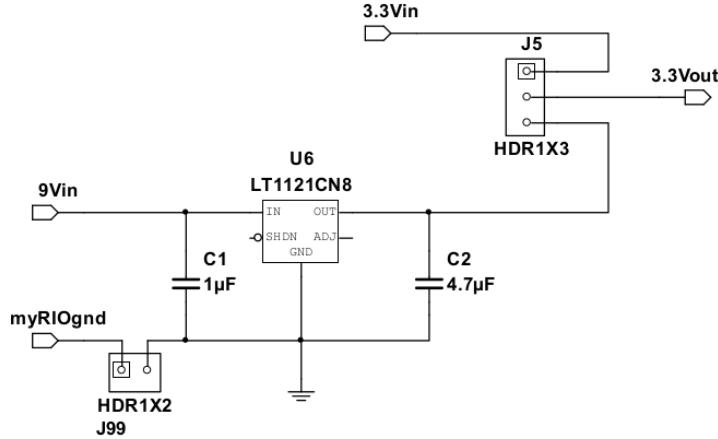


Figure 3: Power supply schematic block

surements, but makes sampling with external instruments (e.g. an oscilloscope) more difficult, because those have their own ground references, which tend to be set by the power grid. To make sampling easier, we need to re-reference the signal to the sampling instrument's ground - to couple it across the two ground references. We were given an optical isolator to use for that purpose.

1.2 Schematics

1.2.1 Generic Components

Some of the bypass capacitors are visible in Fig. 1. There are also bypass capacitors in the filter and isolator blocks. All bypass capacitors were chosen to be the largest value available, $10\mu\text{F}$. This should allow as much ripple (AC) as possible while holding the desired DC voltage. Summary: Tab. 1.

1.2.2 Power Supply

The schematic for the power supply is seen in Fig. 3. According to the datasheet, “a bypass capacitor in the range of $0.1\mu\text{F}$ to $1\mu\text{F}$ is sufficient” [1, p. 8] for the

| RefDes | Value | Why |
|--------|------------------|------------------|
| C1 | $1\mu\text{F}$ | U6 input bypass |
| C2 | $4.7\mu\text{F}$ | U6 output bypass |

Table 2: Values Designed in Section 1.1.2

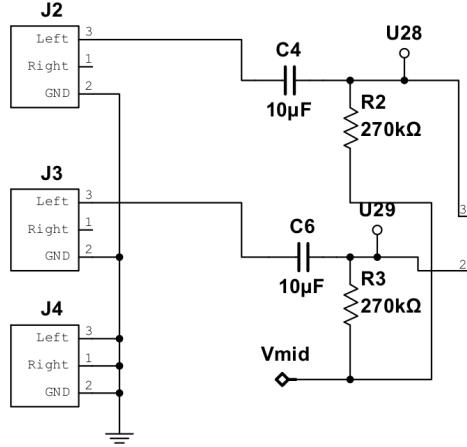


Figure 4: Input filter schematic block

input pin. We selected $C_1 = 1\mu\text{F}$. The output bypass capacitor is recommended to be $> 1\mu\text{F}$ if there is no series resistor [1, p. 10-11]. We selected $C_2 = 4.7\mu\text{F}$. Summary: Tab. 2.

1.2.3 Input Filters

The first section of the project board is the input filters shown in Fig. 4 that precede the instrumentation amplifier. Both are configured as first-order high pass filters that pass frequencies above about 0.05Hz received by the electrodes. These are then received as input to the instrumentation amplifier.

The input resistance is required to be between $250\text{k}\Omega$ and $300\text{k}\Omega$. The only resistor available to us between those is $270\text{k}\Omega$. The other condition is that the breakpoint should be between 0.05Hz and 0.1Hz. Using the code in Section 5.1, we found that only capacitor value available to us that works is $10\mu\text{F}$. Summary: Tab. 1.1.3.

1.2.4 Instrumentation Amplifier

The instrumentation amplifier schematic can be seen in Fig. 5. Using the code in Section 5.2, we found the peak-to-peak amplitude of the signal to be 1.108mV . In order to get the desired signal amplitude of around 1V , we would need a gain of 1000, the maximum the amplifier can supply [2, p. 1]. This corresponds to a

| RefDes | Value | Why |
|--------|---------------------|------------------------|
| C4 | $10\mu\text{F}$ | Input filter capacitor |
| C6 | $10\mu\text{F}$ | Input filter capacitor |
| R2 | $270\text{k}\Omega$ | Input filter resistor |
| R3 | $270\text{k}\Omega$ | Input filter resistor |

Table 3: Values Designed in Section 1.1.3

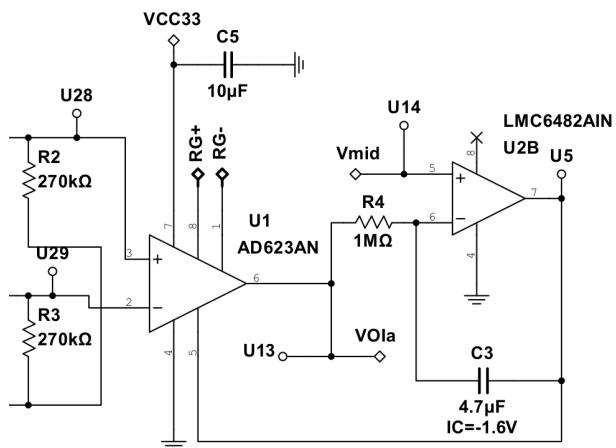


Figure 5: IAmp schematic block

| RefDes | Value | Why |
|--------|-------|---------------|
| R1 | 100Ω | Gain resistor |

Table 4: Values Designed in Section 1.1.4

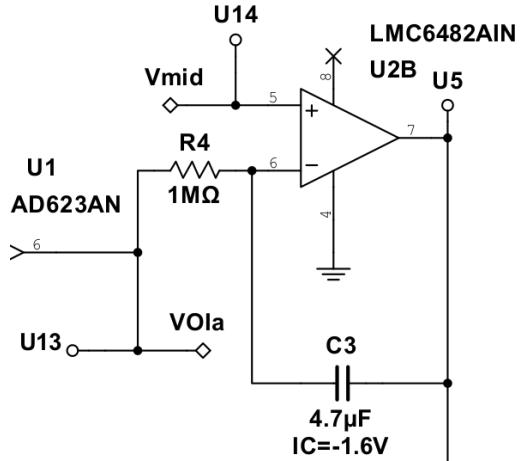


Figure 6: Integrator schematic block

gain resistor value of $R_1 = 100\Omega$ [2, p. 24]. Summary: Tab. 4.

Because we are using socketed chips, we can inject a signal at most points in the circuit to test functional blocks by themselves. We can remove the I-amp to test the input filters. We can inject a signal directly into the I-amp using the test points we placed to test the I-amp. We can remove the I-amp and inject a signal into its output node to test the filter, and we can remove the filter op-amp and inject a signal into its output to test the isolation. The only unknown left to us is the exact range of the signal from the user's skin, and thus the correct value of gain resistor to use. In light of this, we will leave the gain resistor off. We decided to set R1 as our DNI component.

1.2.5 Integrator

Next is the integrator whose schematic is shown in Fig. 6. This operates in the feedback loop of the instrumentation amplifier and attempts to bring signals to around V_{mid} . The integrator is shown in Fig. 6. We are given a restriction on the transfer function, so we'll derive the transfer function. All components are linear (ideally), so we can use superposition. Initially zeroing V_{mid} , this looks like an inverting amplifier:

$$V_{out1} = V_{in} \cdot -\frac{Z_{C3}}{R_4} = -\frac{1}{sR_4C_3}V_{in}$$

| RefDes | Value | Why |
|--------|-------------------|----------------------------------------------|
| C3 | $4.7\mu\text{F}$ | Integrator capacitor, $R_4C_3 = 4.7\text{s}$ |
| R4 | $1\text{M}\Omega$ | Integrator resistor, $R_4C_3 = 4.7\text{s}$ |

Table 5: Values Designed in Section 1.1.5

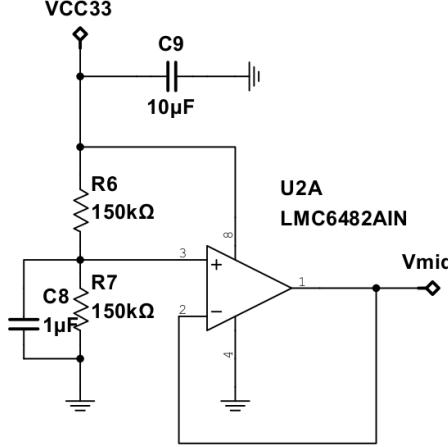


Figure 7: Vmid schematic block

Now zeroing V_{in} , the circuit is a non-inverting amplifier:

$$V_{\text{out}2} = V_{\text{mid}} \cdot \left(1 + \frac{Z_{C3}}{R_4}\right) = V_{\text{mid}} \cdot \left(1 + \frac{1}{sR_4C_3}\right)$$

Adding the two:

$$V_{\text{out}} = V_{\text{out}1} + V_{\text{out}2} = V_{\text{mid}} - \frac{1}{sR_4C_3} (V_{\text{in}} - V_{\text{mid}})$$

This looks like we're re-setting the ground reference to V_{mid} , and then applying an inverting low-pass to the signal difference, which intuitively checks out with what we intend to do: correct the DC offset of the signal using the I-amp's reference terminal. The restriction we're given translates to $1 \leq R_4C_3 \leq 10$ (seconds). The code in Section 5.3 gave us a list of possible values, from which we chose an intermediate value of $R_4C_3 = 4.7\text{sec}$, corresponding to $R_4 = 1\text{M}\Omega$, $C_3 = 4.7\mu\text{F}$. Summary: Tab. 5.

1.2.6 V_{mid} Supply

The next section of the board is the V_{mid} supply whose schematic can be seen in Fig. 7. The supply circuit for V_{mid} is shown in Fig. 2. We are given a few restrictions, so we'll derive the relevant properties. The Thevenin-equivalent

| RefDes | Value | Why |
|--------|---------------------|----------------------------------|
| C8 | $1\mu\text{F}$ | V_{mid} buffer cap |
| R6 | $150\text{k}\Omega$ | V_{mid} voltage divider |
| R7 | $150\text{k}\Omega$ | V_{mid} voltage divider |

Table 6: Values Designed in Section 1.1.6

resistance seen by the + terminal of the op-amp is just the two resistors in parallel:

$$R_{\text{th}+} = R_6 \parallel R_7 = \frac{R_6 R_7}{R_6 + R_7}$$

The time constant τ is C_8 multiplied by the equivalent resistance seen by the capacitor, which conveniently is $R_{\text{th}+}$:

$$f_c = \frac{1}{2\pi\tau} = \frac{1}{2\pi C_8 (R_6 \parallel R_7)}$$

When there's no ripple in the supply voltage, the circuit is just a buffered voltage divider, so we can find V_{mid} :

$$V_{\text{mid}} = \frac{R_7}{R_6 + R_7} V_{CC}$$

Just to confirm our intuitive results, we can do a more formal analysis. The whole circuit is a voltage divider with complex impedance:

$$\begin{aligned} H(s) &= \frac{R_7 \parallel Z_{C8}}{R_6 + (R_7 \parallel Z_{C8})} \\ &= \frac{\left(sC_8 + \frac{1}{R_7} \right)^{-1}}{R_6 + \left(sC_8 + \frac{1}{R_7} \right)^{-1}} \\ &= \frac{1}{sC_8 R_6 + \frac{R_6}{R_7} + 1} \\ &= \frac{1}{sC_8 R_6 + \frac{R_6 + R_7}{R_7}} \\ &= \frac{R_7}{R_6 + R_7} \cdot \frac{1}{sC_8 (R_6 \parallel R_7) + 1} \end{aligned}$$

So our intuitive analysis checks out with the more formal impedance approach. Using the program in Section 5.4, we find that one set of values that satisfies all the conditions is $R_6 = 150\text{k}\Omega$, $R_7 = 150\text{k}\Omega$, $C_8 = 1\mu\text{F}$. This gives $R_{\text{th}+} = 75\text{k}\Omega$, $f_c = 2.122\text{Hz}$, $H(\omega = 0) = \frac{1}{2}$.

1.2.7 Butterworth Filter

The next section of the project board is the fourth-order low-pass Butterworth filter as seen in Fig. 8. This section provides the antialiasing function of the

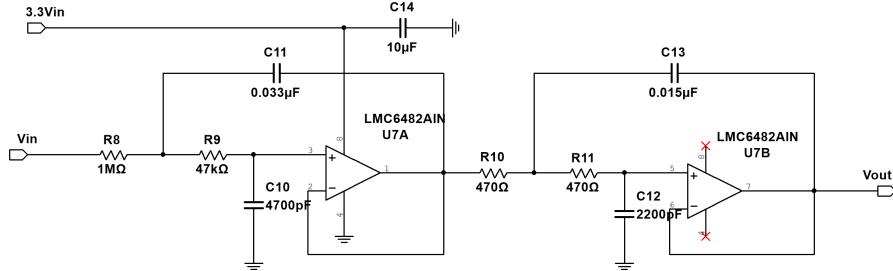


Figure 8: Butterworth filter schematic block

board. The ideal transfer function for a unity-gain fourth-order Butterworth is:

$$|H(j2\pi f)| = \frac{1}{\sqrt{1 + (f/f_c)^8}}$$

Plugging in our values, we can find the required corner frequency:

$$10^{-72/20} = \frac{1}{\sqrt{1 + (500/f_c)^8}}$$

$$f_c = \frac{500}{\sqrt[8]{10^{7.2} - 1}} \approx 62.946\text{Hz}$$

So the corner frequency should be at most 62.946Hz. The Q -values for the two Sallen-Key filter sections are 0.5412 and 1.3065 [3, tab. 1]. Using the second simplification from the Sallen-Key design document [4, p. 5], we wrote a program to brute-force all possible combinations of resistors and capacitors for each section (Section 5.5). Because of the wide selection of capacitors available, we were able to put a very tight constraint on the Q value - it must be within 0.008 of the desired value. The corner frequency requirement was less stringent - within $0.1f_c \approx 6.3\text{Hz}$. The reasoning for this was that any error in the Q values can lead to signal distortion, while error in f_c (as long as the two sections match) only slightly changes which frequencies are passed through. We found a solution for both filter sections with matching f_c values (58.947Hz) and very close Q values: 0.5487 and 1.3056. Summary: Tab. 7.

1.2.8 Isolator

The last section is the isolator which can be seen in Fig. 9. The input section of the optocoupler is shown in Fig. 10. The circuit has negative feedback, because an increase in the LED current will lead to an increase in i_{pd1} , which increases

| RefDes | Value | Why |
|--------|---------------|-------------------------------|
| C10 | 4700pF | C_1 for first-stage filter |
| C11 | 0.033 μ F | C_2 for first-stage filter |
| C12 | 2200pF | C_1 for second-stage filter |
| C13 | 0.015 μ F | C_2 for second-stage filter |
| R8 | 1M Ω | R_1 for first-stage filter |
| R9 | 47k Ω | R_2 for first-stage filter |
| R10 | 470k Ω | R_1 for second-stage filter |
| R11 | 470k Ω | R_2 for second-stage filter |

Table 7: Values Designed in Section 1.1.7

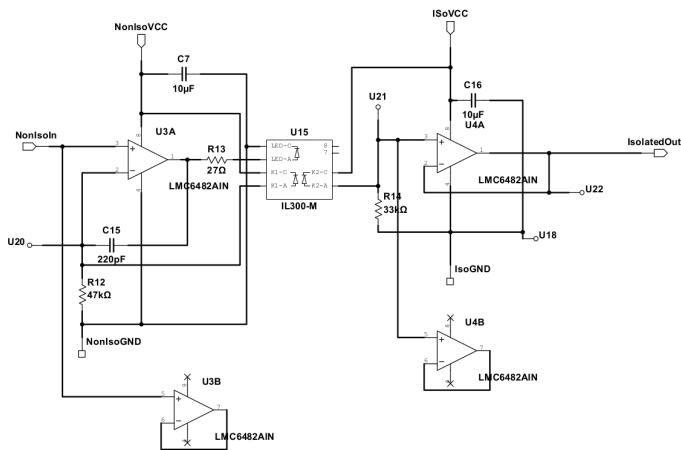


Figure 9: Isolator schematic block

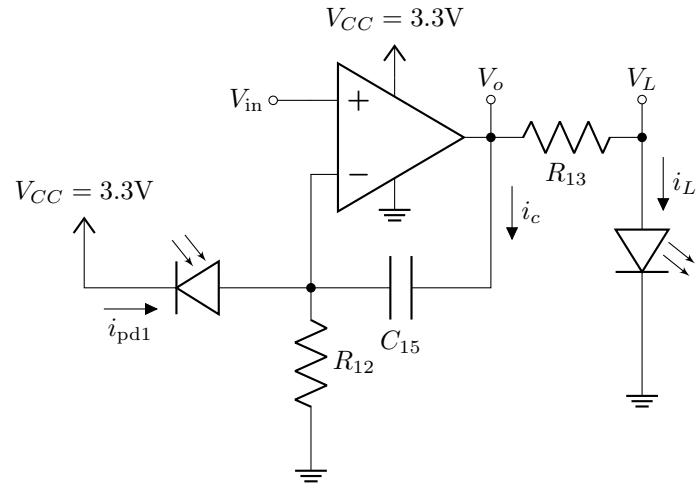


Figure 10: Optocoupler Input Feedback Circuit

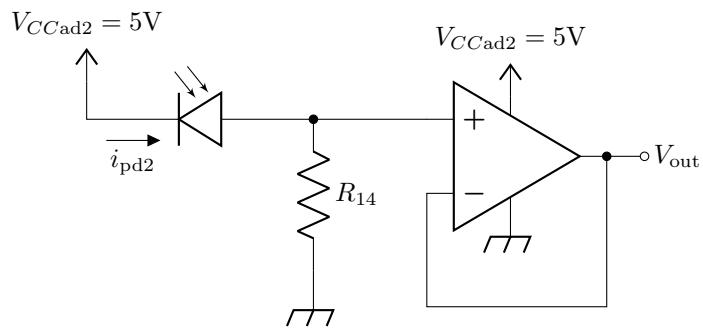


Figure 11: Optocoupler Output Buffer Circuit

the voltage at the inverting terminal. Because of the negative feedback, $V_- = V_{\text{in}}$ (ignoring offset voltage). So:

$$V_- = (i_c + i_{\text{pd1}}) R_{12} = V_{\text{in}}$$

$$i_{\text{pd1}} = \frac{V_{\text{in}}}{R_{12}} - i_c$$

The capacitor current is given by:

$$i_c = \frac{V_o - V_{\text{in}}}{Z_{C15}}$$

Substituting:

$$i_{\text{pd1}} = \frac{V_{\text{in}}}{R_{12}} - \frac{V_o - V_{\text{in}}}{Z_{C15}}$$

We know that $i_{\text{pd2}} = k_3 i_{\text{pd1}}$, so we can find an expression for the output voltage:

$$\begin{aligned} V_{\text{out}} &= i_{\text{pd2}} R_{14} \\ &= k_3 i_{\text{pd1}} R_{14} \\ &= \frac{k_3 R_{14}}{R_{12}} \cdot V_{\text{in}} - \frac{k_3 R_{14}}{Z_{C15}} (V_o - V_{\text{in}}) \end{aligned}$$

Because of the limits of the power rails, $0V \leq V_o - V_{\text{in}} \leq 3.3V$. The optocoupler comes after the low-pass filter, so we can assume the highest frequency we'll see in our signal is around 59Hz. We were told that C_{15} should be on the order of hundreds of pF, so $|Z_C| \sim 5M\Omega$. The optocoupler is in the I bin, so $k_3 \approx 1.3765$ [5, p. 3]. The typical maximum i_{pd2} is $120\mu\text{A}$ [5, p. 2], but to give us some headroom for part variation we'll assume the maximum is $90\mu\text{A}$. We know that the range of the signal is 0V to 3.3V, so to achieve unity gain on the output side we need R_{14} to be at least $3.3V/90\mu\text{A} \approx 36.7k\Omega$. We expect R_{14} to be in the same order of magnitude as that, though - much less than $5M\Omega$, so we can cross off the second term in the V_{out} as negligible:

$$V_{\text{out}} \approx \frac{k_3 R_{14}}{R_{12}} \cdot V_{\text{in}}$$

So the gain is set (approximately) by the ratio of R_{14} to R_{12} . We would like the optocoupler to have approximately unity gain, so $V_{\text{out}} = V_{\text{in}}$:

$$\begin{aligned} V_{\text{in}} &= \frac{k_3 R_{14}}{R_{12}} \cdot V_{\text{in}} \\ R_{12} &= k_3 R_{14} \end{aligned}$$

One combination that satisfies this (and the minimum value constraint) is $R_{12} = 47k\Omega$, $R_{14} = 33k\Omega$. This gives an inverse ratio of 1.424, which is within the range for I-bin optocouplers.

We were given a few more hints for resistor values: first, that the max signal should give us something near the max current for the op-amp. The max current

| RefDes | Value | Why |
|--------|-------|---------------------------|
| R12 | 47kΩ | Optocoupler gain resistor |
| R14 | 33kΩ | Optocoupler gain resistor |
| R13 | 27Ω | Optocoupler LED resistor |
| C15 | 220pF | Optocoupler capacitor |

Table 8: Values Designed in Section 1.1.8

is 30mA [6, p. 3], so to give us a bit of headroom we'll choose 20mA. Second, we were told that R_{13} should drop around 0.5V when operating at max current. With a bit of division, we find that this corresponds to a value of $R_{13} = 25\Omega$. We don't have that available to us, the closest one is $R_{13} = 27\Omega$. Summary: Tab. 8.

1.3 Multisim Simulations

1.3.1 Input Filters

To verify our analytical calculations for our board components, we tested different sections of the circuit for the expected time domain plots and frequency responses. Starting with the input filters, we ran a simulation in Multisim to verify their frequency response. Its schematic is seen in Fig. 4. We saw a high pass filter with corner frequency at around 0.0589Hz (Fig. 12) which meets our breakpoint frequency condition between 0.05Hz and 0.1Hz.

1.3.2 Instrumentation Amplifier

To verify the functionality of the instrumentation amplifier, we first simulated it with no gain resistor to verify its unity gain output. A 60 Hz 0.5V signal was summed with a 300Hz 0.25V signal and compared to a 60Hz 0.5V signal. The expected output for this case should be a 300Hz 0.25V signal, which was approximately observed in Fig. 13, with an output amplitude of 0.2498V.

Next, the gain resistor was set to verify our previous calculations for it. The same 60Hz 0.5V signal was input as the common mode, but now a differential mode of 1mV at 300Hz was input. With this, a 0.904V output amplitude was observed in Fig. 14. With the 100Ω gain resistor, an amplitude of approximately 1V would be expected for that differential mode input amplitude. Although there is some difference between the ideal and simulated circuits, for our purposes this discrepancy should be insignificant.

1.3.3 Integrator

Next, we ran simulations for the integrator whose schematic can be seen in Fig. 6. Because making a bode plot would be time-consuming, we tested the integrator with 1kHz sinusoidal inputs with DC offsets above and below V_{mid} . When the DC offset is below V_{mid} , we expect the output to go to the positive

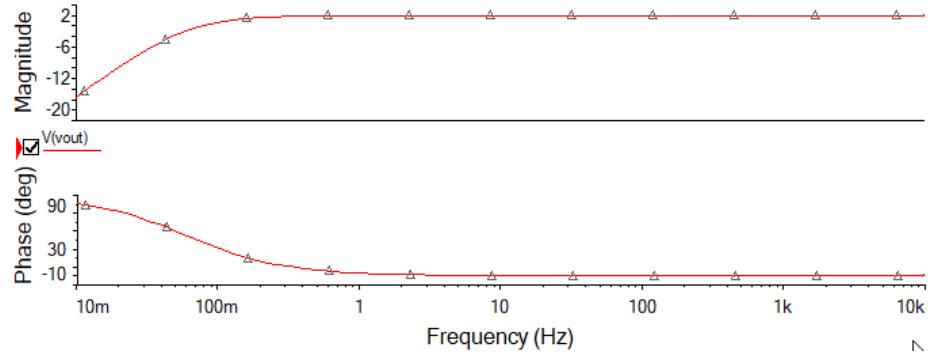


Figure 12: Simulated Bode plot of input filter.

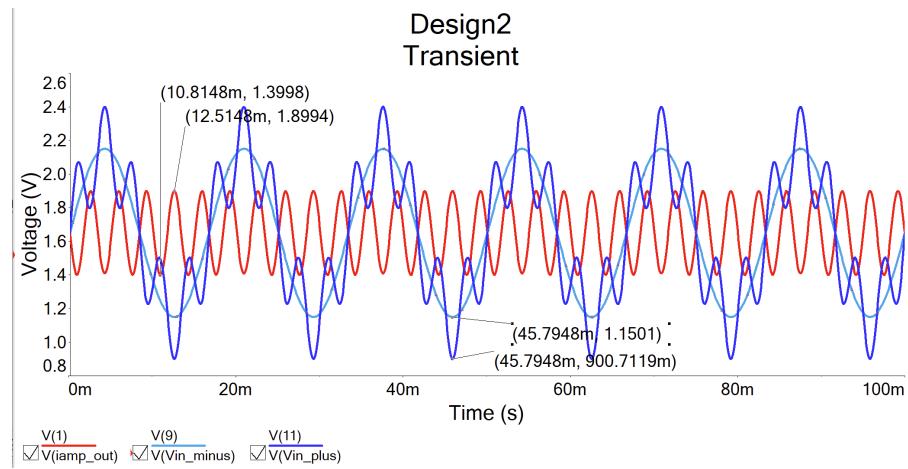


Figure 13: Simulation of instrumentation amplifier without a gain resistor. The common mode is set to 0.5V amplitude with 60Hz frequency and the differential mode is set to 0.25V amplitude and 300Hz frequency.

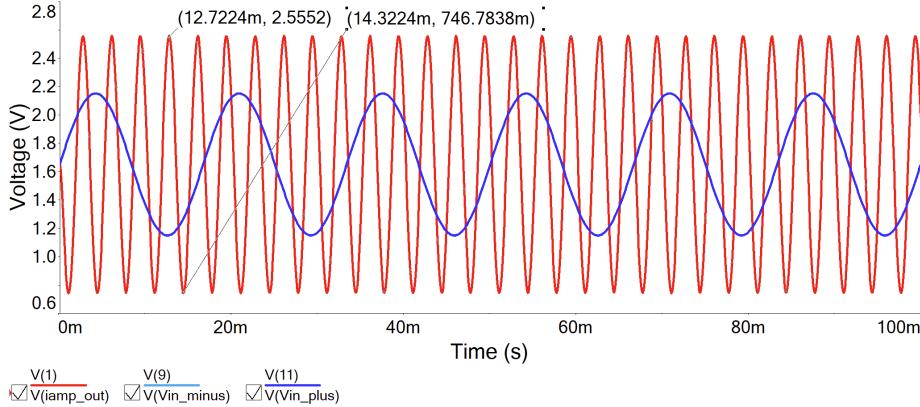


Figure 14: Simulation of instrumentation amplifier with a 100Ω gain resistor. The common mode is set to 0.5V amplitude with 60Hz frequency and the differential mode is set to 1mV amplitude and 300Hz frequency.

rail which we see in Fig. 15. When the DC offset is above V_{mid} , we expect the output to go to the negative rail which we see in Fig. 16.

To help diagnose problems when testing the board experimentally, we ran a few simulations for integrator fault conditions. Inserting the op-amp backwards into the socket, we would see an output like that in Fig. 17. If we had a defective capacitor, we would see something similar to Fig. 18. If the resistor was too small, we would see something similar to Fig. 19.

1.3.4 V_{mid} Supply

Next we ran verifications for the V_{mid} supply whose schematic can be seen in Fig. 7. The simulation result of the step response seen in Fig. 20 for V_{mid} matches what was calculated analytically. We expected a step response that starts off with a standard RC curve that eventually settles to about 1.65V which is half of the input voltage,

1.3.5 Butterworth Filter

Next, we ran simulations for the first section of the Butterworth Filter whose schematic can be seen in Fig. 8. We expected to see an underdamped step response which was verified in Multisim as seen in Fig. 21. We also expected to see a low pass frequency response with a cutoff frequency at 58.947Hz and 74.3dB of rejection at 500Hz which was verified by our simulation as seen in Fig. 22. Performing the numerical verification, we generated a bode plot with a corner frequency of about 60Hz as seen in Fig. 22.

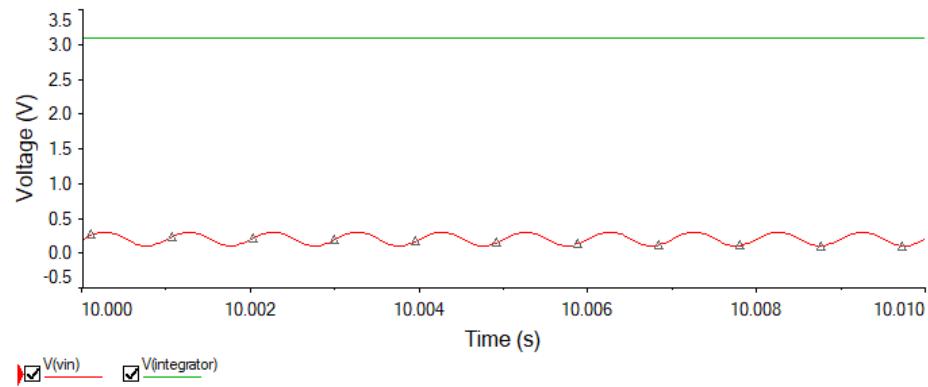


Figure 15: Integrator: Low DC Offset

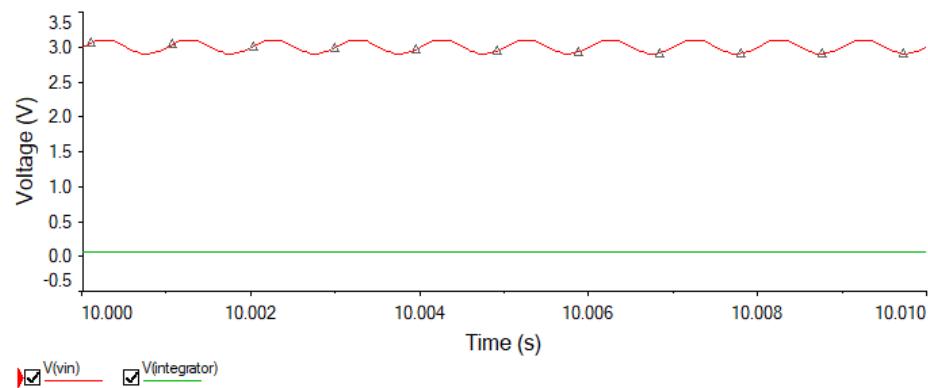


Figure 16: Integrator: High DC Offset

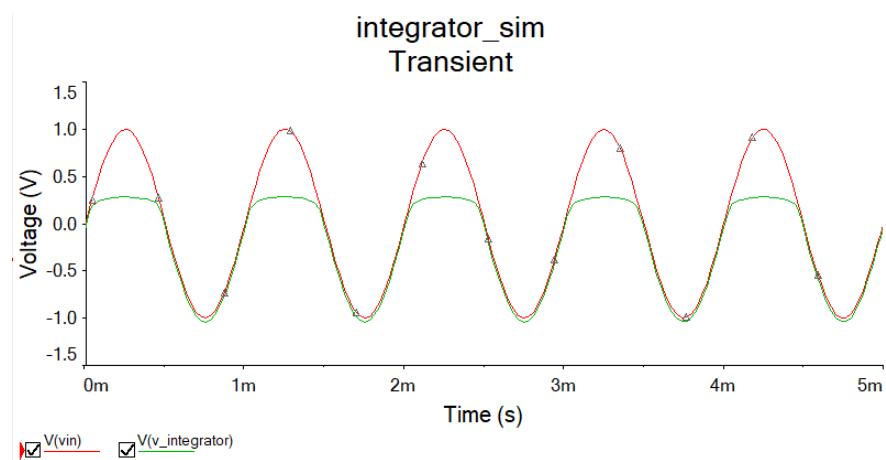


Figure 17: U2 op-amp reversed fault condition simulation results.

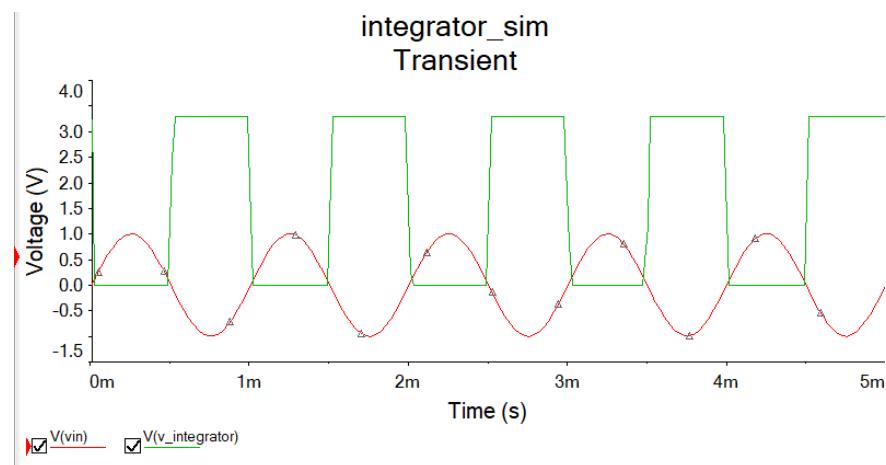


Figure 18: Integrator fault condition with defective capacitor.

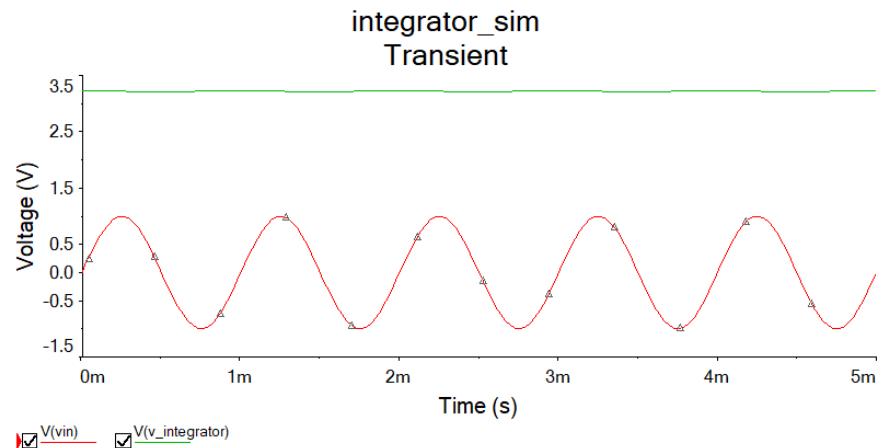


Figure 19: Integrator when resistor is too small.

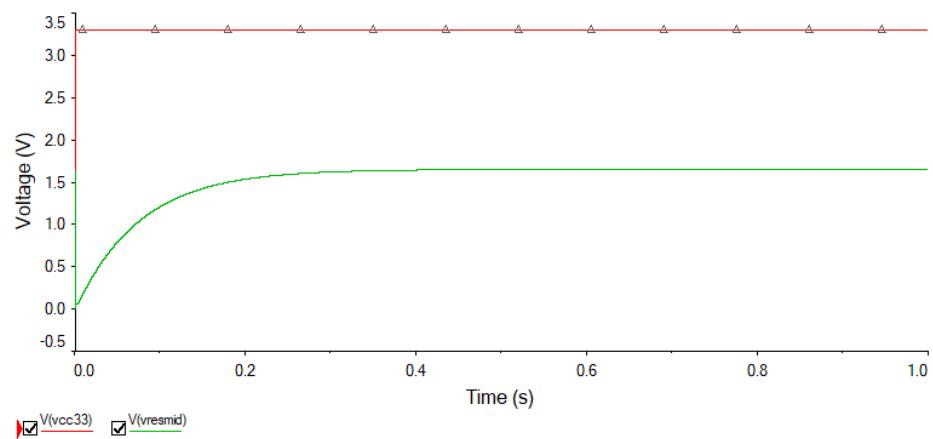


Figure 20: Multisim transient simulation for step response of V_{mid} . Amplitude of V_{CC} is 3.3V and that of V_{mid} is about 1.6V, half of V_{CC} .

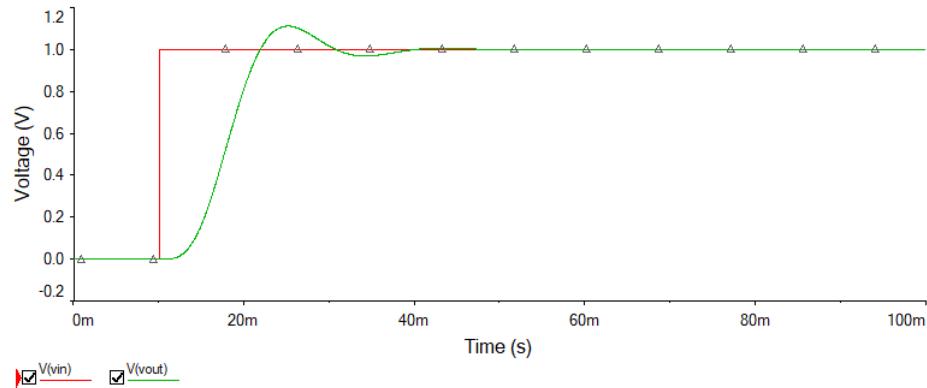


Figure 21: Simulation of step response of Butterworth filter

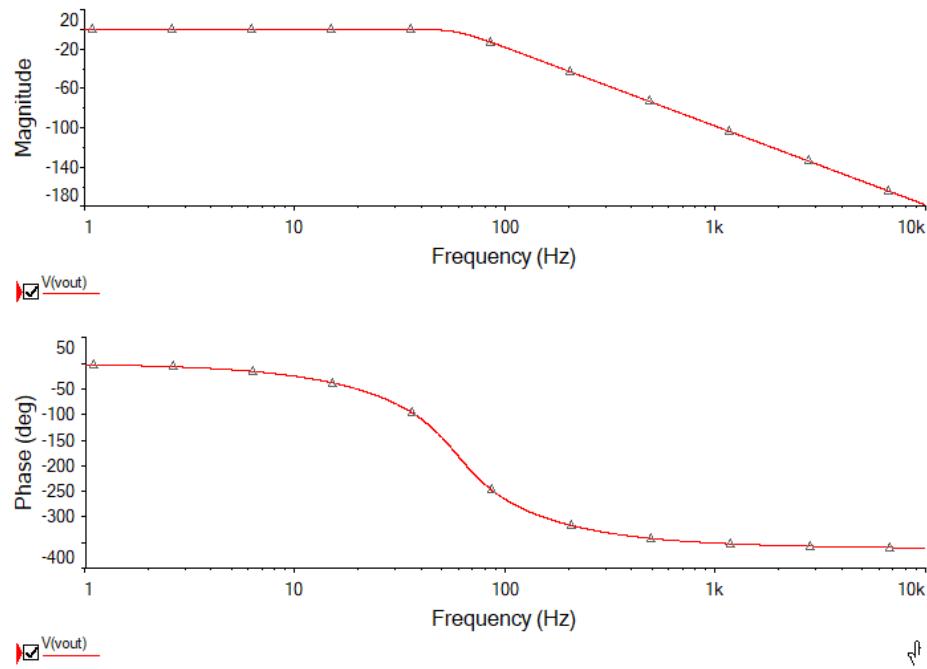


Figure 22: Frequency Response of Butterworth filter

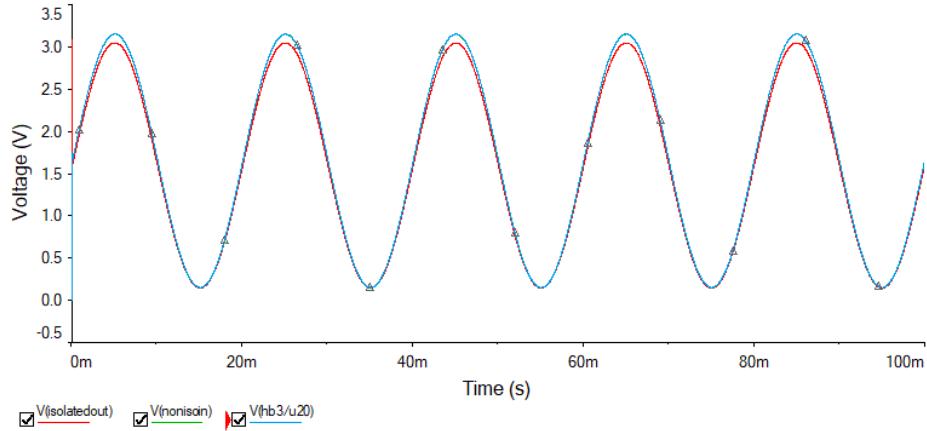


Figure 23: Numerical simulation of isolator.

1.3.6 Isolator

For the isolator, we applied various sinusoidal signals with 1.65VDC, $f < 62\text{Hz}$, and $V_{pk-pk} < 3\text{V}$ expecting the signal on U20 and U21 to be identical to the input and the signal on U22 identical to the input with some scaling. The isolator schematic can be seen in Fig. 9. The simulated response can be seen in Fig. 23 which is what was expected.

We also simulated several fault conditions for the isolator to help diagnose issues that may arise in experimental testing. If the overall gain is too high, we would see scaling and clipping of the output but still perfect scaling of the input (Fig. 24). If the overall gain is too low, we would see the output signal scaled down but still perfect scaling of the input (Fig. 25). If R_{12} is too small, we would see clipping on U20 and large scaling on the output (Fig. 26). If R_{13} is too large, we would see clipping on U20 and the ouput (Fig. 27).

1.3.7 Individual section simulation overview

For all of the simulations performed above, the results generally corresponded to the analytical calculations performed in the previous sections. Any discrepancies observed, such as in the simulation of the instrumentation amplifier, were noted and found to be satisfactory for the purposes of the overall schematic. With this verification, a system level simulation can be performed.

1.3.8 Complete EKG Simulation

The full-system simulation was conducted with the EKG data provided. The results are shown in Fig. 28.

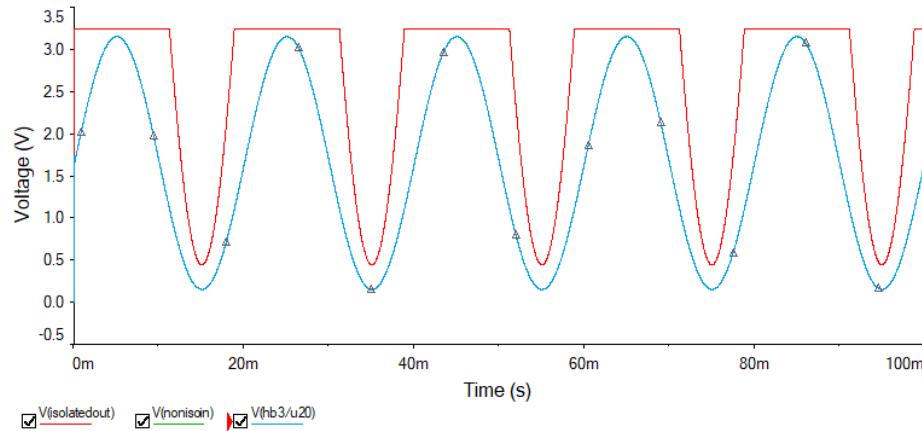


Figure 24: Isolator: Gain too large $\left(k_3 \cdot \frac{R_{14}}{R_{12}} > 1\right)$

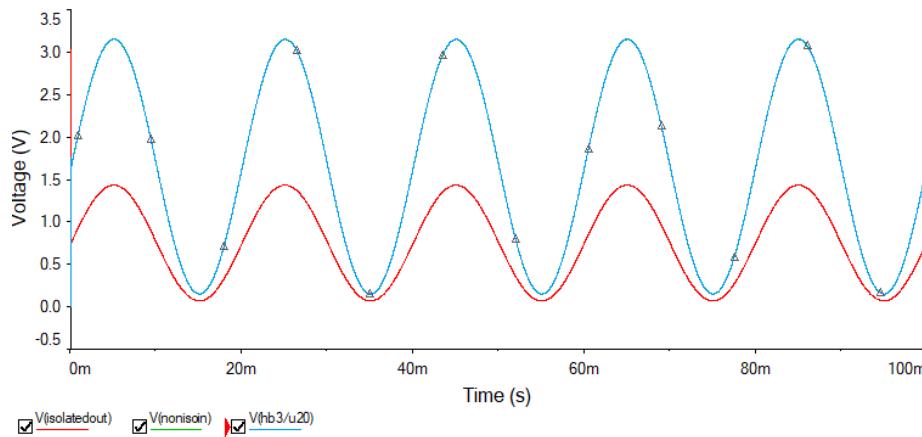


Figure 25: Isolator: Gain too small $\left(k_3 \cdot \frac{R_{14}}{R_{12}} < 1\right)$

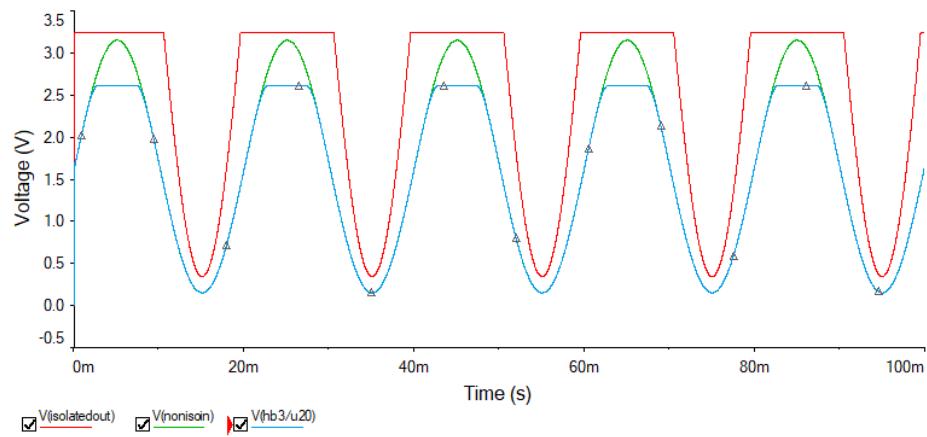


Figure 26: Isolator: R_{12} too small

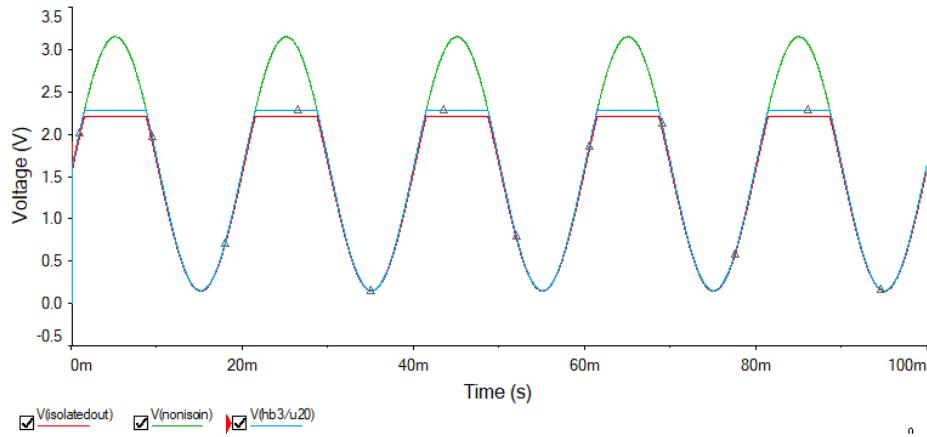


Figure 27: Isolator: R_{13} too large

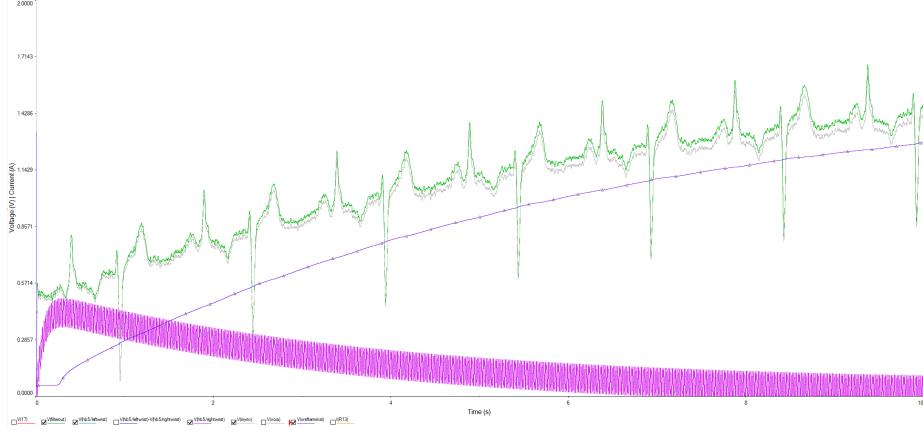


Figure 28: Full-system simulation with LVM data

1.4 Layout

For the board layout in Fig. 29, there are several important elements to note. C5, C7, C9, C14, and C16 are bypass capacitors and were placed within close proximity to each of their respective ICs. The power traces were made to be as short as possible by moving the relevant components close together, so as to eliminate unnecessary noise within the system. On the mechanical layer, the component designations and values were listed to make sure that the manufacturers would be able to properly build the board. For the sake of organization, each component was placed as closely as possible with other components from its design section. Although there are some components which break this rule, in general it was made such that components would align nicely with one another. As a general rule, power routing was done on the copper top and signal routing was reserved for the bottom. However, due to the complexity of the circuit topology, it wasn't always possible to keep this rule, so it is broken in some places (most notably the upper left of the board). The reference designators printed on the silkscreen top were placed in the same orientation relative to the component across the board, further limiting confusion when reading the board. Finally, test points were placed in locations where they could be accessed relatively easily to streamline the experimental verification process.

Following the completion of the board, a series of tests were performed to verify that it would operate as expected. First, connectivity and DRC and netlist tests were performed to verify nothing more needed to be connected or was too close together. Next, the Ultiboard file was exported with the provided UVA class CAM export. The exported folder was then renamed with a CAM file renamer, and sent to FreeDFM to provide a more complete analysis and verification of the board layout. This resulted in the reply in Fig. 30, verifying no critical issues were present in the design. With this verification, the PCB

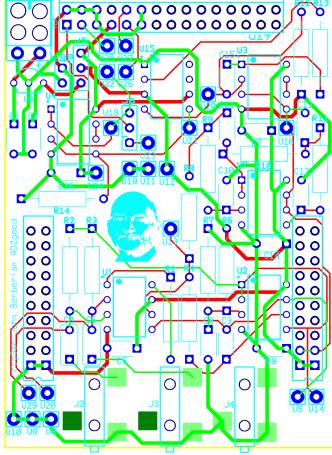


Figure 29: Copper top, copper bottom, and silkscreen layers of PCB.

was ready to be manufactured.

2 Experimental

2.1 Assembly and Testing

2.1.1 Preliminary Tests

After receiving the physical PCB, we ran preliminary tests and tested the different sections of the board. First, we checked connectivity at each node with the multimeter. After verifying connectivity, we ensured that the installed resistors and capacitors were of the right values. When inserting chips into the board, each team member verified the orientation of the chip. Next, we test that the power supply worked correctly by connecting the battery and measuring 3.3V output at the bottom of J5. We also introduced AC noise into the input, ensuring that we still received a 3.3V output.

2.1.2 Isolator

Next, we verified the functionality of the isolator by applying various sinusoidal signals with a 1.65V DC offset, 3Vpk, and 62Hz and seeing if the signals at U20 and U21 were identical to the input and that the signal at U22 was similar to the input, but with some scaling. Initially, we did not measuring the expected responses, but realized the problem was that we did not shunt U3 or U4.



Figure 30: FreeDFM reply verifying board layout.

2.1.3 V_{mid}

Preparing to test V_{mid} , we made sure that it was between the rails at U14 after displaying an RC exponential curve at start-up. The experimental result may be seen in Fig. 31 and matches what we expected from our calculations and simulation after settling from its RC exponential-like curve.

2.1.4 Butterworth Filter

Next, we test the functionality of the Butterworth Filter, observing the frequency response with the scope channels at the filter input and output, U13 and U16, respectively. From our analytical calculations, we expect a corner frequency of 58.947Hz and rejection of 74.3dB at 500Hz. For the experimental testing, we generated a bode plot as seen in Fig. 32 with a corner frequency of about 61.626Hz and rejection of 72.150dB at 499.52Hz, which matched our analytical and numerical values closely. Initially, the output of the filter was railing, but this was from procedural error where we forgot to add a voltage offset.

2.1.5 Input Filters

We then tested the input filters by verifying their corner frequencies. For the positive input filter, we measured the input at J3 and output at U28. The transient passband response may be seen in Fig. 33. For the negative input filter, we measured the input at J2 and output at U29. The transient passband response may be seen in Fig. 34. For the DC conditions, the voltage read with the multimeter was 1.618V for the positive input filter, and 1.624V for the negative input filter. These results match what was expected, with the signal is approximately centered around V_{mid} . When testing the input filters, we initially

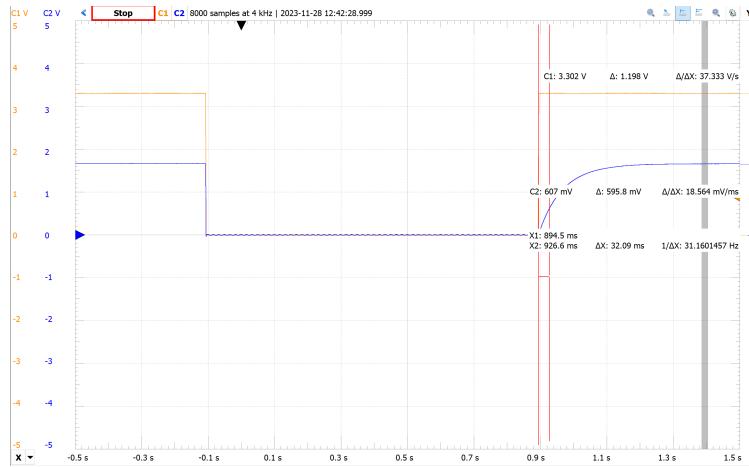


Figure 31: Experimental step response for V_{mid} . Amplitude of VCC is 3.302V and that of V_{mid} is 1.661V, half of VCC.

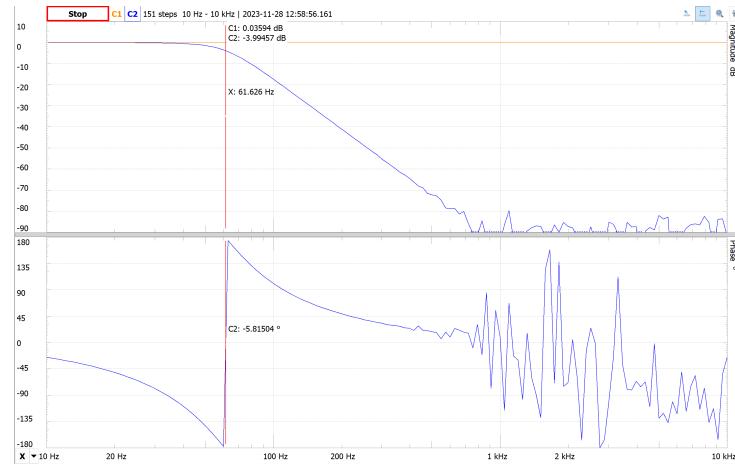


Figure 32: Experimental Bode plot of Butterworth filter design. The corner frequency was measured as 61.626Hz.

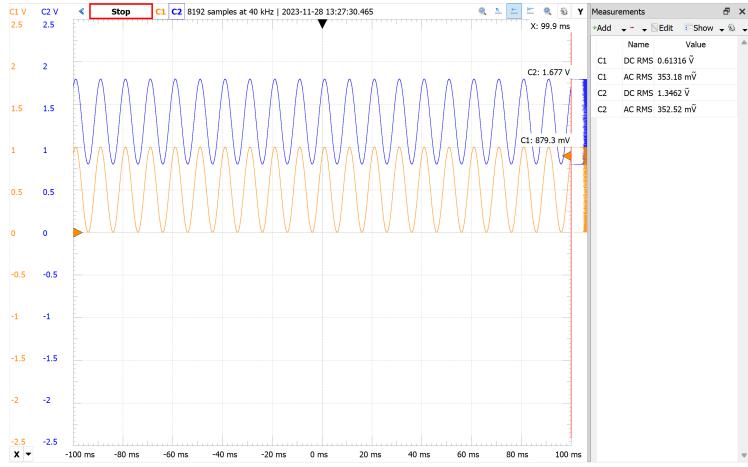


Figure 33: Scope view for the positive input filter.

had issues where current was running through the resistors R2 and R3 when there should have been zero current due to the dangling capacitors. We found that current was being drawn in the scope probes. This issue was resolved after switching from using Liam Timmins's AD2 to Paul Karhnak's AD2.

2.1.6 Integrator

Next, we tested the integrator section by generating an input signal on U13, checking the V_{mid} signal on U14, and verifying the output on U5. Verification involves adding a DC offset less than V_{mid} that should hit the positive rail and a DC offset greater than V_{mid} that should hit the negative rail.

2.1.7 Instrumentation Amplifier

Lastly, we tested the instrumentation amplifier. For the input, we sum a 60Hz noisy signal of a 100mV amplitude and a 1kHz signal of a 50mV amplitude. This, we do experimentally by constructing a summer amplifier. We input the the noisy signal and the output of the summer at U29 and U28 and measure the output at U13.

For this verification, we had initially planned to have a 1V common mode amplitude with 1mV differential mode amplitude, but found that the AD2 could not reliably produce signals this small. Signals of a large size would end up peaking as they met the rails. To verify the functionality of the instrumentation amplifier, we instead removed the gain resistor of 100Ω , allowing the results to be measured in unity gain. The results of this are visible in Fig. 37

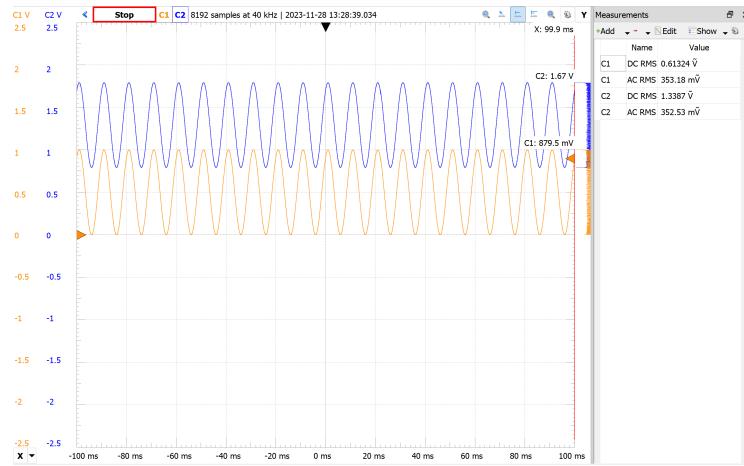


Figure 34: Scope view for the negative input filter.

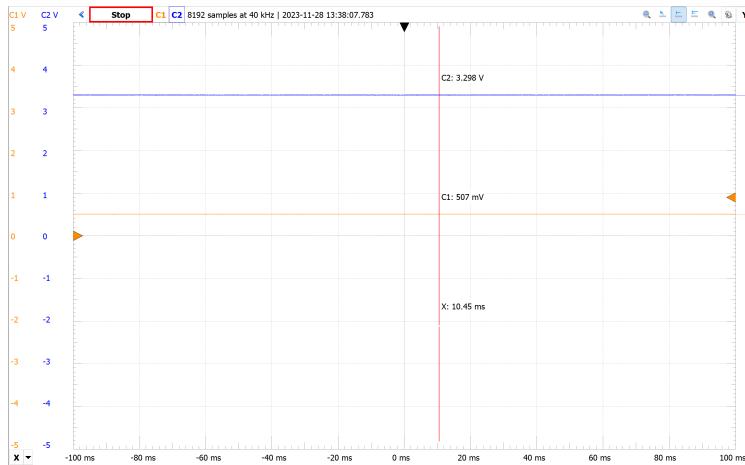


Figure 35: Integrator output hits the positive rail at 3.298V when the DC offset is below V_{mid} .

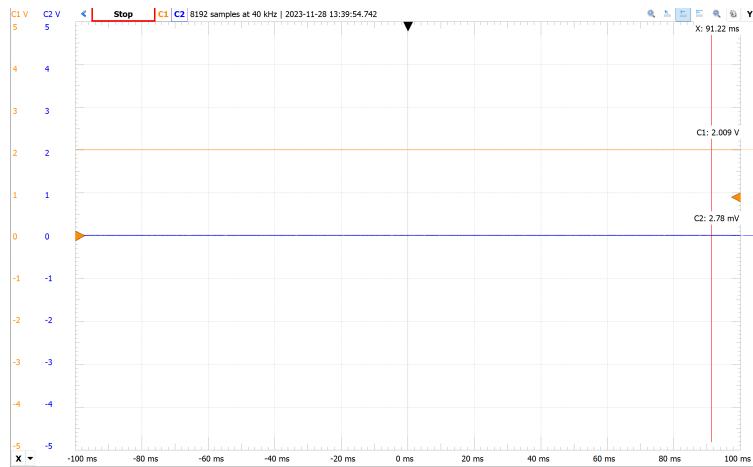


Figure 36: Integrator output hits the negative rail at 2.78mV when the DC offset is above V_{mid} .

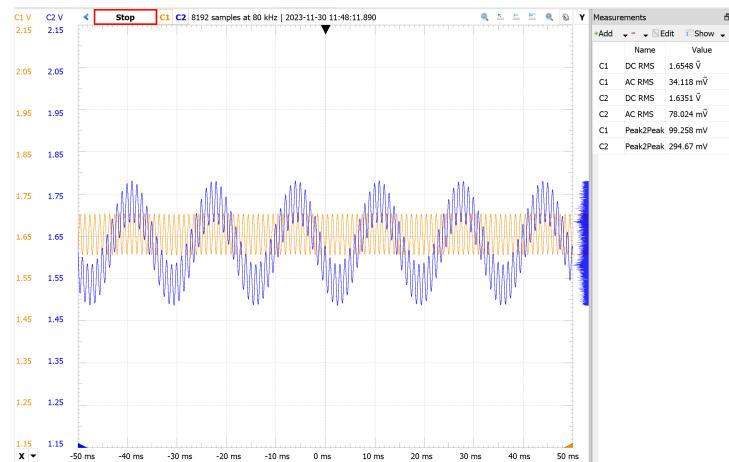


Figure 37: Experimental verification of instrumentation amplifier with 100mV 60 Hz common mode and 50mV 1kHz differential mode.



Figure 38: Electrodes attached to John Berberian’s wrists and ankle to generate the signal visible in Fig. 41.

2.2 Setup Images

Once we verified that the project board operated as expected, we attached the electrodes to the input visible in Fig. 38 and inserted the AD2 header into the AD2. The two wrist electrodes were plugged into J2 and J3 to generate the differential signal, while the ankle electrode served as a ground reference and was plugged into J4. The scope channel was then set to observe the signal with 2 seconds per division to observe 20 seconds of data.

In Fig. 41, Fig. 42, and Fig. 43, the heart beat can be clearly observed, with varying levels of interference visible. Fig. 43 was recorded in a different location from the other two, potentially contributing to the 60 Hz noise visible. Additionally, it was later found that the proximity of the power block to the laptop running waveforms had an observable effect on the 60Hz noise generated, further explaining the discrepancies between the data generated. However in all cases, the peaks representing the heartbeat are clearly visible.

2.3 Data Capture

2.3.1 Data Collection

After verifying the functionality of each component, the board was used to generate signals for John Berberian, Paul Karhnak, and Liam Timmins in Fig. 41, Fig. 42, and Fig. 43, respectively.

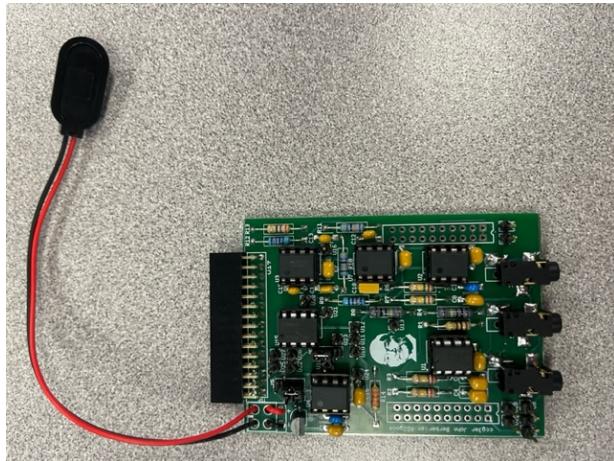


Figure 39: Front view of project PCB.

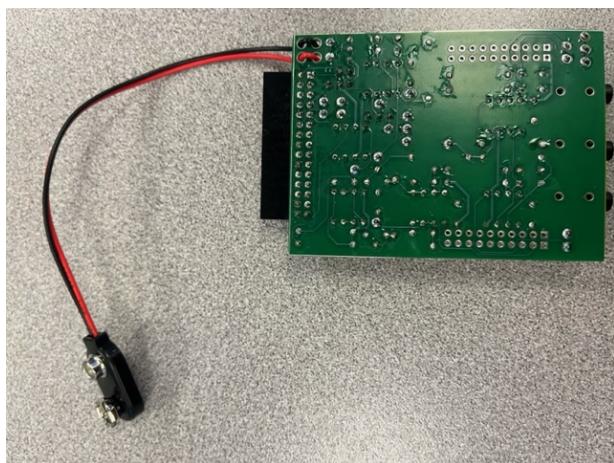


Figure 40: Back view of project PCB.

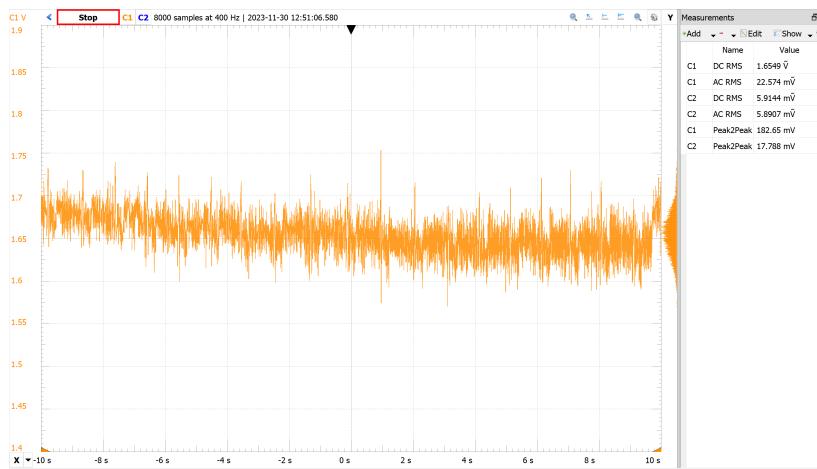


Figure 41: EKG signal measured with John Berberian

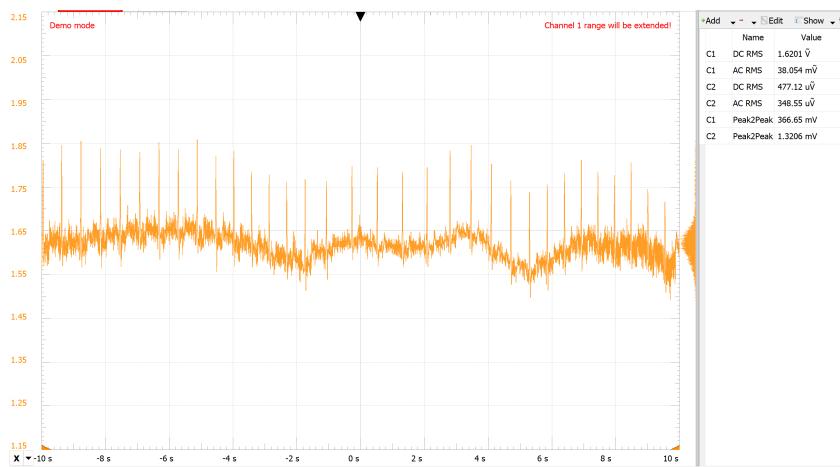


Figure 42: EKG signal measured with Paul Karhnak

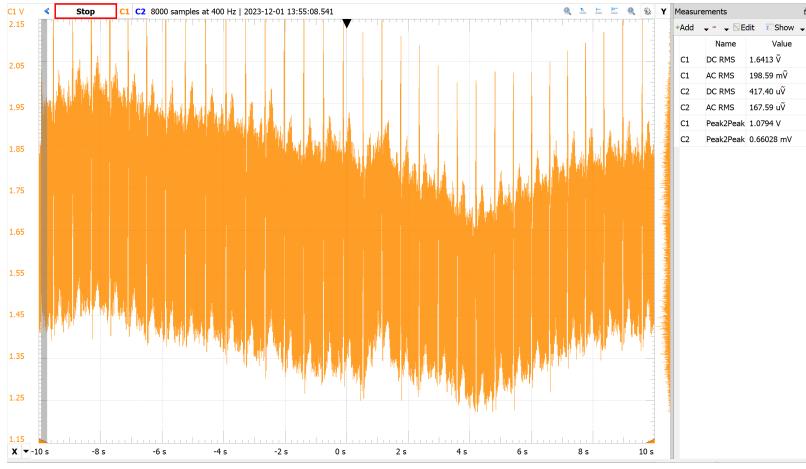


Figure 43: EKG signal measured with Liam Timmins

2.3.2 FIR Filtering

We tried to filter the data using a moving-average filter to remove 60Hz interference. Because we had a sample frequency of 400Hz, we would need a moving-average filter with 6.67 elements in it. We can't do that, so instead we used a convolution of a 6-element filter with a 7-element filter. The frequency response is shown in Fig. 44.

Our cleanest data was from Paul Karhnak, so we'll focus on that for evaluating the filtering. The unfiltered data is shown in Fig. 45. After applying the MA filter, we got a more centered signal, shown in Fig. 46.

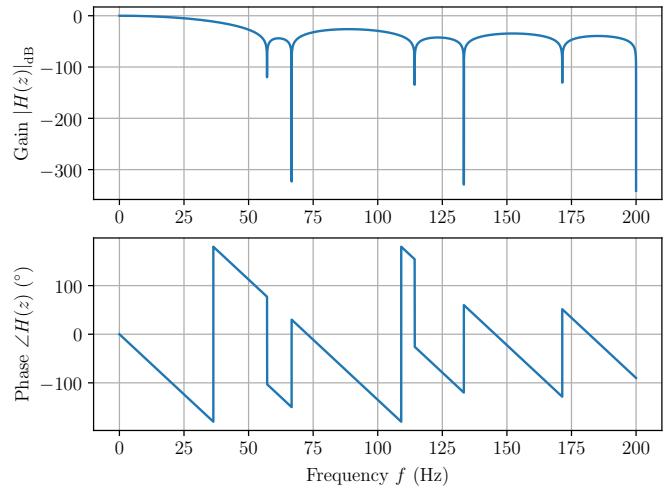


Figure 44: FIR Moving-Average Filter Frequency Response

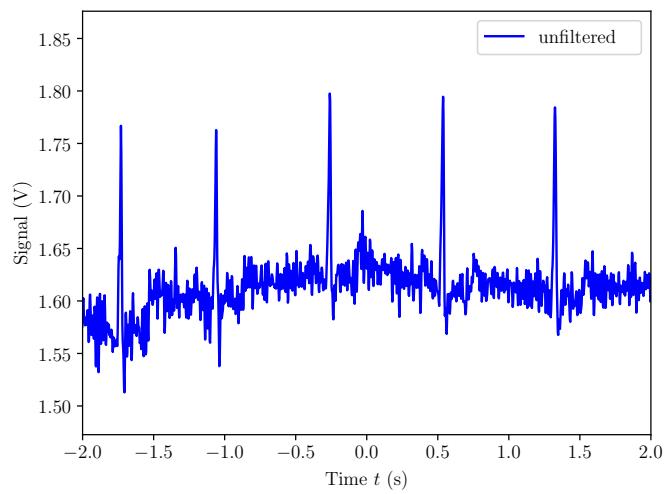


Figure 45: Paul Karhnak's Data, Unfiltered

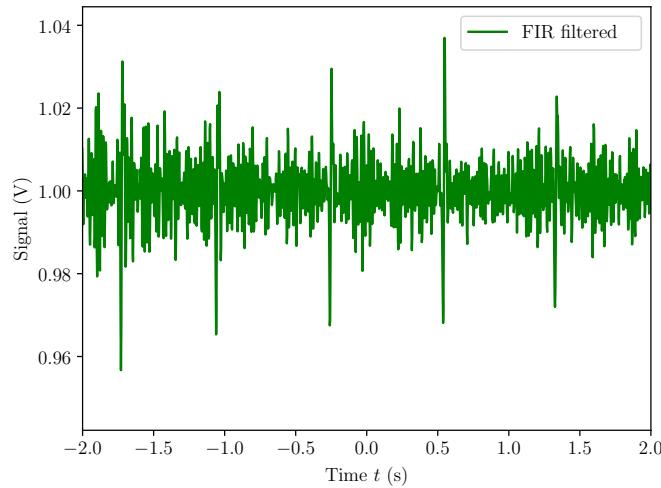


Figure 46: Paul Karhnak's Data, FIR Filtered

We noticed there was still significant noise in the FIR-filtered data, though. We suspect this is because our sample rate, 400Hz, was too low. We guessed that perhaps having nulls at exactly 60Hz and its harmonics would improve our filtering. To achieve this, we designed an IIR notch filter by placing the zero on the unit circle and placing a pole at the same Ω with $r = 0.8$. Convolving this with the MA filter from before, we got the frequency response in Fig. 47. Note the nulls at exactly 60Hz, 120Hz, and 180Hz. When we applied this filter to our data, we got much cleaner results (Fig. 48). The code for all of this is in Section 5.6 and Section 5.7.

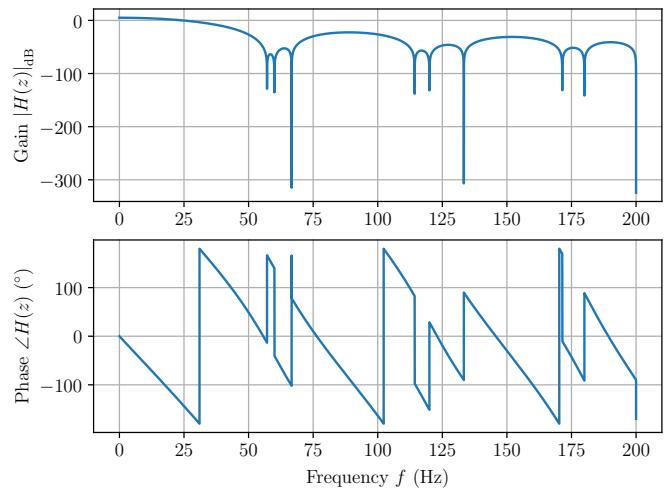


Figure 47: IIR + FIR Moving-Average Filter Frequency Response

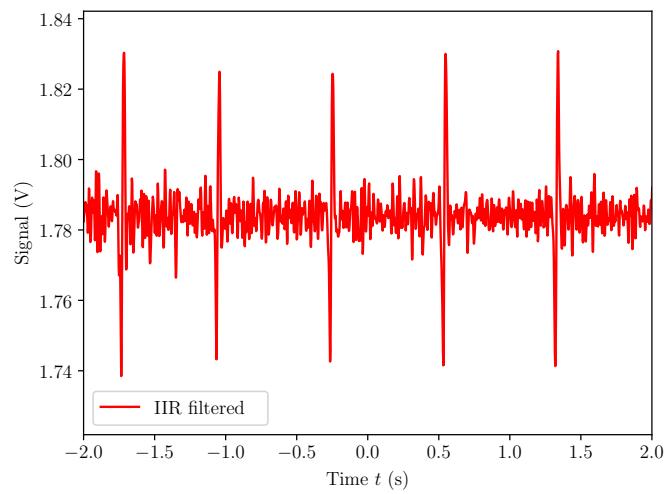


Figure 48: Paul Karhnak's Data, IIR Filtered

3 Conclusion

In comparison to previous semesters, the final project for ECE 3750 was significantly more freeform, resulting in the group needing to be far more organized when approaching the work. Despite this, the flow of work was largely concentrated in bursts dealing with the analytical calculations, designing the board, numerically simulating the designs made, and finally, testing the board experimentally.

For the design itself, the goal was to create a design which would function in the most broad cases as possible. All the design choices made in the background in Section 1.1 and layout in Section 1.4 were in pursuit of this ideal. This ideal was tested in accordance with the most extreme inputs that could be reasonably expected to occur. A notable moment during testing was for the simulation of the instrumentation amplifier, in which our initial testing conditions produced unexpected results. After further analysis of the circuit and reflection on what our expected results would be, we adjusted the testing parameters to more accurately reflect an input we would receive, resulting in the simulation in Section 1.3.2. Similar adjustments were made throughout the simulation and experimental testing procedures in order to verify functionality when more extreme cases failed.

This project made use of the NI Multisim and Ultiboard software to simulate the calculations for the schematics made and design the PCB, respectively. Additionally, FreeDFM was used to verify the functionality of the designed board before manufacturing it. Beyond the software listed above, the bibliography in Section 4 lists the resources which were referenced for designing the component values.

Generally, the biggest improvement that could have been made to this lab is the organization of presentation to the students, both internally, with clearer organization of data and information to refer back to our results, and externally, with clearer outlines and goals for individual steps of the project. There were several times where certain information was unclear, leading to instances where not all the data intended was able to be properly collected, such as with the final digital signal processing section, resulting in unanticipated adjustments. If the submission requirements were more clearly posted at an earlier point in time, this would have likely been avoided.

4 Bibliography

References

- [1] *Micropower low dropout regulators with shutdown*, LT1121, Linear Technology, 1994. [Online]. Available: <https://www.analog.com/media/en/technical-documentation/data-sheets/1121fg.pdf>.

- [2] *Single and dual-supply, rail-to-rail, low cost instrumentation amplifier*, AD623, Analog Devices, 2020. [Online]. Available: <https://www.analog.com/media/en/technical-documentation/data-sheets/ad623.pdf>.
- [3] J. Karki, “Active low-pass filter design,” Texas Instruments, Tech. Rep., Sep. 2002.
- [4] J. Karki, “Analysis of the salen-key architecture: Application report,” Texas Instruments, Tech. Rep., Sep. 2002.
- [5] *Linear optocoupler, high gain stability, wide bandwidth*, IL300, Vishay Semiconductors, 2018. [Online]. Available: <https://www.vishay.com/docs/83622/il300.pdf>.
- [6] *Cmos dual rail-to-rail input and output operational amplifier*, LMC6482, Texas Instruments, 1997. [Online]. Available: <https://www.ti.com/lit/ds/symlink/lmc6482.pdf>.

5 Code

5.1 Input Filter Brute-Force

```

1  #!/usr/bin/env python3
2  import numpy as np
3
4  # Requirement: RC Rin such that 250k <= Rin <=300k
5  validresistors = 270e3
6  validcaps = np.array([
7      10e-12, 22e-12, 27e-12, 33e-12, 47e-12, 100e-12, 220e-12, 270e-12, 330e-12,
8      470e-12, 680e-12, 1000e-12, 1500e-12, 2200e-12, 3300e-12, 4700e-12,
9      0.01e-6, 0.015e-6, 0.022e-6, 0.033e-6, 0.047e-6, 0.1e-6, 0.15e-6, 0.22e-6,
10     0.33e-6, 0.47e-6, 0.68e-6, 1e-6, 2.2e-6, 4.7e-6, 10e-6
11 ])
12
13 corners = np.reciprocal(np.multiply(2 * np.pi * validresistors, validcaps))
14
15 solutions = validcaps[np.all([corners >= 0.05, corners <= 0.1], axis=0)]
16
17 print(validresistors)
18 print(solutions)
19
20 # Selected C=1e-5
21 # Translated C4, C6=10e-6, R2, R3 = 270e3

```

5.2 Input Signal Range Calculations

```
1 #!/usr/bin/env python3
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 plt.rcParams.update({
6     "text.usetex": True,
7     "font.family": "serif",
8     "font.size": 12
9 })
10
11 data = np.loadtxt("ECG-OriginalVals.lvm-modded")
12
13 time = data[:, 0]
14 voltage = data[:, 1]
15
16 print("max: "+str(np.max(voltage)))
17 print("min: "+str(np.min(voltage)))
18 print("range: "+str(np.max(voltage)-np.min(voltage)))
19
20 plt.plot(time, 1000*voltage)
21 plt.xlabel(r"Time $t$ (s)")
22 plt.ylabel(r"Skin Voltage $V$ (mV)")
23 plt.show()
```

5.3 Integrator Value Selection

```
1 import numpy as np
2
3 decade = np.array([10, 12, 15, 18, 22, 27, 33, 39, 47, 56, 68, 82])
4
5 res = np.concatenate((decade, decade * 10, decade * 100, decade * 1000,
6                      decade * 10000, [1e6, 2.2e6]))
7 cap = np.array([
8     10e-12, 22e-12, 27e-12, 33e-12, 47e-12, 100e-12, 220e-12, 270e-12, 330e-12,
9     470e-12, 680e-12, 1000e-12, 1500e-12, 2200e-12, 3300e-12, 4700e-12,
10    0.01e-6, 0.015e-6, 0.022e-6, 0.033e-6, 0.047e-6, 0.1e-6, 0.15e-6, 0.22e-6,
11    0.33e-6, 0.47e-6, 0.68e-6, 1e-6, 2.2e-6, 4.7e-6, 10e-6
12 ])
13
14 # Requirement:  $-1/(sRC)$  in range  $-1/s$  to  $-1/(10s)$ 
15 prodmax = 10
16 prodmin = 1
```

```

17
18 shape = (len(res), len(cap))
19 resimat = np.broadcast_to(res.reshape((shape[0], 1)), shape)
20 capimat = np.broadcast_to(cap.reshape((1, shape[1])), shape)
21
22 prodmat = np.multiply(resimat, capimat)
23 good = np.all([prodmat <= prodmax, prodmat >= prodmin], axis=0).nonzero()
24
25 goodvals = list(
26     map(np.ndarray.tolist, [resimat[good], capimat[good], prodmat[good]]))
27 print("Good values:")
28 for r, c, p in zip(*goodvals):
29     print()
30     print("product={}, R4={}, C3={}".format(p, r, c))
31
32 # Selected R4=1000000.0, C3=4.7e-06
33 # Translated R4=1e6, C3=4.7e-6

```

5.4 V_{mid} Properties Brute-Force

```

1 import numpy as np
2
3 decade = np.array([10, 12, 15, 18, 22, 27, 33, 39, 47, 56, 68, 82])
4
5 res = np.concatenate((decade, decade * 10, decade * 100, decade * 1000,
6                      decade * 10000, [1e6, 2.2e6]))
7 cap = np.array([
8     10e-12, 22e-12, 27e-12, 33e-12, 47e-12, 100e-12, 220e-12, 270e-12, 330e-12,
9     470e-12, 680e-12, 1000e-12, 1500e-12, 2200e-12, 3300e-12, 4700e-12,
10    0.01e-6, 0.015e-6, 0.022e-6, 0.033e-6, 0.047e-6, 0.1e-6, 0.15e-6, 0.22e-6,
11    0.33e-6, 0.47e-6, 0.68e-6, 1e-6, 2.2e-6, 4.7e-6, 10e-6
12 ])
13
14 # Requirement: thevenin resistance seen by + input on op-amp btw. 25k-100k
15 resmax = 100e3
16 resmin = 25e3
17
18 # Requirement: breakpoint frequency between 2 and 5 hz.
19 fcmax = 5
20 fcmin = 2
21
22 # Requirement: Vmid output must be within 5% of Vcc/2 -> voltage divider within 5% of 1/2.
23 vdivmax = 0.5 * 1.05
24 vdivmin = 0.5 * 0.95
25

```

```

26 shape = (len(res), len(res), len(cap))
27 res1mat = np.broadcast_to(res.reshape((shape[0], 1, 1)), shape)
28 res2mat = np.broadcast_to(res.reshape((1, shape[1], 1)), shape)
29 capimat = np.broadcast_to(cap.reshape((1, 1, shape[2])), shape)
30
31 rplus = np.reciprocal(np.add(np.reciprocal(res1mat), np.reciprocal(res2mat)))
32 fc = np.reciprocal(np.multiply(2 * np.pi, np.multiply(capimat, rplus)))
33 vdiv = np.divide(res2mat, np.add(res1mat, res2mat))
34
35 resgood = np.all([rplus < resmax, rplus > resmin], axis=0)
36 fcgood = np.all([fc < fcmax, fc > fcmin], axis=0)
37 vdivgood = np.all([vdiv < vdivmax, vdiv > vdivmin], axis=0)
38
39 good = np.all([resgood, fcgood, vdivgood], axis=0).nonzero()
40
41 goodvals = list(
42     map(np.ndarray.tolist, [
43         res1mat[good], res2mat[good], capimat[good], rplus[good], fc[good],
44         vdiv[good]
45     ]))
46 print("Good values:")
47 for r6, r7, c8, rp, f, vd in zip(*goodvals):
48     print()
49     print("Rplus={}, fc={}, vdiv={}".format(rp, f, vd))
50     print("R6={}, R7={}, C8={}".format(r6, r7, c8))
51
52 # Selected R6=150000.0, R7=150000.0, C8=1e-06
53 # Translated R6=150e3, R7=150e3, C8=1e-6

```

5.5 Filter Value Brute-Force

```

1 import numpy as np
2
3 decade = np.array([10, 12, 15, 18, 22, 27, 33, 39, 47, 56, 68, 82])
4
5 # Requirement: All resistors >= 10k
6 res = np.concatenate((decade * 1000, decade * 10000, [1e+6, 2.2e+6]))
7 cap = np.array([
8     10e-12, 22e-12, 27e-12, 33e-12, 47e-12, 100e-12, 220e-12, 270e-12, 330e-12,
9     470e-12, 680e-12, 1000e-12, 1500e-12, 2200e-12, 3300e-12, 4700e-12,
10    0.01e-6, 0.015e-6, 0.022e-6, 0.033e-6, 0.047e-6, 0.1e-6, 0.15e-6, 0.22e-6,
11    0.33e-6, 0.47e-6, 0.68e-6, 1e-6, 2.2e-6, 4.7e-6, 10e-6
12 ])
13
14 # Corner frequency calculations.

```

```

15 targetfc = 500 * np.power(10**7.2 - 1, -1 / 8)
16 fctol = 0.1 * targetfc
17
18 shape = (len(res), len(res), len(cap), len(cap))
19 res1mat = np.broadcast_to(res.reshape((shape[0], 1, 1, 1)), shape)
20 res2mat = np.broadcast_to(res.reshape((1, shape[1], 1, 1)), shape)
21 cap1mat = np.broadcast_to(cap.reshape((1, 1, shape[2], 1)), shape)
22 cap2mat = np.broadcast_to(cap.reshape((1, 1, 1, shape[3])), shape)
23 nmat = np.divide(cap2mat, cap1mat)
24 mmat = np.divide(res1mat, res2mat)
25
26 fcmat = np.reciprocal(
27     np.multiply(
28         2 * np.pi,
29         np.multiply(np.multiply(res2mat, cap1mat),
30                     np.sqrt(np.multiply(nmat, mmat))))))
31
32 fcerr = np.abs(np.subtract(fcmat, targetfc))
33 # We need to be within tolerance and lower than the max corner frequency.
34 fcgood = np.where(np.logical_and(fcmat < targetfc, fcerr < fctol), True, False)
35
36 # Q calculations
37 # target Q 1
38 tQ1 = 0.5412
39 tQ2 = 1.3065
40 qtol = 0.008
41
42 Qmat = np.divide(np.sqrt(np.multiply(nmat, mmat)), np.add(mmat, 1))
43 Qe1 = np.abs(np.subtract(Qmat, tQ1))
44 Qe2 = np.abs(np.subtract(Qmat, tQ2))
45
46 Qgood1 = np.where(Qe1 < qtol, True, False)
47 Qgood2 = np.where(Qe2 < qtol, True, False)
48
49 good1 = np.logical_and(Qgood1, fcgood).nonzero()
50 good2 = np.logical_and(Qgood2, fcgood).nonzero()
51
52 goodvals1 = list(
53     map(np.ndarray.tolist, [
54         res1mat[good1], res2mat[good1], cap1mat[good1], cap2mat[good1],
55         fcmat[good1], fcerr[good1], Qmat[good1], Qe1[good1]
56     ]))
57 goodvals2 = list(
58     map(np.ndarray.tolist, [
59         res1mat[good2], res2mat[good2], cap1mat[good2], cap2mat[good2],
60         fcmat[good2], fcerr[good2], Qmat[good2], Qe2[good2]
61     ]))

```

```

62
63     print("Good values for filter 1:")
64     for r1, r2, c1, c2, f, ferr, q, qerr in zip(*goodvals1):
65         print()
66         print("qerr={}, ferr={}, q={}, fc={}".format(qerr, ferr, q, f))
67         print("r1={}, r2={}, c1={}, c2={}".format(r1, r2, c1, c2))

68
69     print("\n")
70     print("Good values for filter 2:")
71     for r1, r2, c1, c2, f, ferr, q, qerr in zip(*goodvals2):
72         print()
73         print("qerr={}, ferr={}, q={}, fc={}".format(qerr, ferr, q, f))
74         print("r1={}, r2={}, c1={}, c2={}".format(r1, r2, c1, c2))

75
76     # Selected S1: r1=1000000.0, r2=47000.0, c1=4.7e-09, c2=3.3e-08
77     # Translated R8=1e6, R9=47e3, C10=4700e-12, C11=0.033e-06
78
79     # Selected S2: r1=470000.0, r2=470000.0, c1=2.2e-09, c2=1.5e-08
80     # Translated R10=470e3, R11=470e3, C12=2200e-12, C13=0.015e-06

```

5.6 DSP Filter Calculations

```

1  #!/usr/bin/env python3
2  import numpy as np
3  from numpy.polynomial import Polynomial
4  import matplotlib.pyplot as plt
5
6  plt.rcParams.update({
7      "text.usetex": True,
8      "font.family": "serif",
9      "font.size": 12
10 })
11
12
13 def H(z, b, a):
14     Z = np.reshape(z, (-1, 1))
15     num = np.sum(np.multiply(np.power(Z, -np.arange(len(b))), b), axis=1)
16     den = np.sum(np.multiply(np.power(Z, -np.arange(len(a))), a), axis=1)
17     return np.divide(num, den)
18
19
20 fs = 400
21 fkill = 60
22
23 Omega = np.linspace(0, np.pi, 256000)

```

```

24 f = np.multiply(fs / (2 * np.pi), Omega)
25 z = np.exp(np.multiply(1j, Omega))
26
27 r = 1
28 zeros = np.array([
29     r * np.exp(2j * np.pi * fkill / fs), r * np.exp(-2j * np.pi * fkill / fs),
30     r * np.exp(4j * np.pi * fkill / fs), r * np.exp(-4j * np.pi * fkill / fs),
31     r * np.exp(6j * np.pi * fkill / fs), r * np.exp(-6j * np.pi * fkill / fs)
32 ])
33 r = 0.8
34 poles = np.array([
35     r * np.exp(2j * np.pi * fkill / fs), r * np.exp(-2j * np.pi * fkill / fs),
36     r * np.exp(4j * np.pi * fkill / fs), r * np.exp(-4j * np.pi * fkill / fs),
37     r * np.exp(6j * np.pi * fkill / fs), r * np.exp(-6j * np.pi * fkill / fs)
38 ])
39
40 b = np.real(Polynomial.fromroots(zeros).coef[::-1])
41 a = np.real(Polynomial.fromroots(poles).coef[::-1])
42
43 b = np.convolve(b, np.convolve(np.ones(7) / 7, np.ones(6) / 6))
#b=np.convolve(np.ones(7)/7, np.ones(6)/6)
#a=np.array([1])
44
45
46 print("b = np.array(" + str(list(b)) + ")")
47 print("a = np.array(" + str(list(a)) + ")")
48 transfer = H(z, b, a)
49
50 maxstopgain = 20 * np.log10(
51     max(
52         np.abs(transfer[np.logical_and(
53             np.angle(z) > np.pi / 4,
54             np.angle(z) < np.pi / 3)])))
55 print(maxstopgain)
56
57 plt.figure()
58 plt.subplot(2, 1, 1)
59 plt.plot(f, 20 * np.log10(np.abs(transfer)))
60 plt.grid(which="both")
61 plt.ylabel(r"Gain $\left|H(z)\right|$ dB")
#plt.ylim([maxstopgain - 20, plt.ylim()[1]])
62
63 plt.subplot(2, 1, 2)
64 plt.plot(f, np.angle(transfer, deg=True))
65 plt.grid(which="both")
66 plt.ylabel(r"Phase $\angle H(z)$ rad")
67 plt.xlabel(r"Frequency $f$ Hz")
68
69
70

```

```

71 #plt.show()
72 plt.savefig("problem2_transfer_iir.pdf")
73 plt.savefig("problem2_transfer_iir.png")

```

5.7 DSP Filtering Code

```

1 #!/usr/bin/env python3
2 import sys
3
4 import numpy as np
5 import matplotlib.pyplot as plt
6
7 plt.rcParams.update({
8     "text.usetex": True,
9     "font.family": "serif",
10    "font.size": 12
11})
12
13 data = np.loadtxt(sys.argv[1], skiprows=1, delimiter=',')
14
15 biir = np.array([
16     0.023809523809523808, 0.07963277420845843, 0.1643360245854834,
17     0.28016277549251806, 0.4248695263776431, 0.6015900038521789,
18     0.7783104813267145, 0.8992077084023158, 0.9592112089104156,
19     0.9592112089104157, 0.8992077084023158, 0.7783104813267147,
20     0.6015900038521788, 0.42486952637764297, 0.280162775492518,
21     0.1643360245854834, 0.07963277420845843, 0.0238095238095238
22 ])
23 aiir = np.array([
24     1.0, 1.0756612134042038, 0.7762943994110685, 0.6692797553973278,
25     0.4968284156230841, 0.44059083301036195, 0.262144000000000004
26 ])
27 bfir = np.array([
28     0.023809523809523808, 0.047619047619047616, 0.07142857142857142,
29     0.09523809523809523, 0.11904761904761904, 0.14285714285714285,
30     0.14285714285714285, 0.11904761904761904, 0.09523809523809523,
31     0.07142857142857142, 0.047619047619047616, 0.023809523809523808
32 ])
33 afir = np.array([1])
34 #b = np.array([1.0, -1.1755705045849465, 1.0])
35 #a = np.array([1.0, -0.9404564036679572, 0.6400000000000001])
36
37
38 def apply_filter(signal, b, a):
39     return np.divide(np.convolve(b, signal, mode='same'), a)

```

```

40     np.convolve(a, signal, mode='same'))
41
42
43     time = data[:, 0]
44     voltage = data[:, 1]
45
46     iidata = apply_filter(voltage, biir, aiir)
47     cutiir = max(len(biir), len(aiir)) // 2
48     tiir = time[cutiir:-cutiir]
49     viir = iidata[cutiir:-cutiir]
50     np.savetxt(sys.argv[2] + '_IIR.csv', (tiir, viir), delimiter=',')
51
52     firdata = apply_filter(voltage, bfir, afir)
53     cutfir = max(len(bfir), len(afir)) // 2
54     tfir = time[cutfir:-cutfir]
55     vfir = firdata[cutfir:-cutfir]
56     np.savetxt(sys.argv[2] + '_FIR.csv', (tfir, vfir), delimiter=',')
57
58     plt.figure()
59     plt.plot(time, voltage, 'b', label="unfiltered")
60     plt.xlabel('Time $t$ (s)')
61     plt.ylabel('Signal (V)')
62     plt.xlim([-2, 2])
63     plt.legend()
64     plt.savefig(sys.argv[3] + '.png')
65     plt.savefig(sys.argv[3] + '.pgf')
66
67     plt.figure()
68     plt.plot(tiir, viir, 'r', label="IIR filtered")
69     plt.xlabel('Time $t$ (s)')
70     plt.ylabel('Signal (V)')
71     plt.xlim([-2, 2])
72     plt.legend()
73     plt.savefig(sys.argv[3] + '_IIR.png')
74     plt.savefig(sys.argv[3] + '_IIR.pgf')
75
76     plt.figure()
77     plt.plot(tfir, vfir, 'g', label="FIR filtered")
78     plt.xlabel('Time $t$ (s)')
79     plt.ylabel('Signal (V)')
80     plt.xlim([-2, 2])
81     plt.legend()
82     plt.savefig(sys.argv[3] + '_FIR.png')
83     plt.savefig(sys.argv[3] + '_FIR.pgf')
84
85     plt.show()
86

```

```
87 #fft=np.fft.rfft(voltage)
88 #filtfft = np.fft.rfft(filtered_data)
89
90 #plt.plot(fft, label="unfiltered")
91 #plt.plot(filtfft, label="filtered")
92 #plt.legend()
93 #plt.show()
```
