University of Idaho

709 Deakin Avenue, Moscow, ID 83843

Friday, May 08, 2015

Dr. Normann Fisher
Schweitzer Engineering Laboratories, Inc (SEL)
2350 NE Hopkins Court
Pullman, WA 99163

Dr. Fisher,

Attached, kindly find Team Power Team's final report on the Doubly Fed Induction Generator Simulation and Control project sponsored by SEL.

Best regards,

Team Power Team

Doubly Fed Induction Generator Simulation and Control

Authors:
Drew McKinnon (mcki7291@vandals.uidaho.edu)
Andy Miles (mile4558@vandals.uidaho.edu)
Tiras Newman (newm6560@vandals.uidaho.edu)
Cody Swisher (Swis4861@vandals.uidaho.edu)

Contents

Executive Summary

Currently, there are little data on the response of Type III and Type IV wind turbine generators (WTG) to power system faults. Over-current detection methods, which are the most common form of fault detection, are insufficient to detect when faults occur on systems with Type III/IV WTGs. Our team plans to update an existing RSCAD model of a Type III WTG power system and convert the model to a 690 V system and a 208 V system. The model will simulate a doubly-fed induction generator (DFIG), which is similar to a Type 3 wind turbine, and study the effects of a fault on a model power system. We will also help to develop physical controls that will mimic the proprietary controls that govern the behavior of the Type III WTG. Part of our project will be to take previous work done by prior teams and further the development of the physical and model controls for the physical DFIG that will be used for fault simulations.

Background

Harnessing wind power may to be the energy of the future. With new government subsidies in place, building new "green" energy production facilities such as solar or wind installations, has become a higher priority over traditional generation from fossil fuels such as coal or natural gas. One of the most popular of these "green" production methods is the harnessing of wind energy. Today, there are wind farm installations all over the country with thousands of wind turbine generators producing energy. These energy installations are not only a critical part of our countries infrastructure; they represent a huge monetary investment. According to Energy.gov, in 2012, “wind energy is now the fastest growing source of power in the United States - representing 43 percent of all the new U.S. electric generation capacity in 2012 and $25 billion in new investment.” (http://energy.gov/articles/americas-wind-industry-reaches-record-highs) Therefore protecting these wind turbines from damage and keeping this generation in production under varying conditions has become a concern.

Nearly all new megawatt-scale wind power plants being developed employ either variable-speed doubly-fed asynchronous (Type III) or full converter-based (Type IV) wind turbine generators (WTGs). These two WTG types can produce energy over a wide range of wind speeds, allow for fast and independent control of active and reactive power, limit fault current, and comply with low-voltage ride-through (LVRT) requirements set forth by industry regulatory agencies.

The first and second generation wind turbines (Type I and II), which are basically synchronous generators (Type I = squirrel-cage configuration and Type II = wound rotor configuration) are mostly being phase out and do not have protection issues. For our project we will be looking at the Type III WTG.

Problem Definition:

The newer types of wind turbine generators (WTG), Type III and Type IV, are modeled as Doubly Fed Induction Generators (DFIG). The major difference between Type I/II WTG and Type III WTGs is the addition of a power electronics circuit between the output of the generator (Grid/Stator side) and the Rotor windings (Rotor side).

These power electronics allow variable frequency AC excitation on the rotor circuit basically controlling the currents applied to the rotor. With these controls, not only can the magnitude of the rotor currents be adjusted, the phase angle can also be adjusted. Due to the controls being able to adjust the rotor currents magnitude and phase angle, these WTG's have separate controls over the real and reactive power that they are producing or consuming. By under or over exciting the generator, the WTG can be operated as a synchronous generator with a large range of RPM. This effectively allows the Type III WTG to produce energy during winds that are +/- 50% of synchronous speed which is a huge improvement in wind range over Type I or II WTG. These controls are implemented using back to back, AC/DC, current-regulated, voltage-source converters which act as a control for the output of the WTG. Because of these controls, the responses of these Type III/IV WTGs to faults on the power system are not completely understood.

For faults near Type I/II WTG terminals, the fault current produced by these WTGs can be several times (sometimes 5-6 times) the rated full-load current, only limited by the fault and the WTG impedances. The fault current characteristics for Type I and Type II WTGs are well understood and accurately represented in most commercially available short-circuit analysis tools used by protection engineers. For faults on systems with Type III and Type IV WTGs, have much more complex fault current characteristics. These characteristics are governed by the proprietary controls of the converters used in these generators and are not readily available for analysis. Because of the controls added to these generators, during fault conditions, Type III WTG's tend to behave like a current source evening out the current making the over-current detection scheme (the most common form of fault detection) unreliable in detecting faults. For Type III and Type IV WTGs, the fault current contributions are usually limited to 1.1 to 2.5 times the rated full-load current, following any transients. Not only are these over-current oscillations small, because of the controls circuit, they also tend to dampen out quickly. Due to the lower fault currents and quick dampening of the oscillations, detecting these faults with conventional over-current relays are difficult or impossible. Also, as normal operation, the system may see spikes of currents in the 1.1-2.5 range that should not lead to circuit breakers tripping. When the over-current relays are adjusted to detect the over-currents produced by the Type III WTGs, often, these relays start tripping their circuit breakers at minor disturbances in the system, otherwise known as Nuisance Tripping.

Project Plan

The first semester of our project consisted primarily of defining the problem and learning more on the subject. We spent a great deal of time working through tutorials and trying to figure out how to use the RSCAD software. Since we initially planned to focus primarily on fault detection and protection schemes, we also did a lot of research on fault detection techniques like line current differential.

We assigned roles and responsibilies:

Roles and Responsibilities

- Cody Swisher will be in charge of the Budget.
- Tiras Newman will be in the primary client contact.

- Andy Miles will organize the team meetings.
- Drew McKinnon will be in charge of keeping all documentation.
- Roles will be selected on a volunteer basis.

Since our end goal changed rather substantially around the start of the spring semester, we made a Gantt chart to outline our schedule for the semester. This guideline can be found in Figure 1 below.

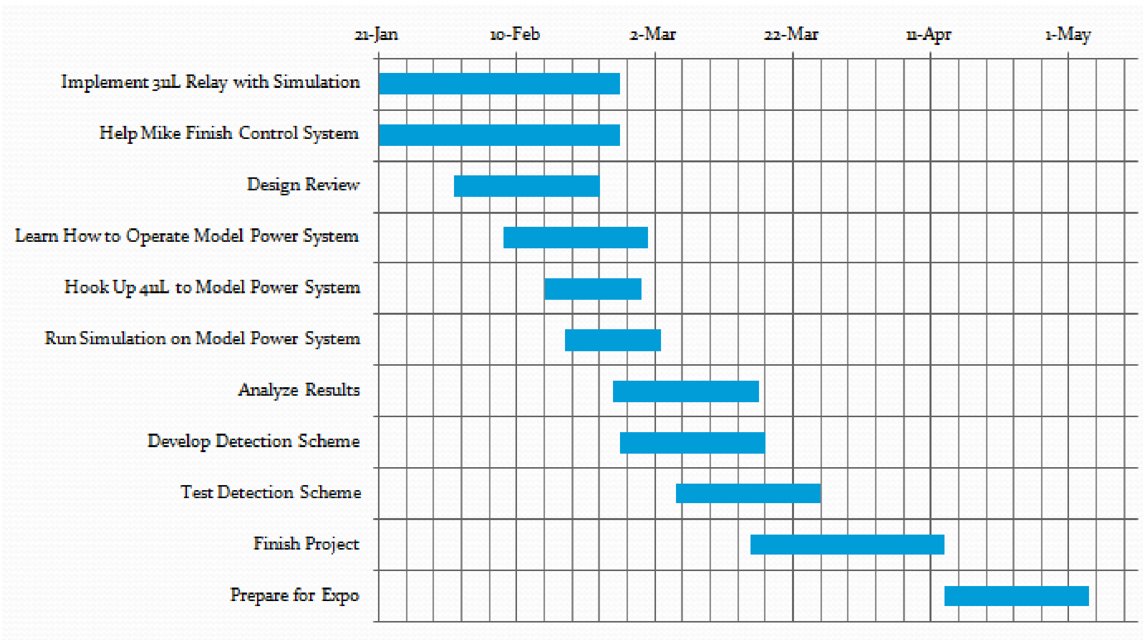

Figure 1: Spring Semester Timeline

Concepts Considered

Since our project did not involve purchasing any equipment, the concepts we considered consisted primarily of what methodology we wanted to use to develop the RSCAD model and physical DFIG controls system to fit the guidelines set forth by our client.

RSCAD Modeling

Before testing the effects of faults on the DFIG, we will first model the system in **RSCAD** software and run simulations using the **RTDS** system. This will give us a better idea of how the system will react to various types of faults before we do physical testing using the model power system. The simulations are carried out in a RSCAD Runtime file. An example of a runtime output can be seen in the figure to right.

The RSCAD model and runtime will also allow us to verify the accuracy of the transient and steady state behavior of our DFIG. We have parameterized the internal impedances of the physical DFIG and have input these values into our model in an attempt to achieve a nearly identical response.

The main decision we needed to make when looking at creating the RSCAD model of the DFIG and transmission system was what voltage we should be generating. To accurately model a typical wind turbine, a 690V generation model would be best for accurately testing fault conditions. On the other hand, a 208V generation model would allow us to better model the physical DFIG.

Model Power System

Once we have a good idea of what to expect, we plan to connect the DFIG to the model power system and test faults in different locations throughout the system. This should verify or disprove the validity of our computer simulations

Physical Generator

Once we have our RSCAD model up and running, we hope to create a working control system for a phycical DFIG located next to the Model Power System. Once we have our DFIG operating properly, we can apply faults to the system and verify that both our simulation and physical setup agree with each other.

SEL Relays

If time permits, we would like to design a detection/protection scheme using SEL relays. We are currently doing research on which relay will allow us to detect and protect faults most effectively. We are currently looking at two different relay options: the **SEL 421** and the **SEL 411L**. Both have a variety of detection and protection technologies that will help us solve the problem. Once we determine which detection technology will work best for our project, we should be able to make a definitive choice of which relay we plan to use

Physical DFIG Controls System

To complete the controls system for the physical DFIG, we began with existing hardware built by a prior senior design team. The current system has been outlined in detail in Appendix A at the end of this report. Our primary focus was to build on the current system, mainly by implementing the microcontrollers that would eventually control the system.

To help us program the microcontrollers, we were given a Simulink model of the complete controls system developed by Tim Lendberg. To come up with a final solution to this problem, we considered two different alternatives: using a Simulink add-on to convert the existing Simulink model into C-code directly or using the Simulink model as a reference to create a program from scratch.

In an attempt to avoid programming the complex program into the microcontroller from scratch, we decided to first attempt to use a Simulink add-on to convert the Simulink model we were given directly to C-code. However, since this add-on was not designed specifically for our Microchip microcontroller, the syntax was not correct and we received several pages of error messages when we tried to compile the code converted by the Simulink add-on. Since this proved to be a problematic method, we began to look into the second method.

Since using a coder to directly convert from Simulink to usable C-code would not work, we decided to look at the project in a different light. We realized that the Simulink model we were looking at actually contained programming that modeled the controls hardware that we already had in place. This meant that not everything in the complex Simulink model would need to be implemented into the final C-code that would be written for our microcontroller. This reduced the scope of our project substantially. At this point, we felt that trying to produce usable code would be too much of a hassle, even if we were to remove the unnecessary pieces from the Simulink model. We decided that programming the microcontrollers from scratch, using the Simulink model as a reference, would likely be easier and would give us a better understanding of how the program operated.

Concept Selection

RSCAD Modeling

Our client wanted us to model a standard 690V to 33kV to 345kV transmission system to allow for robust protection scheme testing. We began our RSCAD modeling of the DFIG system by looking at a model that had been built previously by a graduate student named Rishabh. In modifying this model we started out on the transmission line system modifying its rated voltages and currents to allow for the larger values. Working our way back we removed the collector bus and replaced it with an averaged pi model to simulate our DFIG being hooked right into the transmission system, assuming we had only one wind turbine. Lastly we began modifying the DFIG itself, this was done by taking the parameterized values from our physical DFIG and changing the generator parameters. Next we needed to modify the DFIG controls and adjust the gains so that our machine could sync to the grid enabling proper power output. See Appendix A for a troubleshooting guide to modifying the model.

We ended up creating two separate models: one model generates at 208V while the other generates at 690V. This will allow us to model both wind turbine generation and the generation of the DFIG set at the university.

Physical DFIG Controls System

After struggling with trying to convert the Simulink model into C-code that would be usable with our Microchip microcontroller, we finally decided on designing the C-code from scratch. We think that this solution, although much more difficult and time consuming, will provide a better outcome in the long run. Not only will it likely run smoother and contain less lines of code, it will also be easier to document and will be much easier to understand.

System Architecture

RSCAD Modeling

Two models were developed and verified: one generating at 208V and another generating at 690V. The finalized RSCAD model has been saved to a disc and can be viewed and used for simulation using RSCAD.

Physical DFIG Controls System

Since this portion of the project was incorporated relatively late in the school year, we did not get as much accomplished as we had hoped. The primary outcome of our looking at the physical DFIG controls system was the new understanding of the system. As previously stated, our outlining of the controls system can be found in Appendix B.

Future Work

RSCAD Modeling

The current model steps the 208V up to 33kV and will not produce the same results as the model power system. A future model that uses a transformer with 1:1 turns ratio will likely be needed in the future. The model power system currently runs at 208V, which is the voltage supplied by the generator. This modification would allow for more accurate testing of the physical DFIG controls on the model power system.

Physical DFIG Controls System

Although we did not accomplish all we had hoped to with the DFIG controls system, we feel that we have done a great job of documenting the current status for future teams that wish to pursue this project. We hope that future teams/students will have the opportunity to look deeper into how to implement a microcontroller into the current controls system hardware. This is the main limitation at the moment of the current system. There is currently a graduate student working on this task but if he does not finish the project, we feel that another graduate student with some basic microcontroller knowledge could finish the project relatively quickly. If the project were to be taken on by another senior design team, we feel that they should be able to finish the project within the course of two semesters.

Appendix A: Guide to Modifying Averaged_3.dft

NOTE TO USER

- **Rotor side is in OHMs, AMPs, VOLTS**
- **Grid side is in Per Unit**
- **Use "find" option to replace voltages and currents**

Transmission Side:

- Adjust bus voltages to respected line-line rated voltages
- Adjust transformer turns/voltages for respected line voltages
- Adjust infinite bus voltage to respected voltage

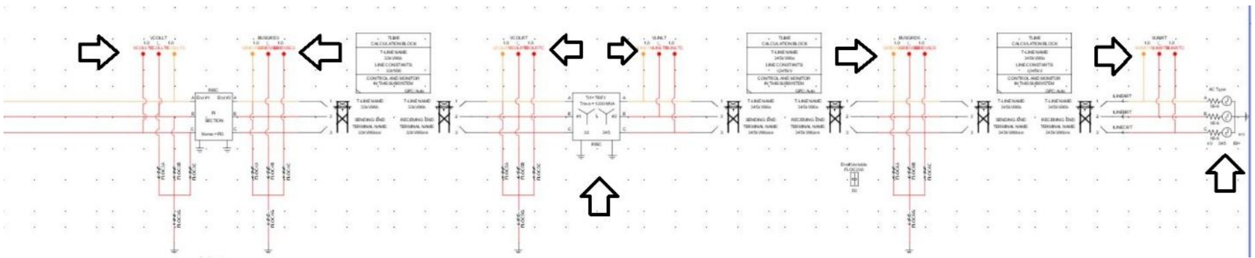

Figure 2: T-Line

DFIG:

Step 1: Voltages and Current Bases

- Adjust DFIG Rated Settings
- Adjust **Bus** voltages to rated L-L voltages *not L-N voltages (see figure 2.)
    - VTROT = 0.690kV
    - ROTVS = 0.690kV
    - DFIGTEA = 0.690kV
    - DFIGTR = 0.690kV
    - BUSGRD = 0.690kV
    - VTA= 0.690kV
- Adjust VSC rated L-N voltages (Blocks (1.878 – 0.563))
- (see Figure 2. Below)
- Run Simulation

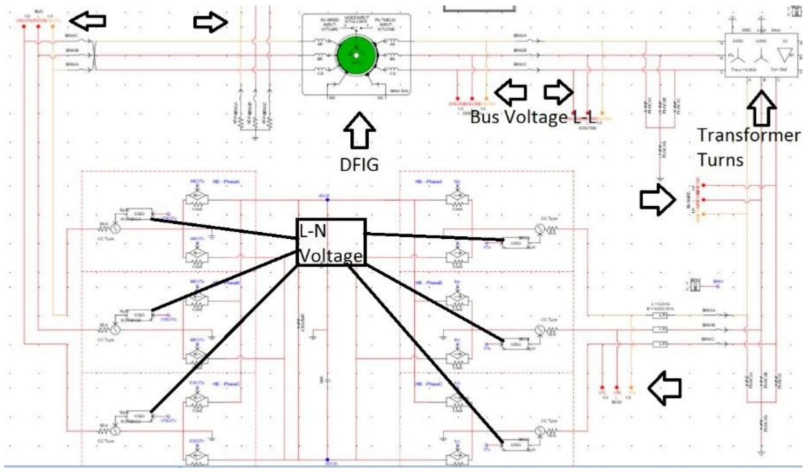

Figure 3: DFIG Voltage and Current adjustment

Simulation #1

- Check infinite bus voltage 230-345kv
- Check DFIG Voltages and currents
- Does it sync up? – (It shouldn't if on a new voltage)
- Currents should not match…that is ok

Step 2: PI Control

- On the Grid Side Current Control there is the Current Tracking look (ISd and ISq)
- Modify the P (proportional gain) values till the current syncs up with the grid.

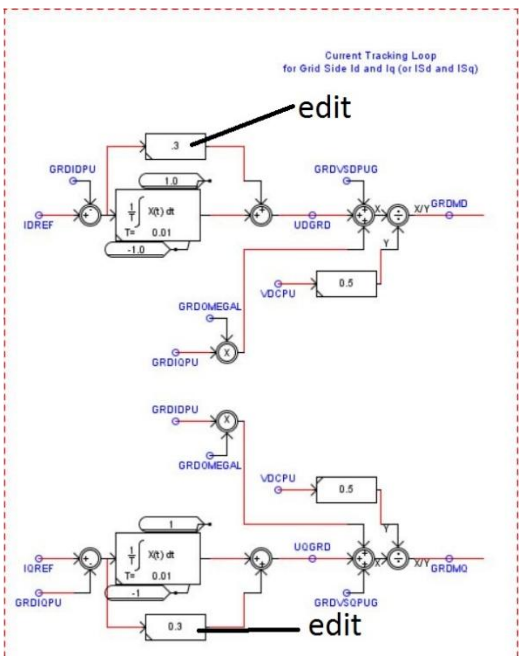

Figure 4: PI control gains

- Usually **Down Is better**

Simulation #2

- Check to see if current syncs up to grid (it should eventually when you get the right gain)

Inductance of the Machine and Grid

- This will be the most difficult gain to set
- I found the further away you go the more oscillatory it becomes
- Take a look at the VDCLINKPU meter – The magnitude should be increasing drastically – The closer you get to the right inductance – the slower the magnitude will increase
- As seen below there are two gains you need to set
    - Grid side inductance (edit first)

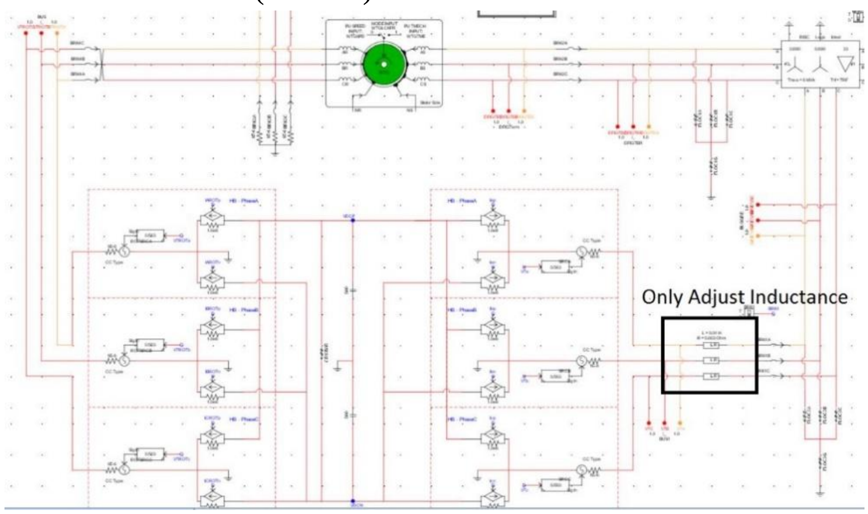

Figure 5: Inductance Grid

    - Grid Side Inductance **inside the Grid Side Current Tracking block**
    - They should be the same value

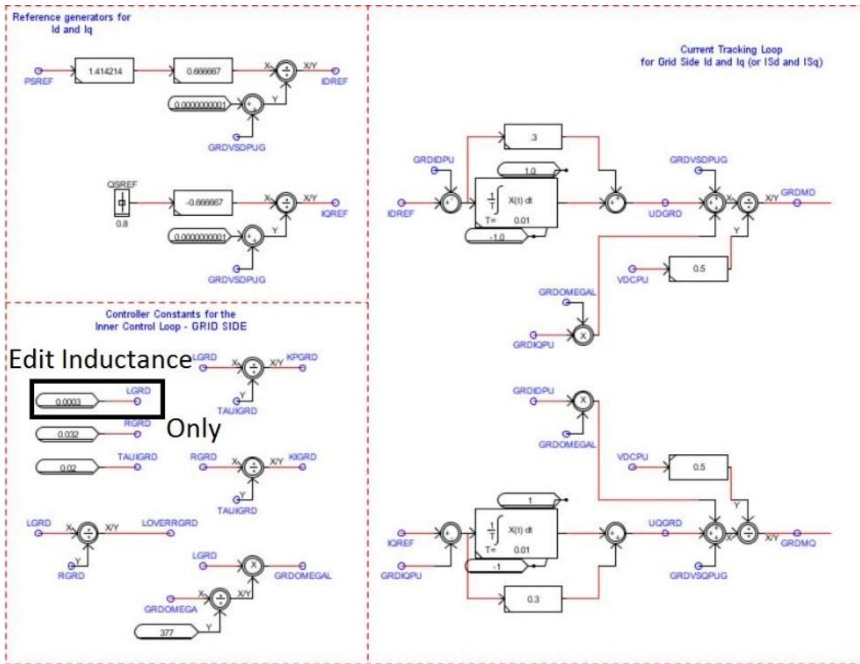

Figure 6: Inductance grid - Current Tracking loop

Troubleshooting Inductance

- We found that the less rated voltage the DFIG has, the less the inductance
  - 2.3kV - 730 A = 0.011H
  - 609V - 10.8 A = 0.0025H
  - 208V - 25.4 A = 0.0008H

Syncing to the grid

- If it is still not synced to the grid repeat varying the inductance till the current syncs to the grid

Adjusting Values

- Check the DFIG parameters
  - GRIDVSA = rated voltage (Line to neutral)
  - IDFIGa = rated current (Line to Line)
  - Ia, Ib, Ic = 1/70th of rated?

Troubleshoot Adjusting Voltages

- The voltages are set within the DFIG and the VSC converter
- They should have already been set earlier

Troubleshoot Adjusting Currents and Power Output

- The Currents and power can be adjusted within the current mode control (rotor side)
- Adjust the two constants (as seen in the figure below) to match your rated current from the machine
- Simulate and check that the current is at rated current

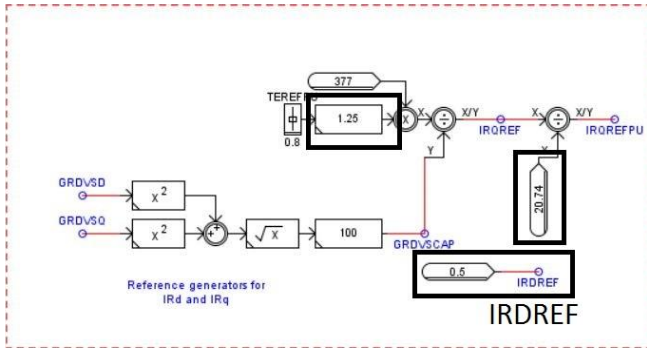

Figure 7: Reference Generators for IRd and IRq

Power Output

- To achieve the right power output on the stator
- Again look at the figure above and vary the gain till the poutstat is what you want (7.46kW) or a combination of P and Q
- The combination of P and Q of the stator can be changed a bit with the IRDref as shown in figure 6.
- Next take a look at the Pout and Qout of the machine
- The changes in Pout and Qout can be changed a little in the PI controller to the right of the reference generators for IRd and IRq (wont effect it too much) But can move it around a bit.

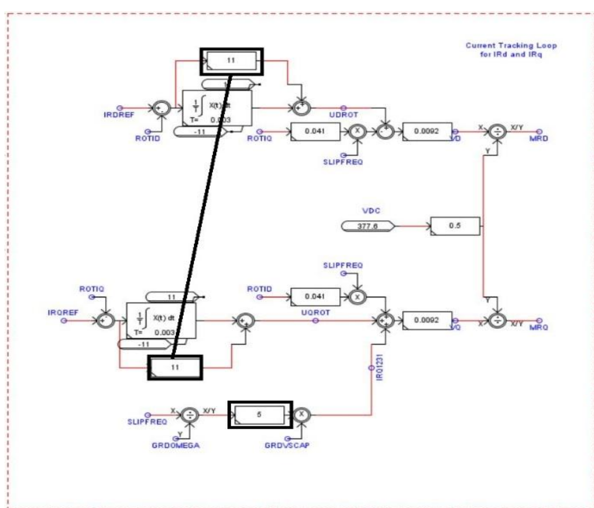

Figure 8: Current Control Rotor

Q&D check

- Bring up graphs for IRQREF
  - It should be in the same ballpark as your current
- Bring up UQROT
  - It should be in the same ballpark as your voltage (roughly)

Appendix B: Notes to Run Averaged_3.dft

Pre-Run

- If you have not used RSCAD before, I would recommend going through a couple of tutorials to familiarize yourself with the GUI and operating shortcuts.

Operation

- In the run mode take note of the breakers
    - BRK1: Grid Side VSC
    - BRK2: DFIG Stator side
    - BRK3: Used to Short the Rotor (leave open)
    - BRK4: Rotor Side VSC
    - BRK5: Collector Bus Breaker
- CBAT
    - Usually left off – for the crowbar logic
- ABCmag
    - Used for t-line voltage (345kV)
- WIND SPEED
    - Two options
        - WTGLCKFR: is for lock/free mode, meaning
            - WTGLCKFR-off and WTGSPD-on is steady state
            - WTGLCKFR-ON and WTGSPD-off. You control the speed of the machine
        - To control the speed you have two options
            - TEREFPU: mechanical torque on the rotor
            - WTGTME: adjusts speed as well

RUN

- To run the model in run time the draft needs to be compiled to a rack
- In runtime you need to select the proper graphs and set the proper breakers as noted above to the settings you prefer
    - BRK1-on
    - BRK2-on
    - BRK3-off
    - BRK4-on
    - BRK5-on
    - WTGLCKFR-off
    - WTGTME-on
- Then run the simulation and press refresh and it should be running.

Appendix C: Outlining of Physical DFIG Controls System

Our team has outlined what we have learned of the physical DFIG controls system thus far and have done our best to document the various pieces to help future students working on this project. Figure 10 and Figure 9 below show the whole of the controls system that we were given at the start of the project.

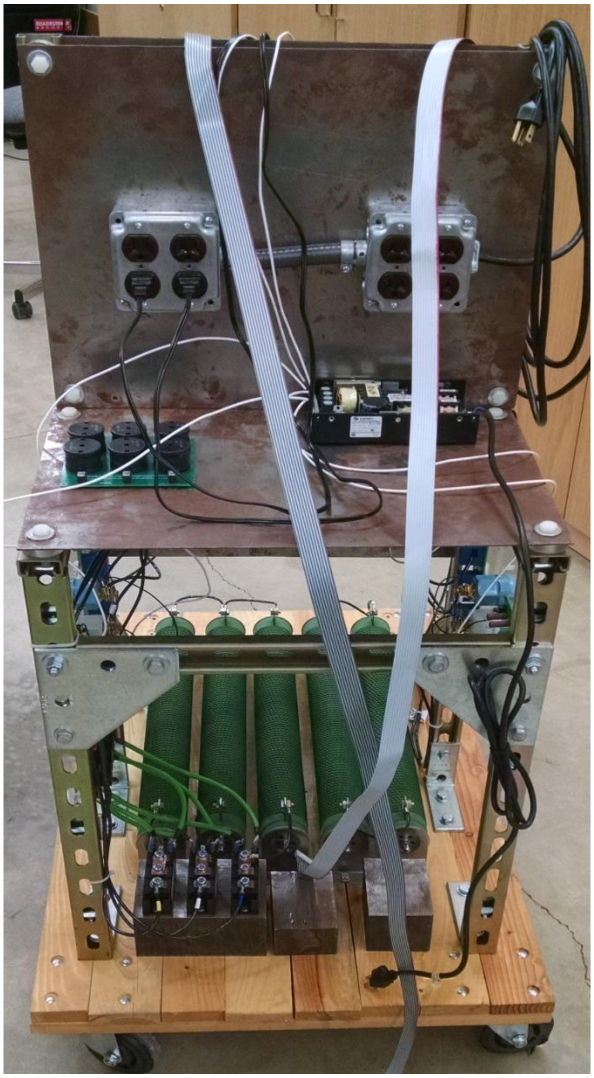

Figure 10: "Back" of DFIG Controls System

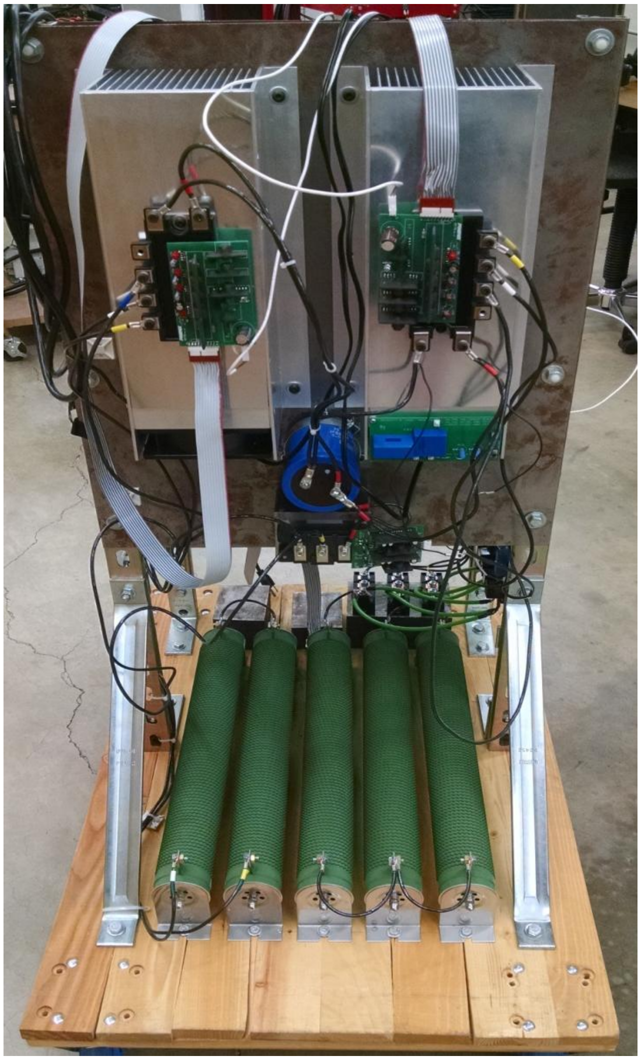

Figure 9: "Front" of DFIG Controls System

Our first step in documenting the system was to look at the schematic given to us by the last senior design group on this project and determine where the various pieces of the system were located. Figure 11 below shows the schematic we received of the controls hardware.

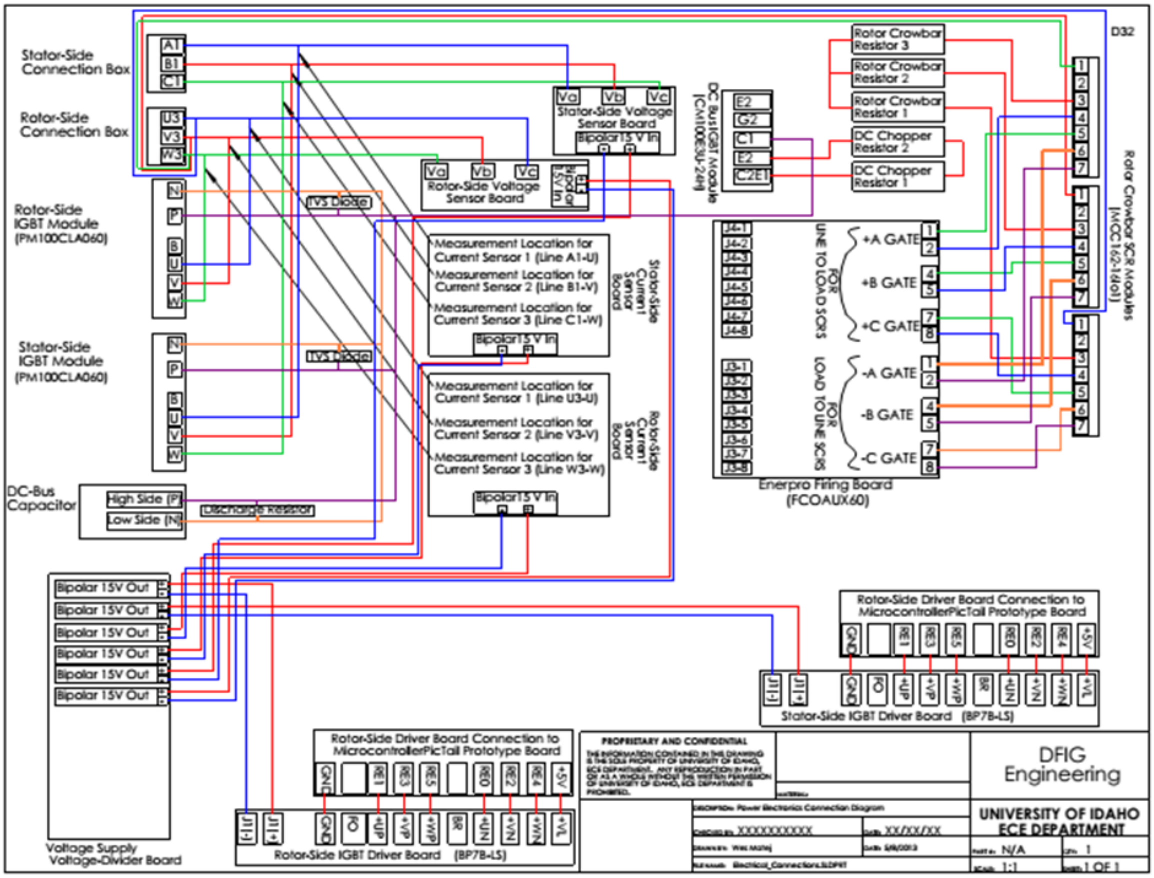

Figure 11: Schematic of DFIG Controls System Hardware

Once we examined the schematic, we cross referenced it with the physical hardware to analyze the data flow through the system. The system was lacking a good source of documentation when we were assigned to the project so we decided it would be beneficial to cross-reference the physical parts of the controls system with the schematic to make it easier for future groups that would be working on furthering the progress of the system design. We also felt that this would help determine what would be needed to program the microcontroller that will eventually run this system. The figures below give a detailed visualization of the different parts of the DFIG controls system that has been built thus far.

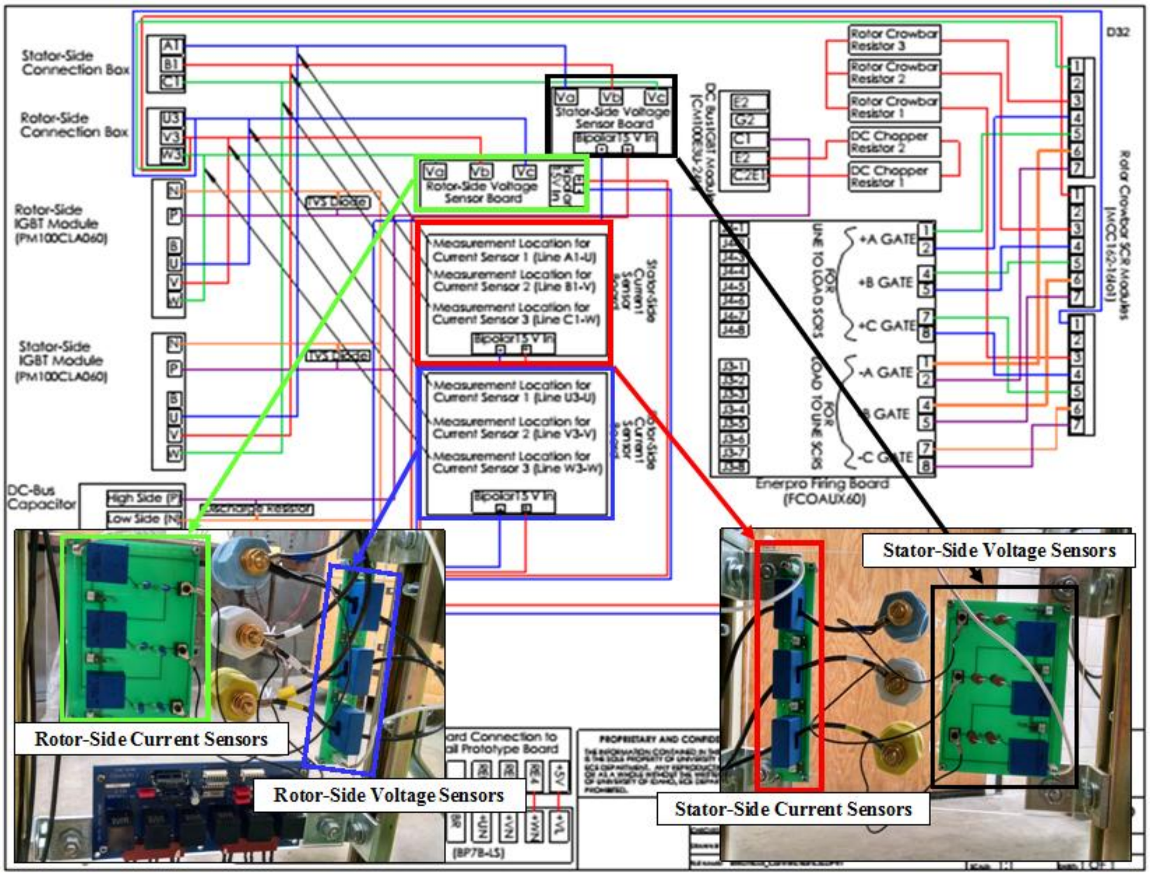

Figure 12: Rotor and Stator Side Voltage/Current Measurements

The rotor and stator voltages and currents are measured using voltage sensor boards and current sensor boards. The measurement data from these boards is sent to the microcontroller and manipulated in code. These boards are powered by the 15V DC power supply. The boards are currently not hooked up completely but have connection pins to send data to pins on the microcontroller.

The IGBT modules pictured in Figure 13 and 14 are currently missing a couple of connections, likely due to transporting the unit. The ribbon cables that communicate with the microcontroller will most likely need to modified in the future to fit the pins on the microcontroller.

When current stored in the rotor needs to be dissipated, the crowbar thyristors fire and allow current to flow through the crowbar resistors. These components can be found in Figure 16 below.

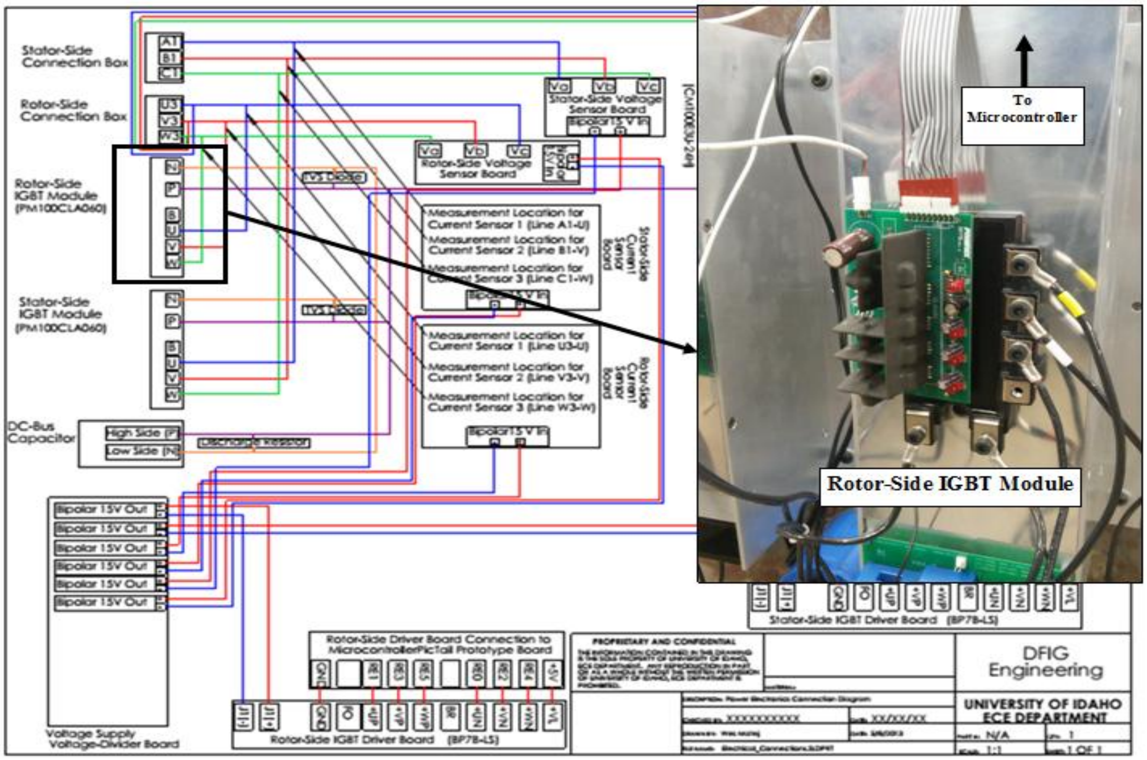

Figure 13: Rotor-Side IGBT Module

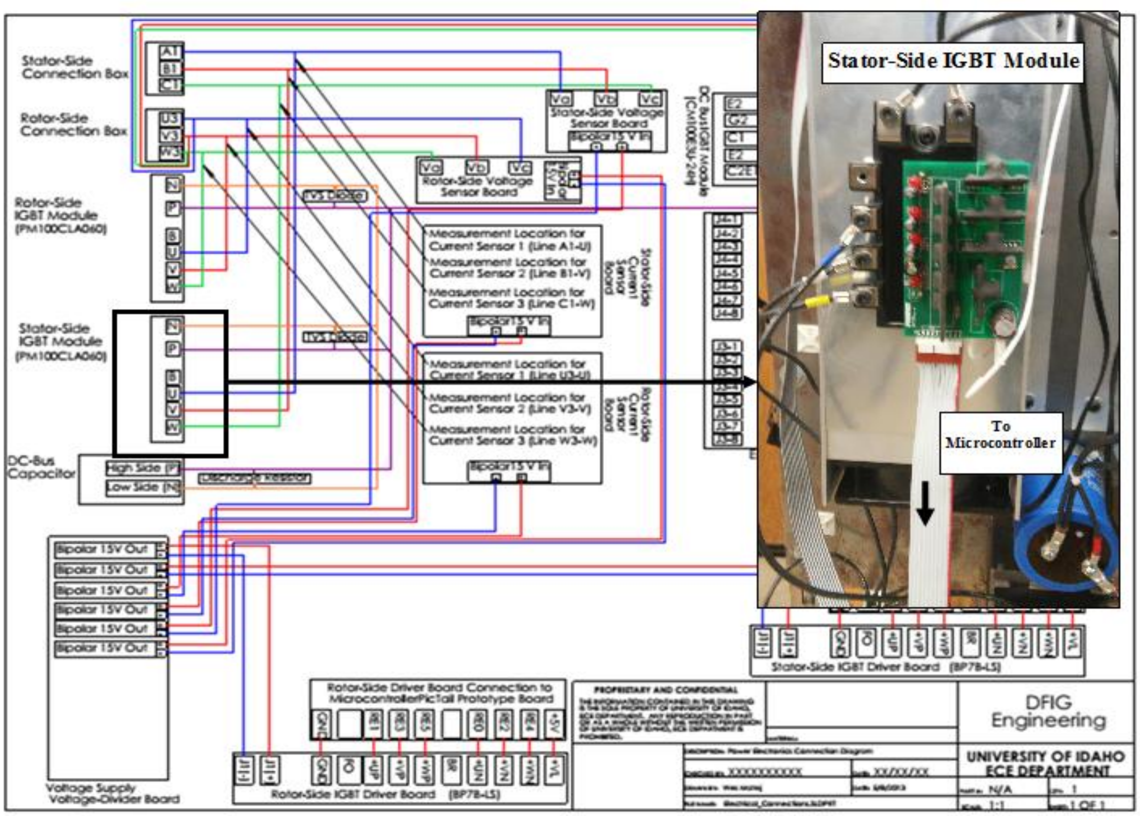

Figure 14: Stator-Side IGBT Module

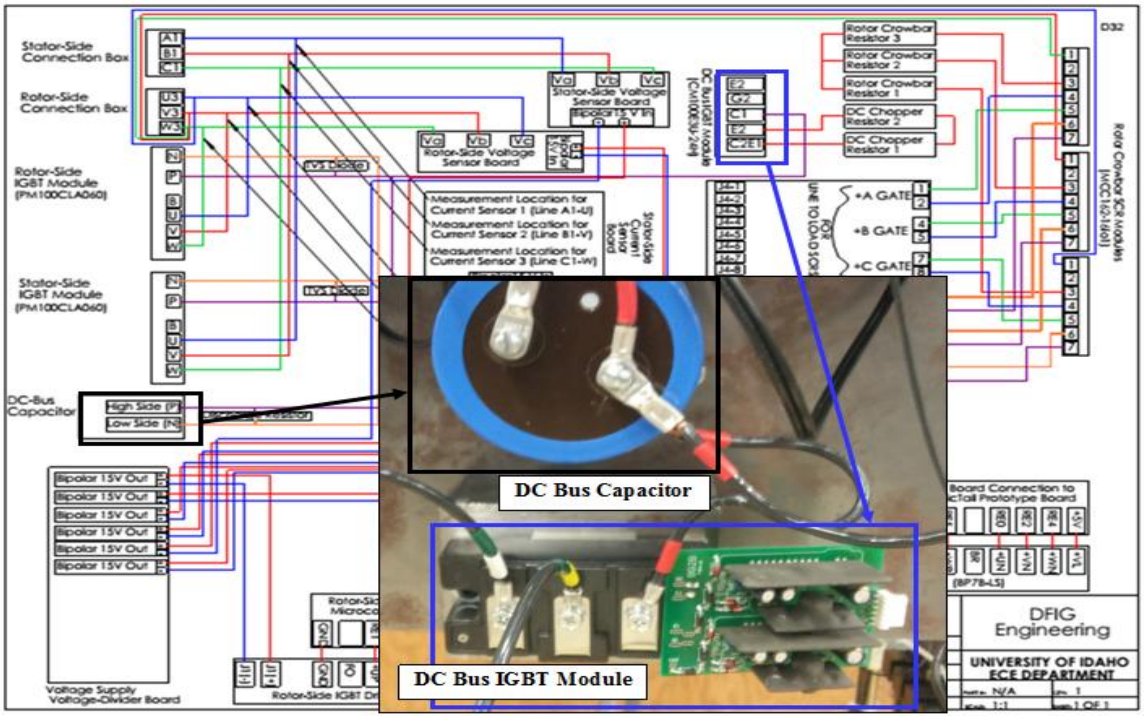

Figure 15: DC Bus Capacitor and IGBT Module

.

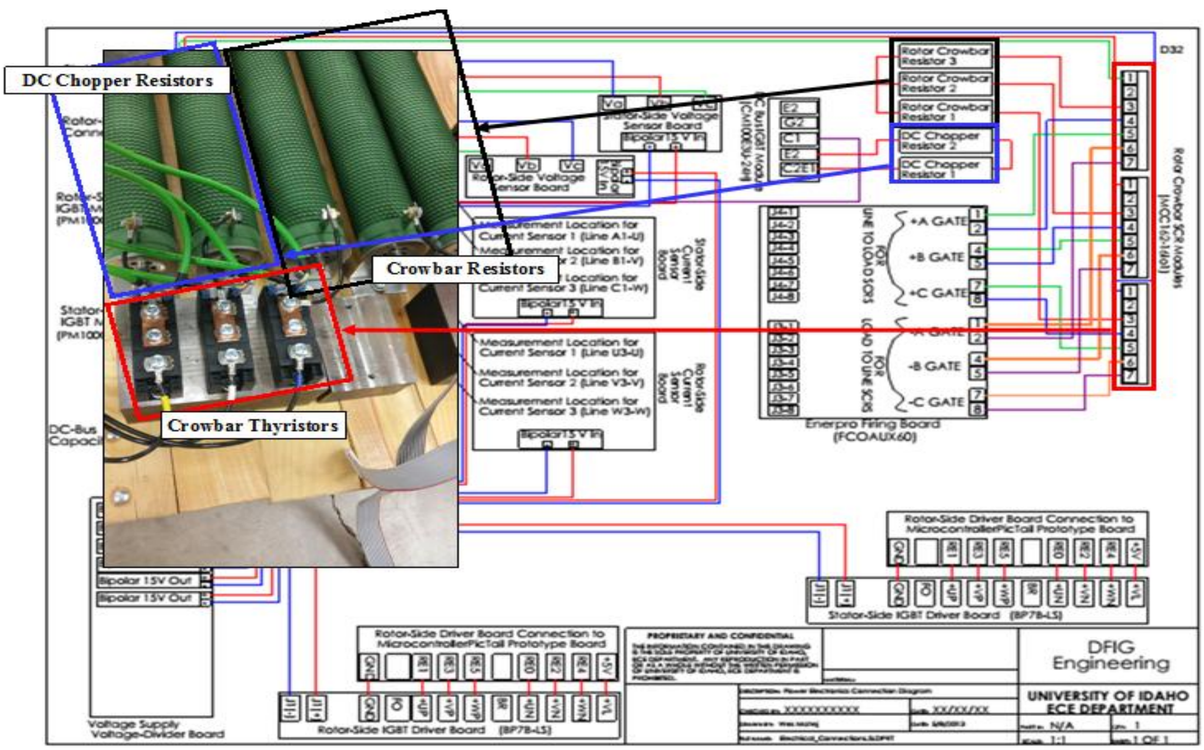

Figure 16: Rotor Crowbar Resistors, Crowbar Thyristors, and DC Chopper

List of Figures Used: