Analysis and Mitigation of Harmonics in Wind Turbine Transformers

A Major Qualifying Project
Submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
In partial fulfillment of the requirements for the
Degree of Bachelor of Science
in
Electrical and Computer Engineering

By:
Stephen Cialdea, Marcus Peart, and Warranyu Walton

April 2012

APPROVED:

_____

Professor Alexander E. Emanuel, Advisor

Abstract

The goals of this project are to study the effects of current harmonics generated by a Doubly-Fed Induction Generator (DFIG) and to create a reduced scale proof-of-concept of a single phase active filtering system designed to reduce non-fundamental current harmonics up to the 13^{th} harmonic. The DFIG study was done through modeling of the system using PSPICE. The active filter was then designed to measure non-linear current and generates a correction current to negate the non-fundamental harmonics via an H-Bridge circuit.

Acknowledgments

We would like to thank the following individuals for their support and guidance during the course of the project:

- Professor Alexander Emanuel for providing weekly advice on gathering background for the project, modeling case studies and building design blocks which were necessary for completing our final design.

- Three-C Electrical Inc. for presenting the issue of overheating in the transformers and for supplying us with their current measuring equipment.

- Princeton Municipal Power and Light for providing us with harmonic current readings produced by the transformers.

- Professor Stephen Bitar for his advice on design and analysis.

Executive Summary

This project dealt with the technology of wind energy. There are numerous popular turbine designs, but this project focused on the doubly fed induction generator (DFIG). This technology has significant market share in the emerging wind energy market due to its low maintenance and technical aspects.

However, there is a particular drawback to DFIG wind turbine systems. Because of the feedback loop used to maintain a constant 60Hz output frequency at variable wind speeds, a source of non-linear current is introduced. This non-linear current is passed through the step-up transformer and the eddy current losses associated with non-linear current causes heating in that transformer. This heating can cause the transformer to age prematurely and forces the wind farm operators to replace them more often, causing them to lose money.

This feedback loop consists of an AC/DC converter hooked up back to back with a DC/AC inverter that powers the rotor in such a way that the stator electric frequency matches the grid. This feedback loop is modeled as a simple rectifier system drawing a constant power of 30% of stator power. This is typical of the maximum power drawn by DFIG back to back converter systems. The harmonics generated is then analyzed using PSPICE. Depending on particular transformer impedances, the current total harmonic distortion can range from between 19.1% to 32.0%. In order to mitigate transformer overheating, either the transformer must be derated, or a filtering system must be installed.

In the second part of the project, a reduced scale single phase active filter was designed to reduce the presence of the current harmonics detected in the line. The objective was to create a system that will operate on a 12Vrms line voltage that could produce a correction current of up to 10A. The

physical circuitry comprises of the sensor system, the microcontroller, and an H-Bridge circuit. The sensor system uses three sensors that measure the line voltage, non-linear current, and correction current. With these measurements, the appropriate sinusoidal waveform is calculated, and the difference between that ideal waveform and the non-linear current waveform becomes the target current. Using the information on the present correction current being generated and the target current waveform, the system will increase or decrease the correction current in order to match that target waveform using the H-Bridge circuit. All logic is performed by the microcontroller system.

To test the system, a rectifier was used as a source of non-linear current and the active filter was connected between the rectifier and the autotransformer. The rectifier circuit draws a peak current of 7A, and the current waveform has a total harmonic distortion of 123.8%. When the current to the auto transformer is measured, the resulting current harmonic is at only 13.4% THD. Overall the system successfully reduced current harmonics present in the line. As a recommendation for future work, we suggest adapting the system for three phase operation as well as higher power rating.

Table of Contents

Table of Figures

1 Introduction

Wind energy is an emerging market in the power industry. As a renewable energy source it is important to evaluate this technology and attempt to alleviate any harm that may arise from its implementation to itself of the system it is integrated into in an effort to make the system more feasible from a technical and economic standpoint. This report focuses on the particular problem that is causing the premature aging and failure of step up transformers in wind farms most likely due to harmonic content in the current supplied from the turbine system.

These transformers used to step up the energy from wind turbine generators to collector busses (WTSU's) are displaying unique and disturbing tendencies throughout the nation. They are producing high levels of combustible gases within the oil, overheating, and failing prematurely. The gas levels are indicating internal partial discharge and overheating. Some have been found to have dangerous levels of hydrogen.

This project will perform all necessary background research to fully understand and mitigate this problem. This investigation will begin by attaining waveform samples of active doubly fed induction generator (DFIG) systems that have particularly high gassing to see if there are any anomalies that may explain this partial discharge. Then a physical scale model and computer simulations of a DFIG system will be created in order to study the non-sinusoidal effects of the system.

As a result of this research an active filter design will then be created to be placed on the secondary side of the step up transformers to mitigate any power quality issues. The design of the active filter may be scaled down, but will prove the concept for high power application.

2 Background

This chapter will explain all background information needed to fully understand the analysis and design performed in this project. This includes transformer, rectification, single and doubly fed induction generators, wind turbine physics and active filter theory.

2.1 Dissolved Gas Analysis

Oil sampling can be used to determine the integrity of a transformer and investigate any faults that may have occurred. In this project, oil sampling will be used to determine which transformers are most heavily affected by the phenomenon in question. In this chapter, the basic theory and methodology of oil sampling, also termed dissolved gas analysis, will be explained.

2.1.1 Chromatography

Oil samples are procured from transformers pertaining to each wind turbine. The samples are then chromatographically analyzed in a chemical laboratory per ASTM D-3612 [1]. These test results display the quantity of gas dissolved in the electrical insulating oil. The gasses examined in particular are seen in Table 1.

Hydrogen	H_2
Methane	CH_4
Ethane	C_2H_6
Ethylene	C_2H_4
Acetylene	C_2H_2
Carbon Monoxide	CO
Carbon Dioxide	CO_2

Table 1: Gasses Examined in ASTM D-3612 [1]

These gasses are synthesized through the fragmentation and rearrangement of carbon-hydrogen and carbon-carbon bonds of the hydrocarbon molecules originating in the oil. The classification of the chemical bond created in the new molecules can be directly associated to the temperature and partial pressure it has been subjected to. This relationship is epitomized in Figure 1.

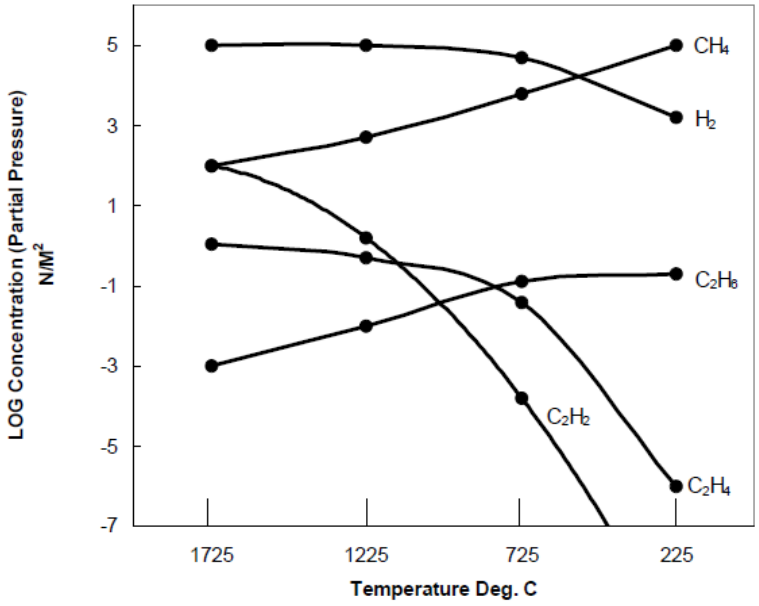


Figure 1: Halstead's Thermal Equilibrium Plot (Specified in IEEE C57.104-2008) [2]

2.1.2 IEEE Standard

The correlations seen in Figure 1 have led to the establishment of IEEE C57.104-2008 [2], which delineates the acceptable levels of each of the gasses presented in Table 1, along with the acceptable volumetric amount of total combustible dissolved gas (TCDG) allowed. The TCDG can be calculated using Equation 1 from the amounts of each gas found through chromatography and oil volume. Notice that CO_2 is not included because it is not a combustible gas.

$$TDCG = \frac{(H_2 + CH_4 + C_2H_6 + C_2H_4 + C_2H_2 + CO)(Volume\ of\ Oil\ in\ Liters)}{10^5}$$

Equation 1: Total Dissolved Combustible Gas Equation [2]

2.1.2.1 Gas Limits

Once the gas levels are known, they can be compared to the limits provided by IEEE C57.104. These limits outline the suggested warning level assigned to the different gas levels. The suggested levels can be seen in Table 2.

Status	Dissolved key gas concentration limits [µL/L (ppm)[a]]							
	Hydrogen (H_2)	Methane (CH_4)	Acetylene (C_2H_2)	Ethylene (C_2H_4)	Ethane (C_2H_6)	Carbon monoxide (CO)	Carbon dioxide (CO_2)	TDCG[b]
Condition 1	100	120	1	50	65	350	2 500	720
Condition 2	101–700	121–400	2–9	51–100	66–100	351–570	2 500–4 000	721–1920
Condition 3	701–1800	401–1000	10–35	101–200	101–150	571–1400	4 001–10 000	1921–4630
Condition 4	>1800	>1000	>35	>200	>150	>1400	>10 000	>4630

NOTE 1—Table 1 assumes that no previous tests on the transformer for dissolved gas analysis have been made or that no recent history exists. If a previous analysis exists, it should be reviewed to determine if the situation is stable or unstable. Refer to 6.5.2 for appropriate action(s) to be taken.

NOTE 2—An ASTM round-robin indicated variability in gas analysis between labs. This should be considered when having gas analysis made by different labs.

[a] The numbers shown in Table 1 are in parts of gas per million parts of oil [µL/L (ppm)] volumetrically and are based on a large power transformer with several thousand gallons of oil. With a smaller oil volume, the same volume of gas will give a higher gas concentration. Small distribution transformers and voltage regulators may contain combustible gases because of the operation of internal expulsion fuses or load break switches. The status codes in Table 1 are also not applicable to other apparatus in which load break switches operate under oil.

[b] The TDCG value does not include CO_2, which is not a combustible gas.

Table 2: Dissolved Gas Concentration Limits **[2]**

As seen in Table 2, there are four "conditions" that denote the severity of the condition of the gassing. The IEEE C57.104 standard also classifies these conditions definitively, which can be seen in Table 3.

	TDCG levels (µL/L)	TDCG rate (µL/L/day)	Sampling intervals and operating procedures for gas generation rates	
			Sampling interval	Operating procedures
Condition 4	>4630	>30	Daily	Consider removal from service. Advise manufacturer.
		10 to 30	Daily	
		<10	Weekly	Exercise extreme caution. Analyze for individual gases. Plan outage. Advise manufacturer.
Condition 3	1921 to 4630	>30	Weekly	Exercise extreme caution. Analyze for individual gases. Plan outage. Advise manufacturer.
		10 to 30	Weekly	
		<10	Monthly	
Condition 2	721 to 1920	>30	Monthly	Exercise caution. Analyze for individual gases. Determine load dependence.
		10 to 30	Monthly	
		<10	Quarterly	
Condition 1	≤720	>30	Monthly	Exercise caution. Analyze for individual gases. Determine load dependence.
		10 to 30	Quarterly	Continue normal operation.
		<10	Annual	

Table 3: Condition Classifications [2]

As seen in Table 3, dissolved gas analysis requires periodic sampling in order to reach conclusions or diagnosis of transformer conditions. Once multiple samples have been taken, the severity of the gassing can be determined by calculating at the rate of increase. This can be calculated using Equation 2.

$$R = \frac{(S_T - S_0) * V * 10^{-6}}{T}$$

Equation 2: Gas Generation Rate Equation per IEEE C57.104-2008 [2]

Where

R Is the rate (Liters/Day)

S_0 Is the first sample (µL/L)

S_T Is the second sample (µL/L)

V Is the tank oil volume (Liters)

T Is the time (Days)

2.1.3 Fault Type Evaluation

Gas patterns can be indications of what type of faults are occurring or have occurred in the transformer. This can be accomplished fairly easily by calculating specific gas ratios and comparing their values to guides that were predominantly empirically deduced. The ratios are as follows:

Ratio 1	CH_4/H_2
Ratio 2	C_2H_2/C_2H_4
Ratio 3	C_2H_2/CH_4
Ratio 4	C_2H_6/C_2H_2
Ratio 5	C_2H_4/C_2H_6

Table 4: Table of Ratios

The methods of analysis discussed in this section use dissimilar ratios to reach their conclusions and will be referred to by their number as seen above.

2.1.3.1 Doernenburg Ratio Method

The Doernenburg ratio method makes use of ratios one through four. For this method to be valid the transformer must be of condition one or higher based on the levels of a combustible gas (all gasses seen in Table 5 with the exception of CO_2). Once this prerequisite is satisfied, the values of ratios one through four can be compared to those in TBL to determine the probable fault type.

Suggested fault diagnosis	Ratio 1 (R1) CH_4/H_2		Ratio 2 (R2) C_2H_2/C_2H_4		Ratio 3 (R3) C_2H_2/CH_4		Ratio 4 (R4) C_2H_6/C_2H_2	
	Oil	Gas space	Oil	Gas space	Oil	Gas space	Oil	Gas space
1. Thermal decomposition	>1.0	>0.1	<0.75	<1.0	<0.3	<0.1	>0.4	>0.2
2. Partial discharge (low-intensity PD)	<0.1	<0.01	Not significant		<0.3	<0.1	>0.4	>0.2
3. Arcing (high-intensity PD)	>0.1 to <1.0	>0.01 to <0.1	>0.75	>1.0	>0.3	>0.1	<0.4	<0.2

Table 5: Doernenburg Ratio Fault Types

The Doernenburg ratio method is simple to use and can be accurate. But, it is suggested to use multiple techniques of analysis when investigating fault type and not all gas patterns can be explained by this method.

2.1.3.2 Rogers Ratio Method

The Rogers ratio method, like the Doernenburg, makes use of ratios to compare to empirically trended conclusions. The ratios in use for this technique are one, two, and five. Once these have been calculated, their levels can be compared to Table 6.

Case	R2 C_2H_2/C_2H_4	R1 CH_4/H_2	R5 C_2H_4/C_2H_6	Suggested fault diagnosis
0	<0.1	>0.1 to <1.0	<1.0	Unit normal
1	<0.1	<0.1	<1.0	Low-energy density arcing—PD[a]
2	0.1 to 3.0	0.1 to 1.0	>3.0	Arcing—High-energy discharge
3	<0.1	>0.1 to <1.0	1.0 to 3.0	Low temperature thermal
4	<0.1	>1.0	1.0 to 3.0	Thermal <700 °C
5	<0.1	>1.0	>3.0	Thermal >700 °C

[a] There will be a tendency for the ratios R2 and R5 to increase to a ratio above 3 as the discharge develops in intensity.

Table 6: Rogers Ratio Fault Types

As with the Doernenburg ratio method, it is suggested that the Rogers ratio method be used in conjunction with other protocols to reinforce deduced conclusions.

2.1.4 Oil Quality Analysis

There are several other tests that are run to evaluate the quality of the oil in the transformer tank. These include the following:

Attribute	ASTM Standard
Moisture Content	D-1533B
Interfacial Tension	D-971
Acid Number	D-974
Color	D-1500
Visual/Sediment	D-1524
Dielectric Breakdown	D-1816
Power Factor at @ 25°C and 100°C	D-924
Specific Gravity	D-1298

Table 7: Oil Quality Tests

These test results can then be compared to IEEE C57.106-2006. The limits outlined in IEEE C57.106-2006 [3] can be seen in Table 8, below.

Test and method	Limit value
Dielectric strength ASTM D1816 kV minimum 1 mm gap[b] 2 mm gap[b]	 20 35
Dissipation factor (power factor) ASTM D924 25 °C, % maximum 100 °C, % maximum	 0.05 0.30
Interfacial tension ASTM D971 mN/m minimum	 40
Color ASTM D1500 ASTM units maximum	 0.5
Visual examination ASTM D1524	 Bright and clear
Neutralization number (acidity) ASTM D974 mg KOH/g maximum	 0.015[c]
Water content ASTM D1533 mg/kg maximum[d]	 25[c]
Oxidation inhibitor content when specified ASTM D2668 Type I oil, % maximum Type I oil, % minimum Type II oil, % maximum Type II oil, % minimum	 0.08 0.0 0.3 >0.08
Corrosive sulfur ASTM D1275	 Not corrosive
Relative density (specific gravity) ASTM D1298 15 °C/15 °C maximum	 0.91

[a] Oil dielectric testing in accordance with ASTM D877 has been replaced by ASTM D1816 in Table 1. See 5.2.1.
[b] Alternative measurements of 1.0 mm (0.04 in) and 2.0 mm (0.08 in), respectively, for gaps.
[c] This value is more stringent than the ASTM D3487 requirement.
[d] Equivalent measurement is parts per million (ppm).

Table 8: IEE C57.106-2006

2.2 Online Transformer Monitoring

This project made use of online monitoring to determine the actual voltage and current characteristics. A high voltage meter is required to determine the power quality due to the high power systems used in this project. Also, having a meter that calculates a wide array of power quality

measurements enables the project team to see any abnormalities that may be occurring that may cause problems quickly and easily. For this project, the Fluke 1750 was chosen.

Figure 2: http://www.transcat.com/catalog/productdetail.aspx?itemnum=45832EL

This meter is capable of recording three phase voltage and current waveforms at periodic intervals. These measurements can be saved on the meter for upwards of a month. Once these "snapshots" are recorded they can be revisited and important data such as harmonic spectrum (up to the 50^{th} harmonic), RMS currents and voltages, kVA, KVAR, etc. are automatically calculated for easy system classifications.

The Fluke 1750 enables the project team to quickly analyze transformers at full power as well as the scale model of the system. These measurements can then be easily compared to the theoretical values found from the computer simulation of the system.

2.3 Transformer Derating

This section deals with the amount that transformers must be derated depending on the amount of harmonics present and why. This is pertinent to the project because we are dealing with transformers that are under the stresses of harmonic signals. It must be decided if these harmonics are significant enough to impact the integrity of the transformer.

Harmonics are either AC voltages or currents that possess frequencies that are integer multiples of the fundamental frequency. The American standard of the fundamental frequency is rated at 60 Hz. The presence of harmonics can add a significant amount of distortion to the ideal sine wave of a system to a point where it can no longer be recognized and the system will become very inefficient. Harmonics are caused by the use of non-linear loads. There are a few techniques that can be used to prevent the existence of harmonics. Such techniques include K-Factor transformers, transformer derating and active harmonic filters. [4]

When non-sinusoidal waveforms are introduced to transformers, there is a derating factor that must be strongly considered. At the base of this problem is the fact that when transformers are supplied with waveforms containing harmonics there is an increase in power loss, which leads to an increase in heating. If a transformer is heated beyond its specification it will age at an increased rate and will need to be replaced prematurely. This aging may happen in the form of partial discharge in the oil or the degradation of the solid insulation, also called the degree of polymerization.

There are three major concerns with load loss in power transformers, which neglects excitation losses. They are as follows:

$$\boldsymbol{P_{LL} = P + P_{EC} + P_{OSL}}$$

Equation 3: Load Loss [5]

Where P_{LL} is the total load loss; P is the total resistive losses (I^2R); P_{EC} is the eddy current losses; and P_{OSL} accounts for all other stray losses. By further investigation, we can see that all three of these losses are related to the input harmonics.

Part of the reasoning behind eddy current losses is explained by the skin effect. The skin effect is the custom for higher frequencies currents to flow or drift in the area along the surface of the conductor (transformer). It occurs due to opposing eddy currents that are induced by the alternating current. Since the harmonic currents occur at higher frequencies, they travel beneath the surface and increase the resistance of the transformer. This increase produces a considerable amount of heating which is detrimental to the state of transformer and is why we see that it impacts the derating of the transformer. [6]

As the harmonics content increases, the RMS value may also be increased as a result. Therefore, it can be seen that the P (I^2R losses) will be increased. It is also known that the eddy current losses are directly related to the square of both the input current and harmonic level, therefore increasing with the amount of harmonic content. Finally, connections and structural parts of the transformer can be affected by harmonic content [7]. This does not have as clear of a relationship as the others.

There are two common transformer rating standards: the K-Factor, the U.S. standard, and the Factor K, the European standard.

$$K = \frac{P_t}{P_f} = \sum_{h=1}^{h=h_{max}} I_h^2 h^2$$

Equation 4: K Factor [8]

$$K = \left[1 + \frac{e}{1+e} \left(\frac{I_1}{I} \right)^2 \sum_{n=2}^{n=N} \left(n^q \left(\frac{I_n}{I_1} \right)^2 \right) \right]^{0.5}$$

Equation 5: Factor K [8]

The K-Factor is a value that regulates the rating of the transformer load as a function of its corresponding harmonic currents and is applied to transformers to determine the magnitude of which they must be derated to insure proper operation and avoid expedited aging. For instance, a K-Factor of 1.0 refers to a transformer that has linear load. The K-Factor values range from 1 to 50, thus the 50th harmonic is the maximum harmonic current that can be corrected by the use of K-Factor transformers. The K-Factor specifies that the total eddy current loss in the transformer will be K times the eddy current loss at the fundamental frequency. This allows manufacturers to design transformers with a large thermal capacity and a low fundamental eddy current loss, in order to reduce the effect of overheating. K-factor transformers are useful because they are designed with values specifically for certain loads and efficiently reduce the effect of harmonics.

In the formula for the K-factor, it can be seen that there are values of “q” and “e”. The q value is an exponential constant dependent on the winding of the transformer. For round rectangular cross section conductors in both windings, the q value is 1.7. The e value is defined as the eddy current loss at the fundamental frequency over the ohmic loss. Its value has been estimated to be about 0.1. Although

transformer derating overcomes the heating effects of harmonics, it is not as efficient as K-factor transformers and active filters because it requires a reduction in the power of the transformer. [9]

The derating values depend on the harmonic content of the power signal, which we will call alpha. This is has an exponential relationship to the harmonic content where a higher alpha value represents a less distorted, more fundamental, waveform. The influence of alpha on the harmonic values can be seen in Figure 3, below (MATLAB Code seen in Appendix B).

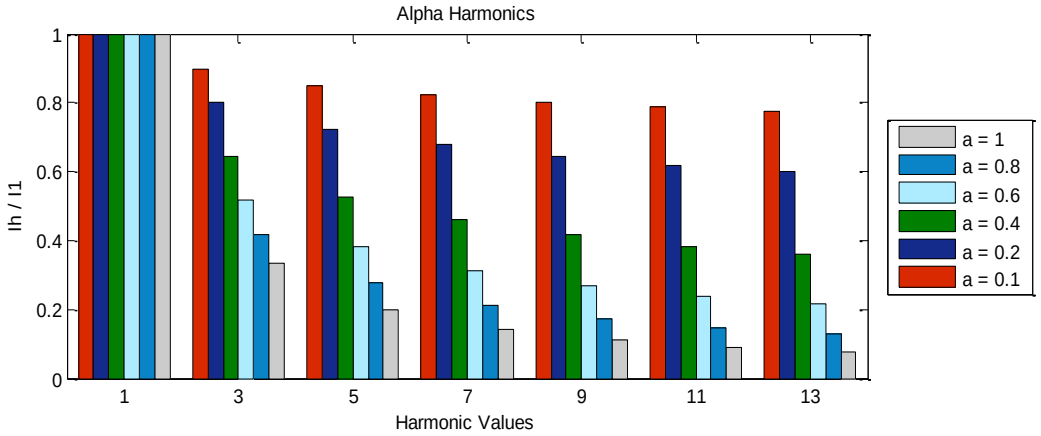


Figure 3: Effect of Alpha Values

From this we can see that this will affect the Factor K through the "e" value, seen in Figure 4, seen below.

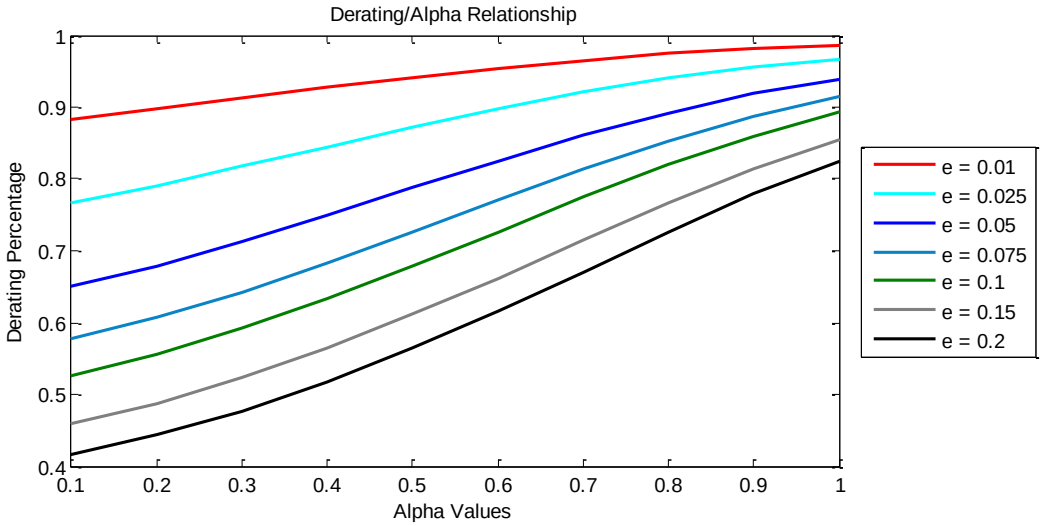


Figure 4: Derating Value vs Alpha

Therefore, as the "e" value increase, more derating is necessary to remove the harmonics present in the system. In practice, a value of 0.1 is assumed when more accurate information is not available. This can then be related to the derating percentage as a function of the alpha value for various "e" values. Finally, these can be related to the eddy current losses, seen in Figure **5** below.

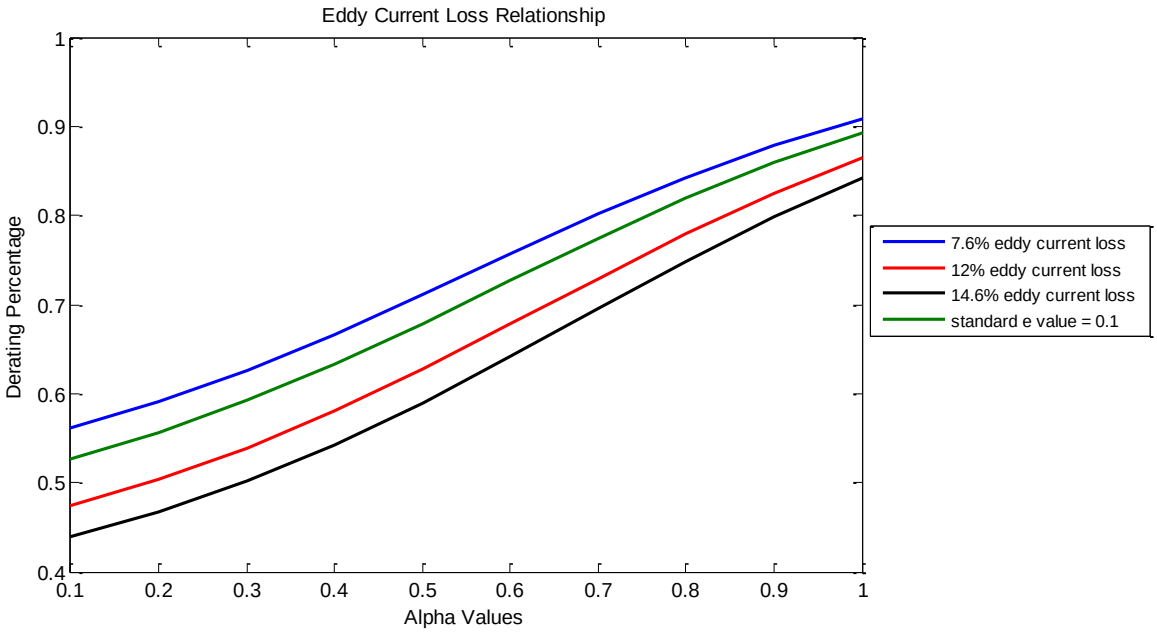


Figure 5**: Eddy Current Loss**

2.4 Single Phase Rectification

A rectifier is a component that converts alternating current (AC) to direct current (DC). An alternating current changes directions over a period of time while a direct current flows in a single direction. A rectifier can output either the half or full waveform of an AC current. A half-wave rectifier produces either the negative or the positive sections of the AC waveform while a full-wave rectifier produces the full waveform as the DC output. A rectifier can control the output current and voltage by means of the components used in the bridge scheme. The rectifier systems that use diodes are uncontrolled rectifiers while those that use SCRs are controlled rectifiers.

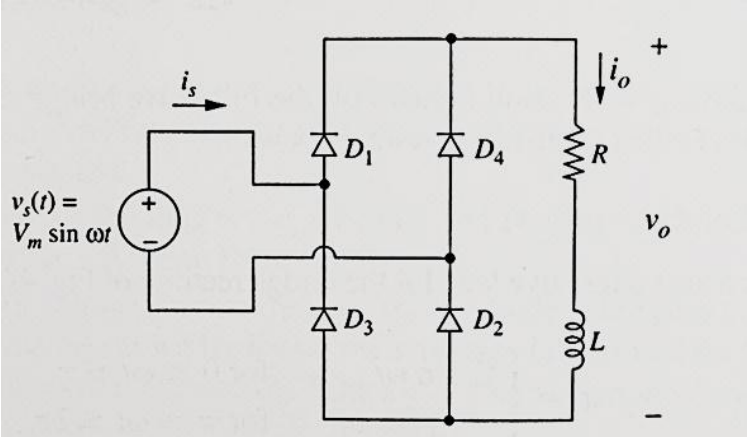


Figure 6: Single Phase Rectifier [10]

A single-phase bridge rectifier can produce the full wave of an AC current by the use of four diodes constructed together, as shown in Figure 6 above. When the positive side of the AC current occurs, diodes D1 and D2 are forward biased while the other two diodes D3 and D4 are reverse biased. On the negative side, diodes D3 and D4 are now forward biased while D1 and D2 also switch their role and become reverse biased. The two pairs of diodes constructed together in this configuration combine their periodical waveforms to create the full wave rectified output that is shown in Figure 7 below.

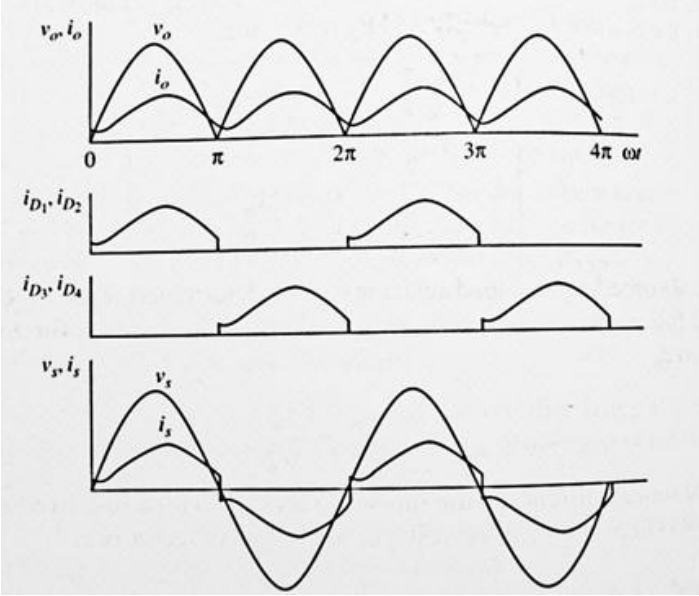


Figure 7: Single Phase Rectifier Waveforms [10]

The average output voltage of the single-phase bridge rectifier is determined by the equation below where V_m refers to the peak value of the AC RMS source.

$$V_O = \frac{2\,V_m}{\pi}$$

Equation 6: Single Phase Rectifier Output Voltage [10]

2.5 Three Phase Rectification

A three-phase bridge rectifier is constructed with the use of a six diode circuit, as shown in the figure below. The circuit is designed with two sets of diodes, the top diodes D1, D3 and D5, and the bottom diodes D4, D6 and D2. Only one diode from each set can conduct at a time due to Kirchhoff's voltage law.

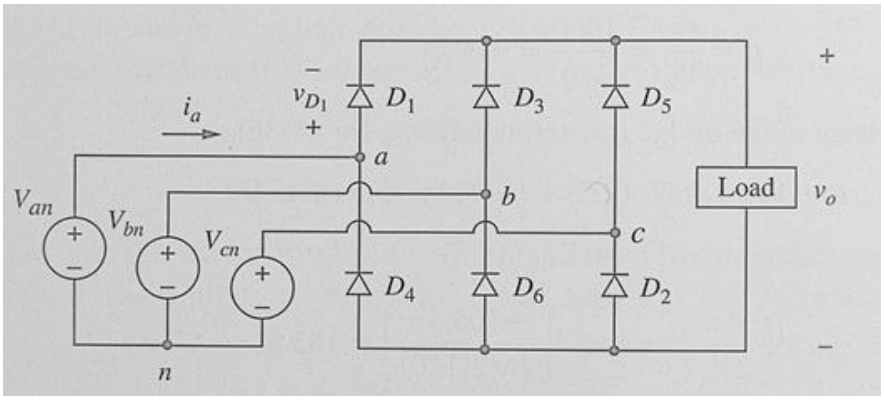


Figure 8: Three Phase Rectifier [10]

The conducting of the diodes depends on the values of the three voltage sources. The top diode conducts when the voltage source that is connected to its anode has the highest value of the three sources. Since V_{an} is the highest source at the start of the source system shown in Figure 9 below, D1 will be the first diode of the top set to conduct. The bottom diode conducts when the voltage source that is connected to its cathode is the lowest of the three sources. Therefore, D6 will be the first diode of the bottom set to conduct because V_{bn} is the source with the lowest voltage. The complete conducting scheme along with the output voltage is also shown in the figure. Since the three-phase rectifier uses more diode switching modes for the alternating current, it produces a lower ripple output voltage than what a single-phase rectifier would produce.

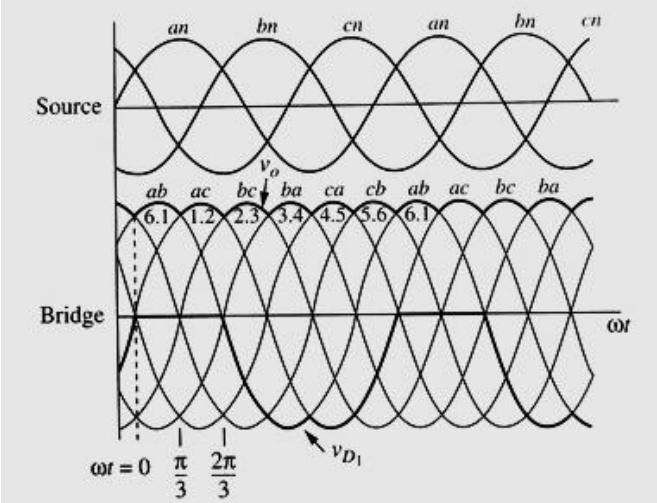


Figure 9: Three Phase Rectifier Output Waveforms [10]

The average output voltage of the three-phase bridge rectifier is determined by the equation below where $V_{m,\,L\text{-}L}$ refers to the peak value of the line-to-line voltage of the three-phase AC RMS source.

$$V_O = \frac{3\,V_{m,L-L}}{\pi}$$

Equation 7: Three Phase Rectifier Output Voltage [10]

2.6 Doubly Fed Induction Generator (DFIG)

There are four main types of wind energy systems. The oldest and most basic system is the fixed speed system. It is apparent from the name that these systems run at a single speed. They also include a geared shaft and a reactive power compensator to help with the turbine's demands. By running at a single speed, these systems cannot take advantage of the full power potential, which will be discussed more in depth later. The second type of system is a dynamic slip control system. This is similar to the fixed speed system, but has variable resistance in series with the rotor. The resistance variation changes the speed and slip of the drive.

A newer system is the direct drive, which eliminates the gear box by converting to DC and then converting back to AC with a DC link connecting the two converters. This system uses permanent magnetic synchronous generation technology, which entails a higher number of poles. It is a low maintenance system due to the simpler mechanical scheme, but is more expensive due to the number of permanent magnets.

The fixed and dynamic speed systems led to the most common wind generation system, the doubly fed induction system, which is what this project is mostly concerned with. Like the direct and dynamic systems, the doubly fed induction is a geared system, but differs by having a feedback loop from stator to rotor. Meaning both the stator and rotor have windings. The feedback loop has a line side converter and a rotor side converter that are separated by a DC link. This allows for varying voltage to be applied to the rotor of the system.

To fully understand this system we must first look at how power is absorbed by wind turbine systems and how single fed induction machines work. From here we will have a proper background to fully understand the doubly fed induction system.

2.6.1 Singly Fed Induction Machine

In order to understand the DFIG system, it is useful to first study a singly fed induction machine. The singly fed induction machine consists of a rotor with windings that are shorted. When the stator is excited by a three-phase voltage source it creates a magnetic field in the gap between the stator and rotor. The stator flux will demand movement at what is called synchronous speed. This is the speed that the actual stator flux wave is moving and its definition is:

$$\omega_{sync} = \frac{2\pi f_1}{p}$$

Equation 8: Synchronous Speed [11]

Where ω_1 is the synchronous speed (radians per second), f_1 is frequency (in Hertz), and p is the number of poles (in pairs). However, this can be converted from mechanical radians (ω_{mech}) to electrical radians (ω_m). This is defined as:

$$\omega_{mech} = \frac{\omega_m}{p}$$

Equation 9: Mechanical Speed [11]

A common term used to define the difference between the mechanical and synchronous speeds is called slip. It is defined as, where S is slip:

$$S = \frac{\omega_{sync} - \omega_m}{\omega_{sync}}$$

Equation 10: Slip [11]

With this value, the direction and operation mode can be determined. These are outlined in Table 9 below.

State	Rotor Direction	Function
S > 1	Opposite stator field ($\omega_m < 0$)	Braking
S = 1	No movement ($\omega_m = 0$)	Transformer
0 < S < 1	Same as stator, but slower ($0 < \omega_m < \omega_{sync}$)	Motor
S < 0	Same as stator, but faster ($\omega_m > \omega_{sync}$)	Generator

Table 9: Slip and Generator Function

This slip value can also help to determine the rotor field velocity and current frequency as:

$$\omega_r = S\omega_{sync}$$

Equation 11: Rotor Field Velocity [11]

$$f_r = Sf_{sync}$$

Equation 12: Rotor Current Frequency [11]

Where ω_r is the rotor field velocity, f_r is the rotor current frequency, and f_{sync} is the stator current frequency.

2.6.2 Doubly Fed System

The doubly fed system has windings on both the stator and rotor side. It makes use of a feedback loop running from stator to rotor that include a line side converter (LSC), a DC link and a rotor side converter (RSC). This can be seen in Figure 10, below.

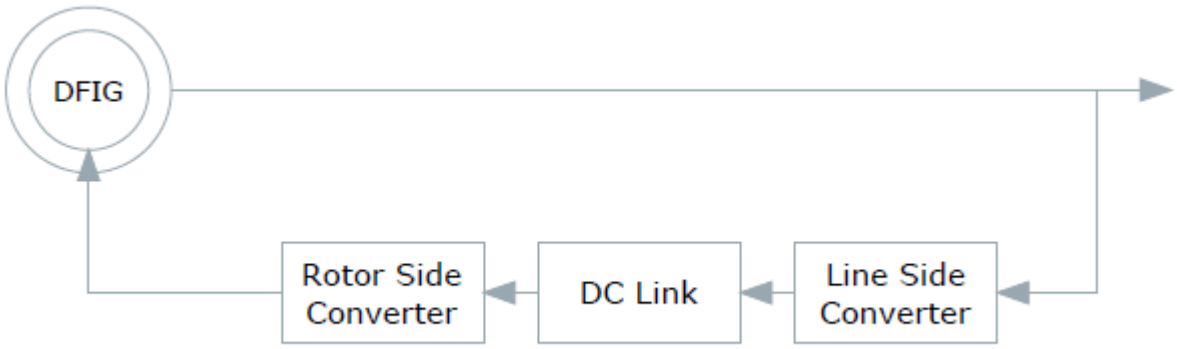


Figure 10: DFIG Feedback Loop

The line side converter will regulate the voltage to the DC link. From there, the rotor side converter will convert the DC link to AC at the frequency required at the rotor. Through the variation of these values, the power taken from the line and injected back into the rotor can be controlled.

Due to the windings on both the rotor and stator, they will both be creating magnetic fields that will interact, creating torque. The torque can be defined as:

$$T = |\psi_s||\psi_r| \cos\theta$$

Equation 13: DFIG Torque [12]

Where *T* is the torque created, ψ_s is the stator magnetic field, ψ_r is the rotor magnetic field, and θ is the angle between the stator and rotor magnetic fields. This shows that the torque is at its maximum when these two fields are normal to each other.

Each the stator and rotor will have a three phase winding configuration. The stator is connected to the power grid and the rotor is connected to the rotor side converter (which is in series with the DC link connected to the grid side converter and back to the grid), these connections will determine the voltages seen at the stator and rotor. There are:

$$v_{sa} = V_s \cos(\omega_1 t) \qquad v_{ra} = V_r \cos(\omega_r)$$

$$v_{sb} = V_s \cos\left(\omega_1 - \frac{2\pi}{3}\right) \qquad v_{rb} = V_r \cos\left(\omega_r - \frac{2\pi}{3}\right)$$

$$v_{sc} = V_s \cos\left(\omega_1 - \frac{4\pi}{3}\right) \qquad v_{rc} = V_r \cos\left(\omega_r - \frac{4\pi}{3}\right)$$

Equation 14: Stator and Rotor Voltages [11]

Where v_{sa}, v_{sb}, and v_{sc} are the stator voltages on phase a, b, and c. Similarly, v_{ra}, v_{rb}, and v_{rc} are the voltages for phase a, b, and c on the rotor. The definition of ω_1 is:

$$\omega_1 = \frac{2\pi f_1}{p}$$

Equation 15: ω_1 [11]

Where f_1 is the grid frequency. The relationship between ω_1 and ω_r is:

$$\omega_r = s\omega_1$$

Equation 16: Rotor Frequency [11]

Where, similarly to the single fed system, the slip of the doubly fed machine is defined as:

$$s = \frac{n_s - n_r}{n_s}$$

Equation 17: Definition of Slip [11]

Where n_s is the synchronous speed and n_r is the mechanical speed of the rotor, both of which are in rpm. The synchronous speed is defined as:

$$n_s = \frac{60 f_e}{p}$$

Equation 18: Synchronous Speed [11]

Where f_e is the electrical frequency.

The system can be represented in a t-circuit with parasitic inductors and resisters on each side of two coils with a high coupling coefficient to signify the power transfer between the stator and rotor where:

R_s is the rotor resistance

L_s is the rotor leakage inductance

L_m is the magnetization branch

L_s is the rotor leakage inductance

R_r is the rotor resistance

sk is the turns ratio between the stator and rotor

i_s is the stator current

i_r is the rotor current

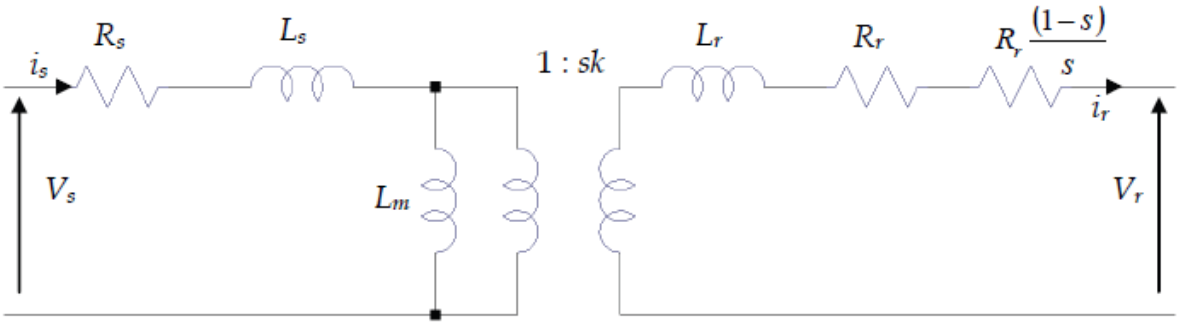


Figure 11: Per Phase Equivalent Circuit of an Induction Machine [12]

The mechanical dissipation can be modeled as $R_r(1-s)/s$. Using this notation the mechanical power can be calculated by:

$$P_{mech} = 3|i_r|^2 \left(\frac{1-s}{s}\right) R_r$$

Equation 19**: Mechanical Power [12]**

The power going through the stator and rotor can be related to each other by:

$$P_r = P_m - P_s = T_m\omega_r - T_{em}\omega_s = -T_m\left(\frac{\omega_s - \omega_r}{\omega_s}\right)\omega_s = -sT_m\omega_s = -sP_s$$

Equation 20**: Relationship between Stator and Rotor Power [12]**

This leads to a similar definition found in the singly fed machine where there are multiple modes of operation depending on the slip. However, in the case of the doubly fed machine there are two per condition of the slip instead of one depending on the direction of the power flow. They are as follows:

Slip	Mechanical Power	Stator Power	Rotor Power	Operation Mode
S > 0	Outward	Inward	Outward	Motor
S > 0	Inward	Outward	Inward	Generator
S < 0	Outward	Inward	Inward	Motor
S < 0	Inward	Outward	Outward	Generator

Table 10: Slip and Generator Power

2.7 Active Filtering

Active filtering uses components such as switches, logic, and possibly microprocessors to mitigate unwanted harmonics. These are much more versatile than passive filters, which have to be tuned to a particular frequency. However, they are much more complicated. For the active filter, the designer can choose the desired output waveform within a certain tolerance.

This is done using power switches. These switches can be toggled depending on whether the current needs to be sunk or pushed from the line. In parallel with the switches will be an energy storage device, such as a capacitor. This will need to keep a voltage that is higher than the line voltage. This energy storage device will create the voltage potential that will force the current flow.

To decide when to toggle these switches, a test signal is generated, in the case of this project, using pulse width modulation output from a microprocessor. This signal is a scaled down version of the desired output. Once the test signal is attained, the circuit must sample the line conditions to compare

it to the ideal signal. Once it is known whether current will have to be added or subtracted the appropriate signals can be sent from the microcontroller to the appropriate switches.

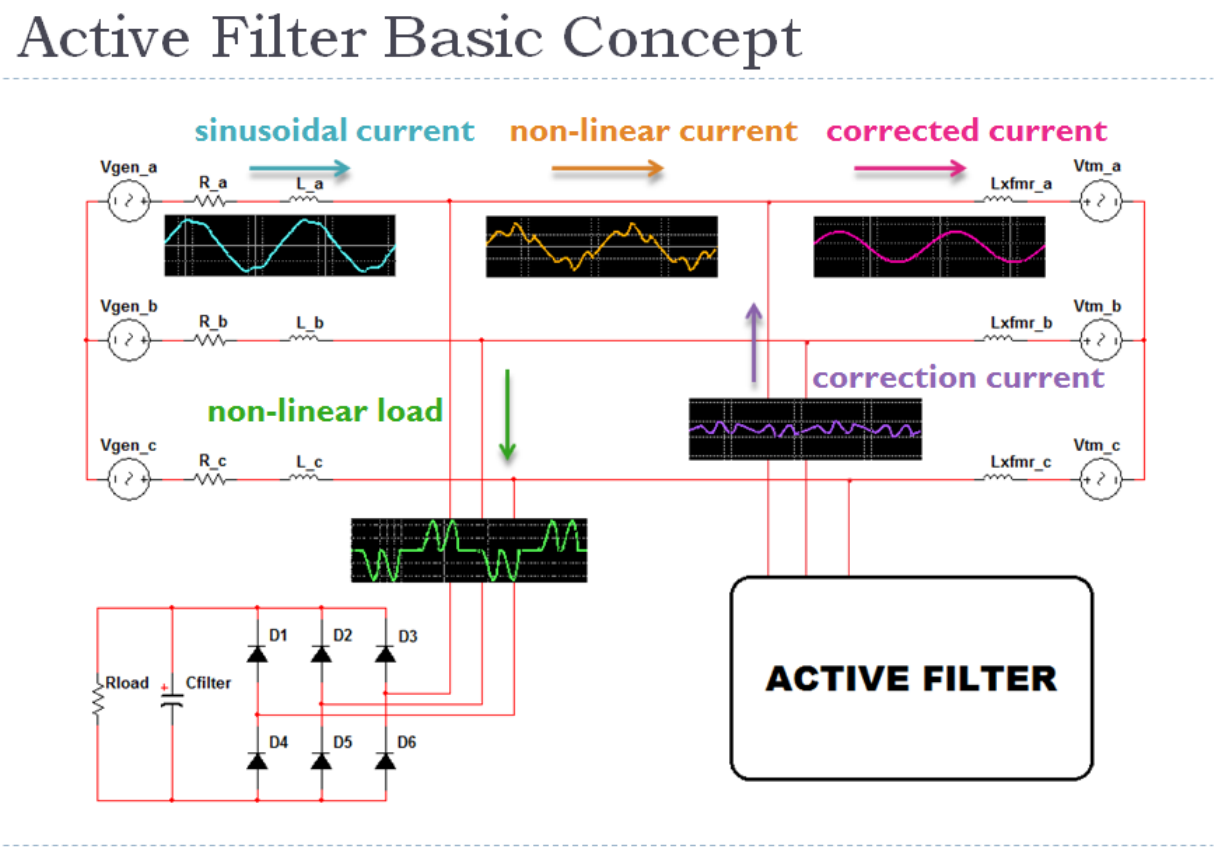


Figure 12: Basic Concept of an Active Filter System

The exact design of all the sensors and switching scheme depend on the application. The particular design and analysis for this project will be explained in chapter 3. In Figure 12 the basic concept of the active filter can be seen. Starting from the left, there is a three phase generation, each generator being 120^{o} out of phase from each other. They then each have line losses associated with them in the form of resistances and inductances. Here, it can be seen that the path splits. The downward path displays the harmonic current that is being pulled by the grid side converter in the feedback system of the DFIG

system. The path continuing to the right shows the harmonically distorted current that is left over. This is the problem current that must be fixed.

It can be seen that the active filter is then in tapped off the line down the line, after the harmonic current is pulled away, but before the transformers and grid connection. This is placed here so that the filter can add or subtract the necessary current from the line to produce a sinusoidal wave. Once this is completed, it can be seen that the transformers see a current that has significantly less harmonics, which is what it is designed to see and will prolong the life of the equipment.

3 Analysis and Design

This chapter will outline the analysis of the DFIG system scale model, computer simulation, and any physical research that was done. All modeling of the system was completed in PSPICE, which can be found in Appendix A.

3.1 Transformer A DGA Analysis

This report outlines the dissolved gas analysis from a transformer in question of being at harm due to harmonics. This analysis is used to determine troubled units so their cases can be studied. The gassing pattern is as follows:

Date	**8/17/2011**	**5/25/2011**	**7/10/2010**
Oil Temp	38	35	40
Hydrogen (H_2)	444	433	89

Methane (CH_4)	1014	810	21
Ethane (C_2H_6)	241	254	5
Ethylene (C_2H_4)	1600	1378	4
Acetylene (C_2H_2)	12	12	0
Carbon Monoxide (CO)	218	186	110
Carbon Dioxide (CO_2)	437	413	329
Nitrogen (N_2)	73873	64701	55382
Oxygen (O_2)	14391	15636	22714
Total Gas	92230	83823	78654

Table 11: Transformer DGA Analysis

Putting this into a gas spectrum we can make some general deductions. This spectrum is seen below.

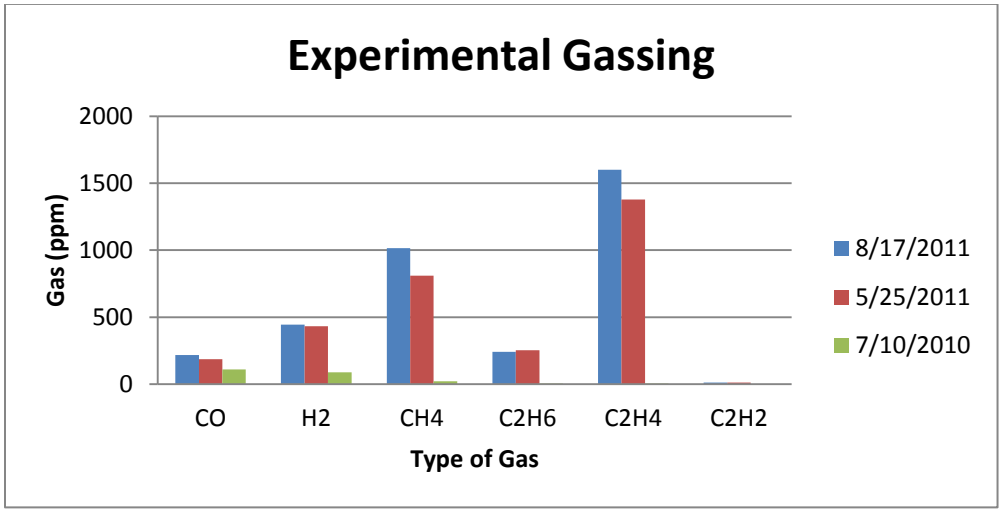


Figure 13: Gas Spectrum

This pattern somewhat resembles the patterns of overheating from IEEE C57.104 [2] seen below but is not conclusive. Therefore, ratios will be used.

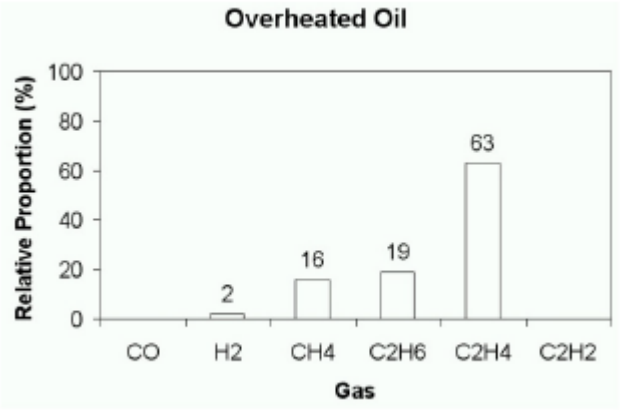


Figure 14: Possible Overheating

3.2 Ratio Analysis

The ratios will be calculated now in order to make use of the Roger's and Doerenburg methods of fault evaluation. For the most recent sample, the ratios are as follows:

Ratio 1	CH_4/H_2	2.2838
Ratio 2	C_2H_2/C_2H_4	0.0075
Ratio 3	C_2H_2/CH_4	0.0118
Ratio 4	C_2H_6/C_2H_2	20.083
Ratio 5	C_2H_4/C_2H_6	6.639

Table 12: Gas Ratios

3.2.1 Doerenburg Method

We can now check these values against the limit concentrations discussed previously, seen below.

Key gas	Concentrations *L*1 [µL/L (ppm)]
Hydrogen (H_2)	100
Methane (CH_4)	120
Carbon monoxide (CO)	350
Acetylene (C_2H_2)	1
Ethylene (C_2H_4)	50
Ethane (C_2H_6)	65

Table 13: Limit Concentrations

In this case, we need to have at least one gas that is more than double the minimum concentration and at least one more surpass a limit as stated in the table 13 as well. Here we clearly have hydrogen, methane, acetylene, ethylene, and ethane well above these limits. Therefore, this analysis applies and we can compare the ratios to the table 14 found below.

Suggested fault diagnosis	Ratio 1 (R1) CH_4/H_2		Ratio 2 (R2) C_2H_2/C_2H_4		Ratio 3 (R3) C_2H_2/CH_4		Ratio 4 (R4) C_2H_6/C_2H_2	
	Oil	Gas space	Oil	Gas space	Oil	Gas space	Oil	Gas space
1. Thermal decomposition	>1.0	>0.1	<0.75	<1.0	<0.3	<0.1	>0.4	>0.2
2. Partial discharge (low-intensity PD)	<0.1	<0.01	Not significant		<0.3	<0.1	>0.4	>0.2
3. Arcing (high-intensity PD)	>0.1 to <1.0	>0.01 to <0.1	>0.75	>1.0	>0.3	>0.1	<0.4	<0.2

Table 14: Ratio Comparison [2]

It can be seen that the ratios clearly show that there has been thermal decomposition in this transformer.

3.2.2 Roger's Ratio Method

The Roger's Ratio method is similar to the Doerenburg Method and should yield similar results (this is a redundant exercise). We will be using the ratios calculated previously to investigate what type of fault, if any, has occurred in this transformer. The table 15 used is seen below.

Case	R2 C_2H_2/C_2H_4	R1 CH_4/H_2	R5 C_2H_4/C_2H_6	Suggested fault diagnosis
0	<0.1	>0.1 to <1.0	<1.0	Unit normal
1	<0.1	<0.1	<1.0	Low-energy density arcing—PD[a]
2	0.1 to 3.0	0.1 to 1.0	>3.0	Arcing—High-energy discharge
3	<0.1	>0.1 to <1.0	1.0 to 3.0	Low temperature thermal
4	<0.1	>1.0	1.0 to 3.0	Thermal <700 °C
5	<0.1	>1.0	>3.0	Thermal >700 °C

Table 15: Roger's Ratio for Key Gasses

Here, we can see that we have a case 5 indicating there has been a thermal fault greater than 700˚C. It was found via the Doerenburg Method that there was a thermal fault and it was also seen in the gassing distribution. The Roger's Ratio method has given a more in depth view of how significant this fault was, which, in this case, is of the highest degree.

3.3 DFIG Harmonics Analysis

This section investigates the harmonics in the doubly fed induction generation system. It will make use of a Fuhrlander turbine for realistic modeling to be done in PSPICE. From these results, proper derating values can be determined.

3.3.1 Base Model

The Fuhrlander FL1500 wind turbine is selected as a basis for our analysis. The manufacturer provides the single-phase equivalent circuit diagram shown below:

Fuhrländer
Friendly Energy

SPECIFICATION – GENERATOR

SPEZIFIKATION – GENERATOR

Voltage:	690 VAC
Frequency:	50 Hz
Speed:	1.000-2.000 rpm
Equivalent circuit diagram:	Values relating to stator side. Temperatures at rated power and +20°C ambient temperature. Rated power and frequency. Applies to values around synchronous speed.
Stator resistance:	R_1 = 0.0056 Ω
Stator leakage inductance:	$X_{1\sigma}$ = 0.0745 Ω
Iron loss resistance:	$R_{F\sigma}$ = 140 Ω
Magnetizing inductance:	X_{1H} = 3.2 Ω
Rotor leakage inductance:	$X'_{2\sigma}$ = 0.0488 Ω
Rotor resistance:	R'_2 = 0.0054 Ω
Rotor voltage:	U_{L-L} = (approx.) 2,000 VAC (s = 1)
Voltage:	U_σ = 690 VAC

Figure 15: Fuhrlander Manufacturer's Specifications [13]

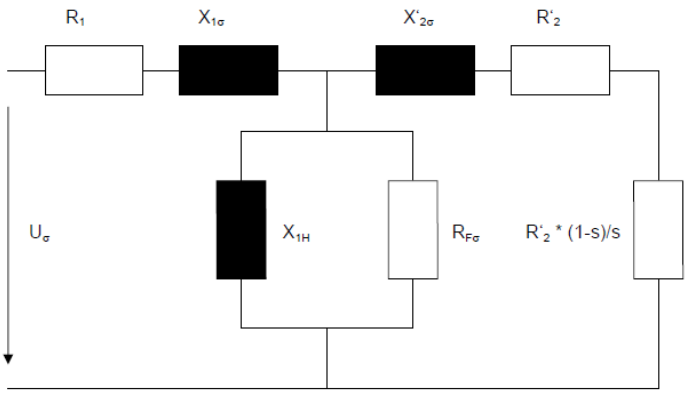


Figure 16: Fuhrlander Single-Phase Equivalent Circuit [13]

The circuit provided shows a single phase of the delta connection and is referenced line to line instead of line to neutral. In order to simplify the analysis, the magnetizing current is ignored and the series impedance is combined. The impedance $R'_2*(1-s)/s$ is also ignored and will be treated as a voltage source later on. The circuit is then converted to a Wye connection equivalent with values referenced from line to neutral as follows:

FL1500 Equivalent Model

Rated Voltage

$V_{LL} := 690V$

$V_{LN} := \frac{V_{LL}}{\sqrt{3}} = 398.372\,V$

Rated Frequency

$f_{Gen} := 50Hz$

$\omega_{Gen} := 2\pi f_{Gen} = 314.159\,\frac{1}{s}$

Combined Delta Series Impedance

$R_1 := 0.0056\Omega$

$X_{1\sigma} := 0.0745\Omega$

$R_{2\_prime} := 0.0054\Omega$

$X_{2\sigma\_prime} := 0.0488\Omega$

$R_{Gen\Delta} := R_1 + R_{2\_prime} = 0.011\,\Omega$

$X_{Gen\Delta} := X_{1\sigma} + X_{2\sigma\_prime} = 0.123\,\Omega$

Equivalent Wye Impedance

$R_{GenY} := \frac{R_{Gen\Delta}}{3} = 3.667 \times 10^{-3}\,\Omega$

$X_{GenY} := \frac{X_{Gen\Delta}}{3} = 0.041\,\Omega$

$L_{GenY} := \frac{X_{GenY}}{\omega_{Gen}} = 1.308 \times 10^{-4}\,H$

Figure 17: Calculation of Generator Impedance Value

Next, a Delta-Wye transformer is connected to the generator equivalent circuit with the Wye side being connected to the generator and the Delta side being connected to a 13.8kV line. For this first model, a 1.5MVA transformer with a 5% per unit impedance is chosen. The calculation for the equivalent Wye connected inductance is as follows:

Transformer Model

Grid - Δ : Y - Generator

Rated Apparent Power

$S_{base} := 1500KV \cdot A$

Rated Frequency

$f_{xfmr} := 60Hz$

$\omega_{xfmr} := 2\pi f_{xfmr} = 376.991 \frac{1}{s}$

Rated Per Unit Impedance

$Z_{pu} := 0.05$

Rated Voltage

$V_{LV\_base} := 690V$

$V_{HV\_base} := 13800V$

Assume transformer impedance to be purely inductive:

$X_{pu} := Z_{pu} = 0.05$

$X_{LV\_pu} := X_{pu} = 0.05$

$X_{HV\_pu} := X_{pu} = 0.05$

Transformer per unit impedance is the same regardless of to which side of the transformer it is referred.

$Z_{LV\_base} := \frac{V_{LV\_base}^2}{S_{base}} = 0.317\,\Omega$

$X_{LV\_base} := Z_{LV\_base}$

$X_{LV} := X_{LV\_pu} \cdot X_{LV\_base} = 0.016\,\Omega$

$L_{LV} := \frac{X_{LV}}{\omega_{xfmr}} = 4.21 \times 10^{-5}\,H$

$Z_{HV\_base} := \frac{V_{HV\_base}^2}{S_{base}} = 126.96\,\Omega$

$X_{HV\_base} := Z_{HV\_base}$

$X_{HV} := X_{HV\_pu} \cdot X_{HV\_base} = 6.348\,\Omega$

$L_{HV} := \frac{X_{HV}}{\omega_{xfmr}} = 0.017\,H$

Figure 18: Calculation of Transformer Impedance Value

Next, the generator voltage and power angle between the generator and grid must be found such that the power delivered is approximately 1.5MW. The calculation is as follows:

Power Angle Calculation

$E_{Gen} := \frac{580 \cdot \sqrt{3}}{\sqrt{2}} V = 710.352\,V$

$E_{Grid} := \frac{563.383 \cdot \sqrt{3}}{\sqrt{2}} V = 690\,V$

In this case, the generator voltage is set slightly higher than the grid voltage, and the power angle is set so that the power transfer approximately 1.5MW.

$X_{GenGrid} := X_{LV} + X_{GenY} = 0.057\,\Omega$

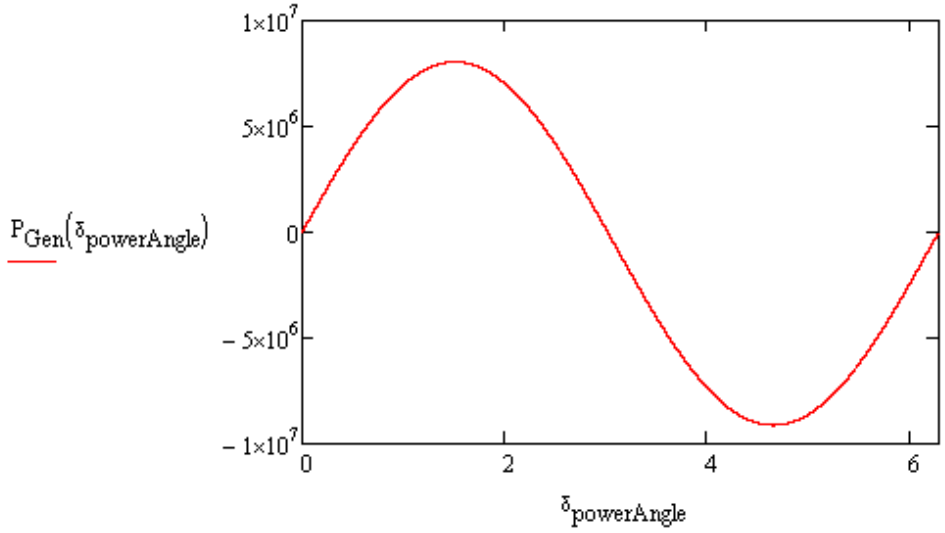
$R_{GenGrid} := R_{GenY} = 3.667 \times 10^{-3}\,\Omega$

$Z_{GenGrid} := \sqrt{R_{GenGrid}^{2} + X_{GenGrid}^{2}} = 0.057\,\Omega$

$\alpha_z := \operatorname{atan}\left(\frac{R_{GenGrid}}{X_{GenGrid}}\right) = 0.064$

$\delta_{powerAngle} := 0deg, 0.01deg .. 359.99deg$

$P_{Gen}(\delta_{powerAngle}) := \frac{E_{Gen} \cdot E_{Grid}}{Z_{GenGrid}} \cdot \sin(\delta_{powerAngle} + \alpha_z) - \frac{E_{Grid}^{2} R_{GenGrid}}{Z_{GenGrid}^{2}}$



$P_{Gen}(9.525deg) = 1.426 \times 10^{6}\,W$

The power angle of 9.525 degrees will be used in following PSPICE simulations.

Figure 19: Power Angle Calculation

With all the parameters calculated, we can build a PSPICE model using the generator and transformer impedances as well as the power angle. The PSPICE circuit for our simplified model can be seen in Figure 20.

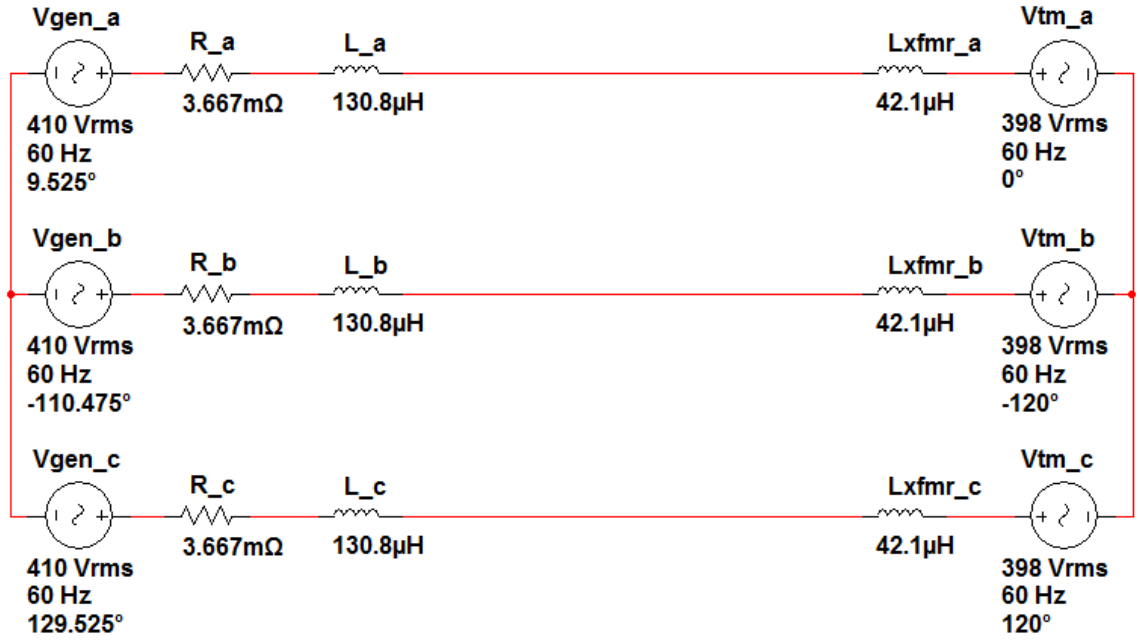


Figure 20: PSPICE Model of the Fuhrlander Generator with No Rectification

The voltage level on the generator side is 410Vrms line-to-neutral or 710Vrms line-to-line. This is set slightly higher than the 690Vrms line-to-line value (398Vrms line-to-neutral) to indicate power flow towards the transformer. The phase angle is set at 9.525 degrees as previously calculated in Figure 19.

3.3.2 Adding a Three Phase Rectifier as a Model for the Rotor Power Converters

During steady-state sub-synchronous operation, where the slip is between zero and one, power flows from the transformer to the Rotor Power Converters to the rotor itself. As mentioned in the

background section, the Rotor Power Converters consists of a Rotor Side Controller, a Grid Side Controller, and a DC link in between. In this analysis, the power flow from the Grid Side Controller to the DC link is modeled as a three phase rectifier, and the power flow from the DC link to the Rotor Side Controller is modeled as a simple resistive load. The model is shown below in Figure 21.

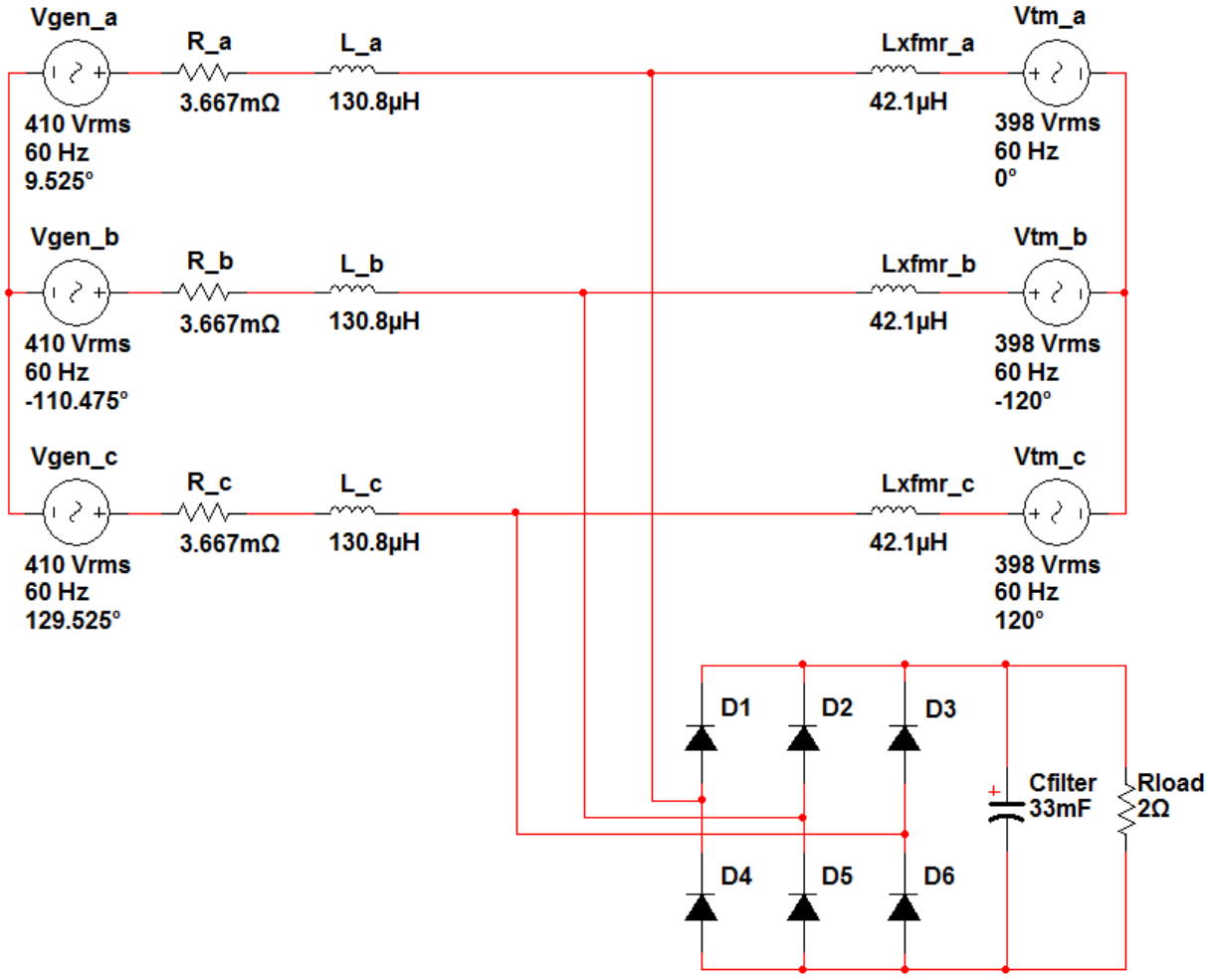


Figure 21: Full PSPICE Model of DFIG System

Thus the rotor power can be virtually adjusted by raising or lowering the load resistance. Assuming a lossless DFIG system, the relationship between P_r and P_s is:

$$P_r = P_m - P_s = T_m\omega_r - T_{em}\omega_s = -T_m\left(\frac{\omega_s - \omega_r}{\omega_s}\right)\omega_s = -sT_m\omega_s = -sP_s$$

Equation 21: Rotor and Stator Power Relationship

The typical DFIG Rotor Power Converter is rated around 30% of the rated power of the induction generator. This means a slip value range of ±0.3. [12] Since the three phase rectifier only models power flow in the sub-synchronous operation, we are only interested in slip values that are greater than zero. For a study of the worst case scenario for harmonics generation, the maximum slip value of 0.3 is chosen. This means that the resistor value must be chosen so that the power across it is 30% of the stator power. As seen in Figure 21, a load resistive value of 2 ohms and filter capacitor value of 33mF is used to simulate the generated harmonic current. Figure 22 shows the output power of the rectifier is approximately 433kW, or about 31.6% of the stator power of 1.37 MW. The output voltage ripple on the DC link is approximately 10V, or about 1%.

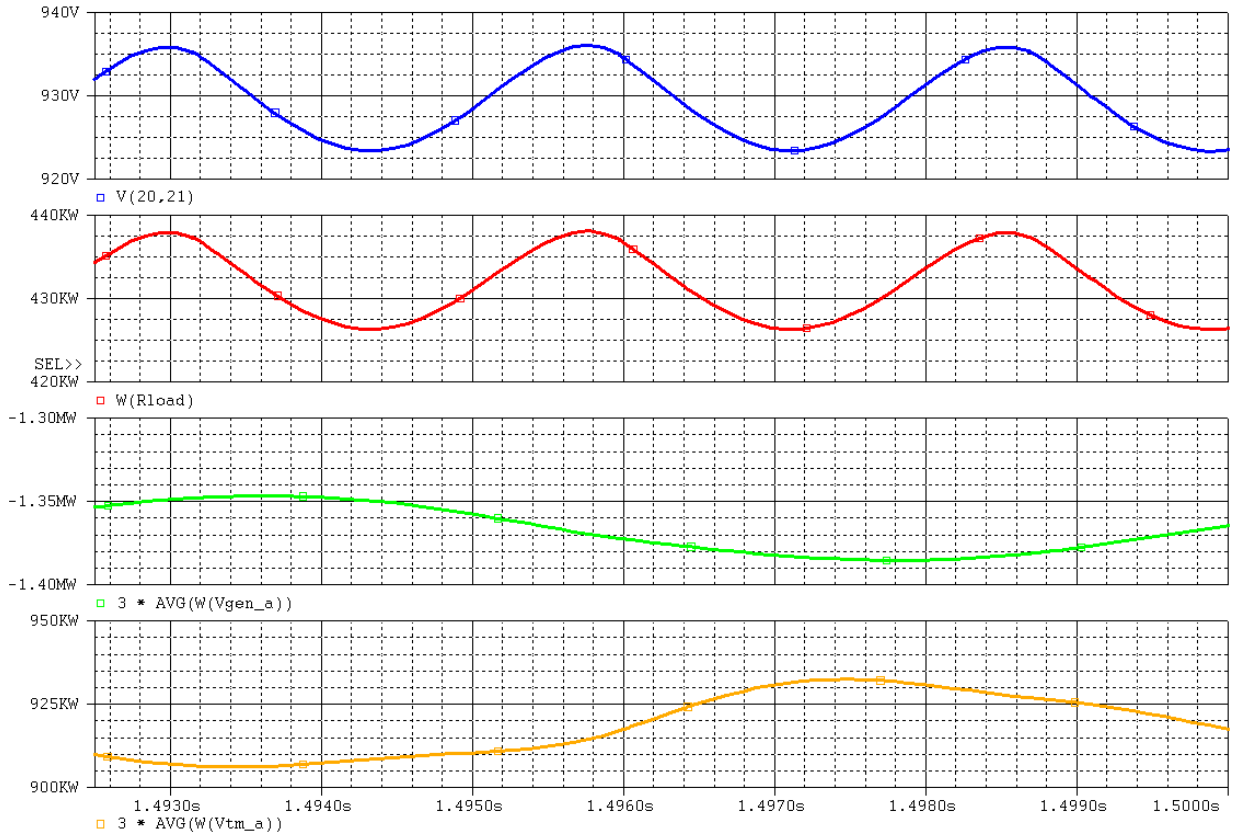


Figure 22: Rectifier Output Voltage and Power, Stator Power, and Power Transmitted to the Line

3.3.3 Analyzing Generated Current Harmonics

With this model, the effects of the harmonics generated by the rectifier system can be simulated. The graphs in Figure 23 below show the current on the stator side, the non-linear current drawn by the rectifier system, and the current seen by the transformer, respectively.

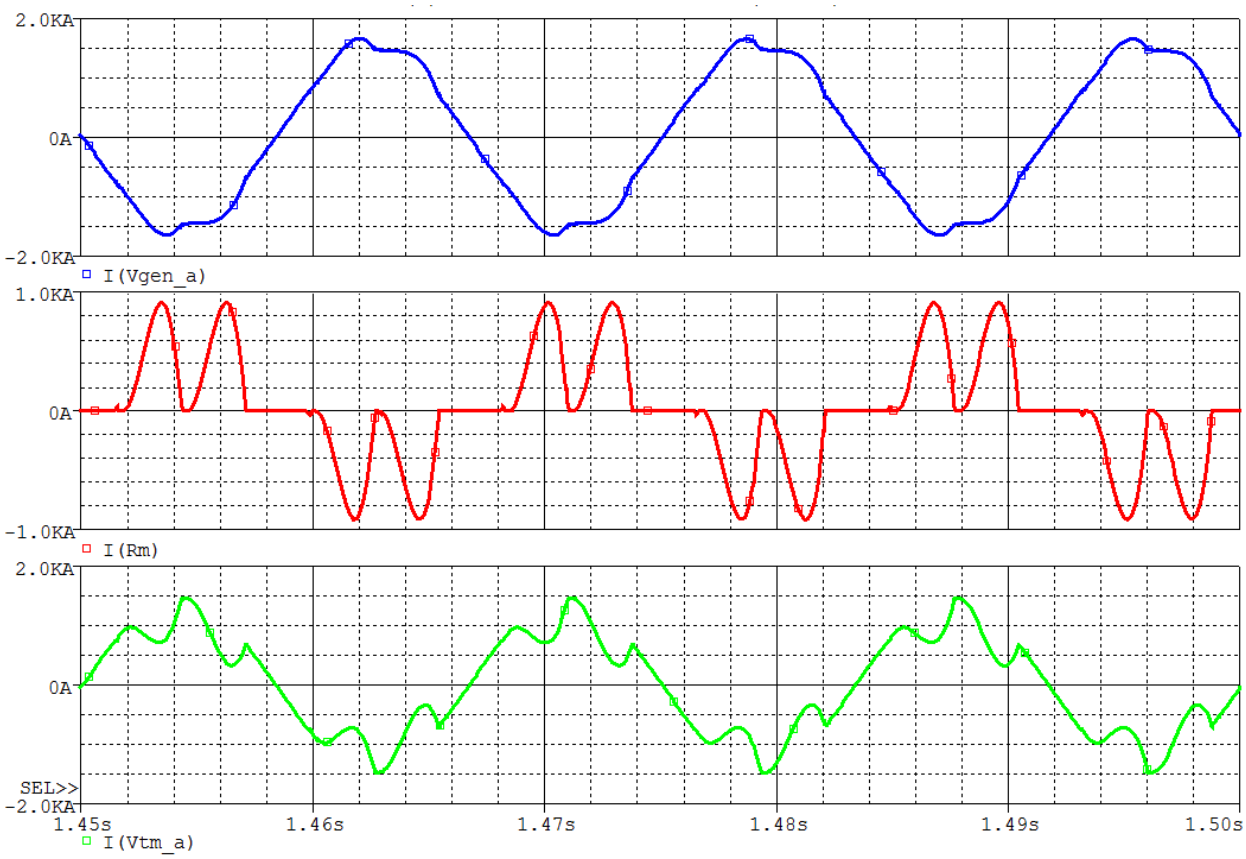


Figure 23: Stator, Rectifier, and Output Currents

To better understand the parameters of the system which generate these harmonics, different impedances were varied and the effects on the harmonics recorded. The results of the study are shown in the Figure 24 and Figure 25 below:

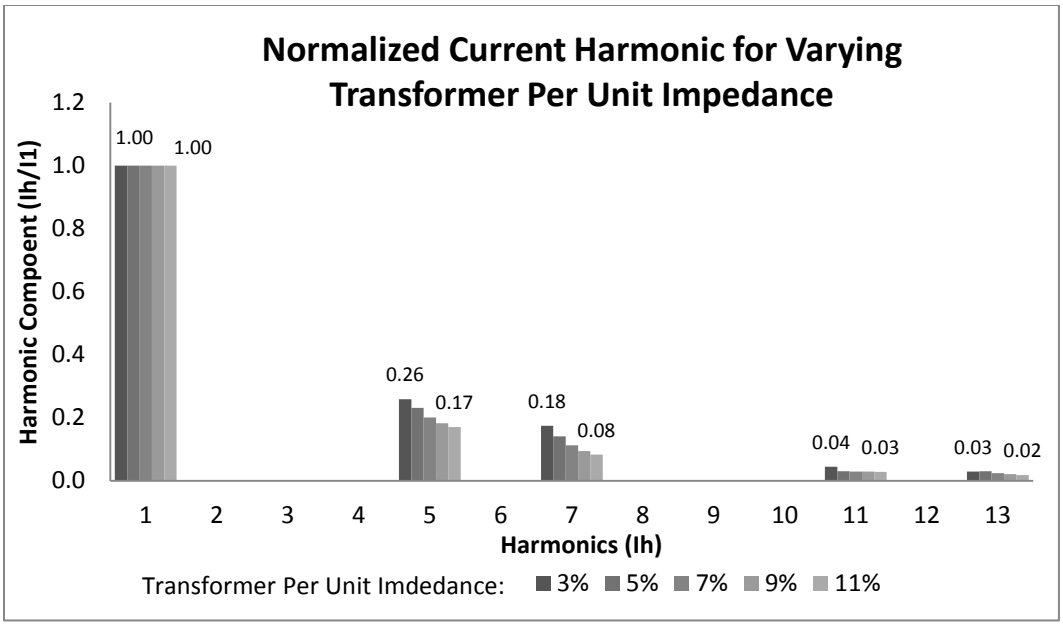


Figure 24: Normalized Current Harmonic for Varying Transformer Per Unit Impedance

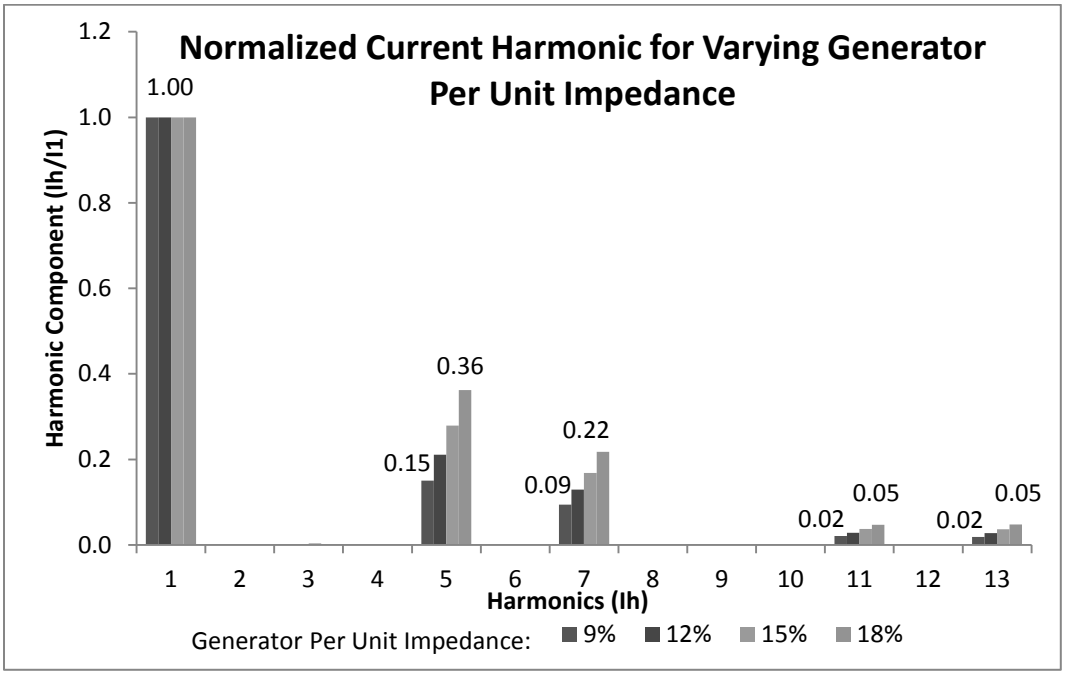


Figure 25: Normalized Current Harmonic for Varying Generator Per Unit Impedance

By increasing the per unit impedance of the connected transformer, the generated current harmonics is reduced. The THD ranges from 19.1% with transformer per unit impedance of 11% to 32.0% THD for transformer per unit impedance of 3%. Also, for generators with a higher per unit impedance, the current harmonics become more pronounced. The THD ranges from 17.7% with generator per unit impedance of 9% to 42.8% THD at generator per unit impedance of 18%.

3.4 Active Filter

The active filter design consists of three major sub-sections: the sensors, the controller, and the H-Bridge. Four sensors are presented although only three are used in this particular design. The first sensor is the non-linear current sensor. The second, based off the same design, is the correction current sensor. The third is a line voltage sensor used to keep track of the timing and synchronization of the injected current. The fourth sensor that is not implemented is for managing the capacitor voltage that is used to power the H-Bridge. In the current design, the capacitor is substituted by a 30V DC power supply. The second sub-section is the controller. This is basically the STM32VL-Discovery evaluation board. The third sub-section is the H-Bridge circuit that actually drives the current to the line. The block diagram for the system is shown below in Figure 26.

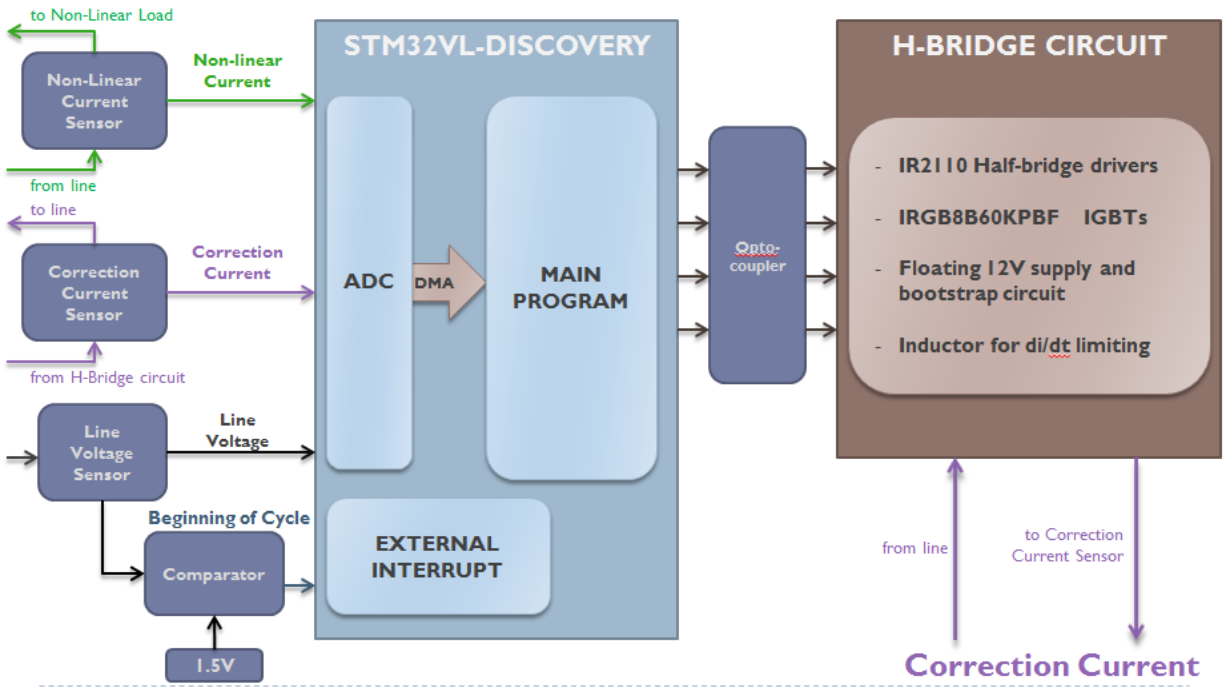


Figure 26: Active Filter Block Diagram

3.4.1 Switching Scheme

In order to know which switches to toggle, there are three different conditions to consider: the line voltage, the target current, and the actual current minus the target current. With these three conditions we can toggle the switches in a way to control the system. The basic truth table is seen below in table 16 where A is $V_{in}>0$, B is $I_{target}>0$, and C is $I_{actual} - I_{target}>0$.

A	B	C	S1	S2	S3	S4	Current	Reason
0	0	0	0	0	0	0	Decrease	Charging
0	0	1	0	1	1	0	Increase	Shorting
0	1	0	1	0	0	1	Increase	Discharging

0	1	1	0	0	0	0	Decrease	Charging
1	0	0	0	0	0	0	Decrease	Charging
1	0	1	0	1	1	0	Increase	Discharging
1	1	0	1	0	0	1	Increase	Shorting
1	1	1	0	0	0	0	Decrease	Charging

Table 16: IGBT Switching Logic

This gives a logic output of (B & ~C) for S1 and S4 and a logic output of (~B & C) for S2 and S3. It turns out that the line voltage does not affect the way the system operates. Figure 27 below shows a PSPICE simulation of this logic system where the first waveform is the line voltage at 60Hz, the second is an arbitrary target current of 300Hz, and the last is the system output current matching the arbitrary target current.

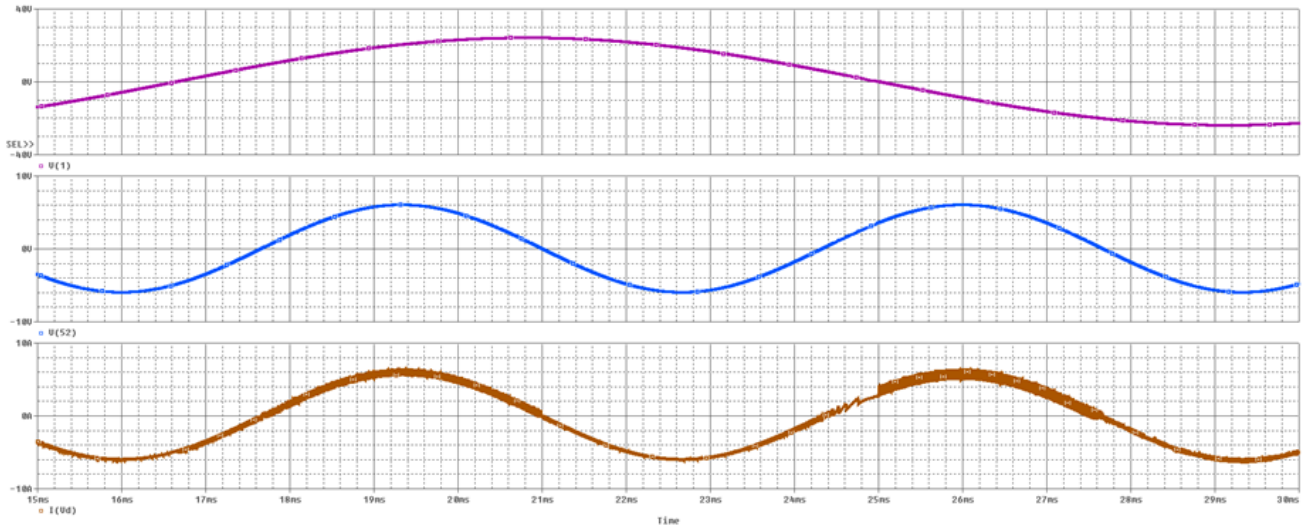


Figure 27: PSPICE IGBT Logic Verification

3.4.2 Voltage Sensing Circuit

For the test signal to be in phase, the voltage on the line will need to be known to the controller. However, the voltage is too high to simply tap directly off the line. The since the microprocessor used in this project can only handle 0-3.3 volts. To solve this problem the use of a step down transformer and some analog circuitry will be made use of, which can be seen in Figure 28 below.

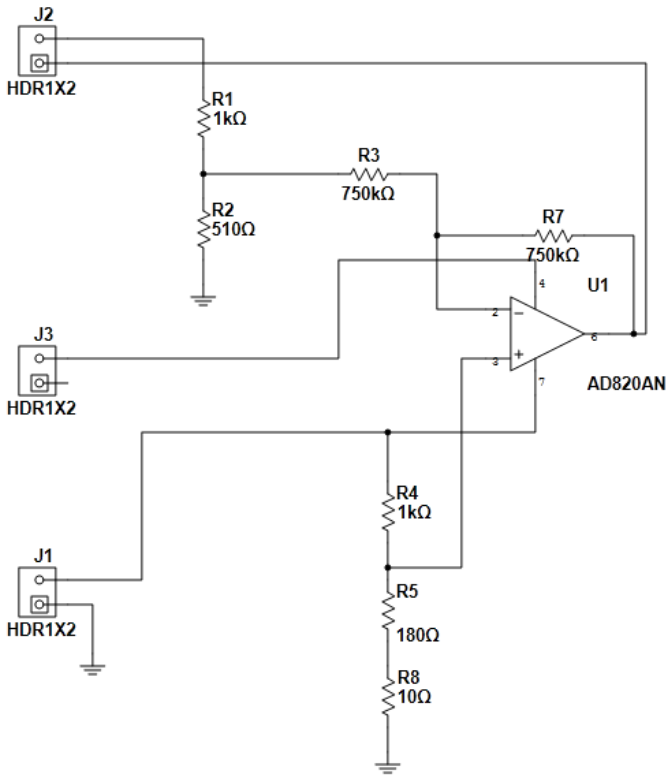


Figure 28: Voltage Sensor Circuit

This circuit makes use of an AD820AN operational amplifier which uses ±5V rails. The positive input was biased in the bottom left circle using a voltage divider to make the waveform positive. A voltage divider was also used for the input off the line to make sure it was 3.3V peak to peak. This

circuit actually inverts the waveform, but is accounted for in the processing of the signal. Common resister sizes were used to make this design practical.

The design of this circuit was created for a scale model of the DFIG system with a line voltage of 12VAC. Therefore, if this design were to be adopted for a full power realization of this design, the biasing would have to be altered to still create an acceptable output for a microprocessor. However, the overall theory and design will hold true.

3.4.3 Current Sensing Circuit

A current sensing circuit was designed in order to continuously monitor the AC current of the line. The initial design uses a Hall Effect sensor IC (ACS714) [14] to measure the line current. The IC possesses an optimized current measurement range of ±20 A and an output voltage that is proportional to AC or DC currents. The output of the Hall Effect sensor is biased at a voltage equivalent to $V_{CC}/2$, which for this design is 2.5 V. The output sensitivity is equal to 100 mV/A. The sensor output voltage must go through a system of op-amps that will reduce the values so that they can be entered and stored in the ADC. The op-amp system consists of an inverting op-amp that will reduce and invert the input voltage and another inverting op-amp that will act as an inverting buffer. The schematic of this circuit is shown in the figure below.

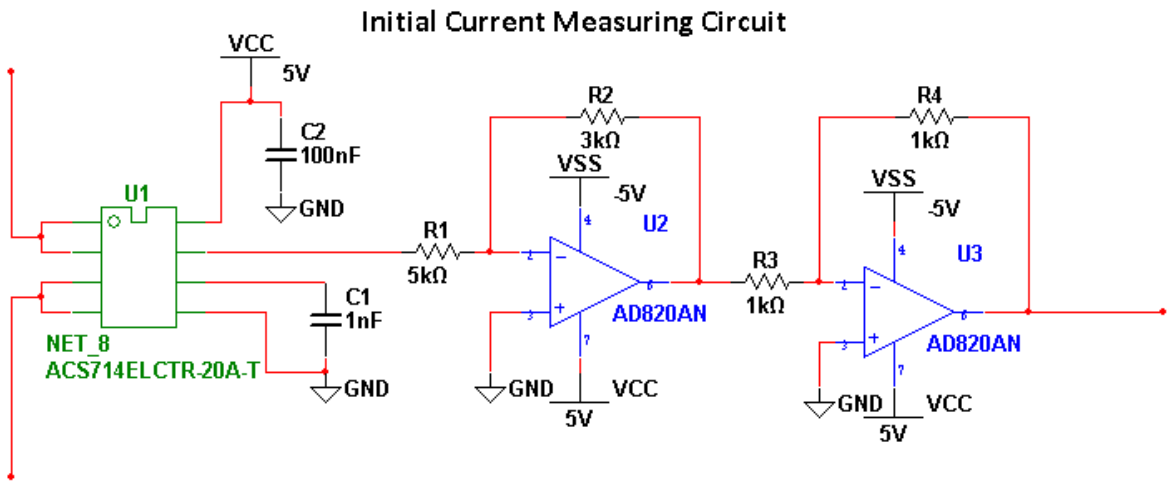


Figure 29: Initial Current Measuring Circuit

The ACS714 Hall Effect sensor was listed in the datasheet to have a 1.5 % total output error. A set of test measurements were performed to confirm that this would be the typical error of the sensor. A basic circuit was created using a 15 Ω resistor in series with a voltage supply and ground. The voltage supply ranged from 2.5 V to 8.5 V and was stepped up by 0.5 V each test. The output of the current sensing circuit was measured and compared with the calculated or expected output to determine the error percentage. The expected output was determined by a set of calculations. First, the current flowing through the basic load was calculated by using ohm's law. Next, the output sensitivity (100 mV/A) and the Hall Effect output were calculated. For example, when the current is 1 A, the output is the sum of the sensitivity (100 mV) and the output voltage bias (2.5 V). Thus, the result is an output of 2.6 V. After this calculation, the output is run through an inverting op-amp, which has a gain of –R1/R2. The values for R1 and R2 are 3 kΩ and 5 kΩ, so the gain is -0.6. Finally, the output of the inverting op-amp is run through an inverting buffer, which has a gain of -1, so that all values can then be entered into the ADC.

Current Sensing Circuit Test Measurements			
Hall Effect: 5 V Power Supply Output Bias: VCC/2			
15 Ω Resistor Load			
Hall Effect Output Sensitivity 100 mV/A			
Test Voltages at 2.5 V to 8.5 V with a 0.5 V step			
Test Voltage (V)	**Calculated Output Voltage (V)**	**Measured Output Voltage (V)**	**Error (%)**
2.5	1.51	1.495	0.9934
3	1.512	1.496	1.0582
3.5	1.514	1.4973	1.103
4	1.516	1.4987	1.1412
4.5	1.518	1.4997	1.2055
5	1.52	1.5016	1.2105
5.5	1.522	1.5027	1.2681
6	1.524	1.5043	1.2927
6.5	1.526	1.505	1.3761
7	1.528	1.5076	1.3351
7.5	1.53	1.5094	1.3464
8	1.532	1.511	1.3708
8.5	1.534	1.512	1.3055

Table 17: Hall Effect Current Test Measurements

The test measurements verified that the circuit was correctly built because the error was around 1.5 %. The figure below demonstrates that the error produces a lower voltage than the calculated output voltage. The Measured output voltage is not perfectly linear like the calculated voltage, but it does increase in value as the test voltages is brought up.

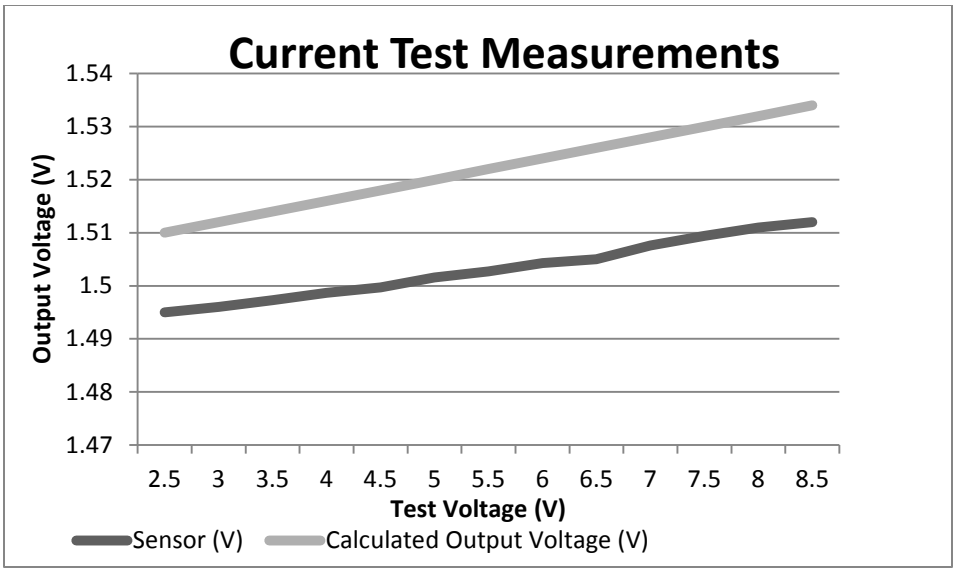


Figure 30: Hall Effect Current Test

It was determined that the active filter would require two current sensors. One was needed to measure the current in the line while the other was needed to measure the current harmonics of the system. The difference between these two currents allows the microcontroller to determine the injection current that is needed to remove the harmonics from the system. Therefore, two current sensors were built both by means of a Hall Effect sensor and an op-amp. The same Hall Effect IC sensors and op-amps were used to create the two circuits since they would require the same operating conditions.

A final current measuring circuit was designed to substitute for the initial design. This circuit, which is shown below, was created in order to reduce the number of components necessary for the sensor. The second op-amp, which acted as an inverting buffer is removed from the circuit and the output of the first op-amp is biased before it is sent into the ADC. The output of the first inverting op-amp is biased in order to produce a non-negative voltage for the ADC. Additional capacitors and resistors were connected to the dual-power supply and the output of the op-amp in order to reduce

error and improve the Signal-to-noise ratio (SNR). The voltage bias is constructed by means of a voltage divider that is connected to the V_+ side of the op-amp.

$$VBias = \frac{V_O + GV_I}{(1+G)}$$

Equation 22: Opamp Bias Voltage

The voltage bias for an inverting amplifier is calculated by using the equation shown above. When the Hall Effect senses zero current, the output is equal to the bias of the sensor (2.5 V). The parameter G, which is equal to 3/5, represents the ratio between the resistors R2 and R1. The output voltage was chosen to be biased at 1.5 V. With V_I set at 2.5 V and V_O set at 1.5 V, the voltage bias was determined to be 1.875 V. A voltage divider consisting of a 30 Ω and 18 Ω resistor series was constructed to produce this voltage bias.

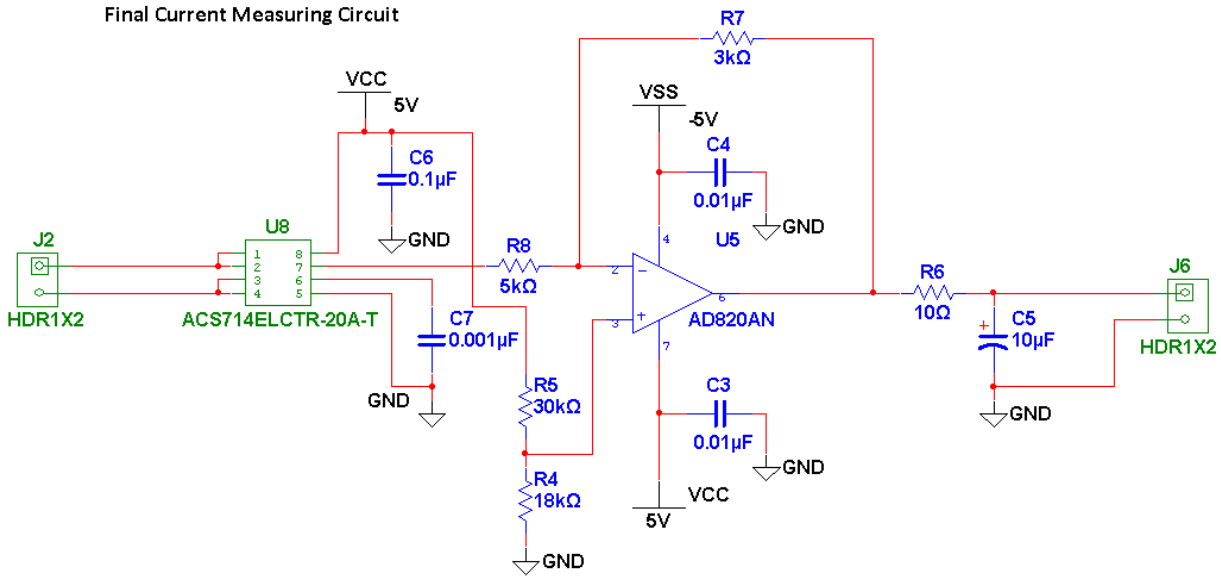


Figure 31: Final Current Measuring Circuit

A set of test measurements were also performed for the final design to observe the functionality of the circuit and the amount of error reduction in the output voltage. The same 15 Ω resistor and power supply were used for these test measurements. The voltage supply ranged from 3.5 V to 8.5 V and was stepped up by 0.5 V each test. The output voltage and error percentage were calculated for both sensors. The two sensors both produced a significantly lower error percentage than the one tested in the original design.

Current Sensing Circuit Test Measurements					
Measured with Agilent Multimeter					
5.2 Ω Resistor Load					
Test Currents at 0 to 1.5 A with a 100 mA step					
Current (A)	**Calculated Output Voltage (V)**	**1st Sensor (V)**	**Error (%)**	**2nd Sensor (V)**	**Error (%)**
0	1.5	1.515	1	1.504	0.267
0.1	1.494	1.509	1.004	1.499	0.335
0.2	1.488	1.504	1.075	1.493	0.336
0.3	1.482	1.498	1.08	1.487	0.337
0.4	1.476	1.492	1.084	1.481	0.339
0.5	1.47	1.487	1.157	1.476	0.408
0.6	1.464	1.481	1.161	1.47	0.41
0.7	1.458	1.476	1.235	1.464	0.412
0.8	1.452	1.47	1.24	1.459	0.482
0.9	1.446	1.464	1.245	1.453	0.484
1	1.44	1.459	1.319	1.447	0.486
1.1	1.434	1.453	1.325	1.441	0.488
1.2	1.428	1.447	1.33	1.436	0.56
1.3	1.422	1.442	1.407	1.43	0.563
1.4	1.416	1.436	1.412	1.424	0.565
1.5	1.41	1.43	1.418	1.418	0.567

Table 18: Current Sensing Circuit Test Measurements (0 to 1.5A)

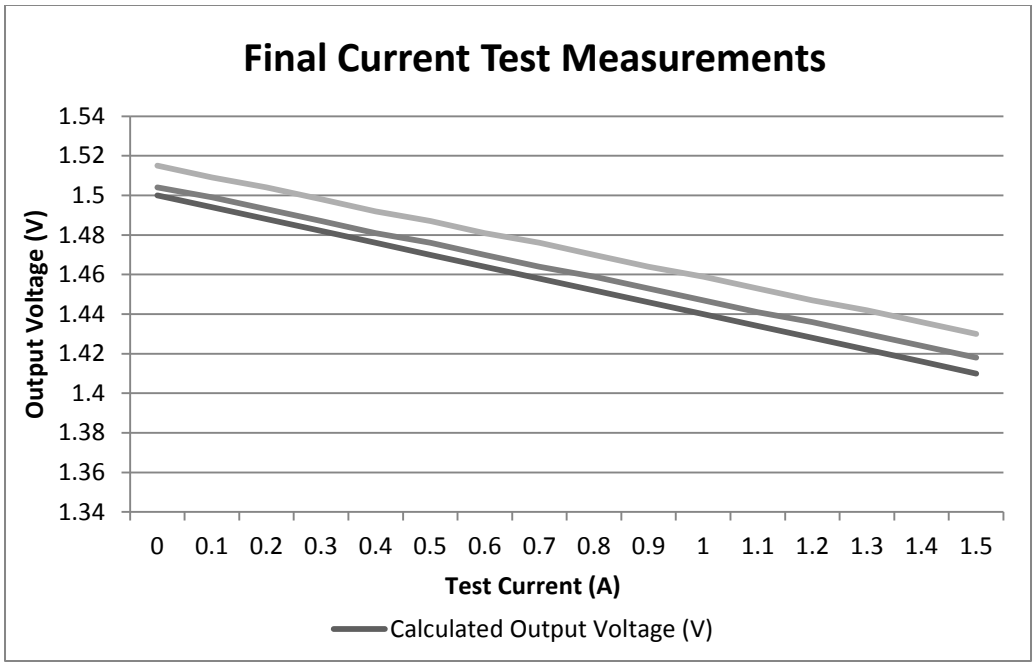


Figure 32: Final Current Measurements (0 to 1.5A)

Current Sensing Circuit Test Measurements					
Measured with Agilent Multimeter					
5.2 Ω Resistor Load					
Higher Test Currents at 2 to 5.5 A with a 0.5 A step					
Current (A)	**Calculated Output Voltage (V)**	**1st Sensor (V)**	**Error (%)**	**2nd Sensor (V)**	**Error (%)**
2	1.38	1.403	1.667	1.393	0.942
2.5	1.35	1.375	1.852	1.365	1.111
3	1.32	1.346	1.97	1.336	1.212
3.5	1.29	1.318	2.171	1.307	1.318
4	1.26	1.29	2.381	1.279	1.508
4.5	1.23	1.261	2.52	1.25	1.626
5	1.2	1.233	2.75	1.221	1.75
5.5	1.17	1.204	2.906	1.193	1.966

Table 19: Current Sensing Circuit Test Measurements (2 to 5.5A)

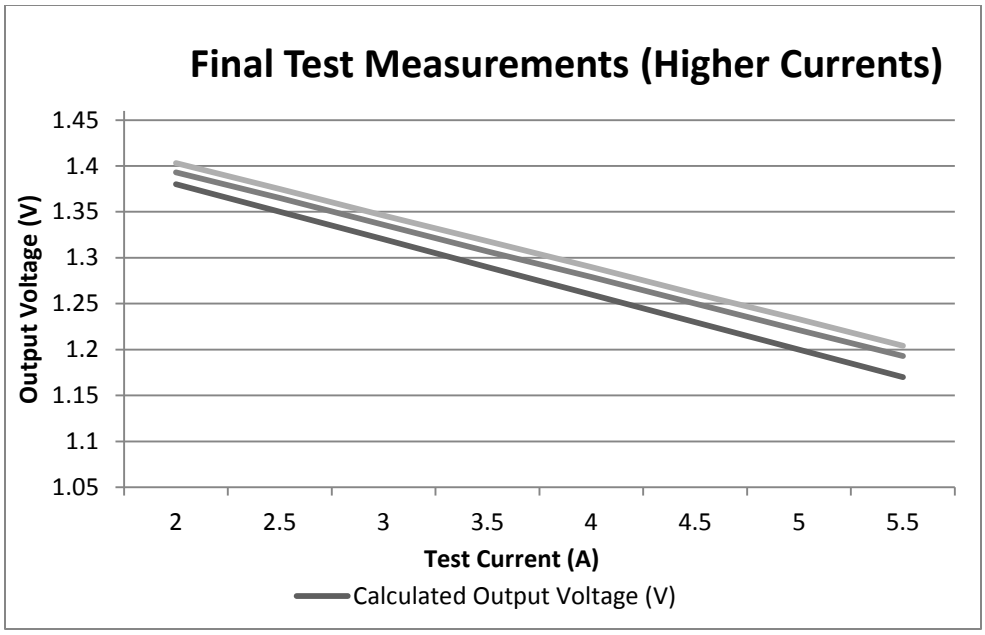


Figure 33: Final Current Measurements (2 to 5.5A)

3.4.4 Power Supply Circuit

A power supply circuit was designed in order to power the active filter circuit by means of the AC wall line. The power supply must support voltages of ± 5 V and + 12 V. This design uses two voltage transformers: a 115 to 12.6 V center-tapped transformer and a 115 to 18 V transformer. The transformers are necessary for reducing the voltage of the AC line in order to be run through the regulators. Bridge rectifiers are required in order to convert the AC voltages into DC before the voltage line values are input into the voltage regulators. The first two regulators used in this circuit include the LM7805 and LM7905, which regulate the higher input voltages and produce the ± 5 V outputs. The LM7812 is the other regulator and receives an input voltage from the higher rated transformer in order to produce the + 12 V output.

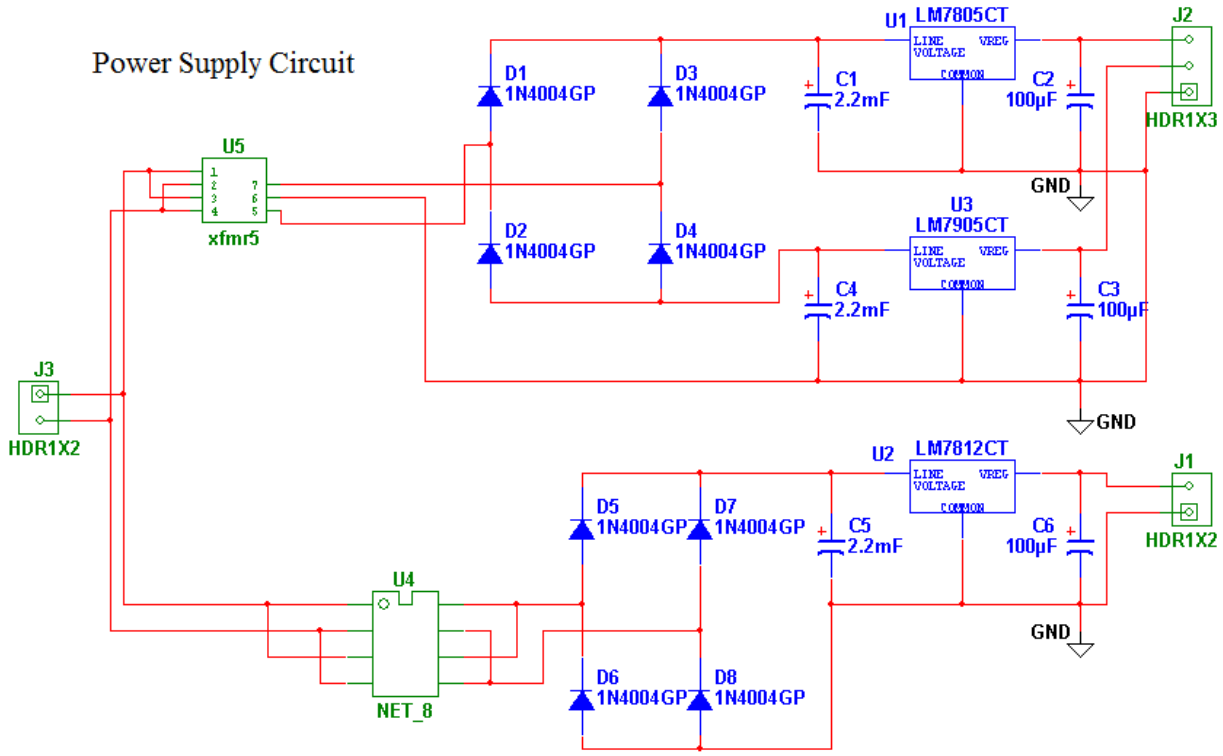


Figure 34: Power Supply Circuit

The power supply circuit was simulated on Multisim and tested on a protoboard to confirm its accuracy. The simulation was tested with 50 Ω resistors and two virtual transformers with fairly similar turns ratios. A 13:1 and a 6.5:1 were used with a 120 VRMS AC Power Supply to produce the AC voltages. The outputs of the two transformers were around 9.23 V and 18.46 V and therefore were large enough to be input into the regulators. The result of the circuit was that it produced output voltages of 4.96 V, -4.8 V and 11.8 V. Since these results were very accurate, the design was then tested on a protoboard.

The protoboard design included the 2.2 mF and 100 uF capacitors that were used in the design shown above. The regulators were all tested with a 15 Ω resistor. The transformers were connected to a 120 VRMS AC outlet. The protoboard design produced output voltages of 4.94V, -4.99 V and 11.8 V. The

test measurements proved that the circuit is designed right and successfully outputs the right values for our power supply.

3.4.5 Capacitor Voltage Sensor Circuit

A voltage sensing circuit was designed in order to monitor the capacitor voltage and determine when to charge and discharge the capacitor. The design uses an analog optocoupler (HCNR201) [15] to create electrical isolation between the input and output of the sensing circuit. Optocouplers prevent rapidly changing voltages from harming or destroying any other components that are connected to the system. The voltage sensing circuit, which is displayed in the figure below, consists of two op amps and two resistors along with the optocoupler IC.

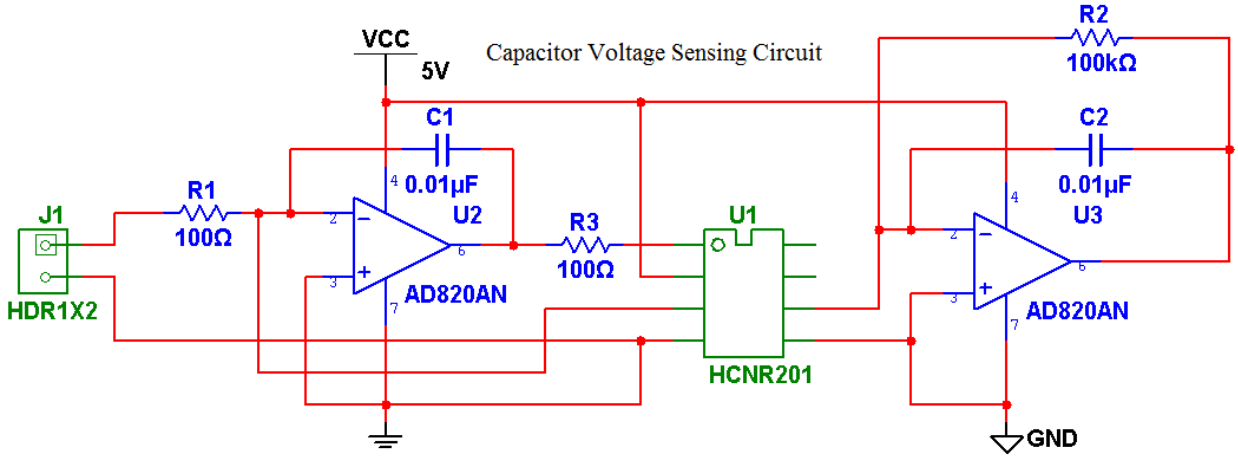


Figure 35: Capacitor Voltage Sensor

The first op-amp is assumed to be a perfect op-amp so that all current flows through the R_1 resistor into the PD1 photodiode located in the IC. When a positive input voltage is applied, the op-amp output moves in the direction of the negative rail causing current to flow through the LED. The output of the LED is detected by the photodiode PD1 and as a result PD1 generates I_{PD1}. Since the current is

assumed to flow only through the resistor, the photodiode current I_{PD1}, shown in the equation below, is calculated by Ohm's Law.

$$I_{PD1} = {}^{V_{IN+}}/_{R_1}$$

Equation 23: Photodiode Current

PD1 and PD2 are equivalent photodiodes and therefore produce a transfer gain K_3 which is determined to be I_{PD2}/I_{PD1} = 1**.** The second op-amp connected with the R_2 resistor act as a trans-resistance amplifier that produces the analog output V_{OUT}, which is calculated in the equation below.

$$V_{OUT} = I_{PD2} \text{ x } R_2$$

Equation 24: Vout

These two equations combined together produce the formula below, which illustrates the linear relationship between the input and output voltage. Since K_3 is essentially unity gain, the relationship is determined by the ratio of the two resistors.

$${}^{\text{VOUT}}/_{\text{VIN}+} = K_3 \text{ x } ({}^{R_2}/_{R_1})$$

Equation 25: Gain

Voltage Measured with Agilent Multimeter			
Test Voltages at 0 to 50 V with a 5 V step			
Capacitor Voltage (V)	Input Divider Voltage (V)	Cap Sensor (V)	Error (%)
0	0	0.003	–
5	0.3	0.29	-3.33
10	0.6	0.58	-3.33
15	0.9	0.87	-3.33
20	1.2	1.161	-3.25

25	1.5	1.451	-3.267
30	1.8	1.741	-3.278
35	2.1	2.03	-3.33
40	2.4	2.32	-3.33
45	2.7	2.61	-3.33
50	3	2.9	-3.33

Table 20: Capacitor Sensor Test

The test measurements proved that the circuit was correctly built because the input/output voltage relationship was linear. The difference between the test voltage and the sensor voltage grew after each voltage step as the test voltage was increased. The average error of this test measurement was calculated to be 3.248%.

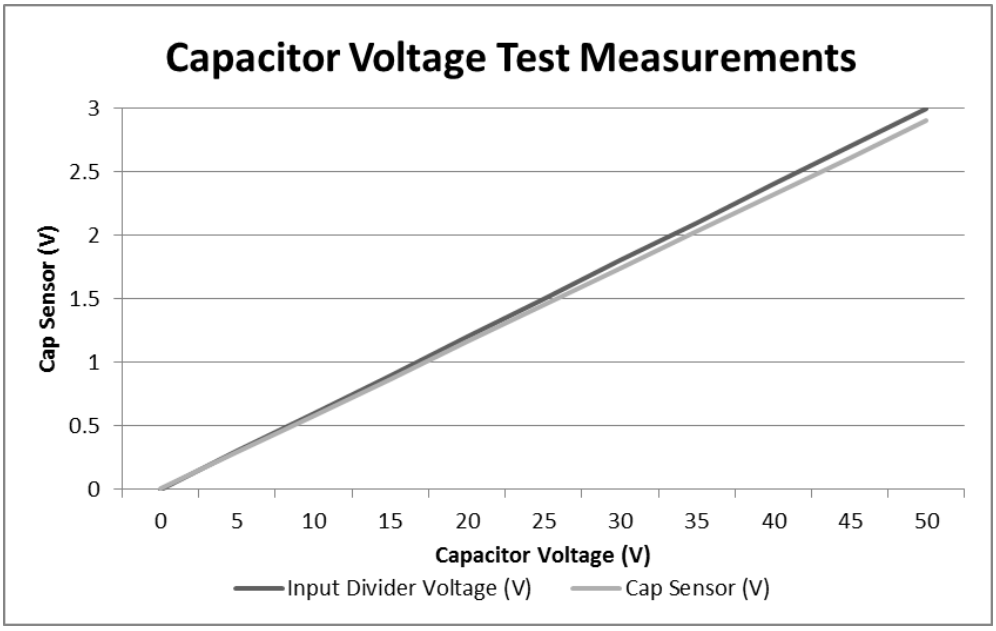


Figure 36: Capacitor Output Voltage

The optocoupler would successfully output the analog input voltage as long as the power supply of the op-amps was greater than or equivalent to the input voltage. In order to power the op-amps with a small power supply and measure a large input voltage range, a voltage divider is attached to the

capacitor. A capacitor voltage of 50 V is reduced to 3 V, so that all necessary voltages can be measured by the circuit. The op-amps and optocoupler are powered by a 5 V supply voltage.

3.4.6 H-Bridge Inverter Circuit

A practical H-Bridge based inverter needed to be designed in order to create a source of correction current to be injected into the line. The circuit is designed to create a maximum injection current of 10A. Four IRGB8B60KPBF IGBTs are used for the H-Bridge switching at 24kHz. These IGBT's have a continuous collector current rating of up to 19A at 100^{o}C junction temperature. The $I_{collector}$ vs V_{ce} graph shows a V_{ce} of approximately 2.2V for a V_{ge} of 12V. In order to drive the IGBT gates, a V_{ge} of 12V needs to be supplied. Isolated power supply for each IGBT proves an expensive an impractical option. Instead a bootstrap based design was chosen. The IR2110 half bridge driver was picked because it can drive both the high and low side IGBTs while only needing one isolated 12V supply. It is also able to drive up to 2A of gate drive current. MCT-6 opto-couplers are used to isolate microcontroller signals from the driver, thus providing complete galvanic isolation between the power circuitry and the logic hardware. The circuit diagram below shows the design of this H-Bridge:

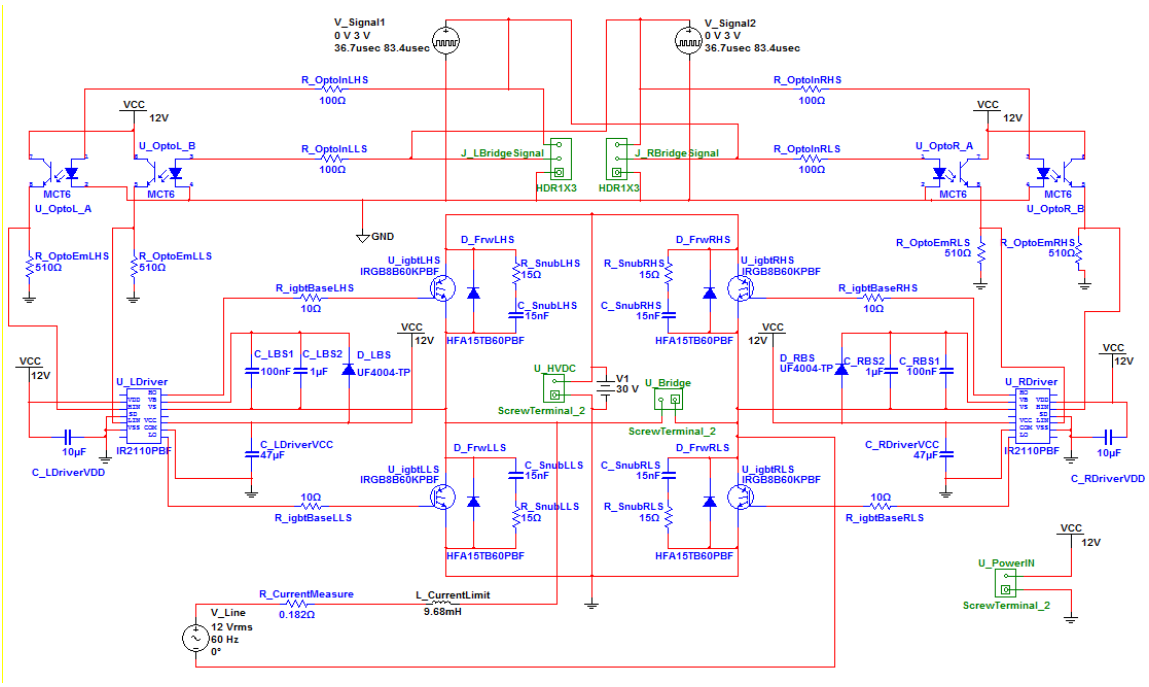


Figure 37: H-Bridge Circuit Schematic

Two square wave signals of opposite polarity are used to drive the circuit. They are set to be high for 1/(24kHz) – 5us (dead time) = 36.7us and with a period of 2*(1/24kHz) = 83.4us. In this design, only one 12V supply is needed to power both the opto-couplers and the two IGBT drivers because of the bootstrap topology. The bootstrap circuit works by charging up the bootstrap capacitors (C_LBS and C_RBS) through the bootstrap diode (D_LBS and D_RBS) when the opposite IGBT pair gets turned on. The capacitors then reach 12V above line voltage. This stored charge is used to drive the IGBT when the state switches. In order to have the bootstrap circuit function correctly, the capacitor size must be large enough to store the charge required to drive the high side IGBT. A film type capacitor was chosen because of its low leakage current. The bootstrap diode is an ultra-fast recovery diode also chosen to minimize leakage current. Component values were determined using *International Rectifier Design Tip 04-4: Using Monolithic High Voltage Gate Drivers* [16] as a guide. The following equation is used to determine the capacitor size:

$$C_{BOOT_{min}} = \frac{Q_{TOT}}{\Delta V_{BS}}$$

Equation 26: Minimum Bootstrap Capacitor Value

Where

$$Q_{TOT} = Q_G + Q_{LS} + \left(I_{LK\_GE} + I_{QBS} + I_{LK} + I_{LK\_DIODE} + I_{LK\_CAP} + I_{DS-}\right) \cdot T_{HON}$$

Equation 27: Total Charge for Bootstrap Capacitor

Q_G : IGBT turn on required Gate charge (IRGB8B60KPBF - 29nC)

I_{LK_GE} : IGBT gate-source leakage current (IRGB8B60KPBF - max. 100nA)

I_{QBS} : Floating section quiescent current (approximate as 0)

I_{LK} : Floating section leakage current (approximate as 0)

I_{LK_DIODE} : Bootstrap diode leakage current (UF4004GP – max. 100uA)

I_{DS-} : Desat diode bias when on (approximate as 0)

Q_{LS} : Charge required by internal level shifters (approximate as 0)

I_{LK_CAP} : Bootstrap capacitor leakage current (approximate as 0 since we are using a non-electrolytic capacitor)

T_{HON} : High side on time (24KHz switching speed gives an average on time of 41.7uS)

Thus $Q_{TOT} \approx 33nC$.

And

$$\Delta V_{BS} \leq V_{CC} - V_F - V_{GE_{min}} - V_{CE_{on}}$$

Equation 28: Bootstrap Capacitor Ripple Voltage

V_{CC} : IC voltage supply (12V)

V_F : Bootstrap forward voltage drop (1.3V)

V_{CEon} : IGBT operating collector-emitter voltage (2.2V)

V_{GEmin} : Minimum maintained gate-emitter voltage ($V_{GE(th)}$ = 4.5V)

Thus $\Delta V_{BS} \leq 4V$

Using equation x, the minimum bootstrap capacitor value is 8.29nF. However, a 1uF capacitor was chosen because the higher capacitance guarantees better performance when the switching time is longer. It will also stay at the charged voltage for a longer time while not significantly affecting the circuit in other respects.

$$R_G \approx \frac{V_{cc} - V_{ge}}{I_{avg}}$$

Equation 29: Determining Gate Resistor Value

Where

$$I_{avg} = \frac{Q_{gc} + Q_{ge}}{t_{sw}}$$

Equation 30: Average Gate Current

From the IRGB8B60KPBF datasheet, we know that Q_{gc} = 14nC and Q_{ge} = 3.7nC. Thus Iavg = 0.4248A and $R_G \approx 17.7\Omega$. A value of 10Ω was tested and found to work. (IRF DT04-4 revA)

The inductor in the circuit is used to limit di/dt. The value is set to 9.68mH, which is the value of the inductor available on hand that is of a workable size. In order to minimize the voltage spike on the IGBT caused by inductive kickback, the HFA15TB60PBF ultrafast soft recovery diode is placed between each IGBT'S common and emitter terminals. An RC snubber is also placed between the common and emitter to further reduce any remaining spikes [17]. The IR2110 SPICE model from the International

Rectifier website was used in this simulation. Below is the Multisim simulated graph of Vce and Vge of the left high side IGBT:

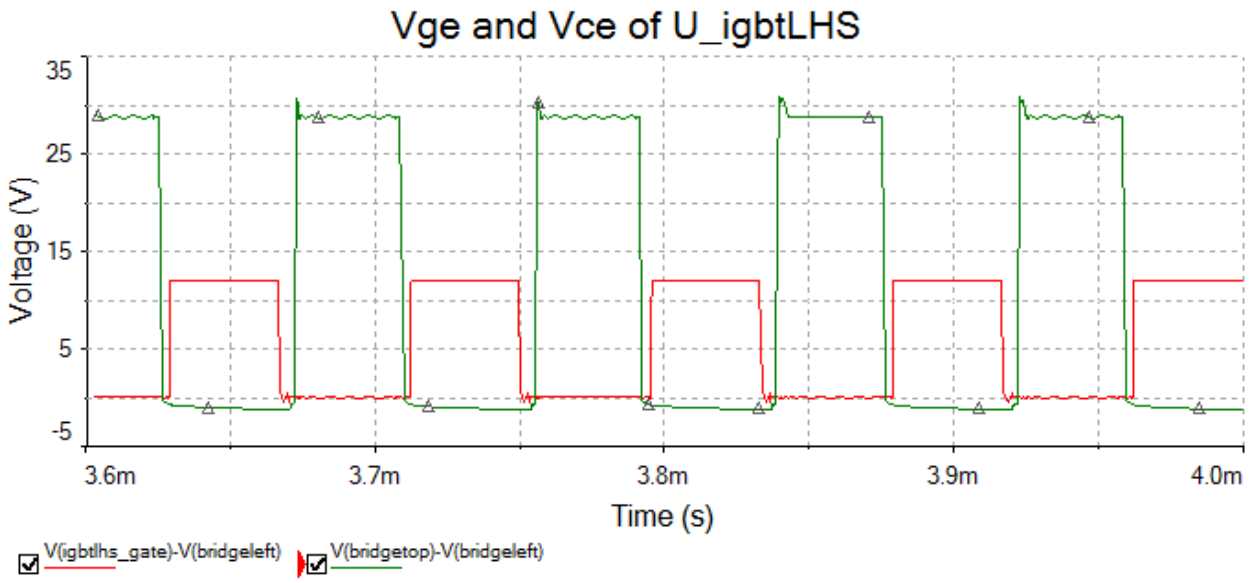


Figure 38: Vge and Vce of Left High Side IGBT

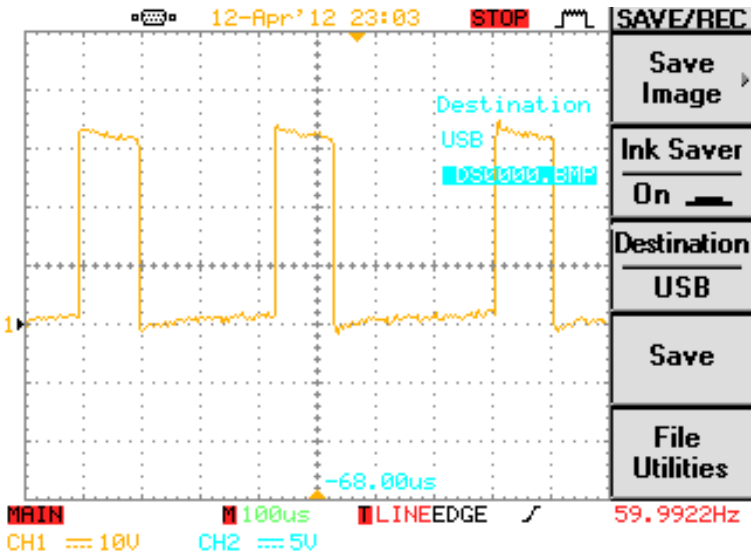


Figure 39: Actual Measured Vce on Left High Side IGBT

From Figure 38 above, Vge is at 12V, which is the voltage the IGBT gate is set to be driven at. The Vce graph shows very little spiking and oscillation. This is bacuse of the freewheeling diode and snubber circuit. Figure 39 shows the actual measured Vce, with waveform similar to that seen in the simulation.

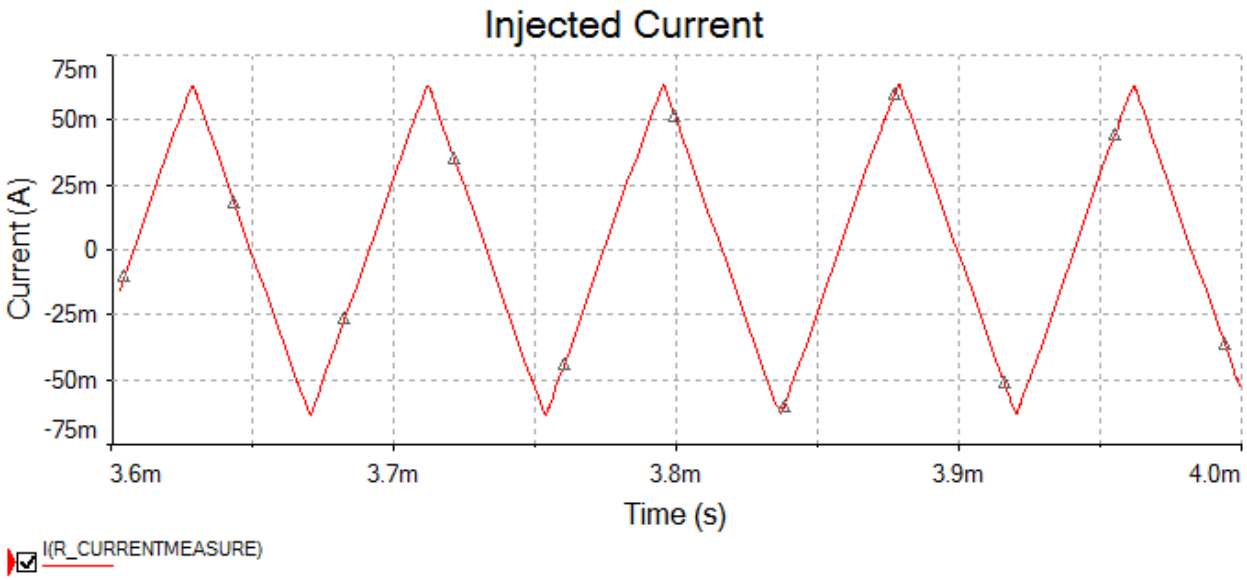


Figure 40: H-Bridge Injected Current (no line voltage aaplied)

Figure 40 above shows the simulated injection current with the source voltage turned off. The di/dt is at 3A/mS.

3.4.7 Microcontroller

The microcontroller used is the STM32VL-Discovery evaluation board. This particular controller is used because of its speed (24MHz), price, availability of free compiler and IDE, and the ease of use of the provided ST Standard Peripheral Library.

The microcontroller's job is to collect the data value of the line voltage, non-linear current, and correction current, then calculate a target correction waveform and implement the switching logic with some dead-time between switching. The basic program flow is shown in Figure 41 below.

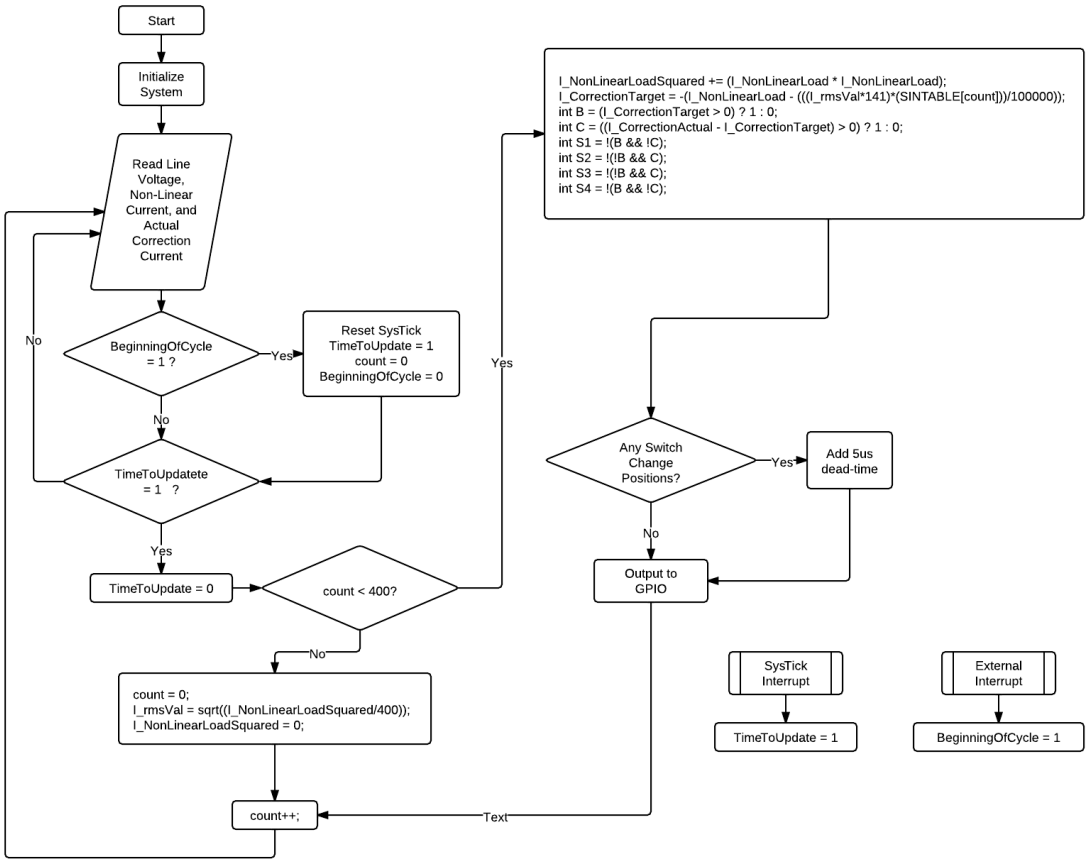


Figure 41: Basic Program Flow Chart

At the start of the program, the system goes through an initialization process where all the hardware peripherals are configured. This includes setting up the timers, setting the ADC in continuous conversion mode, and setting up the DMA vectors to bring converted ADC values directly to the global

variables in the main program. The SysTick timer is set to interrupt the program at 24kHz. This means that for a 60Hz cycle, SysTick will interrupt 400 times. The program is set to loop and continually update sensor values until either the TimeToUpdate flag is set, or when the beginning of a new cycle is detected. Each time SysTick interrupts, the TimeToUpdate flag is set, causing the program to go through the output logic and send out gate drive signals the GPIO pins if the count is less than 400. When it does this, the non-linear current value is squared and summed up with what was already in I_NonLinearLoadSquared. When the 400^{th} count is reached, the program divides that value by 400 and takes the square root of that number. This way, the RMS value of the previous cycle's non-linear current in obtained. The external interrupt is used to set the BeginningOfCycle flag and keeps the program synchronized to the line voltage.

The particular logic for determining the target current is fairly simple. With an RMS value obtained from the previous cycle, a reference sinusoid of the same magnitude can be generated with a sine table. The difference between the present non-linear load value and the reference sinusoid is the target current value. With both the actual correction current value and a target current value, the output for the switches can be determined according to the switching scheme in section 3.4.1. The program then checks if there have been any changes from the previous state and will add a 5us dead-time if that is the case. The dead-time prevents any possible shoot through current in the H-bridge circuitry. In this way, the microcontroller generates an appropriate correction current that will cancel out any of the lower current harmonics.

4 Results

This chapter will outline the results of the physical scale model that was designed and simulated in the previous chapter. To test the filter the grid side converter wars replicated using a rectification circuit. The circuit that was built can be seen in Figure 42.

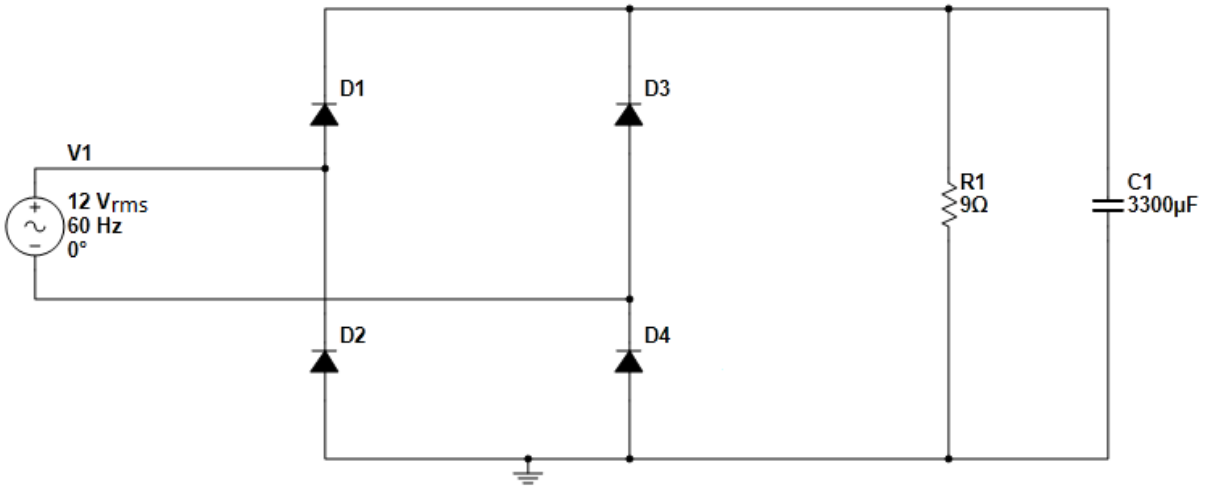


Figure 42: Rectification Test Circuit

This circuit was found to draw a non-linear current that can be seen in Figure 43.

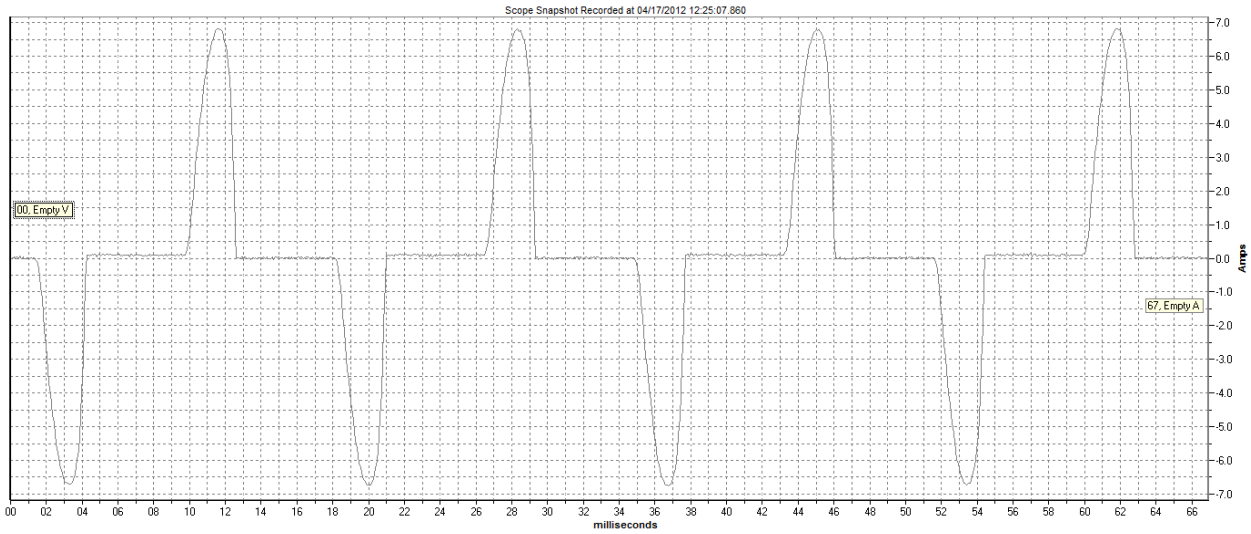


Figure 43: Non-Linear Current Drawn from Rectification

Using the Fluke 1750 to analyze the harmonics in this wave, it was found to have significant triplen harmonics, which can be seen in Figure 44.

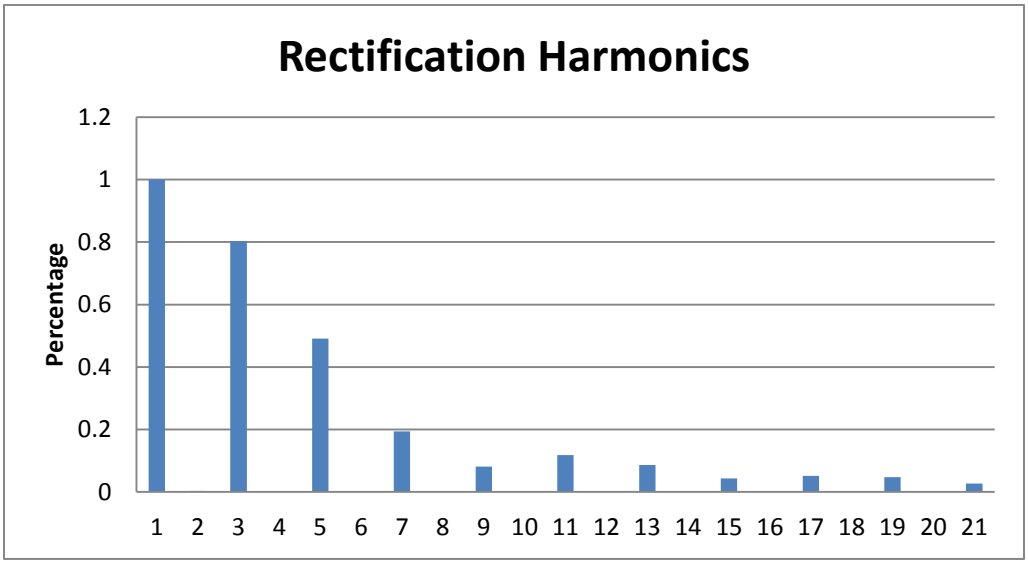


Figure 44: Rectification Harmonics

Seeing that the rectification process created a significant amount of harmonic current, the active filter was then connected to mitigate these harmonics. The resulting waveform can be seen in Figure 45.

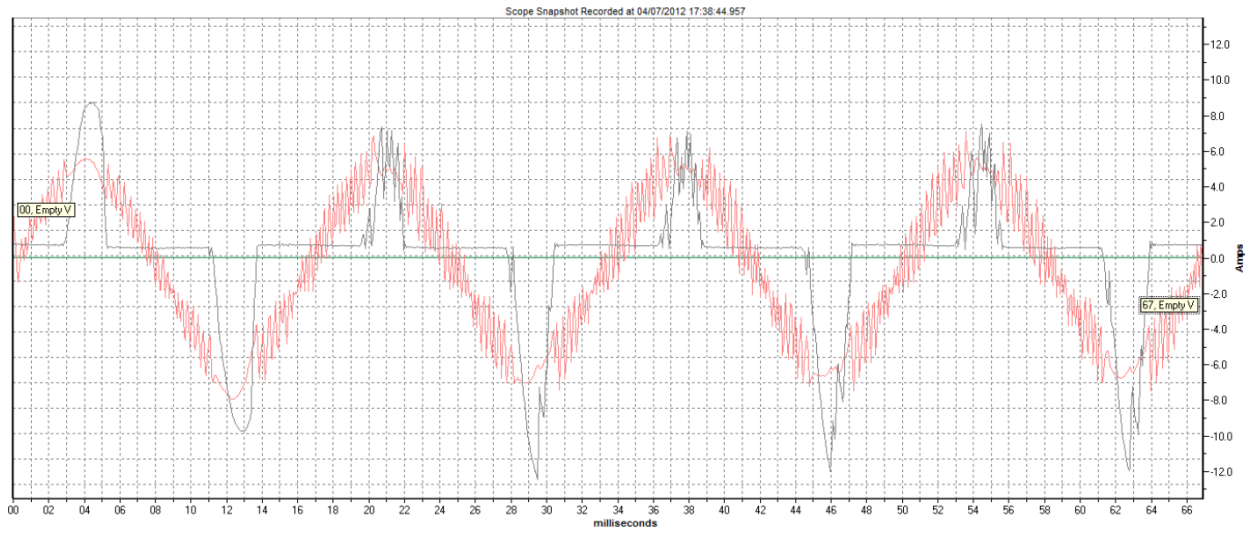


Figure 45: Active Filter Mitigation Oscilloscope

In the oscilloscope reading above the red waveform is the filtered current, and the black waveform is the non-sinusoidal current being pulled to the rectifier. It can be seen that the non-sinusoidal current has even harmonics, which may be due to a slight mismatch in component VI characteristics of the diodes. The noise on the red waveform is the switching from the filter. This is a function of the switching speed and the size of the inductance that is connecting the filter to the system. This can be reduced by increasing the switching speed, increasing the inductance, and/or installing a tuned passive filter. Looking at the harmonics present in this mitigated waveform, which can be seen in Figure 46, the filter has significantly reduced the harmonics, both even and odd.

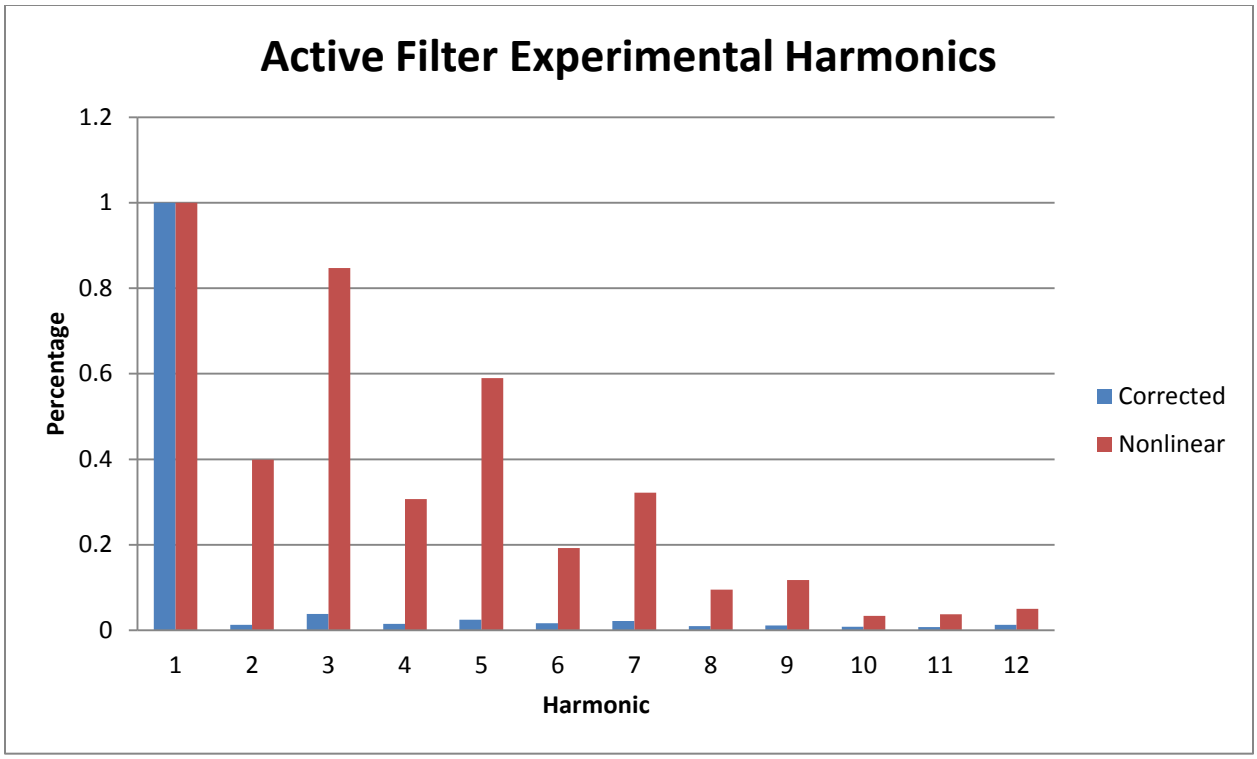


Figure 46: Active Filter Experimental Harmonics

This graph can be easily summarized by looking at the %THD and the even and odd harmonics. This can be seen in Table 21, where "A" is the non-linear current and "B" is the corrected current.

	A	B
% THD	123.8	13.4
% Odd Harmonic	110.0	9.7
% Even Harmonic	56.9	9.3

Table 21: Current Harmonic Comparison

It can be seen that the current harmonics are greatly reduced from 123.8% to 13.4%. Therefore, this system is working as designed to greatly reduce the current harmonics seen by the step up transformer that would be in an actual system.

5 Conclusions and Future Recommendations

Although this system is working as expected, there are possible improvements that can be made to the system. One improvement would be to replace the power supply that acted as the storage device for the H-bridge and replace it with a stand-a-lone source, such as a capacitor. This would require a charging circuit to ensure that the necessary power would always be available from the energy storage device to supply the system with the needed current.

Another possible improvement would be to eliminate the switching noise created by the system. These can be mitigated by either increasing the switching frequency or increasing the inductance that ties the system to the line. Both of these solutions would have to be balanced and can never fully mitigate the switching noise. The most practical solution may be a simple tuned passive filter.

Overall, this project was successful in the design and creation of an active filter to mitigate current harmonics. Simulations of a doubly fed induction generation (DIFIG) system were made to determine harmonic content. A reduced scale active filter design was created and simulated to mitigate current harmonics caused by a non-linear load. The scale model of the filter system was then built and was found to reduce both even and odd current harmonics by over 90%, therefore proving the concept. The model could then be scaled up in power and adapted for a three phase system.

6 Bibliography

[1] ASTM International, "D3612-02: Standard Test Method for Analysis of Gases Dissolved in Electrical Insulating Oil by Gas Chromotography," 2009.

[2] IEEE Power & Energy Society Transformers Committee, "C57.104-2008 - IEEE Guide for the Interpretation of Gases Generated in Oil-Immersed Transformers," 2008.

[3] IEEE Power & Energy Society Transformers Committee, "C57.106-2006 - IEEE Guide for Acceptance and Maintenance of Insulating Oil in Equipment," 2006.

[4] "Harmonics," WE Energies, USA, November 2009. [Online]. Available: http://www.we-energies.com/business/energyeff/harmonics.pdf.

[5] A. Elmoudi, M. Lehtonen and H. Nordman, "Effect of Harmonics on Transformers Loss of Life," *IEEE International Symposium on Electrical Insulation,* 2006.

[6] H. Wheeler, "Formulas for the skin effect," *Proc. IRE,* vol. 30, pp. 412-424, 1942.

[7] L. H. Dixon, "Eddy current losses in transformer windings and circuit wiring," 2003.

[8] N. Jayasinghe, J. Lucas and K. and Perera, " Power System Harmonic Effects on Distribution Transformers and New Design Considerations for K Factor Transformers," *IEEE Sri Lanka Annual Sessions,* 2003.

[9] C. D. Association, "Harmonics, Transformers and K-Factors," *CDA Publication,* vol. 144, 2000.

[10] D. W. Hart, Introduction to Power Electronics, Upper Saddle River, NJ: Prentice-Hall, 1997.

[11] G. Todeschini, "Wind Energy Conversion Systems Based on DFIG Technology used as Active Filters: Steady-State and Transient Analysis," Worcester, MA: Worcester Polytechnic Institute, 2010.

[12] J. Fletcher and J. Yang, "Introduction to Doubly-Fed Induction Generator for Wind Power Applications," University of Strathclyde, Glasgow, 2010.

[13] Fuhrlander, *Specification - Generator FL 1570/ FL 1577,* 2004.

[14] "Automotive Grade, Fully Integrated, Hall Effect-Based Linear Current Sensor IC with 2.1 kVRMS Voltage Isolation and a Low-Resistance Current Conductor," Allegro, [Online]. Available: http://www.allegromicro.com/en/Products/Part_Numbers/0714/0714.pdf.

[15] "HCNR200 and HCNR201: High-Linearity Analog Optocouplers," Avago Technologies, [Online]. Available: http://www.avagotech.com/docs/AV02-0886EN.

[16] A. Merello, A. Rugginenti and M. Grasso, "Using monolithic high voltage gate drivers," [Online]. Available: http://www.irf.com/technical-info/designtp/dt04-4.pdf.

[17] "Designing RC Snubber Networks," CDE Cornell Dubilier, [Online]. Available: http://www.cde.com/catalogs/igbtAPPguide.pdf.

Appendix A – Pspice Simulation Code

DFIG_HARMONICS_XPU_STUDY.CIR

```
.SUBCKT DiodeWSnubber 1 2
D 1 2 DI
Cs 1 3 28n IC=0
Rs 3 2 30k
.MODEL DI D(RS=1m BV=5000 N=0.01)
.ENDS

.SUBCKT ThreePhaseRectifier 1 2 3 4 5
X1 1 4 DiodeWSnubber
X2 2 4 DiodeWSnubber
X3 3 4 DiodeWSnubber
X4 5 1 DiodeWSnubber
X5 5 2 DiodeWSnubber
X6 5 3 DiodeWSnubber
.ENDS

.PARAM phase_a = 9.525
*.STEP PARAM phase_a LIST -70 -40 -10 0 10 40 70
.PARAM phase_b = {phase_a - 120}
.PARAM phase_c = {phase_a + 120}
.PARAM Rgen_value = 3.667m
.PARAM Lgen_value = 130.8u
.PARAM Lxfmr_value = 42.10u
* Varying Xpu of transformer from 3% to 12%
*.STEP PARAM Lxfmr_value LIST 25.26u 33.68u 42.10u 50.52u 58.94u 67.35u 75.77u 84.19u 92.61u
101.03u


Vgen_a 1 50 SIN(0 580 60 0 0 {phase_a})
Vgen_b 2 50 SIN(0 580 60 0 0 {phase_b})
Vgen_c 3 50 SIN(0 580 60 0 0 {phase_c})
R_a 1 4 {Rgen_value}
R_b 2 5 {Rgen_value}
R_c 3 6 {Rgen_value}
L_a 4 7 {Lgen_value} IC=0
L_b 5 8 {Lgen_value} IC=0
L_c 6 9 {Lgen_value} IC=0

Vtm_a 10 0 SIN(0 563.38 60 0 0 0)
Vtm_b 11 0 SIN(0 563.38 60 0 0 -120)
Vtm_c 12 0 SIN(0 563.38 60 0 0 120)
Lxfmr_a 10 7 {Lxfmr_value} IC=0
Lxfmr_b 11 8 {Lxfmr_value} IC=0
```

```
Lxfmr_c 12 9 {Lxfmr_value} IC=0

Rm 7 71 0.01m

Xrect 71 8 9 20 21 ThreePhaseRectifier
Cfilter 20 21 33m IC=0
Rload 20 21 2.0
.FOUR 60 13 I(Lxfmr_a)
.PROBE
.TRAN 100m 1500m 1400m .1m UIC
.END
```

****

ACTIVE_FILTER_V1.CIR

```
.SUBCKT IdealTransistorSwitch 1 2 5 6
D1 1 3 DI
D2 2 1 DI
Rs 1 4 5
Cs 4 2 22n IC=0
S1 3 2 5 6 Switch
.MODEL Switch VSWITCH(RON=1m ROFF=10G VON=1 VOFF=0)
.MODEL DI D(RS=1m BV=5000 N=0.01)
.ENDS

.SUBCKT LogicTransistorSwitch 1 2 5 6
D1 1 3 DI
D2 2 1 DI
Rs 1 4 80k
Cs 4 2 30n IC=0
S1 3 2 5 6 Switch
.MODEL Switch VSWITCH(RON=1m ROFF=10G VON=1 VOFF=0)
.MODEL DI D(RS=1m BV=50000 N=0.01)
.ENDS

.PARAM inductance = .25m
*.STEP PARAM inductance LIST .5m 1m 5m 15m

.SUBCKT ActiveFilterV1 1 2 11 12 13 14 15
R1 1 3 0.1m
L1 3 4 {inductance} IC=0
X2 5 4 12 15 IdealTransistorSwitch
X1 5 2 11 15 IdealTransistorSwitch
X4 4 6 14 15 IdealTransistorSwitch
X3 2 6 13 15 IdealTransistorSwitch
```

```
Vbat 5 6 DC 50
*Cbat 5 6 1000u IC=50
.ENDS

* LOGIC GATES

* NOT GATE
.SUBCKT NotGate 1 2 3
Xnot 2 3 1 3 LogicTransistorSwitch
R 5 2 1k
V 5 3 DC 1
Cout 2 3 5n IC=0
.MODEL Lswitch VSWITCH(RON=1m ROFF=10G VON=1 VOFF=0)
.ENDS

* AND GATE
.SUBCKT AndGate 1 2 3 4
V 5 4 DC 1
Xand1 5 6 1 4 LogicTransistorSwitch
Xand2 6 3 2 4 LogicTransistorSwitch
R 3 4 1k
Cout 3 4 5n IC=0
*.MODEL Lswitch VSWITCH(RON=1m ROFF=10G VON=1 VOFF=0)
.ENDS

* AND GATE 3 INPUT
.SUBCKT AndGate3 1 2 3 4 5
V 6 5 DC 1
Xand1 6 7 1 5 LogicTransistorSwitch
Xand2 7 8 2 5 LogicTransistorSwitch
Xand3 8 4 3 5 LogicTransistorSwitch
R 4 5 1k
Cout 4 5 5n IC=0
*.MODEL Lswitch VSWITCH(RON=1m ROFF=10G VON=1 VOFF=0)
.ENDS

* OR GATE
.SUBCKT OrGate 1 2 3 4
V 5 4 DC 1
Xor1 5 3 1 4 LogicTransistorSwitch
Xor2 5 3 2 4 LogicTransistorSwitch
R 3 4 1k
Cout 3 4 5n IC=0
*.MODEL Lswitch VSWITCH(RON=1m ROFF=10G VON=1 VOFF=0)
.ENDS

* OR GATE 3 INPUT
.SUBCKT OrGate3 1 2 3 4 5
```

```
V 6 5 DC 1
Xor1 6 4 1 5 LogicTransistorSwitch
Xor2 6 4 2 5 LogicTransistorSwitch
Xor3 6 4 3 5 LogicTransistorSwitch
R 4 5 1k
Cout 4 5 5n IC=0
*.MODEL Lswitch VSWITCH(RON=1m ROFF=10G VON=1 VOFF=0)
.ENDS


* S1 LOGIC ( (B&~C) | (~A&~B&C) )  // new B~C
.SUBCKT S1Logic 1 2 3 4 5
Xnotgate1 1 6 5 NotGate
Xnotgate2 2 7 5 NotGate
Xnotgate3 3 8 5 NotGate
Xandgate1 6 7 3 9 5 AndGate3
Xandgate2 2 8 10 5 AndGate
Xorgate1 9 10 4 5 OrGate
*.SUBCKT S1Logic 1 2 3 4 5
*Xnotgate1 3 6 5 NotGate
*Xandgate1 2 6 4 5 AndGate

.ENDS

* S2 LOGIC ( (~B&C) | (A&B&~C) )          //~BC
.SUBCKT S2Logic 1 2 3 4 5
Xnotgate1 2 6 5 NotGate
Xnotgate2 3 7 5 NotGate
Xandgate1 1 2 7 8 5 AndGate3
Xandgate2 6 3 9 5 AndGate
Xorgate1 8 9 4 5 OrGate
.ENDS

* S3 LOGIC ( (A&~B&C) )                                  // ~BC
.SUBCKT S3Logic 1 2 3 4 5
Xnotgate1 2 6 5 NotGate
Xandgate1 1 6 3 4 5 AndGate3
.ENDS

* S4 LOGIC ( (~A&B&~C) )    // new B~C
.SUBCKT S4Logic 1 2 3 4 5
Xnotgate1 1 6 5 NotGate
Xnotgate2 3 7 5 NotGate
Xandgate1 6 2 7 4 5 AndGate3
.ENDS

* 24V
Vs 1 0 SIN(0 24 60)
```

```
Vd 1 2 DC 0
Xaf 2 0 21 22 23 24 0 ActiveFilterV1

*Control

* Actual Current Waveform
Ecia 51 0 VALUE={I(Vd)}

* Ideal Current Waveform
Ecid 52 0 VALUE={-6*SIN((300*PI*TIME)-(.3*PI))}
*Vtriangle 70 0 PULSE(-6 6 0 8.33m 8.33m 1n 16.67m)
*Vsin 71 0 SIN(0 2 120)
*Ecid 52 0 VALUE={V(70) + V(71)}
*Ecid 52 0 VALUE ={6*SIN(2*PI*((10000*TIME))*TIME)}


* Difference Actual-Ideal
Rcd 51 52 1

* V>0?  Output on node 53
Ecv 53 0 TABLE {V(1,0)}=(-1000000,0)(-0.1,0)(0.1,1)(1000000,1)
Rcv 53 0 1M

* Ii>0? Output on node 54
Eci 54 0 TABLE {V(52,0)}=(-1000000,0)(-0.1,0)(0.1,1)(1000000,1)
Rci 54 0 1M

* Ia-Ii>0? Output on node 55
Ecai 55 0 TABLE {V(51,52)}=(-1000000,0)(-0.1,0)(0.1,1)(1000000,1)
Rcai 55 0 1M

* Vcap > 150? Output on node 56
Ecap 56 0 TABLE {V(Xaf.5,Xaf.6)}=(-1000000,0)(139.9,0)(140.1,1)(1000000,1)
Rcap 56 0 1M

Xs1logic 53 54 55 21 0 S1Logic
Xs2Logic 53 54 55 22 0 S2Logic
Xs3Logic 53 54 55 23 0 S3Logic
Xs4Logic 53 54 55 24 0 S4Logic

.PROBE
.TRAN 100m 100m 0 .1m UIC
.END
```

Appendix B – MATLAB Code

```
function [y] = AlphaH(fh, alpha)
%   AlphaH: Produces the harmonics by means of the alpha value
h = fh./60;
y = 1./(h.^alpha);
end
```

Command Script

```
fh = [60 180 300 420 540 660 780];
b1 = AlphaH(fh,0.1);
b2 = AlphaH(fh,0.2);
b3 = AlphaH(fh,0.4);
b4 = AlphaH(fh,0.6);
b5 = AlphaH(fh,0.8);
b6 = AlphaH(fh,1);
h = [1 3 5 7 9 11 13];
bar(h,b1);
hold on
bar(h,b2);
hold on
bar(h,b3);
hold on
bar(h,b4);
hold on
bar(h,b5);
hold on
bar(h,b6);
```

```
function [y] = FactorK(x,q,e)
%   FactorK: Produces the transform derating factor
x1 = (x(1)* (sqrt(sum((x./x(1)).^2))));
N = length(x);
n=1:N;      All integer numbers from 1:N
x2 = sum((n.^q).*((x./x(1)).^2));
x3 = x2-1;
d = sqrt(1 + ((e/(1+e))*((x(1)/x1)^2)* x3));
y = 1/d;
end
```

```
function [y] = FactorKoddp(x,q,e)
%   FactorKoddp: Produces the derating factor percentage (for odd harmonics)
x1 = (x(1)* (sqrt(sum((x./x(1)).^2))));
N = length(x);
n=1:2:((2*N)-1);  Odd integer numbers from 1:N
x2 = sum((n.^q).*((x./x(1)).^2));
x3 = x2-1;
d = sqrt(1 + ((e/(1+e))*((x(1)/x1)^2)* x3));
y = 1/d;
end
```

```
function [y] = FKplot(fh,e)
%   FKplot: Produces derating k values versus alpha values with desired e
%   value
y1 = FactorKoddp(AlphaH(fh,0.1),1.7,e);
y2 = FactorKoddp(AlphaH(fh,0.2),1.7,e);
y3 = FactorKoddp(AlphaH(fh,0.3),1.7,e);
y4 = FactorKoddp(AlphaH(fh,0.4),1.7,e);
y5 = FactorKoddp(AlphaH(fh,0.5),1.7,e);
y6 = FactorKoddp(AlphaH(fh,0.6),1.7,e);
y7 = FactorKoddp(AlphaH(fh,0.7),1.7,e);
y8 = FactorKoddp(AlphaH(fh,0.8),1.7,e);
y9 = FactorKoddp(AlphaH(fh,0.9),1.7,e);
y10 = FactorKoddp(AlphaH(fh,1),1.7,e);
end
```

Command Script ----------

```
fh = [60 180 300 420 540 660 780];
e1 = .076/.924;                     7.6% eddy current loss
e2 = .146/.854;                     14.6% eddy current loss
e3 = .12/.88;                       12% eddy current loss
y1 = FKplot(fh,e1);
y2 = FKplot(fh,e2);
y3 = FKplot(fh,e3);
y4 = FKplot(fh,0.1);
alpha = [0.1:0.1:1];
plot(alpha, y1);
hold on
plot(alpha, y2, 'red');
hold on
plot(alpha, y3, 'black');
hold on
plot(alpha, y4, 'green');
```

Appendix C – C++ Code

```
/*
 ********************************************************************************************************************
 * File:       Main.c
 * Author:  Warranyu Walton
 * Version: 1.8
 * Date:     4/10/2012
 * Brief:     Main file for Active Filter System
 */

/****************************************** INCLUDES and DEFINES ************************************/

#include "stm32f10x.h"
#include "stdlib.h"
#include "math.h"

#define ADC1_DR_Address    ((uint32_t)0x4001244C)    // define ADC address
int __errno;

/************************************ INITIALIZE ST LIBRARY SRUCTURES ***********************************
            - Timer
            - DMA
            - ADC
            - NVIC
            - External Interrupt
            - GPIO
*********************************************************************************************************/

TIM_TimeBaseInitTypeDef  TIM_TimeBaseStructure;
TIM_OCInitTypeDef  TIM_OCInitStructure;
TIM_BDTRInitTypeDef TIM_BDTRInitStructure;
ADC_InitTypeDef ADC_InitStructure;
DMA_InitTypeDef DMA_InitStructure;
EXTI_InitTypeDef   EXTI_InitStructure;
NVIC_InitTypeDef   NVIC_InitStructure;
GPIO_InitTypeDef GPIO_InitStructure;

/************************************** IMPORT EXTERNAL VARIABLES ***********************************/

extern volatile uint8_t EnableCount;
extern volatile uint8_t TimeToUpdate;
extern volatile uint32_t TickCount;
extern volatile uint8_t TIM2_CountNotComplete;
extern volatile uint8_t BeginningOfCycleFLG;

/************************************** INITIALIZE GLOBAL VARIABLES *********************************/

uint16_t TimerPeriod = 0;
uint16_t Channel1Pulse = 0;
__IO uint16_t ADCConvertedValuesTab[4];

/****************************************** INITIALIZE CONSTANTS ************************************/

// Sine wave table
static const int SINTABLE[400] =
{
            0,          16,         31,         47,         63,         78,         94,         110,        125,        141,        156,        172,
            187,        203,        218,        233,        249,        264,        279,        294,
            309,        324,        339,        353,        368,        383,        397,        412,        426,        440,        454,        468,
            482,        495,        509,        522,        536,        549,        562,        575,
            588,        600,        613,        625,        637,        649,        661,        673,        685,        696,        707,        718,
            729,        740,        750,        760,        771,        780,        790,        800,
```

```
	809,	818,	827,	836,	844,	853,	861,	869,	876,	884,	891,	898,
	905,	911,	918,	924,	930,	935,	941,	946,
	951,	956,	960,	965,	969,	972,	976,	979,	982,	985,	988,	990,
	992,	994,	996,	997,	998,	999,	1000,	1000,
	1000,	1000,	1000,	999,	998,	997,	996,	994,	992,	990,	988,	985,
	982,	979,	976,	972,	969,	965,	960,	956,
	951,	946,	941,	935,	930,	924,	918,	911,	905,	898,	891,	884,
	876,	869,	861,	853,	844,	836,	827,	818,
	809,	800,	790,	780,	771,	760,	750,	740,	729,	718,	707,	696,
	685,	673,	661,	649,	637,	625,	613,	600,
	588,	575,	562,	549,	536,	523,	509,	495,	482,	468,	454,	440,
	426,	412,	397,	383,	368,	353,	339,	324,
	309,	294,	279,	264,	249,	233,	218,	203,	187,	172,	156,	141,
	125,	110,	94,	78,	63,	47,	31,	16,
	0,	-16,	-31,	-47,	-63,	-78,	-94,	-110,	-125,	-141,	-156,	-172,
	-187,	-203,	-218,	-233,	-249,	-264,	-279,	-294,
	-309,	-324,	-339,	-353,	-368,	-383,	-397,	-412,	-426,	-440,	-454,	-468,
	-482,	-495,	-509,	-522,	-536,	-549,	-562,	-575,
	-588,	-600,	-613,	-625,	-637,	-649,	-661,	-673,	-685,	-696,	-707,	-718,
	-729,	-740,	-750,	-760,	-771,	-780,	-790,	-800,
	-809,	-818,	-827,	-836,	-844,	-853,	-861,	-869,	-876,	-884,	-891,	-898,
	-905,	-911,	-918,	-924,	-930,	-935,	-941,	-946,
	-951,	-956,	-960,	-965,	-969,	-972,	-976,	-979,	-982,	-985,	-988,	-990,
	-992,	-994,	-996,	-997,	-998,	-999,	-1000,	-1000,
	-1000,	-1000,	-1000,	-999,	-998,	-997,	-996,	-994,	-992,	-990,	-988,	-985,
	-982,	-979,	-976,	-972,	-969,	-965,	-960,	-956,
	-951,	-946,	-941,	-935,	-930,	-924,	-918,	-911,	-905,	-898,	-891,	-884,
	-876,	-869,	-861,	-853,	-844,	-836,	-827,	-818,
	-809,	-800,	-790,	-780,	-771,	-760,	-750,	-740,	-729,	-718,	-707,	-696,
	-685,	-673,	-661,	-649,	-637,	-625,	-613,	-600,
	-588,	-575,	-562,	-549,	-536,	-523,	-509,	-495,	-482,	-468,	-454,	-440,
	-426,	-412,	-397,	-383,	-368,	-353,	-339,	-324,
	-309,	-294,	-279,	-264,	-249,	-233,	-218,	-203,	-187,	-172,	-156,	-141,
	-125,	-110,	-94,	-78,	-63,	-47,	-31,	-16
};

/****************************************** FUNCTION DECLARATIONS ******************************************/

void RCC_Config(void);                          // Configure hardware RCC
void GPIO_Config(void);                         // Configure GPIO
void EXTI9_Config(void);                // Configure External Interrupt
void TIM1_Config(void);                         // Configure Timer 1 for IGBT Test Signals with hardware dead-time generation
void TIM2_Config(void);                         // Configure Timer 2 for actual in program dead-time generation
void TIM1_Run(void);                    // Run Timer 1
void TIM2_Run(void);                    // Run Timer 2
void DMA_Config(void);                          // Configure DMA for circular transfer of data from ADC
void ADC_Config(void);                          // Configure ADC for two channel continuous scan
void SetSysClockTo24(void);             // Setup the system clock using external 8MHz crystal with PLL up to 24MHz
void NVIC_Configuration(void);  // Configure Nested Vectored Interrupt Controller


/****************************************** MAIN PROGRAM ******************************************/

int main(void)
{
	/*
		Initialize System
	*/
	SystemInit();
	SetSysClockTo24();
	RCC_Config();
	GPIO_Config();
	TIM1_Config();
	TIM2_Config();
	NVIC_Configuration();
	TIM1_Run();
```

```
DMA_Config();
ADC_Config();
EXTI9_Config();
SysTick_Config(SystemCoreClock / 24000);                                    // tick every 41.67us

/*
        Set up GPIOC 8 and 9 as diagnostics LEDs
*/
GPIOC->BRR = GPIO_Pin_8;
GPIOC->BRR = GPIO_Pin_9;

/*
        Initialize Variables
*/
int LineVoltage = 0;                    // Keeps the value of the Line Voltage measured from ADC channel 13 (PC.03)
int I_NonLinearLoad = 0;                // Keeps the value nonlinear current measured from ADC channel 11 (PC.01)
int I_CorrectionActual = 0;             // Keeps the value of the correction current measured from ADC channel 12 (PC.02)
int I_NonLinearLoadSquared = 0;         // Non-linear load squared values for RMS calculation
int I_CorrectionTarget = 0;             // Stores target value for the correction current
int I_rmsVal = 0;                       // RMS value calculated after one 60Hz cycle
int count = 0;                          // Step count (there are 400 steps in a 60Hz cycle)
uint8_t S1_prev = 0;                    // Switch 1 previous state
uint8_t S2_prev = 0;                    // Switch 2 previous state
uint8_t S3_prev = 0;                    // Switch 3 previous state
uint8_t S4_prev = 0;                    // Switch 4 previous state
TimeToUpdate = 0;                       // Flag to signify that it is time to calculate and output new switch positions
TickCount = 0;                          // System Tick Count, used to measure time and set the TimeToUpdate Flag
EnableCount = 0;                        // Enable Count Flag
BeginningOfCycleFLG = 0;                // Beginning of Cycle Flag

while(1)
{

        /*
                This section keeps looping and updating voltage and current values. Every 400th of a cycle (24KHz),
                the TimeToUpdate Flag is set based on the SysTick count. When the Flag is set, the program breaks
                out of the loop to perform necessary calculations and output appropriate switch signals to the
                IGBT gate drive IC via optocouplers.
        */
        do
        {
                LineVoltage = (ADCConvertedValuesTab[2] - 2048);            // Updates Line Voltage Value
                I_NonLinearLoad =(ADCConvertedValuesTab[0] - 2083);         // Updates Non-Linear Current Value
                I_CorrectionActual = -1*(ADCConvertedValuesTab[1] - 2041);  // Updates Correction Current Value

                if(BeginningOfCycleFLG)                                     // If it is the beginning of a 60Hz Cycle
                {
                        __disable_irq();                                    // Disable interrupts
                        EnableCount = 1;                                    // Enable Count (only applies to the first
                                                                            // time the beginning of the cycle is
                                                                            // detected after power up)
                        SysTick->VAL = 0;                                   // Reset the SysTick value to 0
                        I_NonLinearLoadSquared = 0;                         // Reset I_NonLinearLoadSquared to 0 so it
                                                                            // can accumulate new values for this cycle
                        TimeToUpdate = 1;                                   // Set TimeToUpdate Flag to 1. This will get
                                                                            // the program out of the update values
                                                                            // loop to the actual switching
                        count = 0;                                          // Reset the program count to 0
                        GPIOC->BSRR = GPIO_Pin_8;                           // Light up onboard LED to mark beginning
                                                                            // of cycle
                        BeginningOfCycleFLG = 0;                            // Reset the Beginnning of Cycle Flag to 0
                        __enable_irq();                                     // Enable Interrupts again
                }
        }while (TimeToUpdate != 1);

        /*
```

```
		When it is time to do the switching, the program checks if it is still less than the 400th count,
		which marks the end of the cycle. If the step count is less than 400, then it will:
			- 	Reset the TimeToUpdate Flag
			- 	Turn off the onboard LED when half a cycle has passed
			- 	Square the new value of the non-linear current and add it to the
				I_NonLinearLoadSquared value
			- 	Calculate the appropriate target current value based on the previously calculated RMS
				current
				from the last cycle and the current non-linear current value.
			- 	Assign Boolean values to denote certain conditions (I_CorrectionTarget > 0, and
				(I_CorrectionActual - I_CorrectionTarget) > 0)
			- 	Perform logic to assign switch states for each of the four switches
			- 	If there are any changes in the switch state, assign a dead-time of 5us where all switches
				are off before making changes
			- 	Output the new switch states
		Otherwise, the end of the cycle has been reached. At this point the RMS value of the nonlinear current
		is calculated by dividing the I_NonLinearLoadSquared by 400 and then taking the square root. Both
		I_NonLinearLoadSquared and Count is then reset to 0.
*/

TimeToUpdate = 0;                                   // Reset the TimeToUpdate Flag

if(count > 200)
{
	GPIOC->BRR = GPIO_Pin_8;                    // Turn off LED to mark half cycle
}

if(count < 400)
{
	// Update I_NonLinearLoadSquared value
	I_NonLinearLoadSquared += (I_NonLinearLoad * I_NonLinearLoad);
	/*
		The I_CorrectionTarget calculation is based off of the difference between the current non-linear
		current value and a sinusoidal value with a peak equal to sqrt(2) times the rms value of the current
		measured last cycle
	*/
	I_CorrectionTarget = -(I_NonLinearLoad-(((I_rmsVal*141)*(SINTABLE[count]))/100000));

	int B = (I_CorrectionTarget > 0) ? 1: 0;                                    // Assign logic condition
	int C = ((I_CorrectionActual - I_CorrectionTarget) > 0)? 1: 0;              // Assign logic condition

	int S1 = !(B && !C);                                                        // Assign Switch 1 state
	int S2 = !(!B && C);                                                        // Assign Switch 2 state
	int S3 = !(!B && C);                                                        // Assign Switch 3 state
	int S4 = !(B && !C);                                                        // Assign Switch 4 state

	if((S1 != S1_prev) || (S2 != S2_prev) || (S3 != S3_prev) || (S4 != S4_prev))
	{
		// switch changing states.. generate dead time of 5us
		GPIOA->BSRR = GPIO_Pin_11 | GPIO_Pin_12 | GPIO_Pin_3 | GPIO_Pin_4;
		TIM2_CountNotComplete = 1;
		TIM2_Run();
		while(TIM2_CountNotComplete);
		TIM2_CountNotComplete = 1;
	}

	GPIO_WriteBit(GPIOA, GPIO_Pin_4, S1);                                       // PA.04 drives S1 (LH)
	GPIO_WriteBit(GPIOA, GPIO_Pin_12, S2);                                      // PA.12 drives S2(RH)
	GPIO_WriteBit(GPIOA, GPIO_Pin_3, S3);                                       // PA.03 drives S3 (LL)
	GPIO_WriteBit(GPIOA, GPIO_Pin_11, S4);                                      // PA.11 drives S4 (RL)

	S1_prev = S1;                                                               // Set S1_prev to current value
	S2_prev = S2;                                                               // Set S2_prev to current value
	S3_prev = S3;                                                               // Set S3_prev to current value
	S4_prev = S4;                                                               // Set S4_prev to current value
```

```
                }
                else
                {
                        count = 0;
                        I_rmsVal = sqrt((I_NonLinearLoadSquared/400));
                        I_NonLinearLoadSquared = 0;
                }
                count++;                                                                        // Increment the step counter
        }
}


/****************************************** FUNCTION DEFINITIONS ******************************************/

void RCC_Config()
{
        // TIM1, GPIOA, GPIOB, GPIOC, GPIOE, ADC, and AFIO clocks enable
        RCC_APB2PeriphClockCmd(RCC_APB2Periph_TIM1 | RCC_APB2Periph_GPIOA |
                                                        RCC_APB2Periph_GPIOE | RCC_APB2Periph_GPIOB |
                                                        RCC_APB2Periph_GPIOC | RCC_APB2Periph_ADC1 |
                                                        RCC_APB2Periph_AFIO, ENABLE);

        RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM2, ENABLE);

        // ADCCLK = PCLK/2
        RCC_ADCCLKConfig(RCC_PCLK2_Div2);
        // Enable DMA1 Clock
        RCC_AHBPeriphClockCmd(RCC_AHBPeriph_DMA1, ENABLE);
}

void GPIO_Config()
{
        GPIO_PinRemapConfig(GPIO_PartialRemap_TIM1, ENABLE);

        //GPIOA Configuration: Channel 1 as alternate function push-pull
        GPIO_InitStructure.GPIO_Pin = GPIO_Pin_7 | GPIO_Pin_8 | GPIO_Pin_9 | GPIO_Pin_10;
        GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
        GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
        GPIO_Init(GPIOA, &GPIO_InitStructure);

        //GPIOB Configuration: Channel 1N as alternate function push-pull
        GPIO_InitStructure.GPIO_Pin = GPIO_Pin_13 | GPIO_Pin_14 | GPIO_Pin_15;
        GPIO_Init(GPIOB, &GPIO_InitStructure);

        //GPIOB Configuration: BKIN pin (connect to ground -- will set break to timer operation if HIGH)
        GPIO_InitStructure.GPIO_Pin = GPIO_Pin_12;
        GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
        GPIO_Init(GPIOB, &GPIO_InitStructure);

        // Configure PC.01, PC.02, PC.04 (ADC channel 11, 12, 13 and 14) as analog input
        GPIO_InitStructure.GPIO_Pin = GPIO_Pin_1 | GPIO_Pin_2 | GPIO_Pin_3 | GPIO_Pin_4;
        GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AIN;
        GPIO_Init(GPIOC, &GPIO_InitStructure);

        // Configure P.C08 and P.C09 as outputs (LEDs)
        GPIO_InitStructure.GPIO_Pin = GPIO_Pin_8 | GPIO_Pin_9 | GPIO_Pin_10;
        GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
        GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
        GPIO_Init(GPIOC, &GPIO_InitStructure);

        // Configure P.A11, P.A12, P.A13, and P.A14 as outputs
        GPIO_InitStructure.GPIO_Pin = GPIO_Pin_11 | GPIO_Pin_12 | GPIO_Pin_3 | GPIO_Pin_4;
        GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
        GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
        GPIO_Init(GPIOA, &GPIO_InitStructure);
}
```

```
void TIM1_Config()
{
	TimerPeriod = (SystemCoreClock / 20000) - 1;
	// The Timer pulse is calculated as follows:
	// - ChannelxPulse = DutyCycle * (TIM1_Period - 1) / 100
	Channel1Pulse = (uint16_t) (((uint32_t) 50 * (TimerPeriod - 1)) / 100);

	TIM_TimeBaseStructure.TIM_Prescaler = 0;
	TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;
	TIM_TimeBaseStructure.TIM_Period = TimerPeriod;
	TIM_TimeBaseStructure.TIM_ClockDivision = 0;
	TIM_TimeBaseStructure.TIM_RepetitionCounter = 0;

	TIM_TimeBaseInit(TIM1, &TIM_TimeBaseStructure);

	// Channel 1 Configuration for PWM mode
	TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_PWM2;
	TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;
	TIM_OCInitStructure.TIM_OutputNState = TIM_OutputNState_Enable;
	TIM_OCInitStructure.TIM_Pulse = Channel1Pulse;
	TIM_OCInitStructure.TIM_OCPolarity = TIM_OCPolarity_Low;
	TIM_OCInitStructure.TIM_OCNPolarity = TIM_OCNPolarity_Low;

	TIM_OCInitStructure.TIM_OCIdleState = TIM_OCIdleState_Set;
	TIM_OCInitStructure.TIM_OCNIdleState = TIM_OCIdleState_Reset;

	TIM_OC1Init(TIM1, &TIM_OCInitStructure);

	// Automatic Output enable, Break, dead time and lock configuration
	// BDTR -- Break and Dead Time Register, get it?

	// Actual Dead time = TIM_DeadTime/SystemCoreClock (24MHz)
	// For a dead time of 5us, TIM_DeadTime = 120

	TIM_BDTRInitStructure.TIM_OSSRState = TIM_OSSRState_Enable;
	TIM_BDTRInitStructure.TIM_OSSIState = TIM_OSSIState_Enable;
	TIM_BDTRInitStructure.TIM_LOCKLevel = TIM_LOCKLevel_1;
	TIM_BDTRInitStructure.TIM_DeadTime = 120;
	TIM_BDTRInitStructure.TIM_Break = TIM_Break_Disable;
	TIM_BDTRInitStructure.TIM_BreakPolarity = TIM_BreakPolarity_High;
	TIM_BDTRInitStructure.TIM_AutomaticOutput = TIM_AutomaticOutput_Enable;

	TIM_BDTRConfig(TIM1, &TIM_BDTRInitStructure);
}

void TIM1_Run()
{
	/* TIM1 counter enable */
	TIM_Cmd(TIM1, ENABLE);

	/* Main Output Enable */
	TIM_CtrlPWMOutputs(TIM1, ENABLE);
}

void TIM2_Config()
{
  /* Time base configuration */
  TIM_TimeBaseStructure.TIM_Period = 250;
  TIM_TimeBaseStructure.TIM_Prescaler = 0;
  TIM_TimeBaseStructure.TIM_ClockDivision = 0;
  TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;
  TIM_TimeBaseInit(TIM2, &TIM_TimeBaseStructure);

  TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_Timing;
  TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;
```

```
  TIM_OCInitStructure.TIM_Pulse = 65535;
  TIM_OCInitStructure.TIM_OCPolarity = TIM_OCPolarity_High;

  TIM_OC1Init(TIM2, &TIM_OCInitStructure);

  TIM_ITConfig(TIM2, TIM_IT_CC1, ENABLE);

}

void TIM2_Run()
{
		/* TIM2 counter enable */
		TIM_Cmd(TIM2, ENABLE);
}


void DMA_Config()
{
		DMA_DeInit(DMA1_Channel1);
		DMA_InitStructure.DMA_PeripheralBaseAddr = ADC1_DR_Address;
		DMA_InitStructure.DMA_MemoryBaseAddr = (uint32_t)&ADCConvertedValuesTab;
		DMA_InitStructure.DMA_DIR = DMA_DIR_PeripheralSRC;
		DMA_InitStructure.DMA_BufferSize = 4;
		DMA_InitStructure.DMA_PeripheralInc = DMA_PeripheralInc_Disable;
		DMA_InitStructure.DMA_MemoryInc = DMA_MemoryInc_Enable;
		DMA_InitStructure.DMA_PeripheralDataSize = DMA_PeripheralDataSize_HalfWord;
		DMA_InitStructure.DMA_MemoryDataSize = DMA_MemoryDataSize_HalfWord;
		DMA_InitStructure.DMA_Mode = DMA_Mode_Circular;
		DMA_InitStructure.DMA_Priority = DMA_Priority_High;
		DMA_InitStructure.DMA_M2M = DMA_M2M_Disable;
		DMA_Init(DMA1_Channel1, &DMA_InitStructure);

		// Enable DMA1 Channel1
		DMA_Cmd(DMA1_Channel1, ENABLE);
}

void NVIC_Configuration(void)
{
		NVIC_InitTypeDef NVIC_InitStructure;

  /* Enable the TIM2 gloabal Interrupt */
  NVIC_InitStructure.NVIC_IRQChannel = TIM2_IRQn;
  NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 1;
  NVIC_InitStructure.NVIC_IRQChannelSubPriority = 1;
  NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;

  NVIC_Init(&NVIC_InitStructure);
}


void ADC_Config()
{
		ADC_InitStructure.ADC_Mode = ADC_Mode_RegSimult;
		ADC_InitStructure.ADC_ScanConvMode = ENABLE;
		ADC_InitStructure.ADC_ContinuousConvMode = ENABLE;
		ADC_InitStructure.ADC_ExternalTrigConv = ADC_ExternalTrigConv_None;
		ADC_InitStructure.ADC_DataAlign = ADC_DataAlign_Right;
		ADC_InitStructure.ADC_NbrOfChannel = 4;
		ADC_Init(ADC1, &ADC_InitStructure);

		// ADC1 regular channel14 configuration
		ADC_RegularChannelConfig(ADC1, ADC_Channel_11, 1, ADC_SampleTime_1Cycles5);
		ADC_RegularChannelConfig(ADC1, ADC_Channel_12, 2, ADC_SampleTime_1Cycles5);
		ADC_RegularChannelConfig(ADC1, ADC_Channel_13, 3, ADC_SampleTime_1Cycles5);
		ADC_RegularChannelConfig(ADC1, ADC_Channel_14, 4, ADC_SampleTime_1Cycles5);
```

```
	// Enable ADC1 DMA
	ADC_DMACmd(ADC1, ENABLE);

	// Enable ADC1
	ADC_Cmd(ADC1, ENABLE);

	// Enable ADC1 reset calibaration register
	ADC_ResetCalibration(ADC1);

	// Check the end of ADC1 reset calibration register
	while(ADC_GetResetCalibrationStatus(ADC1));

	// Start ADC1 calibaration
	ADC_StartCalibration(ADC1);
	// Check the end of ADC1 calibration
	while(ADC_GetCalibrationStatus(ADC1));

	// Start ADC1 Software Conversion
	ADC_SoftwareStartConvCmd(ADC1, ENABLE);
}

void EXTI9_Config()
{
	/* Enable GPIOB clock */
	RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB, ENABLE);

	// Configure PB09 as input floating
	GPIO_InitStructure.GPIO_Pin = GPIO_Pin_9;
	GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
	GPIO_Init(GPIOB, &GPIO_InitStructure);

	// Enable AFIO clock
	RCC_APB2PeriphClockCmd(RCC_APB2Periph_AFIO, ENABLE);

	// Connect EXTI9 to PB.09
	GPIO_EXTILineConfig(GPIO_PortSourceGPIOB, GPIO_PinSource9);

	// Configure EXTI9 Line
	EXTI_InitStructure.EXTI_Line = EXTI_Line9;
	EXTI_InitStructure.EXTI_Mode = EXTI_Mode_Interrupt;
	EXTI_InitStructure.EXTI_Trigger = EXTI_Trigger_Rising;
	EXTI_InitStructure.EXTI_LineCmd = ENABLE;
	EXTI_Init(&EXTI_InitStructure);

	NVIC_InitStructure.NVIC_IRQChannel = EXTI9_5_IRQn;
	NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
	NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;
	NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;

	NVIC_Init(&NVIC_InitStructure);

}

void SetSysClockTo24(void)
{
 /* SYSCLK, HCLK, PCLK2 and PCLK1 configuration ----------------------*/
 /* RCC system reset(for debug purpose) */
 RCC_DeInit();

 /* Enable HSE */
 RCC_HSEConfig(RCC_HSE_ON);

 /* Wait till HSE is ready */
 int HSEStartUpStatus;
 HSEStartUpStatus = RCC_WaitForHSEStartUp();
```

```
 if (HSEStartUpStatus == SUCCESS)
 {
  /* Flash 0 wait state */
  FLASH_SetLatency(FLASH_Latency_0);
  /* HCLK = SYSCLK */
  RCC_HCLKConfig(RCC_SYSCLK_Div1);

  /* PCLK2 = HCLK */
  RCC_PCLK2Config(RCC_HCLK_Div1);

  /* PCLK1 = HCLK */
  RCC_PCLK1Config(RCC_HCLK_Div1);

  /* PLLCLK = (8MHz/2) * 6 = 24 MHz */
  RCC_PREDIV1Config(RCC_PREDIV1_Source_HSE, RCC_PREDIV1_Div2);
  RCC_PLLConfig(RCC_PLLSource_PREDIV1, RCC_PLLMul_6);

  /* Enable PLL */
  RCC_PLLCmd(ENABLE);

  /* Wait till PLL is ready */
  while (RCC_GetFlagStatus(RCC_FLAG_PLLRDY) == RESET)
  {
  }

  /* Select PLL as system clock source */
  RCC_SYSCLKConfig(RCC_SYSCLKSource_PLLCLK);

  /* Wait till PLL is used as system clock source */
  while (RCC_GetSYSCLKSource() != 0x08)
  {
  }
 }
 else
 {
  /* If HSE fails to start-up, the application will have wrong clock configuration.
     User can add here some code to deal with this error */

  /* Go to infinite loop */
  while (1)
  {
  }
 }
}
```