Johnson Le

January 23, 2019

CMPE 167/L

Lab 1 Report

Introduction

This lab introduces two new sensors: flex sensor and vibration sensor. The flex sensor works like a potentiometer by changing resistance based on bending angle. The vibration sensor detects vibrations and outputs higher voltages the stronger the vibration. The goal of this lab is to incorporate these sensors into one circuit to control the speaker used from last lab. For this lab, I collaborated with Talin and helped James with some concepts.

Part 1 – Linearize the Flex Sensor

Method:

The goal here is to linearize the flex sensor to get consistent readings based on the degree of flex. The first step I took was to get readings from the flex sensor onto an A/D pin.

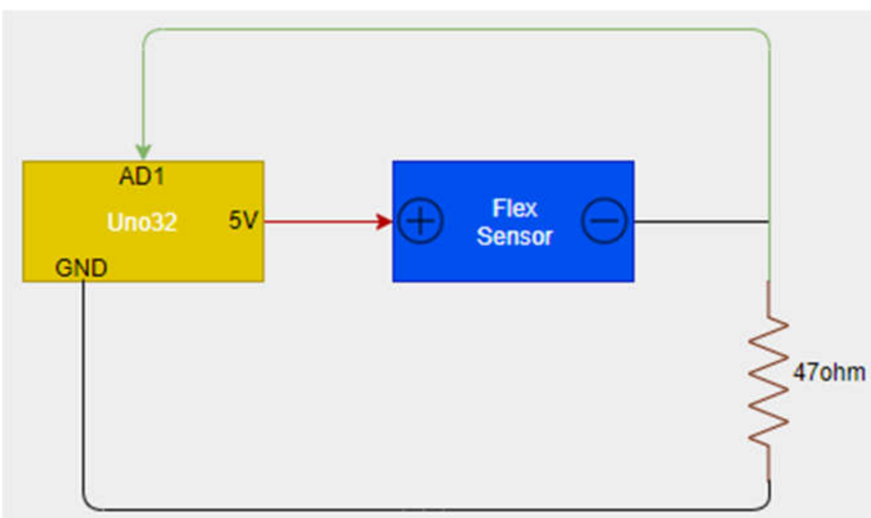


Fig 1 Flex Sensor Wiring

I notice the values ranged to a maximum of 1023. Printing the values onto the OLED, I could easily note down A/D values onto excel. I printed out a protractor to measure the degree of flex and noted degree of flex vs A/D reading. The results are shown in figure 2.

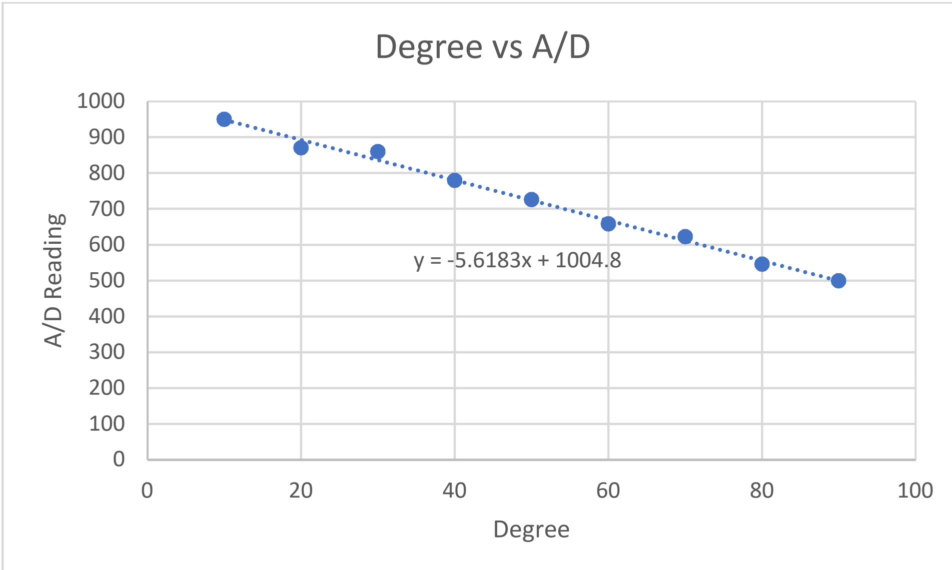


Fig 2 Flex Sensor: Degree Vs A/D Reading

Utilizing excel, I added a best fit line from all the data and got an equation. The equation can be seen in figure 2. The y denotes A/D reading and x denotes the degree of flex. Because the uno32 can't read the degree of flex as an input, I solved for x. This gives me an equation I can code up as a function allowing the A/D readings from the flex sensor to be converted to degrees providing me with linearization of the flex sensor.

Part2 – Capture the Taps

Method:

Moving from the flex sensor, part 2 has us explore the vibration sensor. The vibration sensor generates voltage based on the amount of vibrations it goes through. Before sending any signal from the vibration sensor into the uno32, I had to make sure that the signal generated was within working range for the uno32. I hooked up the vibration sensor in parallel with a resistor and diode. I then hooked up an oscilloscope for the output which is the positive terminal. Striking the sensor, I get peak to peak values ranging from 1.4V to 2.8V averaging 2V depending on how much force I put into striking. This is a safe voltage range to input into the uno32. I tested this by continually striking the vibration sensor with some trigger delay from the oscilloscope.

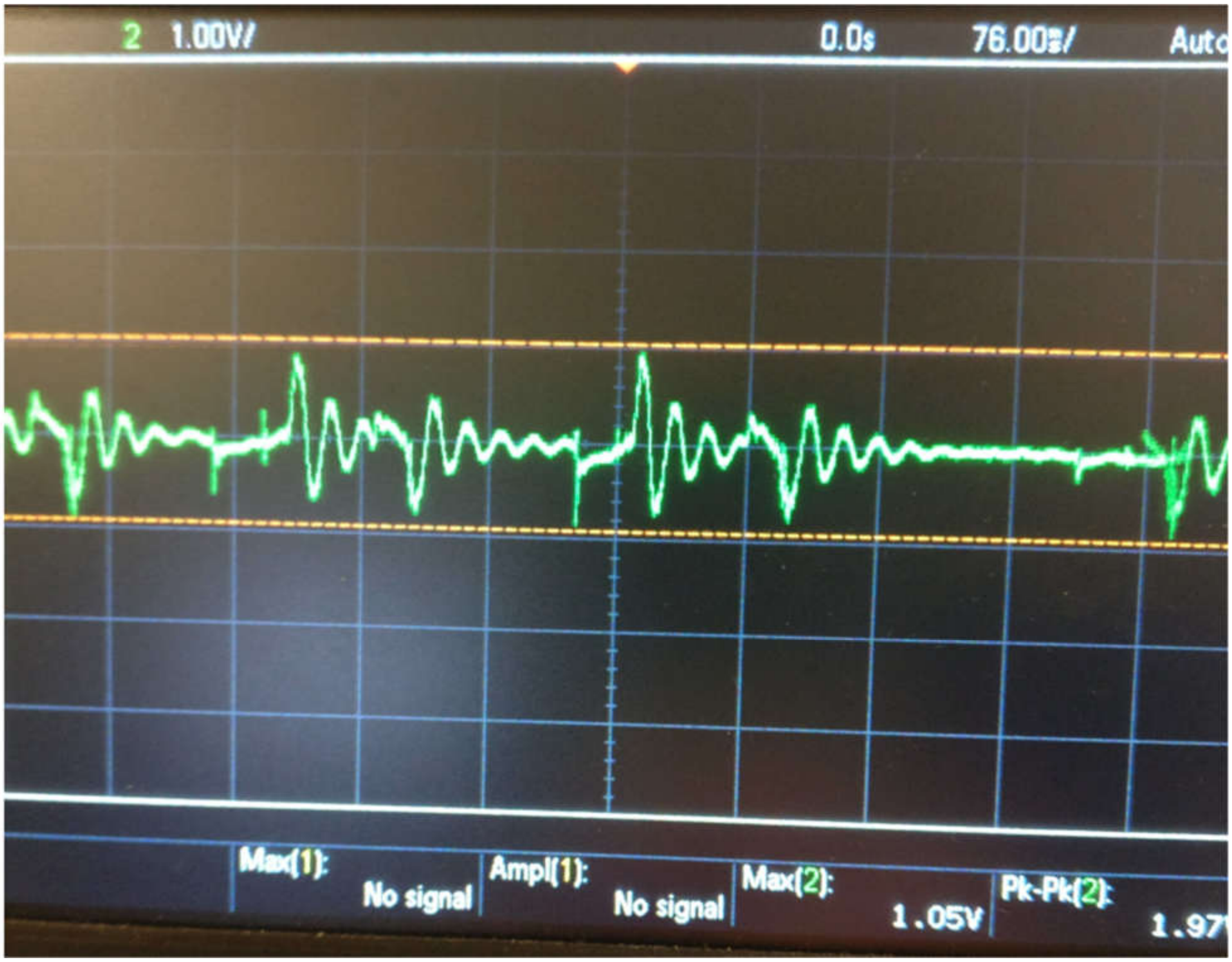


Fig 3Average Striking Voltage

Now I connected the vibration sensor to the uno32 for some coding. The wiring is shown in Figure 4. I outputted the AD value of the vibration sensor onto the OLED to see what values would be appropriate for considering it high. Once that was determined, I made a while loop with a condition that an LED would light up briefly if the vibration sensor was hit.

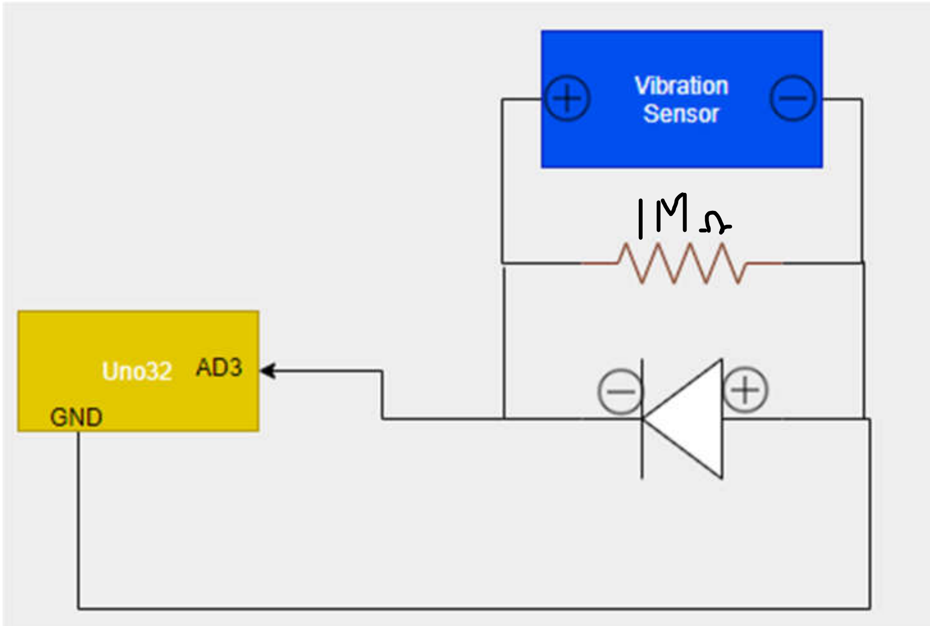


Fig 4Wiring of Vibration Sensor

Part3 – Tone Out Speaker Based On Flex

Method:

Now that I have linearize the flex sensors into degrees from part 1, I can set the tone of the speaker based on how much degree of flex. I decided to only have the sensor flex from 10 to 100 degrees. Anything above 100 would have the same tone as the one at 100 degrees. I then mapped that range to the range I wanted the speaker to have which is 100-880. This range was decided in the previous lab. 100-880 gives nice, clear sounds from the speaker. I incorporated the wiring from Lab0 into my circuit as seen in Figure 5.

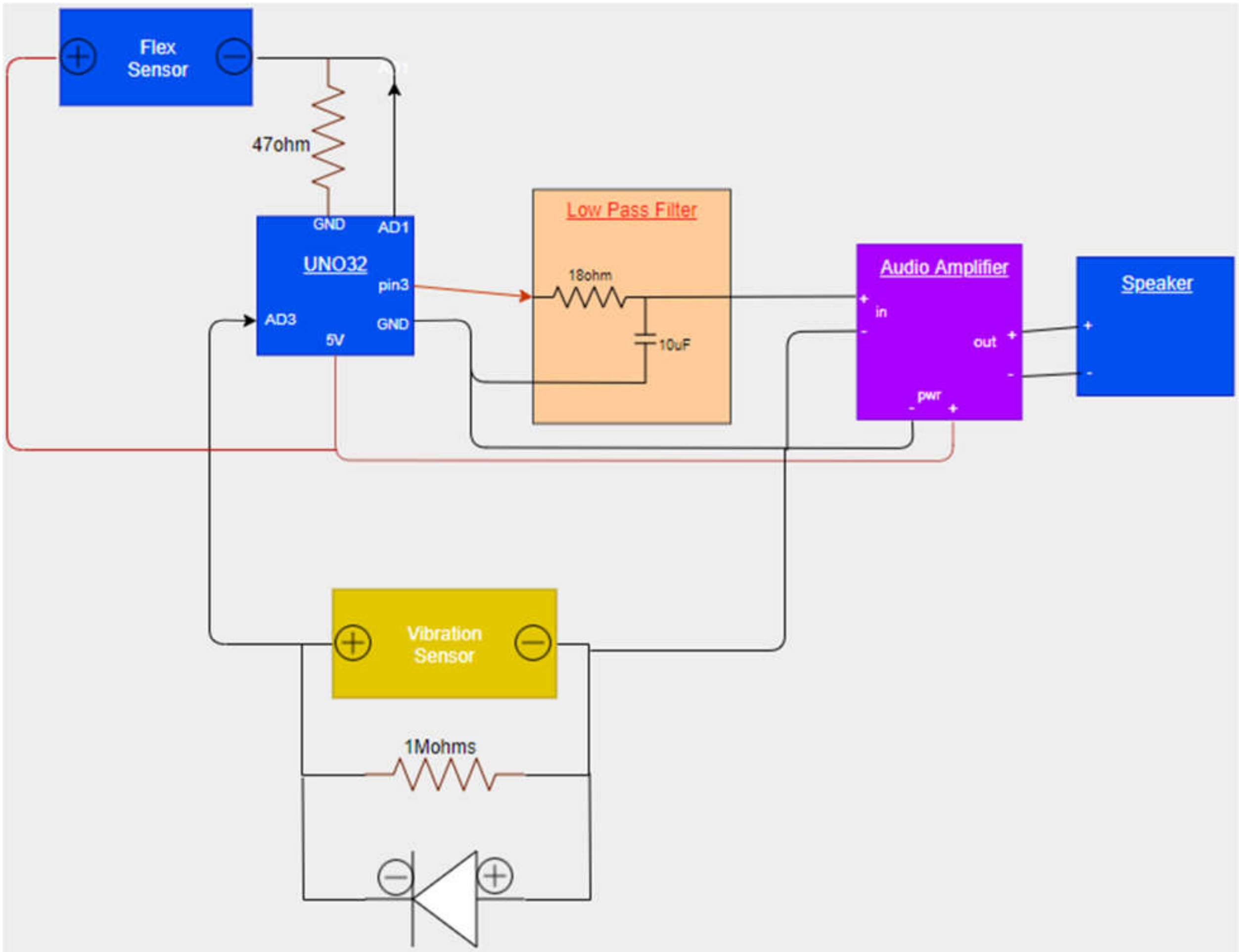


Fig 5Wiring for Everything

Once mapping the range, I used the tone generating library to set the tone based on the degree of flex.

Questions:

Is it noisy? Is it smooth?:

The flex sensor readings are noisy and bounces around rapidly without any filtering. The sound is also not smooth and has the properties of “crackly.”

Is your linearization of the flex sensor correct?:

The flex sensor provides the uno32 with AD values. I take those AD values and throw it into a linear function to get an arbitrary degree of flex. Because I throw it into this function, the flex sensor is linearized.

How could you improve the sound?:

I improved the sound by adding a simple, passive low-pass filter at 880hz. Although this decreases the volume, it reduces some noise allowing for smoother sound transitions. I also only allowed tones between 100 and 880 because tones outside of that range gets funky.

Part4 – Tone Out Based on Tap

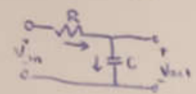
Method:

Now that there is code for interfacing the flex and vibration sensor, we can combine the two. In code, I start the speaker with no sound and constantly check for a vibration detection. If a vibration is detected, I will read the desired tone, set and turn on the speaker for a moment. The duration of the tone is reset if the vibration sensor is hit before the tone goes off. There is no change in hardware and part 4 is all software.

Part5 – Simple Analog Filtering

Method:

Low-pass filter



$Z_R = R$

$Z_c = \frac{1}{sC}$

$s = j\omega$

$$\frac{V_{in} - V_{out}}{Z_R} = \frac{V_{out} - 0}{Z_c}$$

$$\frac{V_{in}}{Z_R} = V_{out}\left(\frac{1}{Z_R} + \frac{1}{Z_c}\right)$$

$$\frac{V_{out}}{V_{in}} = \frac{1}{Z_R\left(\frac{1}{Z_R} + \frac{1}{Z_c}\right)}$$

$$= \frac{1}{1 + \frac{Z_R}{Z_c}}$$

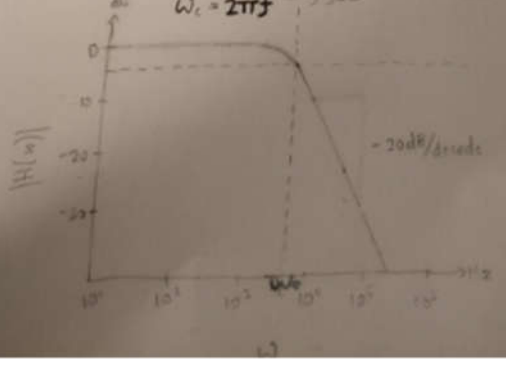
$$= \frac{1}{1 + \frac{R}{1/sC}}$$

$$\frac{V_{out}}{V_{in}} = \frac{1}{1 + sRC}$$

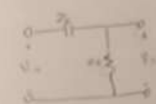
$f_c = \frac{1}{2\pi RC}$, choosing $R =$ [redacted] 18Ω
$C =$ [redacted] $10uF$

$f_c = 880 Hz$, $H(\omega) = 20\log\left(\frac{1}{\sqrt{1 + (\omega/\omega_c)^2}}\right)$

$\omega_c = 2\pi f$ = 5555



High-pass filter



$$\frac{V_{in} - V_{out}}{Z_c} = \frac{V_{out} - 0}{Z_R}$$

$$\frac{V_{in}}{Z_c} = V_{out}\left(\frac{1}{Z_c} + \frac{1}{Z_R}\right)$$

$$\frac{V_{out}}{V_{in}} = \frac{1}{Z_c\left(\frac{1}{Z_c} + \frac{1}{Z_R}\right)}$$

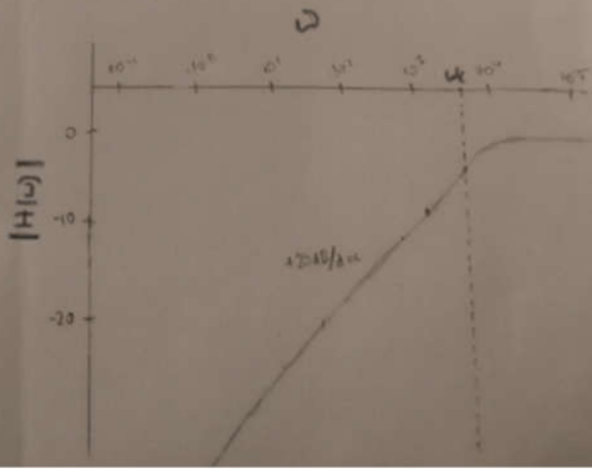
$$= \frac{1}{1 + \frac{Z_c}{Z_R}} = \frac{1}{1 + \frac{1/sC}{R}}$$

$$= \frac{1}{1 + \frac{1}{sCR}}$$

$$= \frac{sCR}{1 + sCR}$$

$$H(\omega) = 20\log\left|\frac{\omega RC}{1 + \omega RC}\right|$$

$$= 20\log\left(\frac{\omega/\omega_c}{1 + (\omega/\omega_c)^2}\right)$$



Band-pass : multiplying the two together

$$L.P: \frac{V_{out}}{V_{in}} = \frac{1}{1+sRC} = V_{out} = \frac{V_{in}}{1+sRC}$$

$$H.P: \frac{V_{out}}{V_{in}} = \frac{sRC}{1+sRC} = V_{out} = \frac{sRC}{1+sRC} \cdot V_{in}$$

1 $$B.P: V_{out} = \frac{sRC}{1+sRC} \cdot \frac{V_{in}}{1+sRC}$$

$$\frac{V_{out}}{V_{in}} = \frac{sRC}{(1+sRC)^2} = \frac{sRC}{1+2sRC+s^2R^2C^2}$$

$$\frac{V_{out}}{V_{in}} = \frac{RCs}{s^2R^2C^2+2sRC+1}$$

$$= \frac{\left[\frac{1}{RC}\right]s}{s^2+2\left[\frac{1}{RC}\right]+\frac{1}{R^2C^2}}$$

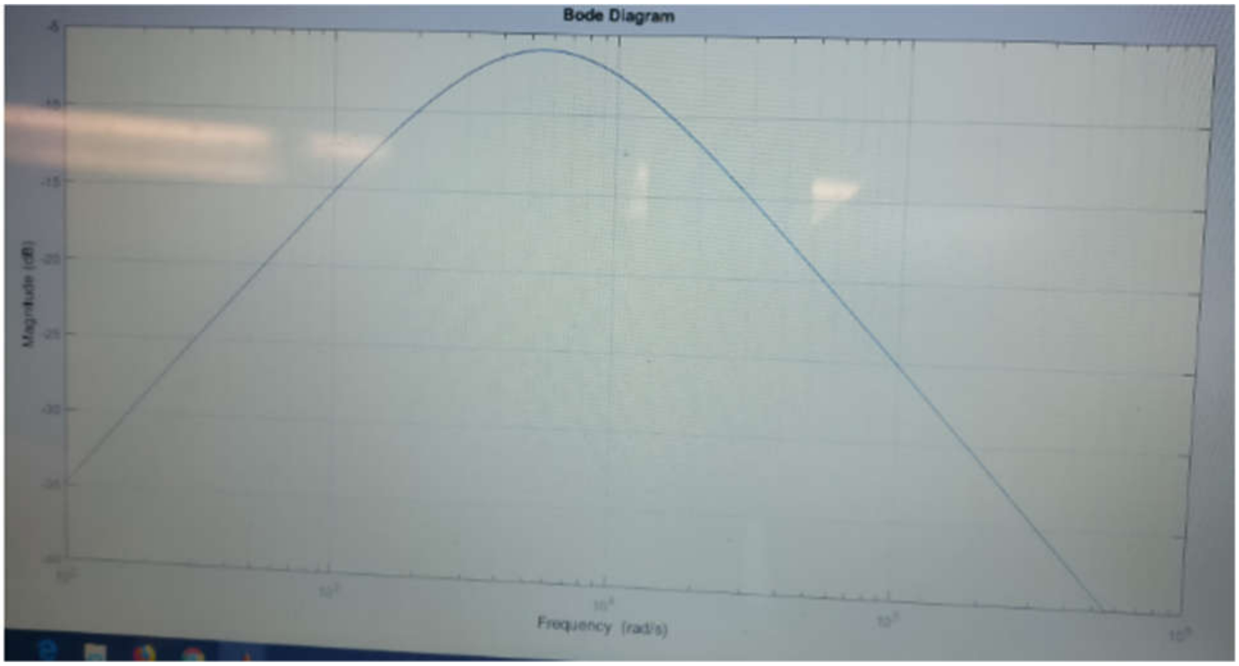
$$\omega_n^2 = \frac{1}{R^2C^2} \Rightarrow \omega_n = \frac{1}{RC}$$

$$\frac{\omega_n}{Q} = \frac{1}{RC} \qquad Q = RC\omega_n = \frac{RC}{RC} = 1$$

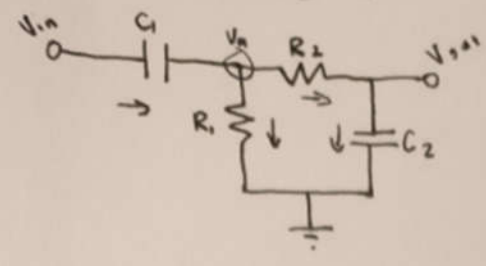
$R = 18\,\Omega$, $C = 10\,\mu F$

$\omega_n = 5555.55$

$Q = 1$


Bode Diagram
Magnitude (dB)
Frequency (rad/s)

Bandpass : using KCL



① $\dfrac{V_{in} - V_a}{1/sC_1} \Rightarrow sC_1 V_{in} - sC_1 V_a = \dfrac{V_a}{R_1} + \dfrac{V_a}{R_2} - \dfrac{V_{out}}{R_2}$

② $\dfrac{V_a}{R_2} - \dfrac{V_{out}}{R_2} = s C_2 V_{out}$

from ② $\Rightarrow \quad V_a = V_{out}\left[1 + sR_2C_2\right]$

$$sC_1 V_{in} = V_a\left[sC_1 + \frac{1}{R_1} + \frac{1}{R_2}\right] - \frac{V_{out}}{R_2}$$

$$= V_{out}\left[(1 + sR_2C_2)\left(sC_1 + \frac{1}{R_1} + \frac{1}{R_2}\right) - \frac{1}{R_2}\right]$$

$$= V_{out}\left[sC_1 + \frac{1}{R_1} + s^2R_2C_1C_2 + \frac{sR_2C_2}{R_1} + sC_2\right]$$

$$\frac{V_{out}}{V_{in}} = \frac{sC_1R_1}{sC_1R_1 + 1 + s^2R_1R_2C_1C_2 + sR_2C_2 + sR_1C_2}$$

if $C_1 = C_2$ & $R_1 = R_2$:

$$\frac{V_{out}}{V_{in}} = \frac{sCR}{s^2R^2C^2 + 3sRC + 1}$$

$$\frac{V_{out}}{V_{in}} = \frac{[R_1C_1]s}{[R_1R_2C_1C_2]s^2 + [R_1C_1 + R_2C_2 + R_1C_2]s + 1}$$

$$= \frac{\left[\frac{1}{R_2C_2}\right]s}{s^2 + \left[\frac{1}{R_2C_2} + \frac{1}{R_1C_1} + \frac{1}{R_2C_1}\right]s + \frac{1}{R_1R_2C_1C_2}}$$

$$W_n^2 = \frac{1}{R_1R_2C_1C_2} \Rightarrow W_n = \frac{1}{\sqrt{R_1R_2C_1C_2}}$$

$$\frac{W_n}{Q} = \frac{1}{R_2C_2} \Rightarrow Q = W_n \cdot R_2C_2 = \frac{R_2C_2}{\sqrt{R_1R_2C_1C_2}}$$

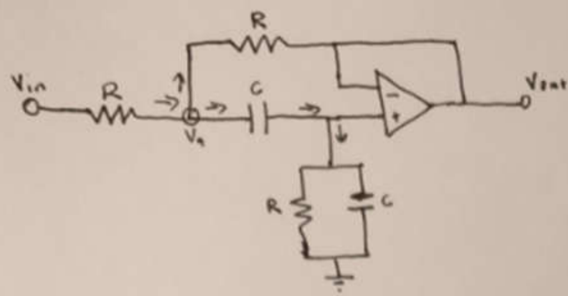
if $R_1 = R_2 = 18\,\Omega$ and $C_1 = C_2 = 10\,\mu F$:

$W_n = 5555.55$

$Q = 1$

From the band pass filter, the convolution of a low-pass and high-pass filter does not differ in quality factor from using KCL from the provided circuit from the lab manual. I think my math is wrong though. Looking at the two bode plots plotted using matlab, I can see the difference in quality factor. The convolution rises to a higher peak.

Sallen-Key



$V_- = V_+ = V_{out}$

$$Z_R \| Z_C = \frac{R/sC}{R + 1/sC} = \frac{R}{sRC + 1}$$

① $$\frac{V_{in} - V_a}{R} = \frac{V_a - V_{out}}{R} + sCV_a - sCV_{out}$$

② $$\frac{V_{out}[sRC+1]}{R} = sCV_a - sCV_{out}$$

from ②: $V_a = V_{out}\left[\frac{1}{sRC} + 2\right]$

from ①: $$\frac{V_{in}}{R} + \frac{V_{out}}{R} + sCV_{out} = V_a\left[\frac{2}{R} + sC\right]$$

plug V_a:

$$= V_{out}\left[\frac{2}{sR^2C} + \frac{5}{R} + 2sRC\right.$$

$$V_{in} = V_{out}\left[\frac{2}{sRC} + 5 + 2sRC - 1 - sRC\right]$$

$$= V_{out}\left[\frac{2}{sRC} + 4 + sRC\right]$$

$$\frac{V_{out}}{V_{in}} = \frac{sRC}{2 + 4sRC + s^2R^2C^2}$$

$$\frac{V_{out}}{V_{in}} = \frac{sRC}{2 + 4sRC + s^2R^2C^2}$$

$$= \frac{\left[\frac{1}{RC}\right]s}{s + 4\left[\frac{1}{RC}\right]s + \frac{2}{R^2C^2}}$$

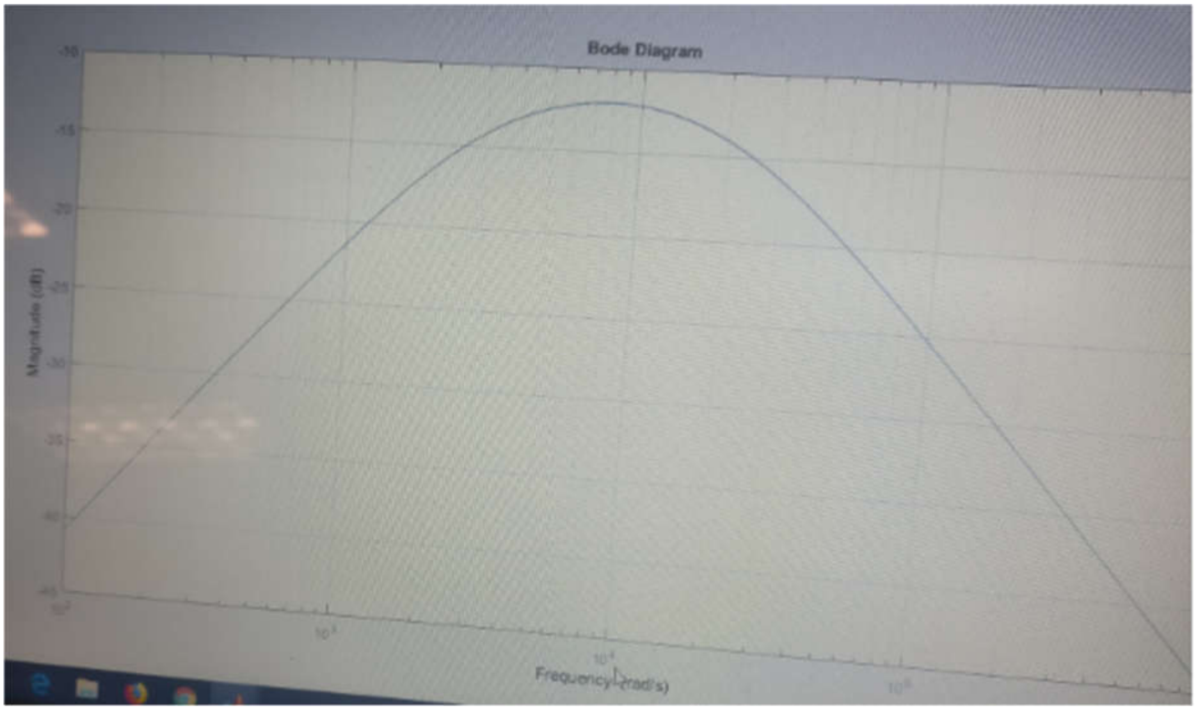
$$\omega_n^2 = \frac{2}{R^2C^2} \Rightarrow \omega_n = \frac{\sqrt{2}}{RC}$$

$$\frac{\omega_n}{Q} = \frac{1}{RC} \Rightarrow Q = \omega_n \cdot RC = \sqrt{2} = 1.414$$

$R = 18\,\Omega$ and $C = 10\,\mu F$

$$\boxed{\therefore \omega_n = 7856\ Hz \quad \text{and} \quad Q = 1.414}$$

$$\frac{V_{out}}{V_{in}} = \frac{\frac{\omega_n}{Q}s}{s^2 + 4\frac{\omega_n}{Q}s + \omega_n^2}$$



From the calculations I made, the sallen-key circuit has a higher quality factor. Using matlab to plot the bode plot, it can be observed that the peak goes much higher as well.

Part6 – Experimental Validation of Analog Filtering

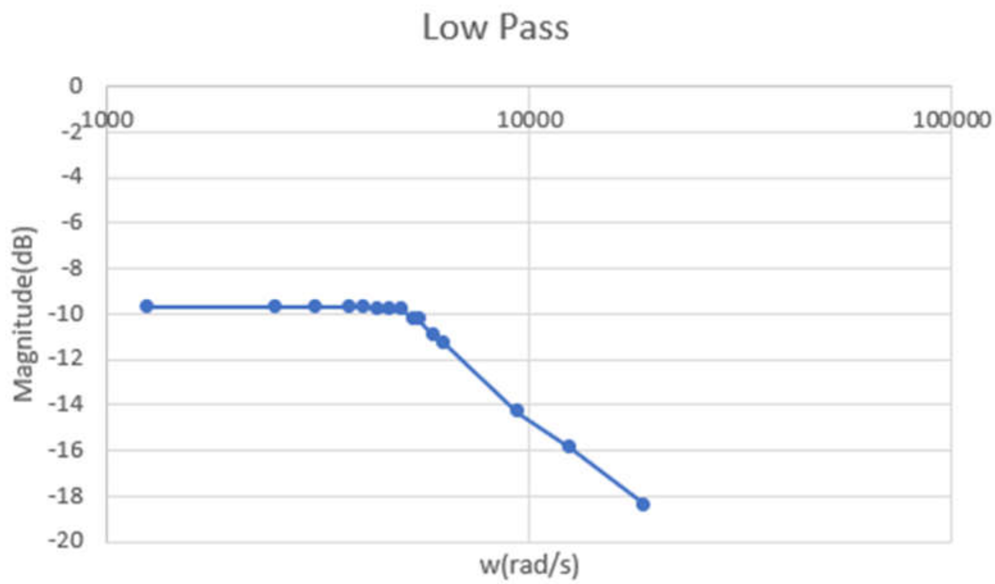
<u>Method:</u>

For part 6, I wired up circuits one by one through the signal generator and oscilloscope. On the signal generator, I set the amplitude to 5V outputting sine waves. Instead of using the sweep function, I manually adjusted the frequency and observed the peak-to-peak value of the output on the oscilloscope. I noted my data(frequency vs peak-to-peak) on excel. Once I obtained enough data, I used calculated angular frequency and magnitude using the following formulas:

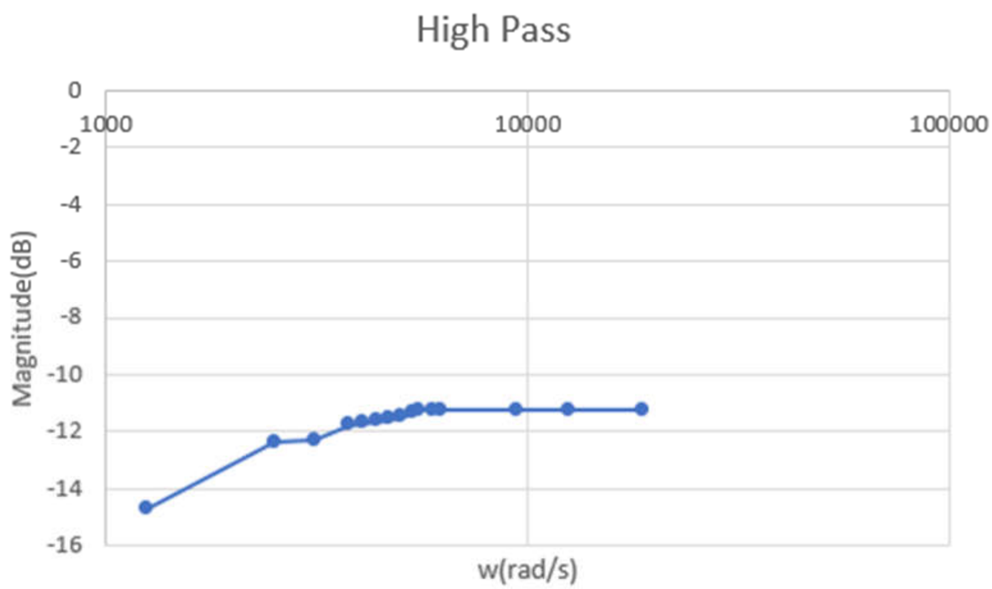
$$\omega = 2\pi f \qquad 20\log\left(\frac{Vout}{Vin}\right)$$

where Vout is the peak-to-peak value and Vin is 5V.

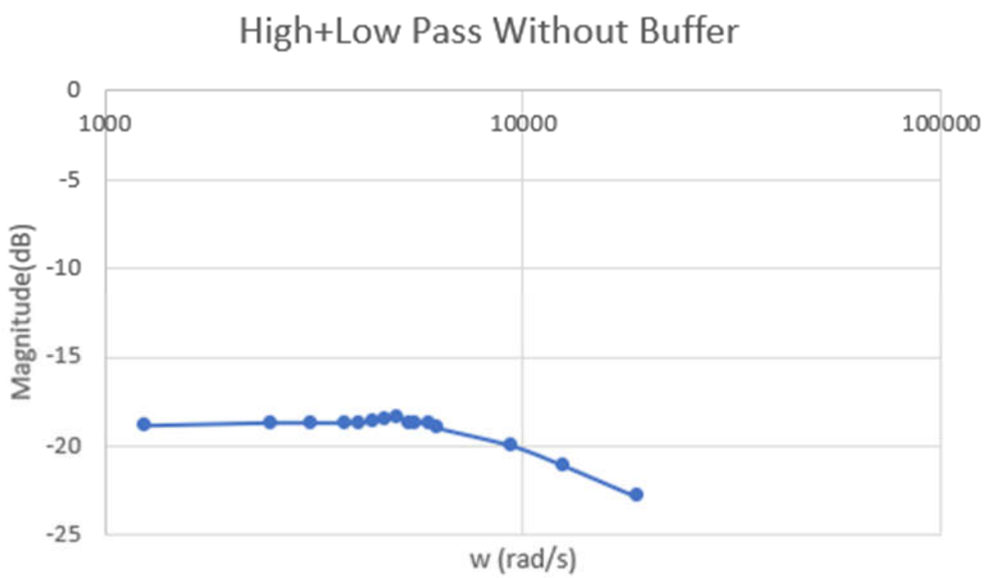
I used excel to do all the math and plotted angular frequency vs magnitude for each circuit. The results are shown below. The plots have the x axis in log base 10 scale but the y axis is kept normal.



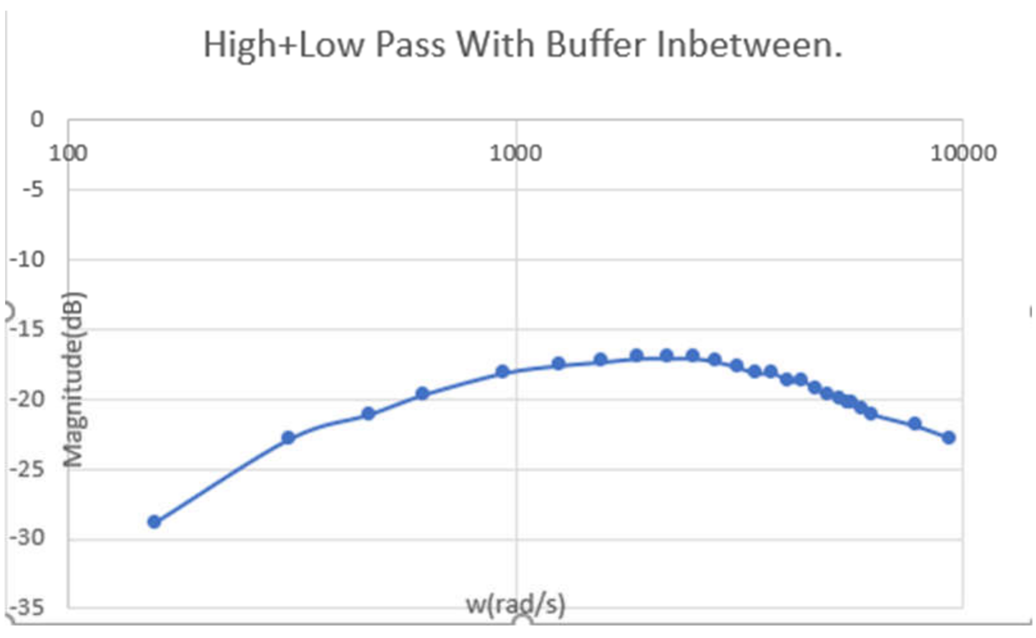
Comparing the experimental to the theoretical, they both start to decay at around the same spot. The peak-to-peak value started to decay passed 880Hz signal from the signal generator. This is seen at around 5555.55 rad/s on the graph.



For the high-pass, I observed it struggle a bit to get to the peak but it still peaked where expected at about 880Hz or 5555.55 rad/s.



The bandpass without buffer has differs a lot from the theoretical plot. From the data that I recorded, it seems to act almost like a low-pass. The highest peak-to-peak value I got was at 750Hz rather than the expected 880Hz although the difference was not drastic.



Adding a buffer in-between the low and high pass filter, I get a very nice plot between angular frequency vs magnitude. This is a much better plot than the previous one without the buffer. This matches the theoretical much more nicely.

Conclusion

Designing this lab for part one through four was simple and didn't take up too much time. The major time sink of the lab was part 5 and 6. Part 5 was the most challenging part as it required me to refresh on old information that I had forgotten. The math was brutal, and many mistakes were made. Part 6 was not too difficult and was the experimental extension of part 5.

The first four parts of the lab was about interfacing the flex and vibration sensors. I found these sensors to be cool and they got me thinking of applications for them especially for virtual reality. Interacting with these was not too difficult if you know what linearization is and make sure you test these sensors before connecting it to the uno32.

I found this lab to be very fun and informative up until part 5 and 6. I did not run into too many issues and think I did a good job of planning to make sure I don't kill the uno32. The only part I'm worried about is part 5 since algebra can easily breed mistakes. Its finally over and looking forward to the next lab.