

Variable Input Constant Output Voltage Regulator

Final Report

Francisco J. Balcazar
&
Alexander Gombert

Project Advisor:
Mr. Gutschlag

Date:
May 15, 2014

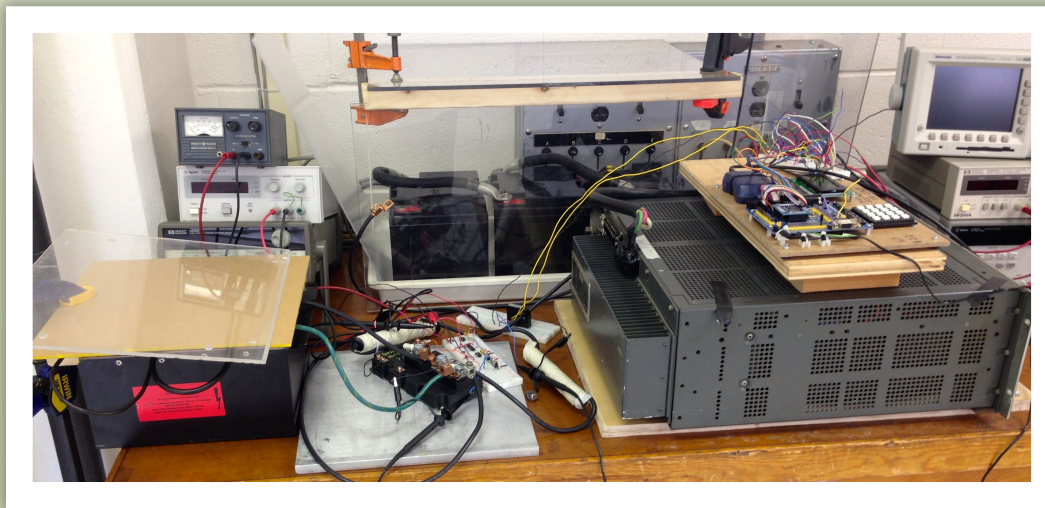


Table of Contents

I. INTRODUCTION	3
II. HIGH LEVEL SYSTEM BLOCK DIAGRAM	3
III. FUNCTIONAL REQUIREMENTS	4
A. MOTOR/GENERATOR	4
B. POWER ELECTRONICS.....	4
C. BATTERY BANKS.....	4
D. CONTROLLER.....	4
IV. DESIGN & SIMULATIONS	4
V. POWER ELECTRONICS	8
VI. CONTROLLER	17
A. CONTROLLER DESIGN	17
1. <i>Original Flow Chart</i>	17
2. <i>Controller Operation</i>	17
3. <i>Final Flow Chart</i>	22
<i>Figure 33: Final Controller Flow Chart</i>	22
VII. EQUIPMENT LIST	23
VIII. REFERENCES	23
IX. INDEX	23

I. Introduction

The need to create a more efficient form of transportation while also considering the ecosystem has led to the development and use of electric vehicles. The primary goal of the “Variable Input Constant Output Voltage Regulator” (VICOVR) project is to redesign the power electronics developed by a 2012-13 capstone project group to significantly increase output current capacity.

A secondary goal will be to design a simple closed loop control system capable of maintaining a constant output current over a wide range of input voltages. Although not included in the goals for this senior capstone project, the VICOVR is ultimately intended to provide a controlled DC current to charge the battery bank in an electric vehicle during regenerative braking.

II. High Level System Block Diagram

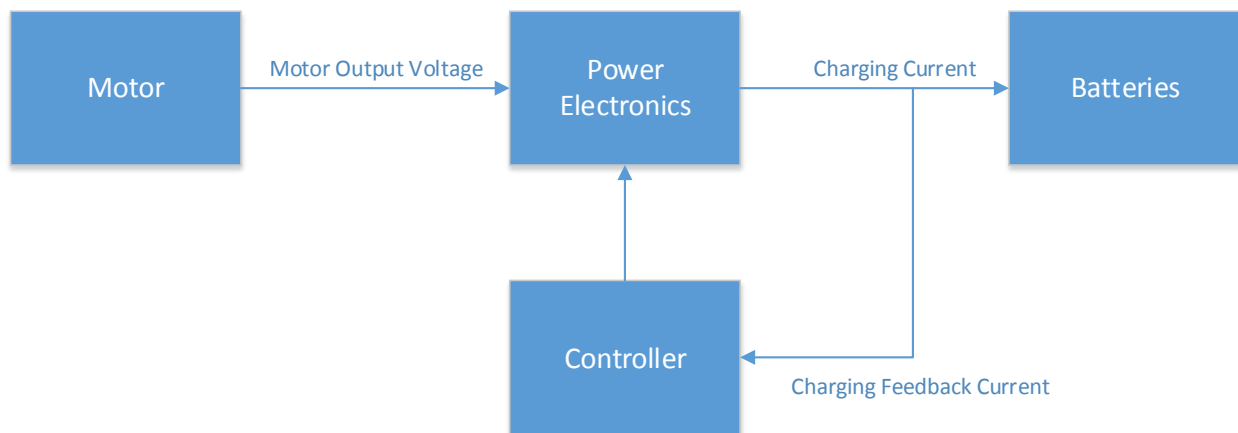


Figure 1: High level block diagram

Figure 1 represents a high-level block diagram of the Variable Input Constant Output Voltage Regulator. The power electronics will convert the widely varying generator (or source) output voltage to a voltage and current range capable of recharging the battery as the available power decreases with input voltage to the converter. The controller will change the duty cycle of a transistor switch in the power electronics (depending on the measured output current) to maintain a charging current during the braking cycle of up to about 48 ampere.

III. Functional requirements

A. Motor/Generator

- The motor/generator output voltage will vary from 5 to 35 volts.
- The motor/generator armature current will range from 20 amps to approximately 160 amps.
- The motor/generator armature resistance of 0.04 ohms.

B. Power Electronics

The power electronics will convert the varying DC input current from the motor/generator to an output current of up to 48 ampere to the battery banks. Each battery bank has a current maximum of 12 amps. The design is for 48 amps because the energy of the car at 30 mph was calculated and the amount recoverable in a 10 second stop with one battery bank was approximately one fourth of the energy required. Hence, the decision to use the four-battery bank assumption.

The current output goals are as follows:

- 10 Amp output at 10 Volts input
- 36 Amp output at 20 Volts input
- 48 Amp output at 25 Volts input
- 48 Amp output at 30 Volts input
- 48 Amp output at 35 Volts input

C. Battery Banks

There will be four battery banks that contain 3 batteries in each bank. Each battery has a maximum current of 12 amps and is 12 volts. The four banks will handle a maximum of 48 amps and 36 volts total.

D. Controller

The controller will measure the current output of the power electronics and adjust the duty cycle of the switch in the power electronics to optimize the current provides to the batteries.

IV. Design & Simulations

To capture an adequate amount of the available energy, the design will be for 4 battery banks of 3 batteries per bank. The batteries have a maximum continuous charge current rating of 12 ampere. With 4 battery banks, that allows for 48 ampere (or about 50 ampere intermittently) and a voltage of 36 volts as stated previously.



Figure 2. The SLA1161 lead acid battery that will make up the battery banks

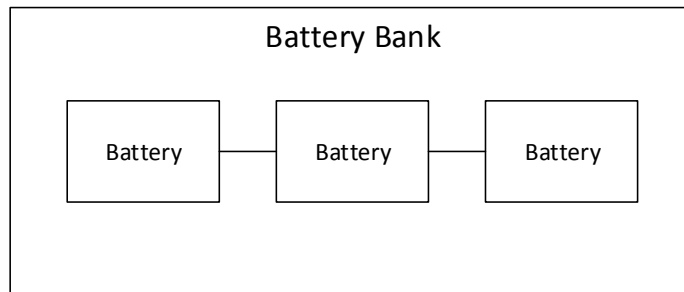


Figure 3. The following shows one battery bank, with each battery bank containing three batteries in series.

The power electronics will convert the widely varying generator (or source) output voltage to a voltage and current range capable of recharging the battery as the available power decreases with input voltage. To accomplish this goal, the boost converter was designed to increase the output voltage as the vehicle comes to a stop. When the switch is closed the supply current flows in the closed loop, which will energize the inductor. When the switch opens, the inductor will reverse polarity and the energy stored in the inductor will then flow through the diode that is now forward biased and provides a current path to the load. When the switch closes again, the filter capacitor helps maintain the output voltage while the input voltage source supplies additional energy to the inductor in preparation for the next switching cycle.

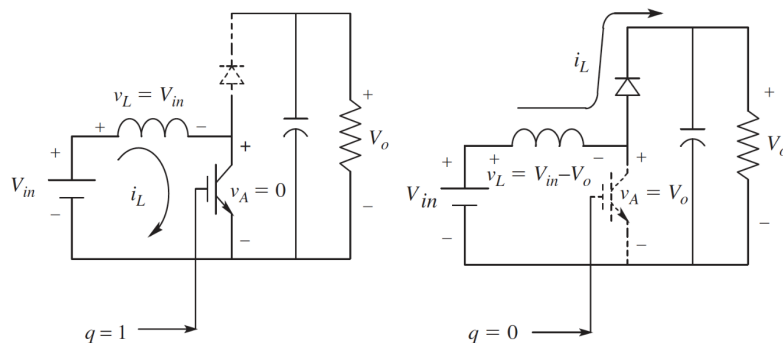


Figure 4. The boost converter to the left with $q=1$ closes the switch, and the figure to the right with $q=0$ opens the switch.

Calculations to estimate the energy available for recovery during regenerative braking while completing a stop in a 10 second interval are shown below.

- $P_{\text{motor}} = (V_c)(I_{\text{Rth}})$, where:
 - P_{motor} is the power of the motor
 - V_c is the Capacitor voltage
 - I_{Rth} is the current into the battery banks.

Calculations for a four battery bank configuration yields the following motor power and maximum available energy recovery for $V_c=36\text{Volts}$ and $I=50\text{ Ampere}$.

- $P_{\text{motor}} = (50\text{ [A]} * 36\text{ [V]}) = 1800\text{ [W]}$
- $\text{Energy} = (P_{\text{motor}})(10\text{seconds}), \text{Energy} = 18,000\text{ [J]}$.

Calculations for Go Kart Energy at 30 miles per hour:

- $E = \frac{1}{2}MV^2$, where:
 - M is the mass of the vehicle in [Kg]
 - V is the velocity of the vehicle [M/Sec]
 - E is the kinetic energy of the vehicle in motion in [J]

The speed of 30mph will be used for the kinetic energy calculations since that is the maximum speed maintainable at the rated power and speed of the motor with a maximum vehicle load of 600 [lbs]. The total available kinetic energy at 30 [mph] with a load of 600[lbs] is therefore approximately 25,000[J].

The circuit shown in figure 3 uses the median value of armature resistance (R_a) found to be approximately 0.04 [Ohms], a calculated inductor value of 50 [uH] for the input and one to serve as a filter for the output, and finally a capacitor value of 2.4 [mF]. The time constant of the battery formed by the battery resistance, R_B , and the filter capacitor needed to be much greater than that of the switching frequency. For that reason it was decided to use a value that was 10 times the value found. Using the following Equations, the values shown in Figure 3 were computed. The results of the Simulation of the circuit shown in figure 3 are shown in figure 4. It can be seen from figure 4 that the steady state current through R_B (the Thevenin equivalent battery resistance) is about 50 ampere, which corresponds to the output current goal mentioned earlier.

- L = inductor component
- E_a = back Emf
- Δi_L = Change in inductor current
- D = Duty cycle
- T_s = Switching time
- C = capacitor component
- V_c = output voltage in the capacitor
- R_{th} = Thevenin Resistor value in the battery

The design equations used are shown below.

- $L = \frac{E_a}{\Delta i_L} DT_s$
- $C = 10 \frac{T_s}{R_{Th}}$
- $\frac{V_c}{E_a} = \frac{1}{1-D} , (V_c > E_a)$

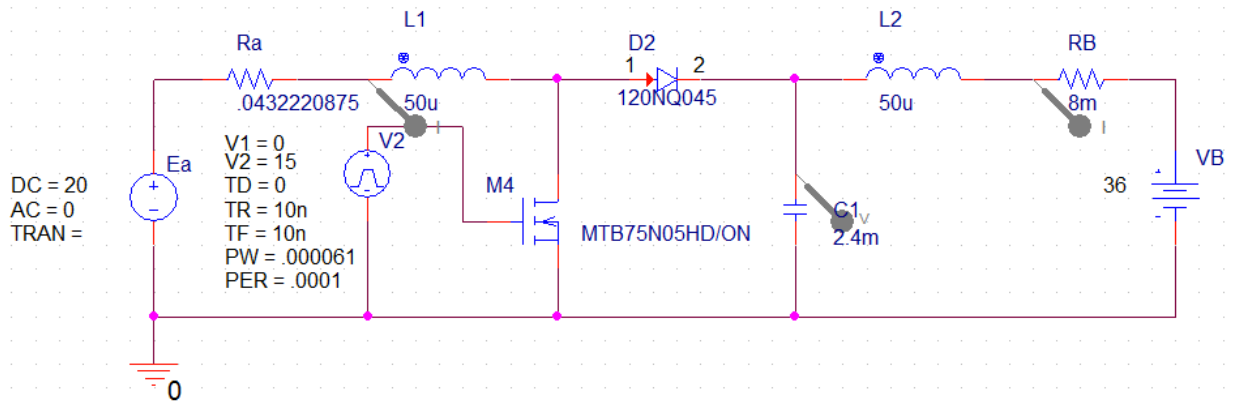


Figure 5: Boost Converter simulation using Orcad Pspice

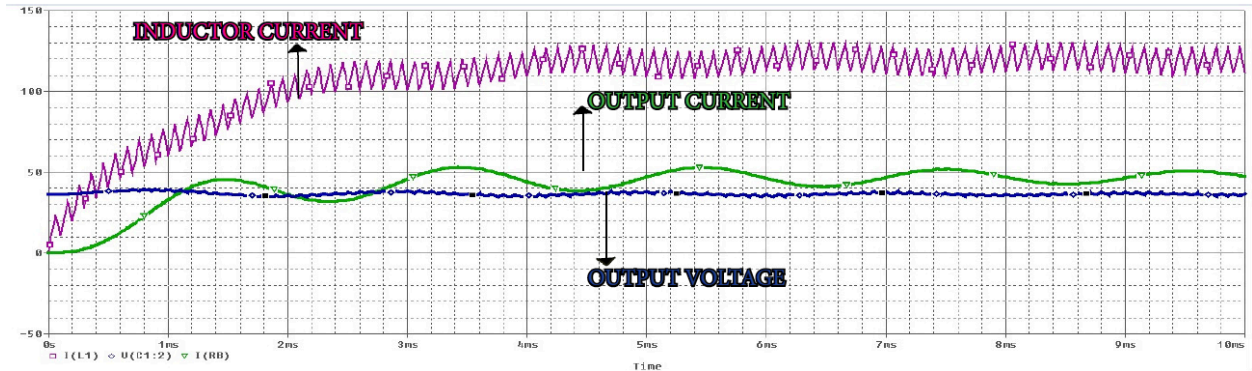


Figure 6: Boost Converter Simulation output for Figure 5

- $V_{in} = 10\text{v}$: $I_{Rb} = 16.2 \text{ [A]}$, $i_{L_avg} = 93.4 \text{ [A]}$, $i_{L_peak} = 101.1 \text{ [A]}$, $V_o = 36.6 \text{ [v]}$ at 85% Duty Cycle
- $V_{in} = 15\text{v}$: $I_{Rb} = 36.2 \text{ [A]}$, $i_{L_avg} = 120.8 \text{ [A]}$, $i_{L_peak} = 131.5 \text{ [A]}$, $V_o = 36.4 \text{ [v]}$ at 75% Duty Cycle
- $V_{in} = 20\text{v}$: $I_{Rb} = 49.3 \text{ [A]}$, $i_{L_avg} = 119.6 \text{ [A]}$, $i_{L_peak} = 133.2 \text{ [A]}$, $V_o = 36.7 \text{ [v]}$ at 61% Duty Cycle
- $V_{in} = 25\text{v}$: $I_{Rb} = 51.1 \text{ [A]}$, $i_{L_avg} = 86.3 \text{ [A]}$, $i_{L_peak} = 101.9 \text{ [A]}$, $V_o = 36.8 \text{ [v]}$ at 43% Duty Cycle
- $V_{in} = 30\text{v}$: $I_{Rb} = 50.5 \text{ [A]}$, $i_{L_avg} = 67.8 \text{ [A]}$, $i_{L_peak} = 81.1 \text{ [A]}$, $V_o = 36.6 \text{ [v]}$ at 27% Duty Cycle
- $V_{in} = 35\text{v}$: $I_{Rb} = 51.7 \text{ [A]}$, $i_{L_avg} = 58.6 \text{ [A]}$, $i_{L_peak} = 65.2 \text{ [A]}$, $V_o = 36.4 \text{ [v]}$ at 12% Duty Cycle

Figure 7: Simulations Results at different input values

The controller will function as indicated in the flow chart shown in figure 5 below. The controller will only be active while the braking system is activated. During braking, it will read the current sensor and determine the level of the battery current compared to the desired current level specified for this project. It will adjust the duty cycle as necessary to maintain the maximum possible charging current, and then repeat the process as the charging current level decreases as the braking process reduces the motor speed.

V. Power Electronics

For this project, the design team decided to build an inductor. To that end, a solid core ferrite rod and number 4 wire gauge were purchased. By using the inductor equation shown below and manipulating it algebraically it was possible to solve for the number of turns need to provide the desired inductor value. Before building the inductor, it was necessary to obtain the value for the permeability of the ferrite core rod to determine the total magnetic field the core could support. The following shows the equations used to design and build system inductors.

$$\text{Area} = \pi r^2 = 3.1415 * (.0047625\text{m})^2 = 7.125573925\text{E-}5 \text{ m}^2$$

- $\mu = \mu_r * \mu_0 = 125 * 4\pi\text{E-}7 = 1.570796327\text{E-}4$
- $L = \mu N^2 A / l \Rightarrow N = ((l * L) / (\mu * A))^{1/2}$
- For 100 $\mu\text{H} = 41.25509832$ turns
- For 50 $\mu\text{H} = 29.17175978$ turns
- For 30 $\mu\text{H} = 22.59634796$ turns

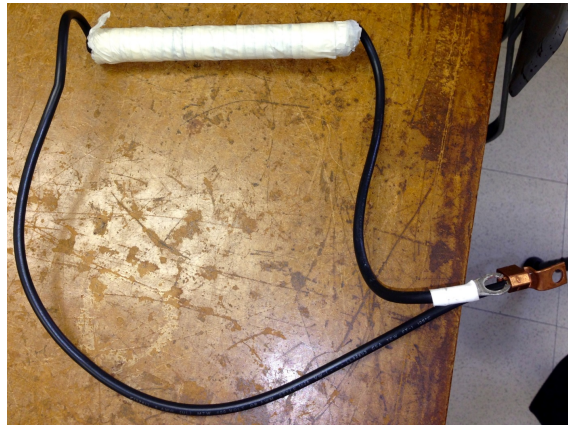


Figure 8. One of two identical wound inductors built for the VICOVR project.

It was also decided to design a gate drive module to control the switching of the IGBT. The gate drive module consists of two HCPL-3120 optically isolated gate drivers, 2 N-Channel power transistors, a DM74LS05 open-collector hex inverter, a bootstrap capacitor, a diode, and various resistors. One HCPL-3120 was used to control the turn-on of the IGBT, and the second to turn-off the IGBT. Since a voltage of 15 [V] was selected for V_{DD} , the high side needs a voltage of 30 [V] to turn-on the IGBT, and for the low side a voltage of 15 [V] was needed to turn-off the IGBT. The circuit design is shown in the following figures, including the finalized gate drive module and the results of the circuit tests.

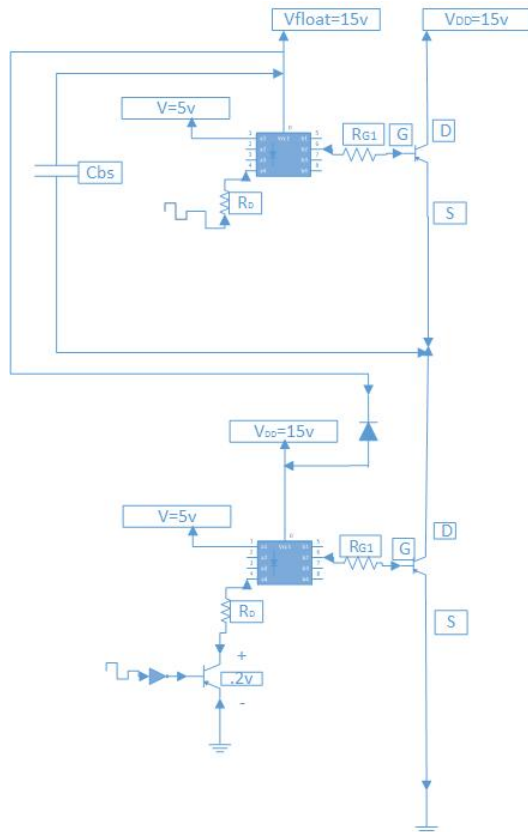


Figure 9. The gate drive module used to turn on and off the IGBT.

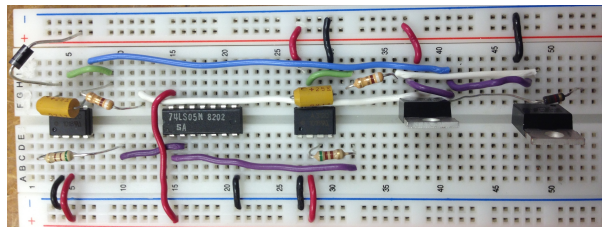


Figure 10. The finalized gate drive module used to turn on and off the IGBT.

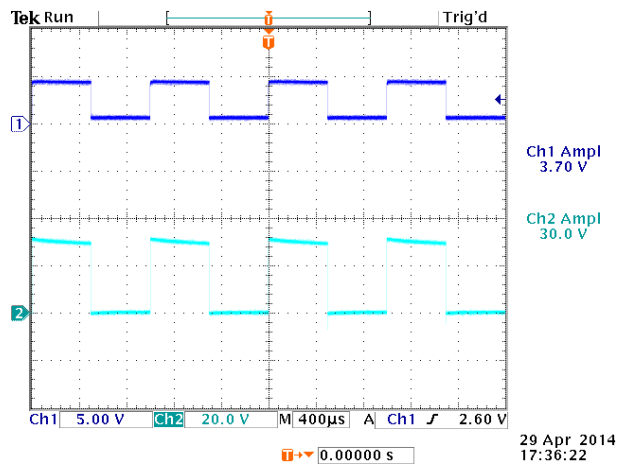


Figure 11. The high side output of the gate drive module used to turn on the IGBT.

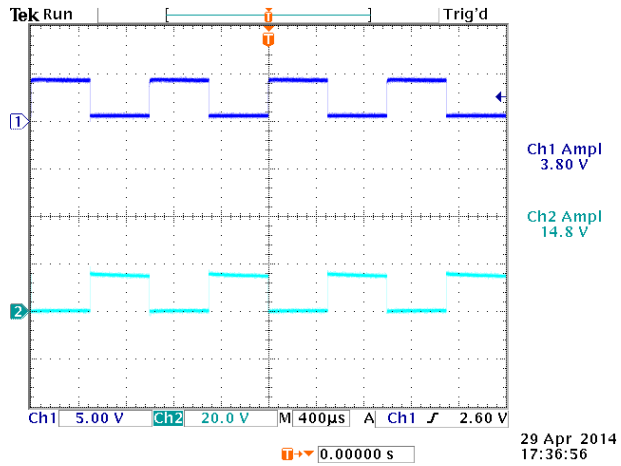


Figure 12. The low side of the gate drive module that will turn off the IGBT.

Once the gate drive module was operational, the gate drive module was then merged with the boost converter. Figure 13 shows the complete circuit with the gate drive module and the boost converter merged. The circuit was then tested with a 15 [V] input and a duty cycle of 50 % that gave us an output of 31.4 [V]. Theoretically an output of 30 [V] should be obtained by the following equation:

$$V_o = \frac{V_{in}}{(1 - D)}$$

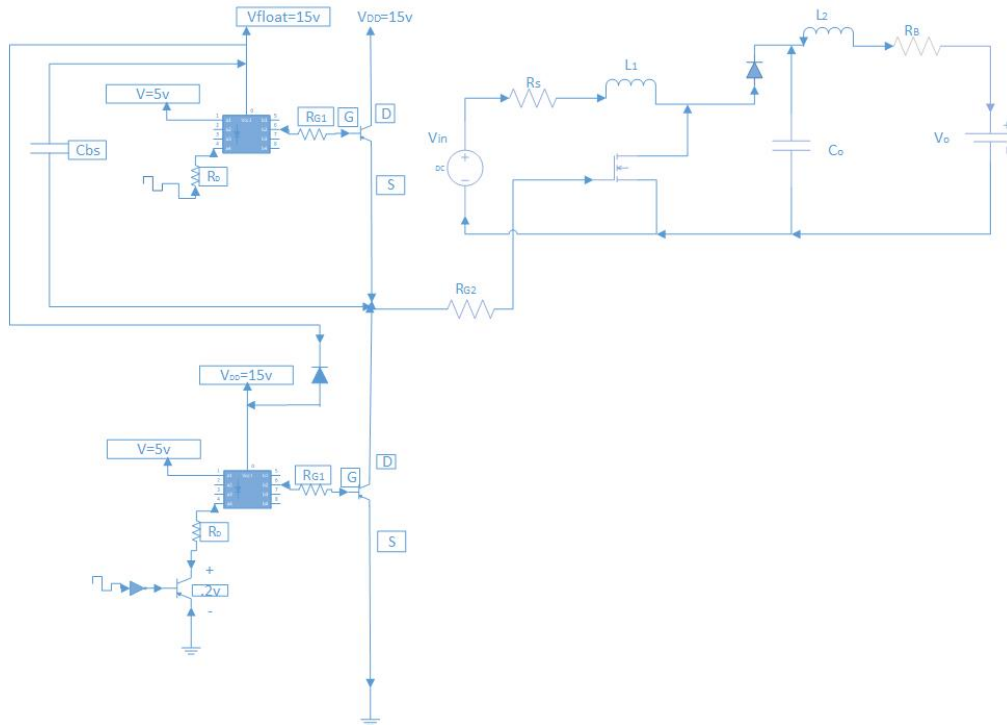


Figure 13. The gate drive module and the boost converter merged.

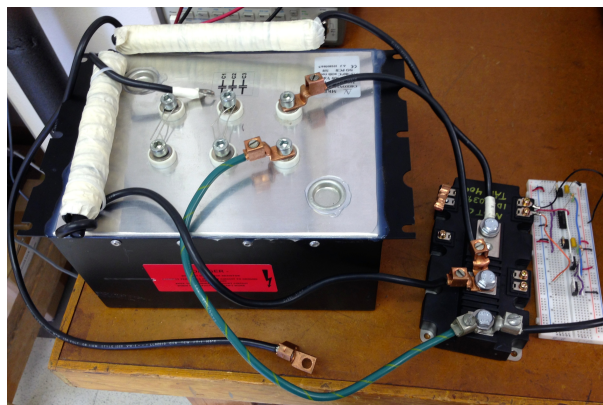


Figure 14. The physical gate drive module and the boost converter merged.

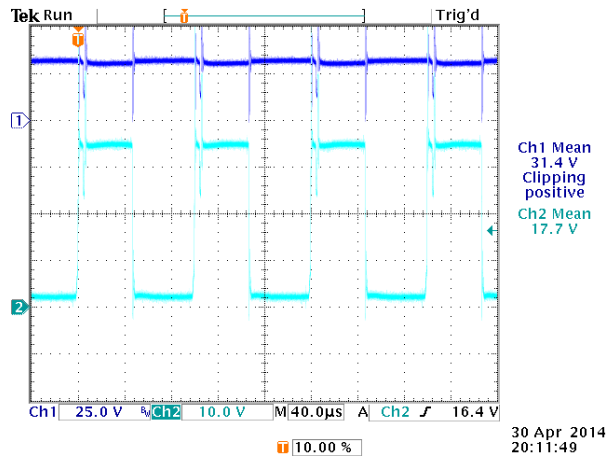


Figure 15. The results of the boost converter:

Channel 1 → Battery charging voltage

Channel 2 → Switching transistor V_{CE} voltage

A major concern was that the output voltage could increase to immense values if the batteries used for the load were to be disconnected or failed internally (i.e., open – circuited). For that reason a protection circuit was designed which would turn off the IGBT if the capacitor voltage were to exceed a specified maximum value. The protection circuit was designed for a 40 [V] threshold using a voltage divider which would provide a 5 [V] output and also a reference voltage of 5 [V]. When the output of the comparator would be less than 40 [V], a value of logic ‘1’ would be applied to the AND gate. Conversely, a logic ‘0’ would be applied to the AND gate if the input was greater than 40 [V]. The AND gate would have one input consisting of the PWM which will alternate between a value of logic ‘1’ and ‘0’, and the second input would be the output of the comparator. The output of the AND gate would then be applied to the gate drive module, thus turning off the IGBT if the over-voltage threshold is exceeded. The over-voltage protection circuit was designed using a single LM311 comparator, a DM74LS08 quad 2 input AND gate, and various resistors. First a resistor value for R1 of 10 [KΩ] was selected, and then the corresponding value for R2 was solved for by manipulating algebraically the voltage divider formula shown below.

$$V_o = \frac{R_2}{R_2 + R_1} V_{in}, \text{ Therefore } R_2 = \frac{(V_o)(R_1)}{(V_{in} - V_o)}$$

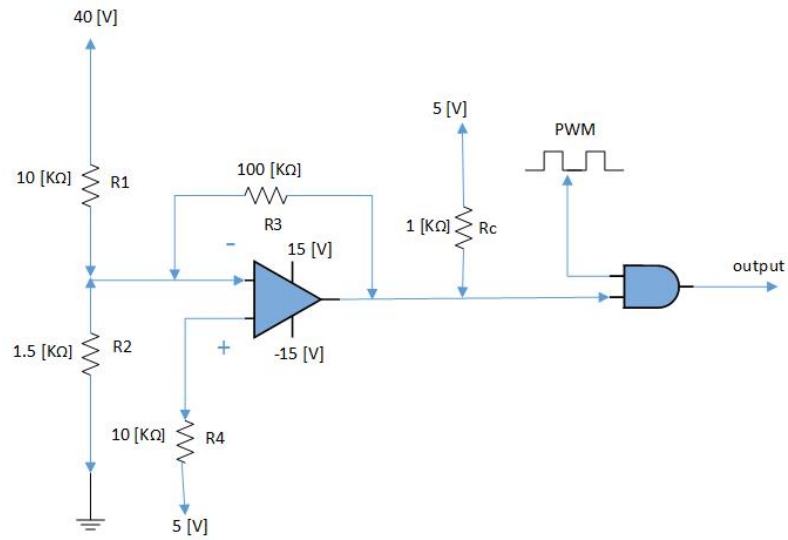


Figure 16. The protection circuit design.

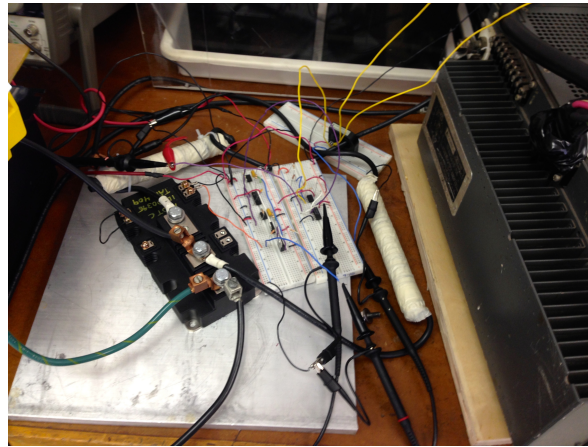


Figure 17. The protection circuit implemented with the power electronics.

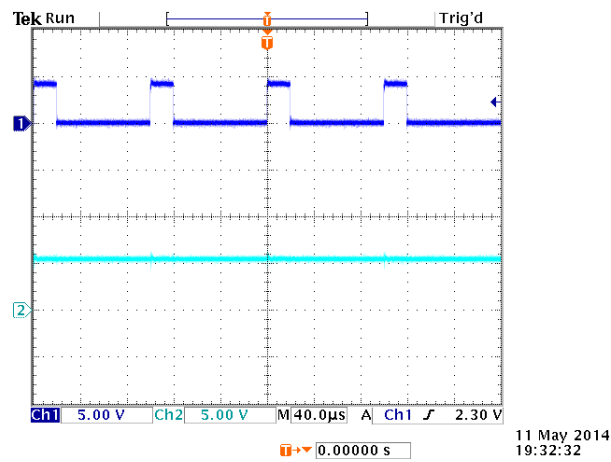


Figure 18. The signal from the protection circuit while less than 40 [V].

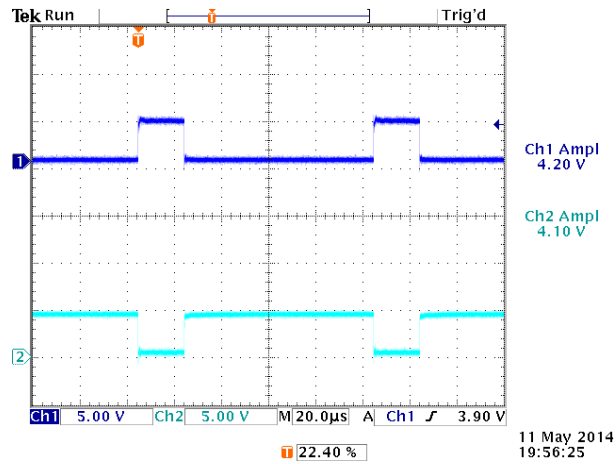


Figure 19. The gate drive module outputs while protection circuit for output voltages less than 40 [V].

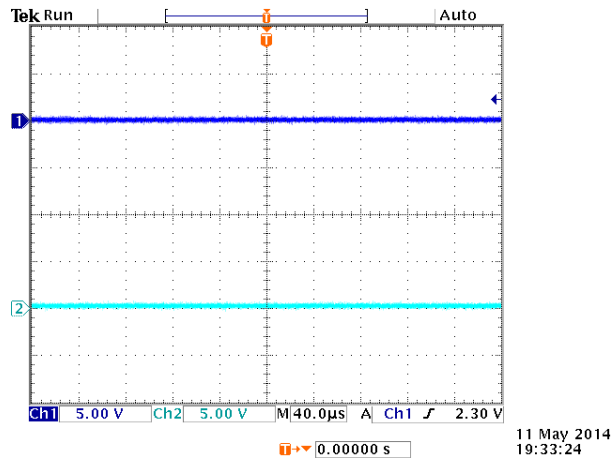


Figure 20. The signal from the protection circuit for output voltages greater than 40 [V].

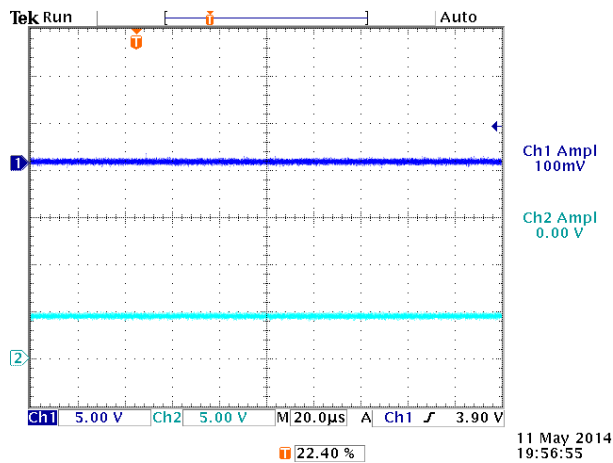


Figure 21. The signal from the gate drive module for protection circuit output voltages greater than 40 [V].

Due to time constraints, it was not possible to test the system over the full range of output currents since a 4-battery bank was not available. An oscilloscope probe was connected to the source inductor to monitor the converter input voltage. The second oscilloscope probe was connected to the gate to see the switching of the IGBT. The inductor voltage reversed polarity and reached the battery voltage, but eventually it could no longer maintain the necessary voltage to keep the diode forward biased. This effect can be seen via the oscillations on channel one from **figure 23**. The oscillations are due to the inductor current flowing into the effective capacitance seen at the collector of the IGBT. The oscillations stop once the IGBT is turned on again.

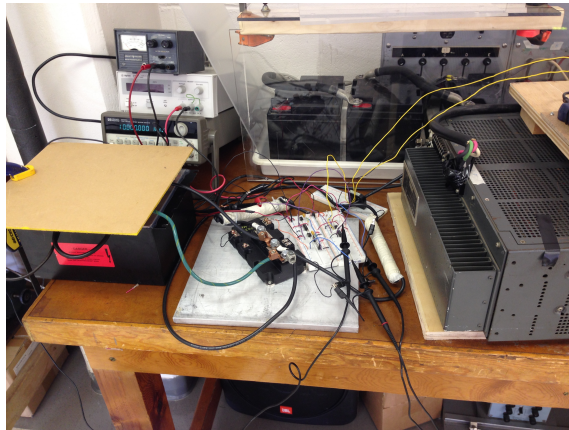


Figure 22. The overall power electronics implemented with all added circuits.

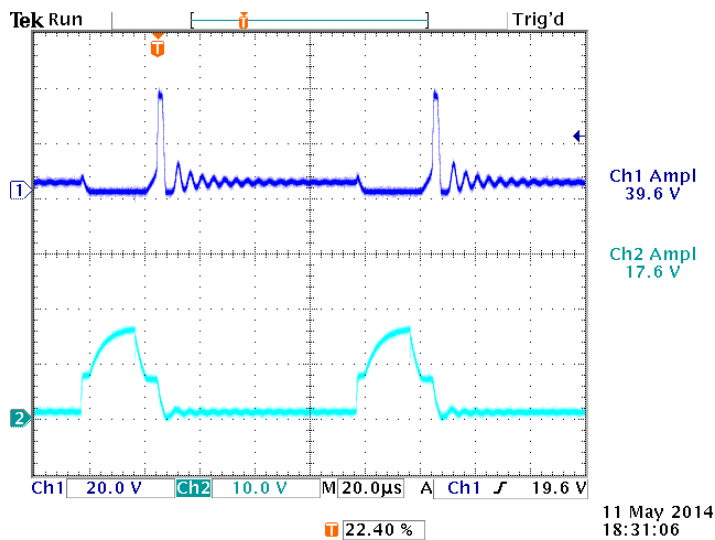


Figure 23. The voltage at the IGBT collector and gate of the switching transistor.

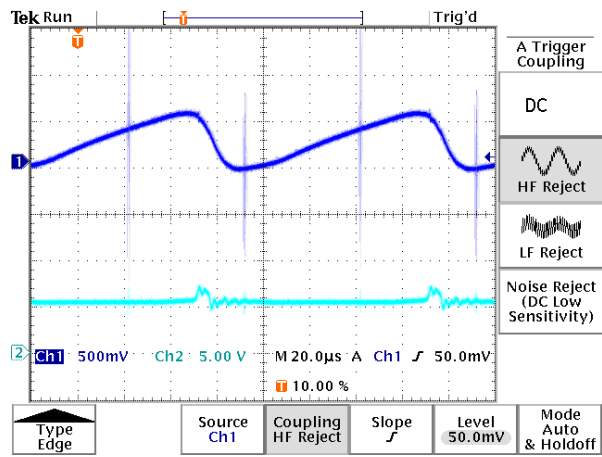


Figure 24. The inductor current and the protection circuit voltage.

VI. Controller

A. Controller Design

1. Original Flow Chart

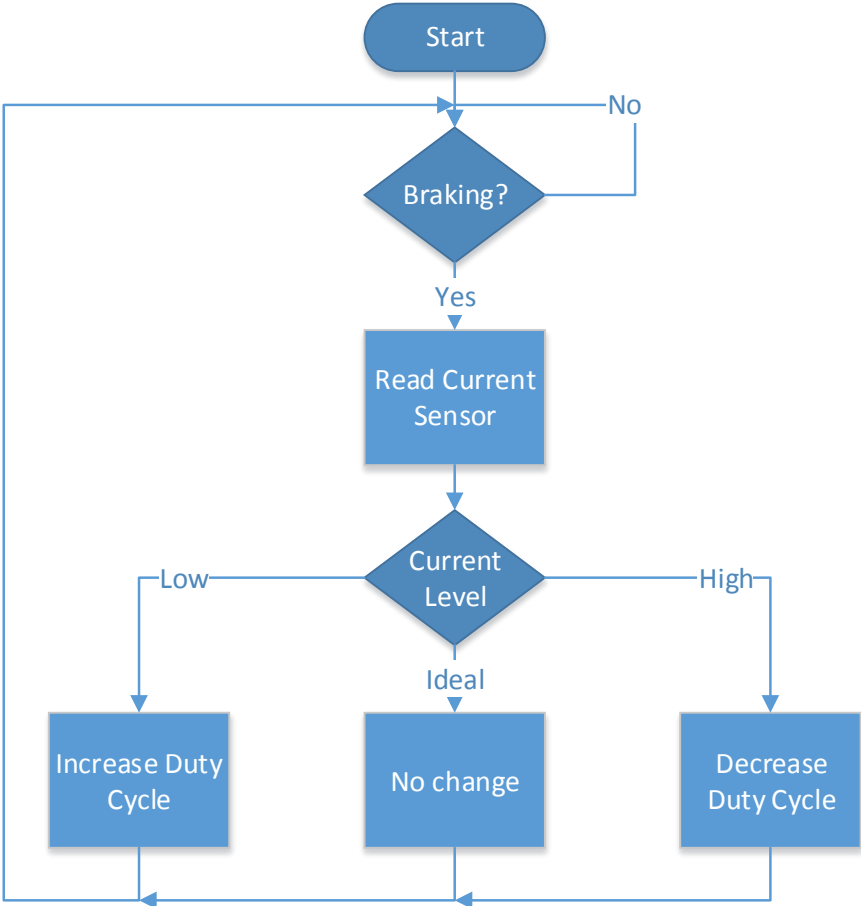


Figure 25: Original Flow Chart for controller

2. Controller Operation

The controller is intended to function as shown in the original flow chart presented in figure 25. The intended application of this system is ultimately a regenerative braking subsystem, so the first test is to determine if the vehicle is in the act of braking. For the purposes of this project, during testing it is assumed that the system is always in the act of braking. The controller will then read in the data from the current sensor, determine if the output current is at the desired level, and adjust the duty cycle accordingly. The duty cycle is used to adjust the switch operation in the power electronics. The flow chart in figure 25 has shortcomings that will be discussed in the final flow chart section.

The system required a functional PWM to control the transistor switch, and the PWM capability of the ATmega128 microcontroller was used for that purpose. The oscilloscope plot in figure 26 demonstrates that the PWM system is functional.

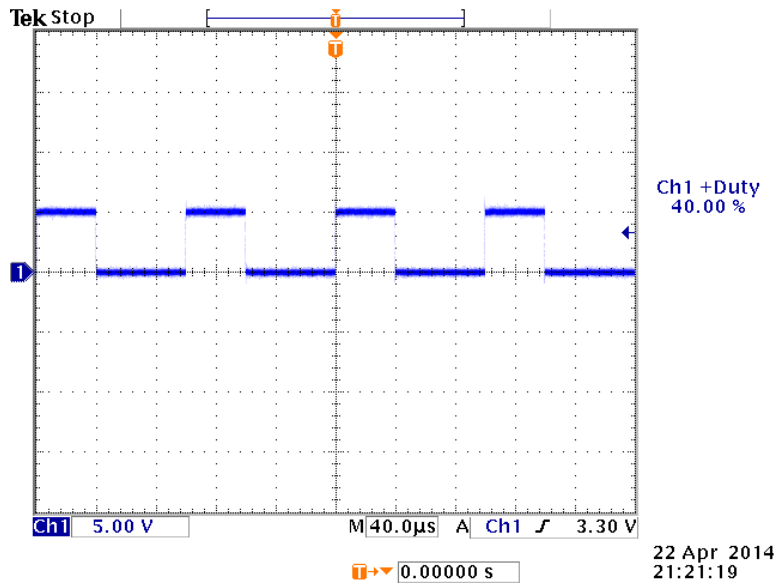


Figure 26: PWM with 40% Duty Cycle

To obtain usable data from the current sensor, an analog-to-digital converter (ADC) on the ATmega128 was used. To test the ADC, an external digital-to-analog converter (DAC) was constructed to verify the operation of the ADC. The DAC operation was verified by incrementing the value sent through PORTC of the ATmega128 (the port used to connect to the DAC) which should display a ramp function when connected to the oscilloscope. The circuitry used for the DAC is shown in figure 27, and the successful output of the DAC is presented in figure 28. The DAC and the op amp are powered by ± 15 V.

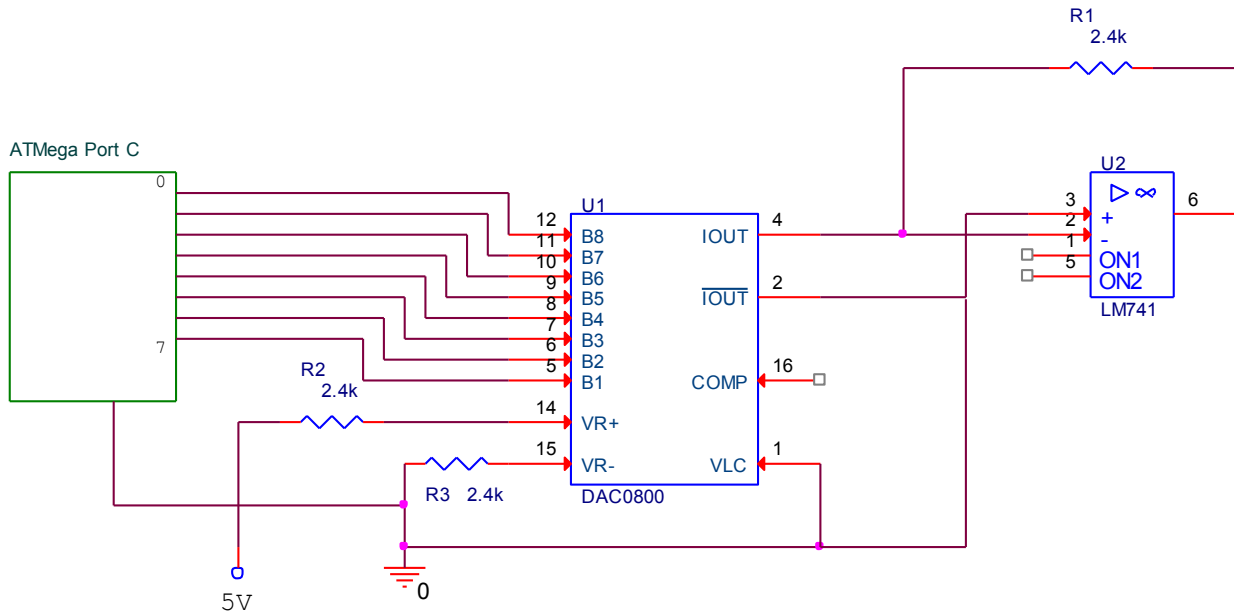


Figure 27: External DAC circuitry

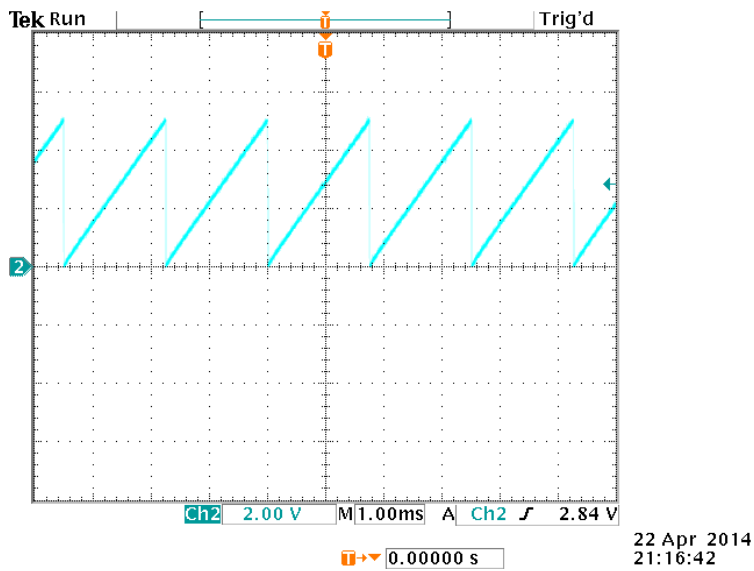


Figure 28: External DAC test

Once the operation of the DAC was verified it could be used to confirm the operation of the ADC. Figure 29 shows the input from the function generator (channel 1) and the output of the DAC (channel 2) after the input signal was processed by the ADC.

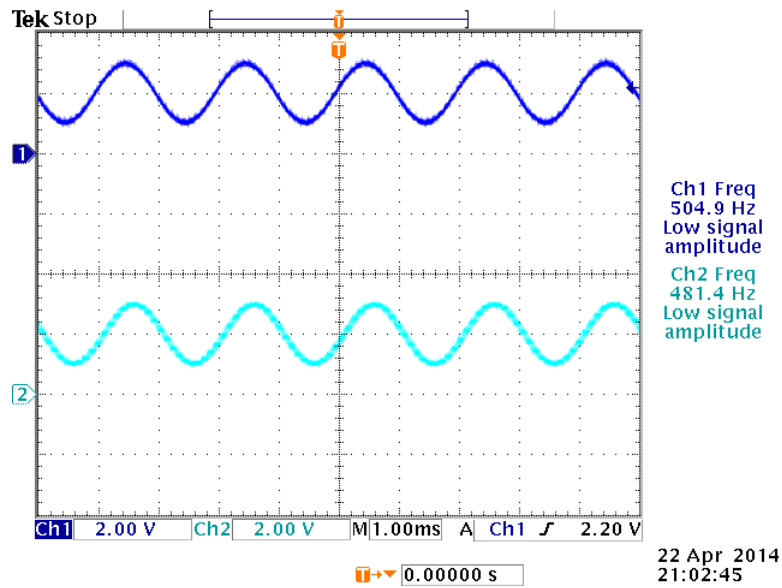


Figure 29: ADC Input and DAC Output

Once it was confirmed that the ADC and DAC were operational, the Hall Effect current sensor was connected to the ADC and DAC.

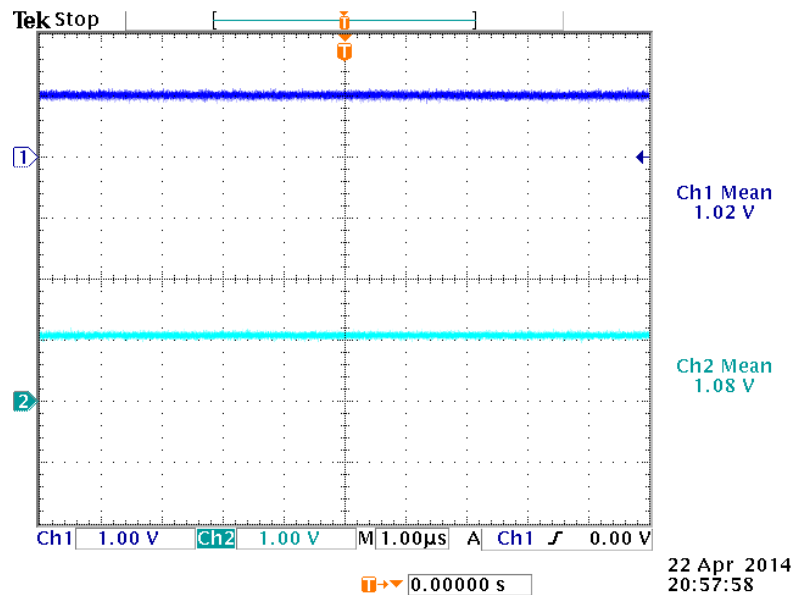


Figure 30: Output of Current Sensor (Channel 1) and Output of DAC (Channel 2)

For the original testing to confirm a changing duty cycle based on the input of the current sensor, a maximum of 85% duty cycle and a minimum of 5% duty cycle were programmed into the code. The threshold for increasing or decreasing the duty cycle is also not at the target of 48 amps for this particular test because it is not necessary to test at high currents for proof of this concept.

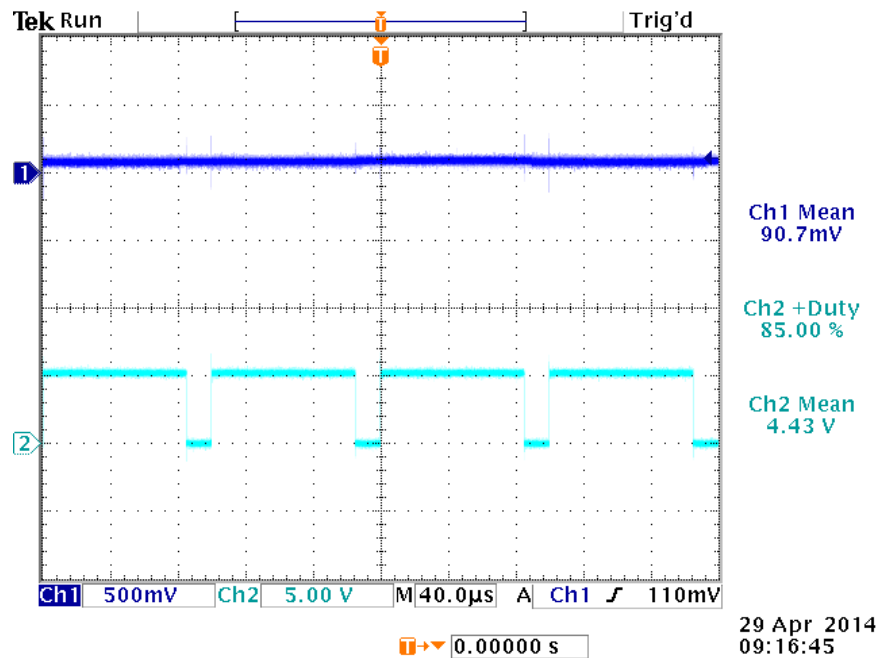


Figure 31: Maximum Duty Cycle (For testing purposes)

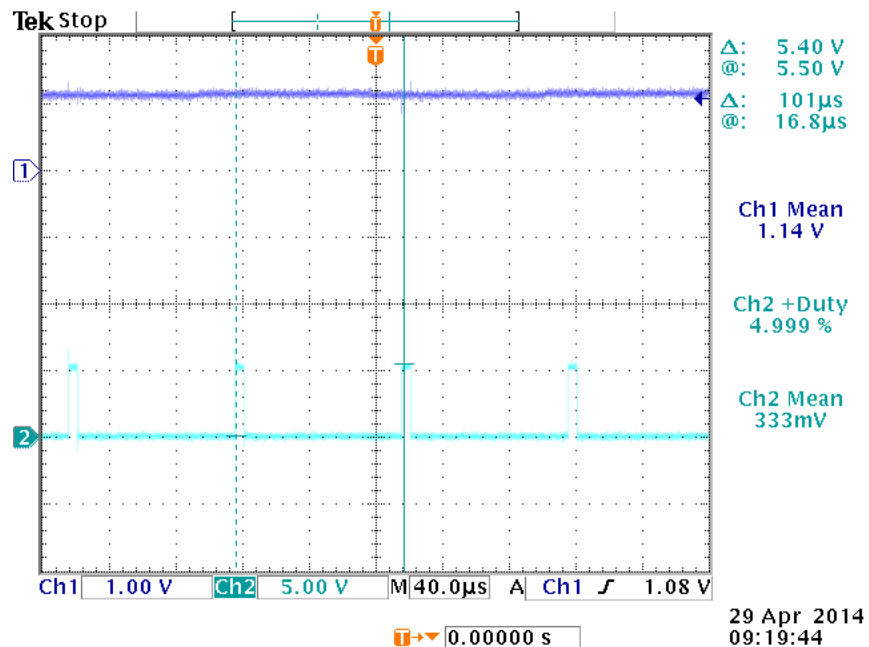


Figure 32: Minimum Duty Cycle

It can be seen from figures 31 and 32 that when the input (Channel 1) from the current sensor is low, the duty cycle is at its maximum and when the input is high (over the threshold set in code for testing) the duty cycle is at its minimum value. It can also be seen in figure 32 that the frequency of the PWM is approximately 10 kHz.

The problem with the current controller setup described is that the 85% and 5% duty cycle boundaries will change during normal system operation. Because of this, another decision point was added to the flow chart as shown in figure 33 below.

3. Final Flow Chart

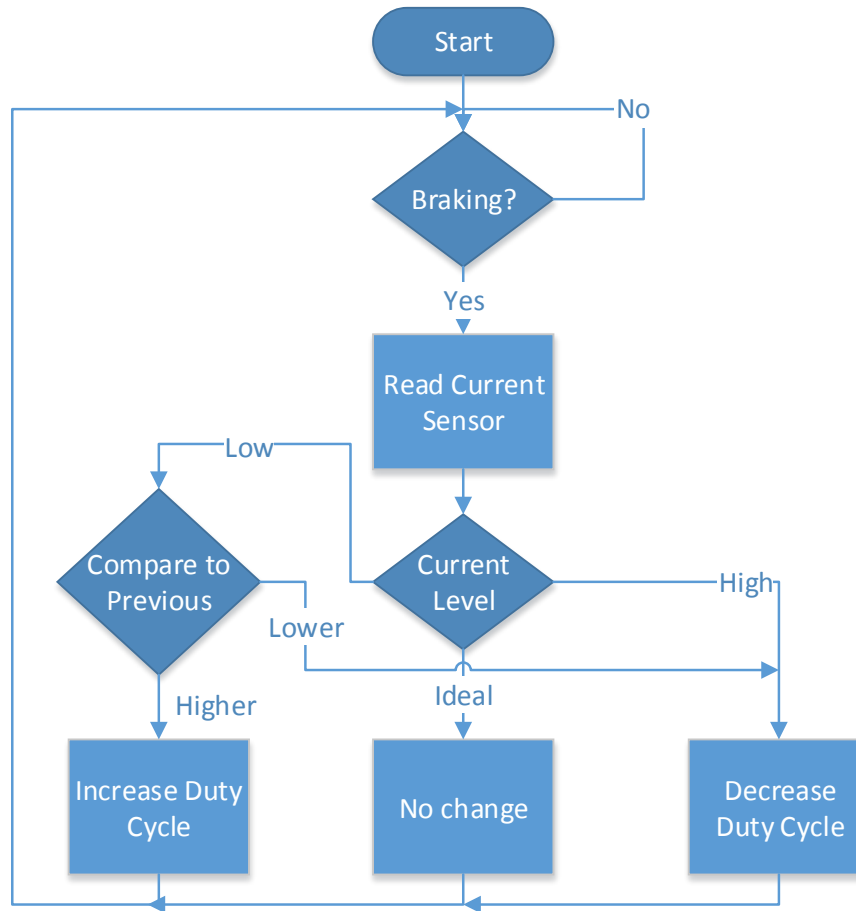


Figure 33: Final Controller Flow Chart

The change in this flow chart is that another check has been added. Rather than setting a minimum and maximum duty cycle, the previous current reading is stored. If the current level is lower than desired but is higher than the last current reading (going in the desired direction), then the duty cycle is increased. However, if the current is both lower than desired and has decreased since the last measurement, the duty cycle is decreased. This is because increasing the duty cycle during the previous iteration of this flow chart has actually decreased the current, meaning that the switch is spending so much time closed that it is not open long enough to send as much charge current to the battery banks as possible (while keeping in mind we do have a maximum continuous charge current limit).

VII. Equipment List

IGBT – FF900R12IP4D
Control Board – ATMega128
Motor – D&D ES-10E-33
Batteries – SLA1161
Current Sensor – L08P050D15
Diode – 120NQ045
Inductor – 50 microHenry
Capacitor – 2.4 milliFarads

VIII. References

- [1] Jaris, Kevin, and Nathan Golick. "Micro Electric Urban Vehicle - PHASE III." Electrical and Computer Engineering Department Bradley University, n.d. Web. 2013.
- [2] Kuhn, Brian, Steven Komperda, and Matthew Leuschke. "Micro Electric Urban Vehicle - PHASE II." Electrical and Computer Engineering Department Bradley University, n.d. Web.
- [3] Mrsolar.com. Interstatebatteries.com, n.d. Web. 2013.
<[http://www.mrsolar.com/pdf/Interstate/Interstate SLA1161.pdf](http://www.mrsolar.com/pdf/Interstate/Interstate%20SLA1161.pdf)>.
- [4] "D&D ES-10E-33 Motor." *Www.evdrives.com*. N.p., n.d. Web. 2013.
<http://www.evdrives.com/product_p/mot-es-10e-33.htm>.

IX. Index

Controller C Code:

```
/*  
 * SeniorProject.c  
 *  
 * Created: 3/4/2014 3:49:54 PM  
 * Author: Alex  
 */  
#include <stdio.h>  
#include <avr/io.h>  
#include <stdint.h>  
#include <avr/interrupt.h>  
#include <util/delay.h>  
#include "lcd.h"  
//#include <stdlib.h>  
//#include <string.h>  
  
char duty_cycle=0, str[5];  
char tens, ones, current=0, tenths;  
unsigned char last_adc;
```

```

void PWM_init()    {

    PORTB=0x00;
    DDRB=0xE0;

    // Timer/Counter 1 initialization
    // Clock source: System Clock
    // Clock value: 2000.000 kHz
    // Mode: Ph. & fr. cor. PWM top=ICR1
    // OC1A output: Connected
    // OC1B output: Connected
    // OC1C output: Connected
    // Noise Canceler: Off
    // Input Capture on Falling Edge
    // Timer1 Overflow Interrupt: Off
    // Input Capture Interrupt: Off
    // Compare A Match Interrupt: Off
    // Compare B Match Interrupt: Off
    // Compare C Match Interrupt: Off
    TCCR1A=0xA8;
    TCCR1B=0x12;
    TCNT1H=0x00;
    TCNT1L=0x00;

    sei();
    ICR1 = 100;    //10 kHz with N=8 (prescaler)
    OCR1A = 50; //% duty cycle
}

void ADC_init(void)  {

    //free running mode
    //8 bit precision
    //interrupt enabled
    //VCC as voltage reference (5V)
    //ADC enable, free running mode - ADEN=1, ADFR=1
    //ADC interrupt enable - Bit3
    //Clock Prescaler = 8(last 3 bits of ADCSRA)
    //ADCSRA = 0xAB;
    //SREG |= 0x80;    //SREG I-bit (required for interrupt)
    //ADMUX = 0x80;    //Voltage reference, left or right adjust, channel, and gain select
    //
    //DDRF = 0x00;    //configure for input
    //PORTF = 0x00;    //turn off pull up resistors
    //ADCSRA |= 0x80;    //enable
    //ADCSRA |=0x40;    //start first conversion
    //

```



```

//code from web -
http://www.avrfreaks.net/index.php?name=PNphpBB2&file=printview&t=56429
//DDRE |= (1 << 2); // Set LED1 as output
//DDRG |= (1 << 0); // Set LED2 as output
ADCSRA |= (1 << ADPS2) | (1 << ADPS1) | (1 << ADPS0); // Set ADC prescalar to
128 - 125KHz sample rate @ 16MHz

```

```

ADMUX |= (1 << REFS0); // Set ADC reference to AVCC
ADMUX |= (1 << ADLAR); // Left adjust ADC result to allow easy 8 bit reading
// No MUX values needed to be changed to use ADC0

```

```

ADCSRA |= (1 << ADFR); // Set ADC to Free-Running Mode
ADCSRA |= (1 << ADEN); // Enable ADC
ADCSRA |= (1 << ADSC); // Start A2D Conversions
ADCSRA |= (1 << ADIE);
sei();

```

```

}

```

```

int main(void)

```

```

{

```

```

//test_lcd();
lcd_init(LCD_DISP_ON);
lcd_gotoxy(5,0);
lcd_puts("VICOVER");
lcd_gotoxy(0,1);
lcd_puts("Duty Cycle = %");
lcd_gotoxy(-4,2);
lcd_puts("Current = . A");
//_delay_ms(50000);
//lcd_puts("New DC: ");
PWM_init();
ADC_init();
OCR1A = 40;
DDRC = 0xFF; //Set portC for output

```

```

while(1)

```

```

{

```

```

//TODO:: Please write your application code
//Test
//PORTC=ADC;
//Test external DAC
//PORTC++; //Send ramp function to PortC to test external ADC
//_delay_ms(.1);
//if(PORTC==0xFF) PORTC=0;
//sprintf(str, "%d", OCR1A);
//
//lcd_gotoxy(13,0);
//duty_cycle= (char) OCR1A;
//duty_cycle = itoa(OCR1A, str, 2);
//lcd_data(str[5]);

```

```

        //lcd_data(duty_cycle);
        //lcd_puts("85");
        //functional code
        //lcd_gotoxy(13,0);
        //tens = (OCR1A/10);
        //ones = (OCR1A%10);
        //lcd_putc(tens+0x30);
        //lcd_putc(ones+0x30);
        //end functional test code
        //lcd_gotoxy(-4,2);
        //lcd_puts("A to start, D to stop");
        //lcd_putc(OCR1A);
        //lcd_putc(tens);
    }
}

ISR(ADC_vect) {

    //Compare value to desired current sensor output
    //adjust duty cycle for PWM to increase or decrease current
    //Interrupt bit cleared by hardware when running ISR
    //Current sensor outputs 4V at 50A
    //Max voltage read by the board is the reference voltage, in this case 5V
    //8 bit precision - 255 = 5V
    //Target of 48 amps = 3.84V, or 195
    //First test with A/D D/A
    //PORTC = ADCH;
    ///PORTC++;
    //if(ADCH <= 50/*195*/ && OCR1A < 85)    {                //tested at 50 so that ~1V
output of current sensor changes duty cycle to min or max

        //OCR1A++;    //increase duty cycle by 1%
    //}else if(ADCH > 50/*195*/ && OCR1A > 5){
        //OCR1A--;    //decrease duty cycle by 1%
    //}

    // -----Test code-----
    /*this section of code demonstrates the functionality of the adc by showing more than just
    min and max duty cycle on display for demonstration purposes*/
    if(ADCH <30 && OCR1A<=85) {
        OCR1A=85;//++;
    }
    else if (ADCH >=30 && ADCH <50)    {
        OCR1A = 70;
    }
    else if (ADCH >=50 && ADCH <70)                {// && ADCH <100)    {
        OCR1A = 55;

```

```

}

else if (ADCH >=70)  { // && ADCH <250)  {

    OCR1A = 40;

}

//else if (ADCH >=150 && ADCH <200) OCR1A = 30;

//else if (ADCH >=200 && OCR1A>5) OCR1A=5;//--;

//else {

    //OCR1A=0;

    //lcd_gotoxy(-4,2);

    //lcd_puts("Error??");

//}

//if(ADCH < 40 /*40 for 10A to reduce risk of battery damage*//*48 for 12A*/ /*195*/)
{
    //if(last_adc<ADCH && OCR1A<95)  {
        //OCR1A++;
    //}else if(ADCH>=last_adc && OCR1A>1)  {
        //OCR1A--; //if current level decreases from increase of duty cycle
decrease duty cycle
    //} //else
    ////{
        ////lcd_gotoxy(-4,2);
        ////lcd_puts("error");
    ////}
//}else if (ADCH >= 40 /*195*/ && OCR1A>1)  {
    //OCR1A--;

//}
//lcd_gotoxy(3,13);
//duty_cycle= (char) ADCH;
//lcd_data(duty_cycle);
lcd_gotoxy(13,1);
tens = OCR1A/10;
ones = OCR1A%10;
lcd_putc(tens+0x30);
//lcd_gotoxy(14,1);
lcd_putc(ones+0x30);
//Display approximate current
current = ADCH*250/1020;
tens = current/10;
ones = current%10;

```

```
tenths = current/100;
lcd_gotoxy(6,2);
lcd_putc(tens+0x30);
lcd_putc(ones+0x30);
lcd_gotoxy(9,2);
lcd_putc(tenths+0x30);
//_delay_ms(500);           //code to show display numbers
};
```