



**MIDDLE EAST TECHNICAL  
UNIVERSITY  
ELECTRICAL & ELECTRONICS  
ENGINEERING**

**EE463 TERM PROJECT FINAL REPORT**

***EMK***

STUDENT 1: ECEM KARAMERCAN & 2304863

STUDENT 2: GOKTUG TONAY & 2305506

STUDENT 3: MERT EREN KANDILLI & 2304855

## Contents

1) Introduction .....	3
2) Project Specifications .....	3
3)Topology Selection .....	4
Three Phase Controlled Rectifier.....	4
Diode Rectifier - Buck Converter .....	6
4) Simulation .....	8
Rectifier .....	8
Buck Converter.....	9
5)Gate Driver Design .....	12
6)Controller .....	19
7) Component Selection.....	23
8)Thermal Calculations.....	24
9)Testing .....	26
10)Demo Day .....	28
11)Conclusion .....	30
12)Appendices.....	31

## 1) Introduction

In this hardware project, we are asked to design a DC Motor Drive. This report presents the development stages of the hardware project. Firstly, different types of AC/DC converter topologies are discussed, using their simulation results their advantages and disadvantages are examined. After this comparison, second part explains the topology selection stage. In the following part our design and simulation results will be displayed. This report also includes the thermal calculations, gate driver design, test results and demo day results.

## 2) Project Specifications

In this project it is aimed to design a AC/DC converter that will be fed from a 3 Phase or 1 Phase AC grid and gives adjustable DC output voltage of maximum 180V. Figure 2.1 and 2.2 show the DC motor that will be driven in this project.



Figure 2.1: DC Motor

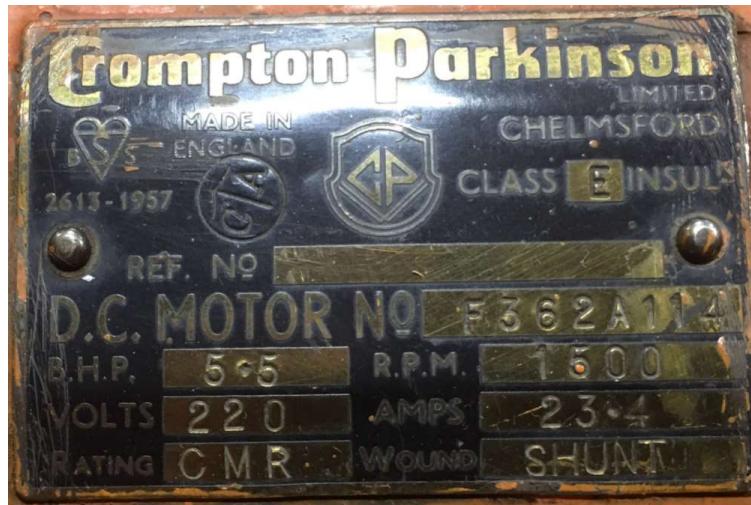


Figure 2.2: DC Motor Specs

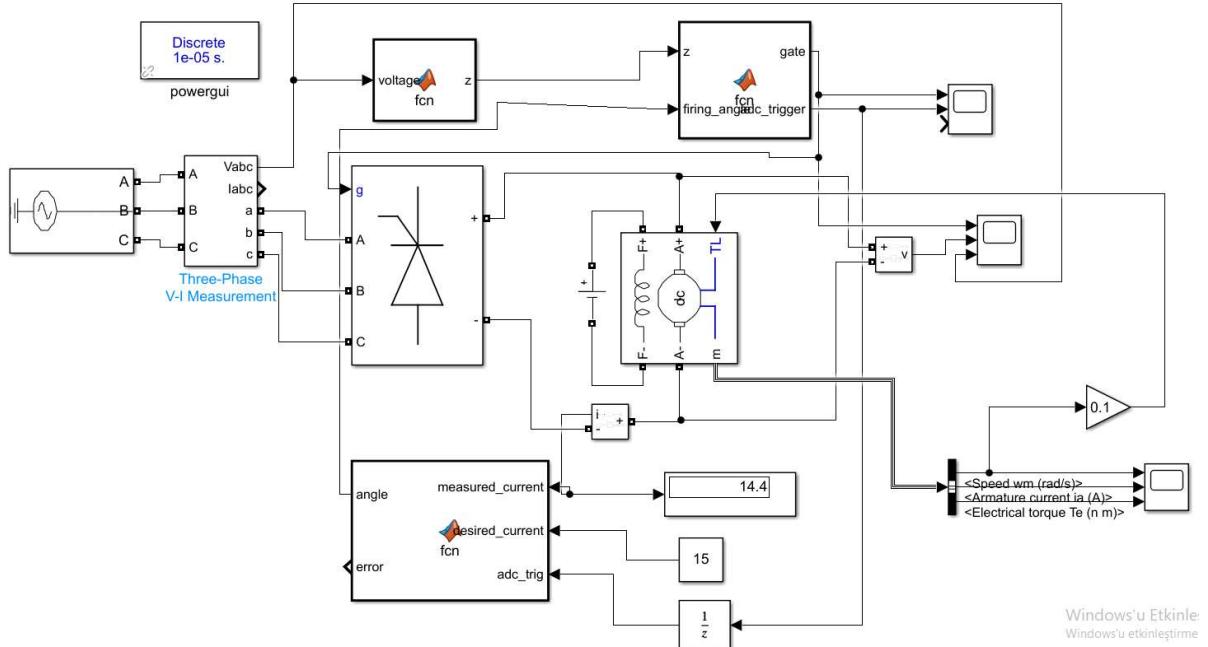
- Armature Winding:  $0.8 \Omega$ , 12.5 mH
- Shunt Winding:  $210 \Omega$ , 23 H
- Interpoles Winding:  $0.27 \Omega$ , 12 mH
- Inertia: TBA

### 3) Topology Selection

For the term project, we discussed and simulated many topologies. Details of it can be found on our simulation report, in this report we will just show a couple of them.

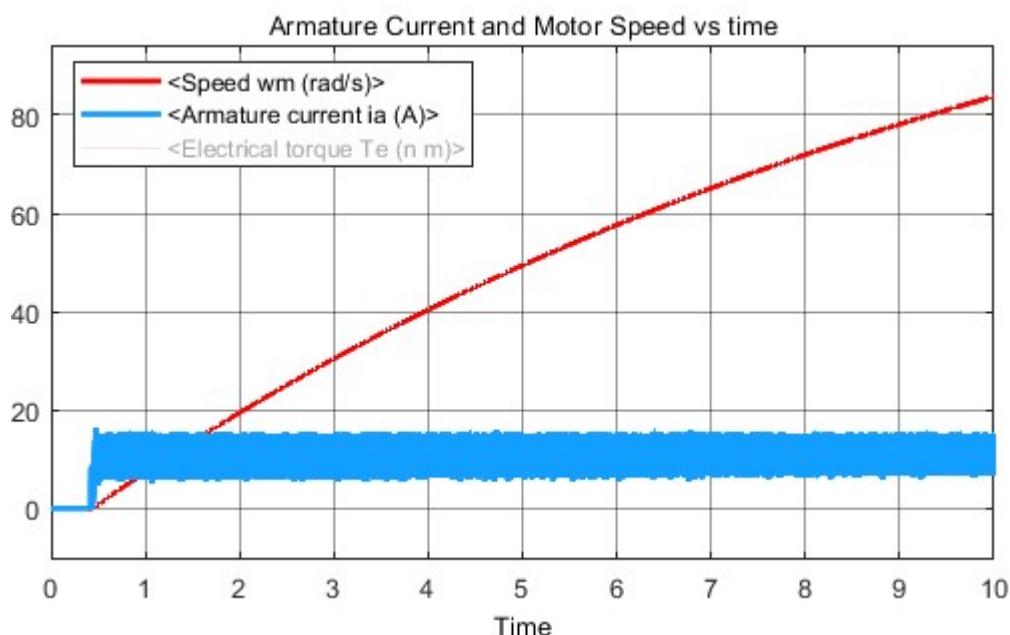
#### Three Phase Controlled Rectifier

Since that topology allows 4 quadrant operation, and more complex one we will just talk about it and our selected topology in this report.



**Figure3.1: Three Phase controlled rectifier**

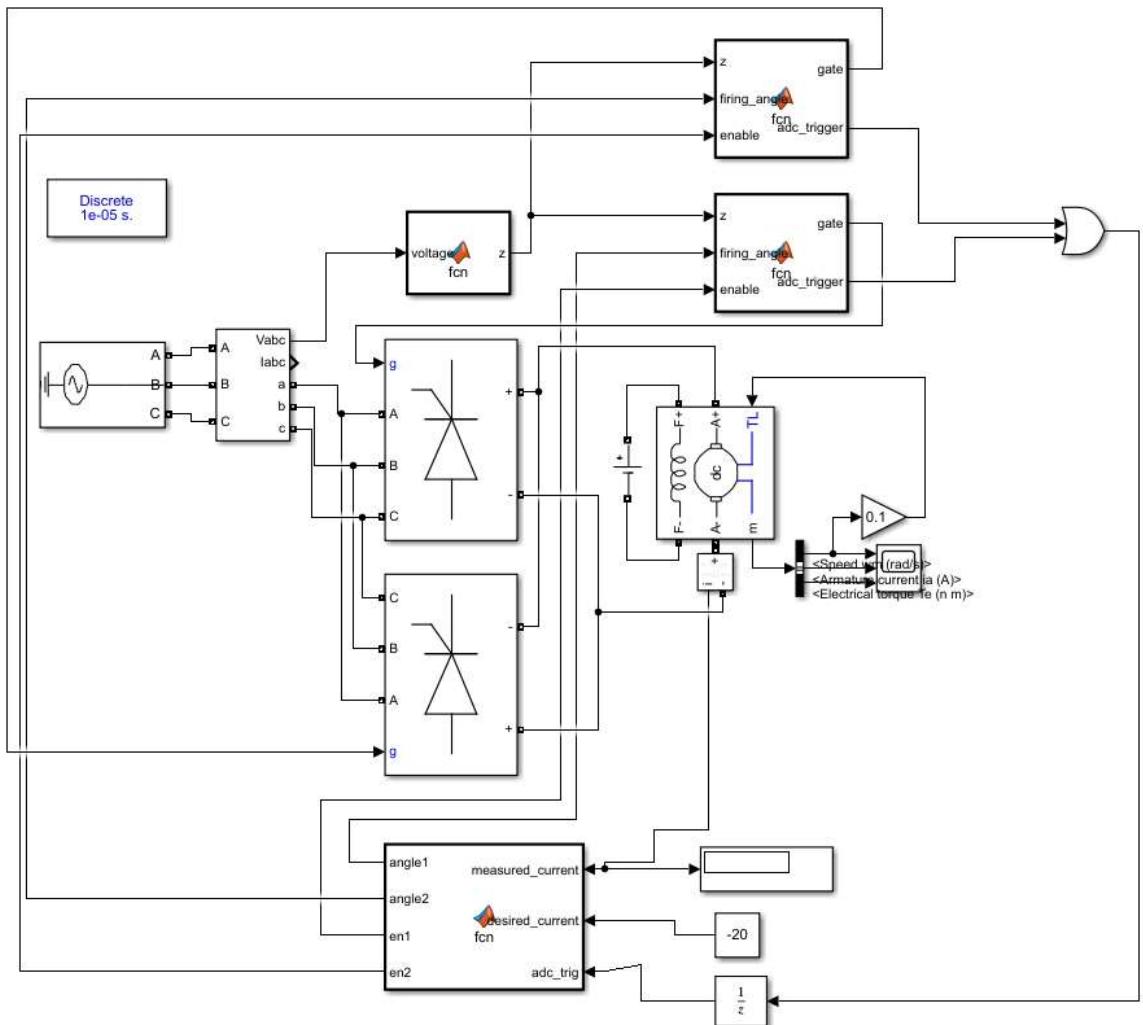
Our design can be seen above, by using matlab functions we implemented zero crossing circuit, ADC of our microcontroller and PWM output.



**Figure 3.2 Armature Current and Motor speed graph**

Since we closed the current loop of our controller, we don't have an inrush current problem.

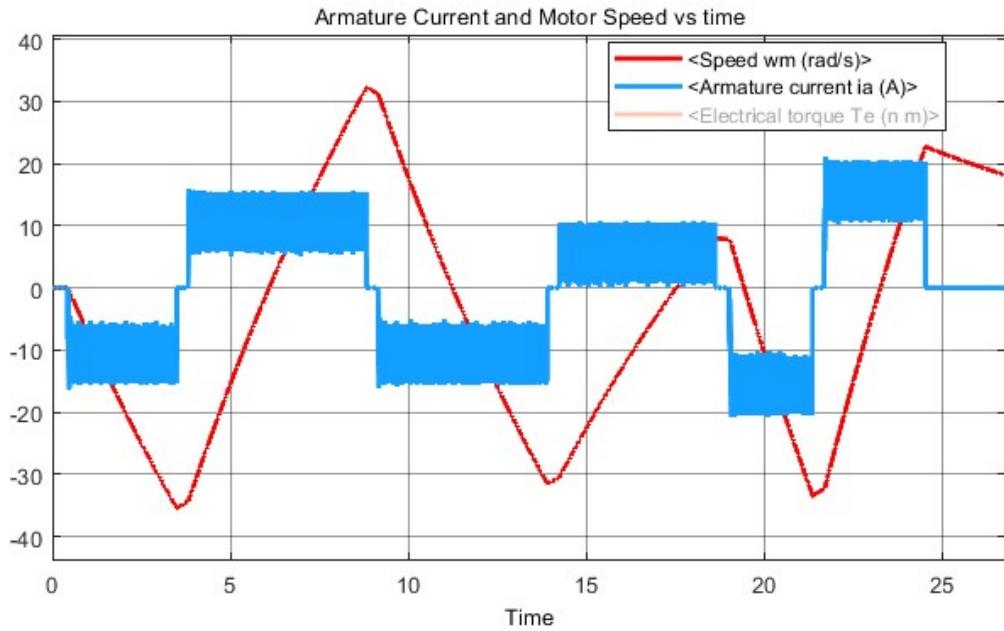
For 4 quadrant operation, we need another thyristor bridge, and control part of it. The edited simulink model of our system can be seen below.



**Figure 3.3: 3 phase controlled rectifier for 4 quadrant operation**

The figure above, shows us that our current configuration works at all quadrants.

This project is not only about our design skills, this topology can be implemented for that project but it is a complex topology. It uses 12 thyristor and drive circuitry of them, developing and testing that circuit will be really time consuming and we have limited time. This means that rather than designing a more complex one we should look for easier solutions. That's why we didn't make thyristor, and chose a different topology.



**Figure 3.4: 4 Quadrant operation using three phase controlled rectifier**

### Diode Rectifier - Buck Converter

In this topology, the circuit is actually composed of 2 different parts. First part is a diode rectifier with a filter to obtain a DC voltage from the 3 phase AC input. Second part is a buck converter to drive the electric machine from that rectified voltage. Since that topology is selected, detailed simulations can be found on the rest of the report and simulation report.

At the buck converter stage, rather than using a regular buck converter, we made a change. In a common buck converter, the switching element is in the high side and requires more complicated drive circuitry. Since our load is a DC machine we can change the location of the switch and design a cheaper PCB.

Common and modified buck converter topologies can be seen below.

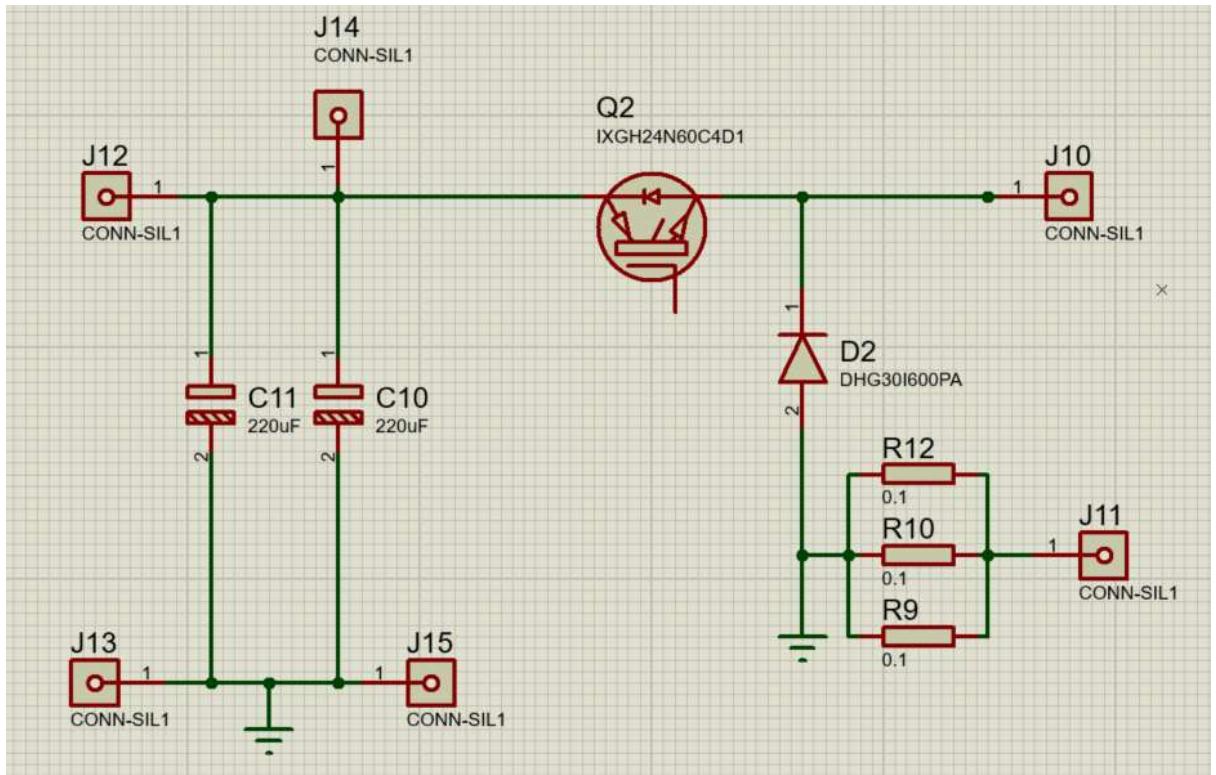


Figure 3.5: Regular Buck converter circuit

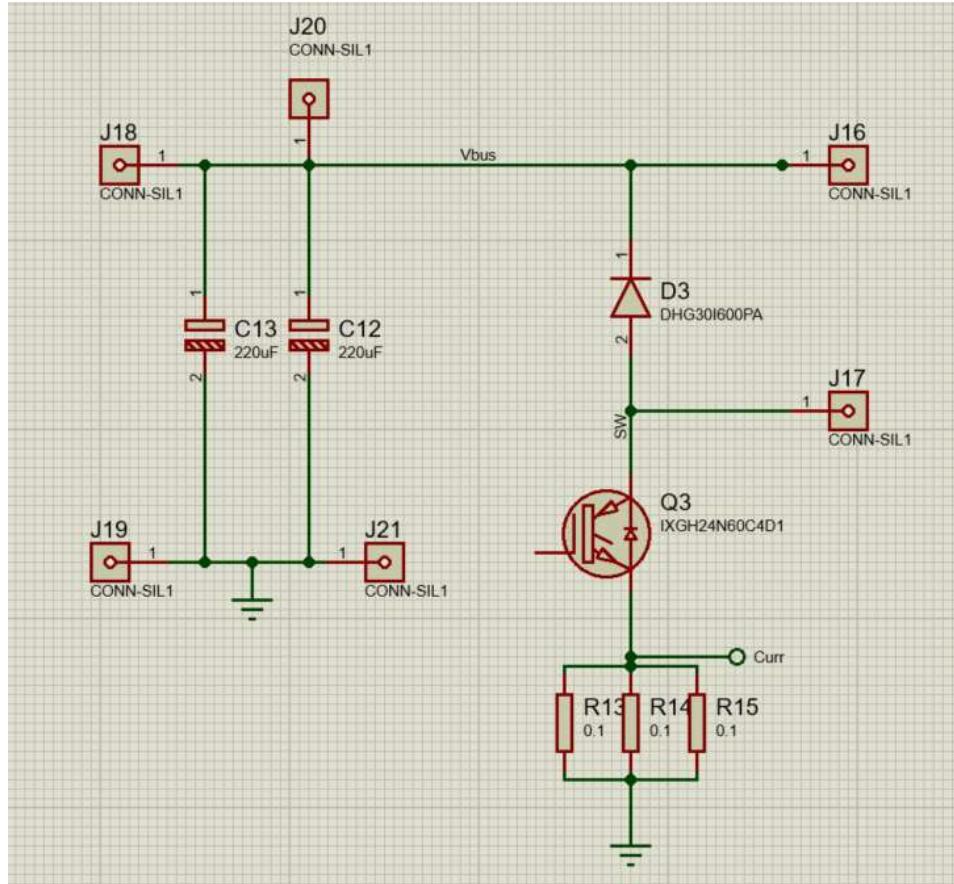
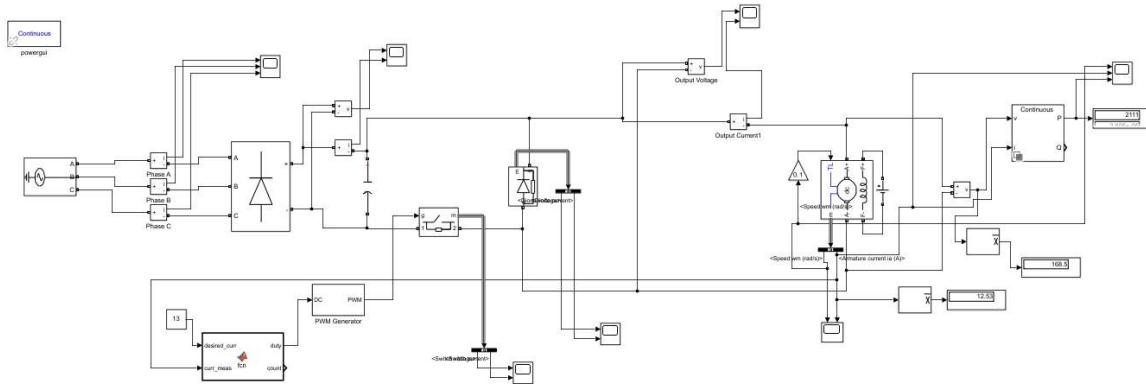


Figure 3.6: Modified buck converter circuit

## 4) Simulation

The three phase full bridge rectifier and buck converter is simulated together with motor at the load side. In the simulations, we loaded the motor with torque that is proportional to its speed. The switch in the buck converter is taken to the low-side to be able to use non-isolated gate driver.

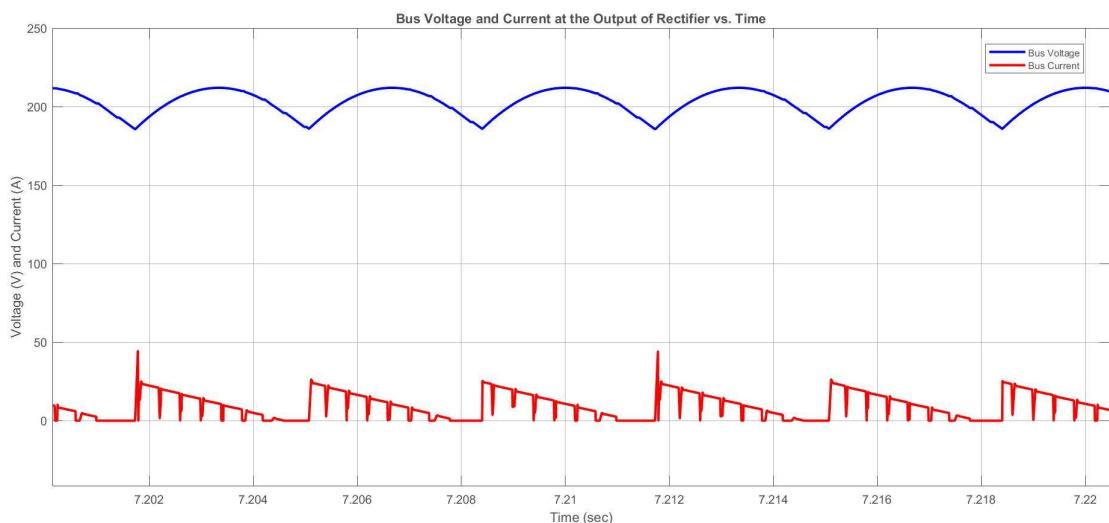
The load current is extremely high at start-up when the system works without any control. Thus, we designed a closed loop control path to set the current limit 13 A. The 13A is selected because we need to obtain approximately 11.1 load current for tea bonus (the load consumes approximately 2kW power). The final version of the design is displayed in Figure 3.1.



**Figure 4.1: Circuit Schematic**

### Rectifier

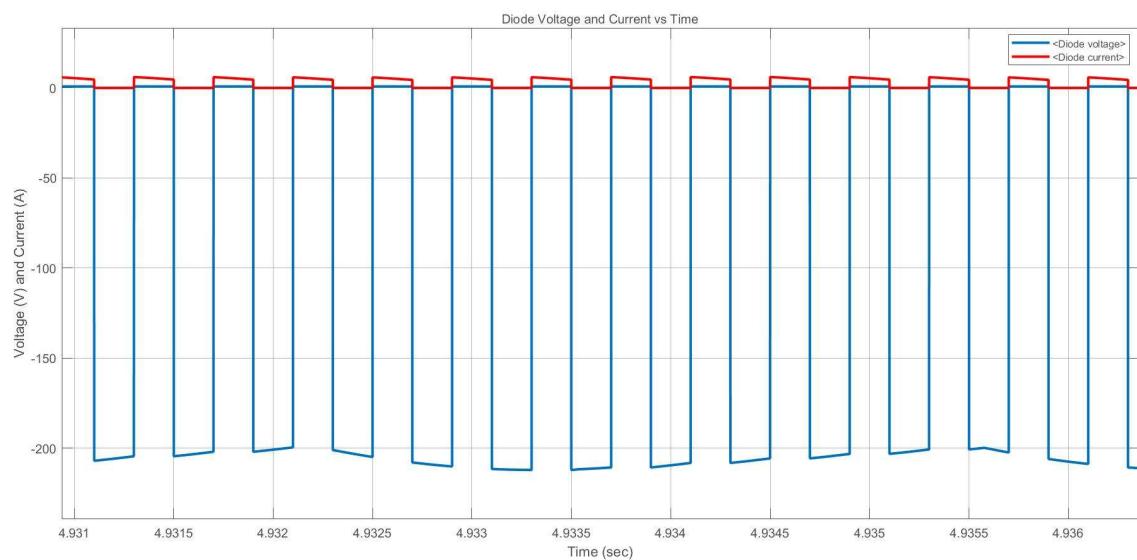
The bus voltage oscillates between 212V and 184.4V with a ripple voltage 27.6V when a 440uF smoothing capacitor is used at the output terminal of the rectifier. The average output voltage of this block is 202.2V. The waveforms are displayed in Figure 3.2 (Duty cycle is 0.9 in this simulation).



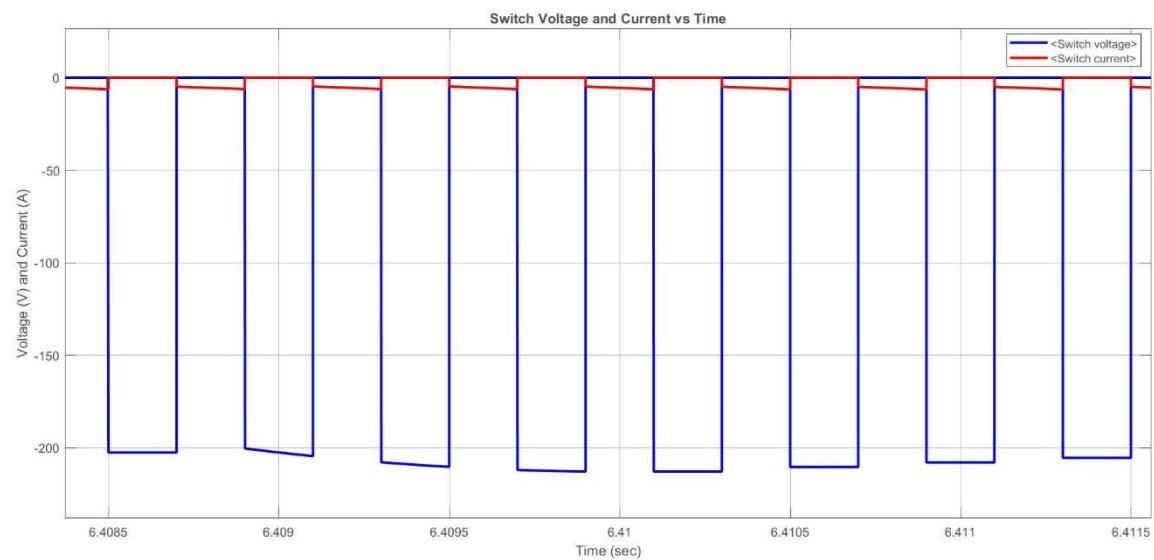
**Figure 4.2: Bus voltage and Current at the Output Terminal of Rectifier**

## Buck Converter

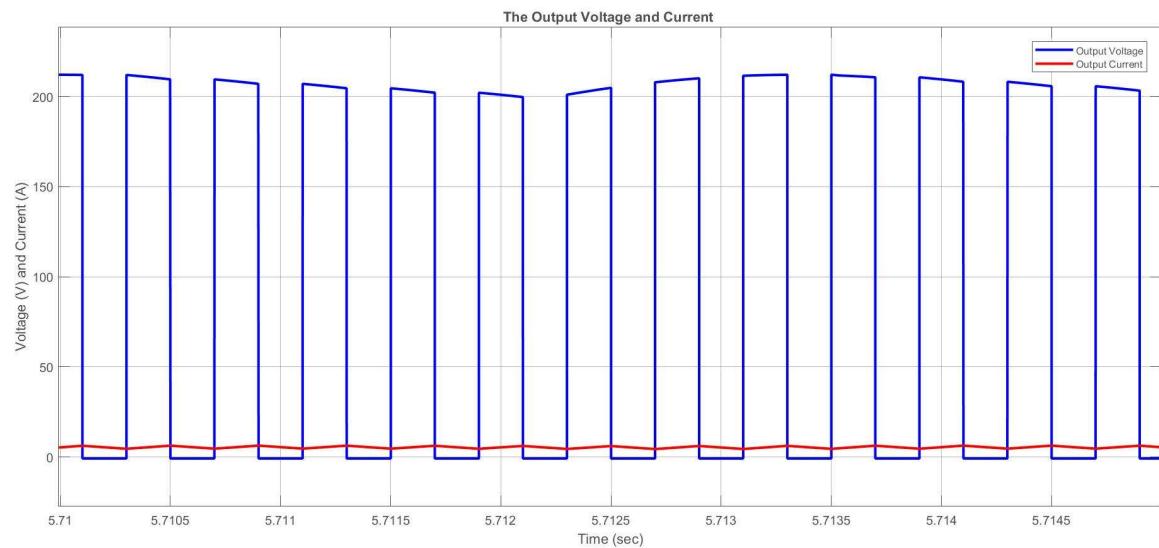
The waveforms for 0.5 duty cycle and 0.9 duty cycle are displayed below.



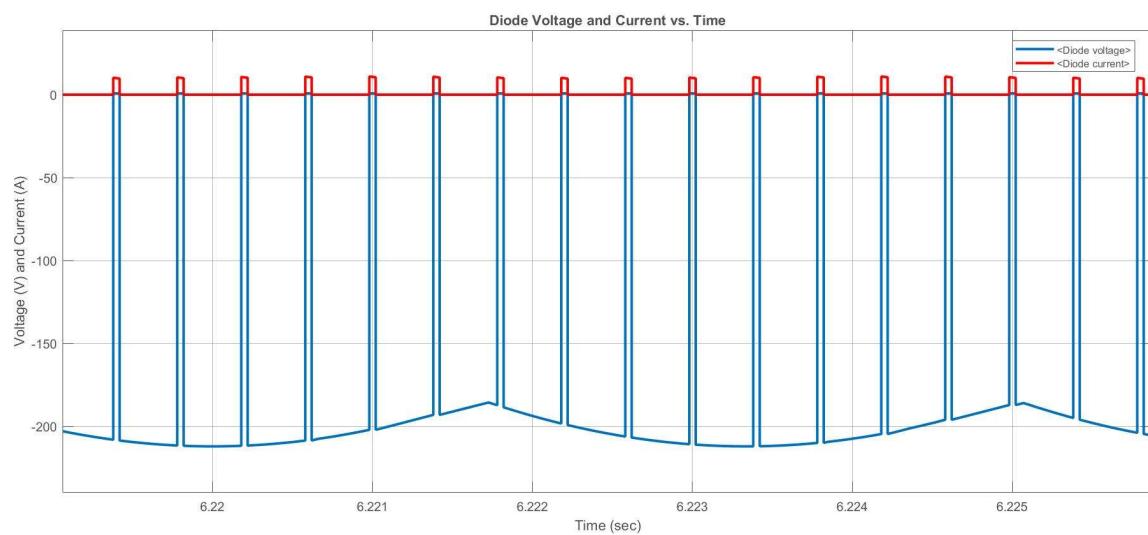
**Figure 4.3: Diode Voltage and Current for 0.5 Duty Cycle**



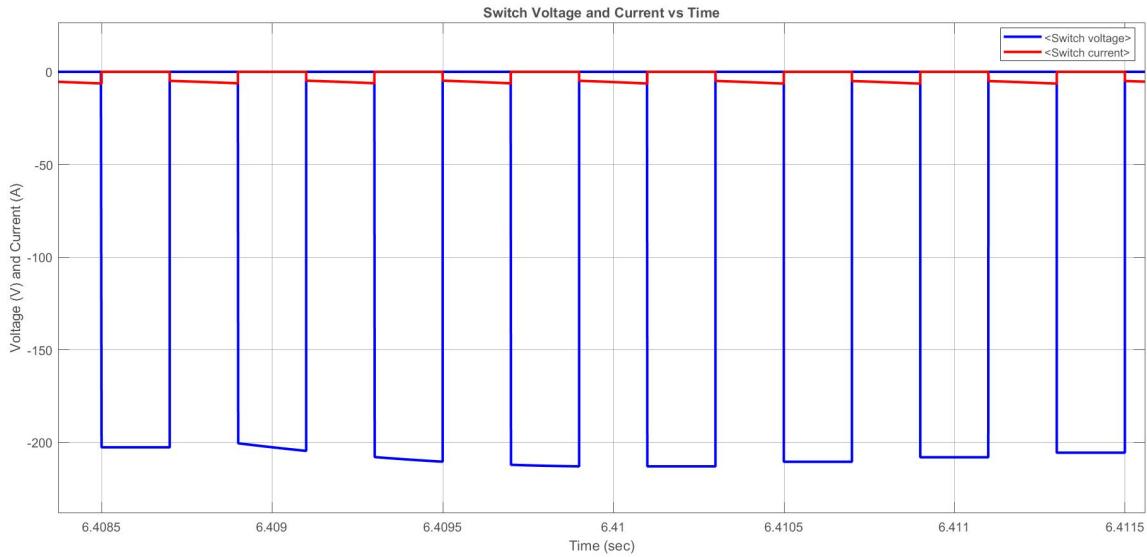
**Figure 4.4: Switch Voltage and Current for 0.5 Duty Cycle**



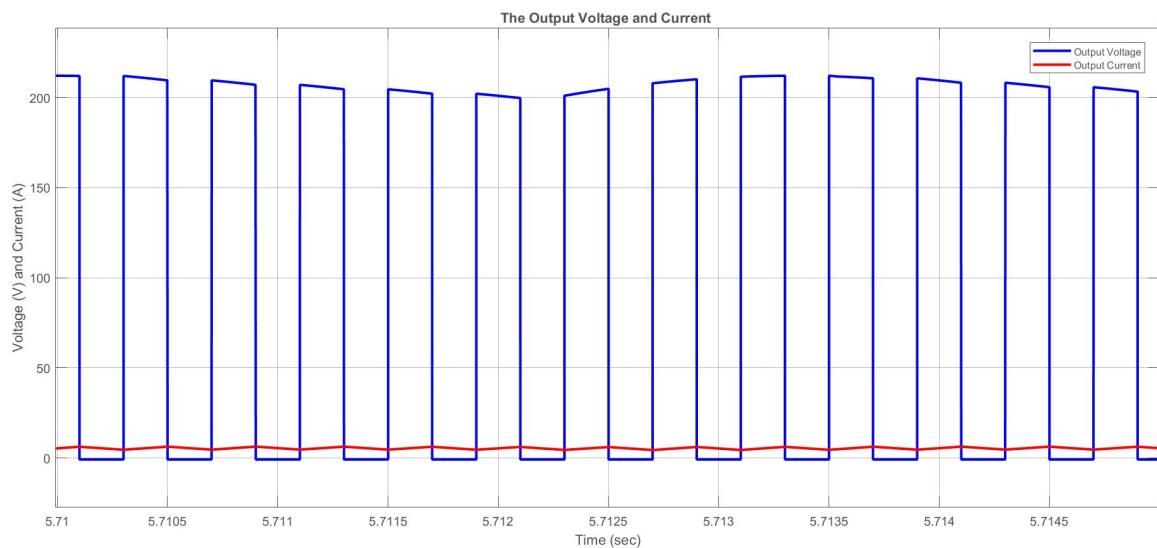
**Figure 4.5: Output Voltage and Current for 0.5 Duty Cycle**



**Figure 4.6: Diode Voltage and Current for 0.9 Duty Cycle**

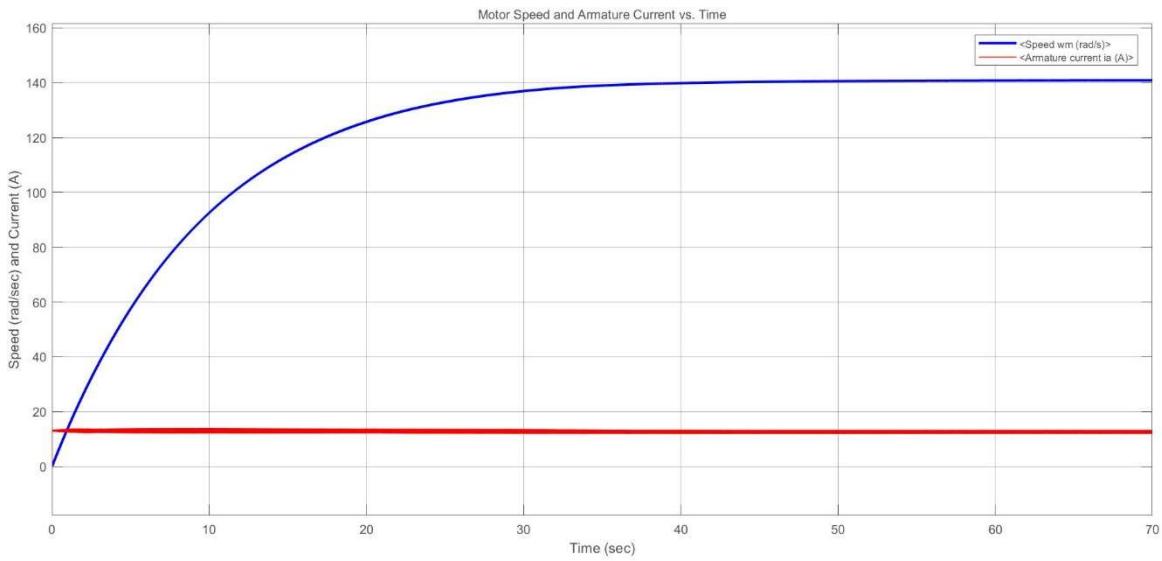


**Figure 4.7: Switch Voltage and Current for 0.9 Duty Cycle**

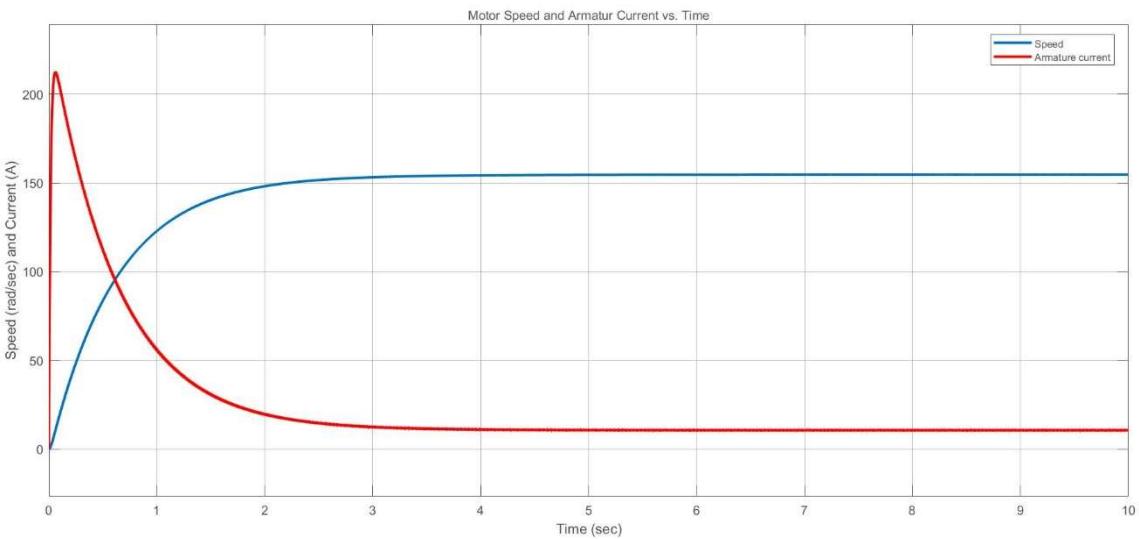


**Figure 4.8: Output Voltage and Current for 0.9 Duty Cycle**

Without any control, the output current reaches 215 A at the beginning, which is unacceptable for our devices current limit. Thus, we designed a closed-loop feedback path to set the current limit 13A. This modification increased the time to reach steady state, but protected our circuit from high current damage. The transient current and speed waveforms with and without any control are displayed in Figure 4.9 and 4.10, respectively.



**Figure 4.9: Motor Speed and Armature Current with Controlled Current**

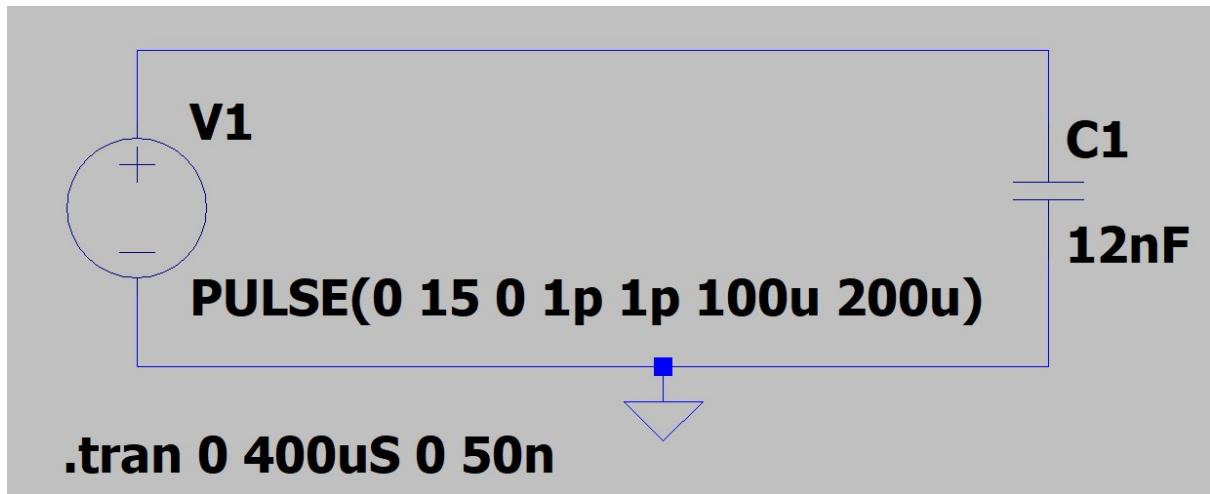


**Figure 4.10: Motor Speed and Armature Current without Controlled Current**

## 5)Gate Driver Design

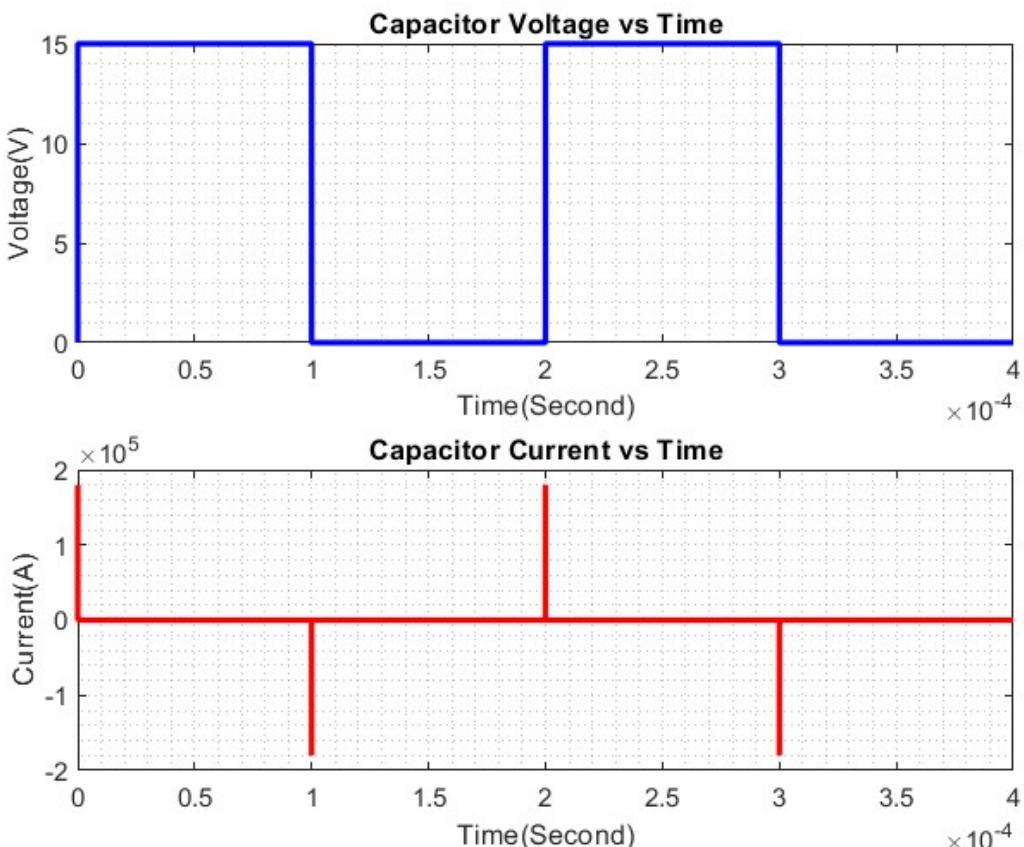
To design a gate drive circuit, we should firstly know why we need such a circuit. In our driver circuit, basically by using a power switch(in our case it is an IGBT) we are controlling the mean voltage applied to the armature of the electric machine. IGBT is a voltage controlled device, and the voltage between gate and emitter terminal is controlling the device. Due to the structure of the IGBT, we could simply think that, voltage of the capacitor between the gate and emitter controlling the device(Actually this capacitance value is a function of a voltage, so usually we need to talk about charges but to simplify the problem we will assume a constant capacitance). Moreover, if we model our control signal as a voltage source, we can make some simulations to understand the need.

According to datasheet of our IGBT,  $Q_g=167\text{nC}$  when  $V_{GE}=15\text{V}$ . If we assume a capacitor whose capacitance is not dependent on the voltage, then the corresponding capacitance value is almost equal to  $12\text{nF}$ . Lets simulate our very idealized circuit using spice.



**Figure 5.1: Idealized gate circuit with 5Khz %50 Duty Cycle**

As can be seen from the figure above, we idealized our gate circuit and applied a 5Khz PWM to CGE to charge and discharge it.



**Figure 5.2 Capacitor current and voltage waveforms**

When we simulate the idealized circuit, we could see the current and voltage waveform above, as can be clearly seen we have currents in the kA range. It is impossible to supply for our control circuit, however this circuit is quite idealized and the parasitic inductances are omitted. Even though we have a perfect layout, we will have the inductances due to our transistor package, assuming that we have a 50nH gate inductance(we will add this inductance series to the capacitor), lets simulate our circuit again.

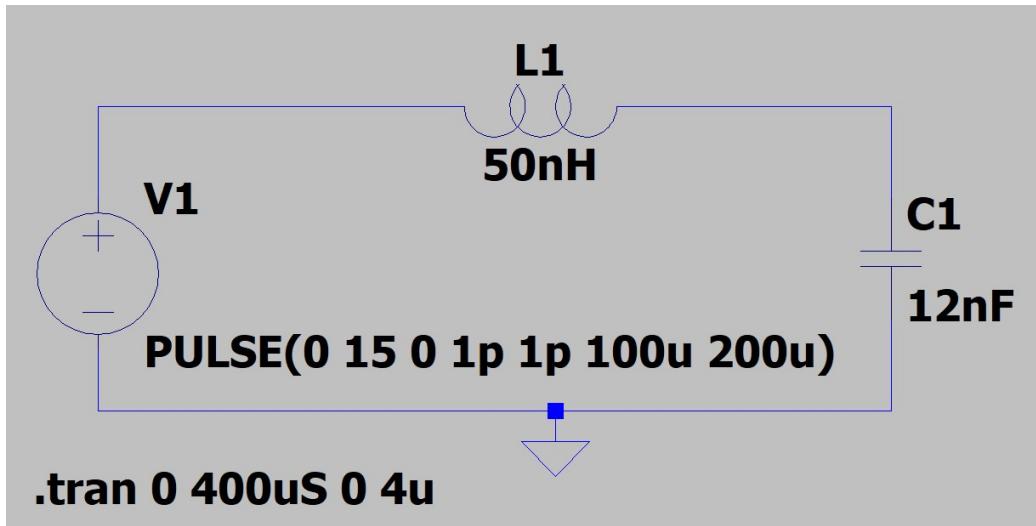


Figure 5.3: Idealized circuit with inductance

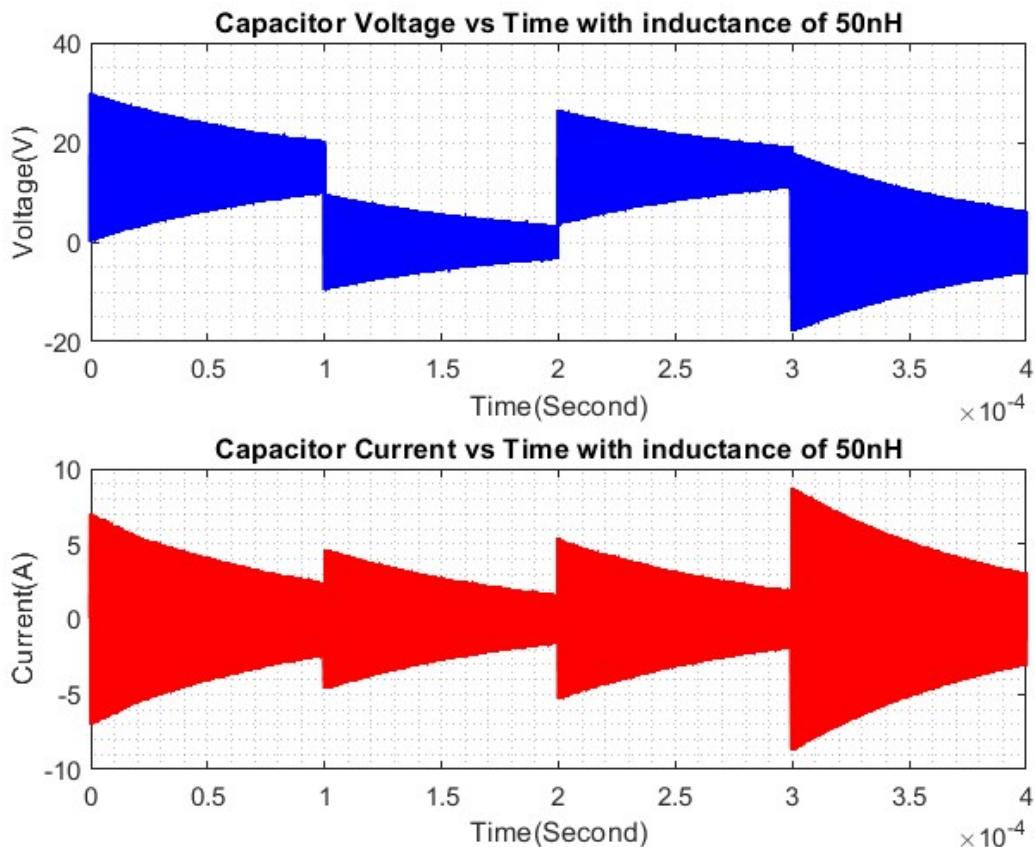
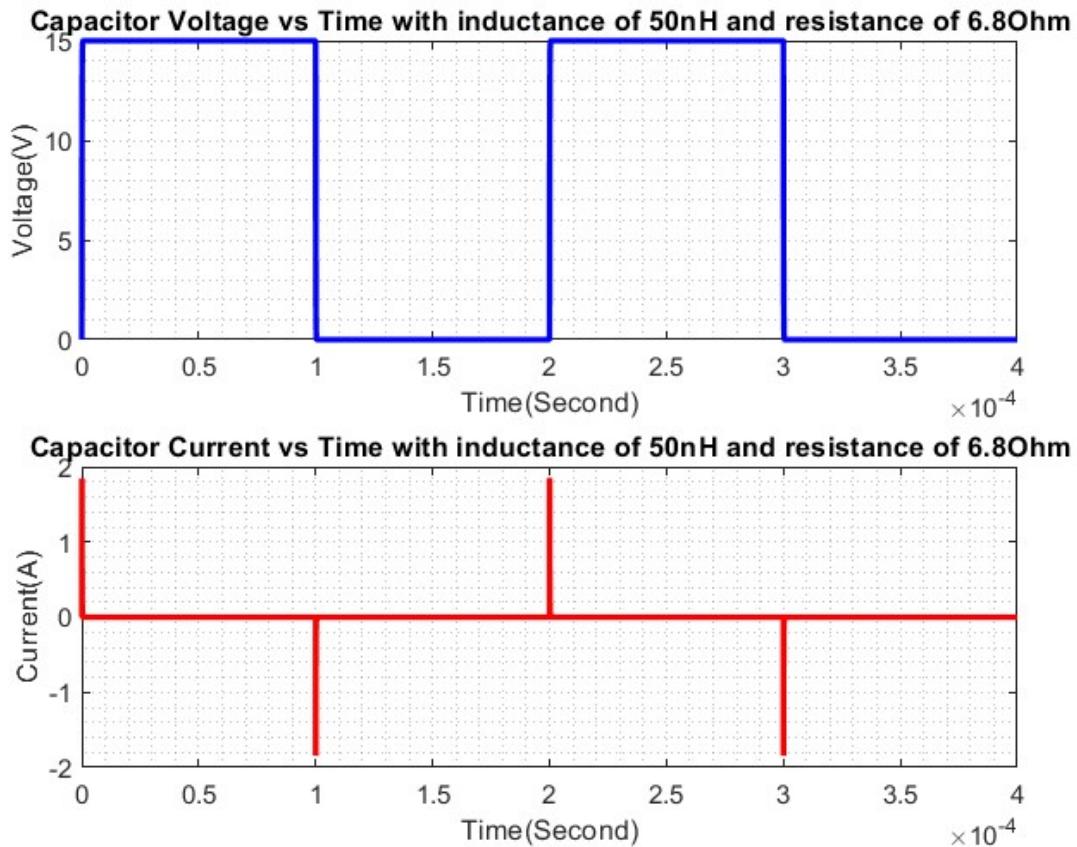


Figure 5.4: Capacitor current and voltage waveforms with added inductance

As it can be seen from figures above, with the inductance we have growing oscillations at the gate. Moreover, the voltage waveform overshoots the safe region and damages the switch. Even though the current through the capacitor is now in the order of amperes(which we can supply somehow), the switch is out of our control. So this means that connecting the pwm source directly to the gate is a bad idea. To create a damping effect and limit the gate current we can add a resistor to that circuit. There are also transistors with integrated small gate resistors however according to the datasheet of our transistor we don't have it.

We should also state that in the simulations above we made a lot of assumptions, in reality every conductor have a resistance and the capacitance is dependent on the voltage value and every voltage source has a output resistance, this means that in reality we could adjust our pwm circuit such that we can connect our gate directly to the pwm circuit. However, to show the effects simply we will continue with our assumptions.

To reduce the ringing and limit the maximum gate current lets add a 6.8 Ohm resistor to our circuit series with the inductor.

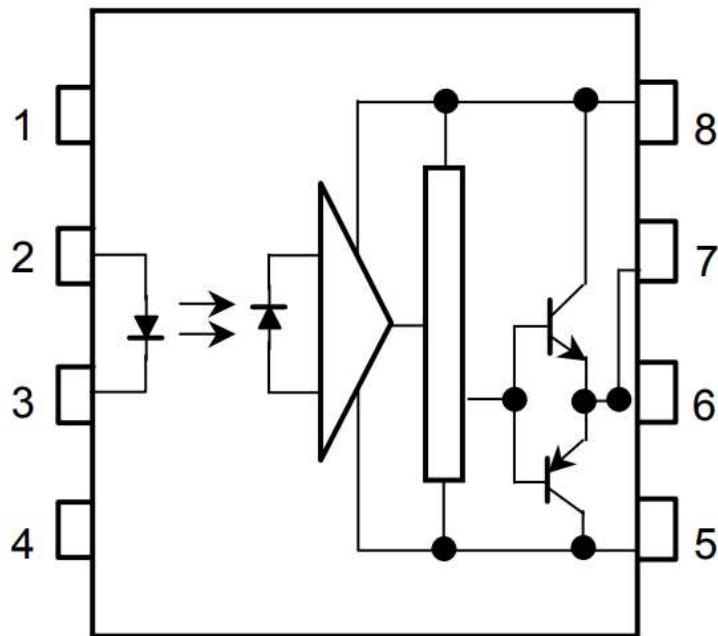


**Figure 5.5: Capacitor current and voltage waveforms with added inductance and resistance**

By adding the resistance, we get rid of the oscillations and limit the capacitor current. However we still need to supply 2A to the gate. Our controller(in our case it is a microcontroller) needed to have a pin that could output pwm with logic 1 with 15V and logic 0 with 0V and the output current of 2A. When we think about it, it is easier to design a controller and then an amplifier/buffer for that current and voltage values than designing a controller pwm stage with that ratings. That amplifier/buffer stage is the gate driver circuit. Of course it is possible to drive IGBT with different gate current, but reducing the gate current will cause higher rise and fall time. Higher rise and fall time means that we will have higher switching loss.

In our design we are using a STM32G4 series microcontroller as a controller. It could output 3.3V and sink/source maximum of 20mA from IO pins. That voltage value is less than the threshold voltage of our IGBT and even though we have enough voltage that current value is really too low to switch at 5 Khz. In order to solve that problem, we used an opto-gate driver TLP250 in our design.

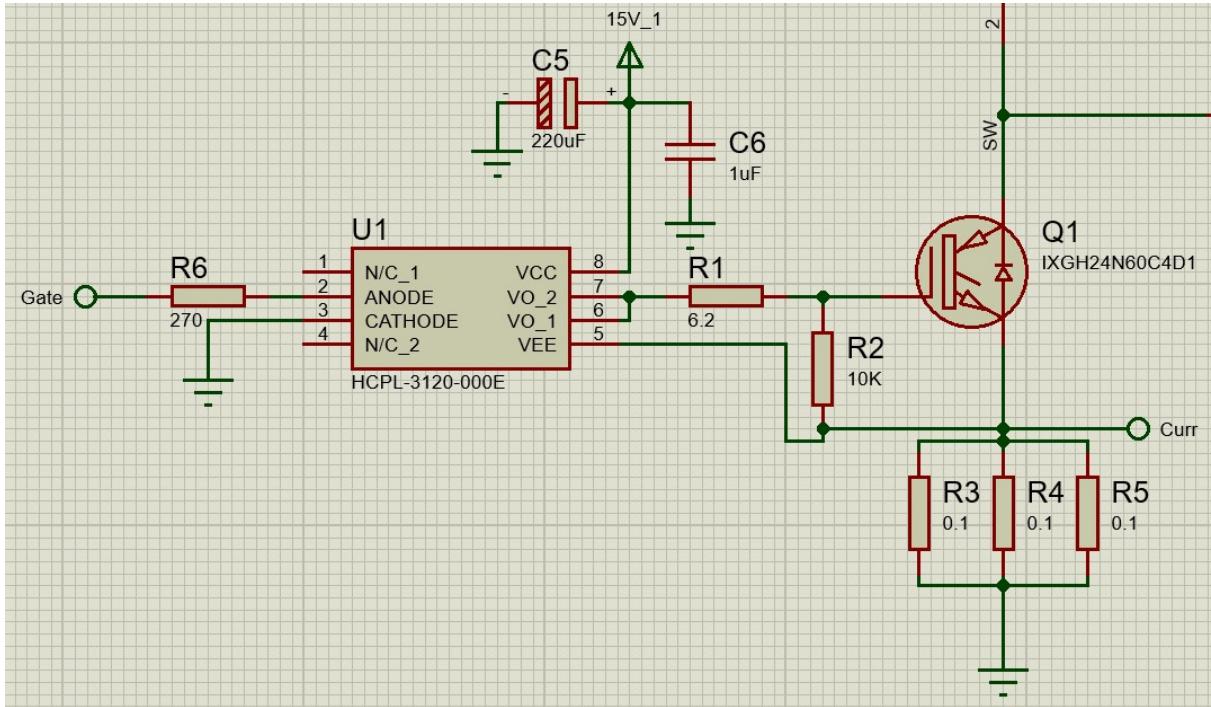
## Pin Configuration (top view)



**Figure 5.6: Internal structure and pinout of TLP250**

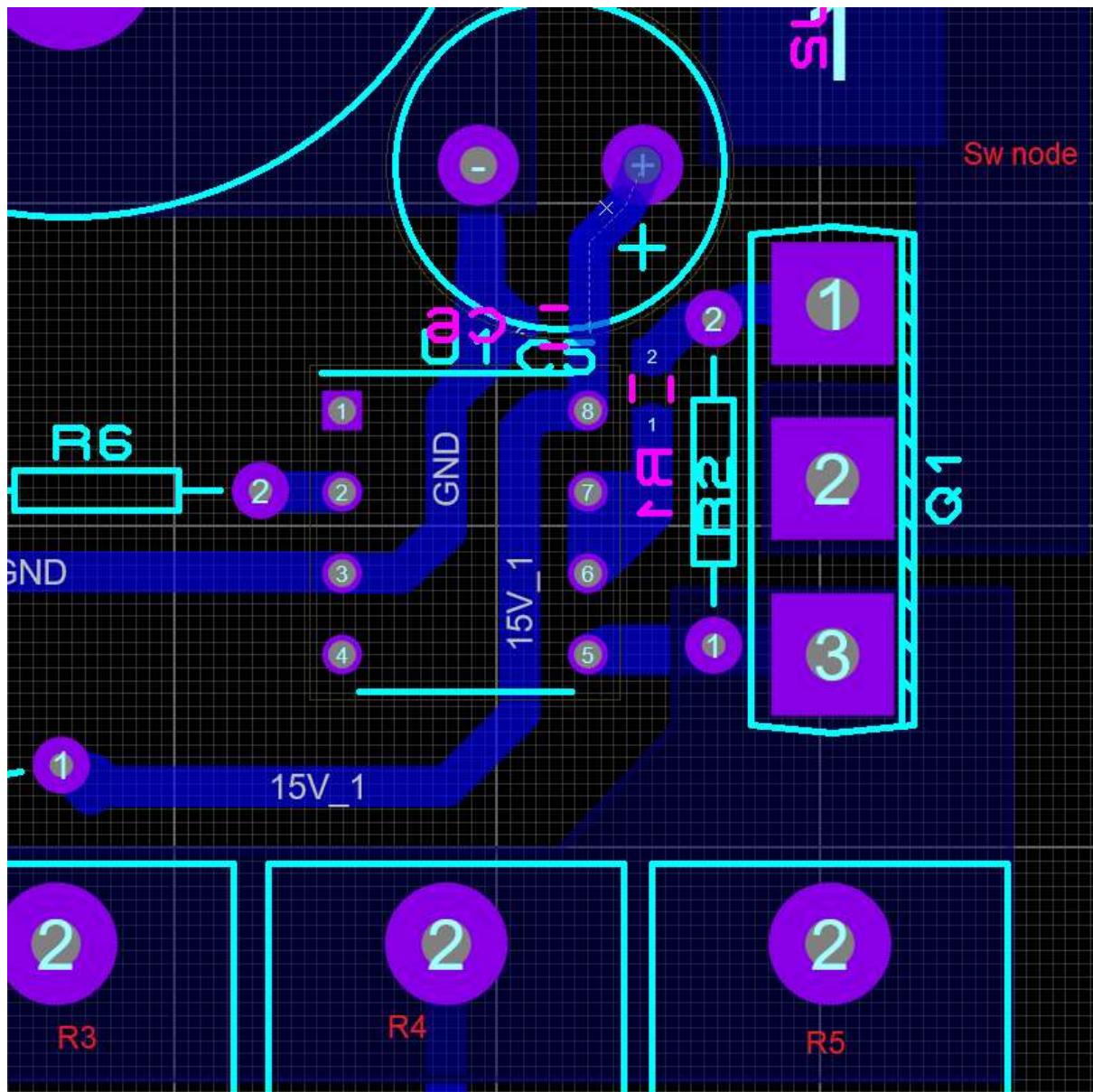
When we look at the internal structure of the TLP250 we can see that, by controlling an LED, we can change the state of output, this means that if there is current through the LED, then the output will pull high, otherwise low. With that structure we could drive our IGBT, moreover this structure actually allows us to isolate the HV side from the controller side. However, we will not use that.

We can see our gate driver in our schematic, since it has the same pinout and package with HCPL3120, we designed it as HCPL3120 but used TLP250.



**Figure 5.7: Our gate drive circuitry**

In the schematic above, we can see our gate driver as U1. Signal which is named as Gate is actually the PWM signal from our controller, C5 and C6 are the decoupling capacitors, R1 is gate current limiting resistor and R2 is to ensure that the switch is not in conduction when there is no energy in the system. Normally we have to connect the Vee pin to circuit GND or a negative voltage relative to the emitter, however we connect it to a variable voltage node. Moreover, we can say that our emitter voltage is not constant at that circuit. We added small resistance between the emitter terminal and circuit ground in order to measure the current passing through the switch. In our design maximum current through that resistors are 15A which means that maximum of 0.5V voltage difference between emitter and ground. This voltage difference is not a problem since our drive voltage(15V) is high enough. However, still according to the datasheet that pin should be connected to the ground. To understand why we didn't connect it directly to the GND we should look at our layout design.



**Figure 5.8: Layout design**

If we examine our single sided layout design, it is clear that connecting the pin 5(or GND pin according to the datasheet) GND will create more parasitic inductance in our design. Hence, to minimize parasitic inductances, we connect pin 5 to the emitter of the IGBT.

## 6)Controller

Main duty of the controller in our design is by taking input from the user as desired current value, measuring the average current and then creating a pwm to maintain that current value. It is actually a quite simple job and can easily be done with an analog controller in a good design. However, due to limited time and available components, and more importantly to achieve a flexible design that can be modified just within a second, we chose to make it digital. In this project we used STM32G431 microcontroller with Cortex M4 core. To understand why we used that microcontroller, we will firstly look at our circuit.

To measure the current through any component, we simply have 3 solutions.

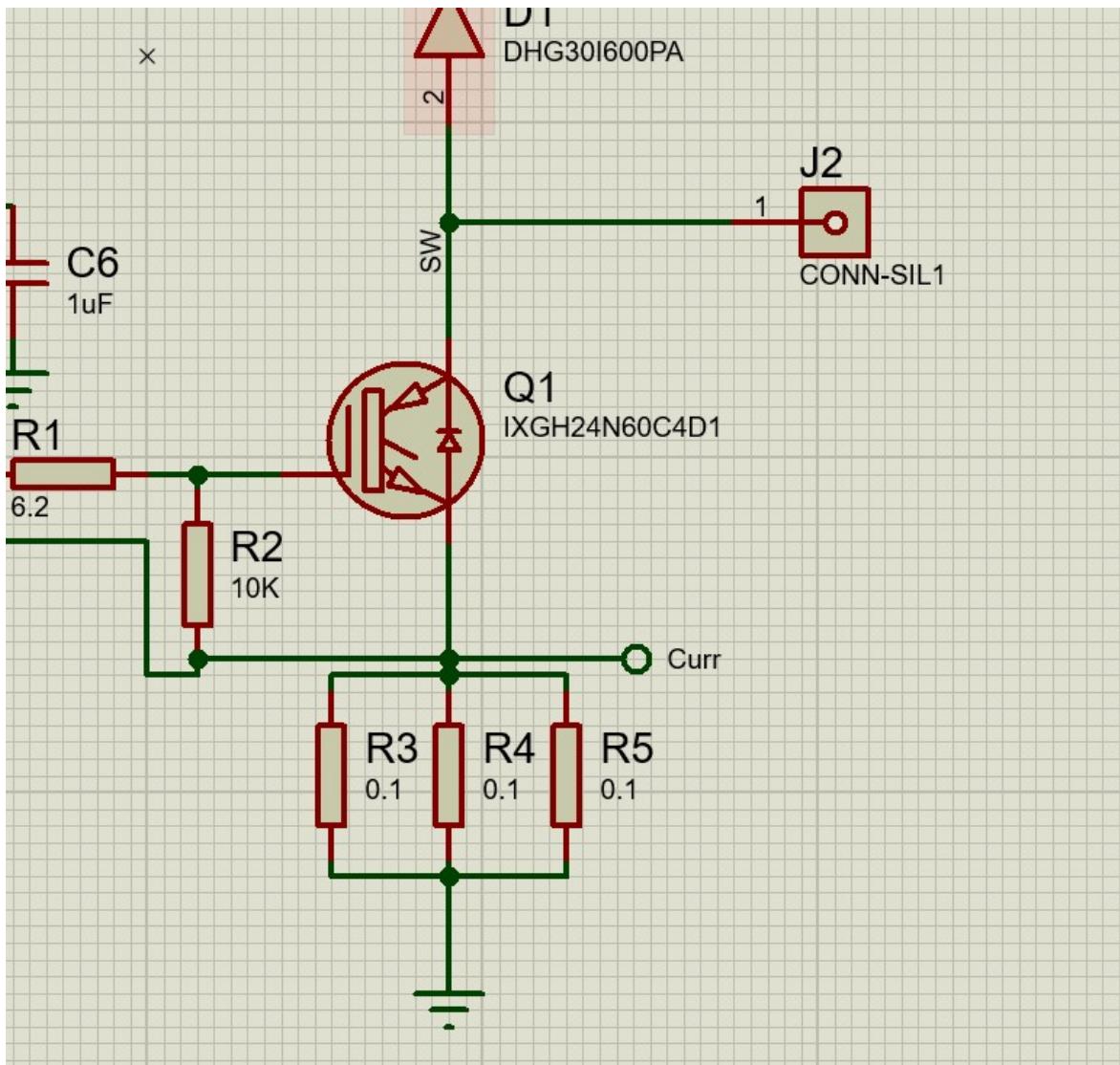
First one is we can use a current transformer, which obviously works with ac only. We could simply add a current transformer to our three phase input and measure the current using the transformer is possible. However, it comes with several drawbacks. Firstly, to measure the current from 0 to 15A, we should have a really well designed and calibrated circuit at the secondary side of the current transformer. Then due to harmonics, we have to implement a true RMS algorithm, which causes high computational cost(our microcontroller could do this). In this method our controller will have the chance of being isolated from the grid.

Secondly, we could measure the current by measuring the magnetic sensors. With that method we could measure both ac and dc, this means that we can locate it anywhere we want. Moreover, there is a lot of ready to use products(Sensors of Allegro and LEM widely used in many sectors, these sensors can be easily bought from Turkey). But it is a rather expensive(around 4 usd for our case) solution.

Lasty, we can add a small resistance to a path then measure the voltage across it. However, we should consider that we are creating a new loss in our circuit. This loss can be directly proportional to our added resistance(called shunt resistance). If it will be too high it will create too much loss, if it will be too small we will possibly have noise and bandwidth problems. To measure current between 0-15A with the ADC reference voltage of 3.3V, we can say that we can put a 0.22 Ohm resistance, but this resistor will create a loss of 49.5W at 15A. This means that we should add a small resistance and then amplify it.

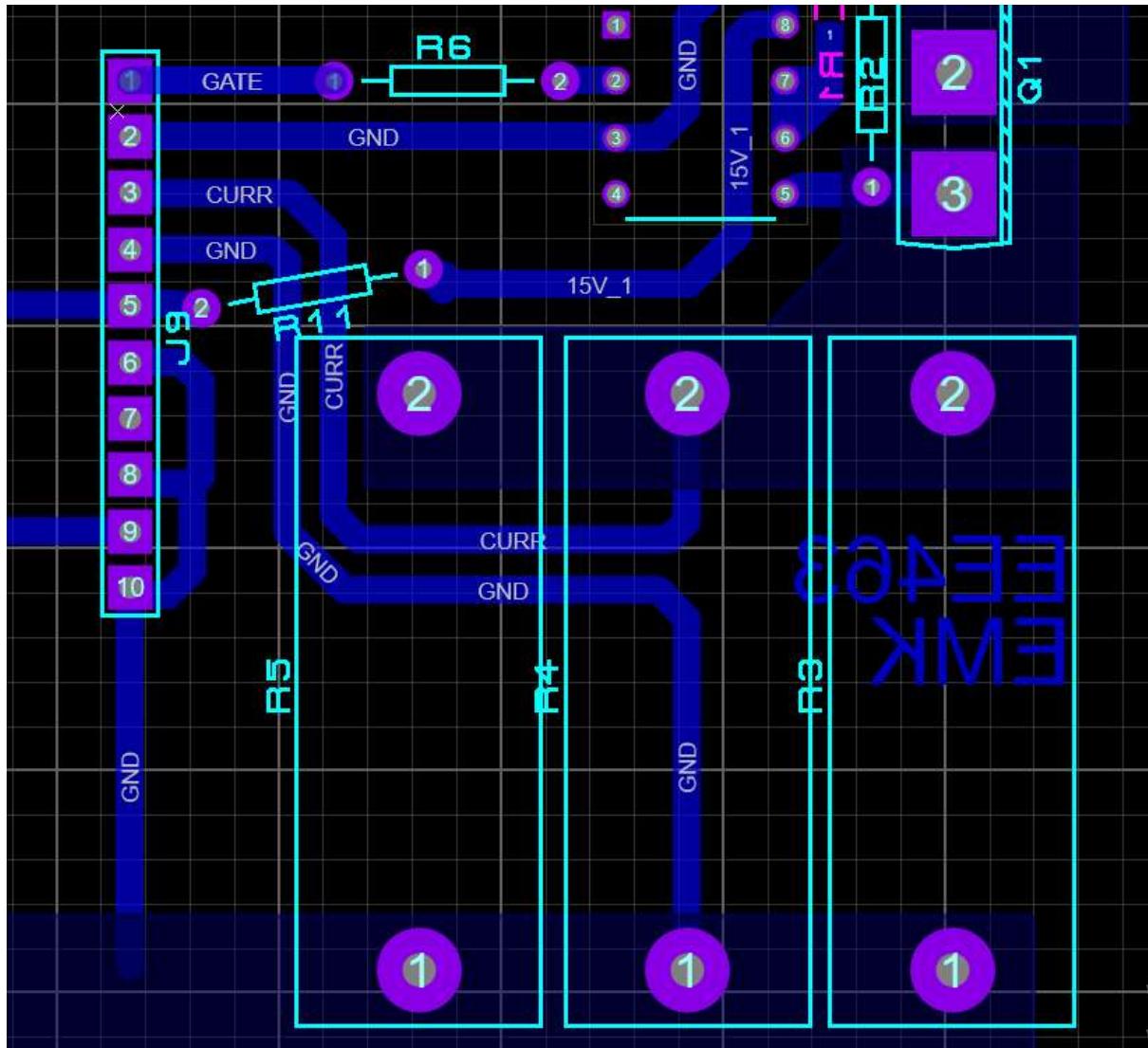
For the amplification purpose, we can use an opamp. However, that opamp should have certain properties. First of all, we will use a single supply and our opamp should swing its output between supply rails or only from gnd rail to a higher point(otherwise we wouldn't be able to measure 0A current or currents below some point). Secondly it should have low input/output offset voltages, it should have a high gain bandwidth product(higher the gain means less output bandwidth, this means that we should have at least GBW of our gain times the frequency of the signal which wants to be measured), high CMRR. There are a lot of products for that purpose but since we are making our circuit single sided using components in Turkey, we have limited options.

For the loss of shunt resistance we decided to go below 10W at maximum. Which means that our sense resistor should have less than 45mOhm. For that purpose we paralleled three 100mOhm cement resistors.



**Figure 6.1: Shunt resistors**

With that configuration, we get approximately 33mOhm shunt resistance and 7.5W loss at maximum current. To correctly amplify over the voltage of those resistors, we have to be careful about layout design. We will not get into details of it, but this method is called as kelvin connection or 4-wire sensing. Our layout design to the amplificator can be seen below.

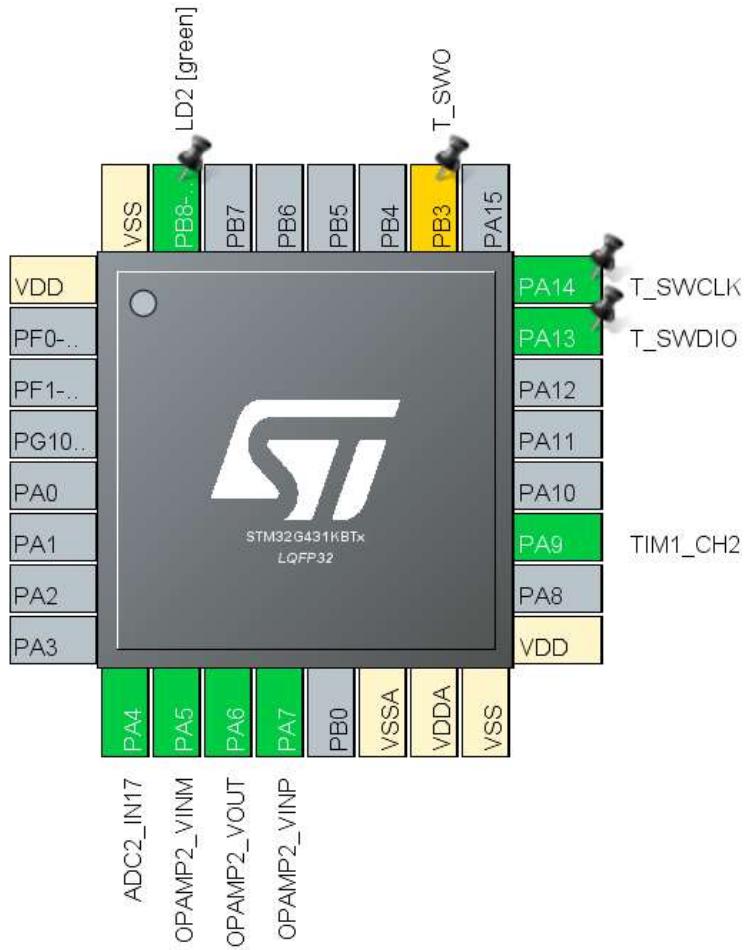


**Figure 6.2: Kelvin connection of shunt resistors**

The paths named as curr and gnd are current sense signals. It is of course better to amplify in a close point is better, however due to limited space in our PCB and design choice of our we should do it in a separate control PCB.

Up to that point, we know that we need a microcontroller to make ADC measurements and create pwm, we need an op-amp to amplify the current signal(otherwise we will have less resolution). At that point, we know that ST have microcontrollers which have both of them. Actually STM also has some microcontrollers with integrated gate drivers but it is harder to find those microcontrollers in crisis. It is possible to choose the cheapest microcontroller with that feature, but unfortunately we have limited time, and the cheapest microcontroller which has a development board in Turkey is STM32G431. That's why we used that microcontroller in our design.

Our code is quite simple but due to HAL libraries supplied from ST, it looks very complicated. So, we will not put it in here completely, we will just look at some important parts. Firstly, we should also say that, rather than discrete components, we used integrated PGA of our microcontroller to sense our current. That means that we only need to connect the sense paths, the required resistors and op amp are inside of our microcontroller.



**Figure 6.3: Pinout of our connections**

As can be seen from figure above, PA4 is a simple ADC input for user input, PA5.6 and 7 are pins of op amp, and PA9 is the PWM output. Other connections are for debugger connection and a simple LED.

When our controller starts to work, it firstly enables the PWM timer and creates a pwm signal with no output. That PWM is used for triggering ADC measurements(since current through shunt resistors are 0 when the switch is not in conduction, we should make our measurements while the switch is in conduction) after that it calibrates ADC and enables op-amp and ADC with DMA to transfer measurements. Then, it checks if the user input, if it does not mean 0A, rather than creating PWM it waits until the input goes to zero. After that it enables the PWM output channel and updates the duty cycle every 2mS by looking at the output of the simple I controller.

```

/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_DMA_Init();
MX_ADC2_Init();
MX_TIM1_Init();
MX_USART2_UART_Init();
MX_TIM2_Init();
MX_OPAMP2_Init();
/* USER CODE BEGIN 2 */
HAL_TIM_PWM_Start(&htim1,TIM_CHANNEL_1);
HAL_GPIO_WritePin(LD2_GPIO_Port, LD2_Pin, GPIO_PIN_RESET);
//HAL_TIM_OC_Start(&htim1,TIM_CHANNEL_1);
HAL_ADCEx_Calibration_Start(&hadc2, ADC_SINGLE_ENDED);
HAL_Delay(100);
HAL_ADC_Start_DMA(&hadc2,(uint32_t *)ADC, 2);
HAL_TIM_Base_Start_IT(&htim2);
HAL_OPAMP_Start(&hopamp2);
HAL_Delay(10);
while(ADC[1]>100){
    HAL_Delay(1);
}

TIM1->CCR2 = 0;
HAL_TIM_PWM_Start(&htim1,TIM_CHANNEL_2);
HAL_GPIO_WritePin(LD2_GPIO_Port, LD2_Pin, GPIO_PIN_SET);

/* USER CODE END 2 */

```

**Figure 6.4: Startup process**

At that point we should mention our ADC measurement time. Since we have a highly inductive load at the input, we can assume that our current waveform's AC part is almost a triangular wave. If we made our measurements at the middle of the rising edge, we will measure the mean current directly for that cycle. That's why our ADC is triggered by our PWM timer. Moreover, since we measured directly mean current, the rest of the code is just control codes. With that controller, our controller closes the current loop, and controls the average armature current.

## 7) Component Selection

- Maximum Voltage on IGBT: 225V      -Maximum Current on IGBT : 15A
- Maximum Voltage on Rectifier: 225V      -Maximum Current on Rectifier: 32A
- Maximum Voltage on Diode: 225V      -Maximum Current on Diode: 15A
- Maximum Voltage on Filter Capacitor: 225V

According to the maximum ratings, following components are selected. Their ratings are enough for our application.

**IGBT:** [IXGH24N60C4D1 N Channel](#)

**Diode:** [DHG30I600PA](#)

**Rectifier:** [35A 1000V Bridge Rectifier](#)

**Filter Capacitor:** [PKL5-400V221MN400](#)

**Cement Resistor:** [PRW05WJW10KB00](#)

**Optocoupler:** [TLP250 DIP-8](#)

**Voltage Regulator:** [7805CV](#)

For 3 Phase rectifier, in order to use available components in power electronics laboratory, we used 2 single phase rectifier module.

## 8) Thermal Calculations

In this part of the report, the thermal analysis of the design is displayed. The main losses in the design are caused by the semiconductor devices. The most significant losses occur in a semiconductor while it is in conduction mode or it is switching. One can find the approximate total loss by adding the conduction loss and switching loss as shown in Equation 4.1.

$$P_{loss} = P_{conductio} + P_{switching}$$

*Equation 4.1*

The conduction losses can be calculated by two methods as shown in Equation 4.2 and Equation 4.3. In the first method, one can use the forward voltage drop of the semiconductor ( $V_f$ ) to calculate the power dissipation. Another possible method is to use the on-resistance value ( $R_{ds}$ ) indicated in the datasheet.

$$P_{conduction} = V_f \times I_{on} \times (\text{Duty Cycle})$$

*Equation 4.2*

$$P_{conduction} = R_{ds} \times I_{on}^2 \times (\text{Duty Cycle})$$

*Equation 4.3*

Switching losses can be calculated as shown in Equation 4.4 by using the turn on energy ( $E_{on}$ ) and turn off energy ( $E_{off}$ ) dissipated while the diode is switching . If these parameters are not indicated in the datasheet, the switching loss can be estimated by using the turn on delay time ( $t_{rise}$ ) and turn off delay time ( $t_{fall}$ ) as displayed in Equation 4.5.

$$P_{switchin} = (E_{on} + E_{off}) \times f$$

*Equation 4.4*

$$P_{switchin} \approx I_{on} \times V_{on} \times (t_{rise} + t_{fall}) \times f$$

*Equation 4.5*

For the rectifier block, two single phase bridge rectifiers (KBPC3510) are combined to construct a three phase full bridge rectifier. The switching energy or delay times are not indicated in the datasheet of the component. The maximum  $V_f$  is indicated as 1.1V in the datasheet. In maximum load case, the average current passing on one diode is 11 A when the diode is in conduction mode. One diode's duty cycle is 1/6. There are 4 diodes that are working in one rectifier while only 2 diodes are working in the other.

$$P_{loss-rectifier} = (1.1V) \times (11A) \times \left(\frac{1}{6}\right) \times 4 = 8.07W$$

$$P_{loss-rectifier} = (1.1V) \times (11A) \times \left(\frac{1}{6}\right) \times 2 = 4.03W$$

The diode used in buck converter is DHG30I600PA. The diode's conduction loss depends on both on-current and duty cycle. The maximum current is 11 A but the conduction loss is low because the duty cycle is 0.1 at that current. Thus, the maximum conduction loss occurs when duty cycle is 0.5 and the current is approximately 6A. Since the switching energy dissipation is not indicated in the datasheet, the Equation 4.5 is used to calculate switching loss. The switching loss is maximum when the duty cycle of the switch is maximum (0.9) since the current is maximum (0.44W). However, the conduction loss is more dominant than switching loss for this application in general. Thus, the worst case is calculated for 0.5 duty cycle.

$$P_{loss-diode,conduction} = 2.27V \times 6A \times 0.5 = 6.81W$$

$$P_{loss-diode,switching} \approx 6A \times 202V \times (40 \text{ ns}) \times (5 \text{ kHz}) = 0.24W$$

$$P_{loss-diode} \approx 7.05W$$

In IGBT, the maximum switching loss and conduction loss occur when duty cycle is 0.9. The calculations are displayed below.

$$P_{loss-IGBT,conduction} = (1.5V) \times (11.2 A) \times 0.8 = 13.44W$$

$$P_{loss-IGBT,switching} = (0.3mJ + 0.4mJ) \times (5kHz) = 3.5W$$

$$P_{loss-IGBT} = 16.94W$$

To find the maximum thermal resistance of the heatsink, the following two equations are used.

$$R_{th-needed} = R_{jc} + R_{th-heatsink}$$

*Equation 7*

$$R_{th-needed} = (T_{max} - T_{ambient}) \div (P)$$

*Equation 8*

For rectifiers;

$$R_{th-needed,rectifier1} = (150 - 25)C^o \div (8.07) = 15.45 C/W$$

$$R_{th-heatsink,rectifier1} = 15.45 \text{ C/W} - 2 \text{ C/W} = 13.45 \text{ C/W}$$

$$R_{th-needed,rectifier} = (150 - 25)C^o \div (8.07) = 31.02 \text{ C/W}$$

$$R_{th-heatsink,rectifier2} = 15.45 \text{ C/W} - 2 \text{ C/W} = 29.02 \text{ C/W}$$

For diode (buck converter);

$$R_{th-needed,diode} = (125 C^o) \div (7.05) = 17.73 \text{ C/W}$$

$$R_{th-heatsink,diode} = 17.73 \text{ C/W} - 0.7 \text{ C/W} = 17.03 \text{ C/W}$$

For IGBT;

$$R_{th-ne,IGBT} = (125 C^o) \div (16.94) = 7.38 \text{ C/W}$$

$$R_{th-heatsink,IGBT} = 7.38 \text{ C/W} - 0.8 \text{ C/W} = 6.58 \text{ C/W}$$

In the implementation phase, we used a huge heatsink (thermal resistance is unknown but it is very low) and connected all of these semiconductor devices to this heatsink. Moreover, we used a fan to improve our thermal parameters.

## 9) Testing

After completing the implementation, we set up a test circuit, we connected a R-L load, then using the thermal camera we observed the thermal ratings of components. In this setup the load current was about 9A.



**Figure 9.1: Thermal Camera Results**

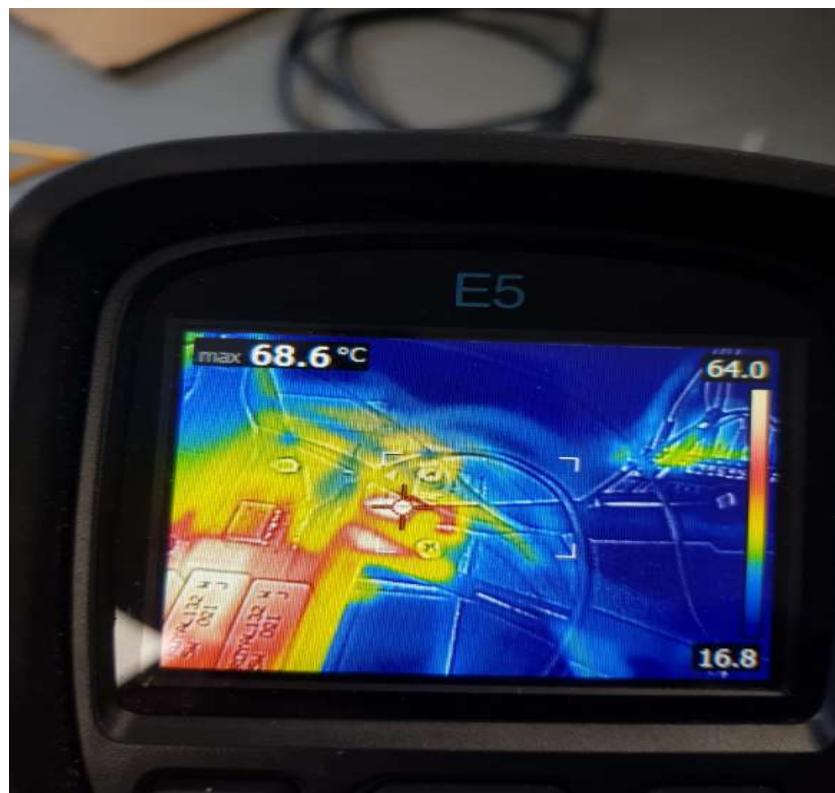


Figure 9.2: Thermal Camera Results

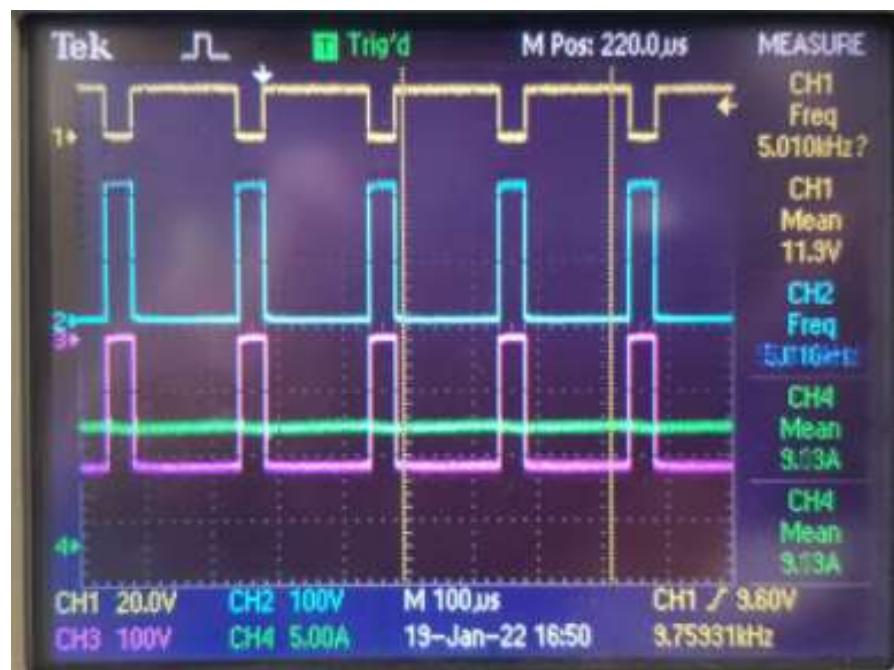


Figure 9.3: Test Setup Waveforms

## 10) Demo Day

In demo day, first we connected our circuit to the motor with no load, after a while motor is loaded with a kettle. Output power was 1.9kW with kettle load. We observed that loading the motor decreases the speed of the motor. Our efficiency is 97.3%.

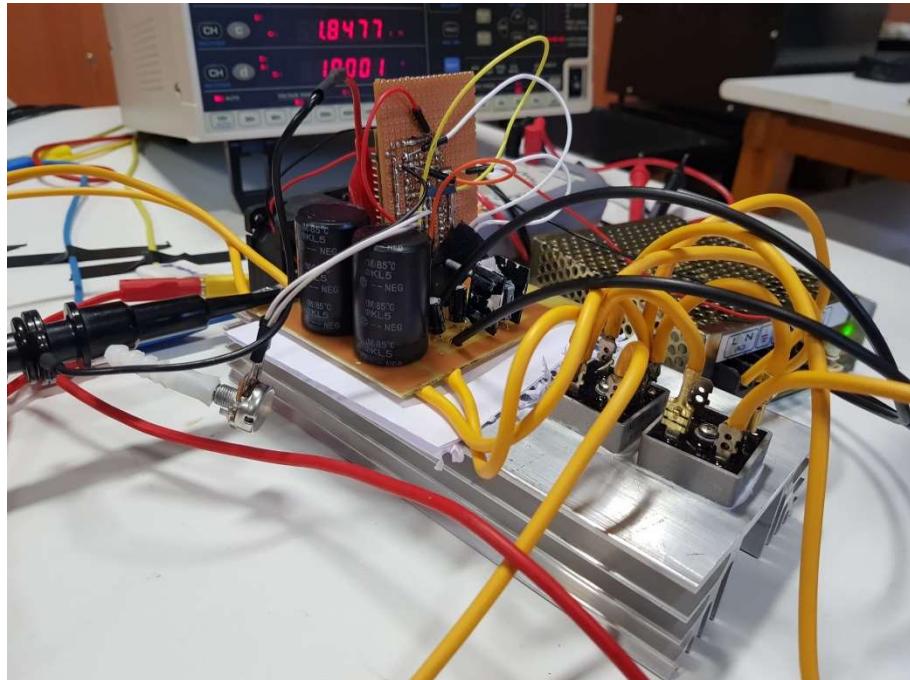


Figure 9.4: Demo Setup

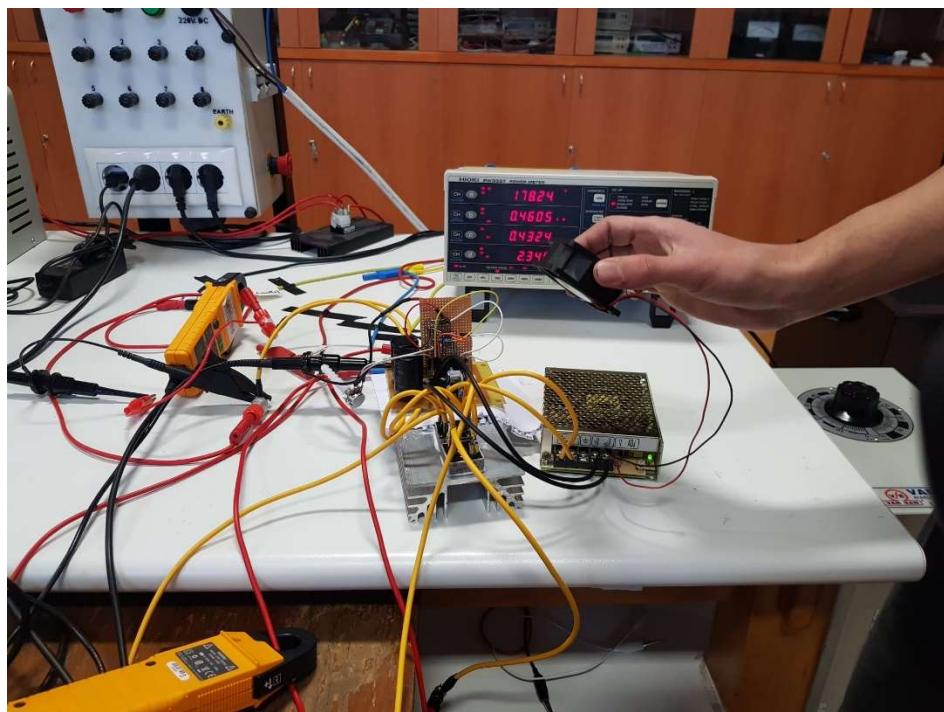


Figure 9.5: Demo Setup

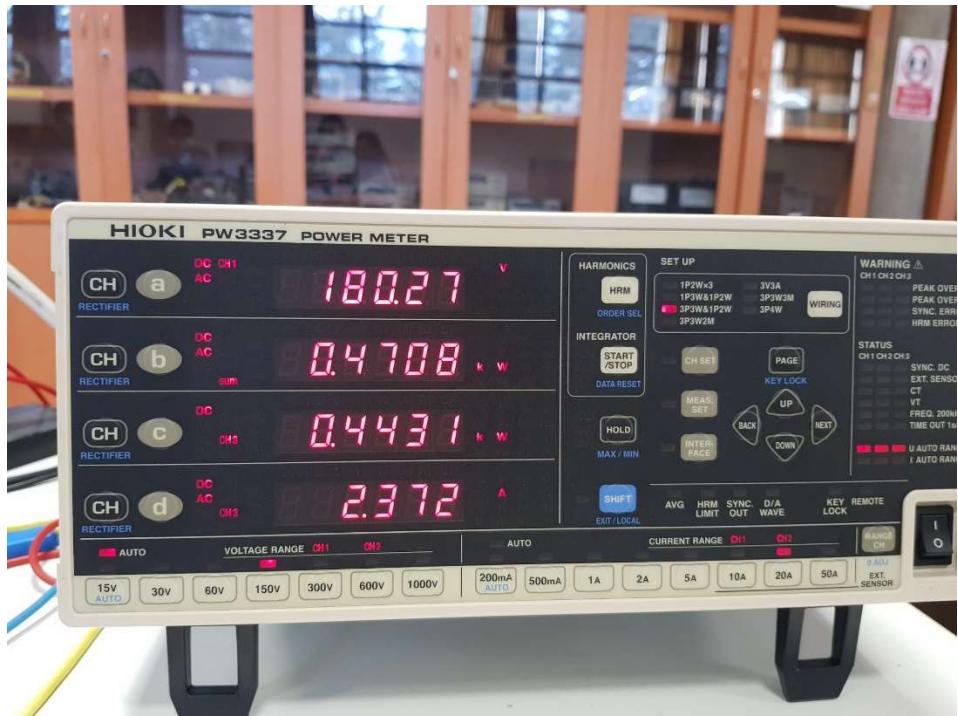


Figure 9.6: No Load Values



Figure 9.7: Kettle Load Values



**Figure 9.8: Full Load Kettle – Thermal Camera Vision of Circuit**

## 11)Conclusion

In this project, the aim is to design a circuit that converts ac input to controllable dc output to drive the specified motor with variable voltages. The three-phase full bridge rectifier plus buck converter topology is selected. First, the simulations made for this topology. Several modifications are made according to the simulation results. First, the switch is moved to the low-side to simplify the gate topology. Moreover, the closed-loop feedback is applied to set the current limit 13A. The gate driver and controller topology are designed with a microcontroller. After the thermal calculations and PCB designs are made, the circuit is implemented. Finally, the circuit is tested with both R-L load and motor.

## 12) Appendices

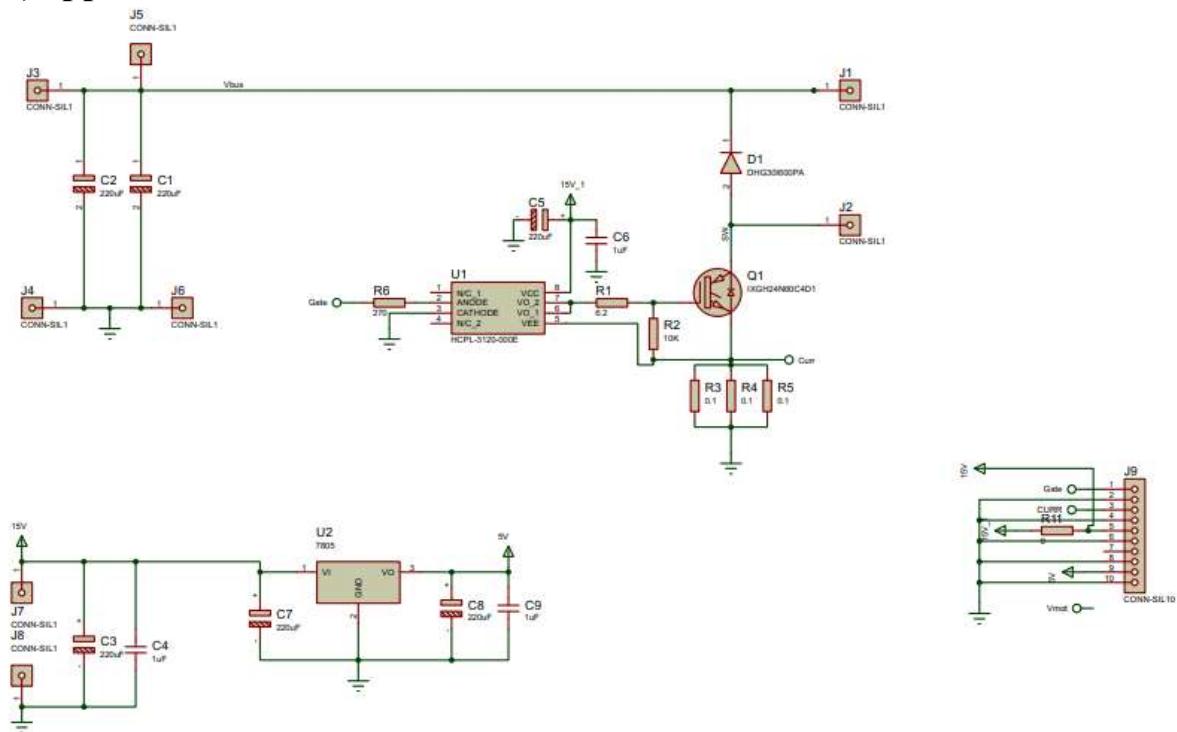


Figure 12.1: Schematic of Power Stage

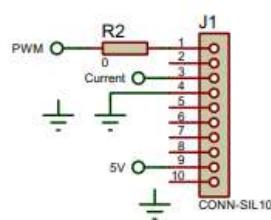
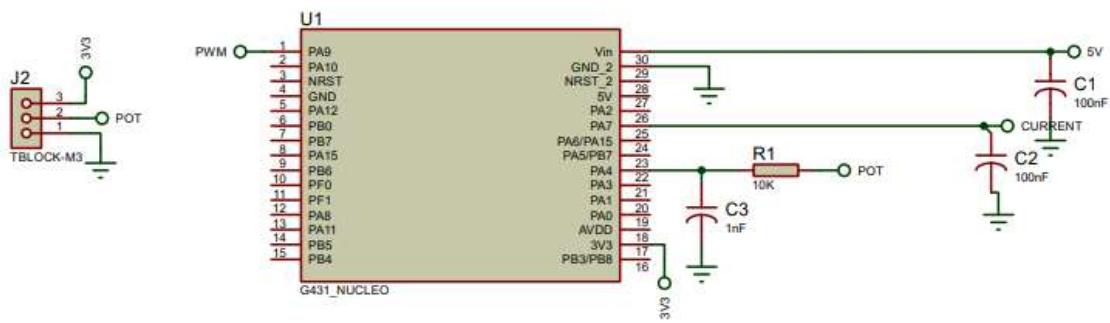


Figure 12.2: Schematic of Control Board

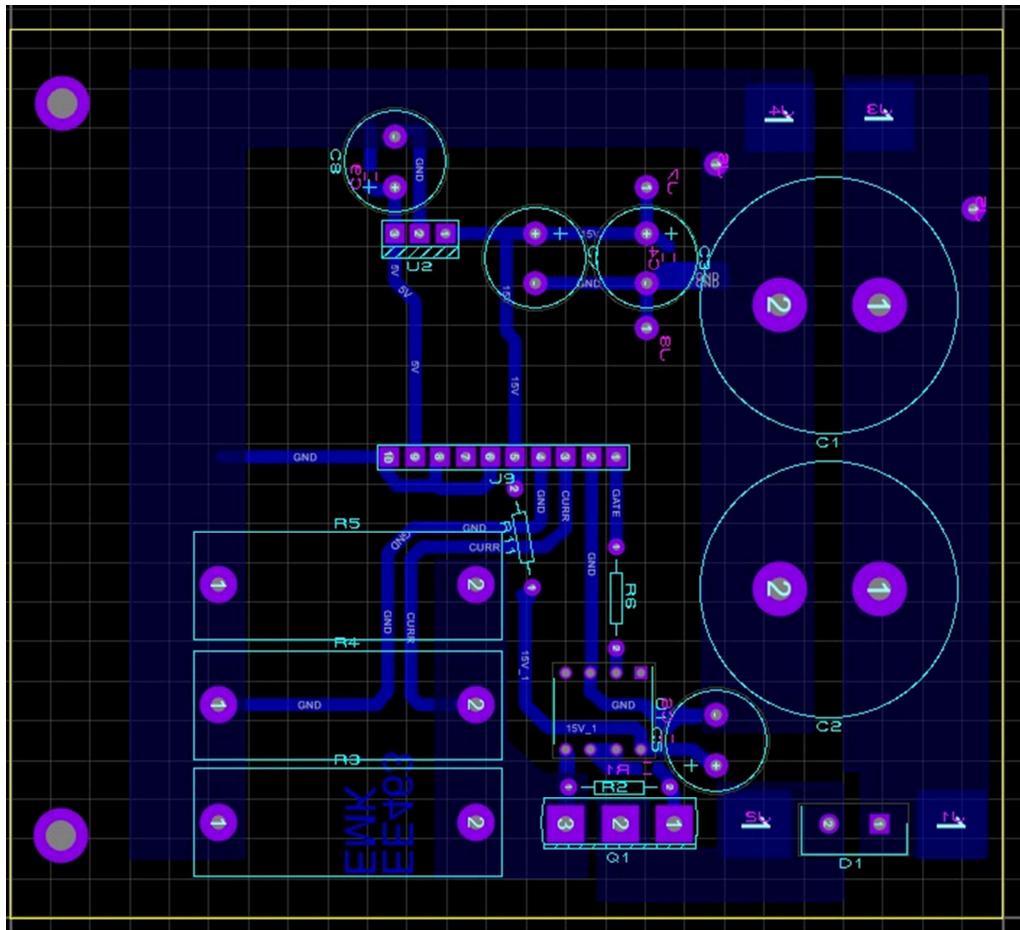


Figure 12.3: PCB Design of Power Stage

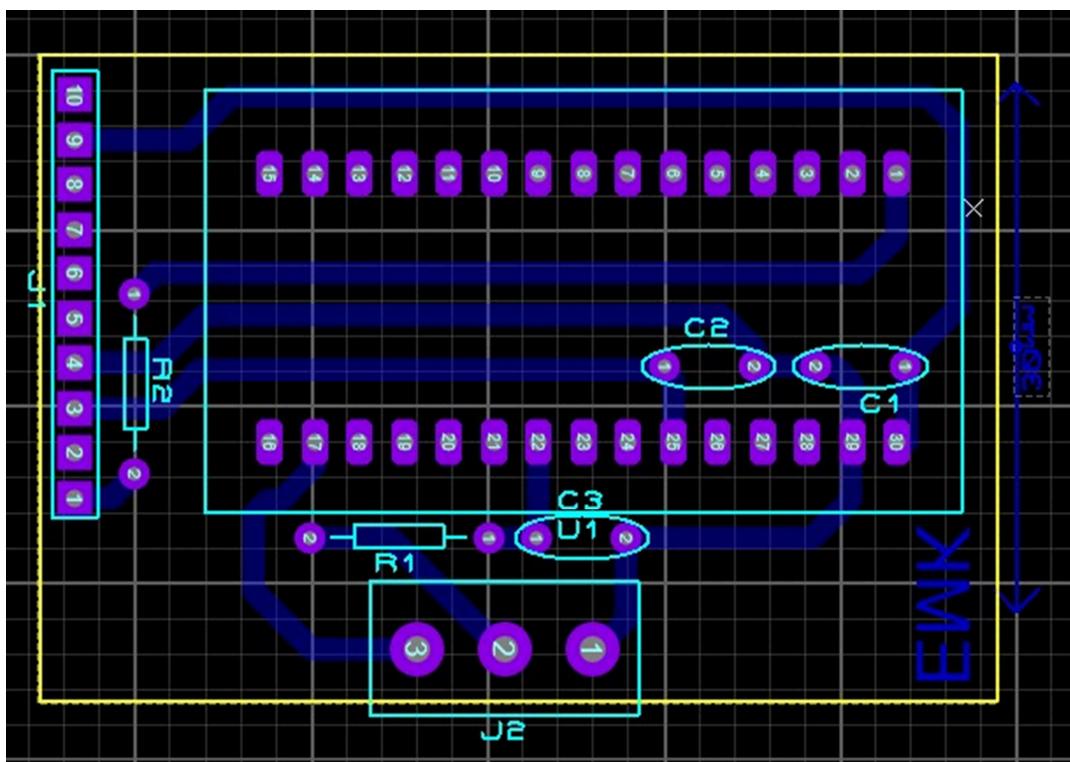


Figure 12.4: PCB Design of Control Board

## **Bonus**

- 1- PCB Bonus
- 2- Single Supply Bonus
- 3- Closed Loop Bonus
- 4- Tea Bonus