Pulse Width Modulated Digital to Analog Converter

Zachary Stow

Abstract:

The purpose of our final project was to be able convert an input of a pulse width modulated signal into a linear output voltage signal. Essentially only letting the DC component frequency of the PWM signal to pass through, while filtering out all the higher frequencies that make up the signal. The filter that was utilized was a 2nd order Bessel LPF. The PWM signal that was used as an input into the filter had a frequency of 500kHz. Therefore, the Bessel filter had a cutoff frequency of 500kHz in order to only let the fundamental frequency of the PWM through. After constructing the filter, the output linear voltage was really low, so a non-inverting op-amp with a high gain was implemented in order to increase the output voltage to around 0.5V. The measured peak to peak ripple of the output voltage was determined to be 12.2mV. From there the effective number of bits was determined to be 4.52.

Introduction:

PWM signals work great for encoding digital signals and driving electrical loads, such a motor for instance. The reason it can be utilized to encode digital signals is because its either on or off—which is the digital aspect of the the PWM signal. Yet, since it's a square wave it can be shown from the Fast Fourier Transform that its comprised of many frequencies. The many frequencies that comprise of the square wave show the analog aspect of the PWM signal.

Therefore, the PWM signal can be converted into a linear analog voltage simply by filtering out all the higher frequencies and only letting the DC component pass through. Below, in Figure 1, a 500kHz, 50% duty cycle, square wave's FFT was plotted. I choose to design a Bessel LPF with a cutoff frequency at 500kHz in order to filter out all the higher frequencies and only letting the fundamental through—this converted the PWM signal into an analog signal.

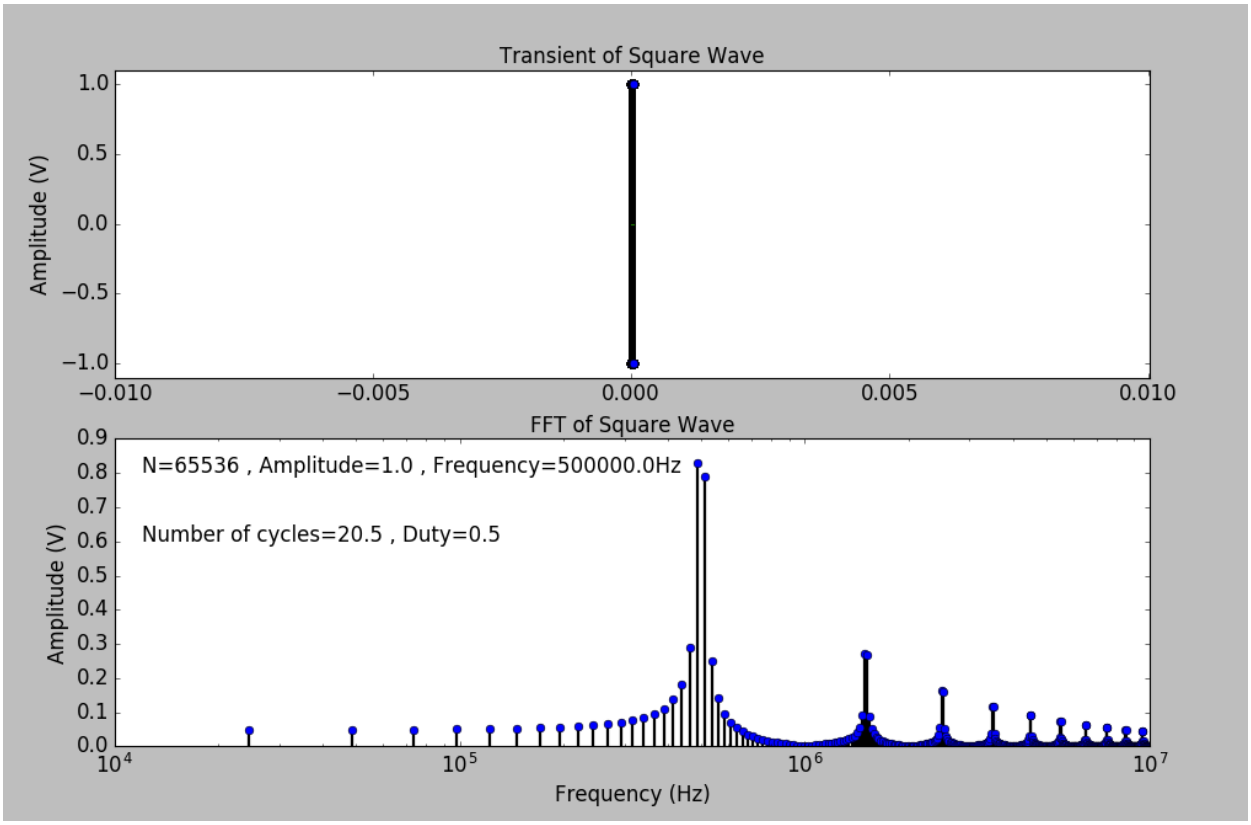

Figure 1: FFT of a 50% duty cycle 500kHz square wave, showing all the frequencies that comprise of the square wave.

A second order filter was picked for implementation for it had a steeper slope after the cutoff frequency, which helped to make sure any higher frequencies right after 500khz were filtered out (or had very little amplification).

Another important feature in regards to determining how well the DAC is working is the effective number of bits (ENOB) that the converter has. The higher the ENOB bits the more accurately the signal is being converted—or the better the resolution.

DAC is very practical, since the many digital applications we have today need to be able to communicate with the many analog applications of today and the past. In this project we are just converting the digital input into a linear output voltage of a signal value, but many filters today take a varying digital input and create a varying linear voltage output that's an analog equivalent.

Theory:

Transfer function of a 2nd order Bessel LPF generated using Python:

$$H(s) = \frac{9.87x10^{12}}{s^2 + 5.44x10^6 s + 9.87x10^{12}}$$

DF2 implementation of the 2nd order Bessel LPF

$$H(s) = \frac{9.87x10^{12}}{s^2 + 5.44x10^6 s + 9.87x10^{12}} \times \frac{s^2}{s^2}$$

$$H(s) = \frac{\frac{9.87x10^{12}}{s^2}}{1 + \frac{5.44x10^6 s}{s} + \frac{9.87x10^{12}}{s^2}}$$

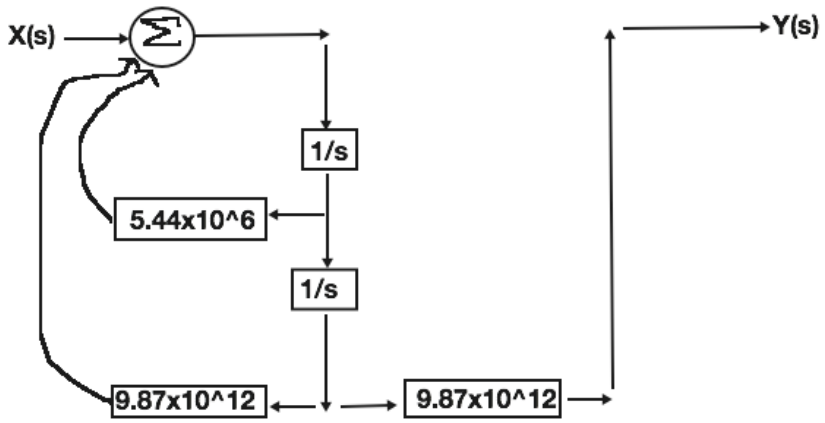

Figure 2: DF2 block diagram of the 2nd order Bessel LPF.

Method used to calculate the values of the resisters and capacitors to get a cut off frequency of 500kHz:

$$a_1 = 5.44x10^6 = \frac{1}{R_1 C_1} \quad let\ C_1\ equal\ 1nF$$

$$R_1 = \frac{1}{5.44x10^6(1nF)} = 184\ \Omega$$

$$a_2 = b_2 = 9.87x10^{12} = \frac{1}{R_1 R_2 C_1 C_2} \quad let\ C_1\ and\ C_2\ equal\ 1nF$$

$$R_2 = \frac{1}{9.87x10^{12}(184\ \Omega)(1nF)(1nF)} = 550\ \Omega$$

Effective number of bits calculation:

$$N_{effective} = \frac{\log\left(\frac{1}{\frac{12.2\ mV}{1V} + \frac{500kHz}{16MHz}}\right)}{\log(2)} = 4.52\ bits$$

Procedure/Methodology:

First, Python was utilized to calculate the transfer function and bode plot of the filter. The second order Bessel LPF was chosen, because its transfer function was much easier to implement over Chebyshev and Elliptical (Chebyshev and Elliptical required techniques not learned in class). Furthermore, the Bessel filter cutoff frequency was chosen to be at 500kHz, since a 500khz PWM was going to be used. Next the transfer function was taken over to Ltspice in order to implemented it into behavior voltage. The behavior voltage resembled the same bode plot as what Python had simulated. The method that was chosen to implement the transfer function was DF2, which resulted in needing four op-amps. In order to make sure the Bessel LPF was operating properly a bode plot of the circuit was generated with a AC input of 1V—while also utilizing real op-amps. In addition, the upper rail was connected to a voltage regulator that converted 9V into 5V. The bode plot of the Bessel filter was close to the ideal, except for it had a gain of about 9dB.

Next an input square wave with a 50% duty cycle and a frequency of 500kHz was feed into the Bessel filter as an input. The output had a really low voltage so a gain was added to the circuit in order to get it to about 0.5V. From there the peak to peak ripple voltage was measured in order to perform the calculation that determined the effective number of bits the circuit had. Lastly, the A_{vol} of the LT1006 open loop gain and GBW was calculated using Ltspice.

Results and Discussion:

```
In [16]: %run "/Users/Zstow/Desktop/Bessel.py"
a= [  1.00000000e+00   5.44139809e+06   9.86960440e+12]
b= [  9.86960440e+12]
```

Figure 3: 2nd order Bessel transfer function coefficients generated using Python.

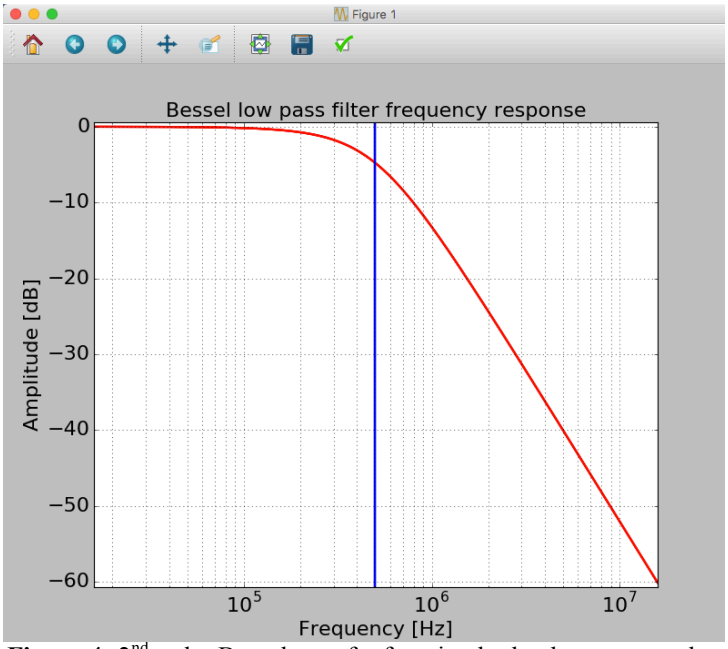

Figure 4: 2nd order Bessel transfer function bode plot, generated using Python.

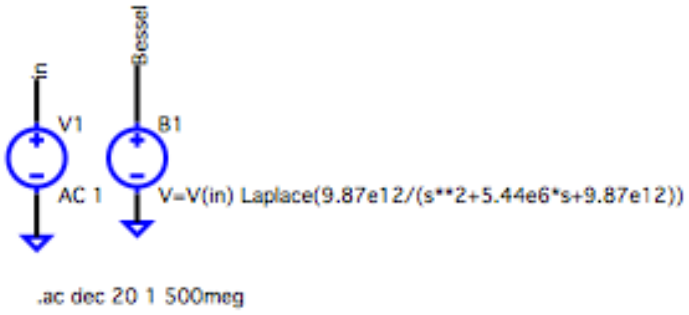

Figure 5: Behavior voltage utilized to plot the 2nd order Bessel transfer function.

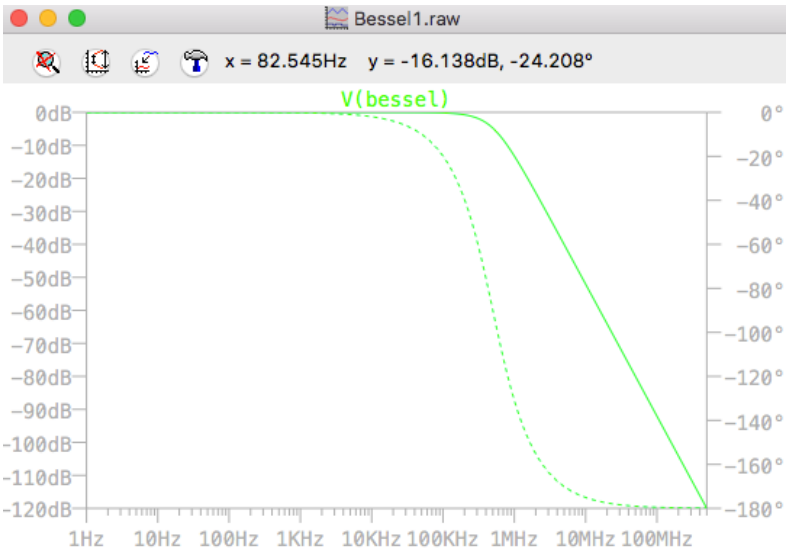

Figure 6: Ideal 2nd order Bessel transfer function bode plot, generated using the behavioral voltage.

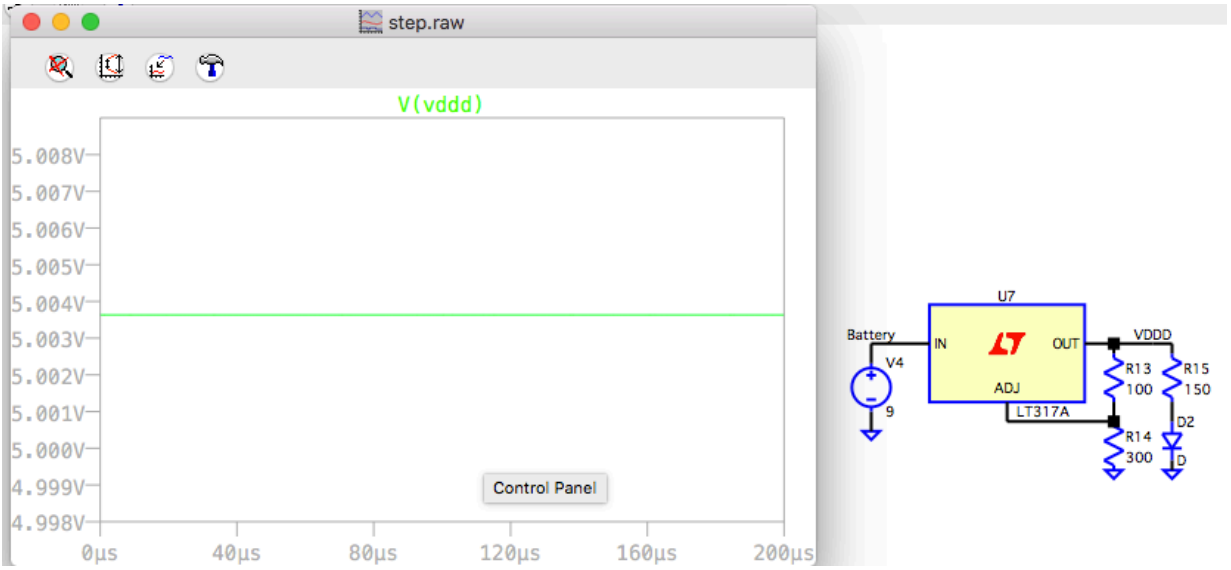

Figure 7: The voltage regulator LM317A is dropping the 9V input to 5V to supply VDD to the op-amps.

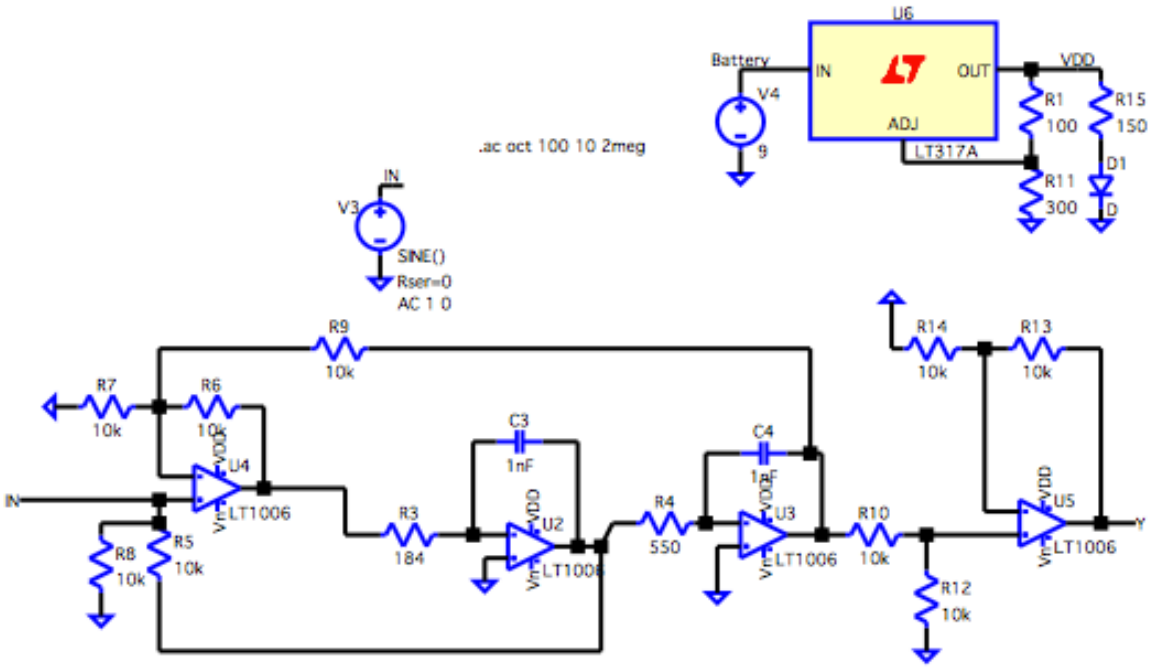

Figure 8: Implementation of the 2nd order Bessel transfer function into Ltspice. Also a voltage regulator was utilized for VDD.

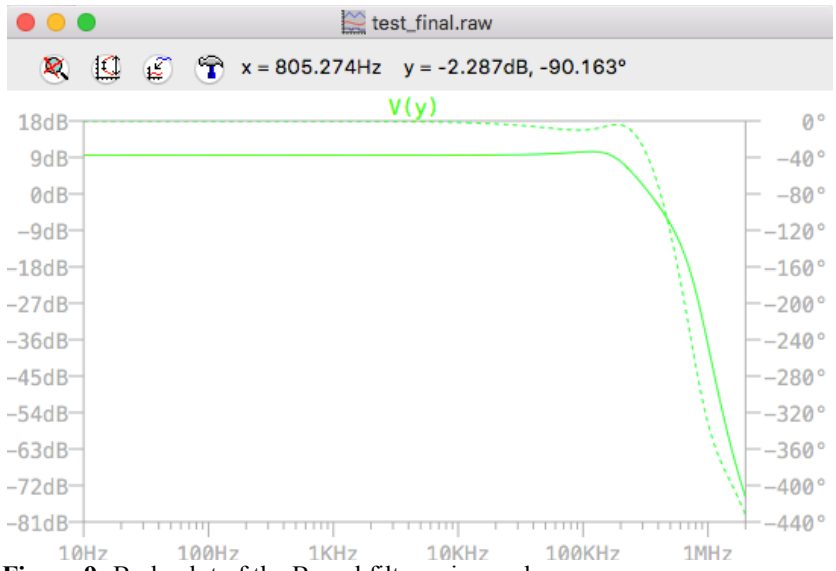

Figure 9: Bode plot of the Bessel filter using real op-amps.

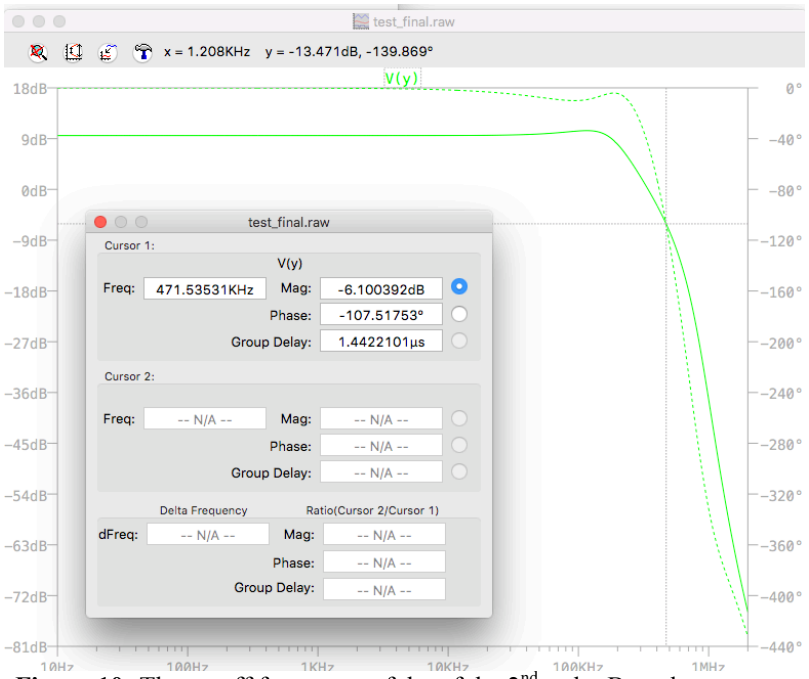

Figure 10: The cutoff frequency of the of the 2nd order Bessel was determined to be at 471.5kHz

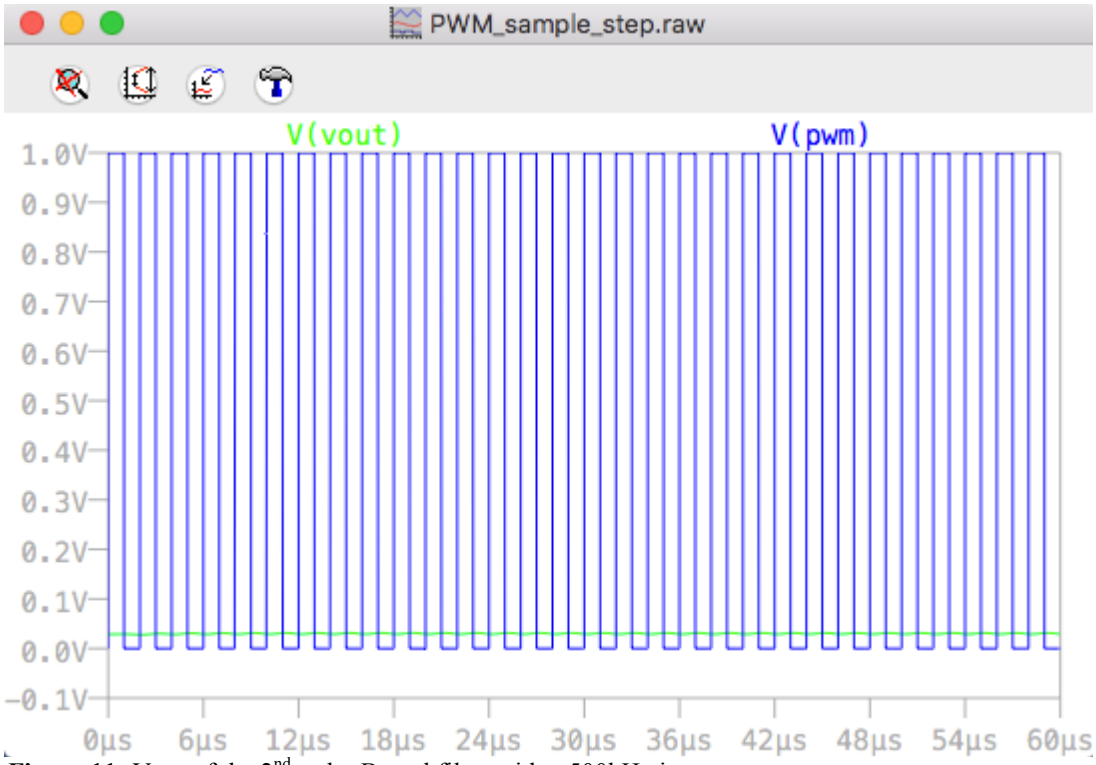

Figure 11: Vout of the 2nd order Bessel filter with a 500kHz input—pwm.

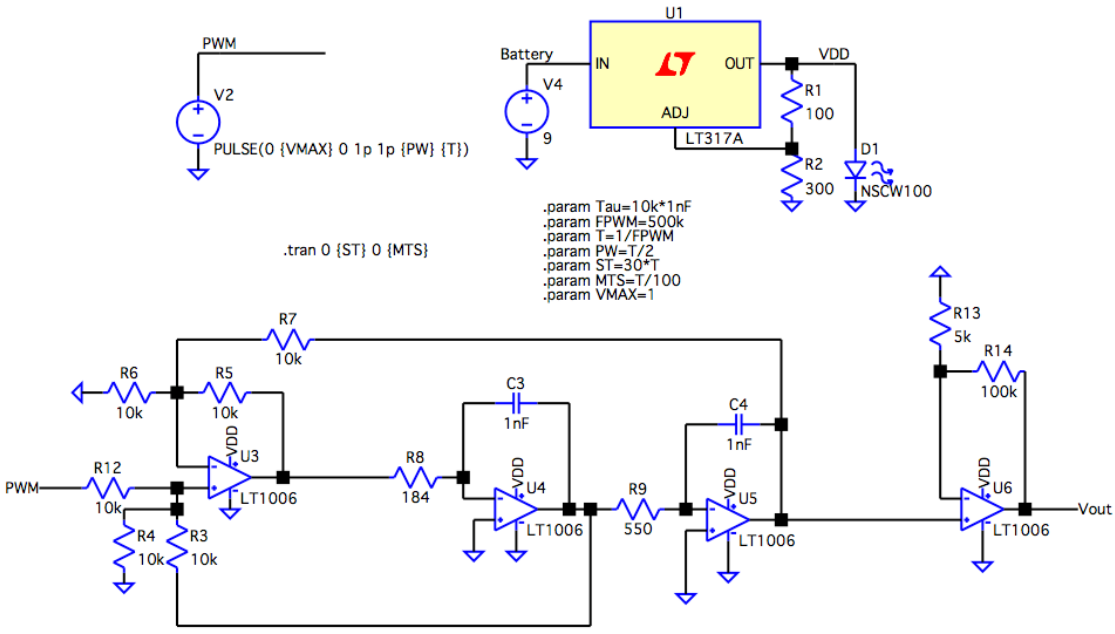

Figure 12: 2nd order Bessel circuit with the last unneeded summing op-amp, turned into a non-inverting op-amp.

Figure 13: New output of the 2nd order Bessel filter—Vout.

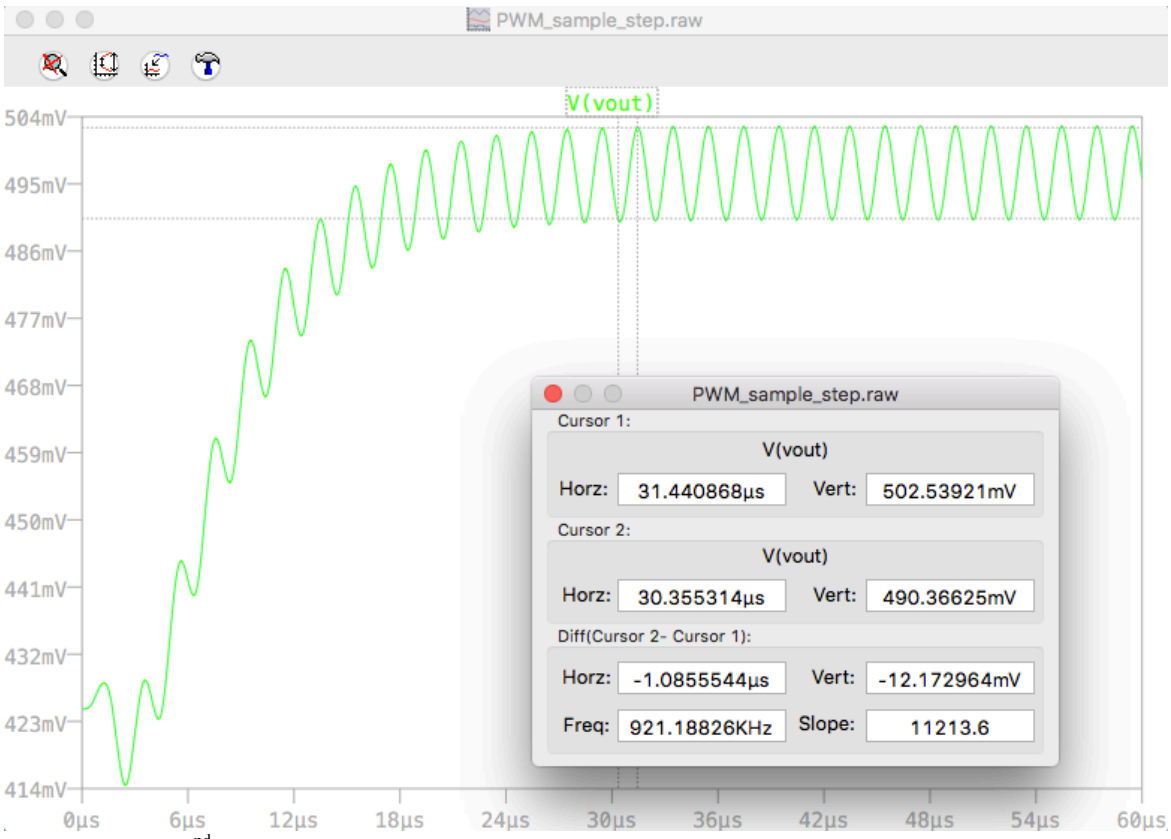

Figure 14: The 2nd order Bessel LPF peak to peak ripple was measured to be 12.2mV.

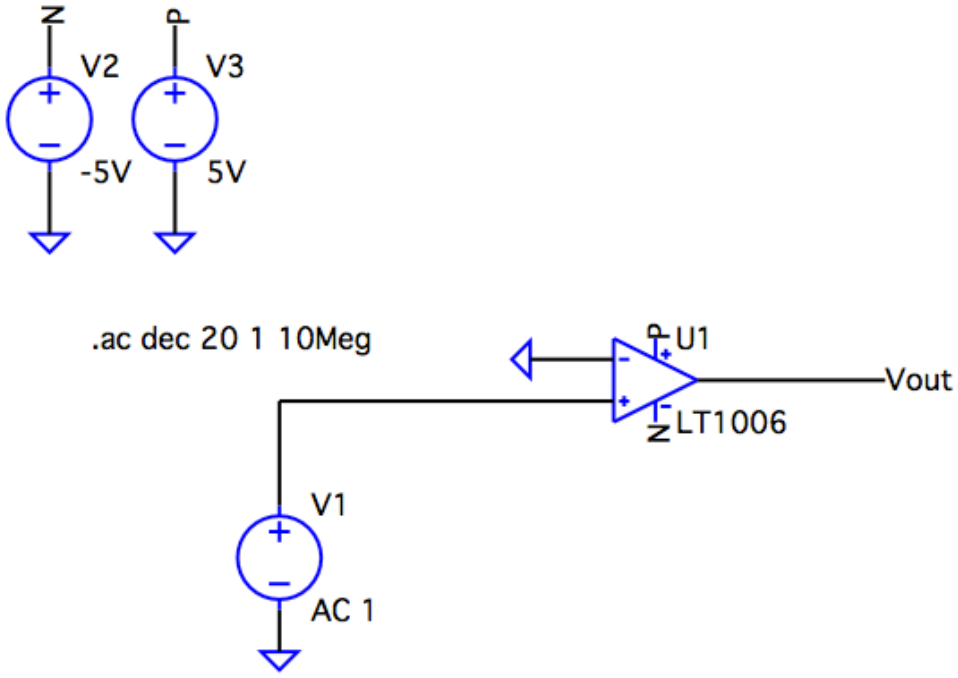

Figure 15: Circuit involving the LT1006 set up with no feedback, allowing the op-loop gain and GWB to be messed.

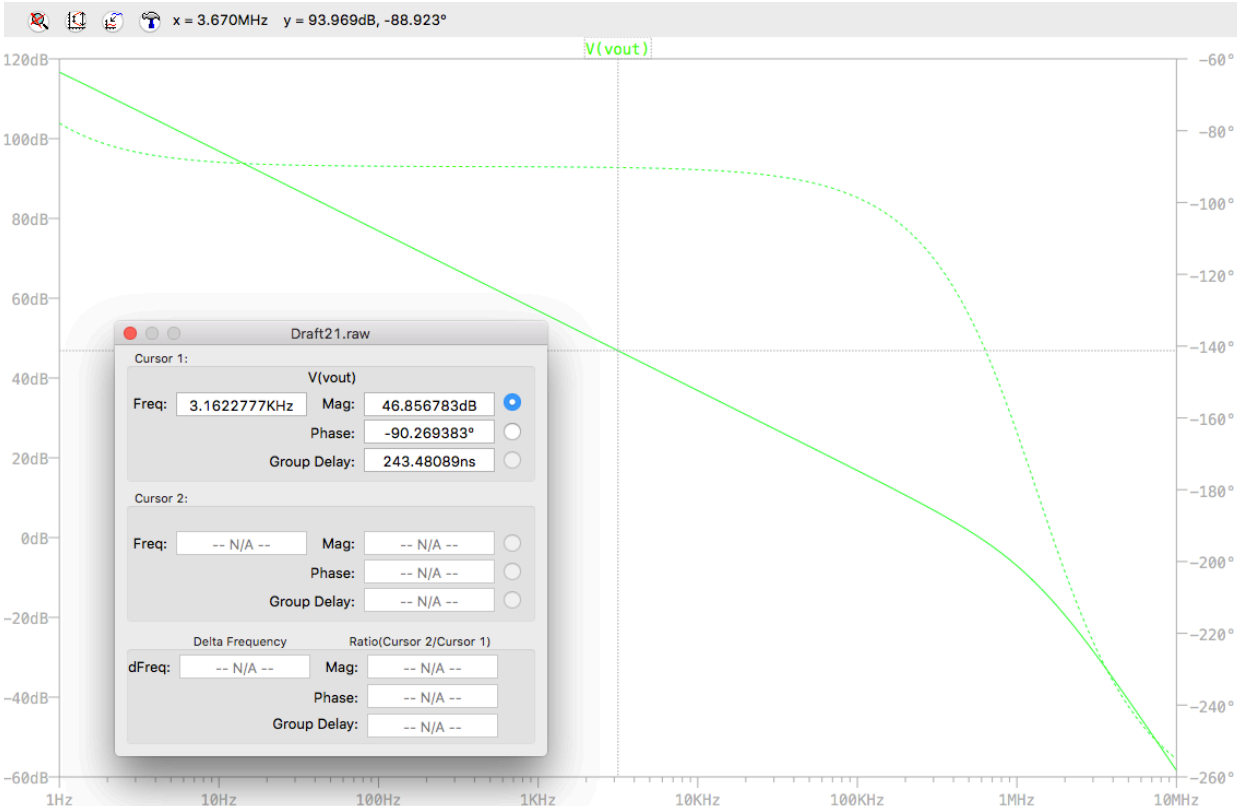

Figure 16: Graph of figure 15 showing part of the process which involved determining A_{vol}.

	AC 1	AC 10	AC 100	AC 1000	AC 10k	AC 100k	AC 1Meg
A_{vol} (V/V)	218.8	18.3	24.5k	200k	w00k	515k	69M
GBW(Hz)	568.3K	2 M	4.8M	10M	24.2M	50M	100M

Table 1: Calculated values for A_{vol} and measured values the GBW at varies AC inputs.

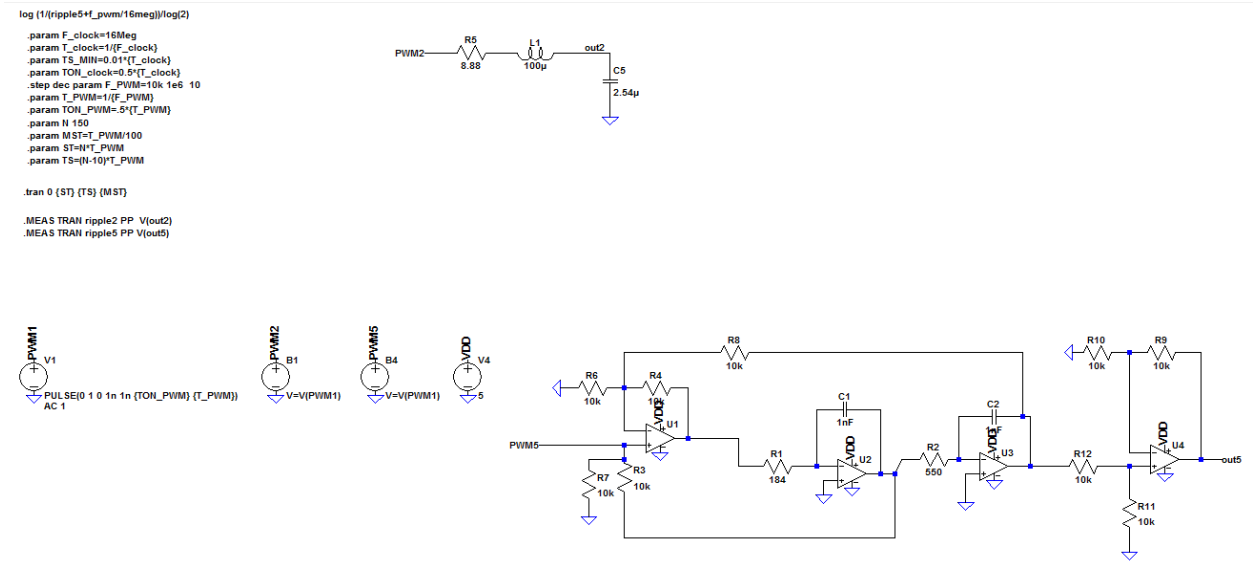

Figure 17: Build circuit in file given by Prof. Parents to calculate the effective number of bits

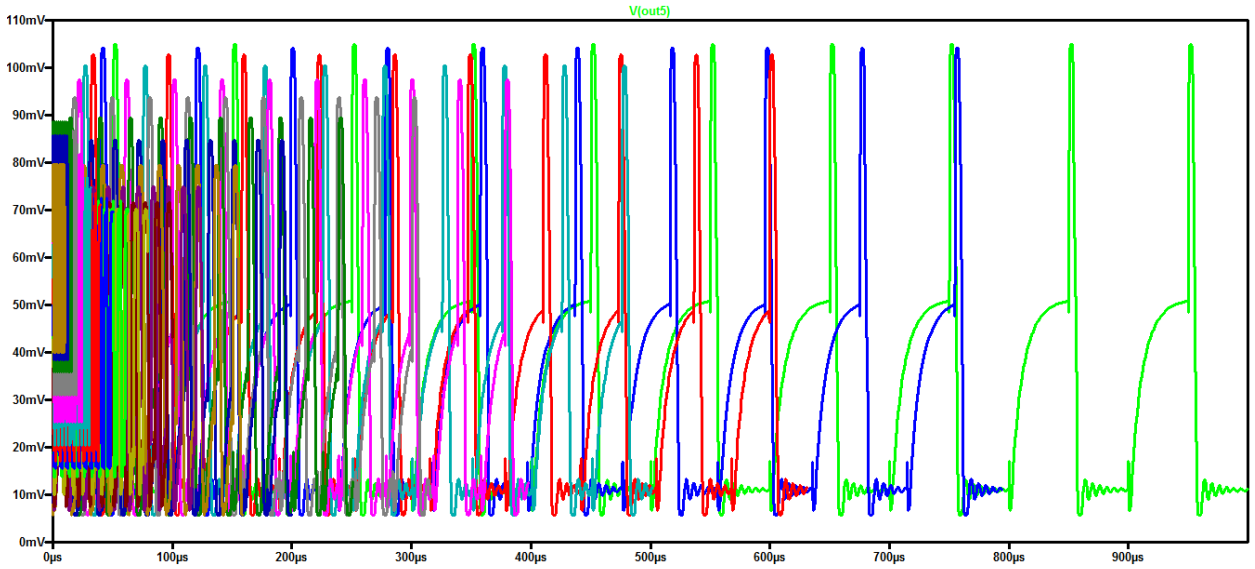

Figure 18: Testing the circuit at many different PWM frequencies.

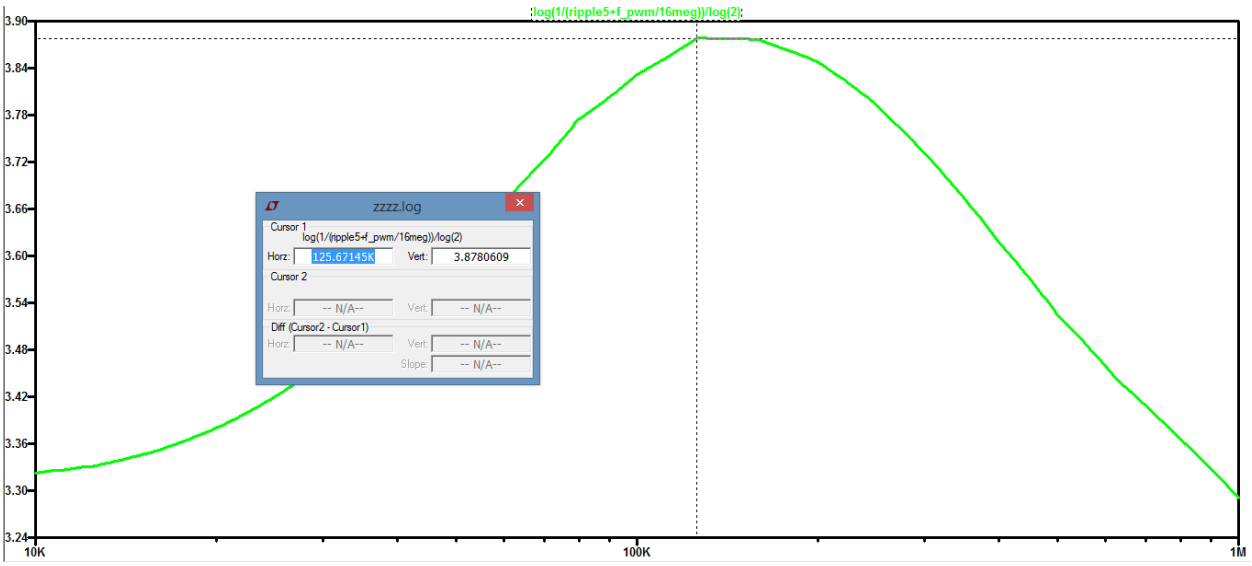

Figure 19: Determined the clock frequency that would produce the highest effect number of bits for my circuit.

Everything seemed to be running smoothly until the 500kHz square wave was used as an input to the circuit. It was also realized at that point that the second summer wasn't needed, for I had originally implemented the transfer function with a second summer also summing an input of one and had realized the summer wasn't needed when I removed it. Yet, when the output of the 500Khz square wave was generated through the filter on Ltspice, I realized the second summer could be used as a non-inverting op-amp in order to increase the gain and get an output around 0.5V.

I also learned upon completing the report that I should have calculated the PWM frequency that would have gotten the highest effective number of bits for my circuit. So I decided to run the simulation anyways. I obtained highest number of bits for my circuit should be 3.87 with a clocking frequency input of 125.6KHz.

Conclusions:

There was a huge time crunch in regards to completing this project, which limited the amount of time needed to completely understand the project. For example, I realized upon completing the project that I should have determined the frequency that would have maximized the ENOB for my circuit. Nonetheless the arbitrary frequency value I choose still managed to generate an ENOB's around 4, which isn't too bad. Also, the implementation was going great up until the 500kHz square wave was used an as input, for I couldn't figure out why the output voltage was so low. Turning the unneeded summer (that was there by mistake) into a non-inverting op-amp with a high gain resulted in a solution for the very low linear voltage output. The peak to peak ripple was increased after implementing the non-inverting op-amp and ended up causing the circuit to have an $N_{effective}$ of 4.52.

Lastly, the highest ENOB was calculated by using Ltspice which determined the PWM clocking frequency that would maximize this value. I did this lastly for I didn't realize I should have done it first. A curious out come from the results was that the maximum number of bits calculated by hand for my circuit was higher than the value that Ltspice had determined.

References:

[1] S.B. Bramble, 'Active Filter Design', [Online]. Available: http://www.simonbramble.co.uk/techarticles/active_filters/active_filter_design.htm. [Accessed: 3- DEC- 2016].

[2] 'Turn or PWM into a DAC', [Online]. Available: http://www.allaboutcircuits.com/technical-articles/turn-your-pwm-into-a-dac/. [Accessed: 3- DEC- 2016].

[3]K.B. Burgess, ‘PWM to DC Voltage Conversion’ 2015. [Online]. Available: http://www.egr.msu.edu/classes/ece480/capstone/spring15/group10/Application%20Notes/Kyle.pdf. [Accessed: 3- DEC- 2016].