Annoted report removes names, code, and several key calculations. Final design was implemented without function generators and was able to run on single Arduino Due microcontroller and a power supply. PCB arrived after presentation so pictures were not supplied in the final report.

Table of Contents

Introduction

The focus of this report is on the design and prototype testing of a DC to AC inverter which efficiently transforms a DC voltage source to an AC source. Electronic devices run on AC power; however, batteries and some forms of power generation such as photovoltaic cells produce a DC voltage so it is necessary to convert the voltage into a source that devices can use.

A low voltage DC source is inverted to create an AC source and this design will take advantage of high-frequency switching ability of MOSFETs. The DC source is transformed into an AC signal using a pulse width modulation control signal. The control signals are used to switch four MOSFETs on and off instantly to create a sinusoidal output. The four MOSFETs are connected in H-bridge form which includes two high side switches and two low side switches. In this report, we detail the design phase of the inverter, then how the inverter's controls were implemented with a digital approach using a microprocessor for the control system, how the circuit was constructed, what adjustments we were required to make, and how the inverter design can be improved if more time was available.

Theoretical Analysis

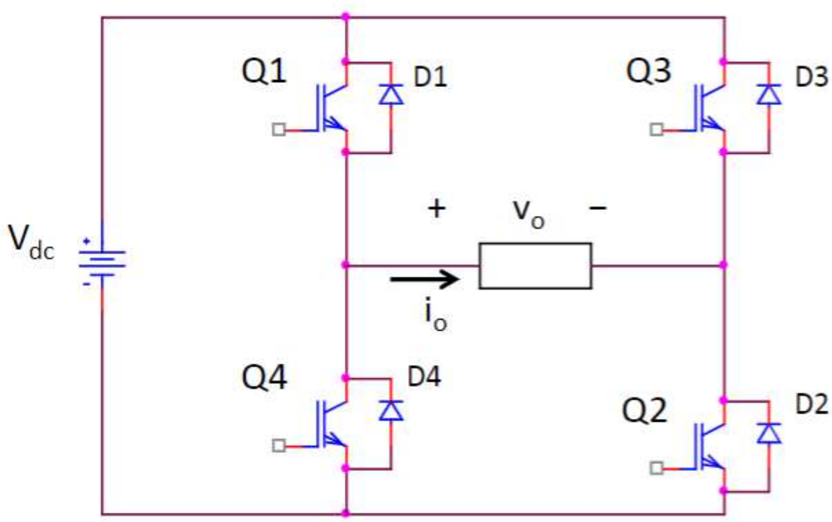

Figure 1. Basic DC-AC inverter with H-bridge

Figure 1 shows the basics of the H-bridge converter using four MOSFETs as switches. The inverter input voltage is a DC voltage source. The load for this project is a simple RL load. Q1 and Q2 are controlled by an identical control signal, while another control signal is sent to the gates of Q3 and Q4 switches. Q1 and Q3 are referred to as the high-side switches, and Q2 and Q4 are the low-side switches.

The control signal of Q1 is to be the inverted signal of Q3 to ensure that both switches are not turned on at the same time, which would cause current to flow to the ground.

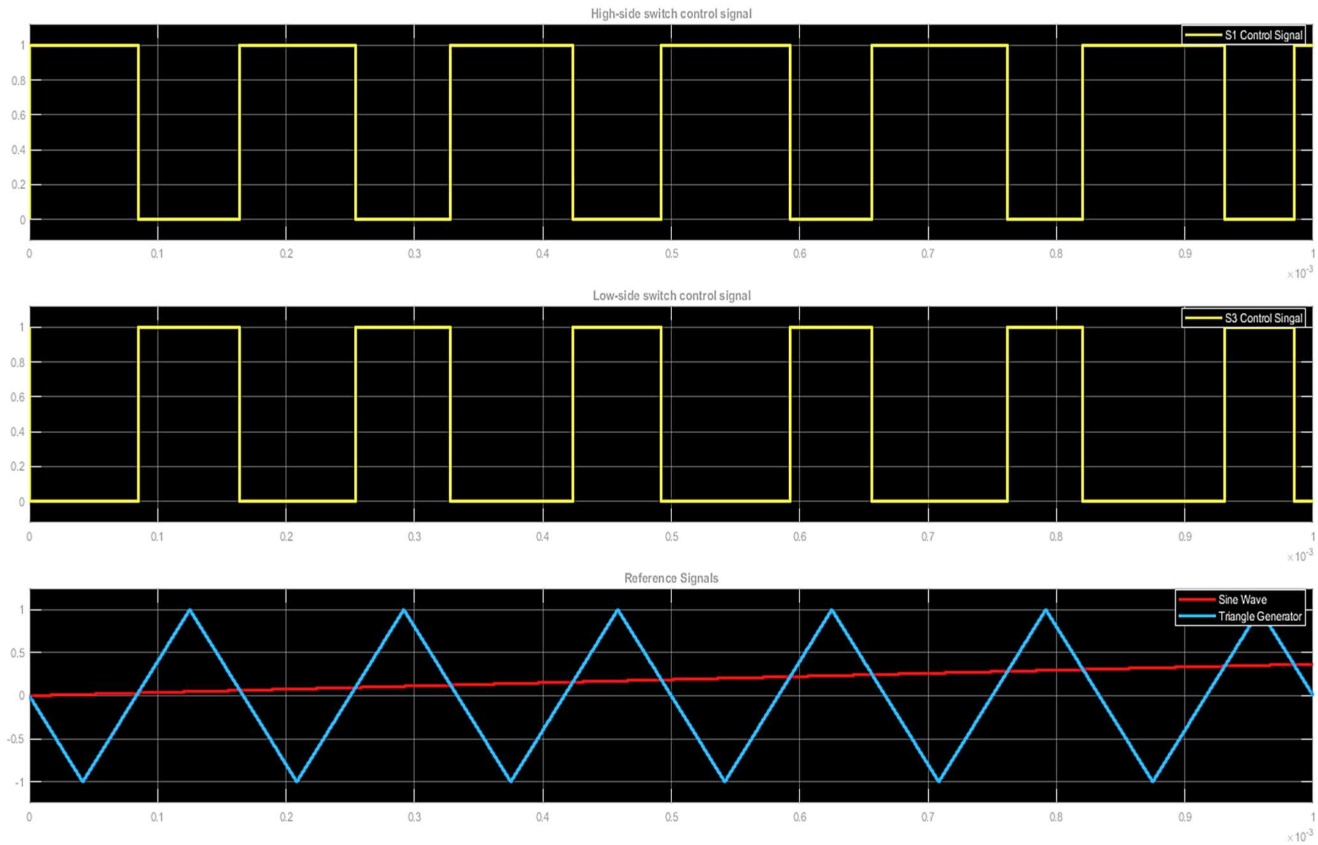

Figure 2. The control signal for high-side and low-side switches on the same leg of H-bridge

Pulse width modulation is based on the comparison between a reference signal and a carrier signal. The reference signal is sinusoidal and has the same frequency as the desired frequency of the output signal. The carrier signal is a triangular signal and provides the switching frequency for the 4 MOSFETs. From Figure 2, when the sine wave is greater than the triangular wave, the high-side switch control signal is set to be high which then turns on the MOSFET. During this time period, the low-side switches have to be off, thus its control signal needs to be set low. This means the control signal is inverted from the high-side switch control signal from the same leg of H-bridge.

Bipolar Switching

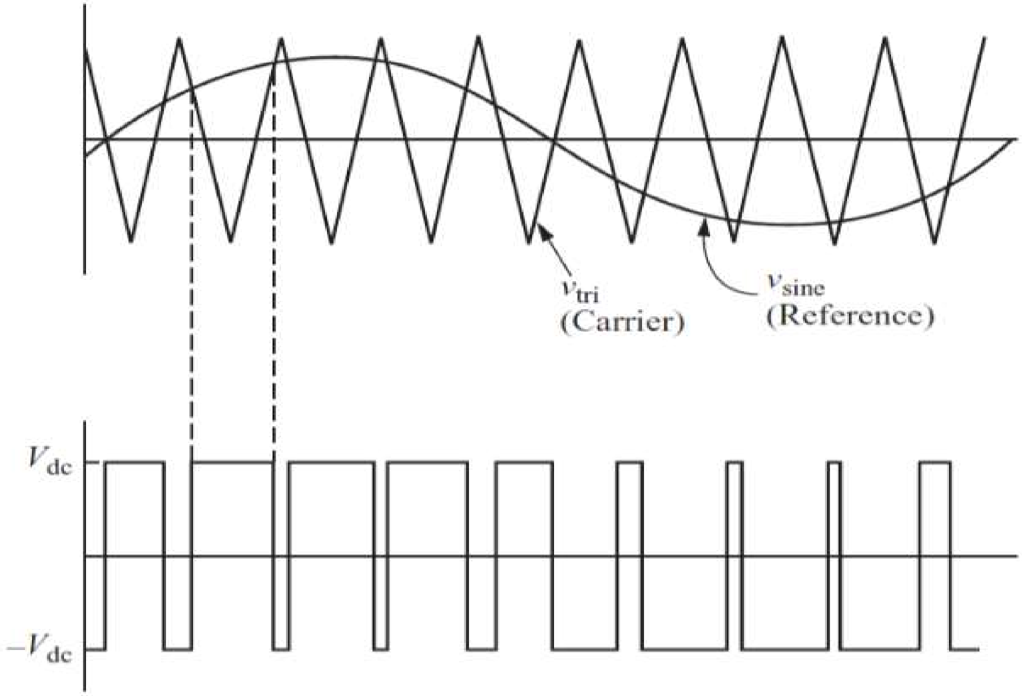

Figure 3. Bipolar switching with a carrier signal and a reference signal, and the output voltage.

Bipolar switching is based on the comparison between a carrier signal and a reference signal. As seen in Figure 3, when the reference signal is smaller than carrier signal, switches Q1 and Q2 are switched on, which allow current to flow through the load in the positive direction (left to right). On the other hand, when the reference signal is greater than the carrier signal, the positive control signal will be sent to the gates of switches Q3 and Q4, which will turn them on. When Q3 and Q4 are conducting, current can flow through the load in a negative direction (right to left). The output voltage alternative between maximum and minimum at high frequency.

Unipolar Switching

Unipolar switching has a similar idea; however, there are two reference signals, a positive sine wave, and a negative sine wave. When the positive sine wave is greater than the triangle wave, switches Q1 and Q2 are on, which results in positive voltage between point A and ground. When the negative sine wave is greater than the triangle wave, the control signal to switches Q3 and Q4 are active; therefore, these switches turn on and create positive voltage between point B and ground. The voltage across the load is the voltage difference between point A and B, as seen in Figure 4. The current flow in the positive direction for half of a period, then it becomes negative for the second half of the period.

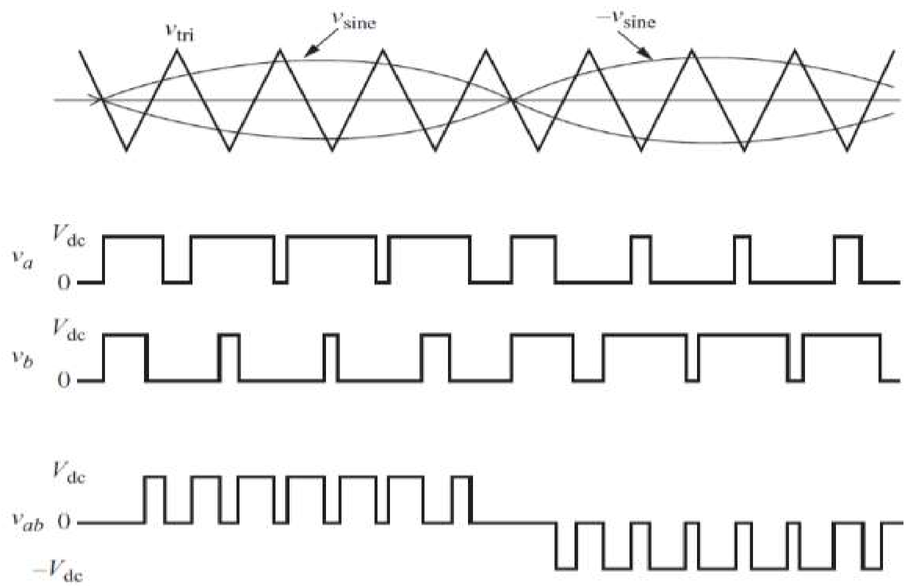

Figure 4. Unipolar switching with a carrier signal and two reference signals, and output voltage.

Filter Design

The load for this project is modeled as an RL low pass filter. Because of the high switching frequency, output signals will have high-frequency noise. To eliminate this noise, the RL filter plays the role of a low pass filter to attenuate high-frequency noise. Since the switching frequency is 6kHz, the cut-off frequency was chosen to be 1/5 of switching frequency, this leads to the cut-off frequency of 300Hz. A resistor of 100Ω was chosen, and the inductance value was calculated as shown in Eqn 1.

$$L = \frac{R}{2\pi f_{cut-off}} = 5\ mH \qquad Equation\ 1$$

Simulation

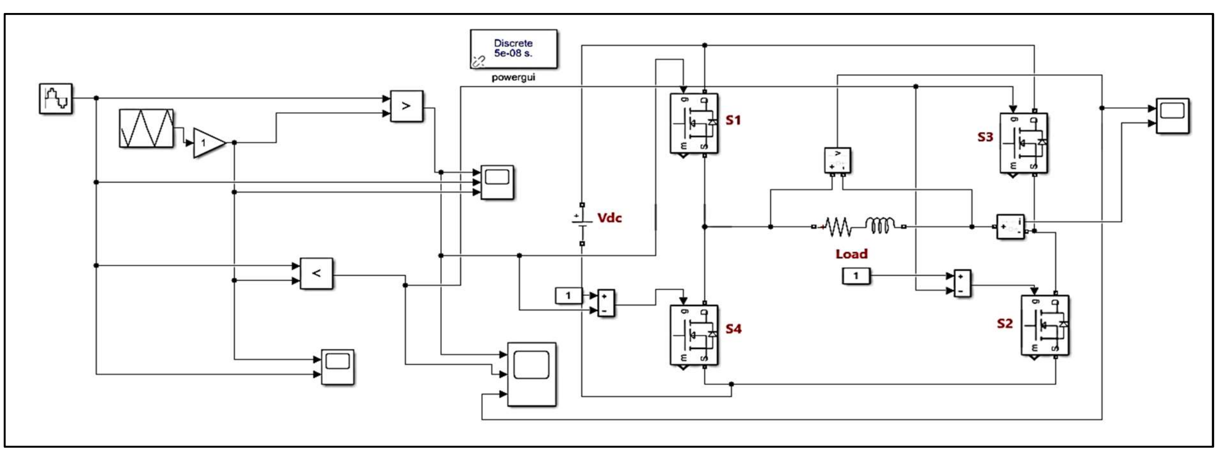

Figure 5. Schematic of bipolar PWM inverter Simulink model.

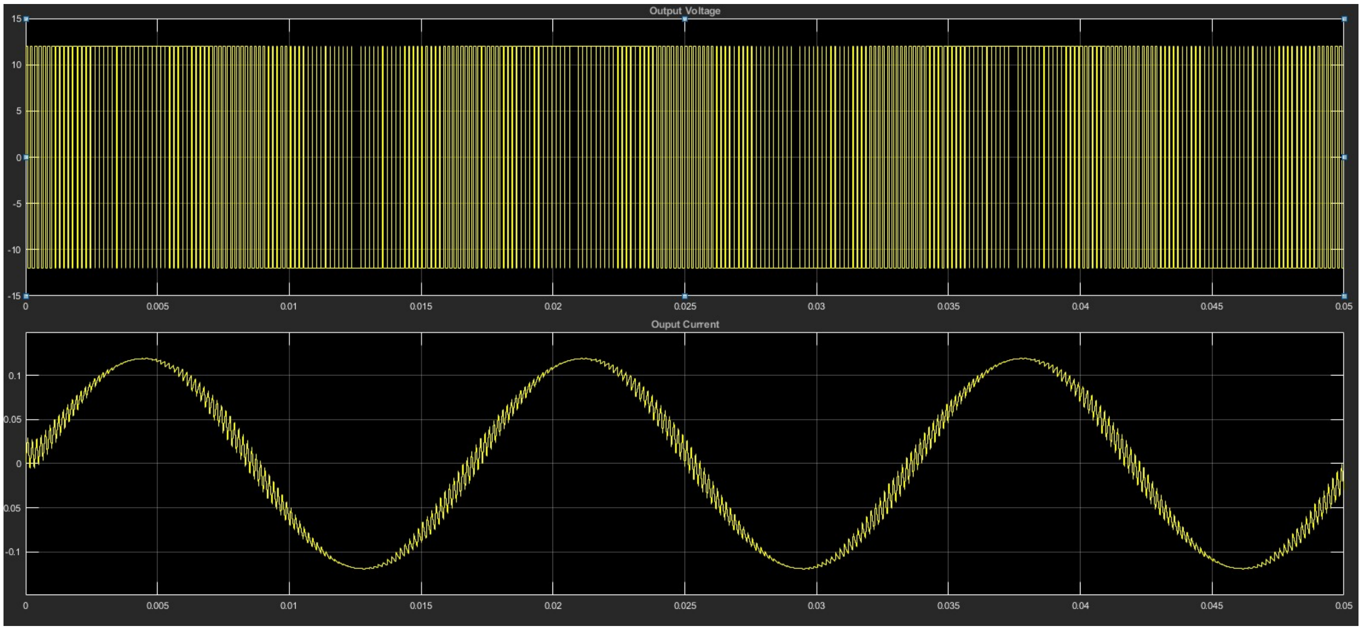

Figure 6. Bipolar switching – Output Voltage and Current

Simulink was utilized to build and theoretically test our design. Figure 5 shows the model built in Simulink with one reference and one carrier signal representing bipolar switching. Output voltage and output current are shown in Figure 6, respectively. Even though the RL filter was designed to cut off high-frequency noise, output current still experienced undesired noise. Because of this fact, the inductance value was increased to 30mH and those simulation results are shown in Figure 7.

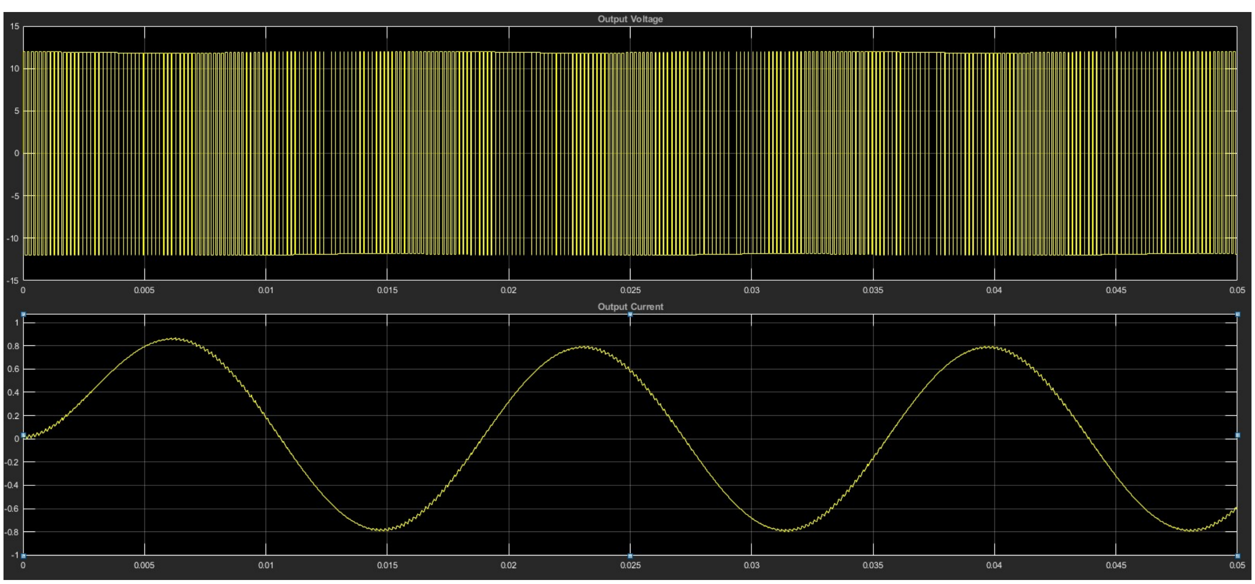

Figure 7. Bipolar switching – Output Voltage and Current with improved RL filter

Unipolar switching was also simulated and the block diagram for that circuit can be seen in Figure 8. In this case, we have two sinusoidal reference signals that are compared to the carrier signal to help realize a unipolar switching inverter. The output current and voltage across the improved load can be seen in Figure 9. The results of Figure 9 do show the correct response of a unipolar switching inverter.

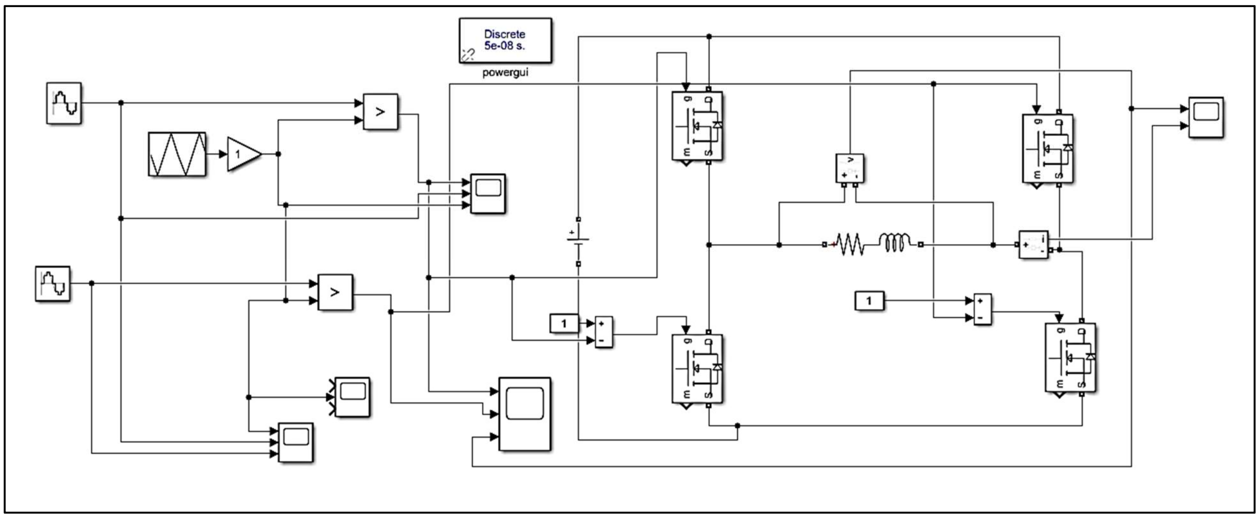

Figure 8. Schematic of unipolar PWM inverter Simulink model.

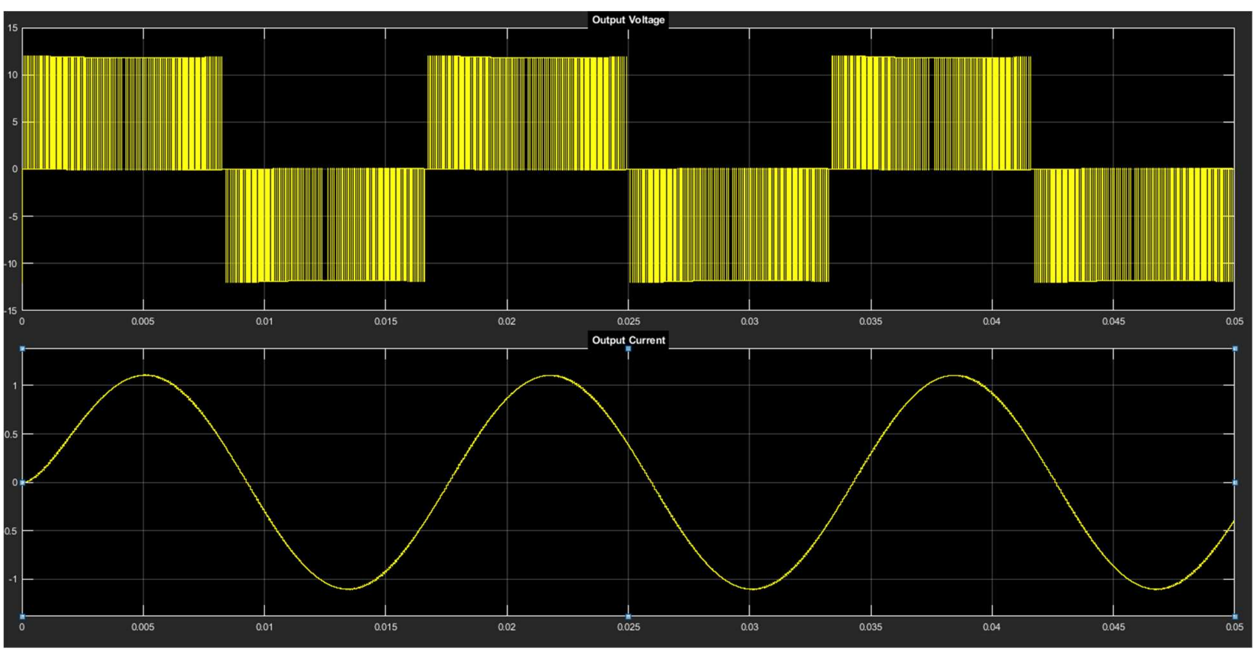

Figure 9. Unipolar switching – Output Voltage and Current with improved RL filter

Control Signal Design

Bipolar Switching

To create the bipolar switching control signal, an Arduino Due microcontroller was used along with two function generators. Since the microcontroller can only utilize analog inputs of 0-3.3V safely, a DC offset was introduced to both sine and triangle waveform inputs of $1V_{DC}$ while adjusting the amplitudes to $1V_{PP}$. This configuration ensured both signals would be strictly between 1V to 3V and could in no way damage the microcontroller.

With the analog inputs created, the microcontroller simply compared the instantaneous values of the reference sine wave at 60Hz and the switching triangular wave at 6kHz prompting switches Q1 and Q2 to be on when the sine amplitude is greater than the triangle amplitude while Q3 and Q4 remained off. When the triangle waveform amplitude was greater than the sine's, then Q1 and Q2 were off and Q3 and Q4 were on. Figure 10 shows the bipolar control signals for Q1 and Q3 created using the Arduino code in appendix 1. Note that they appear nearly identical except one is inverted which is expected for the bipolar configuration.

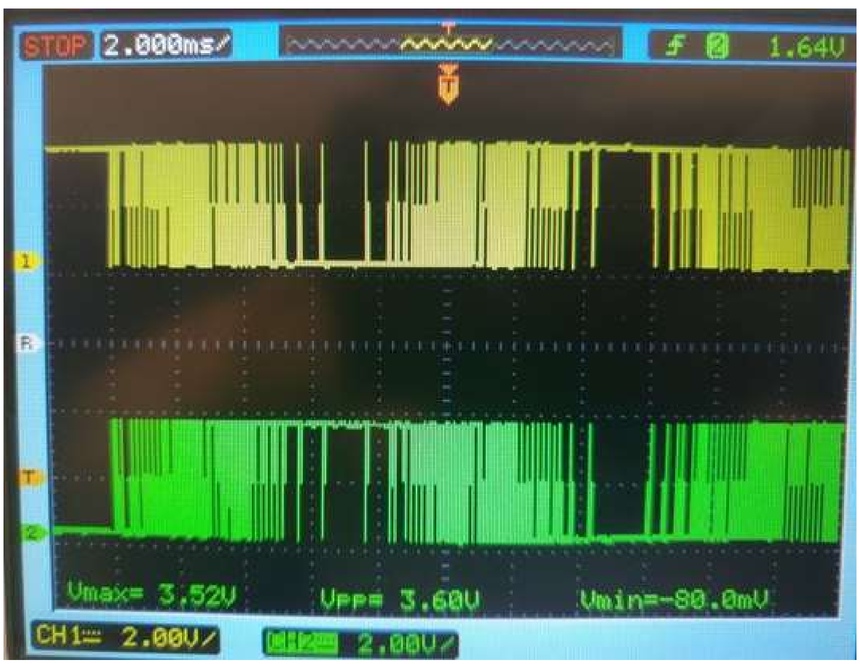

Figure 10. Bipolar PWM control signals produced by Arduino Due.

Unipolar Switching

Using the same configuration for the analog inputs as the previous case, the unipolar control signal was created by a similar method. Since the Arduino's analog inputs are stored as a digital value, it was determined that the negative amplitude reference sinusoid should follow the equation of sin2=-sin1+1123 as shown in appendix 1. This equation prompts sin2 to read the digital equivalent of 1V when sin1 is 3V and the digital equivalent of 3V when sin1 is 1V. Now, switches Q1 is on and Q4 is off when sin1 is greater than the triangle waveform and Q1 is off and Q4 is on if otherwise. Accordingly, Q3 is on and Q2 is off when sin2 is greater than the triangle waveform and Q3 is off and Q2 is on if otherwise. Figure 11 shows the unipolar control signal created by the Arduino microcontroller using the code in Appendix 1. Note that both signals are not inverted of each other and there appears to be a slight shift as expected from the previous section.

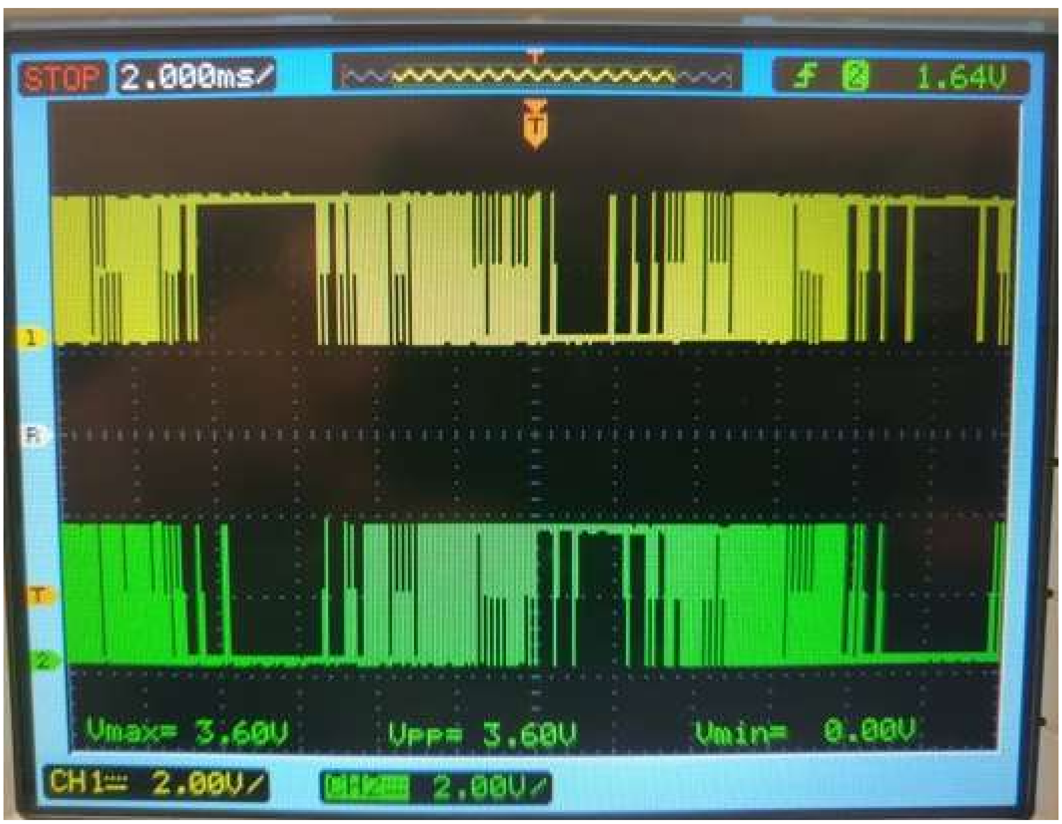

Figure 11. Unipolar control signal for Q1 and Q3.

Design Implementation

This pulse-width modulation DC to AC inverter design can be split into two parts, hardware, and software. The hardware portion is the physical components needed to build the actual circuit and the software portion is the ability for a written code in a microcontroller to output the needed control signal to switch each of the MOSFETs in unipolar and bipolar mode.

Circuit Construction

The hardware portion or physical portion of the design is made up of the necessary components needed to construct the inverter. Each iteration of the circuit was implemented and tested on a breadboard. The required components can be seen in Table 1. 4 Power MOSFETs were used for the switches, 4 optocouplers were used for isolation of the control signal circuit and switching circuit, the load was an RL load with the resistor being 120Ω and the inductor being 30 mH, and 4 more 120Ω resistors were used as pull-up or pull-down resistors for the sources. Not mentioned in the table is the fact that 3 separate power supplies were used with four separate sources.

Table 1 shows the required physical components for the PWM inverter design.

Physical Components			
Category	Type	Qty	Model
Switches	Power MOSFET	4	IRF 840
Isolation	Optocoupler	4	4n30
Load	30 mH Inductor	1	n/a
	120Ω Resistor	1	n/a
Misc	120Ω Resistor	4	n/a

The software portion of the design is the necessary component used for the creation of the control signals that go to each switch of the inverter. In this case, an Arduino Due was used to create an acceptable control signal that was able to handle the high frequency needed to switch the MOSFETs and create a sinusoidal output.

Difficulties

When constructing the circuit, we ran into some difficulties with getting an acceptable output. The first problem was our initial setup of the circuit. The first design was set up the circuit in Figure 1, which gave us an output voltage and current that looked right but had an immense amount of noise thus making that setup unusable. It is also important to note that in that setup, only one DC voltage source was used. [redacted] next step should be to add optocouplers to the circuit. The most notable reason for the need for optocouplers is to isolate the control signal from the high side switches. Two optocouplers were used for the control signal for the low side switches as well. After implementing these, the circuit was not outputting anything worth working on so the decision was made to start from scratch. The problem with the circuit design at this point can be attributed to grounding issues of the optocouplers and the overall circuit. A step-by-step approach was utilized in order to check the functionality of each MOSFET and optocoupler. After constructing the circuit in a piecemeal approach, the design outputted an acceptable output voltage and current. At that point, the setup of the circuit was our final design.

Final Design

Final Design Circuit

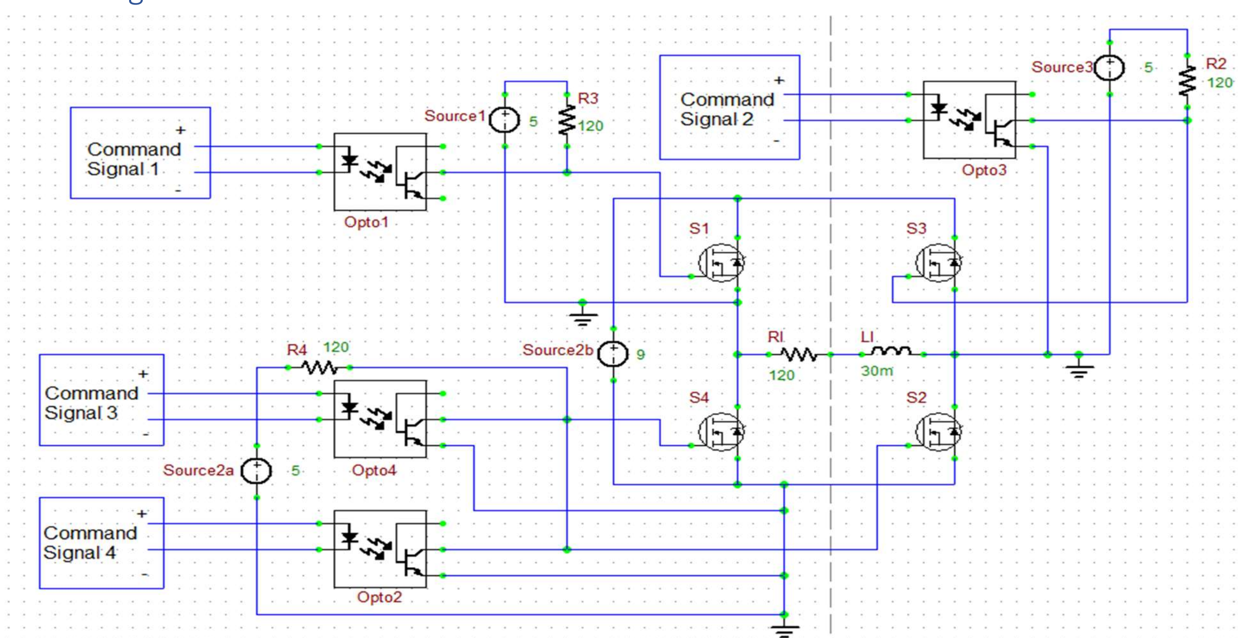

Figure 12. Schematic of the physical PWM inverter circuit.

In order to produce the desired output from the inverter, the final design consists of the usage of one Arduino Due for producing the command signals going to the four MOSFET switches, three power supplies to power to power all four switches along with V_{DC} for the inverter circuit. As shown in Figure 12, the S1 (represents Q1) and S3 (represents Q3) use their own different power supplies and different grounds, meanwhile, S2 (represents Q2) and S4 (represents Q4) use the same power supply that supplies V_{DC} for the inverter circuit and they share the same ground. Altogether there are our different grounding points, one for each power supplies and another grounding point for the Arduino Due's command signals. Building the circuit like so provides isolation for all command signals to the inverter circuit. This will isolate the low power circuit which is the Arduino side to the high side which is the inverter side. In addition, not only does the optocouplers isolate voltages, but the optocouplers also provide noise isolation, thus it helps with eliminating unwanted noise in the output coming from the Arduino. The final design of the circuit was wired onto two breadboards as shown in Figure 13.

Figure 13. The final design of the PWM inverter.

Results

Since the bipolar and unipolar PWM inverter uses the same circuit with the only difference of the command signals coming into the switches of the inverter, the bipolar PWM inverter will be examined first. As shown below in Figure 14, when V_{DC} was 5 volts, the output for a bipolar PWM inverter is ± 5 volts through the full cycle of the waveform which in this case, it has a peak-to-peak value of 10 volts. This waveform will have varying duty cycles and pulses throughout the waveform and the pulses would be thicker during the max and min of the reference sinewave. The voltage output shown was measured with a regular voltage probe, while the output current was measured with a differential probe. In addition, the output current was a nice sinusoidal waveform as expected, and this result is also shown in Figure 14.

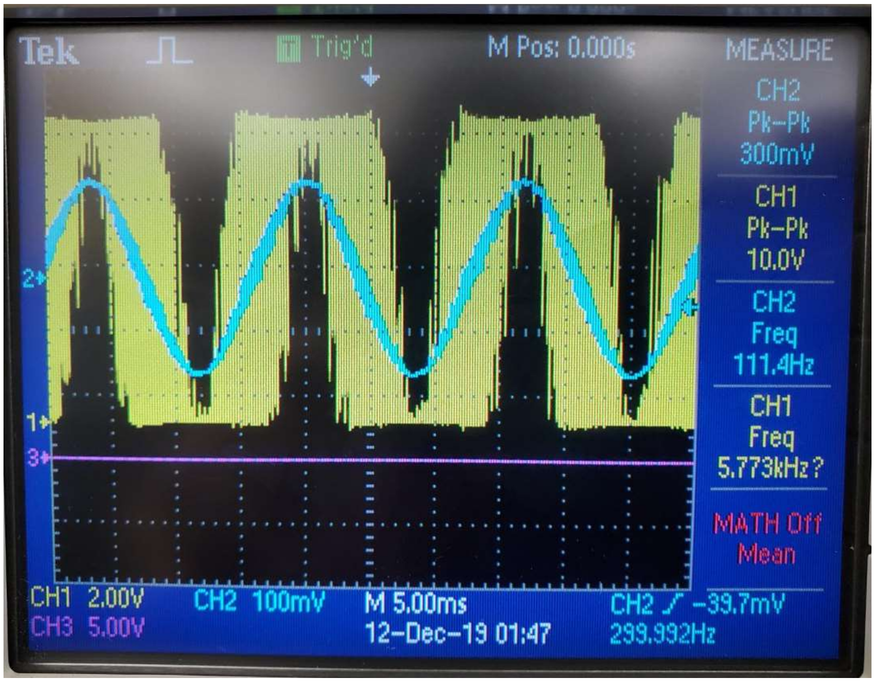

Figure 14. Bipolar PWM inverter voltage and current output with only one differential probe.

Because the load of the inverter is an RL load which means that the load will be an inductive load, it can be assumed that the output current should lag the output voltage. Since it is assumed that the output current should lag the output voltage, the output current will be measured again with a differential probe. This result can be seen in Figure 15, and as shown, there was no change other than the fact that the peak to peak voltage had dropped down to 412 mV. Not sure why the voltage measurement was different with a different probe, but as shown in Figure 15, there still was no lag between the output current and output voltage. This result was then verified but viewing the simulation results of the PWM inverter shown in Figure 6 or Figure 7. Since the simulation results don't show that the output current lags the output voltage, this result was confirmed to be correct. The command signals to S1 and S3 can also be seen in Figure 15 and they are the purple and green waveforms respectively.

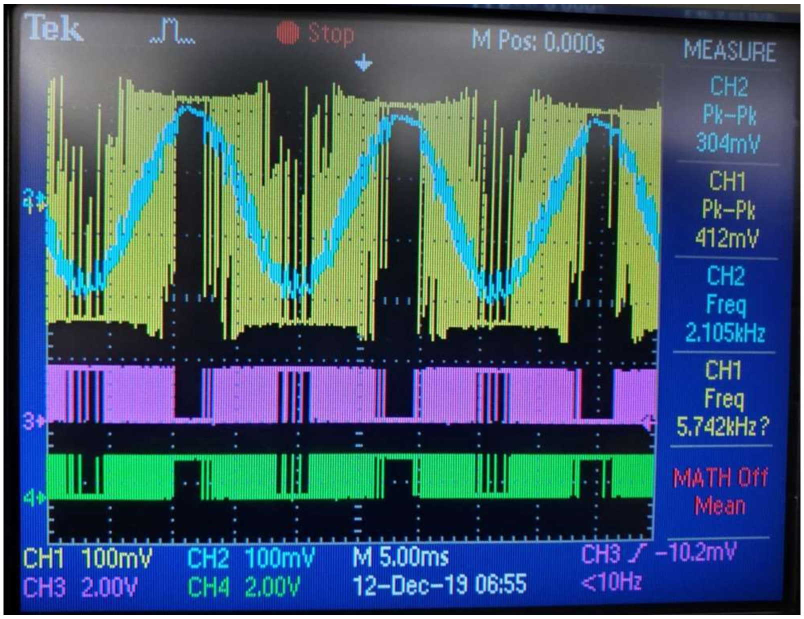

Figure 15. Bipolar PWM inverter voltage and current output measured with two differential probes.

After examining the bipolar PWM inverter, the unipolar PWM inverter was examined simply by changing the command signals going into the switches. This was necessary to see the difference between the output of the bipolar and unipolar PWM inverter. The results of the unipolar PWM inverter can be seen in Figure 16.

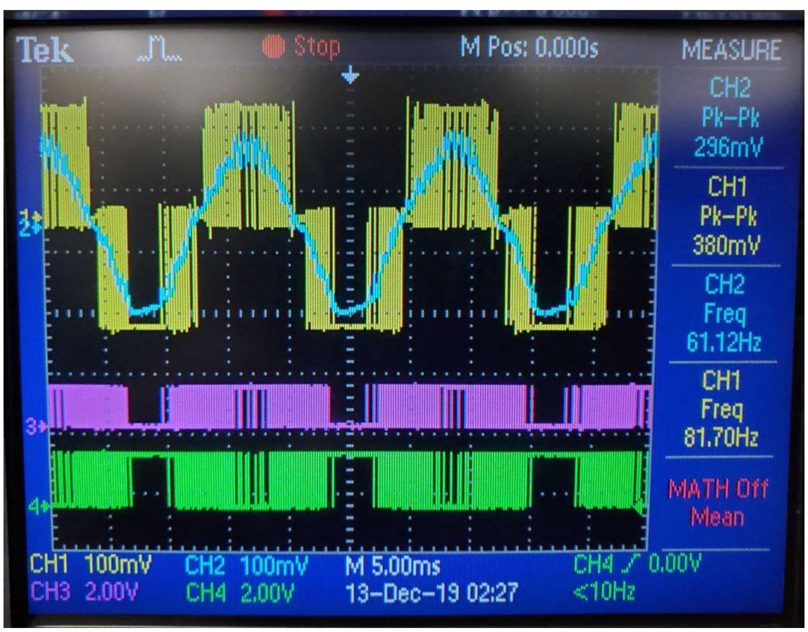

Figure 16. Unipolar PWM inverter voltage and current output measured with two differential probes.

As shown in Figure 16, the results were expected and matched the simulation results of the unipolar PWM inverter in Figure 9. From the results, it can be seen that during the first half of the cycle, the output voltage was only going from positive voltage to zero voltage. On the other hand, during the second half of the cycle, the output voltage waveform was going from zero voltage to a negative voltage. In addition, the output current was a sinusoidal waveform, therefore this result of the unipolar PWM inverter was correct and operated as it should.

Since PWM inverter should also work at low-frequency switching, the inverter circuit was also examined at low-frequency switching. In this case, the switching command going to the S1 would be turned on once and off once in a full cycle, and S2 would also receive the same command but with a slight delay. This can be seen as the green and purple waveform in Figure 17 along with the results of the low-frequency switching inverter. In Figure 17, the yellow waveform represents the output voltage, while the blue waveform represents the output current. Because the switching frequency is very low, the output current waveform cannot be a sinusoidal waveform but takes the shape of the output voltage waveform with some curves caused by the inductor. This result was as expected. In addition, one verification that this result is correct is that when both command signal is off the output voltage is zero. Therefore, overall, all PWM inverter tested and designed for had functioned as expected and gave the right output.

Figure 17. Invert output voltage and current at a low switching frequency.

Discussion

Overall, the design works well. However, if given more time, there are some improvements that could be made. First, the low pass filter should be re-designed. The recommended low pass filter is a Sallen-Key low pass filter, which better eliminates high-frequency noises and harmonic distortion. Since our output voltage is only 10VAC, we can add a step-up transformer to achieve higher voltage which can then be used to power commonly used loads such as fan and light bulb. Our design used four MOSFETs to switch at high frequency; thus, there is power loss in the circuit. One way to improve this is to add a totem pole circuit which includes a PNP and NPN transistor in series. The totem pole circuit would help send control current through MOSFETs' gates faster, resulting in the rising and falling time of MOSFETs being shortened, which decreases switching power loss. Lastly, our final design should be neatly soldered on a prototype board, this would help eliminate potential noise, loose connections, and high resistance of the breadboard to get a better output signal.

Conclusion

This report explores the design and implementation of a DC to AC pulse width modulation inverter operating with bipolar switching and unipolar switching. Both switching modes were simulated using Simulink and an approximate design was obtained for each and an RL load was picked. After confirming that the Simulink models were correct and giving an acceptable output voltage and current, the implementation of the circuit on a breadboard was commenced. Several different configurations of the circuit were tested to try and match the simulated results but there were multiple different problems that came up along the way. Then, after totally rebuilding the circuit from a step by step process, an acceptable output voltage and the current waveform was obtained. In all, the design worked well but if given more time, multiple different improvements would be made to both improve and display the functionality of the DC to AC pulse width modulation inverter.