

HIGH-FREQUENCY GAN-BASED INVERTER CONTROL DESIGN USING PSIM

A Thesis

Presented in Partial Fulfillment of the Requirements for the

Degree of Master of Science

with a

Major in Electrical Engineering

in the

College of Graduate Studies

University of Idaho

by

Ryan P. Ready

Major Professor: Herbert L. Hess, Ph.D.

Committee Members: Yacine Chakhchoukh, Ph.D.; Brian K. Johnson, Ph.D.

Department Administrator: Joseph D. Law, Ph.D.

August 2020

AUTHORIZATION TO SUBMIT THESIS

This thesis of Ryan P. Ready, submitted for the degree of Master of Science with a Major in Electrical Engineering and titled “High-Frequency GaN-Based Inverter Control Design Using PSIM,” has been reviewed in final form. Permission, as indicated by the signatures and dates below, is now granted to submit final copies to the College of Graduate Studies for approval.

Major Professor:

Herbert L. Hess, Ph.D.

Date

Committee Members:

Yacine Chakhchoukh, Ph.D.

Date

Brian K. Johnson, Ph.D.

Date

Department Administrator:

Joseph D. Law, Ph.D.

Date

ABSTRACT

Gallium nitride (GaN) transistors are becoming more common in power electronics. This thesis describes how they can be used to improve solar generators by improving the inverter that they contain. The benefits of using GaN in an inverter include higher efficiency, higher switching speeds, and the ability to reduce the size of some inverter components. The full-bridge inverter topology was chosen for this inverter because of its favorable high-frequency switching characteristics. Using a full-bridge GaN-based inverter evaluation board from Transphorm, Inc., a PI control scheme was successfully designed for this inverter using the simulation software PSIM[®]. This control scheme performed satisfactorily when tested in the inverter showing that PSIM[®] can be a useful design tool for high-frequency GaN-based inverters.

ACKNOWLEDGEMENTS

I would first like to thank Dr. Herb Hess for giving me the opportunity to work on this project while pursuing an advanced degree. I have enjoyed working with and learning from him.

Secondly, I would also like to thank Dr. Brian Johnson for many illuminating power classes as well as the other ECE faculty members whom I had as teachers.

Thirdly, I wish to thank Jason Canaday for his numerous and invaluable contributions to this project. His hard work, focus, and determination were key to this project's successes.

Finally, I'd like to thank James Brainard of Inergy Solar for his practical advice and encouragement throughout the project.

DEDICATION

This thesis is dedicated to my wonderful wife Kathleen without whose support its completion, while theoretically possible, would have been much less probable.

TABLE OF CONTENTS

AUTHORIZATION TO SUBMIT THESIS	ii
ABSTRACT	iii
ACKNOWLEDGEMENTS	iv
DEDICATION	v
TABLE OF CONTENTS	vi
LIST OF FIGURES	viii
LIST OF ACRONYMS	x
CHAPTER 1: INTRODUCTION	1
INVERTERS	1
SOLAR GENERATORS	1
PROBLEM STATEMENT	4
THESIS OBJECTIVES	4
CHAPTER 2: BACKGROUND	6
GAN SEMICONDUCTORS	6
GAN ADVANTAGES	6
GAN DISADVANTAGES	6
INDUSTRY ADOPTION OF GAN	7
LITERATURE REVIEW	7
CHAPTER 3: THE FULL-BRIDGE CONVERTER	8
TOPOLOGY CONSIDERATIONS	8
FULL-BRIDGE TOPOLOGY	8
SWITCHING THE FULL-BRIDGE	11
PULSE WIDTH MODULATION	11
BI-POLAR SWITCHING	13
UNI-POLAR SWITCHING	13

SWITCHING HARMONICS	14
CHAPTER 4: HARDWARE	16
THE TRANSFORM TDPV1000E0C1 SINGLE-PHASE INVERTER	16
THE TI C2000 [™] SERIES MICROCONTROLLER	18
GAN FETs	19
GATE DRIVERS	20
FERRITE BEADS	20
CHAPTER 5: MODELING THE INVERTER	22
PSIM [®]	22
SMARTCTRL [©]	22
SIMCODER [™]	22
MODELLING WITH PSIM [®]	23
CONTROL DESIGN PROCESS	23
INNER CURRENT FEEDBACK LOOP	26
OUTER VOLTAGE FEEDBACK LOOP	33
CHAPTER 6: RESULTS	36
SIMULATION RESULTS	36
HARDWARE RESULTS	38
CHAPTER 7: CONCLUSION AND FUTURE WORK	44
CONCLUSION	44
FUTURE WORK	44
REFERENCES	46
APPENDIX A: INVERTER CONTROL CODE	48
APPENDIX B: HARDWARE SCHEMATICS	55

LIST OF FIGURES

1.1	The Inergy Flex 1500 Solar Generator [1]	2
3.1	The full bridge switch arrangement	8
3.2	With SW1 and SW2 on, the circuit sees $+V_s$	9
3.3	With SW3 and SW4 on, the circuit sees $-V_s$	10
3.4	With both SW1 and SW3 on, the circuit sees $0V$	10
3.5	Comparison of a DC reference voltage V_{ref} (blue) and a 1 kHz sawtooth carrier waveform V_{carr} (red)	11
3.6	Comparison of a 1 kHz triangular carrier waveform (red) to a 60 Hz modulating waveform (blue)	12
3.7	Switching waveform resulting from bi-polar switching at 1 kHz	13
3.8	Switching waveform resulting from uni-polar switching of a full-bridge	14
3.9	An Fast Fourier Transform (FFT) of the bipolar switching waveform shown in Figure 3.7	15
3.10	An FFT of the uni-polar switching scheme shown in Figure 3.8	15
4.1	The Transphorm TDPV1000E0C1 evaluation board	16
4.2	The Transphorm TDPV1000E0C1 simplified circuit	17
4.3	Typical efficiency with 350 Vdc input and 240 Vac output [16]	18
4.4	The TI experimenter's board with the TMS320F28069 card mounted	19
5.1	Initial circuit diagram	24
5.2	Load comprised of two series-connected resistors	25
5.3	Circuit with AC sweep for current sensor frequency response	26
5.4	AC sweep parameters	27
5.5	Results of the AC sweep showing amplitude and phase as a function of frequency	28

5.6	Feedback diagram with selections	29
5.7	Solution map of cross frequency vs. phase margin	30
5.8	PI tuning interface	31
5.9	Parameters for the digital PI block	32
5.10	Inverter circuit with closed loop current feedback	32
5.11	Output voltage and current with closed loop current feedback	33
5.12	Circuit set up to sweep the frequency response of the load voltage	34
5.13	Bode plot of the output	34
5.14	The completed circuit for generating the code for closed loop control	35
6.1	The simulated output voltage and current with combined feedback loops closed	37
6.2	The simulated output voltage and current with the switched load behavior . .	38
6.3	The hardware testing setup	39
6.4	The output voltage waveform (yellow) and output current waveform (green) with the completed feedback loops and a 9.3 Ω resistive load	40
6.5	The output voltage waveform (yellow) and output current waveform (green) with an 11 Ω resistive and inductive load	41
6.6	The output voltage waveform (yellow) and output current waveform (green) with a 6 Ω resistive and inductive load	42
6.7	The output voltage waveform (yellow) and output current waveform (green) with a 4 Ω resistive and inductive load	43

LIST OF ACRONYMS

AC	alternating current
ADC	analog to digital converter
DC	direct current
EMI	electromagnetic interference
eV	electron-volts
FFT	Fast Fourier Transform
GaN	gallium nitride
GaN FET	gallium nitride field-effect transistor
GPIO	general purpose input and output
IDE	integrated development environment
IGEM	Idaho Global Entrepreneurial Mission
INL	Idaho National Laboratory
LBC	Little Box Challenge
PI	proportional-integral
PWM	pulse width modulation
RAM	random access memory
Si	silicon
SiC	silicon carbide

SMD	surface-mount devices
SPWM	sinusoidal pulse width modulation
TI	Texas Instruments
WBG	wide bandgap
ZOH	zero order hold

CHAPTER 1: INTRODUCTION

1.1 INVERTERS

The inverter is an electrical circuit that converts direct current (DC) electrical energy to alternating current (AC) electrical energy. It derives its name from the ability to “invert” the polarity of its output. Direct current is the natural output from solar panels, batteries, and some DC generators. Since most household appliances require AC, inverters are often used to power equipment where there is access to DC power but not AC, e.g. vehicle camping or, since energy can be stored in batteries, as a back-up for when the power grid fails. There are many different types and of inverters. These range in size from large pad-mounted inverters connected to the grid to smaller devices that plug into the cigarette lighter in a car (nowadays referred to as a 12 Volt accessory port; in-car cigarette lighters being no longer in vogue).

1.1.1 SOLAR GENERATORS

Inverters are used in solar generators. A solar generator, like its gasoline counterpart, is a portable source of AC electricity. As a gasoline generator is limited by the fuel in its gas tank, so also the solar generator is limited by the energy capacity of its battery. What makes it “solar” is its capability to be recharged via solar panels to provide additional energy. The solar generator is limited in its portability by its size and weight. As its power capability or energy capacity rises, so also does its size and weight.



Figure 1.1: The Inergy Flex 1500 Solar Generator [1]

To improve their solar generator's utility and appeal, companies like Inergy Solar of Pocatello, ID are working to reduce the size and weight of their solar generators while simultaneously increasing their efficiency and power. Figure 1.1 above shows their current model of solar generator. With this goal, Inergy Solar partnered with the University of Idaho and received an Idaho Global Entrepreneurial Mission (IGEM) grant to develop a more powerful inverter based on recent improvements in gallium nitride (GaN) semiconductors. A GaN-based inverter has the potential to improve all of these aspects.

The scope of work for this project detailed the desired final specifications for this inverter [2]. They are as follows:

1. Inverter.

- Capable of 6 kW continuous output with an input of 48 Volts DC.
- Split phase output of 220 V and 110 V .

- Capable of 40, 60, and 400 Hz operation.
- 12 kW surge, overload, short circuit, and over temperature protection.
- Harmonic suppression via hardware and software.
- Individually addressable and switchable AC outputs.
- Temperature, voltage, current, and load sensors with accessible connections.
- An active, closed loop liquid cooling system.

2. Master Control Module

- Raspberry Pi with nanogrid control software to be jointly developed with Idaho National Laboratory (INL).
- Data logging software utilizing standard communication protocols to provide a data visualization screen for user interfacing.
- Embedded control system accessible by wireless or user interface display.
- WiFi module.
- Bluetooth module.
- PLC module.
- Security system.
- Inverter control software.

3. Motherboard

- Primary DC power bus.
- Battery bank input connections.
- AC input/output connections.
- Sensor connections.

- Active cooling connections.
- Mechanical/structural anchors.
- Master disconnect switch.
- Peltier temperature control/cooling system.
- Wiring harness connections from motherboard to control panel.
- Inverter bus.
- Charge controller buss.
- AC charger bus.
- Master control module buss.

As can be seen, the desired final product was to have been a full-featured solar generator ready for the commercial market. Due to the ambitious scope of the project, the project was broken down into several smaller, more manageable subsets.

1.2 PROBLEM STATEMENT

As a subset of the larger project, we want to design and verify a control scheme for a GaN-based inverter. This will be a basic control scheme that sets output voltage and frequency. Additionally, we want to identify any challenges that might arise from using GaN in a circuit and determine how they can be mitigated.

1.3 THESIS OBJECTIVES

This objective of this thesis is to design a basic control scheme for a single-phase, GaN-based inverter using the circuit modelling software PSIM[®]. This control scheme will use current and voltage feedback to provide the inverter with a stable 60 Hz sinusoidal output at a constant voltage. This initial control scheme will serve to verify PSIM[®] as a design tool for further GaN inverter development.

So far, I've given an introduction to solar inverters and the goal of this thesis. Chapter 2 will cover what GaN is and the benefits and challenges it brings to the project. Chapter 3 will discuss the selection of the inverter topology and switching scheme best suited for this purpose. Chapter 4 will give an overview of the GaN-based evaluation board. Chapter 5 will cover building the simulation model and developing the control scheme. Chapter 6 will present the results of the simulation of the inverter control scheme and its results as implemented in hardware. Finally, Chapter 7 will conclude with a summary of the thesis and what's yet to be accomplished.

CHAPTER 2: BACKGROUND

2.1 GAN SEMICONDUCTORS

Gallium nitride is a wide bandgap (WBG) semiconductor. The bandgap refers to the amount of energy needed for an electron to jump from the valence band to the conduction band. This energy is measured in electron-volts (eV). GaN has a bandgap energy of 3.39 eV, silicon (Si) has a bandgap energy of 1.12 eV, and silicon carbide (SiC), another WBG semiconductor, has a bandgap energy of 3.26 eV [3]. This wide bandgap, along with its crystal structure, gives GaN many of its intrinsic benefits.

2.1.1 GAN ADVANTAGES

There are several advantages to using GaN that make it attractive for power converters. These include lower intrinsic leakage current, higher operating temperatures, low on-resistance, higher breakdown voltage, and faster switching speeds [3]. Compared to Si, GaN has almost half the total switching losses [3], has about 30 percent faster electron mobility, and is about 10 times less likely to fail in high voltage designs [4]. With these characteristics, using GaN transistors in a circuit can increase efficiency, enable higher power density, and reduce the size and number of some components, thereby reducing overall size.

2.1.2 GAN DISADVANTAGES

The disadvantages to using GaN are that, compared to Si, some GaN transistors are more expensive to produce for what may only be a marginal improvement [5]. With manufacturing improvements and economies of scale the cost to produce GaN transistor is projected to decrease [3]. Also, with GaN devices in a circuit, there is a greater need to design the circuit for maximum electromagnetic interference (EMI) attenuation. This additional complexity can also increase design time and board cost.

2.1.3 INDUSTRY ADOPTION OF GAN

The power semiconductor industry is adopting GaN and SiC at an increasing rate. Next-generation converters for increased vehicle electrification, smart devices, and other applications are driving this demand. The GaN market is projected to increase from revenues of \$27 million in 2016 to \$320 million in 2022 [6]. This increased demand and production will likely drive down the cost of these devices.

In an effort to accelerate the adoption of WBG devices, Google started their Little Box Challenge (LBC) with a \$1 million prize in 2014. The design goal of this competition was to create a 2 *kVA* inverter with a power density greater than 50 *W/in*³ [7]. This event successfully helped to raise awareness and industry adoption of GaN.

2.2 LITERATURE REVIEW

The choice of topology for the inverter was informed by a survey [7] of designs used in the Google LBC which found most competitors used a full-bridge topology. [8], [9], [10], and [11] are power converter textbooks used to obtain circuit equations and compare pulse width modulation (PWM) switching methods for the full-bridge inverter. Using a PI controller for inverter feedback and control was borrowed from [12] and a helpful PSIM[®] tutorial was accessed here [13].

Additionally, [3] and [14] provided an in-depth analysis of GaN devices and their dynamics in a circuit that served to broaden our understanding of future layout considerations as the inverter is scaled up.

CHAPTER 3: THE FULL-BRIDGE CONVERTER

3.1 TOPOLOGY CONSIDERATIONS

There are myriad circuit topologies that work as an inverter. Choosing the best topology can be difficult and challenging. This project needed a topology that would have minimal side effects from high frequency switching, high power capability, and relatively simple implementation. To satisfy these needs, a full-bridge topology was chosen. The full-bridge, or H-bridge, is a proven topology with many examples featured in classic power electronic textbooks.

3.2 FULL-BRIDGE TOPOLOGY

The full-bridge is a switching topology that consists of four switches arranged as shown in Figure 3.1. The switch symbols in the circuit represent semiconductor transistors but, for ease of illustration, are here shown as switches. Here V_s represents a DC source voltage and V_{sw} represents the AC output of the bridge.

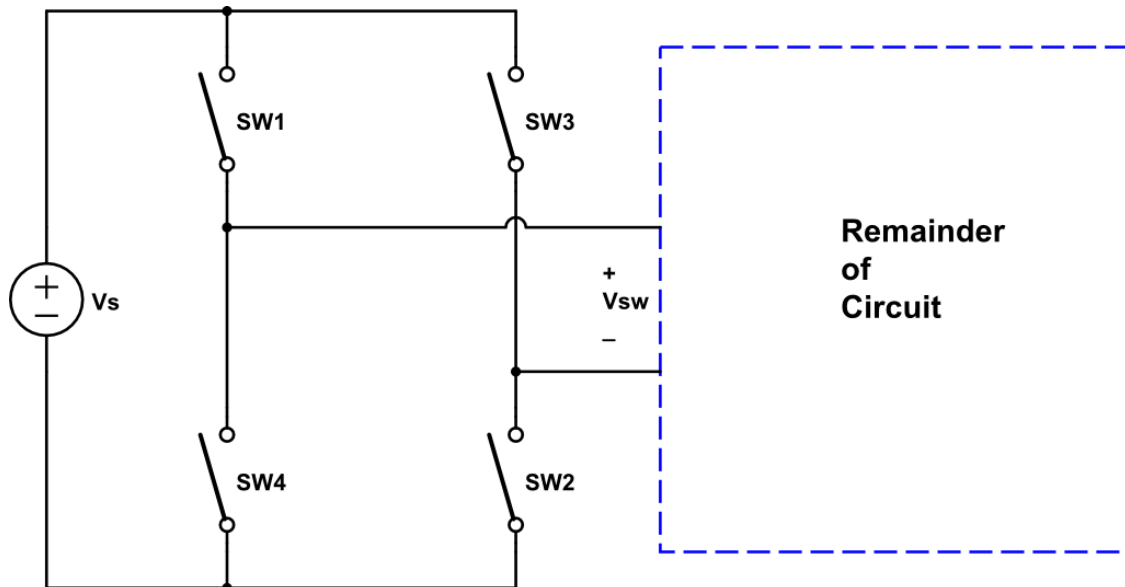


Figure 3.1: The full bridge switch arrangement

The full-bridge topology has an advantage for high power and high-frequency when compared to some other topologies. The maximum voltage across an open switch in a full-bridge is only V_s as compared to $2 * V_s$ in other converter topologies such as the forward or push-pull [9]. It is important to have low voltage stresses on the switches to increase life expectancy, and improve dv/dt related transients.

The full-bridge is comprised of two “legs” where SW1 and SW4 comprise one leg and SW3 and SW2 are the second leg. The switches of each leg must never be on (conducting) at the same time or else the source voltage, V_s , would be shorted. To send energy to the remainder of the circuit SW1 and SW2, as a pair, are turned on alternately with SW3 and SW4. Figures 3.2 and 3.3 show the path of conduction through the bridge when each switch pair is conducting.

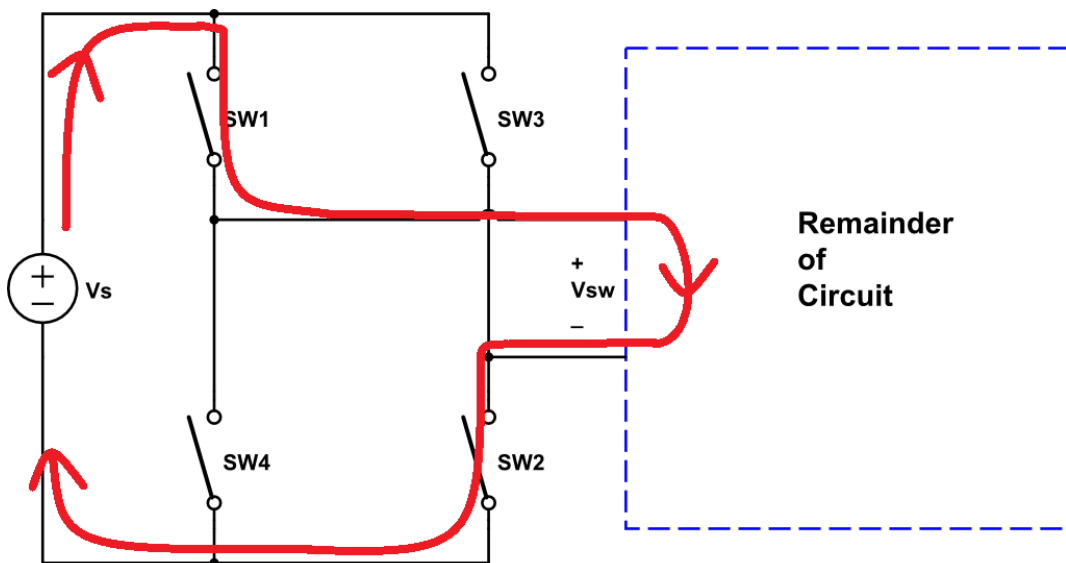


Figure 3.2: With SW1 and SW2 on, the circuit sees $+V_s$

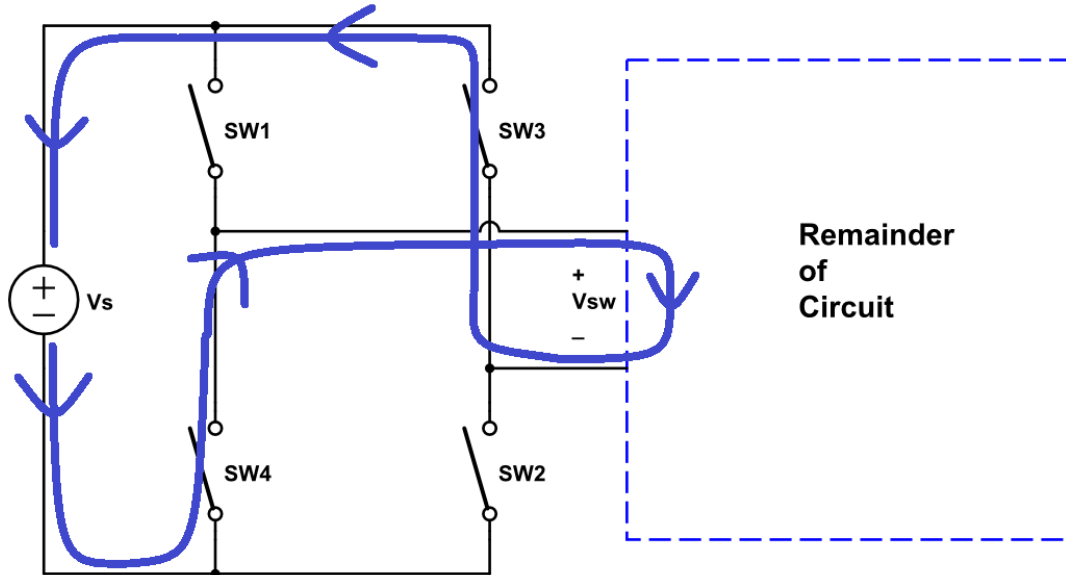


Figure 3.3: With SW3 and SW4 on, the circuit sees $-V_s$

Additionally, there is a third state where the voltage across the circuit is zero. This is shown in Figure 3.4 and occurs when either both high side switches are on with both low side switches off, or visa versa. This effectively shorts the circuit output.

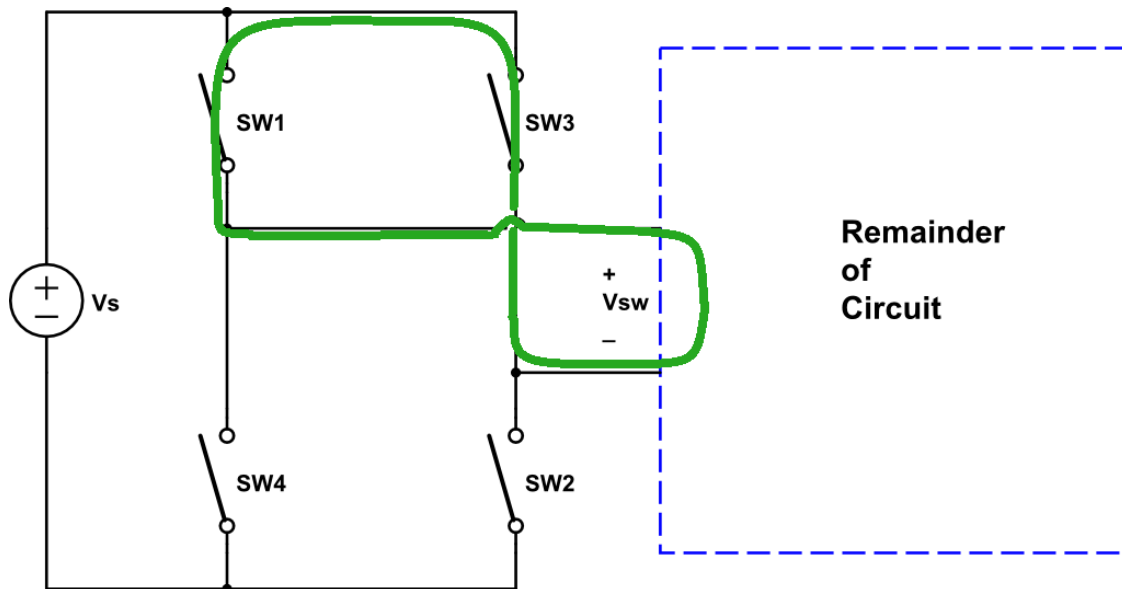


Figure 3.4: With both SW1 and SW3 on, the circuit sees $0V$

3.3 SWITCHING THE FULL-BRIDGE

To operate the converter, the switches need to be switched on and off according to a set pattern to get the desired output. This can be done many ways. One of the simplest, for full-bridge converters, is a method called PWM.

3.3.1 PULSE WIDTH MODULATION

PWM is a method that can be used to generate a pattern of fixed or varying width pulses to control the full-bridge output [15].

To generate this pattern, two waveforms are compared. The carrier waveform is usually a triangular or sawtooth wave and the other is called the modulating waveform. The modulating waveform is sinusoidal for inverters and a DC reference for DC-DC converters. Figures 3.5 and 3.6 show both of these waveforms.

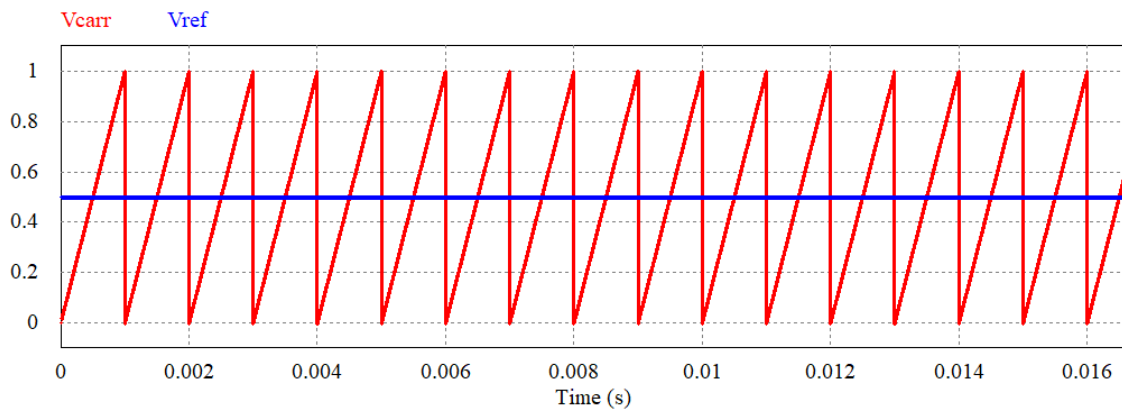


Figure 3.5: Comparison of a DC reference voltage V_{ref} (blue) and a 1 kHz sawtooth carrier waveform V_{carr} (red)

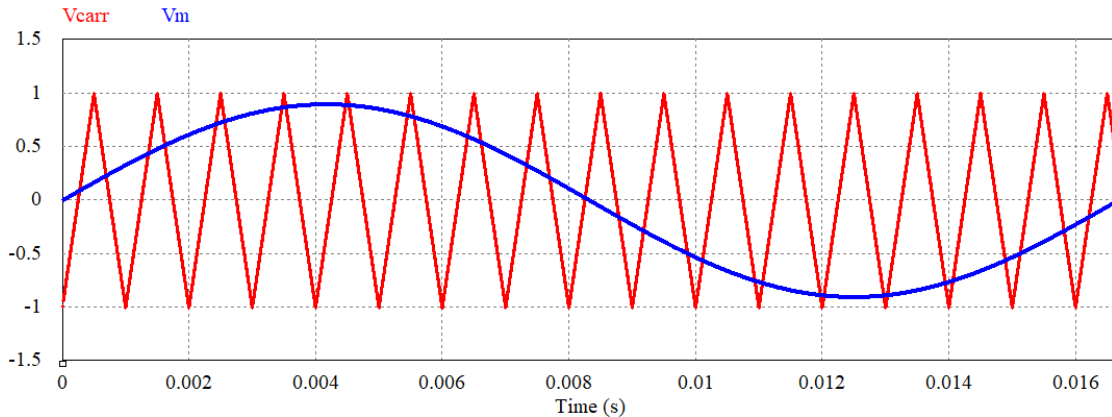


Figure 3.6: Comparison of a 1 kHz triangular carrier waveform (red) to a 60 Hz modulating waveform (blue)

The PWM scheme compares these waveforms to each other in a comparator and, based on their relationship to each other, the comparator output is either high or low. This output provides the switching signals to the switches. The following equations relate the output to the waveforms where V_C is the comparator output:

$$V_m > V_{carr}, V_C = 1 \quad (3.1)$$

$$V_m < V_{carr}, V_C = 0 \quad (3.2)$$

It can be seen that with a sinusoidal modulating waveform, the width of the pulse increases as the modulating wave reaches its peak and decreases as it approaches the lowest portion of its cycle. This form of PWM in inverter applications is also called sinusoidal pulse width modulation (SPWM).

There are two important relationships between the carrier and modulating waveform. These are called the frequency modulation ratio m_f and the amplitude modulation ratio m_a [9]. The frequency modulation ratio is the ratio of the frequency of the carrier wave to the frequency of the modulating or reference wave.

$$m_f = \frac{f_{carr}}{f_{ref}} \quad (3.3)$$

The amplitude modulation index is the ratio of the amplitude of the modulating waveform to the amplitude of the carrier waveform.

$$m_a = \frac{mod_a}{carr_a} \quad (3.4)$$

3.3.2 BI-POLAR SWITCHING

Bi-polar switching is the the simplest method of switching a full-bridge. Each diagonal pair of switches are turned on alternately based on the PWM output. Figure 3.7 shows the output of the bridge using the PWM generation waveforms shown in Figure 3.6.

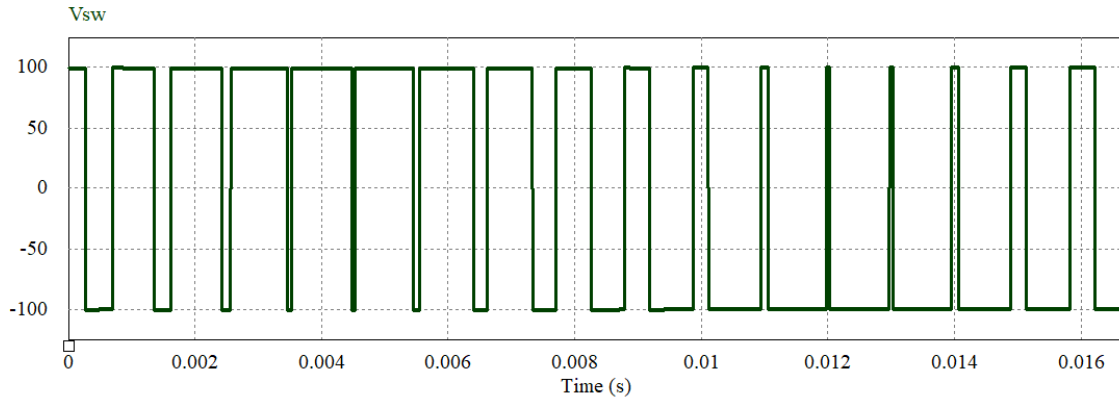


Figure 3.7: Switching waveform resulting from bi-polar switching at 1 kHz

When the PWM output is high, SW1 and SW2 are on while SW3 and SW4 are off. When the PWM output is low, SW3 and SW4 are on while SW1 and SW2 are off. This results in a voltage swing of $2V_s$ as its output switches between $+V_s$ and $-V_s$.

3.3.3 UNI-POLAR SWITCHING

There is another way to switch a full-bridge inverter that reduces the voltage swing. Instead of switching between $+V_s$ and $-V_s$, it switches between $+V_s$ and 0 for the positive half-cycle of V_m and between $-V_s$ and 0 for the negative half-cycle. This is called uni-polar switching. To accomplish this, SW1 is kept on for the positive half-cycle while SW3 and SW2 alternate. With SW1 and SW2 both on V_{sw} is equal to V_s , and when SW1 and

SW3 are both on, V_{sw} is zero. For the negative half-cycle SW4 is kept on while SW3 and SW2 alternate. Figure 3.8 shows the resultant switching waveform from using uni-polar switching. This uses the same 60 Hz modulating waveform and 1 kHz carrier frequency that's shown in Figure 3.6.

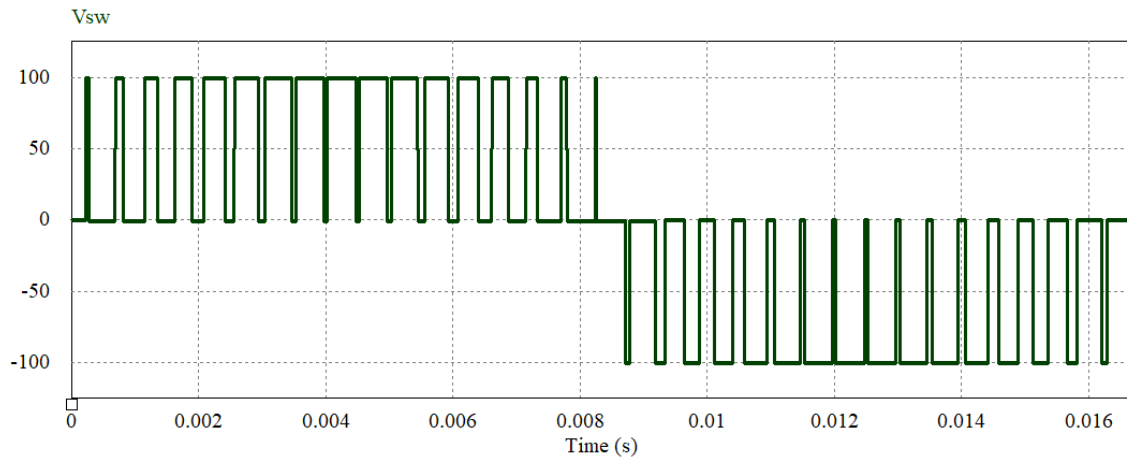


Figure 3.8: Switching waveform resulting from uni-polar switching of a full-bridge

3.3.4 SWITCHING HARMONICS

There are benefits to use uni-polar switching over bi-polar switching [8]. These include lessened voltage spikes from the turn-on and turn-off transients of the transistor switches and improved harmonics when compared to bi-polar switching. Since the output of the uni-polar scheme more closely resembles an actual sine wave, it will have fewer undesirable harmonic components. Figures 3.9 and 3.10 show a Fourier spectrum of the bi-polar uni-polar switching waveforms shown in Figures 3.7 and 3.8, respectively.

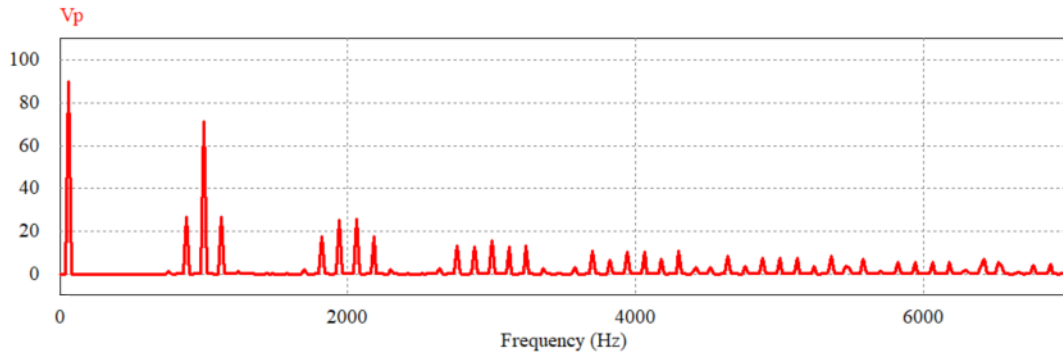


Figure 3.9: An FFT of the bipolar switching waveform shown in Figure 3.7

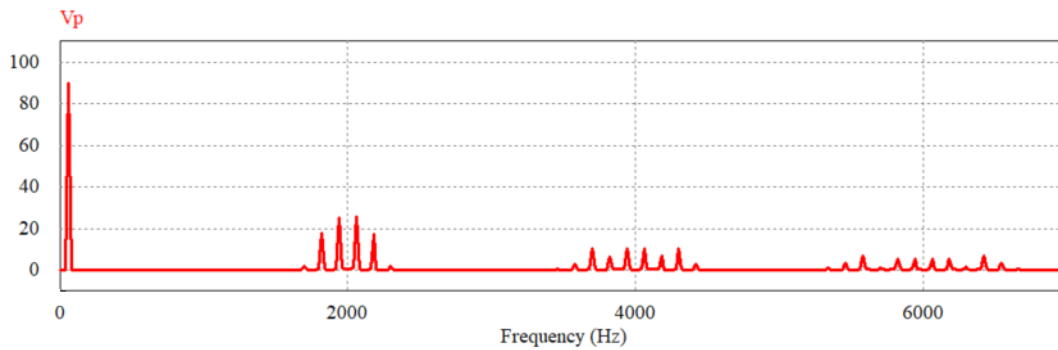


Figure 3.10: An FFT of the uni-polar switching scheme shown in Figure 3.8

As can be seen, with the same 1 kHz switching frequency, the uni-polar scheme has fewer harmonic components. For these examples $m_f = 16.67$ and the higher harmonics occur at intervals of the frequency of f_{carr} . However, these figures are mainly for illustrative purposes. The frequency of the carrier waveform and hence, m_f , will be much higher in practice. Then the high frequency harmonics can be filtered out with a low pass filter on the output stage. Because of its reduced harmonic content, this project uses the uni-polar switching scheme.

CHAPTER 4: HARDWARE

4.1 THE TRANSPHORM TDPV1000E0C1 SINGLE-PHASE INVERTER

The Transphorm TDPV1000E0C1 evaluation board was used as a test bed to design and build the full-bridge inverter switching and control scheme. This is a single-phase high-frequency GaN-based inverter built by Transphorm for use as a design reference and for testing and evaluating their GaN devices. This inverter is shown in Figure 4.1.

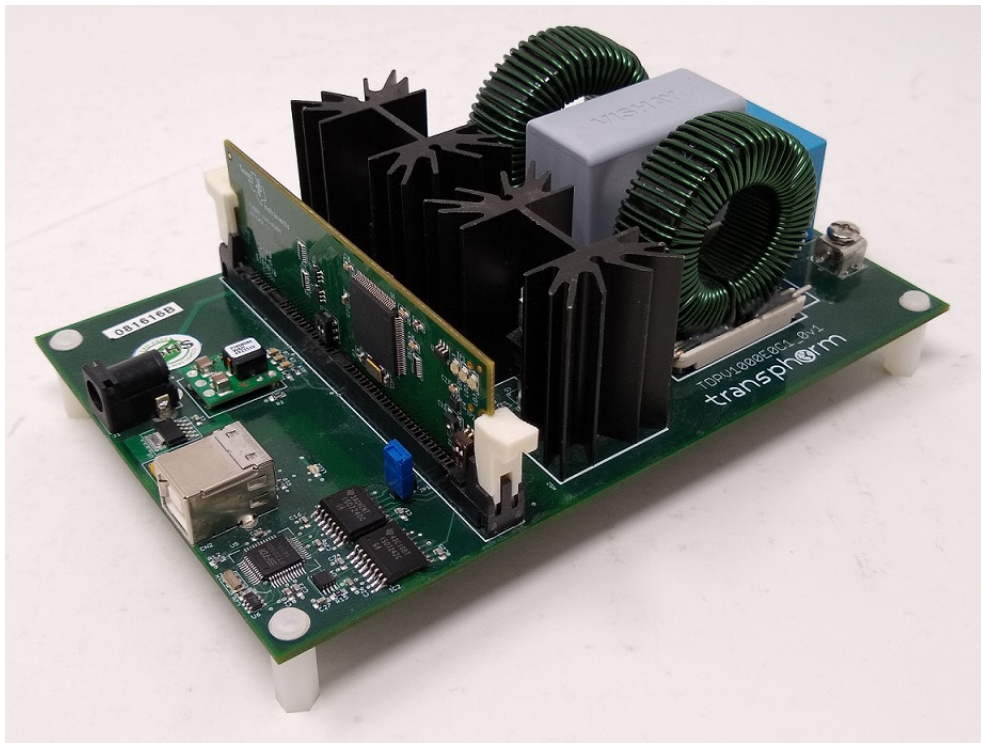


Figure 4.1: The Transphorm TDPV1000E0C1 evaluation board

The specifications for the Transphorm TDPV1000E0C1 inverter [16] are as follows:

- 0-400 V_{dc} input.
- Output of $V_{dc}/\sqrt{2}$ V_{rms} at 50/60 Hz , up to 1000 VA .

- Can support a PWM frequency of 100-200 kHz .

Figure 4.2 shows a simplified circuit of this inverter. This inverter has a separate 12 V input to supply power to the microcontroller, current sensors, gate drivers, and other needed hardware.

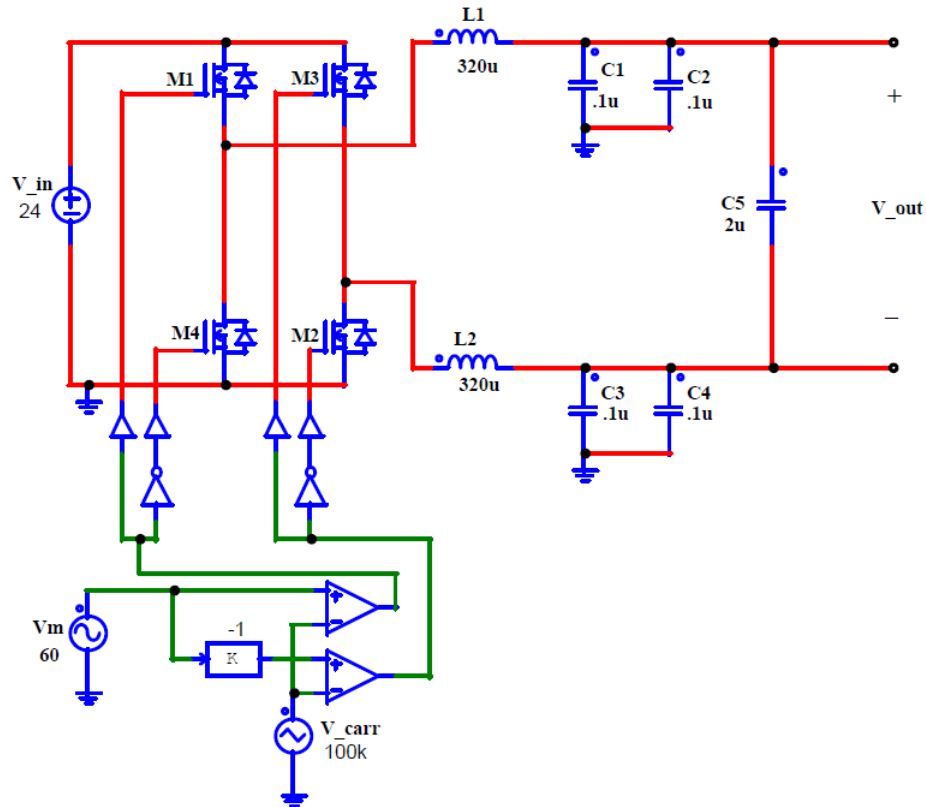


Figure 4.2: The Transphorm TDPV1000E0C1 simplified circuit

Figure 4.3 shows a typical efficiency curve of this inverter. As seen, it reaches its peak efficiency at approximately 450 W of output power. However, it's not readily apparent if this figure includes the losses from the power needed to supply the on-board control circuitry.

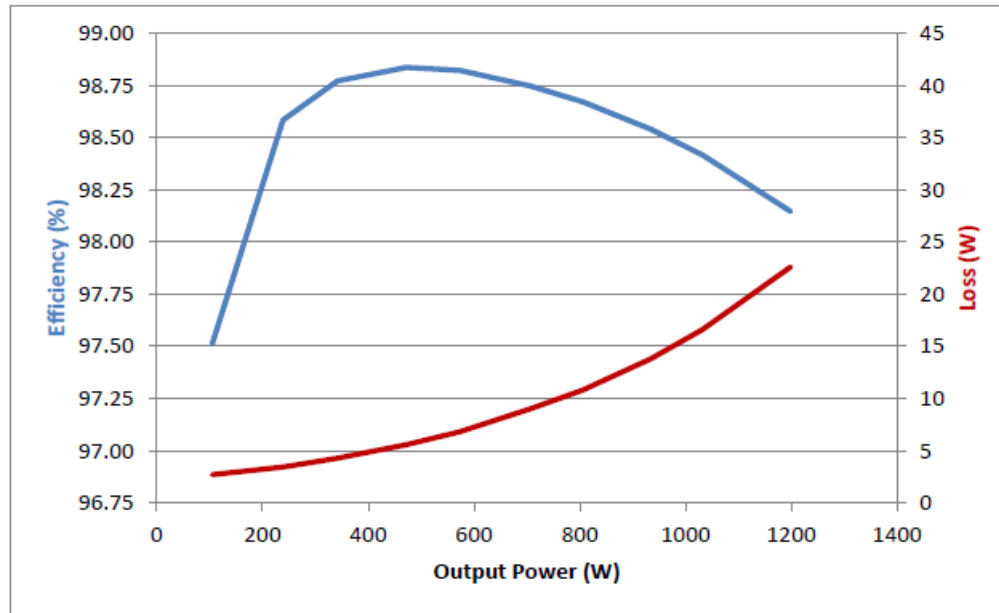


Figure 4.3: Typical efficiency with 350 V_{dc} input and 240 V_{ac} output [16]

4.1.1 THE TI C2000TM SERIES MICROCONTROLLER

A Texas Instruments (TI) C2000TM Series microcontroller is the brain that runs the inverter. It controls the switching of the full-bridge and can be configured to monitor any measurable aspect of the inverter and respond appropriately if the condition calls for it.

Within the many members of the TI C2000TM family, the TMS320F28069 was chosen for this project. Of the many microcontrollers available, this one has the most desirable features [17]. These included:

- A 90 MHz clock to enable high frequency switching.
- 16 configurable PWM channels, 8 of which can be enabled with high resolution.
- The analog to digital converter (ADC) has 12 bits of resolution plus noise filtering and clamping diode protection on its input pins.
- 256 KB of flash and 100 KB of random access memory (RAM).

- Many communication ports and general purpose input and output (GPIO) pins for future peripherals.

The TMS320F28069 ordered came as part of a experimenter's kit which meant that it was mounted on a board with easily accessible pins. These pins, when hooked up to an oscilloscope, helped to verify its output. This board also readily connects via USB to a computer for programming. A picture of this board is shown in Figure 4.4.

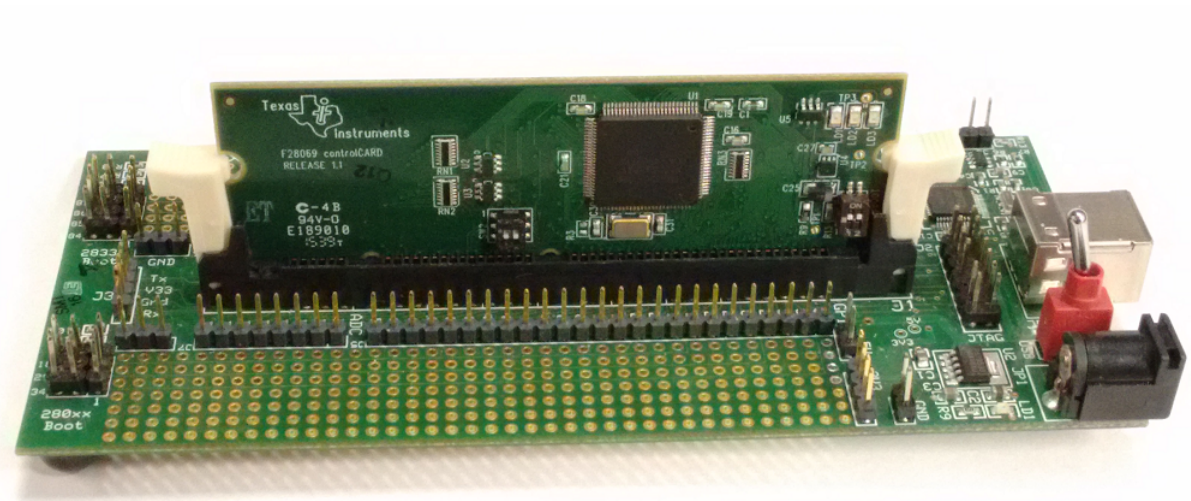


Figure 4.4: The TI experimenter's board with the TMS320F28069 card mounted

This Transphorm evaluation board came equipped with a TI TMS320F28035 controller from the same TI C2000 family. It was replaced with this TMS320F28069 card.

4.1.2 GAN FETs

The gallium nitride field-effect transistor (GaN FET) devices on this board were made by Transphorm. There are several manufacturers of GaN devices but few that make them in TO-220 and TO-247 packages. This inverter board uses their TPH3206PS transistors in TO-220 package. These are rated at 17 A of continuous current and 600 V [18].

The packaging of these devices is a very important consideration. All high power converters need some type of heat sink to dissipate heat, thereby keeping the switching de-

vices at an acceptable operating temperature. With many surface-mount devices (SMD)s, mounting a suitable heat sink can be challenging due to their small surface area. With the TO-220 and TO-247 through-hole packages, the heat sink is easily attached to the back of the device. At the time this project started, Transphorm's devices were the only ones available with this package.

4.1.3 GATE DRIVERS

Since the output of a microcontroller is insufficient to drive most FETs, high voltage gate drivers are needed to supply the necessary voltage and current to the gates of these transistors to turn them on and off. There are a few capacitances associated with the gate such as the gate to source C_{gs} and the gate to drain C_{gd} that need to be charged before the desired voltage can be established. Thus, the gate drivers need to be able to supply that current quickly.

The Transphorm inverter uses a Silicon Labs SI8230 gate driver. This is an isolated dual gate driver, a single package that drives the gates of the high and low sides of each full-bridge leg. These were recommended by Transphorm to use for driving their GaN FETs.

4.1.4 FERRITE BEADS

Ferrite beads are passive components that are essential in high-frequency converters to suppress some of the inherent noise and EMI associated with high-frequency circuits. They can either be inserted in series or wrapped around a conductor. When in a circuit and high frequencies are present, the ferrite bead appears resistive to those high frequencies and converts and dissipates some of that energy as heat. There are many different shapes and styles of ferrite beads that, based on their properties, will suppress different frequencies. These are an important component to mitigate high frequency noise and as such they are common in high frequency design. Transphorm's inverter has these throughout their design.

These are just a few of the design considerations and components that Transphorm used in this board to enable high-frequency operation[16]. Good high-frequency design is essential for any GaN power converter.

CHAPTER 5: MODELING THE INVERTER

5.1 PSIM[®]

PSIM[®], developed by Powersim Inc. of Rockville, MD, is the simulation software used to model this inverter. It is designed for use in modelling power electronics, motor drives, and power conversion systems [19]. In conjunction with PSIM[®], two other modules proved invaluable to this project. These are the SmartCtrl[©] and SimCoder[™] modules. PSIM[®] version 11.1.3 was used for this project.

5.1.1 SMARTCTRL[©]

SmartCtrl[©] is the module used to design the feedback loops to regulate the output voltage of the inverter. A correctly modeled inverter enables a quick design process. With a correct model, many different types of control schemes can be devised and implemented in the simulation model and, subsequently, implemented in hardware with the SimCoder[™] module. I used version 3.0 for this project.

5.1.2 SIMCODER[™]

PSIM[®] was chosen for this project primarily because of the SimCoder[™] module. This module has many function and digital control blocks specific to different members of the TI C2000[™] family of microcontrollers that can be used in simulation. When used in a PSIM[®] simulation, PSIM[®] will also generate the C-code needed to implement that function on the microcontroller. This code is then easily imported into Code Composer Studio, TI's integrated development environment (IDE), and loaded onto the microcontroller. This ability is extremely invaluable for rapid prototyping, and also reduces the need to have someone who is adept at programming these microcontrollers.

5.2 MODELLING WITH PSIM[®]

To model the inverter I first built the circuit in PSIM[®] using non code generating blocks. I will walk through the design of the control scheme using this PSIM[®] model along with the SmartCtrl[©] module. Lastly, once the control scheme is finished, the model will be built with code generating blocks to allow SimCoder[™] to generate the microcontroller code.

5.2.1 CONTROL DESIGN PROCESS

PSIM[®]'s SmartCtrl[©] module uses the measured frequency response of a circuit to approximate the transfer function of its plant. The plant is a mathematical representation of output divided by input. When viewed in a feedback block diagram, the system's output is fed back and compared to its set point. The resultant difference, or error, is then compensated, and that compensated signal is then fed back into the plant so that the output of the plant has less error [20]. SmartCtrl[©] has built-in compensator models and sensor models so that it can automatically insert those models into the block diagram and calculate values needed to correct the system output.

To regulate the output of this inverter, a proportional-integral (PI) control scheme was designed. The PI control scheme was chosen because it can give a small steady-state error [12]. It is also a simpler control scheme than some to understand and implement and so it was deemed sufficient for the first hardware prototype. The transfer function of the SmartCtrl[©] PI compensator is defined [21] as

$$G(s) = Kp * \frac{1 + Ti * s}{Ti * s} \quad (5.1)$$

The control scheme for this inverter has two feedback loops; an inner current loop and an outer voltage loop. The inner current loop uses feedback from the current sensor while the outer control loop uses feedback from the voltage sensors at the output.

To initiate the design process, a model of the inverter was built in PSIM[®] using non code generating blocks. This starts out as the same circuit pictured in Figure 4.1 but now has an added load with current and voltage sensors. This circuit is shown in Figure 5.1.

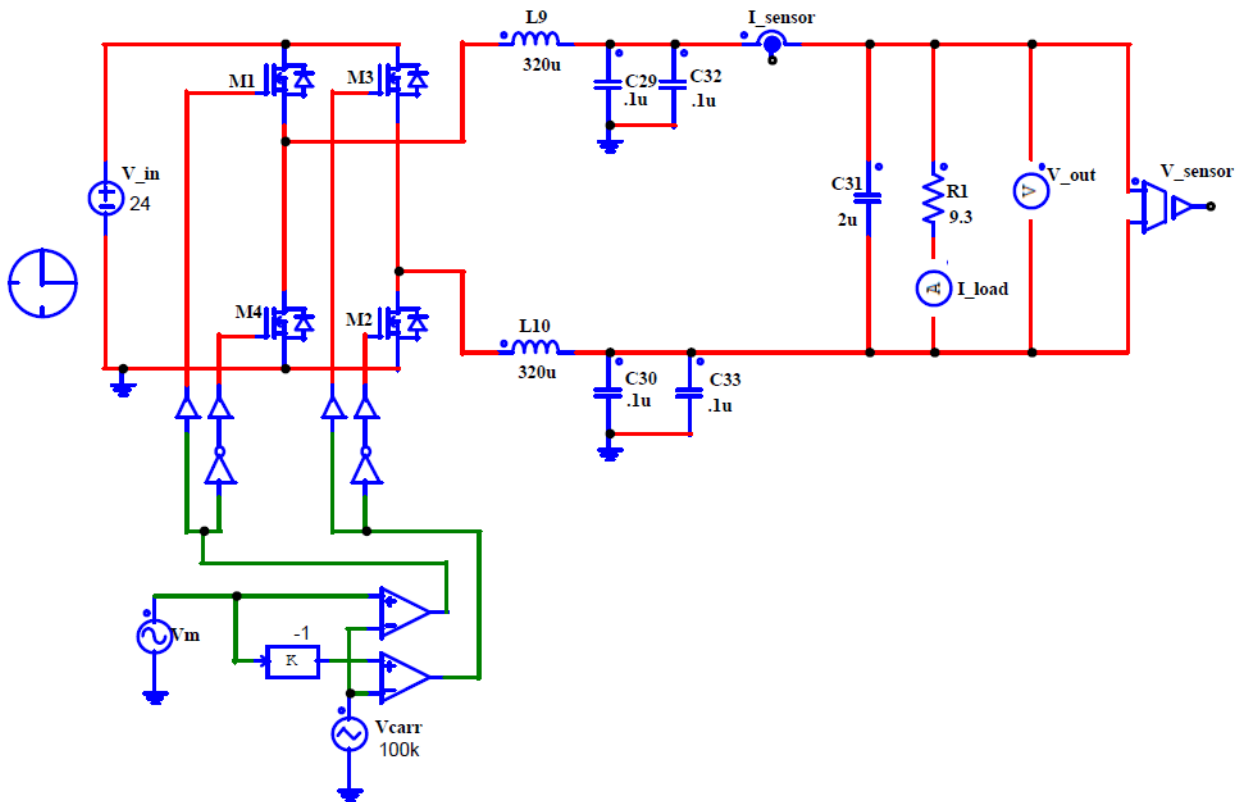


Figure 5.1: Initial circuit diagram

The load is modelled as a resistive load of $9.3\ \Omega$. This load value corresponds to the measured resistance of the two series-connected MEMCOR $5\ \Omega$, $53\ W$ resistors shown in Figure 5.2. The input voltage is limited to $24\ V$ as that is the maximum DC voltage considered safe in the lab for a solitary person to work with and the switching frequency is $100\ kHz$.

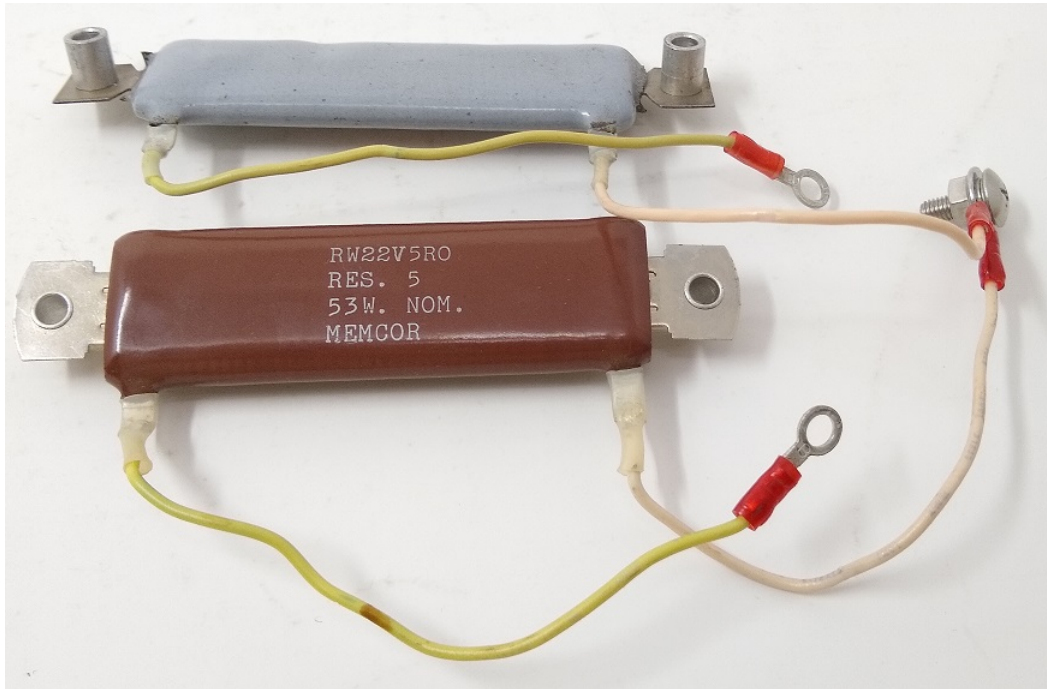


Figure 5.2: Load comprised of two series-connected resistors

In this circuit the GaN FETs are modelled as ideal switches to reduce simulation length. The current sensors on this board are the Allegro ACS712-20A Hall effect linear current sensors. The sensitivity of this sensor is 100 mA/V with a 2.5 V DC offset. On the board, the output of this sensor is further reduced by a voltage divider in order to scale the signal for the $0\text{-}3 \text{ V}$ input range of the ADC. This voltage divider provides a scaling factor of 0.661 , which, when multiplied by the sensitivity, becomes 0.0661 . To sense the voltage, a pair of identical voltage dividers are located on either side of the load capacitor. This provides a scaling factor of 0.008 for the load voltage. On the board, these two signals are compared in the ADC to determine the load voltage. For the initial control design, neither the voltage dividers for the voltage nor the current sensors are needed since they can simply be specified. When the code-generating circuit is built, they will be added so that their signals are within the constraints of the ADC.

5.2.2 INNER CURRENT FEEDBACK LOOP

The control is designed in two stages: first the inner current feedback loop is designed and secondly the outer voltage feedback loop is added with the current feedback loop in place. This circuit model starts out as open loop inverter for the design process.

To initiate the design process of this converter an AC sweep of the inverter is needed to obtain the frequency response of the current sensor output. To do this, a sinusoidal perturbation is added to the input of the comparator and an AC sweep probe is added to the output of the current sensor. The perturbation source injects a range of frequencies in steady-state in the time domain and the probe measures the response due to the injected frequencies. This gives a transfer function which can then be imported into SmartCtrl[©] to determine the parameters of the PI compensator. This circuit is shown in Figure 5.3.

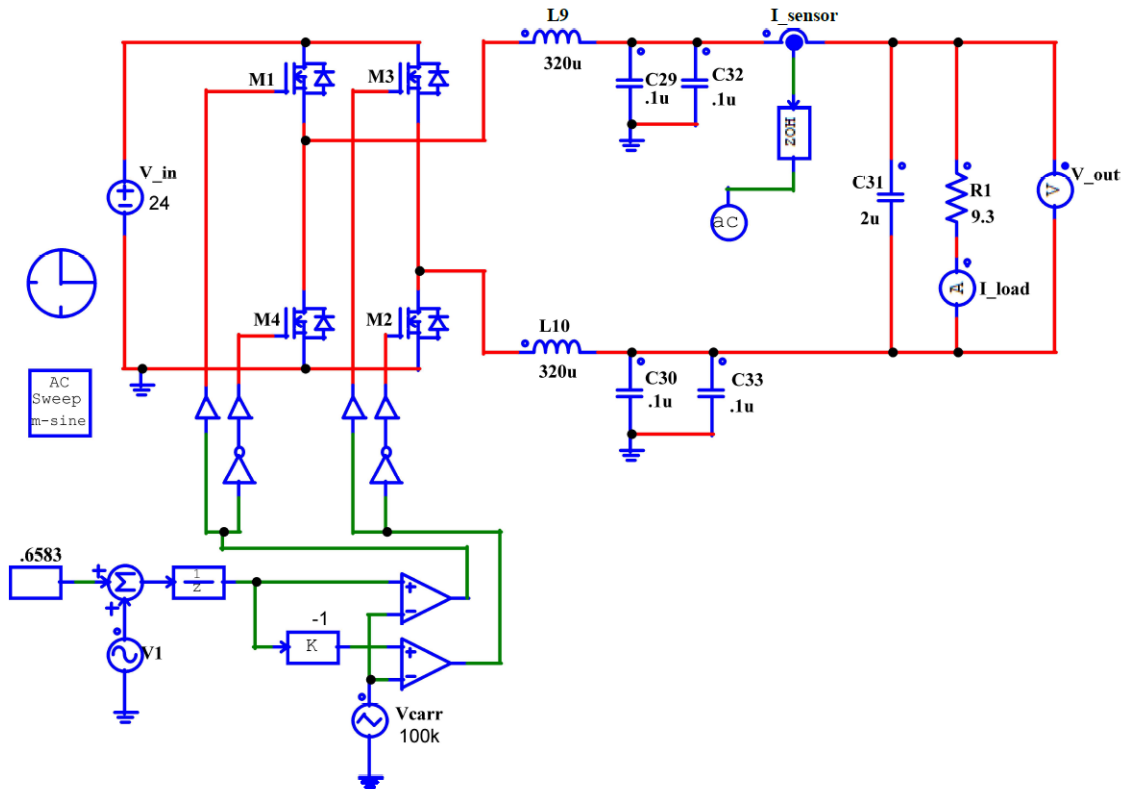


Figure 5.3: Circuit with AC sweep for current sensor frequency response

The AC sweep block in the diagram is where the sweep parameters are specified. This was swept from 10 Hz to 60 kHz . This was started at 10 Hz because starting from that frequency reduces the time needed for the solve. It ends at 60 kHz because as the ending frequency approaches the switching frequency divided by two, or 50 kHz for a it may run into Nyquist sampling problems which can interfere with the result (see the phase response in Figure 5.5 at 60 kHz). Figure 5.4 shows the parameters entered for this sweep.

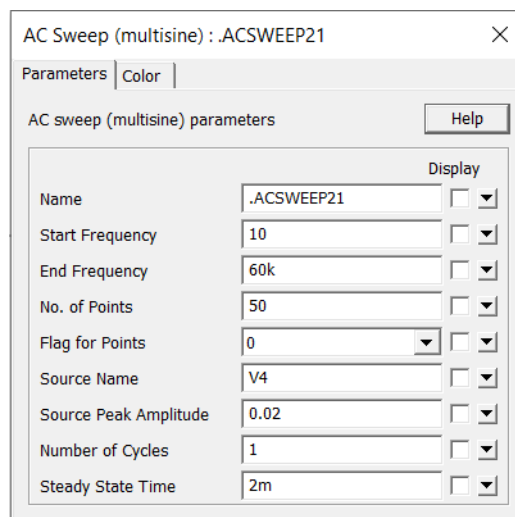


Figure 5.4: AC sweep parameters

Also, since this will be a digitally controlled inverter but the model is in the time domain, a digital response of the sweep can be synthesized by adding a $1/z$ unit delay block to the input of the comparator and a zero order hold (ZOH) block to the output of the current sensor. Both of these have a 100 kHz sampling frequency identical to the switching frequency.

The output voltage and current values to control were specified semi-arbitrarily since I'm using an artificially low voltage to develop this control scheme. I chose a peak output voltage of 15.8 V and with the 9.3 Ω resistive load, a peak current of 1.69 A results.

Instead of a sinusoidal modulating waveform to provide a comparative waveform to the carrier wave, I'm specifying a fixed duty cycle for the AC sweep and will then use

this value as the peak voltage of a sinusoidal modulating waveform to give an AC output. Given the ratio of the specified 15.8 V output to the 24 V input, a duty cycle of $D=.6583$ was calculated. This comes from the equation that $V_{out} = D * V_{in}$.

The simulation was run with a time step of 0.1 μs with a total run time of 0.07 s. The resultant Bode plot of this sweep is shown in Figure 5.5.

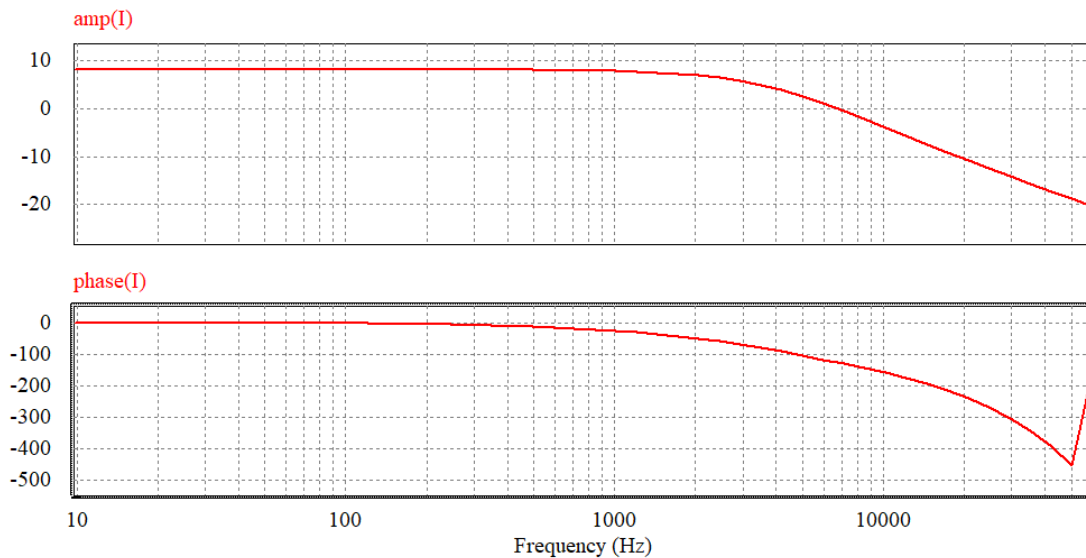


Figure 5.5: Results of the AC sweep showing amplitude and phase as a function of frequency

The AC sweep also generates a text file of the frequency response which is imported into SmartCtrl[©] to design the control loop. After exporting into SmartCtrl[©], the switching frequency, transfer function type (voltage or current), and the value of the controlled variable are entered. Next, the type of current sensor is specified (Hall effect sensor), its gain (0.0661), and the type of compensator desired (PI). Then the peak and minimum voltage of the carrier waveform is specified, as well as its rise time. These are:

$$V_p(V) = 1.0$$

$$V_v(V) = -1.0$$

$$tr(s) = 5\mu$$

Once all of this is done, as shown in Figure 5.6, a solution map is generated to help determine the optimal values for K_i and K_p .

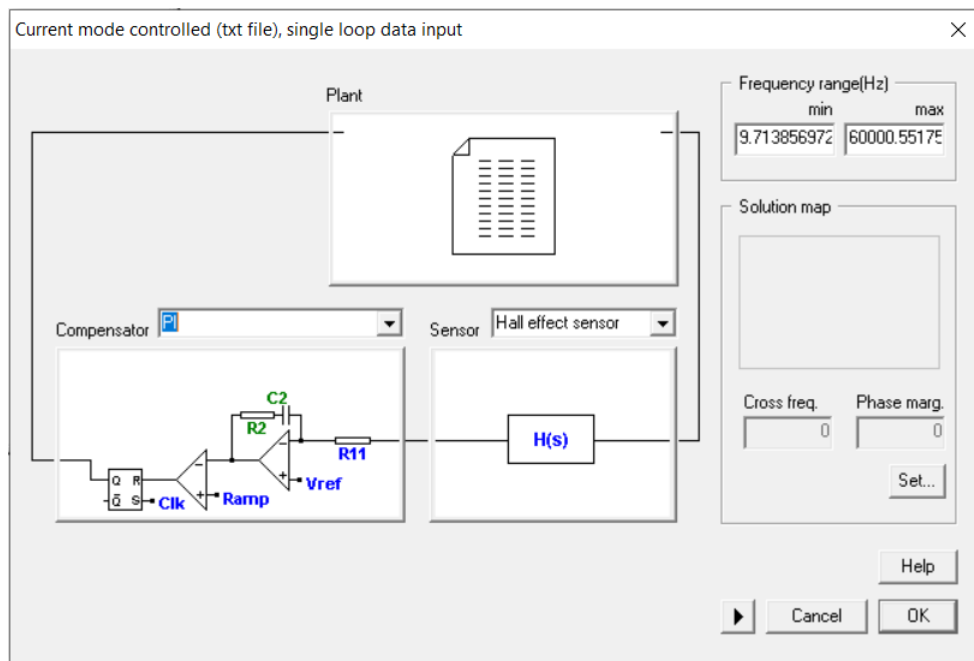


Figure 5.6: Feedback diagram with selections

This solution map, shown in Figure 5.7, shows a plot of cross frequency vs. phase margin.

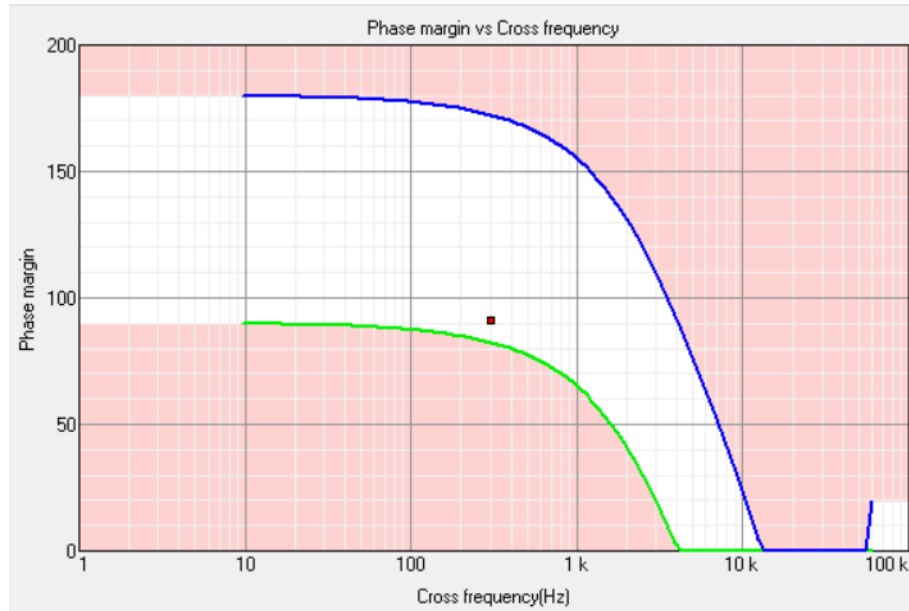


Figure 5.7: Solution map of cross frequency vs. phase margin

In the solution map, the white area is the area for which there exists stable values of K_i and K_p (denoted here as K_p and T_i , respectively) and the pinkish area denotes unstable values. When moving the dot around the solution map, SmartCtrl[©] generates the closed loop response for those values of K_p and T_i . This interface is helpful for tuning K_p and T_i . Tuning is an iterative process. Starting at a phase margin of 60° , the values for K_p and T_i that corresponded to that phase margin were implemented in the hardware. Based on the output waveform, I increased the phase margin until the values for K_p and T_i produced a clean sinusoidal voltage output. This tuning interface is shown in Figure 5.8.

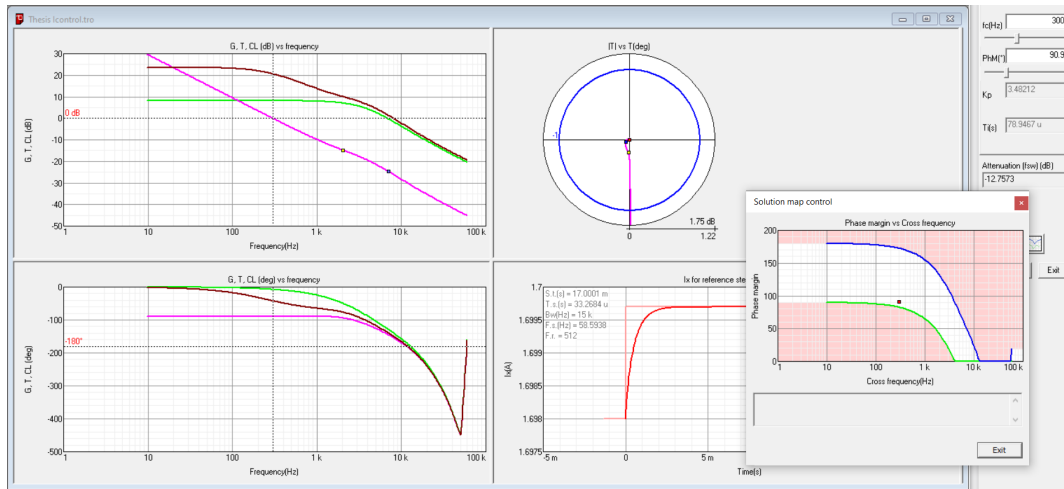


Figure 5.8: PI tuning interface

Once the values for K_p and T_i are selected, the current feedback loop on the model can be closed with a PI function block. The values ultimately chosen were:

$$K_p = 3.495$$

$$T_i = 78.9271 \mu s$$

which correspond to a phase margin of 90° for the closed loop transfer function. With these values, a digital PI block, with a limiter, was added to the circuit and the current feedback loop closed. Figure 5.9 shows the parameters of digital PI control block. Figure 5.10 shows the new circuit and Figure 5.11 shows the voltage and current output of the inverter with just the current feedback loop closed.

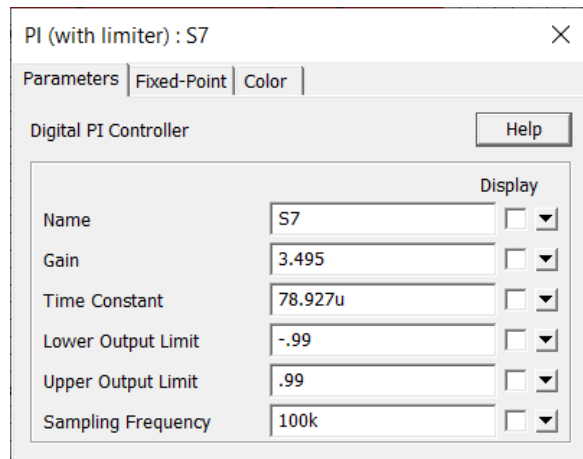


Figure 5.9: Parameters for the digital PI block

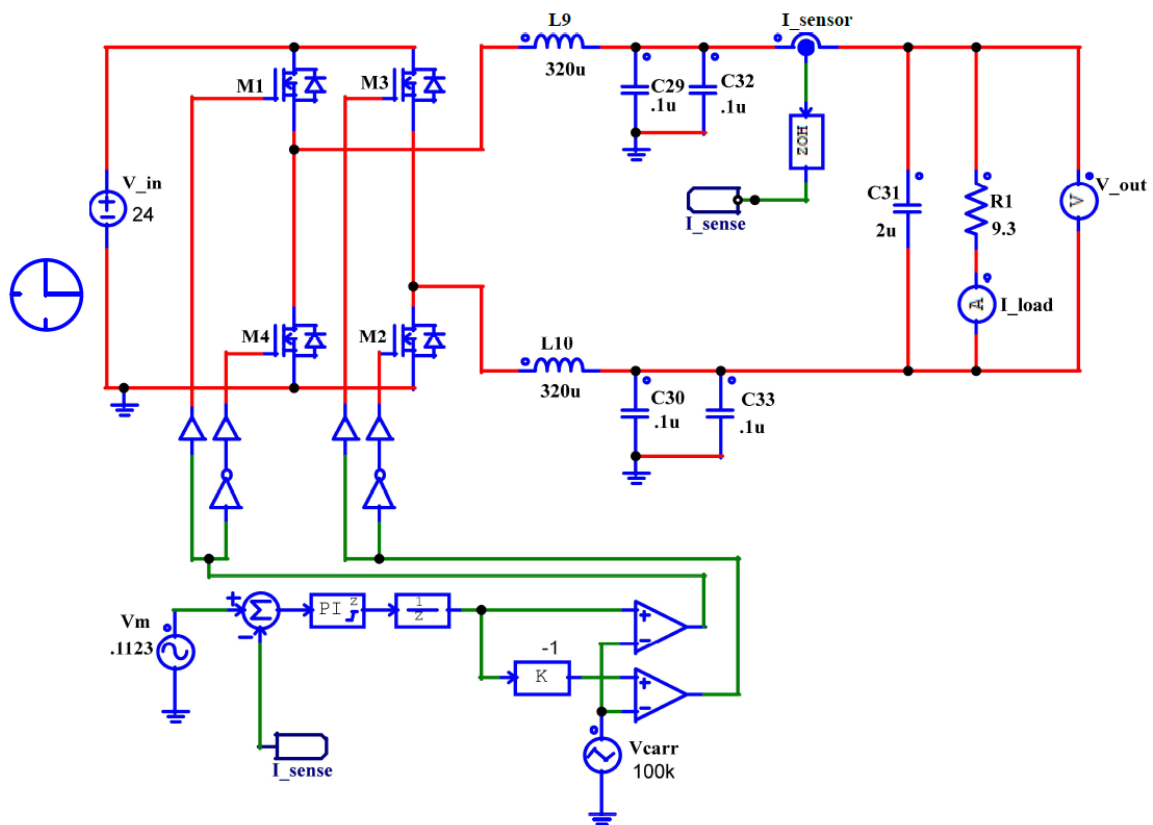


Figure 5.10: Inverter circuit with closed loop current feedback

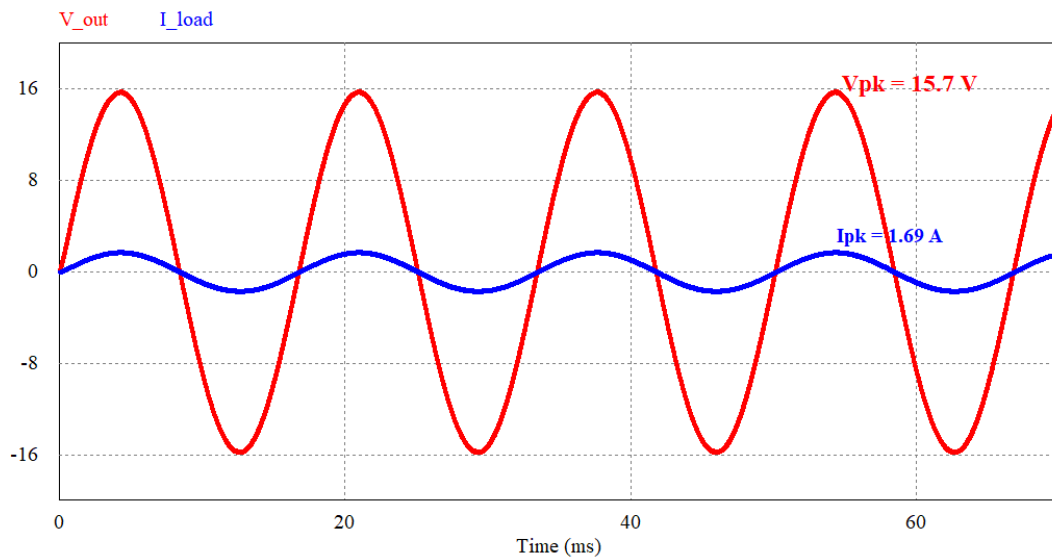


Figure 5.11: Output voltage and current with closed loop current feedback

As seen from Figure 5.11, the output voltage and current are close to the specifications.

5.2.3 OUTER VOLTAGE FEEDBACK LOOP

The steps for designing the outer voltage loop are the same as for the current loop. A voltage sensor is added across the load with an AC sweep probe on its output. Figure 5.12 shows this circuit.

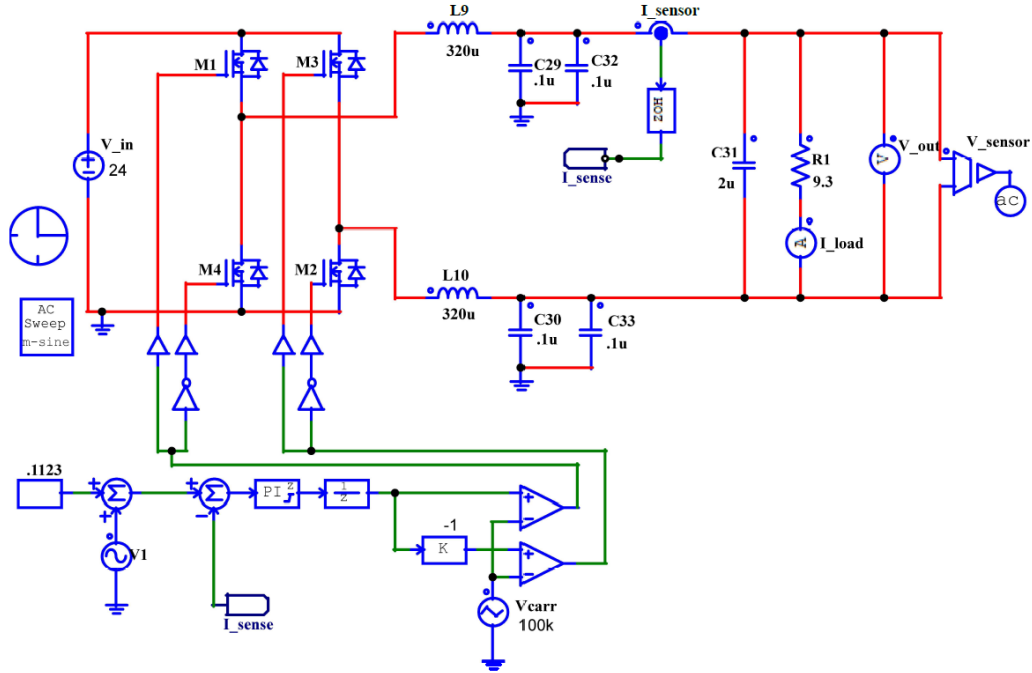


Figure 5.12: Circuit set up to sweep the frequency response of the load voltage

The AC sweep of this circuit used the same parameters as the first sweep. The Bode plot of the frequency response is shown in Figure 5.13.

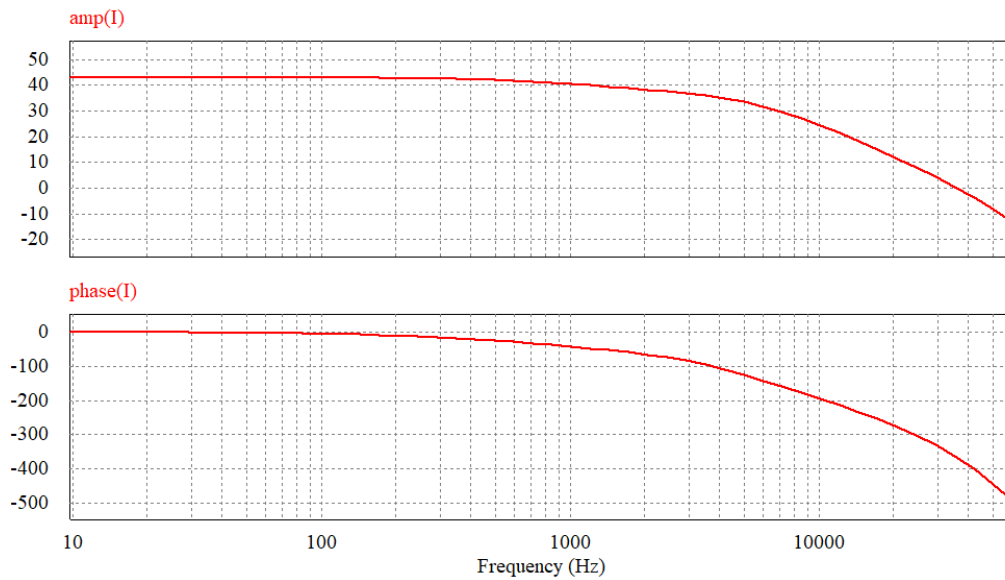


Figure 5.13: Bode plot of the output

This frequency response was imported into SmartCtrl[©] and using the same steps as

before generated a solution map to determine the PI coefficients. They are:

$$K_p = 0.157203$$

$$T_i = 173.17\mu s$$

With these values, the final circuit was built using code generating blocks specific to the TI C2000F28069 microcontroller. This circuit is shown in Figure 5.14.

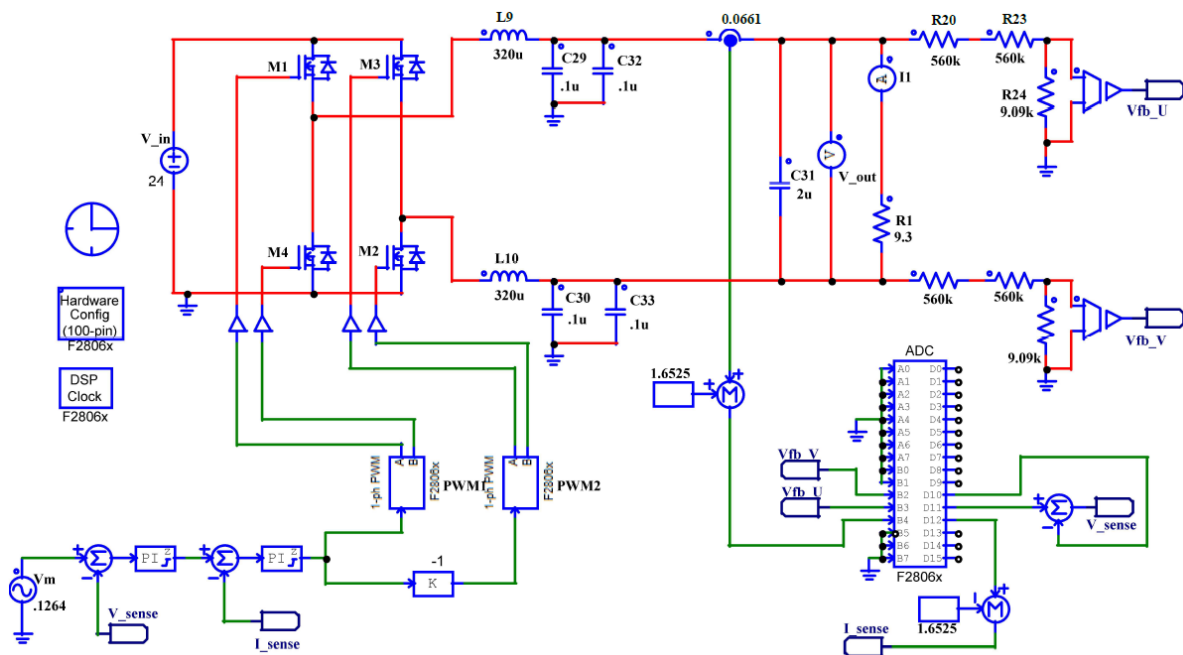


Figure 5.14: The completed circuit for generating the code for closed loop control

With this circuit built, I generated the C-code for implementing this control scheme on the Transphorm inverter. Using TI's Code Composer Studio, the code was loaded onto the TI microcontroller which was then placed in the inverter board. This code is given in Appendix A.

CHAPTER 6: RESULTS

After the control scheme was finished, I tested the inverter by measuring its voltage and current output for different loads. I used several pieces of equipment for this. First was an Agilent U8002A DC power supply. This power supply could output from 0-30 V with up to 5 A of current. To view the waveforms I used an Agilent MSO7034B oscilloscope. I used a P6100 100MHz oscilloscope probe to measure voltage and an Aim Tti I-prober 520 to measure the current. Additionally, I used a Cenco rheostat as a variable load. This rheostat was rated for 6.2 A and adjusted up to 11 Ω of resistance. Also used was a FLUKE 337 true RMS clamp meter.

6.1 SIMULATION RESULTS

Figure 6.1 shows the simulated output of the inverter with the combined inner current and outer voltage feedback loops closed. It is a stable 60Hz sine wave. The peak output voltage (red) measures 14.69 V while the peak output current (blue) measures 1.58 A . There is no perceptible phase shift between the voltage and the current.

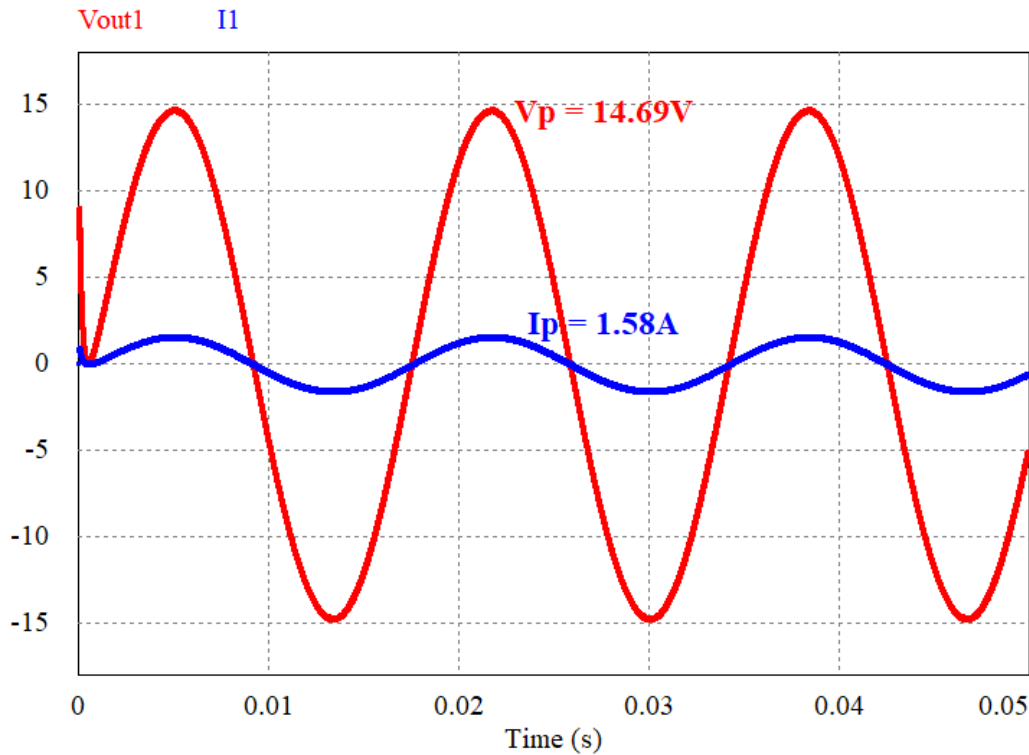


Figure 6.1: The simulated output voltage and current with combined feedback loops closed

Additionally, I added a load with a 10 mH inductor in series with a $9.3\ \Omega$ resistor to the simulation. After 23.6 ms , the load in the simulation switched to this load and the purely resistive load was switched out. This was done to simulate the start up behavior of a motor load on the control scheme. The voltage and current waveforms for this are shown in Figure 6.2.

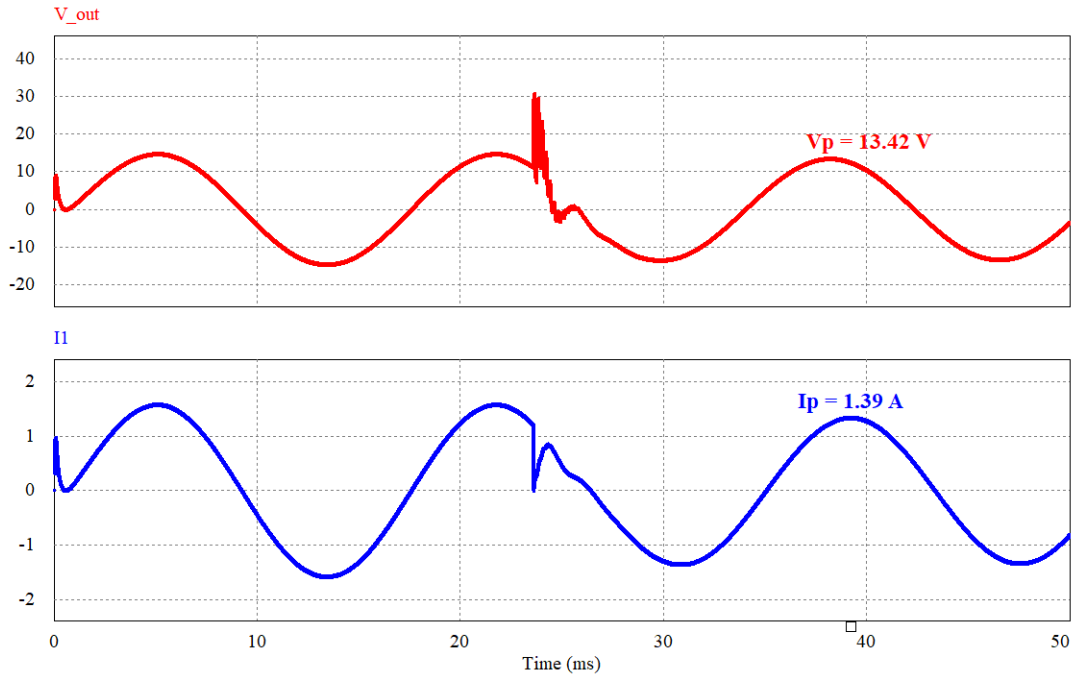


Figure 6.2: The simulated output voltage and current with the switched load behavior

As shown, the switched in load introduces some undesirable voltage spikes but is resolved within a quarter cycle. Afterwards, the peak output voltage drops slightly to 13.42 V and the peak current drops to 1.39 A. The measured phase shift is 23° . This load was unable to be tested in hardware.

6.2 HARDWARE RESULTS

Figure 6.3 shows the hardware testing setup used. Due to the COVID-19 pandemic, this setup was located at my residence.

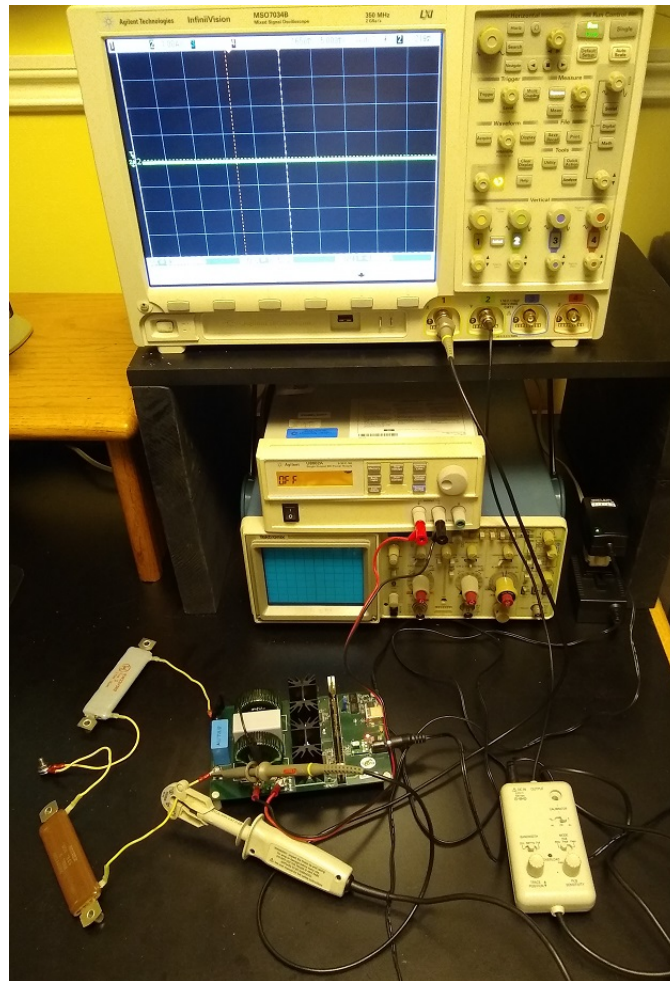


Figure 6.3: The hardware testing setup

With a 24 V input, the peak voltage across the load was measured at 16.9 V and the peak current was measured at 1.49 A. The frequency of both waveforms measured 60.2 Hz. The oscilloscope's on-board measurement function was used to acquire these values. Figure 6.4 shows these waveforms. The output voltage is scaled to 5 Volts/division while the output current is scaled to 1 Ampere/division.

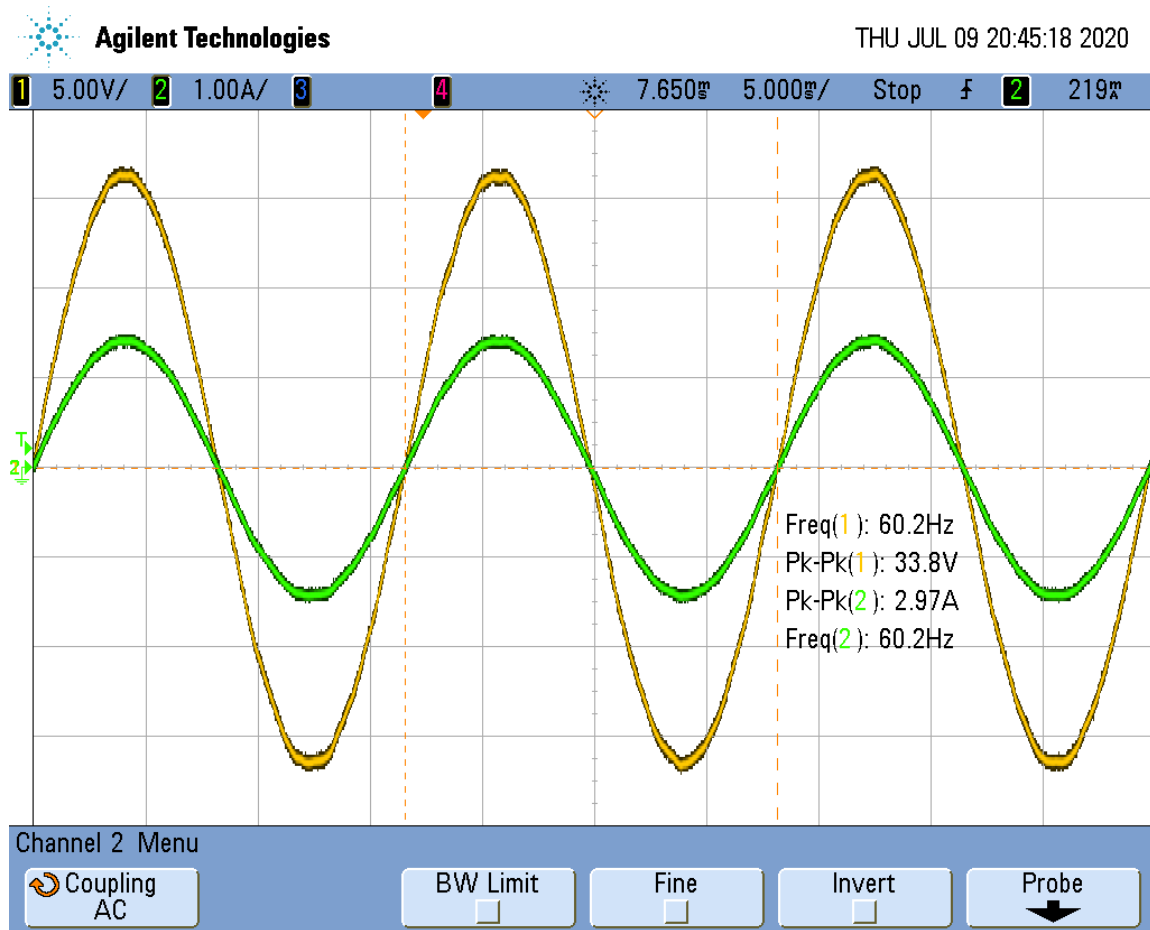


Figure 6.4: The output voltage waveform (yellow) and output current waveform (green) with the completed feedback loops and a 9.3Ω resistive load

Next, I measured the output of the inverter as it was connected to a varying load. This load was a Cenco rheostat that was adjustable from 0-11 Ω . Additionally, this rheostat was wire-wound around an air core and, as such, had a small amount of inductance. The output was measured at load values of 11 Ω , 6 Ω , and 4 Ω . Figures 6.5, 6.6, and 6.7 show these output waveforms, respectively.

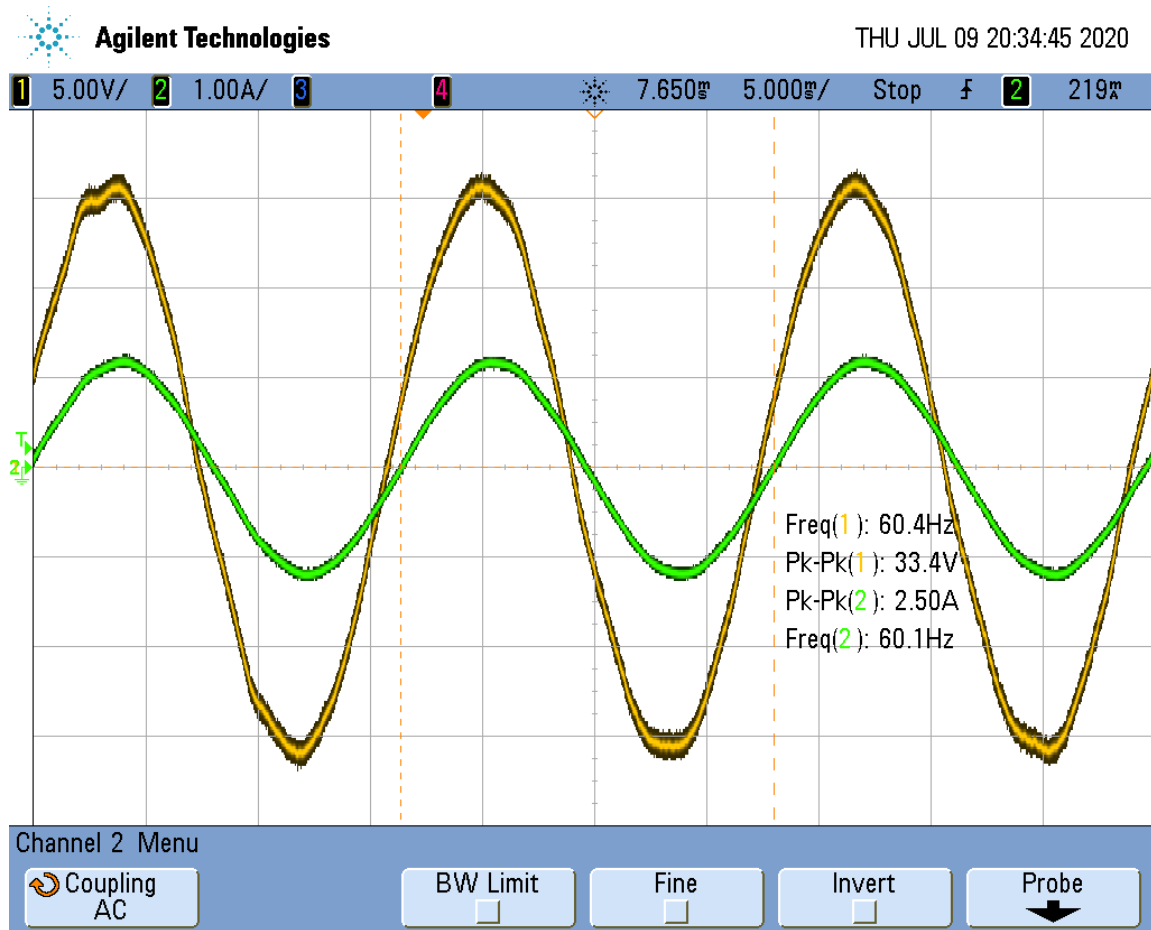


Figure 6.5: The output voltage waveform (yellow) and output current waveform (green) with an $11\ \Omega$ resistive and inductive load

At $11\ \Omega$ the peak output voltage measures $16.7\ V$ and the peak output current measures $1.25\ A$.

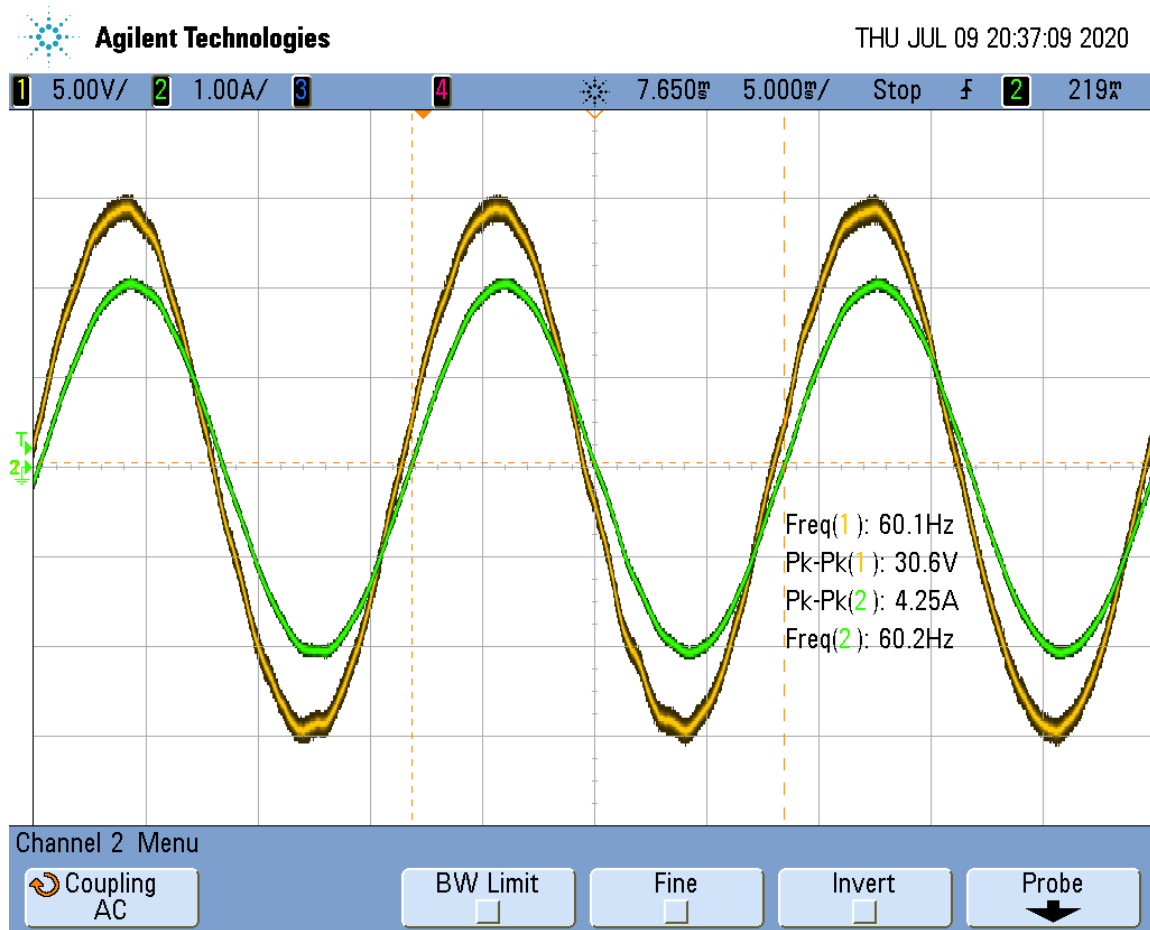


Figure 6.6: The output voltage waveform (yellow) and output current waveform (green) with a $6\ \Omega$ resistive and inductive load

At $6\ \Omega$ the peak output voltage measures $15.3\ V$ and the peak output current measures $2.13\ A$.

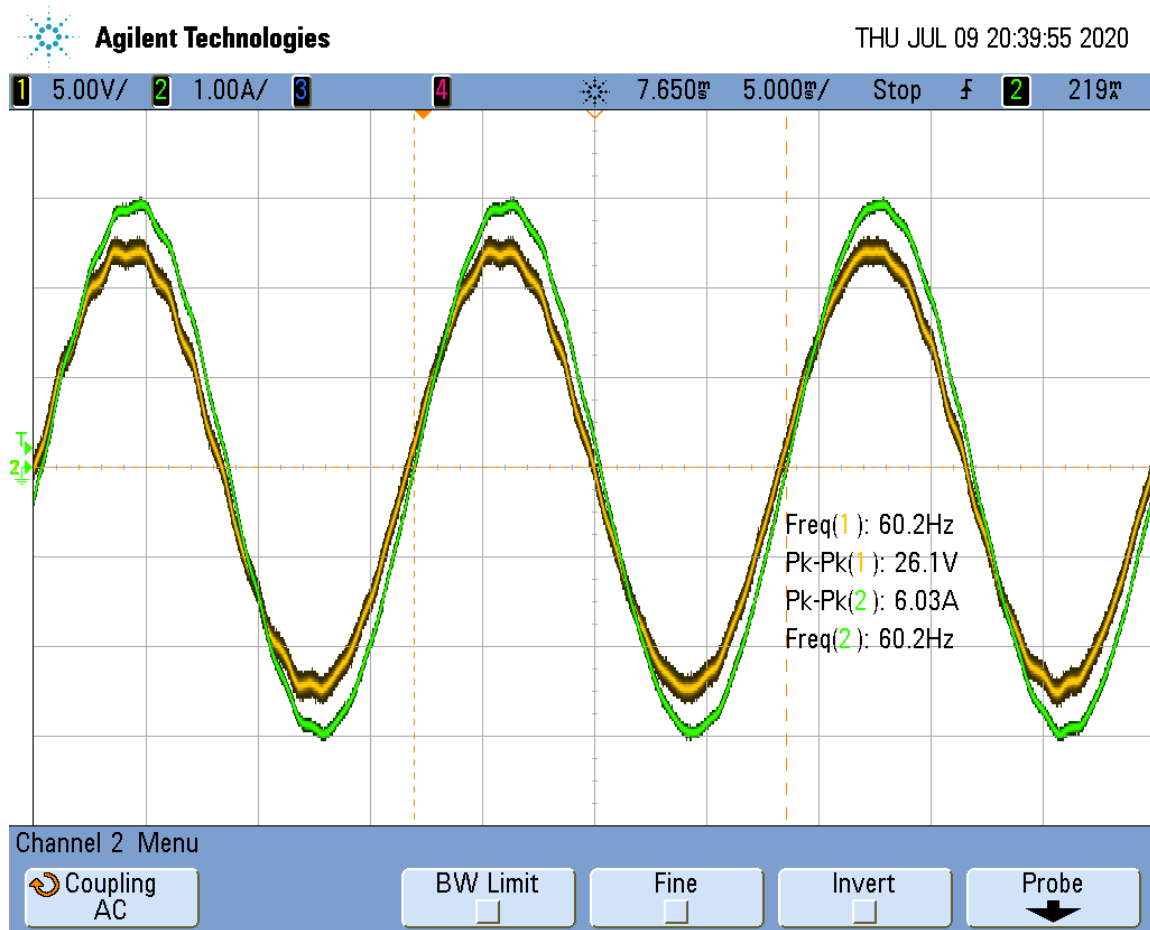


Figure 6.7: The output voltage waveform (yellow) and output current waveform (green) with a $4\ \Omega$ resistive and inductive load

At $4\ \Omega$ the peak output voltage measures 13.05V and the peak output current measures 3.02 A .

The hardware results of the $9.3\ \Omega$ load that was modelled in PSIM[®] are very close to the simulation results. Overall, this control scheme performs satisfactorily in regulating the output voltage and frequency of the inverter.

CHAPTER 7: CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

The design goal of a 60 Hz controlled output voltage on a high-frequency GaN-based inverter was successfully implemented using a control scheme designed and verified with PSIM[®]. As this thesis has shown, designing a control scheme for a GaN-based circuit using PSIM[®] is straightforward and simple. The control design process for a GaN-based inverter follows the same principles as other systems. As long as the individual components and layout are properly designed, classical feedback and control methods can be used to develop a control scheme for a GaN-based circuit without additional complications. PSIM[®] has proven to be an effective tool for rapidly developing and implementing a control scheme on a TI microcontroller and will be useful for the remainder of the project.

7.2 FUTURE WORK

As this was a small step toward the completion of the larger project, there is more work to do. PSIM[®] can be used in several ways to help further this project.

Firstly, we need to refine the simulation model to more accurately mimic the real-world hardware. Transphorm offers Spice models of their transistors. Spice is an open-source circuit simulation software. With PSIM[®] Version 12 and later, these Spice models can be coupled with PSIM[®] so that these can replace the ideal switches used in this initial model. With these models, a more accurate frequency response is obtained and thus a more accurate control scheme can be designed.

Secondly, we can offer additional over-current and over-voltage protection for the inverter by defining those conditions in the simulation circuit and setting up the logic to turn off the inverter when needed. PSIM[®] has this ability.

Thirdly, the inverter hardware needs to be scaled up to the original specifications of 6

kW . At the same time, design specifications regarding the controlled output need to be refined for optimal operation of motor loads.

Lastly, as the inverter reaches its final design, PSIM[®] can generate the code needed to integrate many of the desired peripherals by adding those blocks to the simulation.

REFERENCES

- [1] *Deposit For Flex 1500 Power Station*, <https://inergytek.com/collections/solar-generators/>, Accessed: 07-13-2020.
- [2] *Scope of Work*, Internal Communication, Jul. 2016.
- [3] A. Lidow, J. Strydom, M. de Rooij, and D. Reusch, *GaN Transistors for Efficient Power Conversion*, 2nd ed. John Wiley & Sons, Ltd., 2015.
- [4] *Silicon vs. Gallium Nitride (GaN): Properties and Applications in Semiconductors*, <https://www.arrow.com/en/research-and-events/articles/gan-vs-silicon-semiconductor-materials-compared>, Accessed: 07-23-2020.
- [5] A. Bindra, “Tapping the Benefits of Monolithic Gallium Nitride Power Integrated Circuits,” *IEEE Power Electronics Magazine*, vol. 3, no. 3, pp. 4–6, Sep. 2016. DOI: 10.1109/MPEL.2016.2585473.
- [6] A. Bindra, “Wide-Bandgap Power Devices: Adoption Gathers Momentum,” *IEEE Power Electronics Magazine*, vol. 5, no. 1, pp. 22–27, Mar. 2018. DOI: 10.1109/MPEL.2017.2782404.
- [7] K. A. Kim, Y.-C. Liu, M.-C. Chen, and H.-J. Chiu, “Opening the Box: Survey of High Power Density Inverter Techniques From the Little Box Challenge,” *CPSS Transactions on Power Electronics and Applications*, vol. 2, no. 2, pp. 131–139, Jun. 2017. DOI: 10.24295/CPSSTPEA.2017.00013.
- [8] N. Mohan, T. M. Undeland, and W. P. Robbins, *Power Electronics*, 3rd ed. John Wiley & Sons, Inc., 2003.
- [9] D. W. Hart, *Power Electronics*. McGraw Hill Education, 2011.
- [10] M. H. Rashid, *Power Electronics: Devices, Circuits, and Applications*, 4th ed. Pearson Education, Inc., 2014.

- [11] A. L. Pressman, *Switching Power Supply Design*, 2nd ed. McGraw-Hill, 1998.
- [12] S. Buso and P. Mattavelli, *Digital Control in Power Electronics*, 1st ed. Morgan & Claypool, 2006.
- [13] *Inverter Design with Average Current and Voltage Loop Control*, https://youtu.be/25c8_trrmSo, Accessed: 09-15-2017.
- [14] “Designing Hard-Switched Bridges with GaN,” Transphorm Inc., Tech. Rep., Sep. 2014.
- [15] M. H. Rashid, “Power Electronics: Devices, Circuits, and Applications,” in, 4th ed. Pearson Education, Inc., 2014, ch. 5, p. 216.
- [16] “TDPV1000E0C1 Single-Phase Inverter Evaluation Board,” Transphorm Inc., Tech. Rep., Dec. 2014.
- [17] *TMS320F28069 Piccolo Microcontroller*, www.ti.com/product/TMS320F28069, Accessed: 04-07-2018.
- [18] *TPH3206PS*, tph3206ps.12, Transphorm Inc., Jan. 2017.
- [19] *PSIM User’s Guide*, Version 11, Powersim Inc., 2017.
- [20] J. Chiasson, *An Introduction to System Modeling and Control*, 1st ed. John Chiasson, 2017.
- [21] *SmartCtrl User’s Guide*, Version 3.0, Carlos III University, Mar. 2016.

APPENDIX A: INVERTER CONTROL CODE

```

/*****
// This code is created by SimCoder Version 11.1.3.2 for F2806x Hardware
//Target
// SimCoder is copyright by Powersim Inc., 2009-2018
// Date: April 30, 2020 13:22:29
// Created by Ryan Ready
*****/

#define GLOBAL_Q 20
long GlobalQ = GLOBAL_Q;      // Used for legacy GEL & Graph Debug.
#include "IQmathLib.h"
#include "PS_bios.h"
#define GetCurTime() PS_GetSysTimer()
#define PWM_IN_CHECK // To lower PWM value setting time, comment out
this line if PWM duty cycle values are strictly limited in the range.

interrupt void Task();

const Uint16 PSD_CpuClock = 90; // MHz
extern _iq fGblVref;
extern _iq fGblVk;
extern _iq fGblVm;
extern _iq28 fGblV_v;
extern _iq28 fGblV_u;

```

```
_iq fGblVref = 0;
_iq fGblVk = 0;
_iq fGblVm = 0;
_iq28 fGblV_v = 0;
_iq28 fGblV_u = 0;

interrupt void Task()
{
_iq20 fP1, fS5, fSUM1, fSUM4, fS6, fSUM5, fSUM2, fRef0;
_iq28 fADC1_12, fADC1_10, fADC1_11;
_iq30 fC2;

{

static _iq29 wt = _IQ29(0 / 360.);
static _iq29 dwt = _IQ29((60) * 1.0 / 100000L);
fRef0 = _IQ29sinPU(wt);
wt += dwt;
if (wt >= _IQ29(1.0)) wt -= _IQ29(1.0);
fRef0 = _IQ20mpyIQX(fRef0, 29, _IQ30(.12648), 30);
}

#ifdef _DEBUG
fGblVref = fRef0;
#endif
```

```

fADC1_11 = PS_GetAcAdc(11);
fADC1_10 = PS_GetAcAdc(10);
fSUM2 = ((fADC1_11) >> 8) - ((fADC1_10) >> 8);
fSUM5 = fRef0 - fSUM2;

{ // backward Euler
static _iq20 out_A = 0;
fS6 = out_A + _IQ20mpyIQX(_IQ30((157.203E-3)/((173.178E-6)*100000L)),
30, fSUM5, 20);
fS6 = (fS6 < _IQ20((-0.99))) ? _IQ20((-0.99)) : ((fS6 > _IQ20(0.99))
? _IQ20(0.99) : fS6);
out_A = fS6;
fS6 += _IQ20mpyIQX(_IQ30((157.203E-3)), 30, fSUM5, 20);
fS6 = (fS6 < _IQ20((-0.99))) ? _IQ20((-0.99)) : ((fS6 > _IQ20(0.99))
? _IQ20(0.99) : fS6);

}

fADC1_12 = PS_GetDcAdc(12);
fC2 = _IQ30(1.6525);
fSUM4 = ((fADC1_12) >> 8) - ((fC2) >> 10);
fSUM1 = fS6 - fSUM4;
{ // backward Euler
static _iq20 out_A = 0;
fS5 = out_A + _IQ20mpyIQX(_IQ30(3.495/((78.9271E-6)*100000L)), 30,
fSUM1, 20);
fS5 = (fS5 < _IQ20((-0.98))) ? _IQ20((-0.98)) : ((fS5 > _IQ20(0.98))
? _IQ20(0.98) : fS5);

```

```
out_A = fS5;
fS5 += _IQ20mpyIQX(_IQ29(3.495), 29, fSUM1, 20);
fS5 = (fS5 < _IQ20((-0.98))) ? _IQ20((-0.98)) : ((fS5 > _IQ20(0.98))
? _IQ20(0.98) : fS5);
}

#ifdef _DEBUG
fGblVk = fS5;
#endif

#ifdef _DEBUG
fGblVm = fS5;
#endif

fP1 = _IQ20mpyIQX(fS5, 20, _IQ30((-1.0)), 30); // fS5 * (-1.0)

#ifdef _DEBUG
fGblVv = fADC1_10;
#endif

#ifdef _DEBUG
fGblVu = fADC1_11;
#endif

// Start of changing PWM2(1ph) registers
// Set Duty Cycle
```

```

#ifdef PWM_IN_CHECK
if (fP1 <= _IQ20((-1.0))) {
PWM_CMPA(2) = 0;
} else if (fP1 >= _IQ20(2 + (-1.0))) {
PWM_CMPA(2) = PWM_TBPRD(2);
} else {
#else // PWM_IN_CHECK
{

#endif
_iq20 _val = fP1;
PWM_CMPA(2) = _IQ1mpyIQX(PWM_TBPRD(2), 1, _val - _IQ20((-1.0)), 20-(1));
}

// End of changing PWM2(1ph) registers
// Start of changing PWM1(1ph) registers
// Set Duty Cycle

#ifdef PWM_IN_CHECK
if (fS5 <= _IQ20((-1.0))) {
PWM_CMPA(1) = 0;
} else if (fS5 >= _IQ20(2 + (-1.0))) {
PWM_CMPA(1) = PWM_TBPRD(1);
} else {
#else // PWM_IN_CHECK
{
#endif
#endif

```

```

_iq20 _val = fS5;
PWM_CMPA(1) = _IQ1mpyIQX(PWM_TBPRD(1), 1, _val - _IQ20((-1.0)), 20-(1));
}
// End of changing PWM1(1ph) registers
PS_ExitAdcIntr(1, M__INT1);
}

void Initialize(void)
{
PS_SysInit(0, 10, 18);
PS_StartStopPwmClock(0); // Stop Pwm Clock
PS_InitTimer(0, 0);
PS_AdcInit();
{
    int i;
    /* TAdcAttr: Channel No., Soc No., Trig Src, INTADC#,
    Window Size, Gain */
    const TAdcAttr aryAdcInit[3] = {{12, 0, ADCTRIG_PWM1, 1, 6, _IQ28(1.0)},
{10, 1, ADCTRIG_PWM1, 1, 6, _IQ28(1.0)},
{11, 2, ADCTRIG_PWM1, 1, 6, _IQ28(1.0)}};
    const TAdcAttr *p = aryAdcInit;
    for (i = 0; i < 3; i++, p++) {
        PS_SetAdcChn(p->nIntrNo, p->nChnNo,
        p->nSocNo, p->nTrigSrc, p->nWindSz, p->nGain);
    }
}
}

```



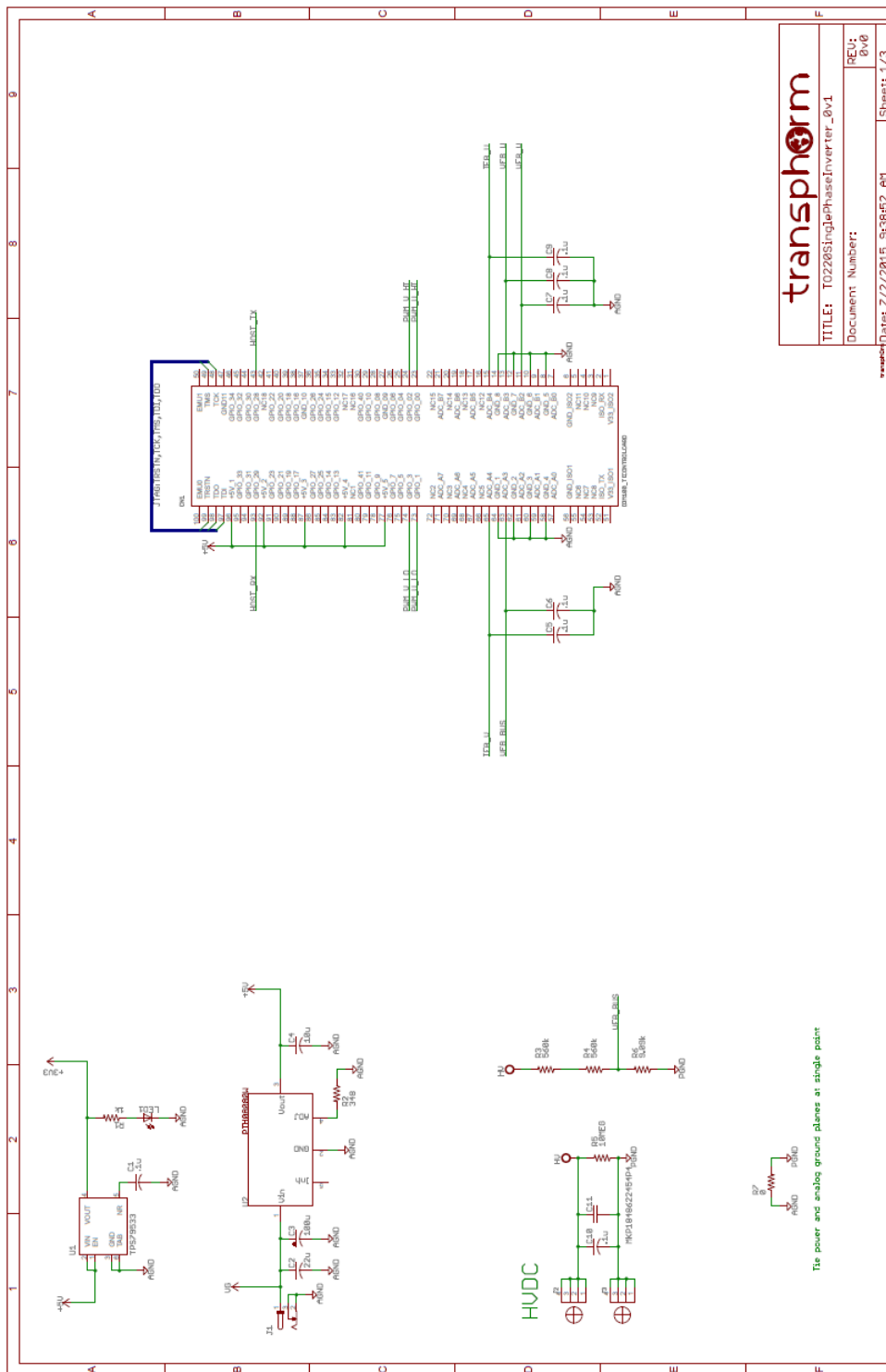
```

PS_InitPwm(1, 0, 1, _IQ8(1.0e6/((double)100000*1)),
_IQ24((50E-9) * 1.0e6), PWM_TWO_OUT, HRPWM_DISABLE);
// pwnNo, waveType, frequency, deadtime, outtype, UseHRPwm
PS_SetPwmIntrType(1, ePwmIntrAdc, 1, _IQ24(0));
PS_SetPwmVector(1, ePwmIntrAdc, 2, 0, Task);
PS_SetPwmTzAct(1, eTZHighImpedance);
PS_SetPwmRateSH(1, _IQ24((0 - (-(1.0))) * 1.0 / 2));
PS_StartPwm(1);
PS_InitPwm(2, 0, 1, _IQ8(1.0e6/((double)100000*1)),
_IQ24((50E-9) * 1.0e6), PWM_TWO_OUT, HRPWM_DISABLE);
// pwnNo, waveType, frequency, deadtime, outtype, UseHRPwm
PS_InitPwmPhase(2, _IQ24(0 * (1.0 / 360.0)));
PS_SetPwmIntrType(2, ePwmNoAdc, 1, _IQ24(0));
PS_SetPwmTzAct(2, eTZHighImpedance);
PS_SetPwmRateSH(2, _IQ24((0 - (-(1.0))) * 1.0 / 2));
PS_StartPwm(2);
PS_StartStopPwmClock(1); // Start Pwm Clock
}

void main()
{
Initialize();
PS_EnableIntr(); // Enable Global interrupt INTM
PS_EnableDbgm();
for (;;) {
}
}

```

APPENDIX B: HARDWARE SCHEMATICS

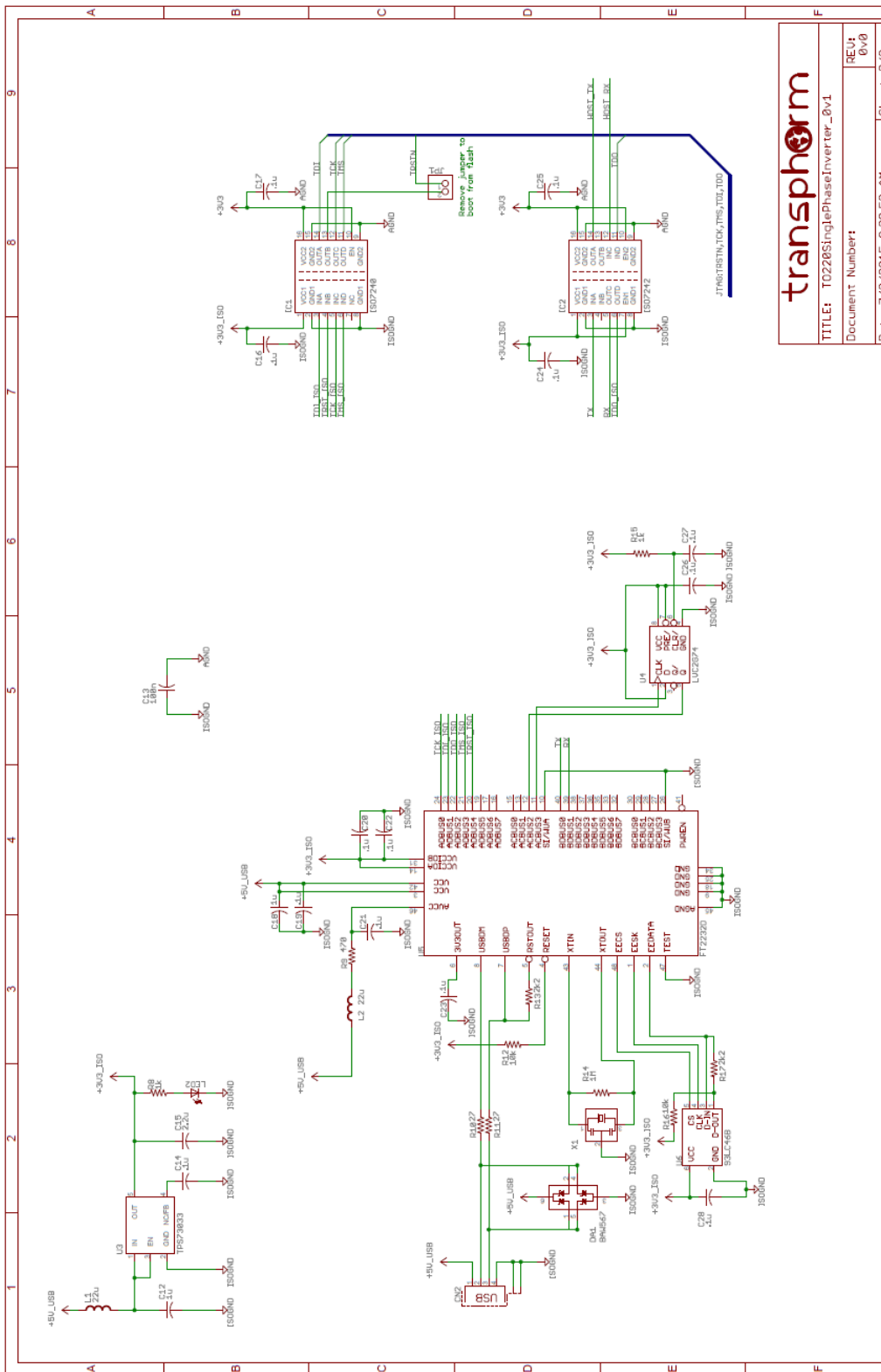


transphorm

TITLE: T0220SinglePhaseInverter_0v1

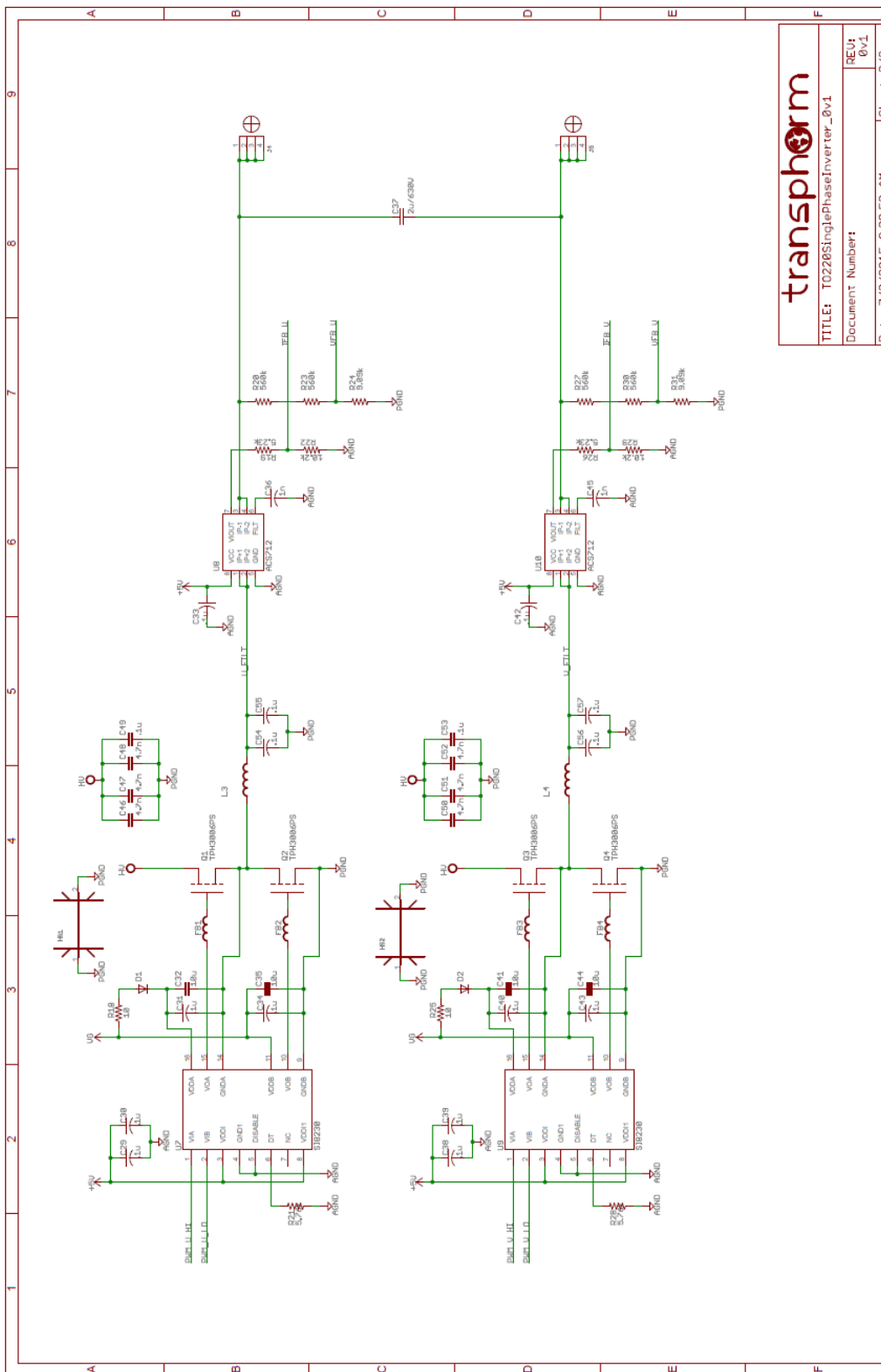
Document Number: REV: 0v6

Date: 7/2/2015 S:3852 P:1 Sheet: 1/3



transphorm

TITLE: T0220SinglePhaseInverter_0v1
 Document Number:
 Date: 7/2/2015 9:38:52 AM
 Sheet: 2/3



transphorm

TITLE: T0220SinglePhaseInverter_0v1
Document Number:
Date: 7/2/2015 9:38:52 AM
Sheet: 3/3