Florida State University Libraries

Electronic Theses, Treatises and Dissertations The Graduate School

2012

Real-Time High Speed Generator System Emulation with Hardware-in the-Loop Application

Nicholas Stroupe

Follow this and additional works at the FSU Digital Library. For more information, please contact lib-ir@fsu.edu

THE FLORIDA STATE UNIVERSITY

COLLEGE OF ENGINEERING

REAL-TIME HIGH SPEED GENERATOR SYSTEM EMULATION WITH HARDWARE-IN-THE-LOOP APPLICATION

By

NICHOLAS STROUPE

A Thesis submitted to the
Department of Electrical and Computer Engineering
in partial fulfillment of the
requirements for the degree of
Master of Science

Degree Awarded:
Summer Semester, 2012

Nicholas Stroupe defended this thesis on June 22, 2012.

The members of the supervisory committee were:

Chris S. Edrington
Professor Directing Thesis

Simon Y. Foo
Committee Member

Helen Li
Committee Member

The Graduate School has verified and approved the above-named committee members, and certifies that the thesis has been approved in accordance with university requirements.

This work is dedicated in loving memory to my mother, Kathy Stroupe, whose love and unrelenting support throughout my life has made all of this possible.

ACKNOWLEDGEMENTS

I would like to thank my academic advisor, Dr. Chris S. Edrington, for giving me the opportunity to succeed in the graduate academic environment. I'd like to acknowledge his hard work in providing many students such as myself within the Energy Conversion and Integration Thrust at the Center for Advanced Power Systems an opportunity for continued education through funding, guidance, and encouragement of publication and conference attendance. I'd also like to acknowledge Dr. Edrington for his forward thinking and strong abilities to match research interests with applicable thesis topics.

I want to thank Dr. Amin Hasanzadeh for his support and mentorship during my graduate studies. Dr. Hasanzadeh has not only offered technical support but also personal advice that I will take on with me in my career and future endeavors. Through his steadfast and honest approach, he has shown to me that I am more capable than I believe. I want to thank my colleagues in the Energy Conversion and Integration Thrust for the friendship, support, and shared stories of agony in times of frustration.

I want to give a very special thanks to my mother, Kathy Stroupe, whose year-long battle with cancer ended peacefully this past January during the final months of my thesis preparation. Without her guidance, love, and unwavering support in anything I've ever been involved in, this achievement would not be possible. Most importantly, I'd like to thank her for teaching me to never give up by both encouragement and through example. There is no one who would love to hear about this accomplishment more than her and I can't thank her enough for giving me the confidence and opportunity to achieve my goals. Lastly, I would like to thank my wonderful girlfriend, Erin Cantwell, who has stood by me in this difficult time and inspired me to keep my head up and to always see the brighter side of things.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AC	:	Alternating Current
ADC	:	Analog to Digital Converter
AES	:	All-Electric-Ship
CAPS	:	Center for Advanced Power Systems
CHIL	:	Controller Hardware-in-the-Loop
DAC	:	Digital to Analog Converter
DC	:	Direct Current
DSC	:	Digital Signal Controller
EMF	:	Electromotive Force
FPGA	:	Field Programmable Gate Array
HIL	:	Hardware-in-the-Loop
IGBT	:	Insulated Gate Bi-polar Junction Transistor
kHz	:	Kilohertz
kW	:	Kilowatt
LE	:	Logic Element
MHz	:	Megahertz
MIL	:	Model-in-the-Loop
MTG	:	Micro Turbine Generator
MVDC	:	Medium Voltage Direct Current
MW	:	Megawatt
NLDL	:	Non-Linear Dynamic Loads
Nm	:	Newton Meter
PHIL	:	Power Hardware-in-the-Loop
PMG	:	Permanent Magnet Generator
P.U.	:	Per Unit
PWM	:	Pulse Width Modulation
RFT	:	Reference Frame Theory
RK4	:	Runge Kutta 4^{th} Order

RPM	:	Revolution Per Minute
SPI	:	Serial Peripheral Interface
TI	:	Texas Instruments

ABSTRACT

The emerging emphasis and benefits of distributed generation on smaller scale networks has prompted much attention and focus to research in this field. Much of the research that has grown in distributed generation has also stimulated the development of simulation software and techniques. Testing and verification of these distributed power networks is a complex task and real hardware testing is often desired. This is where simulation methods such as hardware-in-the-loop become important in which an actual hardware unit can be interfaced with a software simulated environment to verify proper functionality. In this thesis, a simulation technique is taken one step further by utilizing a hardware-in-the-loop technique to emulate the output voltage of a generator system interfaced to a scaled hardware distributed power system for testing.

The purpose of this thesis is to demonstrate a new method of testing a virtually simulated generation system supplying a scaled distributed power system in hardware. This task is performed by using the Non-Linear Loads Test Bed developed by the Energy Conversion and Integration Thrust at the Center for Advanced Power Systems. This test bed consists of a series of real hardware developed converters consistent with the Navy's All-Electric-Ship proposed power system to perform various tests on controls and stability under the expected non-linear load environment of the Navy weaponry. This test bed can also explore other distributed power system research topics and serves as a flexible hardware unit for a variety of tests. In this thesis, the test bed will be utilized to perform and validate this newly developed method of generator system emulation.

In this thesis, the dynamics of a high speed permanent magnet generator directly coupled with a micro turbine are virtually simulated on an FPGA in real-time. The calculated output stator voltage will then serve as a reference for a controllable three phase inverter at the input of the test bed that will emulate and reproduce these voltages on real hardware. The output of the inverter is then connected with the rest of the test bed and can consist of a variety of distributed system topologies for many testing scenarios. The idea is that the distributed power system under test in hardware can also integrate real generator system dynamics without physically involving an actual generator system. The benefits of successful generator system emulation are vast and lead to much more detailed system studies without the draw backs of needing physical

generator units. Some of these advantages are safety, reduced costs, and the ability of scaling while still preserving the appropriate system dynamics.

This thesis will introduce the ideas behind generator emulation and explain the process and necessary steps to obtaining such an objective. It will also demonstrate real results and verification of numerical values in real-time. The final goal of this thesis is to introduce this new idea and show that it is in fact obtainable and can prove to be a highly useful tool in the simulation and verification of distributed power systems.

CHAPTER ONE

INTRODUCTION, MOTIVATION AND ORGANIZATION

1.1 Introduction

Distributed generation is becoming increasingly popular due to the emerging emphasis of renewable energy source integration and many other inherent security and efficiency benefits. The incorporation of distributed generation in today's energy infrastructure often needs advanced use of power electronic devices and control methodologies and is currently a very active branch of research. In aid to this research effort, many different types of studies are involved ranging from system modeling in software to small scale system development on actual hardware for testing purposes. A combination of simulation and hardware testing can be utilized to study the effects of a generator system and is the objective of this thesis. This form of study is referred to as generator system emulation and is a very new and developing form of study. This involves hardware, software and hardware-in-the-loop (HIL) integration to emulate the generator system of a distributed power system in hardware for in depth analysis and system studies. In fact, very little evidence in research literature today exists in which this procedure is demonstrated and documented. It is the goal of this thesis to introduce and explore such a procedure and prove that is a viable option for distributed generation simulation and research.

Generator system emulation is a practice in which a generator system consisting of a prime mover and electric generator is virtually simulated in real-time and emulated on real hardware via a controllable voltage/current source. Generator emulation is a new concept in simulation/verification with much room for research and development for various applications. The implementation of such a procedure coupled with a real distributed system introduces the possibility of a wide range of system studies. Using such a practice eliminates the need of physically involving the generator system in testing or development of control strategies which brings about many advantages and possibilities. The design, implementation and results of the system designed to achieve generator emulation is described and analyzed in the work of this thesis. This thesis demonstrates the system development, implementation, verification and

analysis of results of generator system emulation in hardware and compares it with the expected results performed in software simulation.

1.2 Motivation and Contributions

This thesis aims to aid in the research of the Navy's all-electric-ship (AES) medium voltage direct current (MVDC) system [1]. The MVDC system carries the same characteristics of a distributed generation system in that it is small scale and not directly connected to a large grid system. The MVDC also uses many power electronic converters and is subject to large load pulses that need to be analyzed and studied in the most realistic environment possible. By using generation emulation, the effects of the non-linear loads and the switching of the power converters can be readily studied on the specific generator, as well as the system as a whole. Using generator emulation for studies on the MVDC system leads to a more realistic approach to investigation of the MVDC system as well as allows for new control strategies to be readily tested in a much more practical atmosphere involving a real system in hardware.

The focus of the research done in this thesis is to develop and introduce a method of electric generator emulation via two coupled 3-phase bridge converters. The goal is to calculate the dynamics of a coupled gas turbine and electric generator system and direct the output of a three phase inverter to emulate the calculated 3-phase alternating current (AC) voltages. The basics of the experimental setup and objectives are shown in Figure 1.1. Figure 1.1 depicts how the coupled 3-phase bridge converter set can be theoretically replaced by a non-existent generator system. The non-existent generator system is shown above with the dotted lines, the real hardware is shown below. The idea of generator emulation is that the inverter shown in Figure 1.1 will be controlled to respond identically to the non-existent generator system in terms of voltage amplitude and frequency. This means that the load/converter will be electrically excited as if it is connected to the stator terminals of a certain generator system while actually connected to a three phase inverter. The purpose of the rectifier is to convert the incoming three phase AC from the grid to direct current (DC). This DC power is then available to the inverter to create an AC voltage at a desired amplitude and frequency.

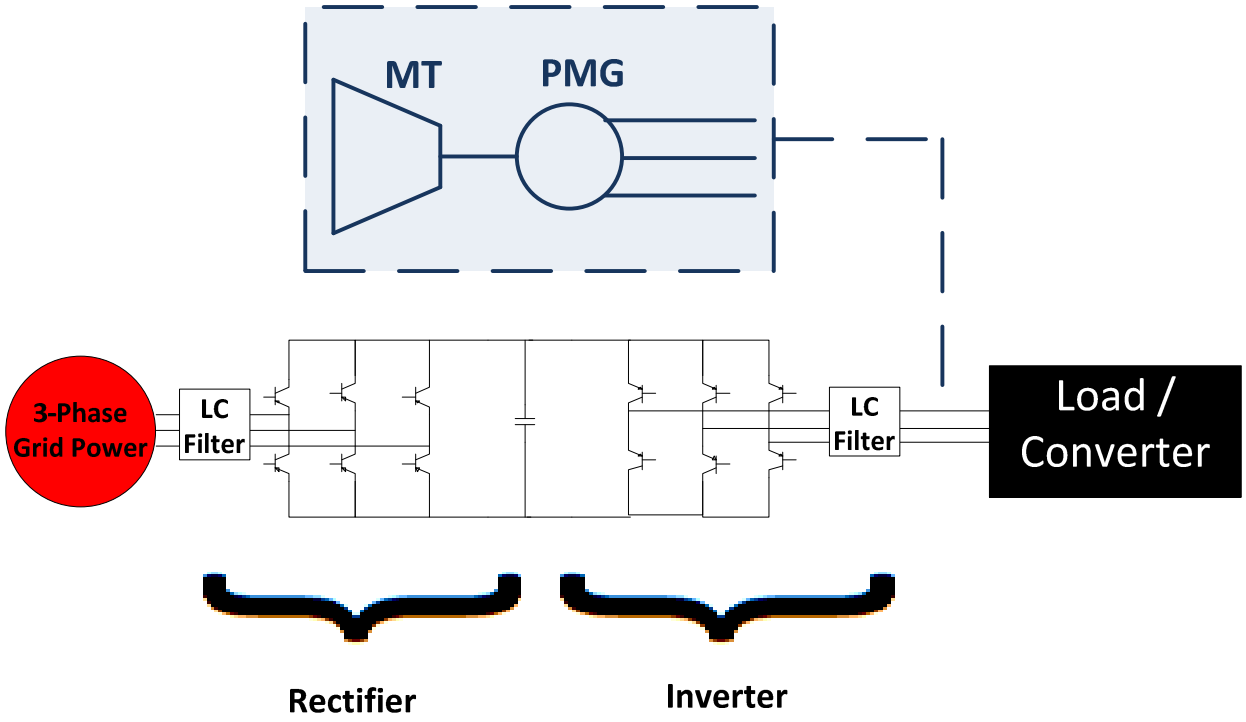


Figure 1.1: Generator Emulation Realization.

As previously stated, the emulation of the generator system will involve the use of HIL implementation. HIL refers to a form of simulation in which a particular hardware component interacts with a simulated environment or a mathematical model. Advanced simulation involves the use of real-time models and HIL interface to take validation efforts one step further which introduces many new advantages [2]. HIL can exist in a number of forms but most popularly including power HIL (PHIL) or controller HIL (CHIL). PHIL involves the use of incorporating physical power hardware to a simulation while CHIL refers to physically involving a controller with a simulated environment. Both forms of HIL allow for more in depth validation techniques by actually involving the real hardware in simulation. The form of HIL that most closely matches what is performed in this work has been previously referred to as model-in-the-loop (MIL) [3]. MIL refers to a form of HIL in which a simulated model that is the device under test is introduced to an actual hardware environment. This type of procedure is performed since physically involving the generation system would be a large constraint to costs and safety during the testing phase.

The benefits of generator system emulation via MIL allows for detailed investigation of control and system responses to various loading conditions on real hardware without the

drawbacks of an actual generator system such as costs, space, configurability and safety [3]. Generator emulation requires the dynamics of the system be computed accurately and in real-time. Also, because the generator will react in accordance to system behavior, such as load changes, the model needs access to system measurements of voltage and current as fast as possible and preferably within the same computational time step. Therefore, a digital system of this complexity requires the use of a field programmable gate array (FPGA) with multiple analog-to-digital converters (ADC) and at least six pulse width modulation (PWM) pulses to the inverter's insulated-gate bipolar junction transistors (IGBT).

In this thesis, a multi-platform computation system using an Altera DE2 development board EP2C35F672 FPGA, a TMS320F28335 Texas Instruments (TI) digital signal controller (DSC) and a dSPACE unit with a PC running dSPACE ControlDesk together is created for this emulation experiment. The FPGA is used for computation of the dynamic equations due to the inherent parallel architecture that can be exploited for rapid calculation of mathematically demanding systems [4]. The DSC is used for the analog readings of the system, PWM output to the inverter, and for future computation of control algorithms for the generator system. The dSPACE is used for real-time data acquisition and user interface on a PC terminal. The computational platform is shown in Figure 1.2. As can be seen, the computational platform will communicate via serial peripheral interface (SPI) and RS-232. The details and specifications of the computational platform will be further explained in Chapter 4.

This thesis will demonstrate the HIL capability using an inverter to output the calculated voltage references while the DSC will sense the activity of the distributed system in response to loading. For the purpose of demonstration and initial testing, the load on the output of the inverter will be a constant three phase resistor which represents a power factor corrected (PFC) converter for DC rectification. This resistance is considered unknown to the computation platform which must sense this value through current and voltage readings at every time step.

An objective of this research is to not only create and use a generator emulation platform, but to also explore the limitations and possibilities that exist. One objective is to investigate how accurately high speed generator systems can be emulated and to maximize the possibilities that FPGA parallel processing can provide. A popular high speed generator system is the micro

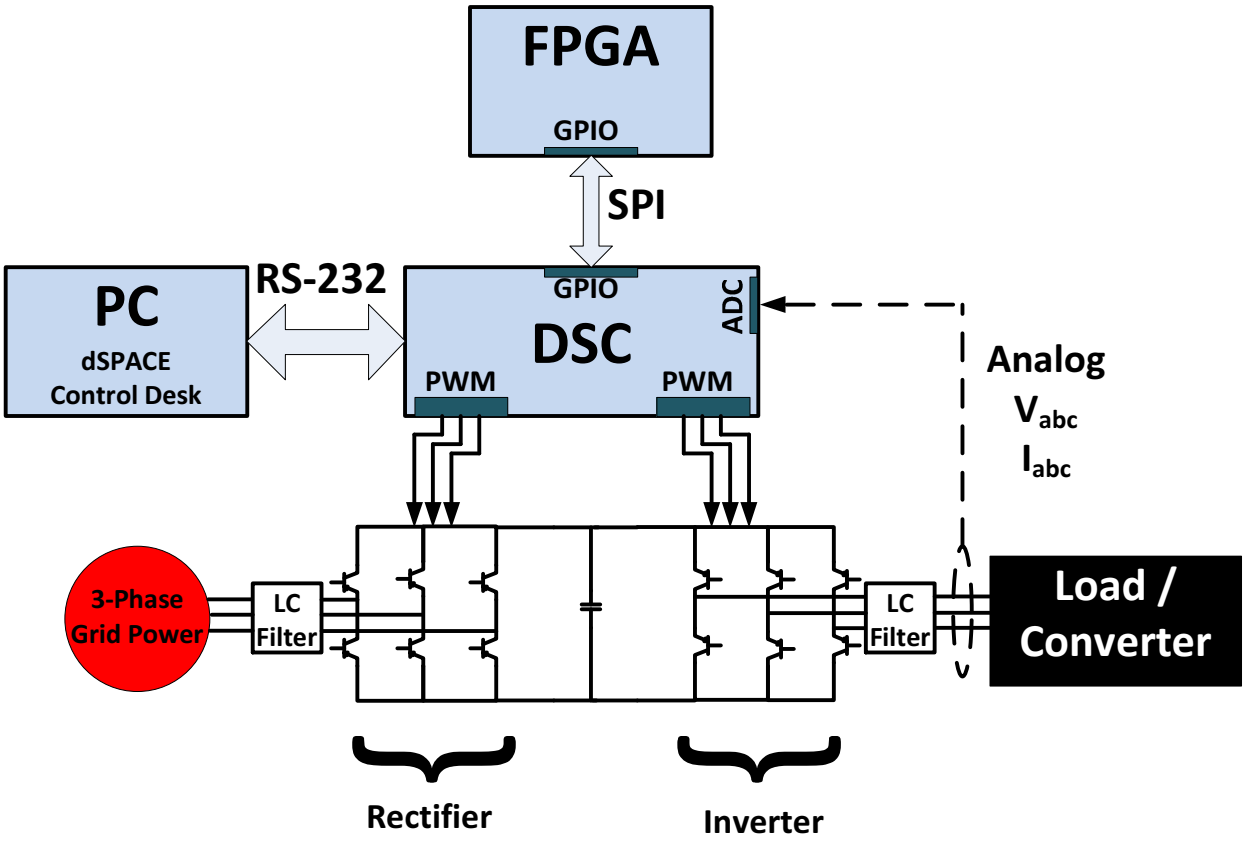


Figure 1.2: Multi-platform Computation System for Generator Emulation

turbine generator (MTG) system consisting of a lower rated high speed gas turbine coupled to an electric generator, most often the synchronous generator and more specifically, the permanent magnet generator (PMG).

The contributions of this work can be summarized as follows:

- Development of a computational platform capable of real-time dynamic calculations of a high speed generator system consisting of both the micro turbine as well as the PMG.
- Detailed description on the process in which an inverter can be controlled via PWM and how the system loads are measured and implemented into the generator dynamics calculation for HIL.
- Proof of concept by experimentation and validation of results.
- Study on the capabilities of the developed system with analysis on the limiting factors and options for further development.

1.3 Thesis Organization

The structure of this thesis is as follows. Chapter 2 will provide a review and description of the MTG system along with the mathematical models for the micro turbine and PMG as well as the full mathematical integration. Next in Chapter 3, the multi-platform real-time computation system that was developed for the emulation will be discussed. The numerical analysis and implementation of the dynamic equations into a digital environment with benefits and advantages of various ODE solving strategies is explained. The real-time simulation of the PMG and MTG system using the FPGA and the multi-platform system will be analyzed and verified. Next in Chapter 4, the HIL interface and experimental set up is presented with the introduction of the Non-Linear Loads Test Bed. In Chapter 5, the results and validation is shown and discussed. Finally, Chapter 6 will provide the conclusion of topics explored and discuss opportunities for further development and future studies.

CHAPTER TWO

MATHEMATICAL MODELS AND DERIVATIONS

In this chapter, the workings of the PMG as well as micro turbine will be presented along with the mathematical models and derivations. First, the permanent magnet machine overview and dynamic equations using the 2-axis *qd* equations will be presented. Next, a brief review of gas turbine and micro-turbine modeling methods will be presented along with a further detailed description of the specific model chosen for this emulation experiment.

2.1 Permanent Magnet Synchronous Generator

2.1.1 Structure and Dynamics Overview

The PMG is gaining popularity among smaller scale distributed generation systems, typically on the order of kilowatts (kW). The advantages of using the PMG over the traditionally used wound rotor synchronous machines are a higher efficiency and power density [5]. The increase in efficiency is due to the lack of a necessary field excitation since the magnetic field created by the rotor is due to the inset permanent magnets. Due to this efficiency and because the flux of the rotor cannot be increased via field windings, the PMG makes for a strong contender in many lower power distributed generation applications. It is also assumed that the stator windings are identical and sinusoidally distributed and displaced by 120° to obtain the sinusoid back electro-motive force (EMF). A cross-sectional view of a simple 2-pole permanent magnet machine can be viewed in Figure 2.1. Although Figure 2.1 only depicts the stator windings as one winding per phase, there actually exists a series of windings for each phase with a sinusoidal distribution of the number of turns in each stator slot. The windings run the length of the shaft and return symmetrically according to Figure 2.1. Each of the phases is internally wye connected at a neutral point. In Figure 2.2, the circuit diagram of the permanent magnet machine is shown which depicts the electrical characteristics of each phase and shows how the stator voltages can be expressed as functions of the current, stator resistance, and lumped inductance on each phase which takes into consideration all mutual inductances.

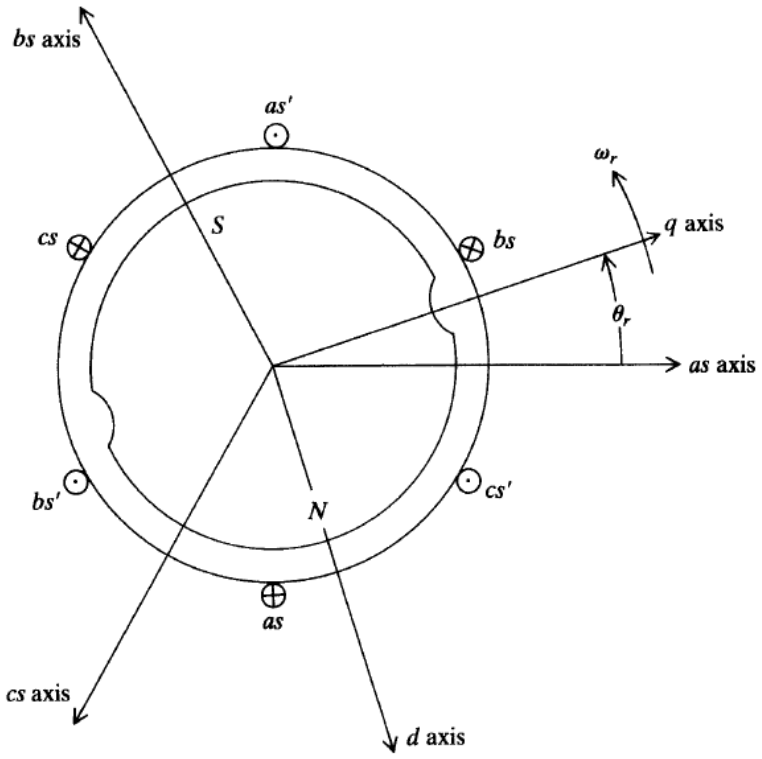


Figure 2.1: Cross-sectional view of a 2-pole, 3-phase permanent magnet machine [6]

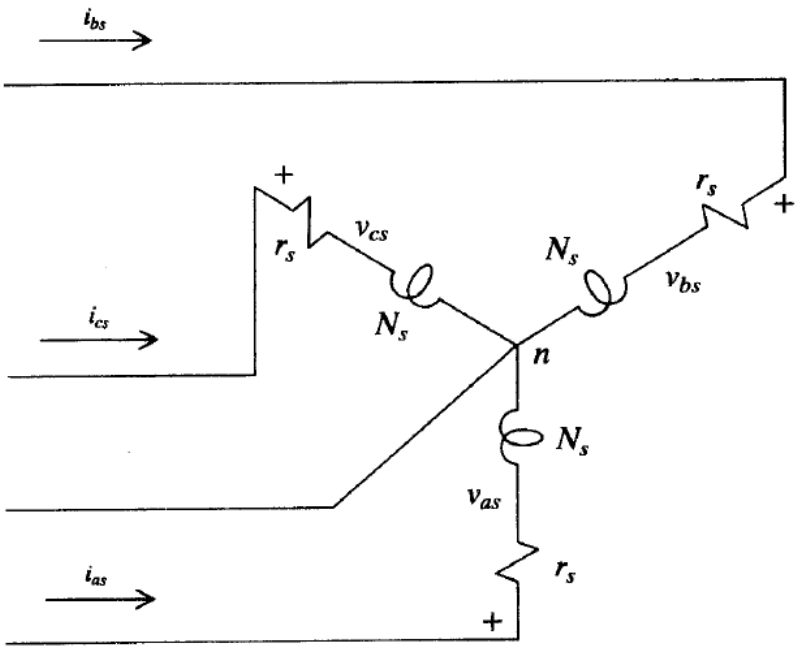


Figure 2.2: Circuit diagram of 3-phase permanent magnet machine [6]

When the permanent magnet machine is being used as a generator, the spinning rotor due to a mechanical torque will create a rotating magnetic field from the permanent magnets in which induces the EMF on the stator windings. This induced voltage in machine variables can be defined as follows [6]:

$$\boldsymbol{v}_{abcs} = \boldsymbol{r}_s \boldsymbol{i}_{abcs} + p\boldsymbol{\lambda}_{abcs} \tag{2.1}$$

Where the flux linkage of each phase is determined by,

$$\boldsymbol{\lambda}_{abcs} = \boldsymbol{L}_s \boldsymbol{i}_{abcs} + \boldsymbol{\lambda'}_m \tag{2.2}$$

The subscript *abcs* refers to the *abc* phase variables of the stator and is a 3x1 vector corresponding to the exact value on each of three different electrical phases of the stator. It is also important to take note that p is simply the derivative, $\frac{d}{dt}$, operator. $\boldsymbol{\lambda'}_m$ is the flux linkage established by the permanent magnets viewed from the stator phase windings and is a function of rotor position and the magnetic strength properties of the permanent magnets, this variable is also a 3x1 vector. The magnitude of $p\boldsymbol{\lambda'}_m$ would be the open circuit voltage induced in each stator phase winding. The inductance matrix $\boldsymbol{L}_s$ is derived according to [6] and is dependent on the position of the rotor and takes into account mutual inductances of all phases as well as leakage inductances. As can be seen, with so many variables dependent upon the position of the rotor, the calculations become rigorous and require a large amount of computing power, making this especially strenuous for real-time application. This is where the utilization of reference frame theory (RFT) comes into use in which simplifies the *abc* equations into *qd0* equations in which the 0 axis is zero for balanced three phase *abc* variables. It can be shown how utilization of RFT eliminates the dependency of rotor position and thus reduces the complexity of the differential equations involved in the dynamic calculations. A brief history and overview of RFT is presented as follows.

2.1.2 Reference Frame Theory Overview

RFT began its origins with R.H. Park for electric machine analysis in the late 1920's in order to simplify the complicated time-varying models used at the time [7]. The transformation used was fixed in the rotor. Later in the 1930's, H.C. Stanley developed a transformation that was fixed on the stator [8]. G. Kron also referred variables to a reference frame in synchronism with the magnetic field [9]. Then, in 1965, it was noted that all these transformations were all related directly by what today is called RFT and the arbitrary reference frame [10].

RFT expresses several difference types of frames. All frames use the same concept with the exception of a different rotational reference speed represented by ω. Examples of common frames are: stationary (where ω = 0), arbitrary (where ω is equal to an arbitrary speed selected by the user) and synchronous (where the ω selected for the transformation is exactly the same as that of the input waveform). The governing equations for *abc* (input waveform) to *qd0* and vice versa transformations are as follows [6]:

$$\boldsymbol{f}_{qd0s} = \boldsymbol{K}_s \boldsymbol{f}_{abcs} \tag{2.3}$$

$$\boldsymbol{f}_{abcs} = \boldsymbol{K}_s^{-1} \boldsymbol{f}_{qd0s} \tag{2.4}$$

$$(\boldsymbol{f}_{qd0s})^T = \left[f_{qs}\ f_{ds}\ f_{0s}\right] \tag{2.5}$$

$$(\boldsymbol{f}_{abcs})^T = [f_{as}\ f_{bs}\ f_{cs}] \tag{2.6}$$

$$\boldsymbol{K}_s = \frac{2}{3}\begin{bmatrix} cos\,\theta & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) \\ \sin\theta & \sin(\theta - \frac{2\pi}{3}) & \sin(\theta + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \tag{2.7}$$

$$\boldsymbol{K}_s{}^{-1} = \begin{bmatrix} cos\,\theta & \sin\theta & 1 \\ \cos(\theta - \frac{2\pi}{3}) & \sin(\theta - \frac{2\pi}{3}) & 1 \\ \cos(\theta + \frac{2\pi}{3}) & \sin(\theta + \frac{2\pi}{3}) & 1 \end{bmatrix} \tag{2.8}$$

In equations (2.3) - (2.6), f can represent voltage, current, flux linkage or electric charge. The superscript T represents the transpose of a matrix. Finally s represents variables, parameters and transformation used with stationary circuits such as the stator of a machine. This transform is very convenient as any of the main values of interest in a machine can be adapted to it.

The equations of transformation could be thought of as if f_{qs} and f_{ds} are directed along stationary paths that are separated by 90° rotating at the specified angular velocity ω. In analyzing Figure 2.3, it is important to note that when ω is zero and the initial displacement is also zero, the q-axis aligns with the a-axis and the d-axis is orthogonal at -90°. The 0-axis is also orthogonal and will be positioned along the z-axis. As previously stated, this magnitude will be zero so long as the *abc* system is balanced and can be observed using (2.3).

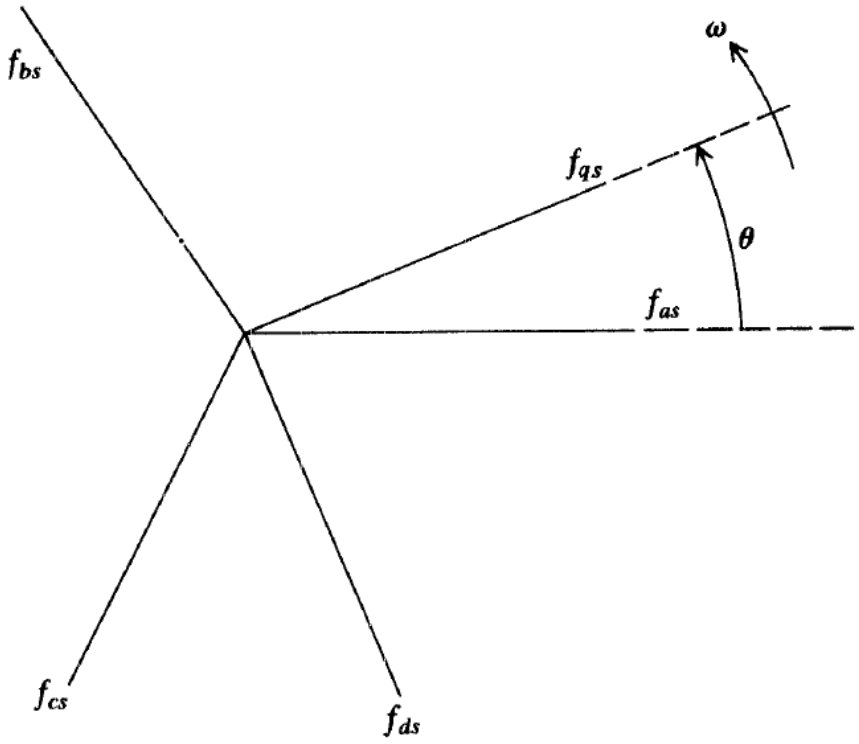


Figure 2.3: *abc* and rotating *qd0* coordinates [6].

2.1.3 *qd0* Dynamic Equations of PMG

In this study, the common *qd0* permanent magnet machine equations are used to solve for the stator currents of the PMG. The derivation of how (2.1) and (2.2) can be expressed in *qd0* using

(2.3)-(2.8) is derived in [6] and additionally shown in Appendix A. The derivation involves the transformation of the resistive terms as well as the inductance terms and derivative operator. The 2-axis equations presented in [6] and Appendix A can be used for isolated generator operation and expressed in terms of the *qd0* currents in generator notation and using the synchronous reference frame as follows:

$$i_d = \frac{1}{L_d}\int\left(-i_d(R_L + r_s) + \omega_e\lambda_q\right)dt \tag{2.9}$$

$$i_q = \frac{1}{L_q}\int\left(-i_q(R_L + r_s) - \omega_e\lambda_d\right)dt \tag{2.10}$$

$$\omega_r = \frac{1}{J}\int\left(T_{em} + T_{mech}\right)dt \tag{2.11}$$

$$\omega_e = \frac{P}{2}\omega_r \tag{2.12}$$

$$T_{em} = \frac{3}{2}\frac{P}{2}\left(\lambda_d i_q - \lambda_q i_d\right) \tag{2.13}$$

$$\lambda_q = i_q L_q \tag{2.14}$$

$$\lambda_d = i_d L_d + \Phi \tag{2.15}$$

ω_e refers to the angular speed of the electrical frequency while ω_r refers to the angular speed of the rotor. These two angular speeds are related by the number of poles, *P*, used in the machine as indicated by (2.12). T_{mech} is the applied torque in Newton-meters (*Nm)* and is positive for generator action and Φ is the magnetic flux linkage of the permanent magnets in volt-seconds (V-s). It can be noticed that V_q and V_d is redefined in terms of the load resistance, R_L. This is because the stator voltage is directly related to the current being drawn and the resistive load of the PFC device. The load resistance, R_L, is the equivalent Thévenin resistance of the PFC converter or resistive load at a particular time instant.

Real power output of the generator can also be solved for as follows [6]:

$$P_e = \frac{3}{2}\left(V_q I_q + V_d I_d\right) \tag{2.16}$$

Equations (2.9) – (2.15) are used to for the dynamic calculation of the stator currents in the *qd0* variables using the synchronous reference frame. This means that the ω used for transformation is the electrical speed and can be used for the inverse transformation to obtain the *abc* stator currents. The input into the system when being used as a generator is the mechanical torque in *Nm* on the rotor shaft and the voltage on the stator. In the case of this study and experiment, the load is simply characterized as a resistance which can accurately represent only a real power draw and does not consider the effect of complex power draw. The reasoning behind this assumption is further explained in Chapter 4.

2.2 Micro-turbine Structure and Modeling

The MTG system is gaining popularity among lower power distributed systems typically ranging from around 25kW to 500kW [11]. There are essentially two types of MTGs, which differ based on the shaft construction [12]. One is a high speed single-shaft unit with both the compressor and turbine mounted on the same shaft as the electrical synchronous machine. In this case, the rotor speeds typically range from 50,000-120,000 revolutions per minute (RPM). The other type of MTG is a split-shaft design in which uses a gearbox to reduce the mechanical speed of the output shaft to the electric generator. The split-shaft design often eliminates the necessary power electronic interface at the terminals of the electric generator and is more commonly found in propulsion drives [13]. Alternatively, the directly coupled topology results in a higher efficiency system over the split shaft design since the power loss in the gearbox is eliminated. However, this results in a higher rotational frequency of the rotor which often needs a power electronic interface to obtain the desired electrical frequency [14]. In the AES MVDC system, the high electrical frequency need only to be rectified to DC and therefore requires no need for a split shaft design and power efficiency will be maximized using a single shaft micro turbine. Micro turbines are considered a part of a general evolution in gas turbine technology, since techniques incorporated into the larger machines to improve performance can be typically found in micro-turbines as well. The physical working components of a gas turbine are shown in Figure 2.4 and are described as follows.

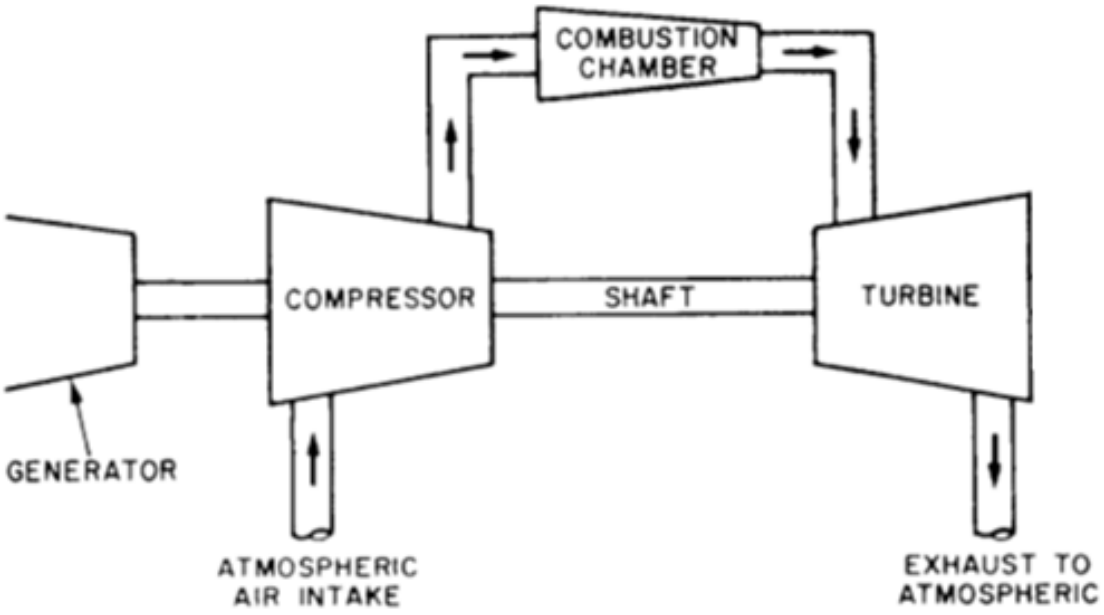


Figure 2.4: Components of a Single-shaft Gas Turbine [15].

The working of the single-shaft gas turbine can be described using the Brayton Cycle [15]. During the Brayton Cycle, air enters the atmospheric air intake. As the air enters the compressor the air pressure is increased and volume is decreased. The compressed air then enters the combustion chamber where it is mixed with fuel and burned. During combustion, the pressure remains the same but the increase in temperature causes an increase in volume. The gases then expand through the turbine where the rotor turns the kinetic and thermal energy into mechanical energy and thus causes a torque on the shaft. The cycle is then repeated. The mathematical modeling of the gas turbine and the micro-turbine has been performed in various works ranging from utilization of the thermodynamic equations governing the Brayton Cycle to models derived using transfer functions with parameters obtained from experimental measurement [14], [16] - [20].

The modeling of a micro turbine in literature is widely based on a two prevalent models. One being the heavy-duty gas turbine model by W. I. Rowen [16], and the other being the formerly Western System Coordinating Council (WSCC) compliant GAST model [17]. The heavy-duty gas turbine model by Rowen was derived and validated specifically for the GE 5001-

9001 gas turbine models with speed ratings from ~3000 – 5100 RPM and power ratings for 18MW – 106MW. The GAST model is slightly simplified from the Rowen model but has been reported to inaccurately represent the turbine dynamics when the temperature control becomes active [18], [19]. Despite these two models being developed for higher power rated systems on the order of MW, a significant amount of publications in MTG modeling utilize these models with adjusted time constants and control parameters to more accurately capture the faster transients observed in MTGs. Other work exists in MTG modeling using a thermodynamic approach [14], [20]. The model chosen for this work was the sing shaft heavy duty gas turbine model derived by Rowen due to the high popularity and various work of validation provided in the research community. Also, although a lack of strong evidence of verification of this model exists in the smaller high speed micro turbines, this model was chosen for this work due to the easy shift to heavy duty gas turbines on the order of MW generation for experiments involving the AES system.

The model shown in Figure 2.5 is a simple single-shaft structure derived according to [16]. The input to the system is the rotor speed which is often based on the attached generator dynamics and the rotor speed reference in per-unit *(p.u).* The output is mechanical torque applied to the shaft which may serve as input to the generator model. The model consists of three control loops and a minimum value selector. The minimum value selector sends forward the minimum value of each three control loops. The control loops are designed in such a way that during normal operation the speed control will be the active control while the acceleration control takes over in times of startup or faults to ensure a safe deceleration and the temperature takes over when high temperatures are reached and is often active in extended times of exceeded power rating activity. The resulting value is then passed through a minimum/maximum limit that corresponds to the fuel intake capability of the gas turbine. Next, a series of transfer functions corresponding to the valve positioner, fuel system and compressor delay is derived to determine the energy into the turbine dynamics to calculate applied torque on the shaft.

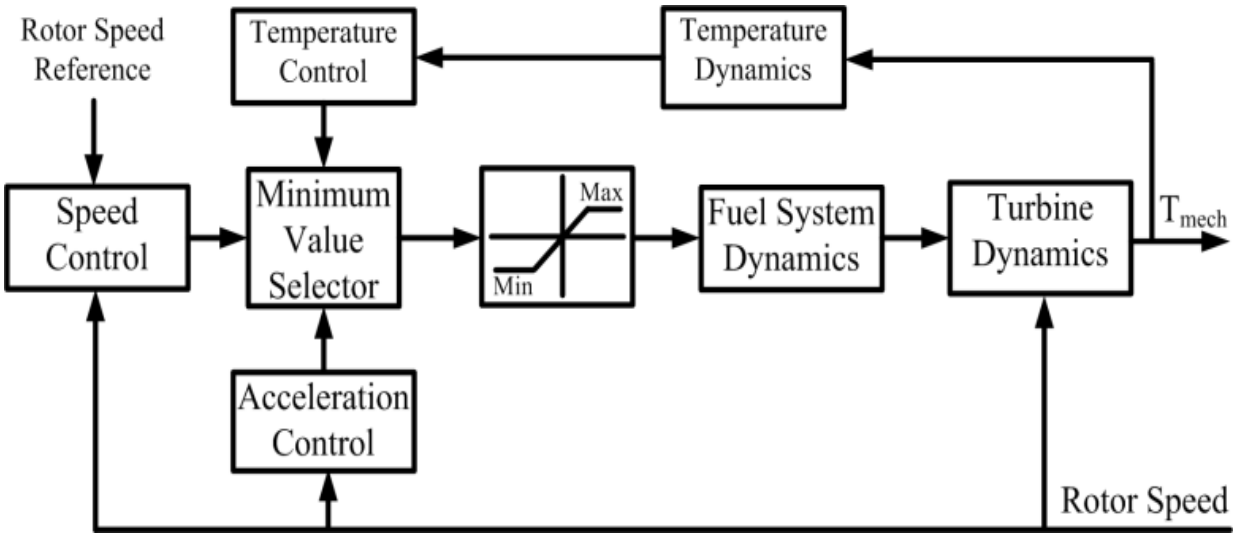


Figure 2.5: Block diagram of heavy-duty gas turbine model.

The exact transfer functions and equations for the micro turbine used in this thesis is presented in Appendix C and derived from [16] and [21]. As can be seen, the temperature control loop has been omitted due to space limitations on the Altera DE2 FPGA. This does not affect the dynamic response so long as the micro turbine operates within the power rating and can be further enforced by tightening the maximum fuel inlet allowance or by adding a power output controller. This control loop can be implemented in the future by either upgrading to a larger FPGA or by limiting the amount of fractional bits of variables within the FPGA.

2.3 Full MTG Mathematical Model Integration

The mathematical models of the micro turbine as well as the PMG can be integrated together according to Figure 2.6. As can be seen, the input into the system is the desired rotor speed of the single shaft system. The calculated rotor speed from the PMG dynamics according to (2.11) is then set as input into the micro turbine dynamics and used according to Figure 2.5. It is also important to note that the rotor speed output of the PMG needs to be converted into the per unit (*p.u)* system based on the rated rotor speed as the base value. This is because the micro turbine dynamics are derived using the *p.u* system. The output torque on the shaft also needs to be converted into real torque by multiplying by the base torque which is derived from the rated speed and power of the MTG system as shown in (2.17).

$$\tau_b = \frac{P_b}{\omega_b} \tag{2.17}$$

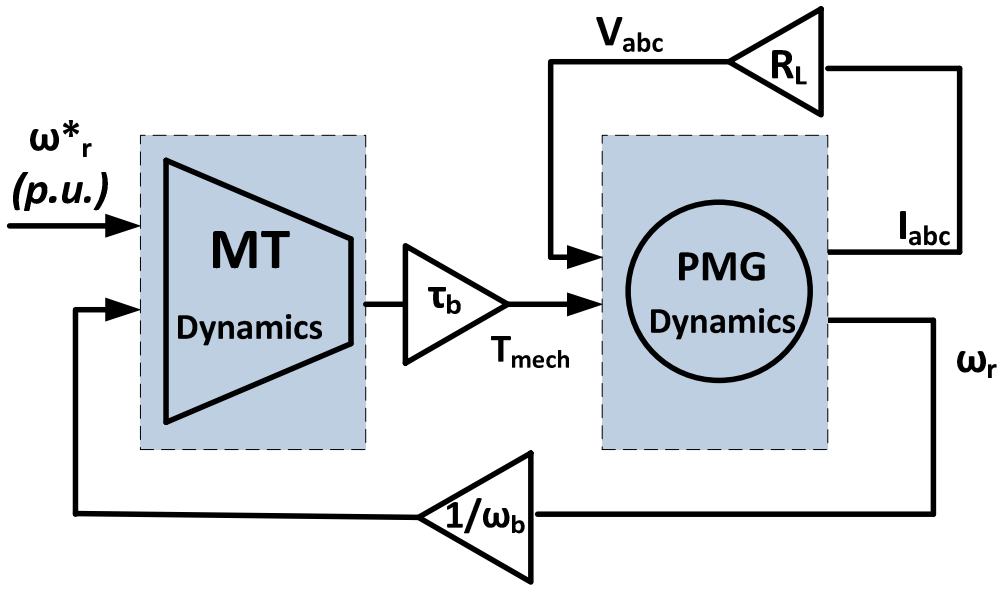


Figure 2.6: MT and PMG Mathematical Model Integration

This converted real torque is then passed into the PMG dynamic calculations. The stator voltage input to the PMG dynamics can be seen to be dependent on the resistive load across the stator of the PMG. This model assumes a purely resistive load consistent with the idea of a converter attached to the stator of the generator controlled as a power factor correcting device. The value of this resistive load then becomes a function of the power draw on the output of the PFC converter.

CHAPTER THREE

MULTI-PLATFORM REAL-TIME COMPUTATION SYSTEM

Real-time simulation is a requirement for HIL application and thus generator emulation. A real-time simulation is defined as a simulation of a system in which the simulation output corresponds with the real system in both numerical value as well as time. It can be noticed when using typical simulation software such as Simulink, the results will exist on a time axis and may be correct, however, the amount of time it takes for the simulation to compute may be faster or slower than the range of simulation time. It can be noticed that when simulating large systems involving many differential equations or mathematical operations, the real-time it takes to simulate a certain time period in simulation may take much longer and therefore not be able to simulate in real-time. Much of the machine real-time realizations performed today are done with the help of a costly real-time simulation platform such as dSPACE, Opal-RT, or Real-Time Data Simulator (RTDS) [4],[22]. Much of the platforms that exist today are based on parallel processing, high performance computer clusters and more recently field-programmable gate-arrays (FPGA) [4]. Other implementations use digital signal processors to obtain similar results [23]. Simulation using a sequential platform such as a digital signal processor cannot be simulated at the speed or at as small of a time step as is possible with an FPGA. The FPGA uses reconfigurable logic in a parallel architecture to compute and perform logic simultaneously whereas a processor performs tasks one step at a time. The FPGA parallel capabilities make machine modeling an ideal application due to the large amount of arithmetic involved that can be solved very rapidly when properly exploited. The choice of platform depends on the objectives and environment of the model.

This chapter will present a multi-platform computation system that was developed specifically for the real-time generator emulation. The platform consists of an FPGA, a DSC and PC working together to ensure real-time simulation and emulation of the generator system. The platform also consists of two different forms of communication protocols using both Serial Peripheral Interface (SPI) and RS-232. This chapter will also demonstrate the method in which the generator dynamics can be simulated in real-time, how it is implemented into the FPGA, and a real-time verification experiment.

3.1 System Architecture

The system developed for this real-time simulation is shown in Figure 3.1 and is a multi-platform HIL computation system. This system is developed for ease of data acquisition and for involvement of HIL application for generator emulation. The overall structure involves the Altera DE2 EP2C35F672 development board FPGA for real-time computation of the generator dynamics while a DSC performs control algorithms, interfaces with a control terminal PC using dSPACE ControlDesk, and interfaces to real hardware through the ADC input and PWM outputs. The topology of this system is explained in this section as follows.

The dSPACE unit is a rapid prototyping instrument used in various applications for development of digital system designs. The dSPACE unit can be interfaced with a PC and contains a software block set that can be used within the Simulink environment. dSPACE provides the capability to offload Simulink models into the computationally efficient dSPACE hardware that has the ability to run models in real-time at relatively low time steps on the order of micro-seconds. The dSPACE unit can also interface with various devices through a number of communication methods from analog to digital I/O or RS-232. This gives dSPACE the ability to run Simulink models in real-time and send and receive data between various devices which makes it ideal for data acquisition and user interface in digital systems. The data acquisition is performed using the dSPACE software, ControlDesk, which allows direct access to simulating input and outputs to view real-time results or change input variables.

The FPGA will compute the real-time dynamic calculations of the desired simulation. This Chapter will first demonstrate the real-time PMG simulation on the FPGA and then demonstrate the full real-time MTG simulation. The FPGA will interface with the DSC which will supply the input to the dynamic equations as the FPGA will compute the output and send back to the DSC. The FPGA will communicate to a Texas Instruments (TI) TMS320F28335 DSC through SPI. The SPI core implemented within the FPGA was obtained from an open source provider and interfaced with the FPGA code [24]. The FPGA is configured as the master device, while the DSC operates as the slave. The FPGA will initiate transmit of data at the beginning of each time step and thus send the new state variables, just after they are calculated. The DSC will communicate to the PC running the dSPACE real-time environment through RS-232. Additionally, the DSC will be a key component in HIL by its interaction with the three-phase inverter. The DSC will send the pulses to the IGBTs of the inverter via the PWM digital output

and will collect readings of voltage and current on the output of the inverter using transducers and the ADC inputs. The purpose of each component of the multi-platform computation system can be summarized in Table 3.1.

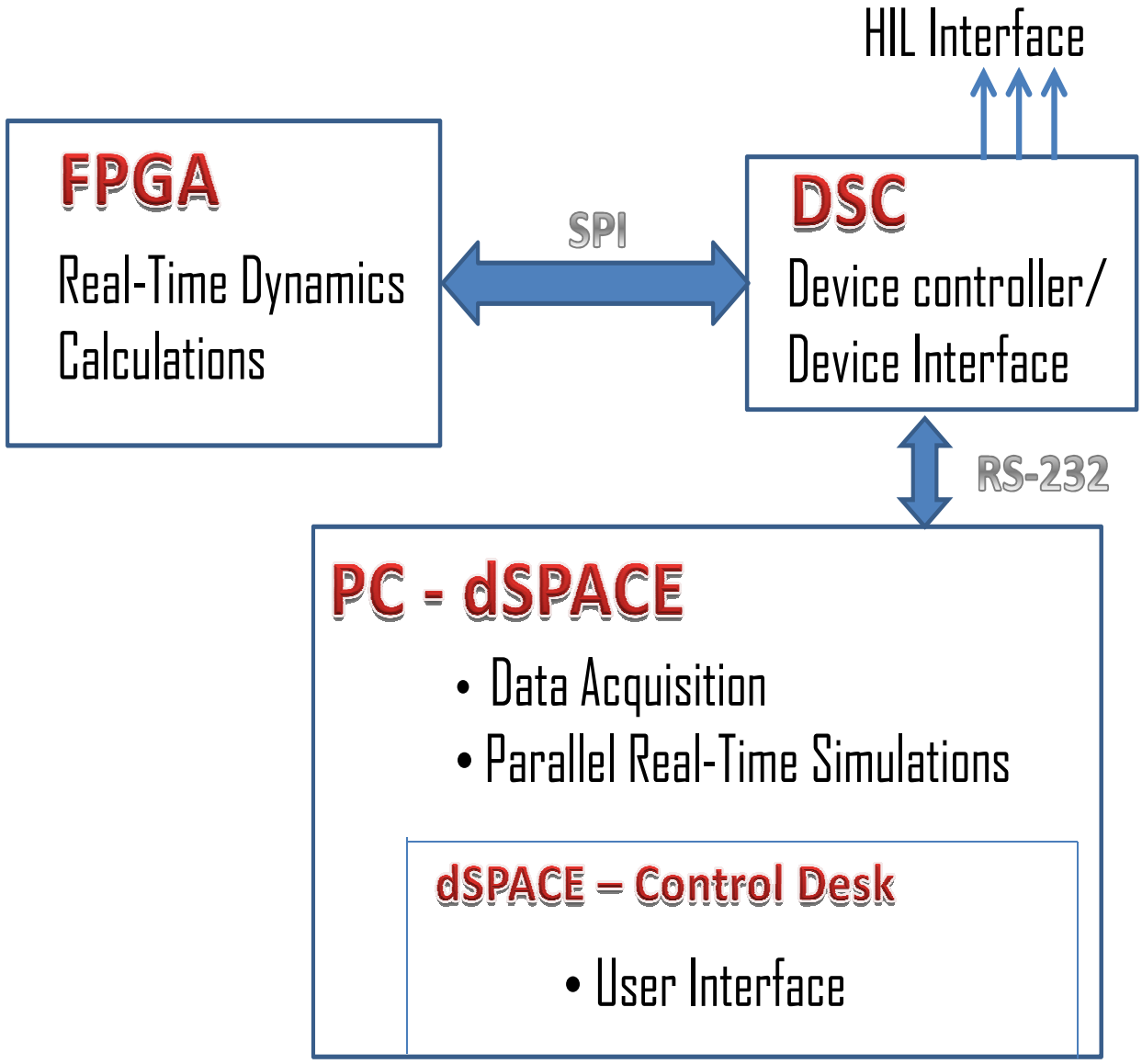


Figure 3.1 Architecture of multi-platform real-time computation system.

TABLE 3.1

Multi-platform computation system components and purposes

Device	Purpose
FPGA	Real-Time calculation of MT and PMG dynamics
DSC	Performs PWM generation, interfaces to hardware and user interface on PC
PC/dSPACE	Supports real-time interface in which control variables can be modified and serves as data acquisition in which variables can be viewed and collected.

3.2 Real-Time Simulation of PMG on FPGA

This section demonstrates a simplified method with use of the FPGA to implement a real-time dynamic simulation of a PMG. The structure aims to reduce the coupled ordinary differential equations (ODE) involved in the model into a simple set of equations utilizing Euler's Method that can be easily programmed to an FPGA for real-time simulation suitable for real-time interaction. The dynamic model implemented in this design is the two-axis *qd0* model [6].

Much of this past work in FPGA machine modeling has been done using a multiple clock cycle process [25], [26]. The structure and implementation in this work demonstrates a simple process that uses one clock and takes advantage of the IEEE arithmetic and numeric types libraries. The following section demonstrates a guideline and way of simplifying the coupled ODEs involved in PMG machine simulation into a set of equations using Euler's method that can be solved at an easily adjustable time-step and in real-time. The following subsections will explain the numerical analysis method chosen as well as how it is used to solve the PMG ODE's and finally how it is implemented into the FPGA and verified in real-time.

3.2.1 Equations and Model Verification

The ultimate goal is for the PMG dynamics to be realized on the FPGA and able to run and be interacted with in real time. The process of achieving such a goal was taken one step at a time as explained in this subsection as listed follows.

- **Step 1**: Verify mathematical models and differential equations via Simulink.
- **Step 2**: Reproduce results in MATLAB using .m code and a numerical analysis method to solve the ODEs.
- **Step 3**:Implement equations and numerical analysis method in VHDL and verify the results and timing against MATLAB and Simulink simulations.

As can be seen, the first step performed is to simulate the generator using Simulink. This involves using the graphical interface and automatically generated ODE solving technique at a desired time step to first verify the mathematical model. This mathematical model was derived in [6] as previously explained and is verified against the commonly used SimPowerSystems Toolbox or PLECS Toolbox. Each of the previously mentioned tool boxes have machine models available for use by the user and are widely referenced and validated. This means that the differential equations used in Simulink to model a PMG should yield the same results of the models used in the tool boxes for verification. The parameters used in the machine modeling are experimentally obtained and often published in literature for different machines under simulations. These parameters typically include the stator resistance, rotor inertia, leakage inductances, pole count, and flux due to the permanent magnets of the rotor. After the mathematical model tested by the user matches the results accepted tool box models and the results shown in literature simulations, the next step is to create a set of equations that can be solved in a discrete platform.

As previously stated, this work is done automatically in Simulink and simply allows the user to graphically solve any set of differential equations by simply graphically connecting variables when needed and using integration block sets. In reality, solving such a set of coupled differential equations require precise use of an ODE solving method, specific time step analysis, and proper coupling of state variables in the differential equations where required. At this stage, it is advantageous to create the simulation in a discrete solving platform such as MATLAB code.

In using MATLAB, an *.m file* was created to solve the coupled differential equations involved in the PMG simulation while involving a numerical ODE solving technique. Subsection 3.2.1 will go into further detail on the exact numerical solving technique used and for what reason. After the *.m file* has been verified to yield the same results as the Simulink model, the set of equations are now ready to be implemented in a digital platform such as microcontroller, digital signal processor, or FPGA.

The main goal of the digital platform implementation is to now match the *.m file* results from a functional perspective and also with a heavy emphasis on correct timing of results, especially for real-time application. In implementation of the PMG simulation on the FPGA, the first step is to validate the FPGA is calculating the correct values and was performed by outputting various variables to the seven segment display. In outputting certain variables, such as the rotor speed to the seven segment display, the start up speed increase can be viewed up until the steady state speed is achieved. Once steady state rotor speed is established, this value can be compared to the simulated results of both the *.m file* and the Simulink simulations. If this value is the same than it is apparent that the calculations are functionally correct and the next step is to verify that the timing is also correct. The way in which timing is implemented and validated is performed in the remainder of this chapter but involves more detailed analysis of the results, often requiring output of data to graphical real-time interface such as dSPACE for data acquisition.

3.2.2 Numerical Solver of ODEs

A numerical analysis method is required to solve a system of ODEs in a discrete platform. The Euler methodology has been selected in this work. The advantages and preference of this approach is explained as follows as well as the implementation of this method for (2.9) – (2.11). Euler's Method is often viewed as the most simplistic form of ODE solving but may be susceptible to error. On the other hand the Runge-Kutta 4^{th} (RK4) order is a commonly used method due to the high accuracy achieved [27]. This form of numerical analysis has been performed in a similar project on a higher end FPGA, the Xilinx Virtex-5 [26]. However, due to the higher amount of computations involved in RK4, implementation of such a method on an FPGA will result in much more silicon usage compared to a simpler method such as Euler's Method. In fact, these two methods were implemented solving the ODE's of a PMG as both an *.m file* in MATLAB and also on the DE2 and compared. The *.m file* for comparison of both

methods is presented in Appendix B. Figure 3.2 shows the rotor speed calculations using both Euler Method and RK4 during start-up of a PMG with parameters shown in Table 3.2 and using a resistive load of 10 Ω with an input torque of 10 *nm*. This plot was created using the attached *.m file* in Appendix B. It can be noticed that at small time steps there is a negligible difference in accuracy between the two methods, however, the amount of logic elements (LE) used in an FPGA for Euler's method is much less which is very advantageous in large designs.

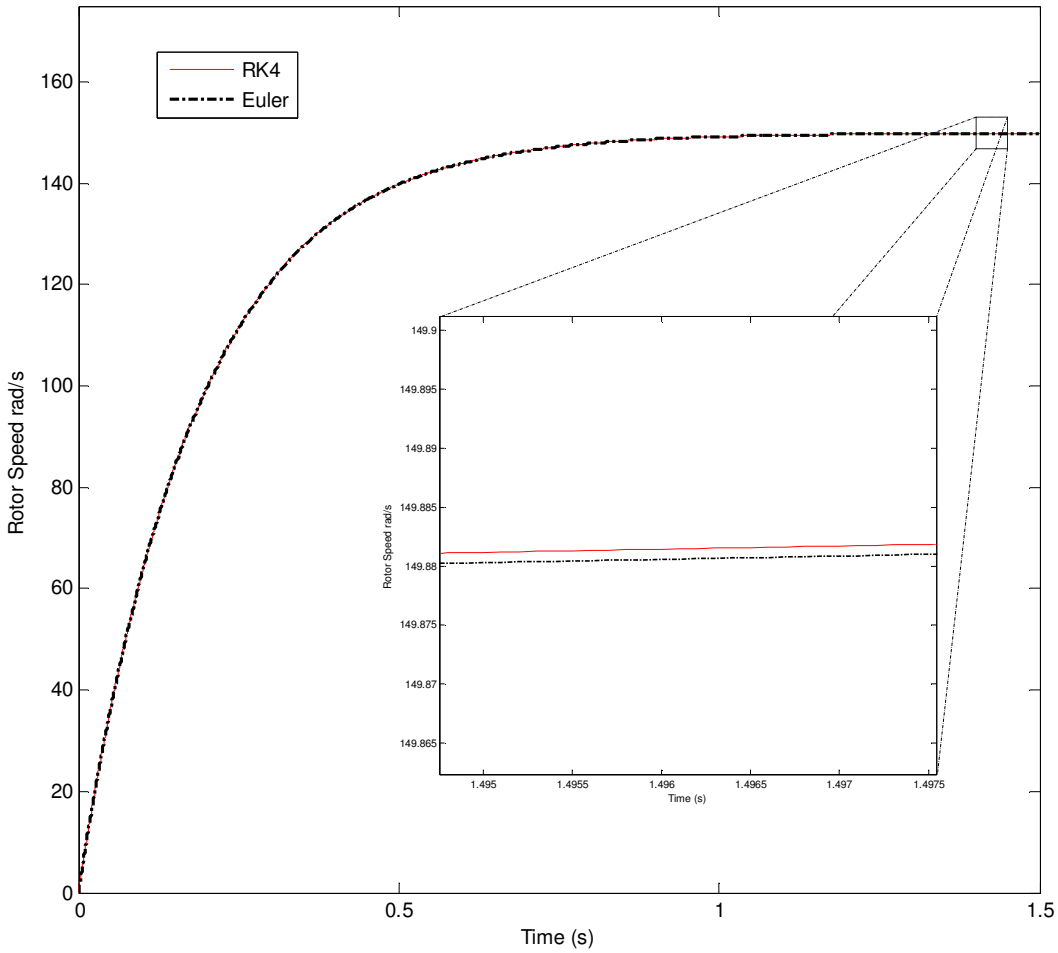


Figure 3.2 Euler method and RK4 comparison of PMG simulation.

The difference in LE usage of the two methods for the simulation done in this work using the Altera DE2 board under the same parameters and fixed point precision is shown in Figure 3.3.

This argument is also further demonstrated in [25]. From this experimentation, it is shown that Euler's method can be used due to the very small time-step and also because the use of the *qd0* model eliminates much of the time varying coefficients. Euler's method for integration and ODE solving is shown as follows [27].

$$y' = f(x, y) \tag{3.1}$$

$$y = y_0 + hf(x, y) \tag{3.2}$$

y_0 is the previously calculated value and h is the value of the desired time step.

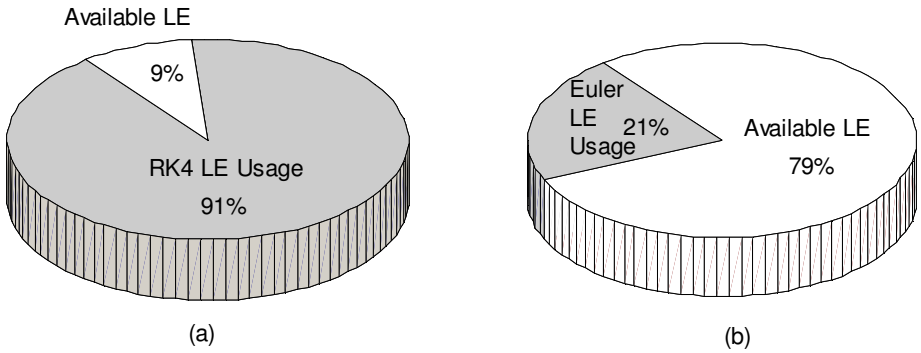


Figure 3.3. Percentage of LE usage on Altera DE2 development board. (a) RK4 method (b) Euler method.

3.2.3 PMG Equations Using Euler's Method

Using Euler's Method and rearranging (2.9) – (2.15) yield the following six equations suitable for the FPGA. The subscript (0) represents the previous time-step value, which is the value retained in the latch which will be explained in the next subsection.

$$f_q = \frac{1}{L_q}\left(-i_{q_{(0)}}(r_l + r_s) - \omega_{r_{(0)}}\frac{P}{2}(i_{d_{(0)}}L_d + \Phi)\right) \tag{3.3}$$

$$f_d = \frac{1}{L_d}\left(-i_{d_{(0)}}(r_l + r_s) + \frac{P}{2}\omega_{r_{(0)}}i_{q_{(0)}}L_q\right) \tag{3.4}$$

$$f_w = \frac{1}{J}\left(\frac{3P}{4}\left((i_{d_{(0)}}L_d + \Phi)i_{q_{(0)}} - i_{q_{(0)}}L_q i_{d_{(0)}}\right) + T_{mech}\right) \tag{3.5}$$

$$i_d = i_{d_{(0)}} + hf_d \tag{3.6}$$

$$i_q = i_{q_{(0)}} + hf_q \tag{3.7}$$

$$\omega_r = \omega_{r_{(0)}} + hf_w \tag{3.8}$$

These equations are derived and separated accordingly to be easily implemented in a VHDL structure to take advantage of the parallel architecture and real-time timing constraints. It will become clear in the next subsection how these equations are defined and separated into concurrent and sequential clock triggered processes.

3.2.4 FPGA Implementation and Data Representation of PMG Dynamics

For real-time simulation to take place, the clock speed and the computational time step, *h*, must be taken into consideration and are dependent upon one another. If a particular computational time step is used, the state variables must be updated at a period equal to this value. Since the DE2 has two on-board clocks of 27 MHz and 50 MHz a clock divider must be implemented to achieve the desired frequency. For this study, a computational time step of 1 μs was chosen. This time step was arbitrarily chosen and can be decreased if desired, though a limit does exist due to propagation and other delays. This means that the state variables must be updated at 1 MHz and a clock divider to reduce the 27 MHz clock to 1 MHz is used. Ideally, this would be accomplished by dividing the 27 MHz clock by a factor of 27, however, it was discovered that measurement via an oscilloscope may be necessary to achieve the desired frequency. This is because the clock speeds may not be exact and because real-time simulation depends on the accuracy of the clock speed, it is important to test and adjust the clock divider to achieve the desired speed. A simple structure shown in Figure 3.4 is implemented in VHDL which uses a process to update the state variables and implements a latch to retain the values on the rising clock edge. The arithmetic of equations (3.3) – (3.5) is computed asynchronously using simple combinational logic, while equations (3.6) – (3.8) are solved for and retained using a latch on the rising clock edge. Equations (3.6) – (3.8) are only computed on every rising edge

of the 1 MHz clock and will thus be computing in real-time using a computational time step of 1µs. Therefore, $h = 1e^{-6}\ s$.

The choice of time-step was chosen with regards to desired degree of accuracy and limitations outside of the FPGA. These limitations will be explained in Chapter 5 and are dependent upon the communication delay and program run-time of the DSC. As long as the time-step meets the criteria shown in Figure 3.5, the time-step can be used for the calculations within the FPGA. This criterion is that the time it takes to compute the new state variables plus the asynchronous functions depicted as *x* must be less than the computational time-step, *h*, in which the next state variables begin calculation.

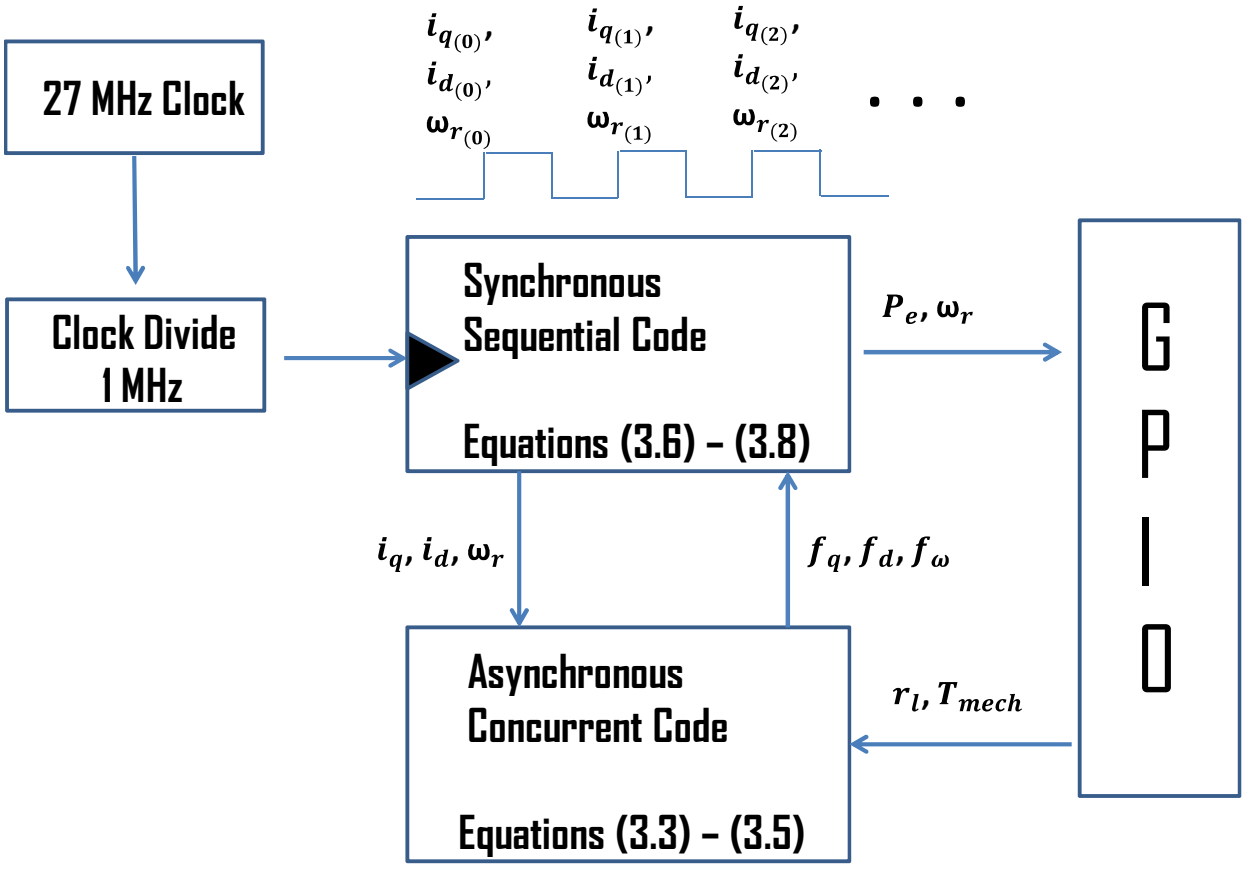


Figure 3.4. Block diagram of FPGA coding architecture of PMG real-time simulation.

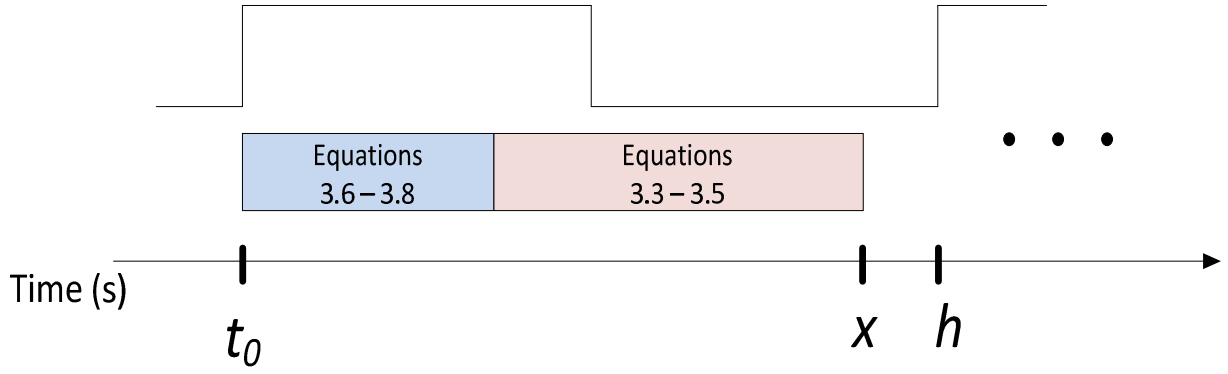


Figure 3.5. Timing constraint for FPGA real time computation.

The FPGA implementation described in this paper makes use of the Fixed Point library originally designed for use In the IEEE 2008 standard [28]. Utilization of this library allows for the advantages of fixed point accuracy with user friendly arithmetic and conversion functions. This is very useful in that many of the parameters involved in machine design are very precise and require the accuracy of fractional bits. This library allows for the programmer to assign the signals with desired accuracy in terms of the number of fractional bits. Floating point is also offered in this library though fixed point is preferred over floating point due to the fact that precision can be adjusted to save silicon space on the FPGA chip, a limited resource on FPGAs.

3.3 Real-Time Verification Experiment and Results of PMG Simulation

This section verifies the real-time simulation of an isolated PMG performed on an FPGA and compares the results to the widely accepted MATLAB/SIMULINK toolbox, SimPowerSystems model [29]. The real-time simulation will be verified using dSPACE which will run the SimPowerSystems model in real-time and allow for input operations to both the SimPowerSystems model and the model running on the FPGA simultaneously. Keep in mind that dSPACE is only used for verification as one of the advantages to implementing the design on the FPGA DE2 board is to demonstrate the ability to achieve real-time simulation at a smaller time step without the costs of a real-time simulation platform. In using dSPACE, real-time interactions with both models are possible by varying the input torque and the resistive load on the stator via dSPACE Control Desk. When using this model for HIL realization, these inputs

will be directed from the DSC making decisions based on the analog readings of the interfaced distributed power system. This sort of topology is being used and demonstrated for studies of a ship power system [3]. However, in the previously mentioned work, the simulation is being performed on a PC based system and is unable to reach the small time steps achievable on a FPGA thereby limiting the observable transients and also capping the frequency of which the generators can be run. Additionally, the simulation performed in the previously mentioned work involves DC rectification on the output of the generator and thus emulates DC voltage instead of the direct stator three-phase AC output that this work provides.

The structure of this specific simulation shown in Figure 3.6 will allow for real-time interaction through the GPIO of the DE2 board. The DE2 board will accept inputs of torque in *Nm* and load resistance in terms of ohms. The DE2 board will also output rotor speed in *rad/s*, and electrical power output in watts (W). Although any variable calculated within the model can be set as output, rotor speed and power was selected to compare with an identical model running in dSPACE. The real-time environment of dSPACE will allow the parameters of resistive load and shaft torque to be adjusted via a user interface and sent to the GPIO of the FPGA and updated within the comparison simulation in dSPACE. The torque input will be replaced with the micro turbine model with rotor speed commands in the experiment discussed in Section 3.4.

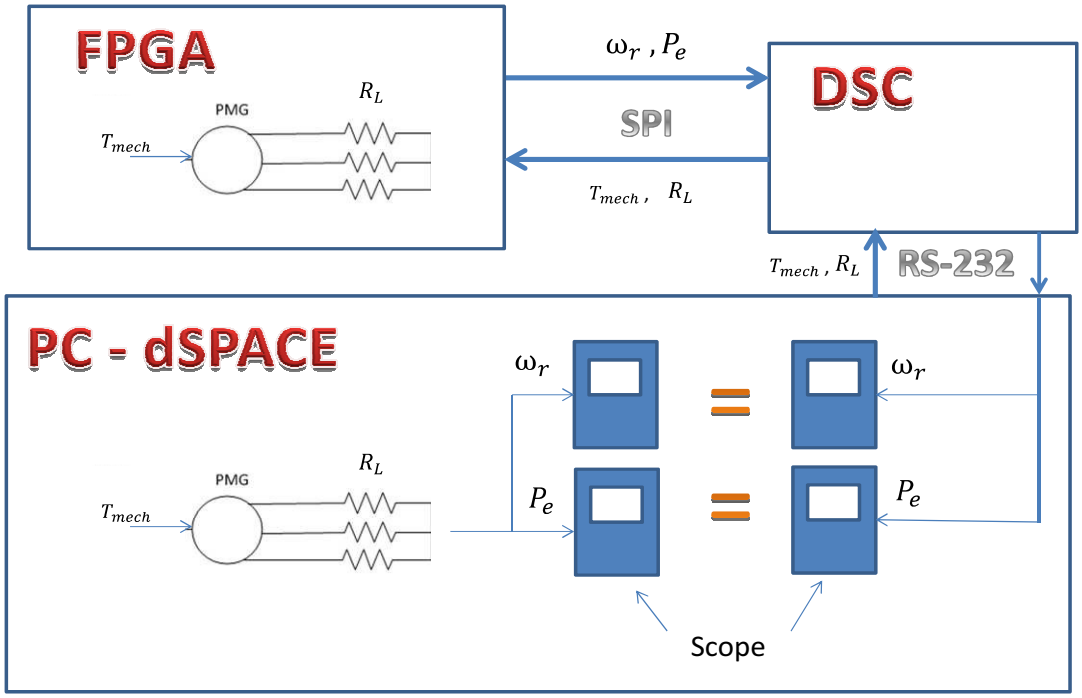


Figure 3.6. Structure of FPGA timing verification compared with the real-time dSPACE simulation.

The experimental setup in hardware is shown in Figure 3.7.

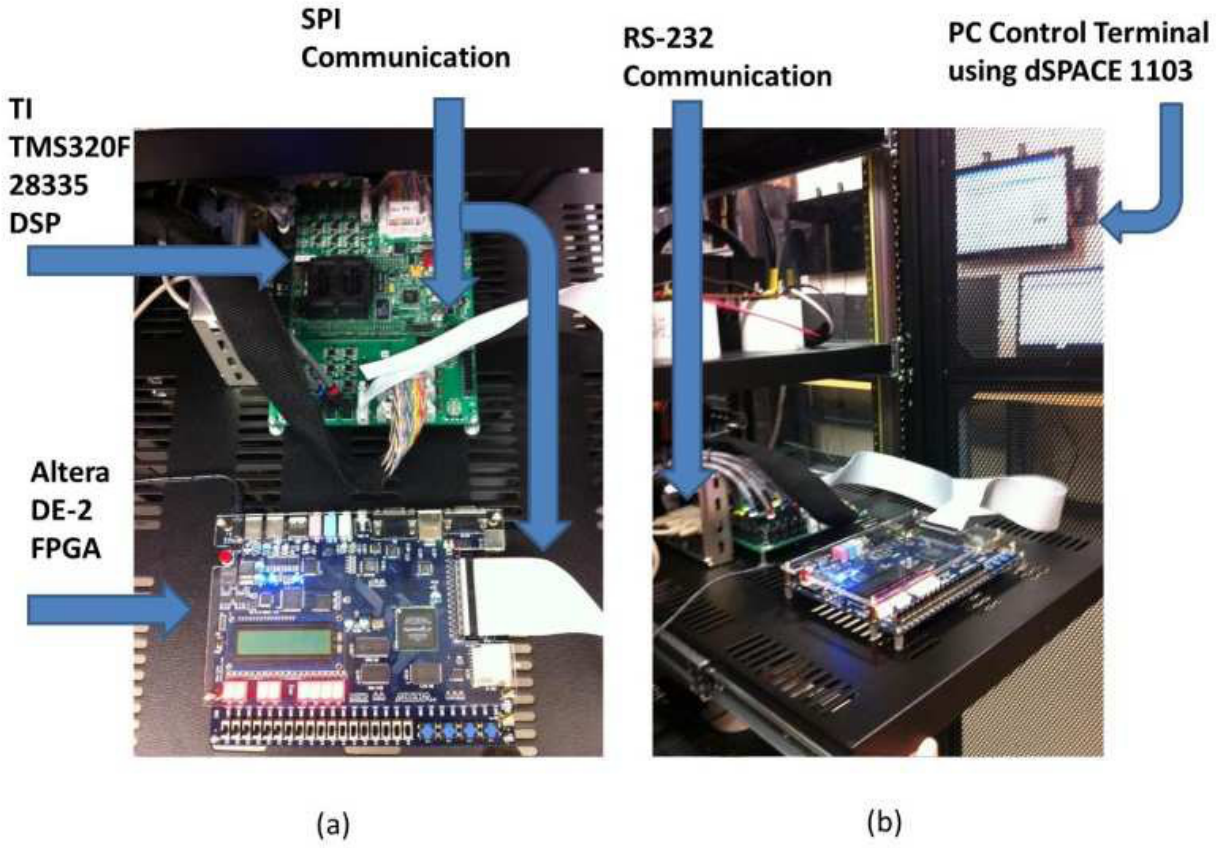


Figure 3.7. Hardware experimental setup. (a) Front view – Displays FPGA and DSC connection interface. (b) Side view – Displays control terminal (PC) and communication interface between DSC and dSPACE terminal (dSPACE terminal not pictured).

As previously mentioned, the model running on the FPGA is compared with the SimPowerSystems permanent magnet machine model simulated in dSPACE using the same parameters shown in Table 3.2 obtained from [30]. The parameters used in this simulation pertain to an 8-pole PMG rated for 11kW at 3600 RPM. This machine was arbitrarily selected to demonstrate the effectiveness of the FPGA model and can be replaced with the parameters of the user's interest. These two models are running in real-time and are both under the same changes in input of mechanical torque on the shaft and resistive load on the stator. In Figure 3.8, the stator resistance is kept at a constant value while the torque undergoes a step change and then another step change back to zero. The rotor speed curve accelerates to a certain value when the torque is applied and then decelerates once the torque is brought back to zero. It is important to

notice how the rotor speed calculated on the FPGA follows closely with what is calculated using the SimPowerSystems model in dSPACE. In Figure 3.9, both the stator resistance and the shaft torque are modified as the PMG models react appropriately. Again, the FPGA model follows the expected values of the compared equivalent model.

TABLE 3.2

Parameters of PMG [30]

Parameter	Symbol	Quantity
Magnetic flux of permanent magnets	Φ	0.1723 V·s
Rotor inertia	J	0.0121 kg·m²
Number of poles	P	8
d-axis inductance	L_d	0.0026 H
q-axis inductance	L_q	0.0026 H
Stator resistance	r_s	0.45 Ω

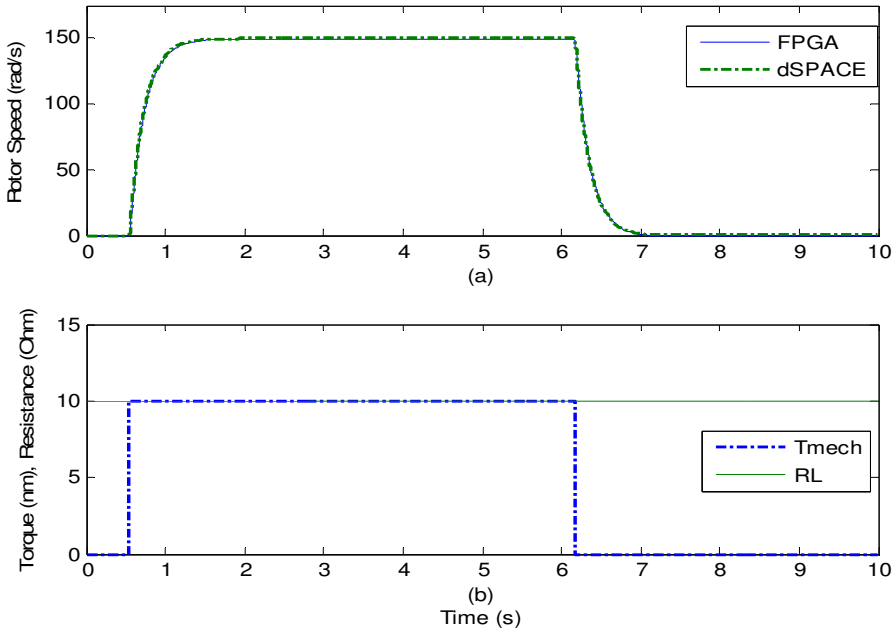


Figure 3.8. Real-time FPGA calculation verification with variable mechanical torque input. (a) Rotor speed of PMG of both FPGA and dSPACE model under the same excitation and parameters. (b) Excitation of PMG, constant resistive load on the stator with a step increase of mechanical torque on the rotor.

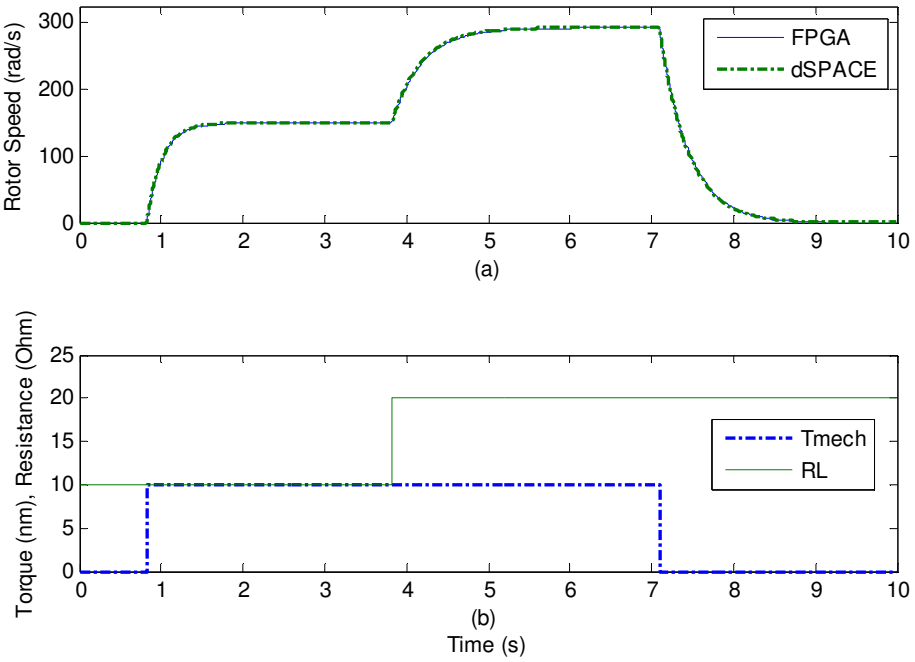


Figure 3.9. Real-time FPGA calculation verification with variable mechanical torque input. (a) Rotor speed of PMG of both FPGA and dSPACE model under the same excitation and parameters. (b) Excitation of PMG, step resistive load on the stator with a step increase of mechanical torque on the rotor.

Finally in Figure 3.10, the electrical power outputs of the machines are compared during various step changes in applied shaft torque. The electrical power is calculated using (2.16). By analyzing the data, it is evident that the model simulated in real-time using the FPGA reacts equivalently as the widely accepted SimPowerSystems model simulated in dSPACE. Also, the FPGA is solving at a smaller time step than is performed in dSPACE and can thus react to a wider range of input pulses and accurately simulate higher speed machines than is possible in dSPACE or many other real-time simulation platforms. This simulation proves that the set of equations utilizing Euler's method and the simple method in which they are implemented onto the FPGA can indeed offer a much cheaper alternative to real-time simulation platforms and even exceed them.

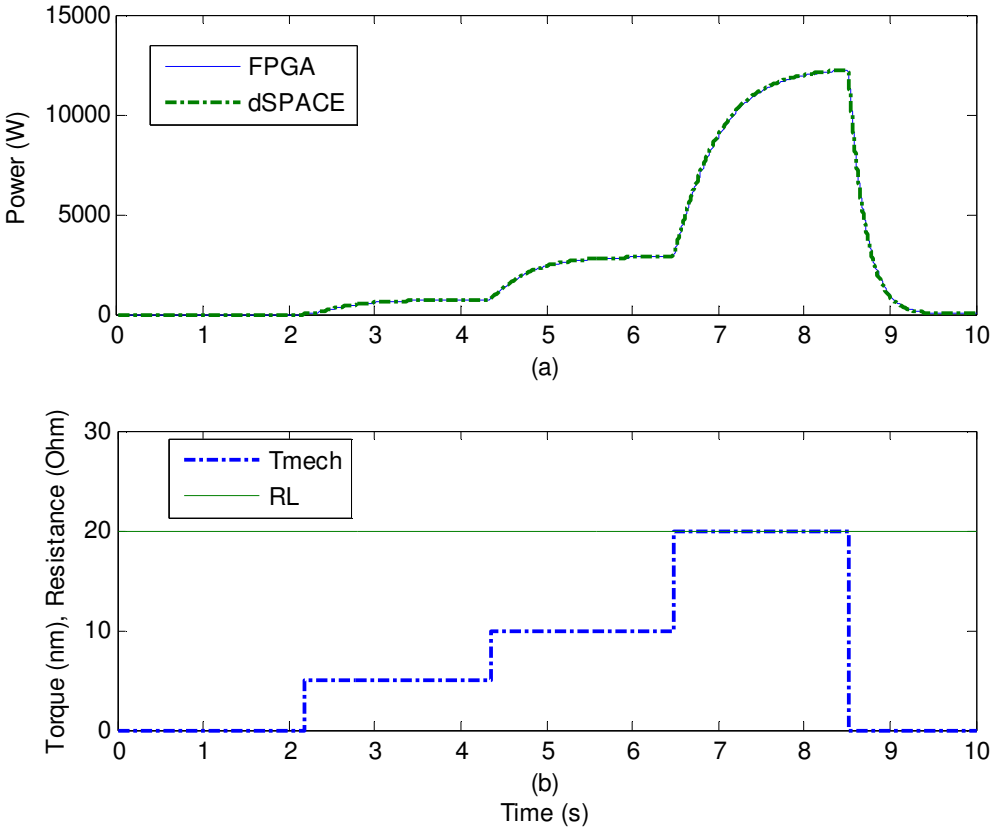


Figure 3.10. Real-time verification of power calculation on FPGA. (a) Power output of PMSG of both FPGA and dSPACE model under the same excitation and parameters. (b) Excitation of PMSG, constant resistive load on the stator with a series of step increases of mechanical torque on the rotor.

3.4 Real-Time Simulation of MTG on FPGA

The simulation of the micro turbine system is derived in [16] and involves a series of transfer function expressed in the *s*-domain. As previously stated, the micro turbine dynamics can be seen in Appendix C. In order to realize the transfer functions in a discrete platform, the equations need to be converted into a state-space representation in which the state variable can be solved for using a numerical analysis technique such as Euler's method [31]. The output can be solved for using algebra and is a function of the state variable. The steps taken to solve a transfer function and time delays in a discrete environment are shown in Figure 3.11.

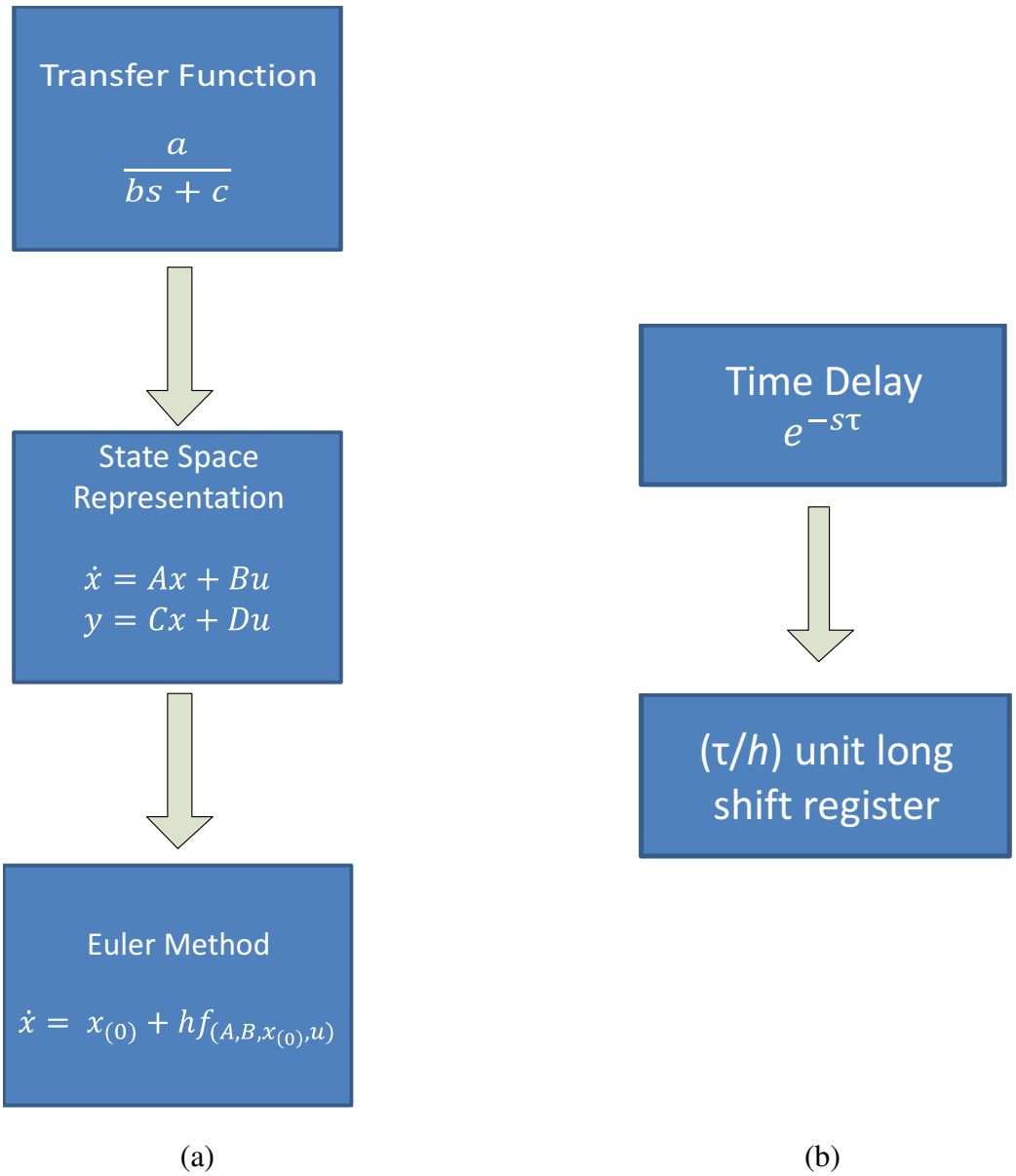


Figure 3.11. Discrete realization of transfer functions. (a) Transfer function in the *s*-domain conversion for realization in a discrete platform. (b) Time delay in the *s*-domain realized in a discrete platform.

As can be seen, the transfer function can be expressed in state space form involving two equations, the first equation describes a differential equation in which the state variable is *x*. in solving this differential equation, the value of the variable in the next state is solved for at each time step and can be used to solve the second equation which describes the output of the system. This output is a function of the state variable and can also involve the input to the system.

Realizing a series of transfer function in a discrete environment involve a series of state equations solved at the mathematical time-step as previously described in Section 3.2 and coupled so as the output of one system is calculated as the input into the following state-space equation and solved accrodingly. An example of this discrete realization for the transfer function of the valve positioner is shown as follows:

$$F_s(s) = \frac{K_v}{T_v s + c} \tag{3.9}$$

where $K_v, T_{v,}\ and\ c$ are the extracted parameters of the valve system of a particular gas turbine. When translating into state space form according to [31], the next states as well as output equations are derived as follows.

$$\dot{x} = \frac{-c}{T_v} x + u \tag{3.10}$$

$$y = \frac{K_v}{T_v} x \tag{3.11}$$

Likewise to what was performed previously in validating the PMG discrete dynamics, a *.m file* was created to prove Euler's method sufficiently solved the dynamics and matched exactly with the Simulink expected results. The *.m file* created for this experiment is shown in Appendix D and involves a directly coupled micro turbine with PMG.

The type of structure implemented onto the FPGA is similar to the previously explained PMG dynamic equations in that the differential equations can be solved for in real-time so long as the state variables are solved for and updated at the same period as the computational time-step. The output equations can be solved for concurrently and do not need to be clock triggered since these signals only update as the state variable or input changes. This sort of structure allows for an integration with the PMG dynamics in a similar fashion in which the set of equations can be divided into two subsets, synchronous and concurrent sets. It is also important to keep in mind that the values used as input into the micro turbine equations needs to be converted into *p.u.* and converted back when used in the PMG dynamic equations.

A similar experiment as the verification of the PMG alone described in Section 3.3 is

performed to validate the FPGA computation of the directly coupled MT with PMG. In this verification, dSPACE was used similarily as a user interface and a means of changing the simulated resistive load on the stator terminals. The speed reference was always kept constant at 1 *p.u.* The verification setup can be viewed in Figure 3.12. It is important to mention that the simulation of the coupled micro turbine and PMG simulation was unable to run in real-time in dSPACE due to the mathematical complexity. Therefore, a non-real-time simulation was run in Simulink in which the excitations were matched in time with what was performed during the FPGA simulation with the excitation change using dSPACE. The resulting data from each experiment was then collected and graphed on the same plot to show the comparison of results of the rotor speed calculations on a similar time axis.

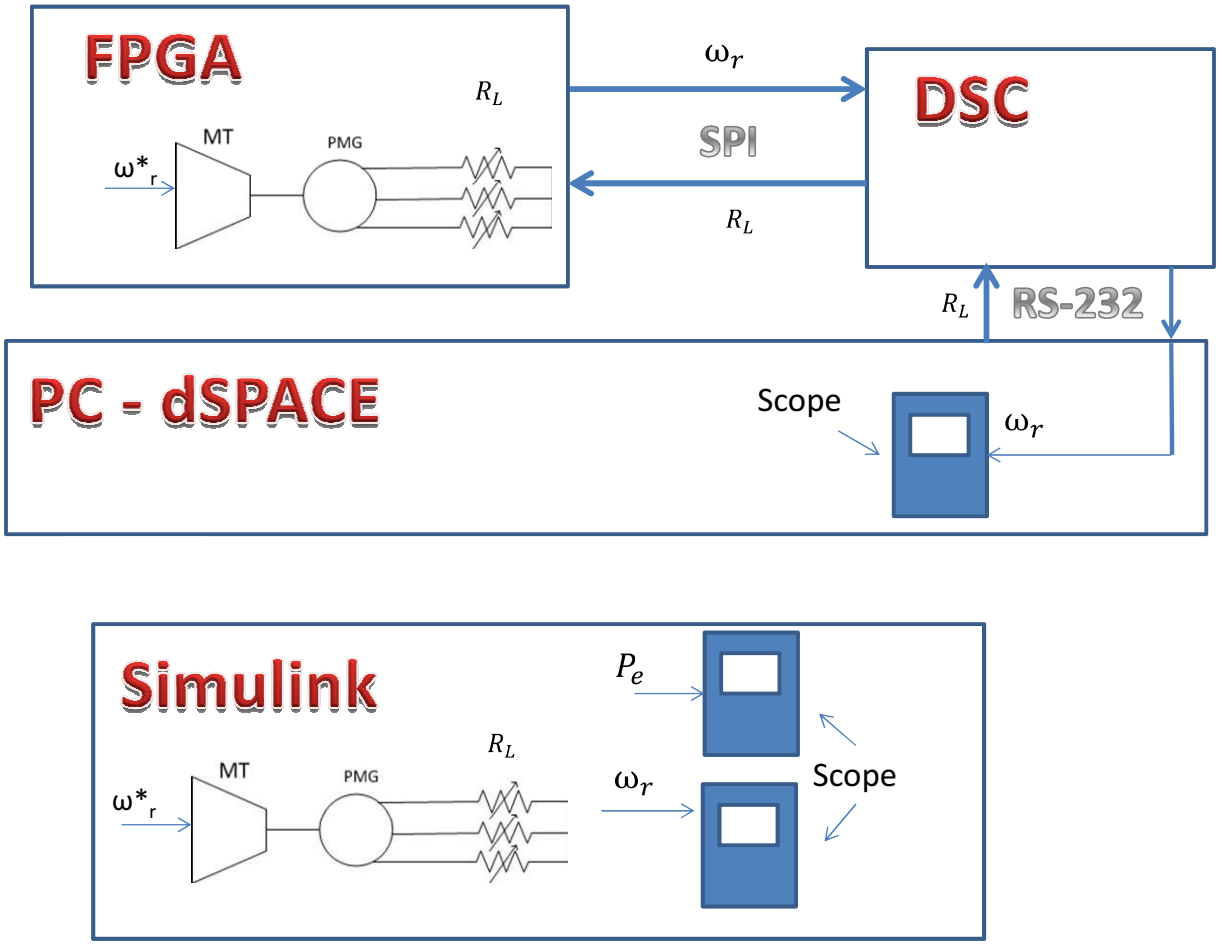


Figure 3.12. Structure of FPGA timing verification of MTG simulation.

Figure 3.13 shows a simulation of the dynamics computed in the FPGA and also an identical

model in simulink using the same parameters and excitations in time. The parameters used in the experiment are taken from [21] and shown in Table 3.3. The excitations to the simulation is a rotor speed reference of 1 *p.u.* and a varying resistive load on the stator. Graph (a) shows the rotor speed dynamics of both the FPGA calculations and the Simulink. The reason the FPGA results begin in steady state is because the simulation had been running for some time before the recording of results had begun, whereas the Simulink simulation was run from initial startup. Graph (b) shows the resulting output electrical power of the generator system obtained form the Simulink experimentation where it can be seen that the rated power is obtained with a resistive load of 125Ω. At this rated power, the speed obtained on the rotor is just below the no-load rated speed as expected. When the resistive load is decreased meaning that an increased power load is applied, the rotor speed drops and the electrical power output increases. When the resistive load is increased meaning a decreased power load, the rotor speed accelerates and the output shaft power decreases. The slight observance of error at the higher rotor speed is due to rounding of variables in the FPGA in the fixed point format. A solution to this problem is to allow a higher number of fractional bits to each variable which in turn increases the silicon usage. Therefore, to mitigate the error completely, a larger FPGA with more logic elements can be implemented to allow for increasing of fractional bits.

TABLE 3.3
Parameters of MTG [21]

MTG Rating	$P_b = 400\,kW, \quad \omega_{r_b} = 70{,}000\,RPM$
Speed Controller	K = 25, T_1 = 0.4, T_2 = 1.0, Z = 3
Fuel Controller	K_v = 1, T_v = 0.05, c = 1, K_3 = 0.77, K_6 = 0.23, K_f = 1, T = 0, T_f = 0.04
Compressor-turbine Combination	T_{CR} = 0.01, T_{TD} = 0.04, T_{CD} = 0.2, K_{HHV} = 1.2
Acceleration Control	K_{ia} = 100, Acceleration Limit = 1
PMG	r_s = 12.5 mΩ, $L_d = L_q$ = 0.000165 H, φ = 0.2388 V.S, P = 8, J = 0.011 kg.m^2

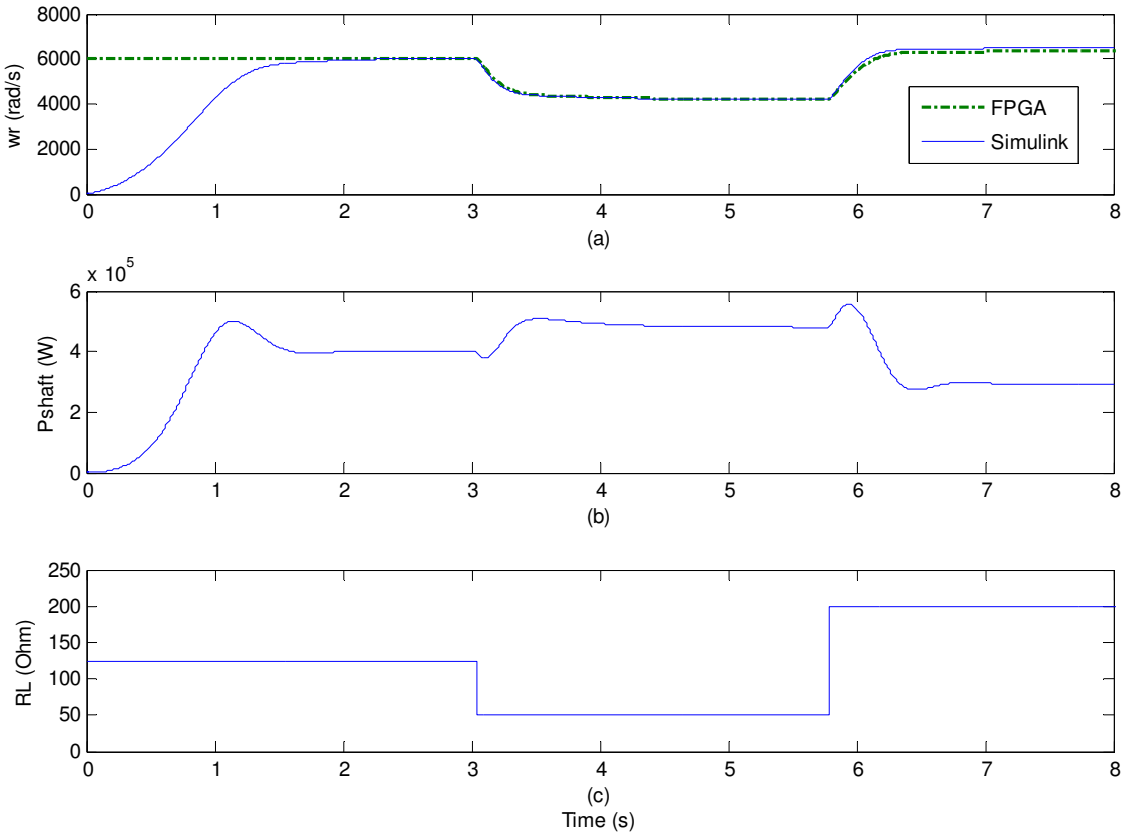


Figure 3.13. MTG real-time simulation verification results. (a) rotor speed comparison of the FPGA computed dynamics and the Simulink computed dynamics. (b) Electrical power output as computed by the Simulink simulation (c) Adjustable resistive load on the stator controlled in real-time through the user interface on dSPACE Control Desk.

CHAPTER FOUR

HIL EXPERIMENTAL SET-UP AND TEST BED

4.1 Non-linear Dynamic Loads Test Bed

The Non-Linear Dynamic Loads (NLDL) Test Bed is being developed within the Energy Conversion and Integration research thrust at CAPS. A main focus of the test bed is to investigate the MVDC system of the Navy's AES research. The topology of the AES shown in [1] is heavily emphasized on a common MVDC rail for powering of all ship system components. The benefit of such a topology is that all generator units can be used at any time for any component of the ship system. Traditional power systems aboard ships would typically have the prime mover systems directly coupled to propulsion motors and would therefore be limited to only supplying power to propellers thereby requiring alternative generation units supplying electric power to the other ship systems. With this new developing power system, all available power can be used in any specific area of the ship system.

The NLDL Test Bed is a small scale mock up of this system and is shown in the diagram of Figure 4.1. As can be seen, the input power from the grid is first introduced to the active front end – inverter (AFE-INV) converter combination that has been discussed in Chapter 1. The AFE is simply used as a diode rectifier in this experiment. The purpose of the emulation in this thesis is to present a method in which the inverter supplying the three-phase AC power to the neutral point clamp (NPC) converter is controlled to reflect the generator system dynamics as shown in the dotted line box represented the simulated component. It is from this point that the three-phase AC enters the NPC converter and power is rectified to a two rail DC system with a neutral point. There are many advantages to this type of converter such as the choice of two different DC voltage levels when connecting a device from rail to rail, or rail to neutral. The other advantages to this type of converter are control oriented in which the converter can be controlled as a PFC device. This means that the switches are controlled in such a manner that the current draw of the device is in phase with the voltage supply leading to only a real power draw. This type of control leads to a more efficient design with the elimination of complex power flow and also allows the NPC to be characterized as a variable resistor as seen

from the AFE-INV standpoint.

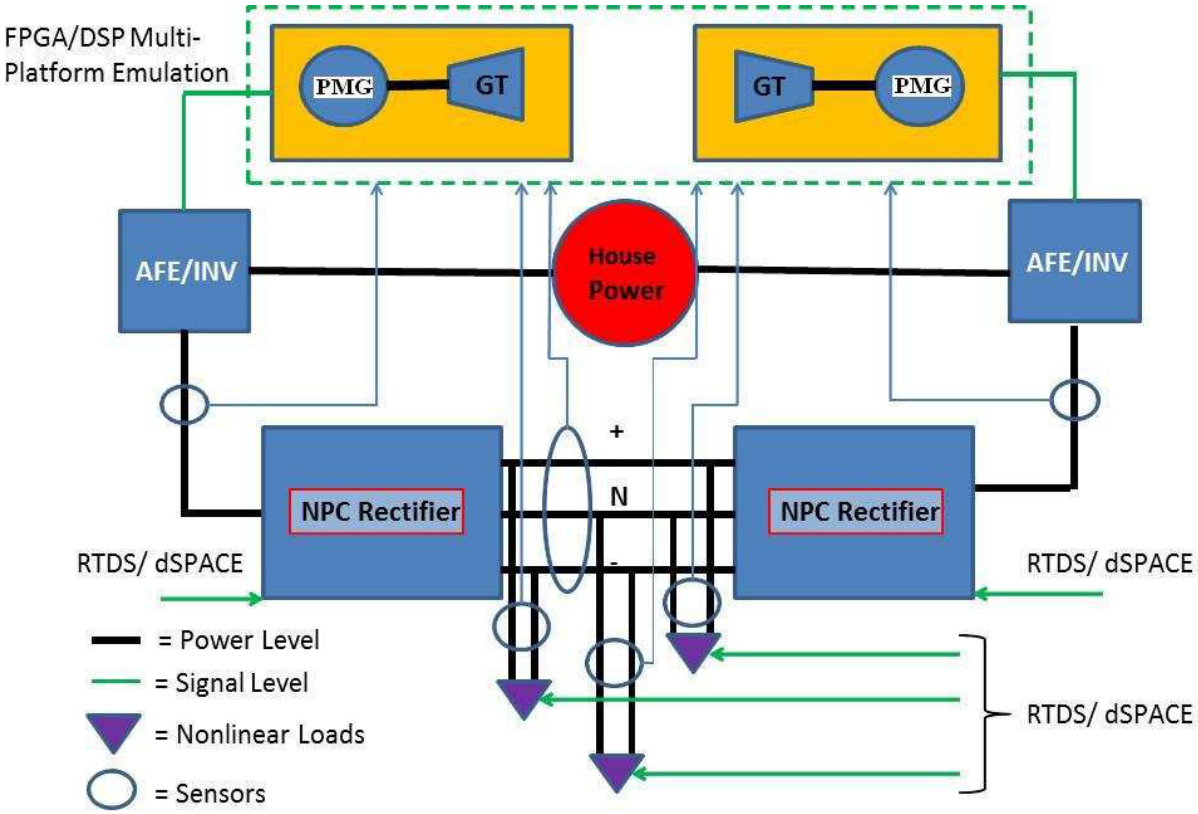


Figure 4.1. Diagram of NLDL Test Bed

After the power is rectified to DC via the NPC, various load banks or converters for electric machine drives can be attached to the DC rail to test the system under loaded conditions. Other HIL methods of load simulation may also be implemented via RTDS or dSPACE. This sort of load testing is of interest to the Navy for testing of the stability of such a power system and for introduction of new control methods that can be tested on a real hardware setup. With the introduction of generator emulation, the full effects of the switching of converters as well as various forms of power loads can be tested and the performance of the generator units can be studied without the actual need of a physical generator unit working within the test bed. This is the driving concept and ultimate goal for generation emulation with the EC&I Thrust at CAPS. The physical NLDL Test Bed is depicted in Figure 4.2. As can be seen, there are two identical cabinets holding the components shown in Figure 4.1. This gives the possibility of testing two

generator systems supplying power to a shared DC bus. This topology is consistent with the MVDC structure shown in [1] without use of any auxiliary generators.

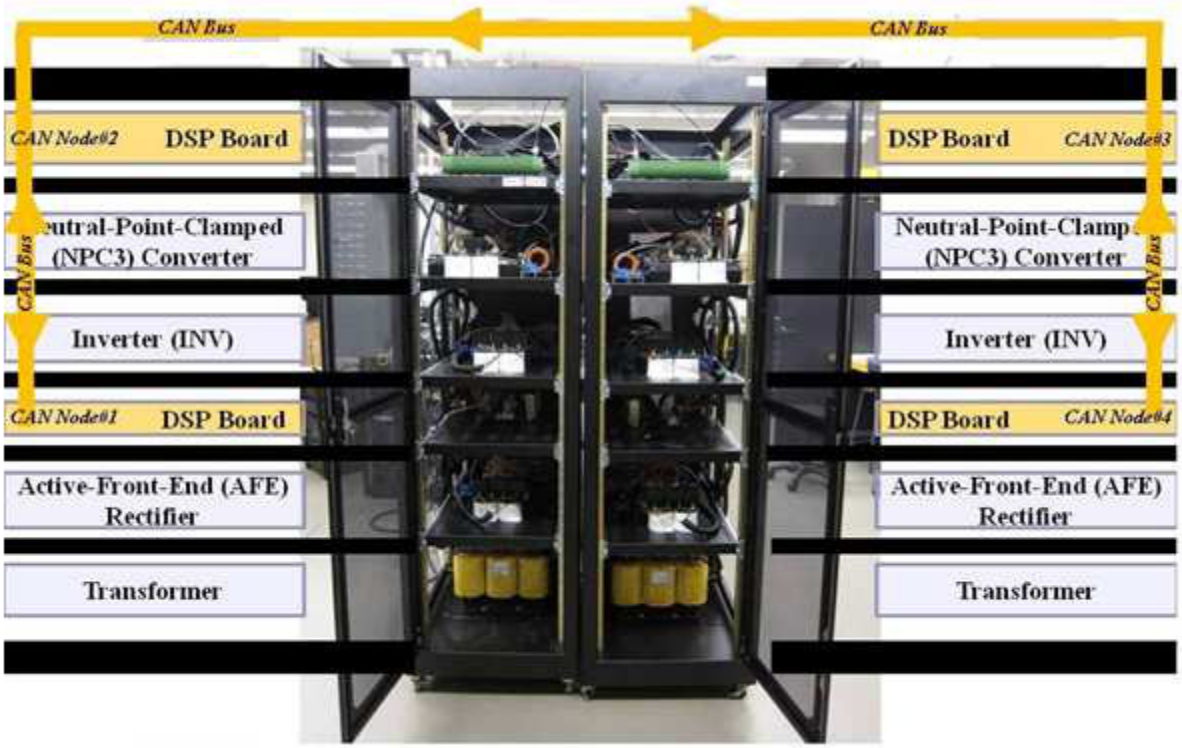


Figure 4.2. Physical picture of NLDL Test Bed with description of shelf contents.

4.2 HIL Experiment

The focus of this experiment is to emulate the generation dynamics by controlling the terminal characteristics of the AFE-INV which is then attached to the NPC converter with PFC controls for DC rectification. In this experiment, the NPC will be replaced by a wye-connected resistive load and the DC rail will not be utilized for simplification of initial tests. This is acceptable at this point in testing due to the previously described PFC characteristics of the NPC converter resulting in a variable resistor performance. This experiment also only considers one generator and does not study the effects of multiple generators coupled via the DC bus. The HIL topology shown in Figure 4.3 is developed for the real-time simulation, data acquisition and the

HIL application for generator emulation via the three phase inverter. The overall structure involves the FPGA DE2 board for real-time computation of the machine dynamics while a DSC performs PWM generation and measurement sampling, interfaces with a control terminal (PC), and acts as an interface to real hardware through voltage/current transducers with ADCs and PWM signaling.

As previously discussed, the DE2 board is used in calculating the dynamics of the MTG system. By looking at equations (2.9) and (2.10), it is observed that the equations for the PMG utilize a voltage in, current out orientation. This is because when looking at (2.1), it can be seen that the stator voltages are a function of the derivative of the flux linkage. Since computing the derivative may sometimes be unstable and yield undefined results in times of step changes, integrals are often preferred when solving ODEs. Due to this, the voltage in, current out orientation is used and the current is solved for using the stator voltage as an input variable. In (2.9) and (2.10), this stator voltage is redefined in terms of current and resistance across the stator windings via Ohm's Law. Since the inverter is controlled as a voltage source device, the calculated current is then multiplied by the stator resistance to obtain the stator voltage of the new time-step that must be realized on the output of the inverter. This load resistance is measured in the test bed and therefore allows for a dynamically changing value which is observed when directly coupled with a PFC device under varying output loads. However, due to this orientation and requirement of the resistive load when solving for stator voltage, the resistive load used in this calculation is lagging by one time step. This one time-step delay is investigated and determined that since the time step is small and only 20 μs that there is no significant error or observable effect in the HIL dynamics. However, in the future when implementing a significant and rapidly changing load resistance, this effect may need further investigation.

The DE2 board will accept inputs of speed reference in *p.u.* and load resistance R_L in terms of ohms. The DE2 board will also output rotor speed in *rad/s* and stator voltage, v_q and v_d in voltage. The variables computed within the FPGA are performed using fixed-point notation but is converted to 32-bit floating point for use in the DSC. The FPGA will communicate to a TI TMS320F28335 DSC through SPI. The FPGA is configured as the master device, while the DSC operates as the slave. The FPGA will initiate transmit of data every 20 μs. By enabling the SPI communication every 20 μs, the three 32-bit floating point variables (v_q , v_d and ω_r) have enough time to be transmitted at a SPI clock speed of 12.5 Mhz. The time period also gives

enough time for the DSC to run the PWM and measurement algorithms. Figure 4.4 shows a diagram of the SPI communication interface. The DSC will communicate to the PC running the dSPACE real-time environment through RS-232.

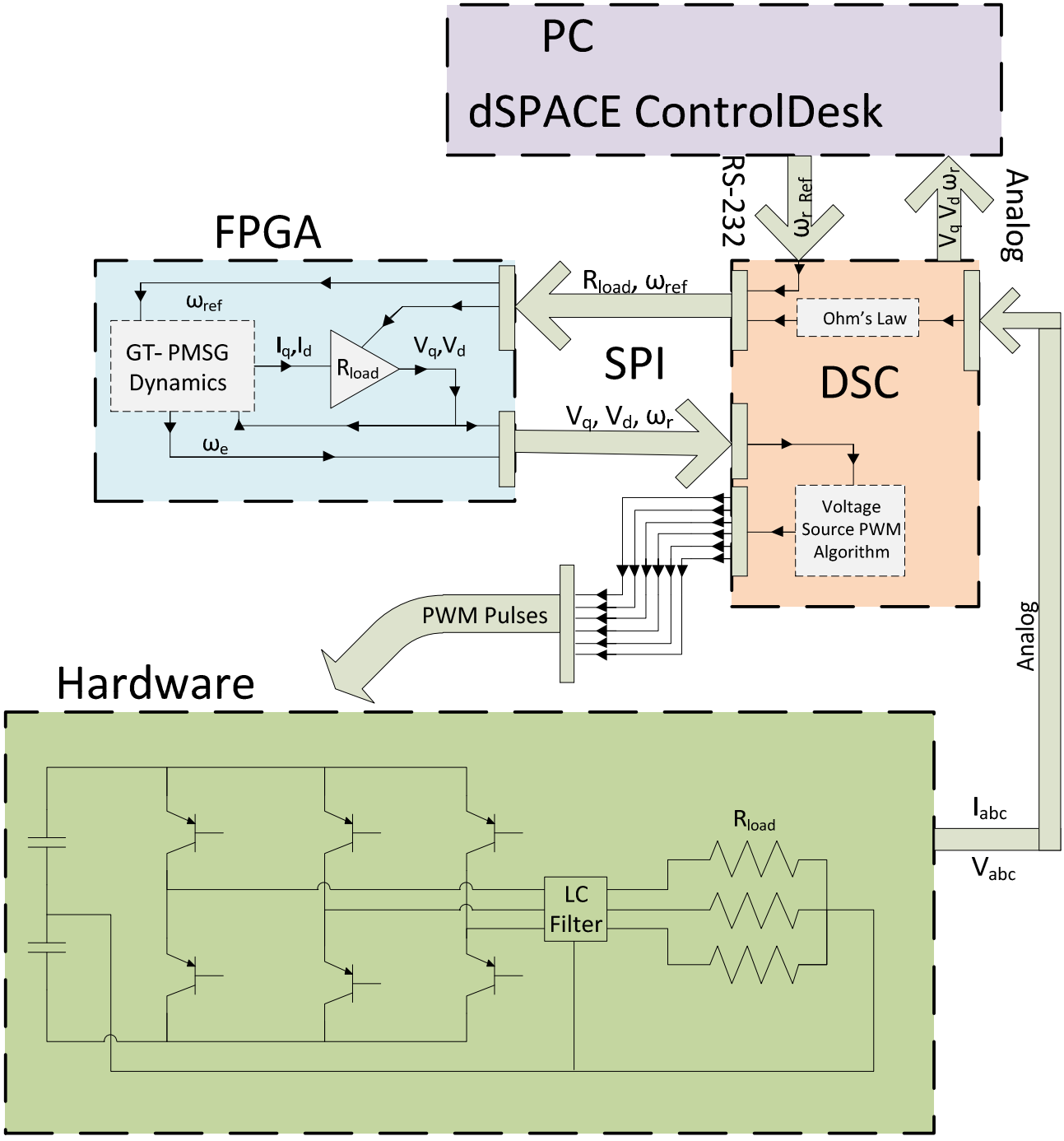


Figure 4.3. HIL structure of the multi-platform computation system with hardware interface.

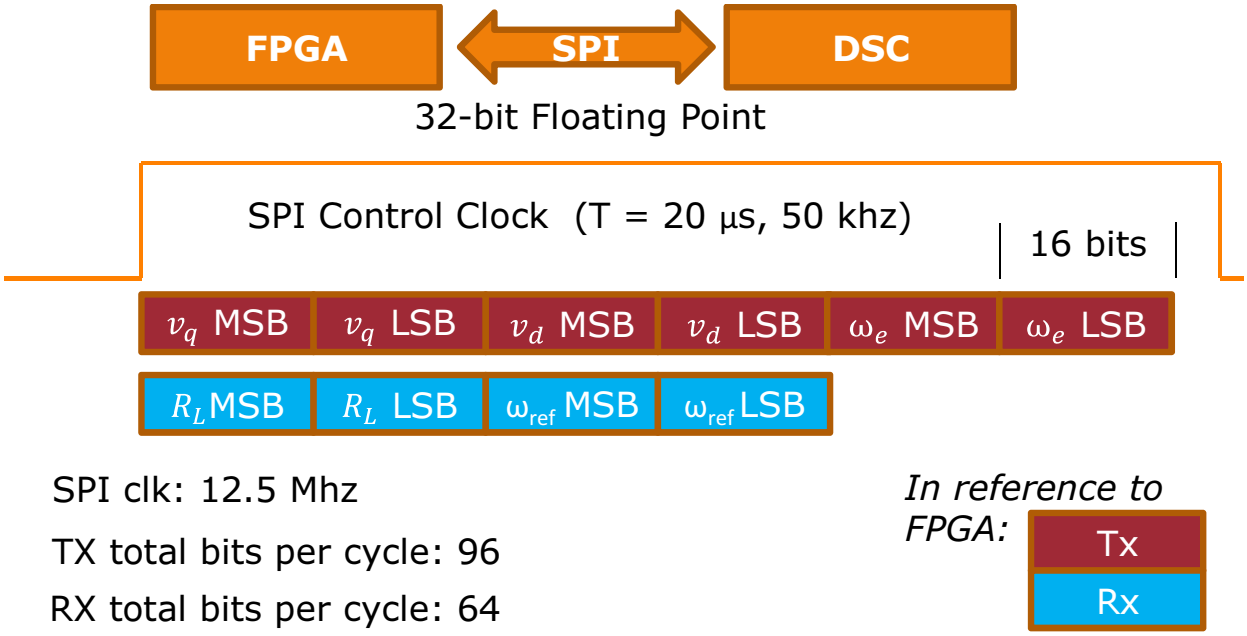


Figure 4.4. FPGA and DSC SPI communication interface.

4.2.1 Structure of Code and Role of the DSC

As can be seen, the FPGA is used to compute the MTG dynamics in real-time and transmit data through SPI with the DSC. The objective of the DSC is to interface with the three phase inverter. The DSC will calculate the voltage references using RFT based on the *qd0* voltage values and angular electrical speed ω_e sent from the FPGA [16]. This would be using the inverse transformation to obtain the three-phase *abc* variables from the calculated *qd0* equations using (2.4). It will then use the references and the voltage source PWM algorithm to transmit the pulses to the six IGBTs via 6 PWM pins. This method effectively commands the inverter to output voltage amplitude and frequency based on the FPGA dynamically calculated values. The next objective of the DSC is to sense the loading characteristics of the system in terms of three phase voltage and current. This will allow the resistance on the output of the inverter to be determined via Ohm's Law and then transmitted to the FPGA to be included in the calculation of dynamics. The readings of voltage and current are first converted to root mean square (RMS) values using (4.1) and (4.2) respectively before calculating the resistance using Ohm's Law in (4.3).

$$V_{rms} = \sqrt{\frac{1}{n}(V_1^2 + V_2^2 + \cdots + V_n^2} \tag{4.1}$$

$$I_{rms} = \sqrt{\frac{1}{n}(I_1^2 + I_2^2 + \cdots + I_n^2} \tag{4.2}$$

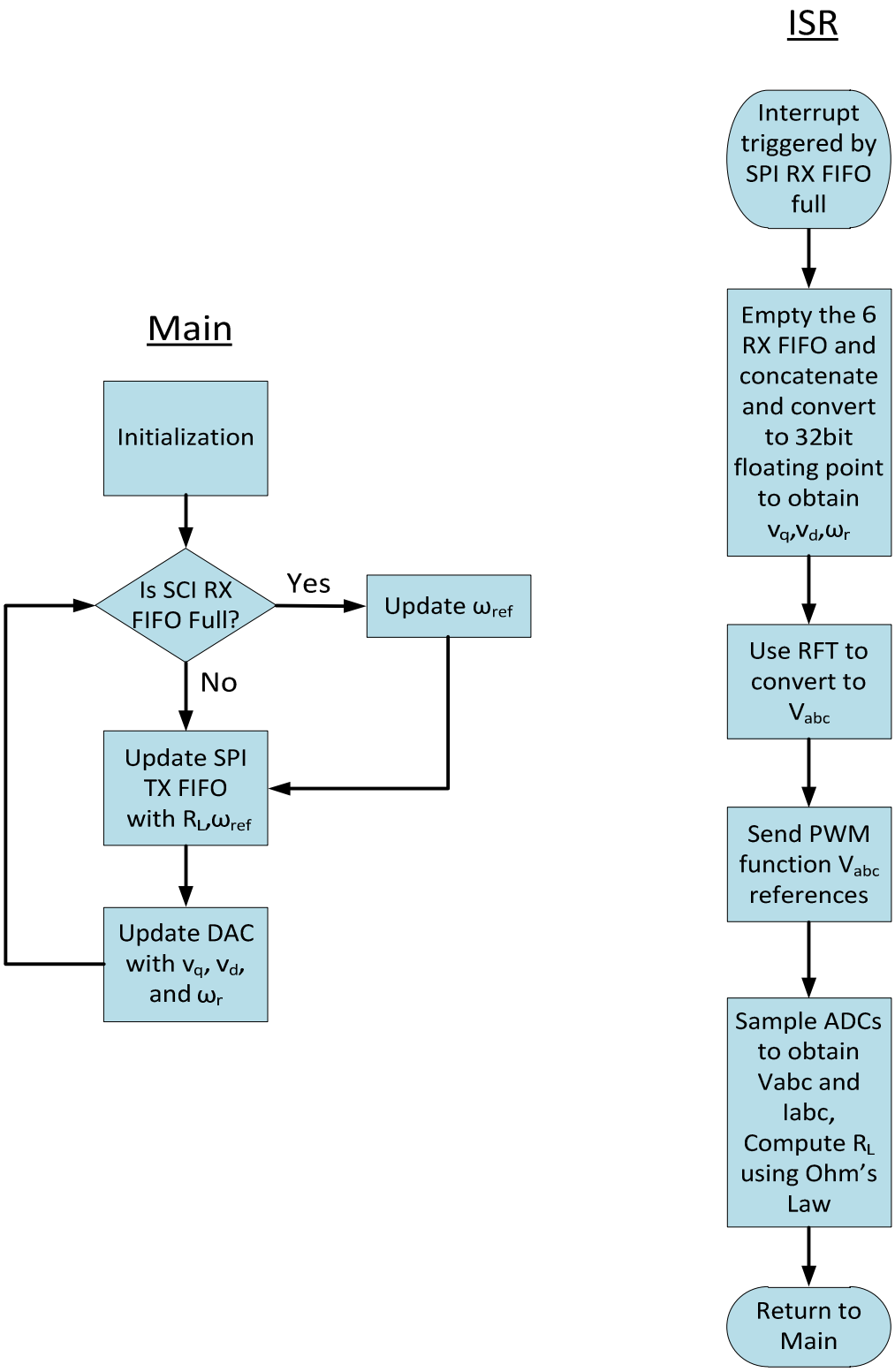
$$R = \frac{V_{rms}}{I_{rms}} \tag{4.3}$$

The DSC uses 500 numbers of samples (n = 500) to compute the RMS values which is chosen to make sure at least several periods of the waveform will be sampled in calculating the RMS. The DSC then has one final objective which will be to accept the speed reference from the user interface PC and send this to the FPGA to include in dynamic calculations. The code implemented onto the DSC can be summarizing using the pseudo-code flow chart shown in Figure 4.5.

In this figure it is seen that in the main function, the pointer will continually loop until the interrupt is called signaling received data via SPI from the FPGA. In the main loop, the algorithm will check to see if data has entered into the SCI RX first-in-first-out (FIFO) register which would mean that data has been sent through the RS-232 connection from the dSPACE unit. In the event that there is data in the FIFO, the ω_{ref} variable will be updated with this value in the DSC and the FIFO will then clear. Next, the SPI TX FIFO will be updated with the R_L and ω_{ref} variables which are the variables that will be sent from the DSC to the FPGA upon transmission initiated by the FPGA. Next the digital to analog converter (DAC) is updated with the variables v_q, v_d and ω_r to be sent to the analog input of the dSPACE unit for data acquisition. v_q, v_d, R_L and ω_r are calculated in the interrupt service routine (ISR) as follows.

The ISR is initiated when the SPI RX FIFO has received all the expected bits from the FPGA according to Figure 4.4. This means that the variables v_q, v_d and ω_r have been sent from the FPGA and received by the DSC and the first step for the DSC is to store these values as variables. Next, since the variables had to be converted to 32-bit floating point format and sent in sections of 16 bits, the values need to be properly concatenated and casted into floating point 32-bit format so that the DSC can properly perform the required computations. Next, the DSC performs the inverse reference frame transformation to obtain the three phase *abc* values using

the angular position, θ_e, derived by using Euler's formula of the electrical frequency ω_e, calculated by using (2.12). In using Euler's formula to calculate θ_e, a computational time-step of 20 µs is used because the interrupt is called every 20 µs since that is how often the FPGA initializes transmit of data. Once the *abc* variables of voltage is calculated these values are sent into the PWM function which generates the pulses and duty cycles to send from the PWM outputs into the gate drivers of the IGBTs. Next, the ADC's of each phase of voltage and current are sampled and R_L is computed using Ohm's Law and the RMS values as previously described.



(a) (b)

Figure 4.5. DSC pseudo-code. (a) Main function. (b) Interrupt Service Routine.

CHAPTER FIVE

FULL MTG HIL EXPERIMENTAL RESULTS

The results produced in this Chapter use the HIL experiment set-up on the NLDL Test Bed as described in Chapter 4. The parameters of the MTG system were used from [21] and shown in Table 3.3. However, in this experiment the pole count on the PMG was reduced to 4 poles to reduce the electrical frequency and voltage observed on the stator terminals. The purpose of the results of this HIL emulation is to demonstrate the ability to control the three phase inverter output to reflect the dynamically calculated stator voltage of a specific MTG system. The results will compare real data computed in the FPGA dynamics with consideration of a real hardware load and compare it to a Simulink simulation of the same system as previously performed. The verification experiment is shown in Figure 5.1. As can be seen, an MTG system with a rotor speed reference input and constant resistive load on the stator terminals will be simulated in Simulink while the same MTG system using the same parameters will be emulated using the HIL interface in connection with the real resistive load. In order for the HIL emulation experiment to be verified, the FPGA calculated variables as well as the real voltage and frequency output of the inverter must match the Simulink results of the same MTG system. It is important to note that the Simulink simulation was unable to run in real time using dSPACE due to the mathematical complexity and the step changes in speed reference were matches to what was performed during the real-time HIL emulation. This aspect further demonstrates the necessity and capabilities of FPGA implementation in real-time simulation experiments. The experiment results will analyze the real voltage measurements on the output of the inverter and compare the voltage and frequency to the simulated expected values under changes in speed commands directed from the user interface via dSPACE. This experiment will aim to prove the method in which the generator emulation is performed and show that such an operation can be accurately executed and correspond accordingly to expected results.

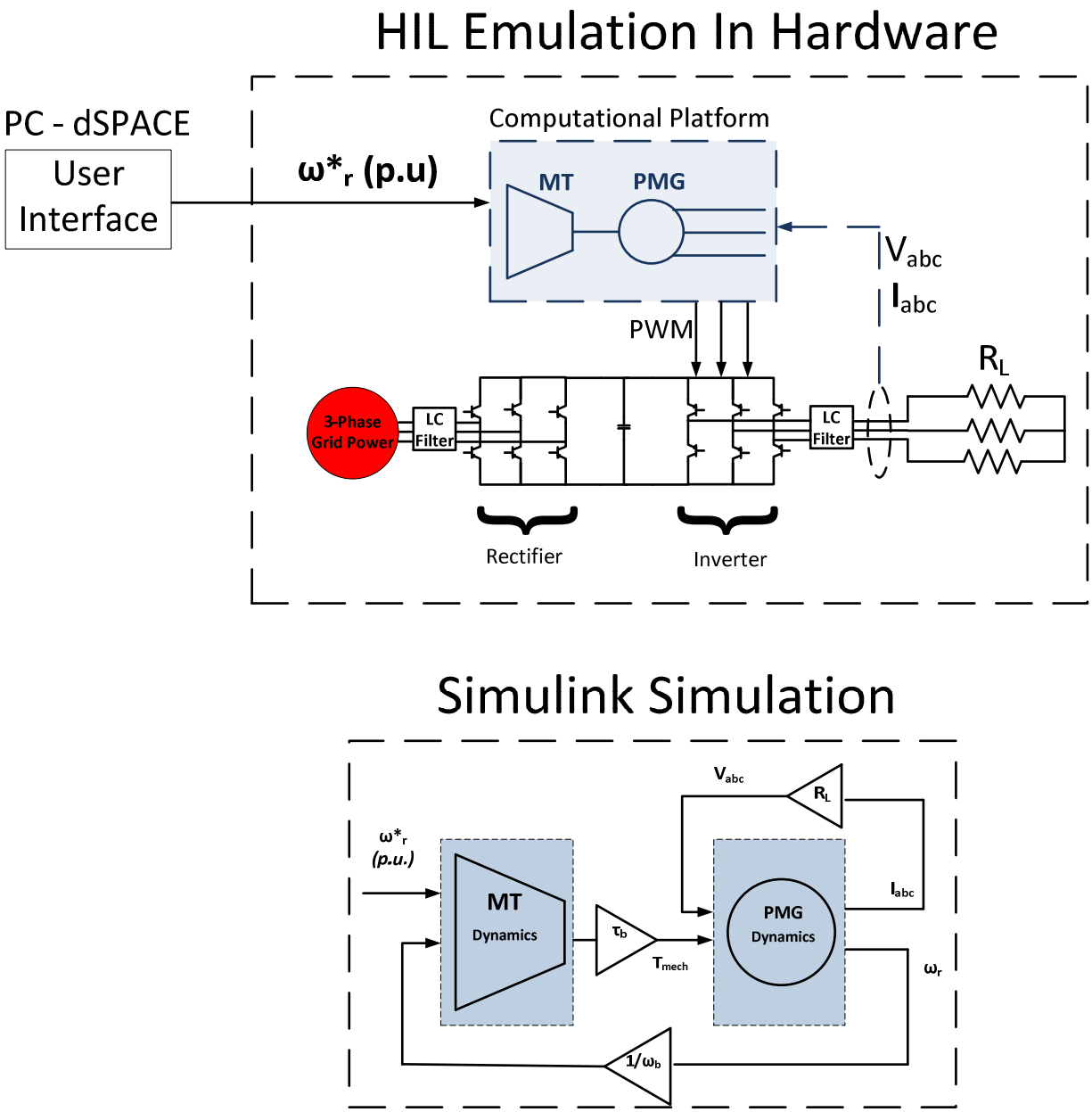


Figure 5.1. HIL emulation verification against Simulink simulation experimental setup.

5.1 Limitations and Scaling of Values

Due to limitations of the power hardware in the test bed, the real voltages and power levels performed in the simulation are not obtainable and must be realized at a scaled down level. The parameters of the generator system being emulated corresponds to a 400kW at 70,000 RPM rated MTG, therefore, the realized voltage on the output of the inverter is scaled down by a factor of 0.01V. This factor is adjustable and was chosen to realize the voltage levels below the maximum voltage that the inverter could produce. This maximum voltage is dependent on the duty cycle and DC voltage. Therefore, in order to increase the maximum achievable voltage on the output of the inverter, one must eventually increase the DC voltage.

In this experiment, the voltage on the DC link is produced by the AFE converter without the use of voltage controls and simply used as a diode rectifier. Because of this, the voltage on the DC link is strictly dependent upon the input three phase AC voltage supplied by the transformer of the input power to the cabinet. It will be discussed in the next chapter how controls in the AFE can increase the voltage on the DC link and thus allow for a wider range of obtainable emulated voltage on the output of the inverter. For the purpose of this proof of concept experiment, the amplitude of the emulated voltage was not important so long as the scaled value corresponded appropriately to the expected value in simulation. With a scaling factor of 0.01V, this means that at every instant, the measured voltage must be multiplied by a factor of 100 to obtain the actual voltage of the emulated MTG system.

Additionally, the rotor speed command was limited to a range of 0.4 *p.u.* to 0.6 *p.u* due to the obtainable switching frequency of the IGBTs of the inverter. Because the DSC only receives new voltage commands every 20 μs, a maximum switching frequency for the IGBTs is 50 kHz. With a switching frequency of 50 kHz, the realized output sinusoids are limited to a maximum frequency of around 1.5 kHz without significant distortion. Since it was experimentally discovered that at rated load conditions with a speed reference of 1 *p.u.*, the electrical frequency is close to 2 kHz, it was decided to reduce the commanded speed to a range of 0.4 *p.u.* to 0.6 *p.u* to produce resulting frequencies closer to 1 kHz on the real hardware. This is another system limitation that will be explored in Chapter 6 and can be increased in future studies.

One final scaling operation is that the real resistance used in the experiment was multiplied by a factor of 5 before being used in the dynamic equations. This was because the resistors available on hand were 20Ω resistors which would reflect a heavily loaded MT system if used as

is, thus reducing the observed rotor and electrical frequency. In scaling the three phase resistance up to a value of 100Ω for the dynamic calculations, higher frequency and a more closely related output to rated values are observed. This has no effect in proving that the obtained voltage and frequency match the expected values, it only effects verifying power measurements which is not included in this experimental verification and would require using this resistance scale to match the simulated results.

With the experimental setup as previously explained, Table 5.1 shows the suggested maximum realizable excitation parameters for generator emulation possible on the NLDL Test Bed without implementing the possible improvements discussed in Chapter 6.

TABLE 5.1

Maximum Realizable Excitation of Inverter Output

Excitation Parameter	**Maximum Value**
Voltage amplitude	30V
Fundamental frequency of inverter output	~1.5 kHz
Switching Frequency	50 kHz

5.2 Results

The waveforms shown in Figure 5.2 is the output data from the FPGA over time which include v_d, v_q and ω_r. This data is compared to the Simulink and can be seen to be calculating the same results with minimal error. This calculated data from the FPGA is the result of the dynamic equations as well as the measured resistance of 20Ω multiplied by the resistive base value of 5, with a speed reference of 0.5 $p.u$. In Figure 5.2(a), the v_d alteration is almost zero with respect to output voltage value of ~1700V and the difference between simulation and emulation results are negligible concerning the precision limitation. This error also has to do with the resolution of the digital to analog converter (DAC) output of the DSC. From these results, it is evident that the computational platform is sampling the voltage and current, calculating the load resistance, and computing the generator dynamics in terms of stator voltage and rotor speed correctly and in real-time. This data is compared to the Simulink simulation and

can be observed to be calculating the correct dynamics with reasonable error. These values are then sent on to the DSC to compute the *abc* references to be realized on the inverter. The importance of verifying these values is evident.

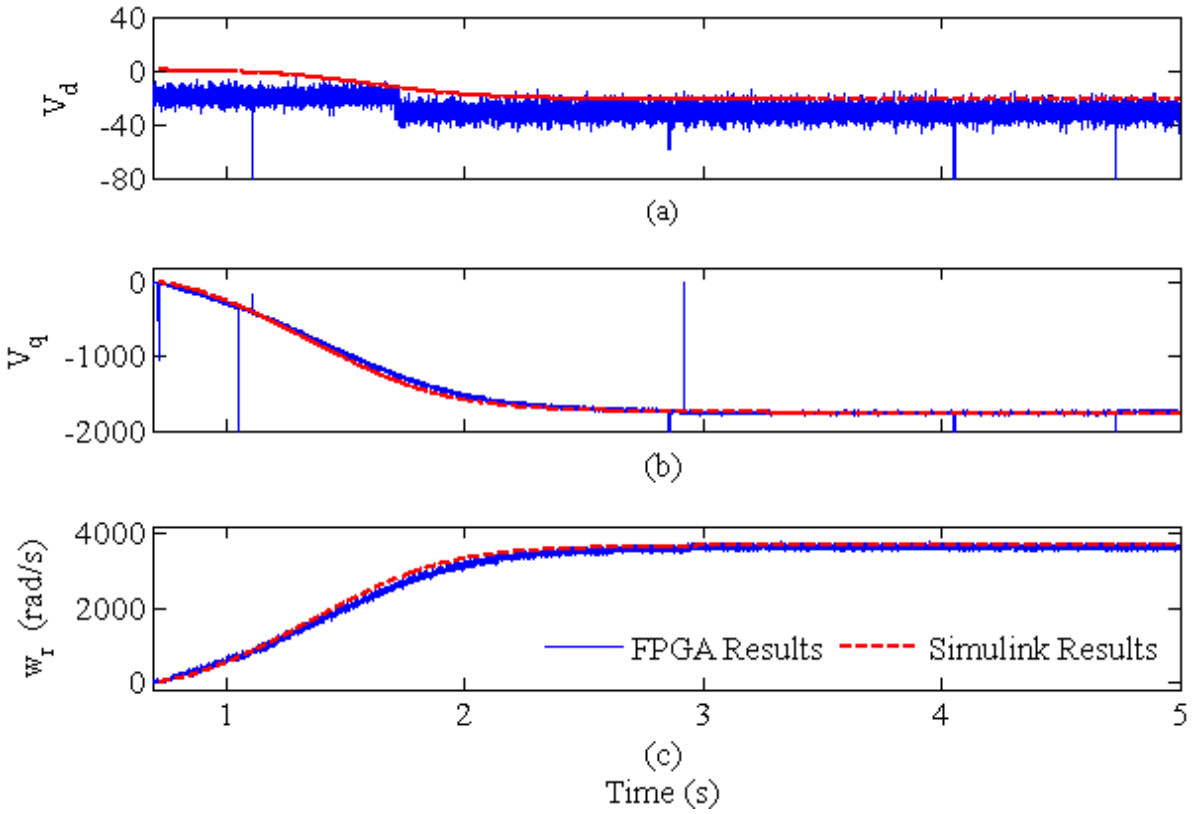


Figure 5.2. Start-up transients of HIL FPGA verification experiment at 0.5 *p.u.* speed reference. (a) Comparison d-axis voltage of HIL Emulation with Simulink simulation. (b) Comparison q-axis voltage of HIL Emulation with Simulink simulation. (c) Comparison rotor speed of HIL Emulation with Simulink simulation.

The second test in Figure 5.3 shows the same system under changes of speed reference. These changes are sent from the user interface of the PC terminal running dSPACE to the DSC which then forwards on to the FPGA via SPI communication. It is important to note how the FPGA calculated dynamics match the expected values from the Simulink simulation with the same changes of speed reference at the same times. Again, the data shown is the output of the FPGA consisting of v_d, v_q and ω_r and compares it to a simulation running in Simulink under the same parameters and excitations. This proves that the calculation is working in real-time through the computational platform and can respond to input commands of speed reference and the sensed resistive load.

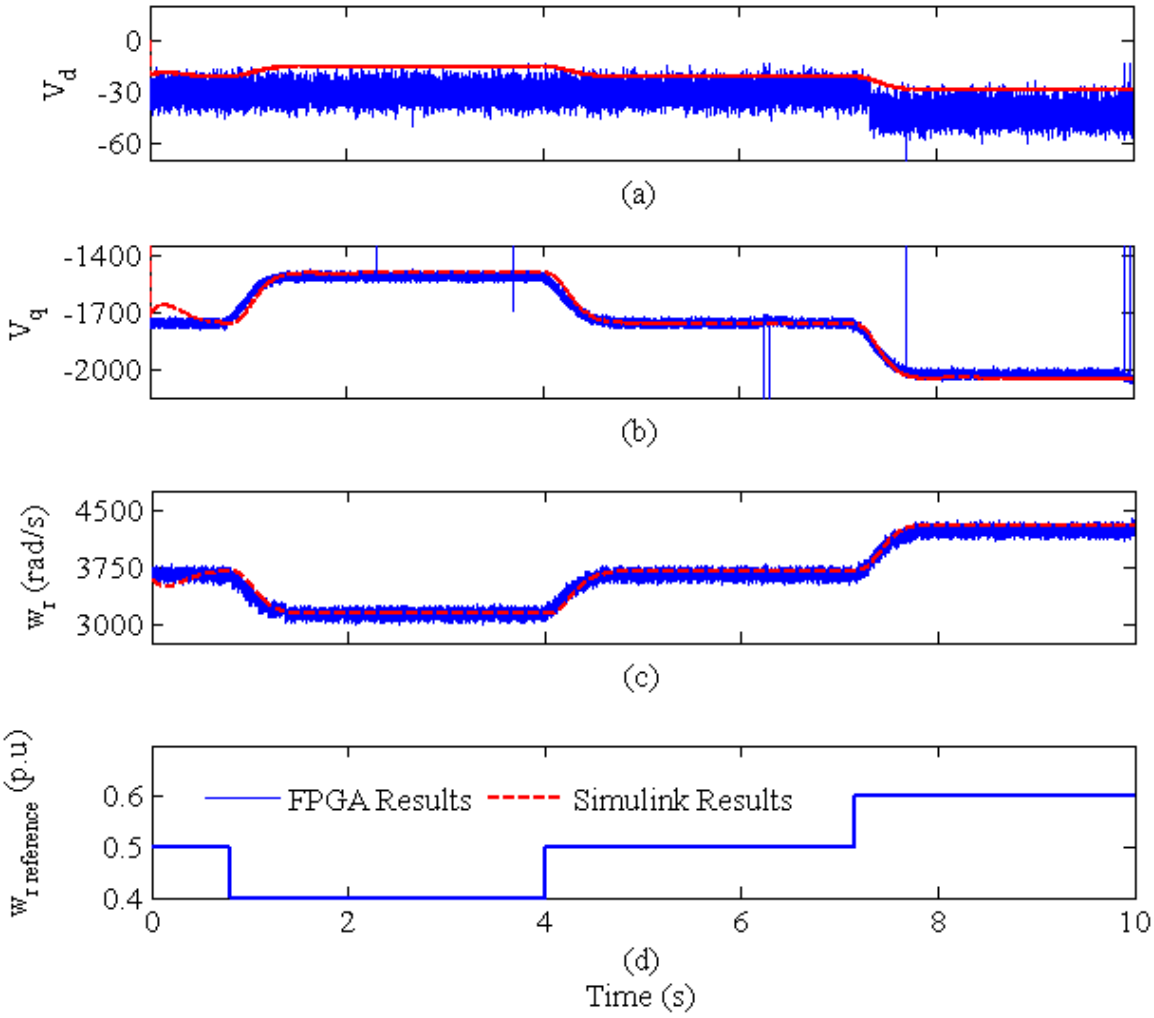
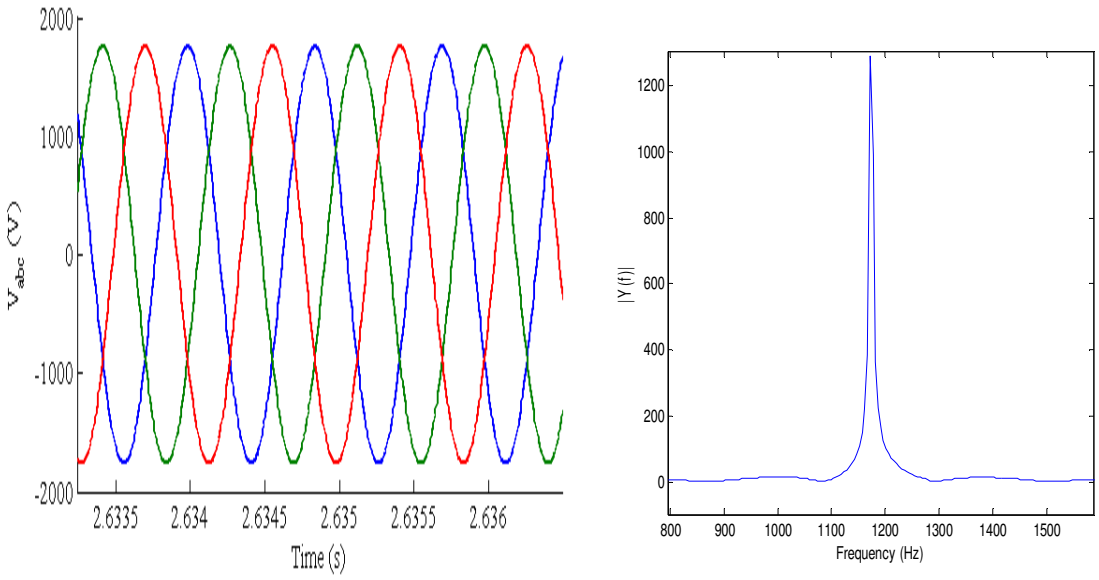


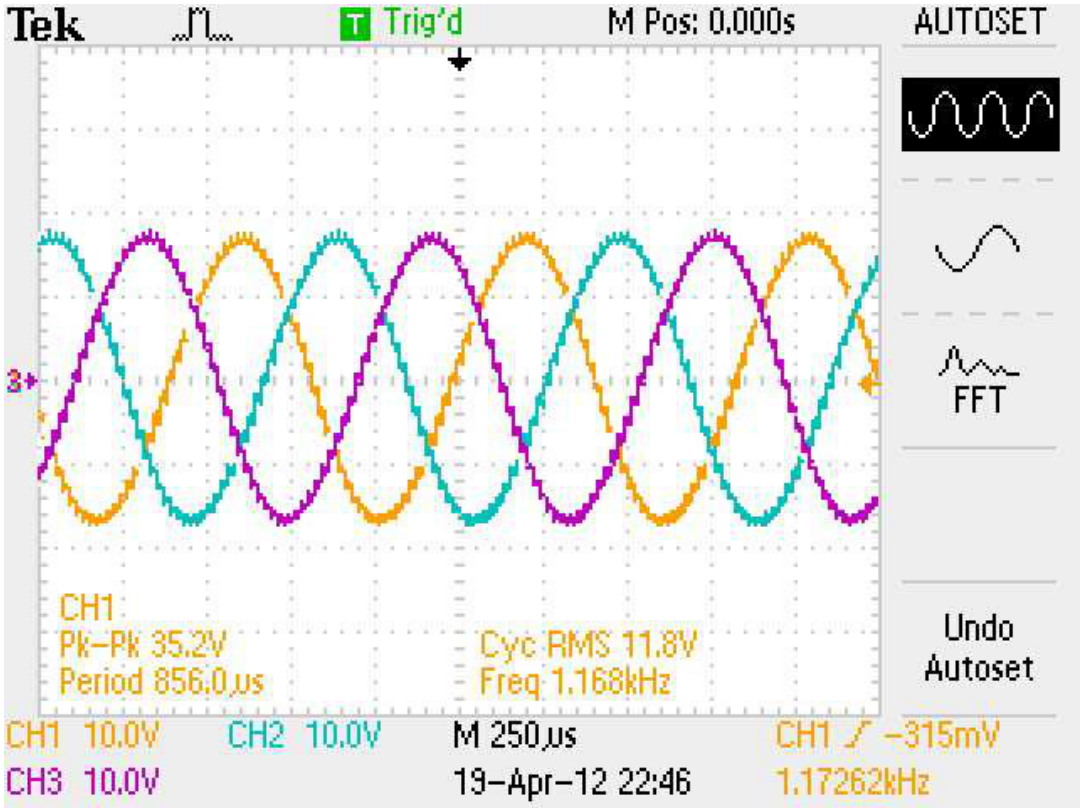
Figure 5.3. HIL FPGA verification experiment under various speed references. (a) Comparison d-axis voltage of HIL Emulation with Simulink simulation. (b) Comparison q-axis voltage of HIL Emulation with Simulink simulation. (c) Comparison rotor speed of HIL Emulation with Simulink simulation. (d) Rotor speed reference input from user interface via dSPACE.

Thus far, the dynamic calculations and therefore voltage references sent from the FPGA dynamics are validated and are now sent for the inverter to be emulated. Figure 5.4 (a) shows the Simulink *abc* stator voltage resulting in the steady state condition of 0.5 *p.u.* speed reference and 100Ω resistive load on the stator. As can be seen, the voltage amplitude is less than the rated 3,000V since the commanded speed is half the rated value. Also, it can be noticed that the electrical frequency shown in the fast Fourier transform (FFT) of Figure 5.4 (b) is again less than the rated value due the slower command of the rotor speed. Figure 5.4 (c) shows the output

voltage via an oscilloscope from the inverter of this same condition during the HIL experiment. It can be noted that the peak-to-peak voltage of the oscilloscope reading matches the expected results of the Simulink simulation when multiplied by the base value of 100V. The frequency can also be verified between the two results of ~1.2 kHz by comparison of the FFT of the simulated voltage shown in Figure 5.4 (b) and measured frequency from the oscilloscope in Figure 5.4 (c). Figure 5.5 show the same steady state results of the same scenario previously stated except uses a reference speed of 0.6 *p.u*. With the fast speed reference, it can be noticed that the voltage output of the stator as well as electrical frequency is increased as expected. Again, the voltage of the oscilloscope can be verified to match the expected Simulink results multiplied by the scaling factor. The frequency can also be verified.

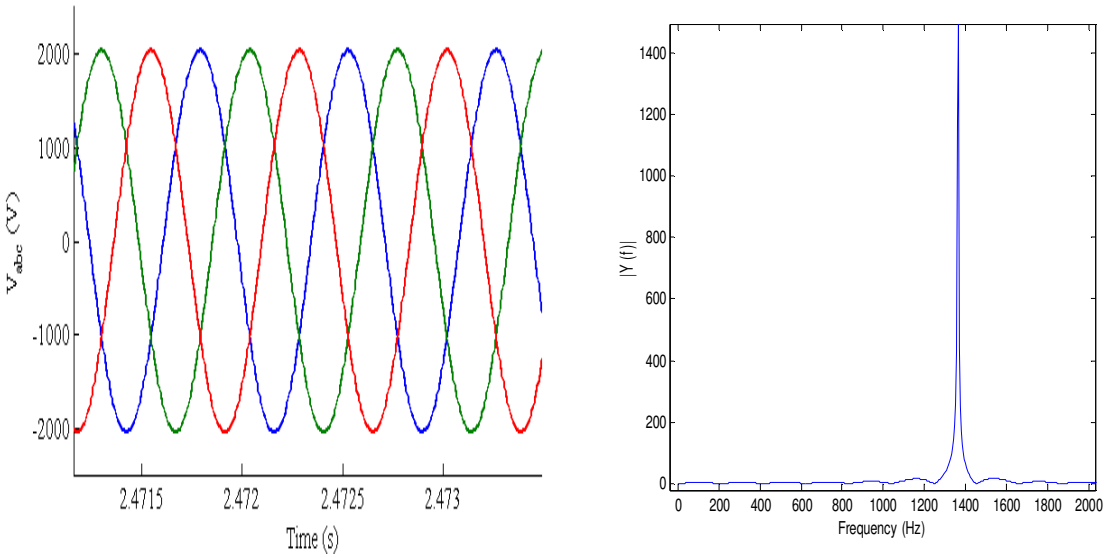


(a) (b)

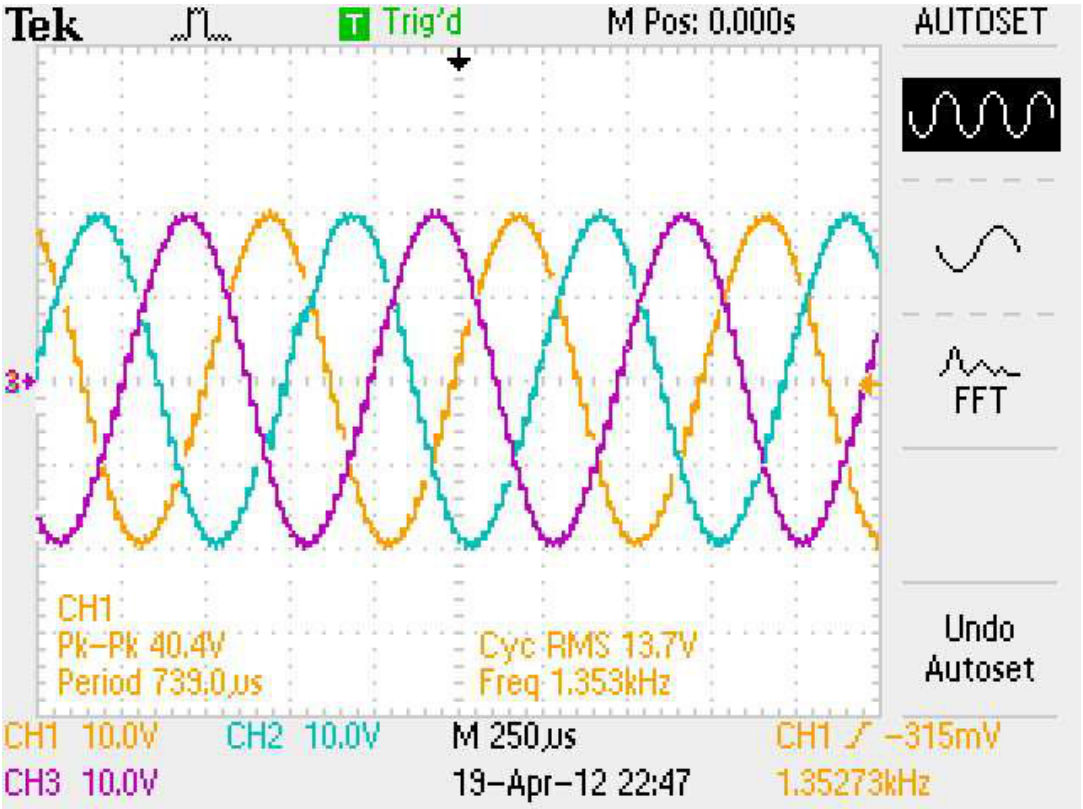


(c)

Figure 5.4. Simulated steady-state voltage and HIL hardware emulated voltage comparison. (a) Simulated steady-state voltage waveform with 0.5 *p.u.* speed reference. (b) FFT of steady state simulated voltage. (c) emulated real voltage measured at the output of the inverter



(a) (b)



(c)

Figure 5.5. Simulated steady-state voltage and HIL hardware emulated voltage comparison. (a) Simulated steady-state voltage waveform with 0.6 *p.u.* speed reference. (b) FFT of steady state simulated voltage. (c) emulated real voltage measured at the output of the inverter.

CHAPTER SIX

CONCLUSION AND FUTURE WORK

6.1 Conclusion

As can be seen, high speed generator emulation is obtainable and a very usable tool for system studies. This sort of HIL simulation will prove to be especially important in studies of the Navy's AES MVDC topology and for various distributed generation networks. In using the generator system emulation strategy discussed in this thesis, various studies of distributed systems can be performed in many different areas ranging from converter control to system stability on real hardware without the need of a physical generator system. The advantages of using the generator emulation approach rather than including a real generator system in testing is the large cost savings and also safety of equipment and personnel. With generator emulation, many more scenarios can be tested that push the limit of what can be achieved without devastating consequences. Additionally, when emulating large generator units in the range of MW as will be utilized in the AES, small scale testing cannot be performed at such high power levels. In this case, generator emulation introduces the ability to scale values to safe power levels while still studying the dynamic effects of the higher power characteristics. This thesis produces the following contributions to this new and developing field of testing and simulation.

- Design and implementation of a multi-platform digital system capable of performing the computationally intensive PMG and micro-turbine dynamics in real-time at a respectable 1μs time-step using both an FPGA and a DSC with a PC for data acquisition and real-time user interface.
- Implementation of a simulated high speed generator system MIL interface using control of a three phase inverter to emulate generator dynamics.
- Full HIL implementation by virtual realization of resistive load in the real-time simulating dynamics by active measurements of the real load in hardware.
- Verification of real time dynamic calculations of MTG system

- Verification of the HIL interface using the MIL generator system emulation with a real resistive load in hardware.
- Analysis of the results with options for improvement and further topics of study for future generator emulation implementations.

Generator system emulation is a very new and exciting practice in HIL system studies and introduces many opportunities for testing in future distributed grid topologies and control. The work described in this thesis has aimed to build a foundation and create a first stepping stone into the emerging practice of generator system emulation.

6.2 Future Studies

There is much room for development and growth in generator system emulation. Generator control strategies can be implemented to test system compatibility and topics such as load sharing and load response can be tested. Also, the limitations discussed in the previous chapter can be improved upon through several methods.

The limitation of the maximum voltage amplitude can be alleviated as follows. As previously explained, the maximum voltage obtainable on the output of the inverter is dependent upon the input DC link voltage. This voltage can be actively controlled via the AFE and was not implemented in this research. Future studies can incorporate AFE controls to increase the available DC voltage or to add complexity to the generator emulation in which this DC voltage could be linearly related or controlled by other system parameters such as output torque of the gas/micro turbine. However, although the DC voltage could be controlled by using AFE controls, this adds more required functionality of the DSC, increasing the required run-time, which may require an increase of the communication initiation time step to make sure the DSC has completed its' ISR.

The second limitation being the maximum fundamental frequency obtainable on the output of the inverter may be mitigated by increasing the switching frequency of the IGBTs. Currently, the switching frequency used in 50 kHz because that is the rate in which the new voltage references are received by the DSC. Using a faster switching frequency would be redundant because there would be no change of directed voltage. Therefore, in order to increase the

frequency, a communication initiation between the FPGA and DSC must be triggered at a faster rate. In this experiment, the reason an increase in communication initiation rate is not performed is because the SPI clock speed of 12.5 MHz cannot be increased on the FPGA. Therefore, the transfer of all 96 bits from the FPGA to the DSC cannot be increased and can therefore not reduce the 20µs initiation of communication time-step. However, this issue needs further exploration by implementing a different SPI core as the problem may not be a hardware issue, but rather a coding issue. If the FPGA processing of the SPI communication sequence can be performed at a faster rate than 12.5 MHz, the communication delay decreases and thus the communication can be initiated at a smaller time-step, allowing an increase in switching frequency to not be redundant. However, an increase in switching frequency reduces the efficiency of the IGBTs and may result in power losses that could result in unacceptable performance or damage, careful consideration needs to be practiced when increasing the IGBT switching frequency. With a higher switching frequency and therefore higher achievable fundamental component frequency, higher speed generator units can be studied such as distributed generator systems in the 50kW – 200kW in which rotor speeds by reach up to 100RPM. For the purpose of the Navy AES systems, such high frequency studies are not necessary due to the slower rotor speeds of the MW systems.

Finally, much of the communication speed and DSC run-time limitations can be solved completely by development of a full FPGA computation system. This system would use the FPGA for dynamic calculations as well as interface to measurements and dSPACE for data acquisition. Such a method would require development of a PCB for ADC and level-shifting of signals to and from the FPGA and IGBT drivers. The FPGA would also need to be large enough in terms of logic elements to fit the required amount of computations and interfaces. As can be seen, there are many options for further exploration and development into generator system emulation.

APPENDIX A

VOLTAGE EQUATIONS *ABC* TO *QD0* TRANSFORMATION DERIVATION [6]

Transformation of resistance term to *qd0* frame

$$\boldsymbol{v}_{abc} = \boldsymbol{r}_s \boldsymbol{i}_{abc}$$

$$\boldsymbol{v}_{qd0} = \boldsymbol{K}_s \boldsymbol{r}_s \boldsymbol{K}_s^{-1} \boldsymbol{i}_{qd0}$$

$$\boldsymbol{K}_s \boldsymbol{r}_s \boldsymbol{K}_s^{-1} = \boldsymbol{r}_s$$

Therefore,

$$\boldsymbol{v}_{qd0} = \boldsymbol{r}_s \boldsymbol{i}_{qd0}$$

Transformation of inductance term to *qd0* frame

$$\boldsymbol{v}_{abc} = \frac{d}{dt} \boldsymbol{\lambda}_{abc}$$

$$\boldsymbol{v}_{qd0} = \boldsymbol{K}_s \frac{d}{dt} [\boldsymbol{K}_s^{-1} \boldsymbol{\lambda}_{qd0}]$$

$$\boldsymbol{v}_{qd0} = \boldsymbol{\lambda}_{qd0} \boldsymbol{K}_s \frac{d}{dt} [K_s^{-1}] + \boldsymbol{K}_s \boldsymbol{K}_s^{-1} \frac{d}{dt} [\boldsymbol{\lambda}_{qd0}]$$

$$\boldsymbol{K}_s \frac{d}{dt} \boldsymbol{K}_s^{-1} = \omega \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{v}_{qd0} = \omega \boldsymbol{\lambda}_{dq} + \frac{d}{dt}[\boldsymbol{\lambda}_{qd0}]$$

Where,

$$\boldsymbol{\lambda}_{dq} = \begin{bmatrix} \boldsymbol{\lambda}_d \\ -\boldsymbol{\lambda}_q \\ 0 \end{bmatrix}$$

$$\lambda_q = L_q i_q \qquad \lambda_d = L_d i_d + \lambda'_m$$

$$L_q = L_{ls} + L_{mq} \qquad L_d = L_{ls} + L_{md}$$

$$L_{md} = \frac{3}{2}(L_A + L_B) \qquad L_{mq} = \frac{3}{2}(L_A - L_B)$$

L_B is 0 for round rotor machines

APPENDIX B

PMG DYNAMIC SIMULATION COMPARISON USING EULER AND RK4

```
clear;


rl = 10;                        %Resistive Load
Tmech = 5;                      %Mechanical Torque Input


rs = .45;                       %PMG Parameters
Ld = .0026;
Lq = .0026;
phi = .1723;
h = .0001;
P = 8;
J = .0121;

R = rl+rs;
L = 1/Ld;

Tstop = 1;                       %desired simulation time
iterations = Tstop/h;


lambdaq = 0;                      %initial conditions
lambdad = 0;
id(1) = 0;
iq(1) = 0;
x = 0;
x2 = 0;
Tem = 0;
w(1) = 0;
k(1) = 0;


% Runge Kutta 4th Order  Implementation ----------------------------------

for j=2:iterations,

id(j) = daxis(id(j-1), x, rl, rs, Ld, h);      %Runge Kutta Function for id
iq(j) = qaxis(iq(j-1), x2, rl, rs, Lq, h);     %Runge Kutta Function for iq
w(j)  = omega(w(j-1), Tem, J, Tmech, h);

Tem = (3/2)*(P/2)*((iq(j)*(((id(j)*Ld)+phi)))-(id(j)*iq(j)*Lq));
x = (iq(j)*Lq)*w(j)*P/2;              % x= (lamdaQ*omega
x2 = ((id(j)*Ld)+phi)*w(j)*P/2;       % x2 = ((LamdaD)+phi)*omega
```

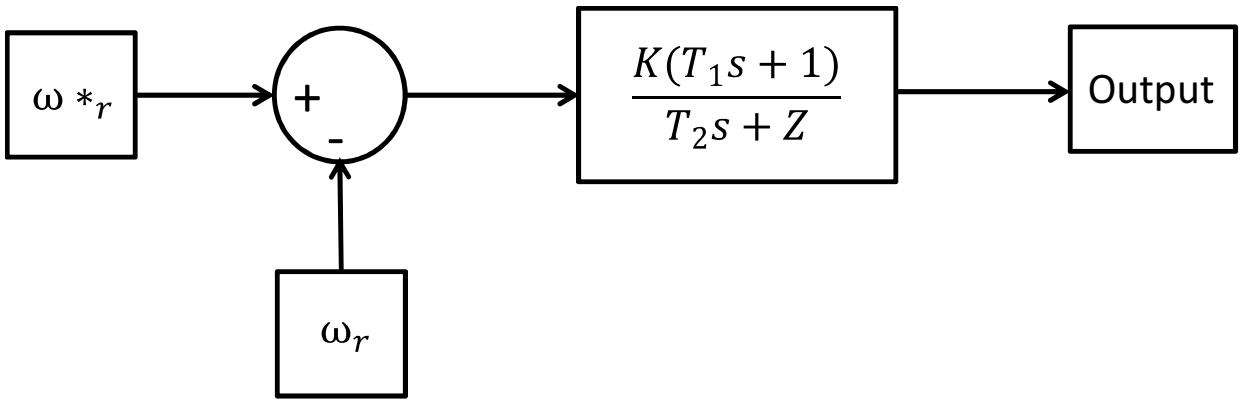
```
k(j) = k(j-1)+h;


end


p = plot(k, w); hold on;                    % Plot rotor speed of RK4 Method
set(p,'Color','red','LineWidth',1)
xlabel('Time (s)')
ylabel('Rotor Speed rad/s')


% Euler Implementation-----------------------------------------------------

for j=2:iterations,

fd(j) = (-L*R*id(j-1))+(L*x);
id(j) = id(j-1)+(h*fd(j));

fq(j) = -(L*R*iq(j-1))-(L*x2);
iq(j) = iq(j-1)+(h*fq(j));

fw(j) = (1/J)*(Tem+Tmech);
w(j) = w(j-1)+(h*fw(j));


Tem = (3/2)*(P/2)*((iq(j)*(((id(j)*Ld)+phi)))-(id(j)*iq(j)*Lq));
x = (iq(j)*Lq)*w(j)*P/2;                % x= (lamdaQ*omega
x2 = ((id(j)*Ld)+phi)*w(j)*P/2;         % x2 = ((LamdaD)+phi)*omega
k(j) = k(j-1)+h;


end

q = plot(k, w); hold on;              % Plot rotor speed of Euler Method
set(q,'Color','black','LineWidth',1);


% End ----------------------------------------------------------------


--------------------------------------------------------------------------------


function [id] = daxis(id, x, rl, rs, Ld, h)

R = rl+rs;
L = 1/Ld;
```

```
g1 = (-L*R*id)+(L*x);
    g2 = ((-L*R)*(id+(.5*h*g1)))+ (L*(x+(.5*h)));
    g3 = ((-L*R)*(id+(.5*h*g2)))+ (L*(x+(.5*h)));
    g4 = ((-L*R)*(id+(h*g3)))+ (L*(x+(h)));
    did = (h*(((1/6)*g1)+((1/3)*g2)+((1/3)*g3)+((1/6)*g4)));
    id = id+did;
end

-------------------------------------------------------------------------------

function [iq] = qaxis(iq, x2, rl, rs, Lq, h)

R = rl+rs;
L = 1/Lq;



    g1 = -(L*R*iq)-(L*x2);
    g2 = (-(L*R)*(iq+(.5*h*g1)))- (L*(x2+(.5*h)));
    g3 = (-(L*R)*(iq+(.5*h*g2)))- (L*(x2+(.5*h)));
    g4 = (-(L*R)*(iq+(h*g3)))- (L*(x2+(h)));
    diq = (h*(((1/6)*g1)+((1/3)*g2)+((1/3)*g3)+((1/6)*g4)));
    iq = iq+diq;
    diq;

end

-----------------------------------------------------------------------

function [w] = omega(w,Tem, J, Tmech, h)



    g1 = (1/J)*(Tem+Tmech);
    g2 = (1/J)*((Tem+(h/2))+Tmech);
    g3 = g2;
    g4 = (1/J)*((Tem+h)+Tmech);
    dw = (h*(((1/6)*g1)+((1/3)*g2)+((1/3)*g3)+((1/6)*g4)));
    w = w+dw;

end
```

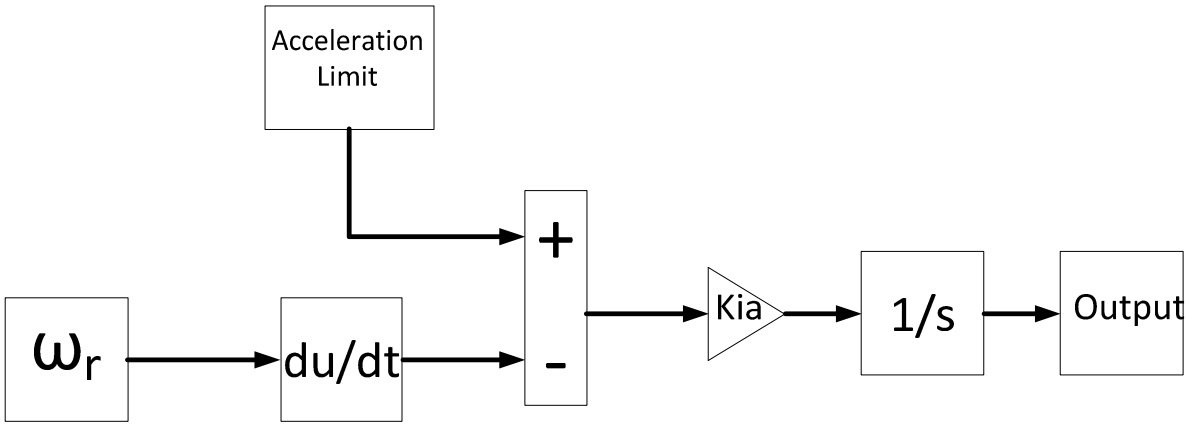
APPENDIX C

MICRO TURBINE TRANSFER FUNCTIONS AND CONTROL SCHEMES [16], [21]

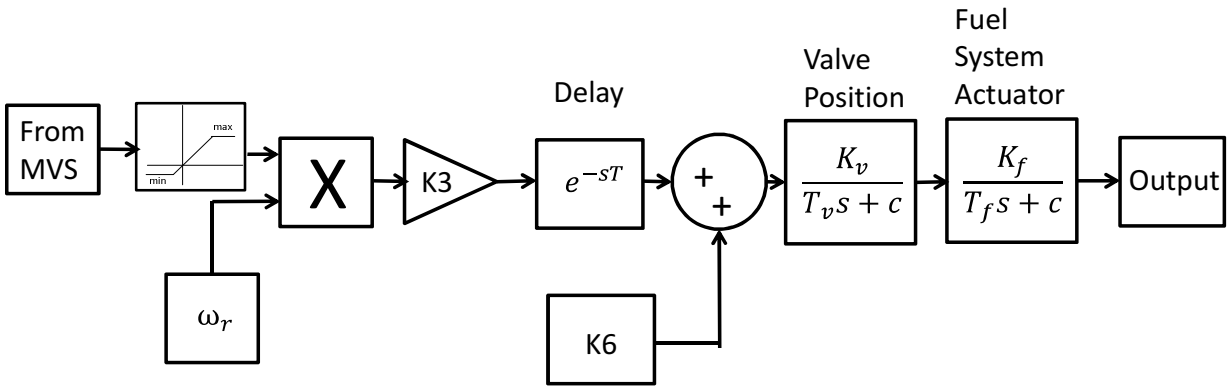
Speed Control:



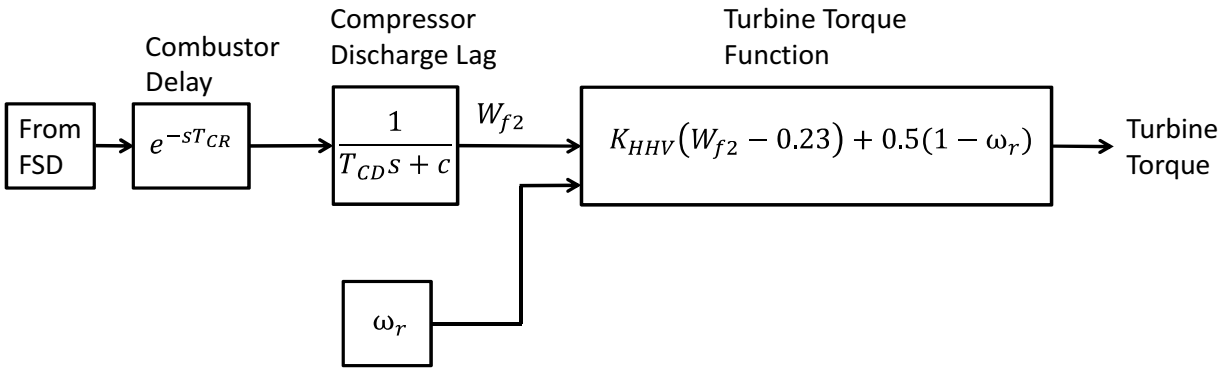
Acceleration Control:



Fuel System Dynamics:



Turbine Dynamics:



APPENDIX D

MTG DISCRETE SIMULATION USING EULER'S METHOD IN MATLAB

```
clear;
clc;

%Simulation Time Step and End Time-------------------------------------

h = .000001;
Tstop = .01;                        %desired simulation time
iterations = Tstop/h;


%Gas Turbine Parameters ------------------------------------------------
%   Adopted from "Modeling of Micro turbine Power Generation Systems"
%   Taylor and Francis

%Rating:  400kW, 70,000 RPM

wb = 7330.3828;   % base speed in rad/s
Pb = 400000;
wref = 1;
K = 25;
T1 = 0.4;
T2 = 1;
Z = 3;
kia = 100;
AccelLim = 1;
FuelUpperLimit = 5;
FuelLowerLimit = -1;
K3 = .77;
K4 = .8;
K6 = 0.23;
Kv = 1;
Tv = .05;
C = 1;
Kf = 1;
Tf = .04;
Tcr = .01;
Tcd = .04;
Khhv = 1.2;

%Initial Conditions
SpeedGov(1) = 0;
werror(1) = 0;
SpeedGovOut(1) = 0;
k(1) = 0;
Alpha(1)= 0;
Acceleration(1) = 0;
Accelin(1) = 0;
```

```
ValvePosition(1) = 0;
ValveIn(1) = 0;
ValveOut(1) = 0;
FuelOut(1) = 0;
FuelSystem(1) = 0;
CompressorIn(1) = 0;
CompressorOut(1) = 0;
CompressorDischarge(1) = 0;

% PMG Parameters----------------------------------------------------------
%   Adopted from "Modeling of Micro turbine Power Generation Systems"
%   Taylor and Francis

rl = 50;
rs = .0125;
Ld = .000165;
Lq = .000165;
phi = .2388;
P = 4;
J = .011;

R = rl+rs;
L = 1/Ld;
id(1) = 0;
iq(1) = 0;

lambdaq = 0;                    %initial conditions
lambdad = 0;
x = 0;
x2 = 0;
Tem = 0;
w(1) = 0;
k(1) = 0;

fd(1) = 0;
fq(1) = 0;
fw(1) = 0;

Tmech = 0;
Torque(1) = 0;

for j=2:iterations,

Tmech = Torque(j-1)*(Pb/wb);   %pu
fd(j) = (-L*R*id(j-1))+(L*x);
id(j) = id(j-1)+(h*fd(j));

fq(j) = -(L*R*iq(j-1))-(L*x2);
iq(j) = iq(j-1)+(h*fq(j));

fw(j) = (1/J)*(Tem+Tmech);
w(j) = w(j-1)+(h*fw(j));
```

```
Tem = (3/2)*(P/2)*((iq(j)*(((id(j)*Ld)+phi)))-(id(j)*iq(j)*Lq));
x = (iq(j)*Lq)*w(j)*P/2;              % x= (lamdaQ*omega)
x2 = ((id(j)*Ld)+phi)*w(j)*P/2;       % x2 = ((LamdaD)+phi)*omega

win(j) = w(j)/wb;   %for GT (pu)

SpeedGov(j) = SpeedGov(j-1)+(h*((-(Z/T2)*SpeedGov(j-1))+werror(j-1)));

Acceleration(j) = (win(j) - win(j-1))/h;
Accelin(j) = (AccelLim - Acceleration(j))*kia;

Alpha(j) = Alpha(j-1)+(Accelin(j)*h);

ValvePosition(j) = (((-(C/Tv)*ValvePosition(j-1))+ValveIn(j-
1))*h)+ValvePosition(j-1);

FuelSystem(j) = (((-(C/Tf)*FuelSystem(j-1))+ValveOut(j-1))*h)+FuelSystem(j-
1);


CompressorDischarge(j) = (((-(C/Tcd)*CompressorDischarge(j-
1))+CompressorIn(j-1))*h)+CompressorDischarge(j-1);

werror(j) = wref - win(j);
SpeedGovOut(j) = ((K/T2)-
((K*T1*Z)/(T2^2)))*SpeedGov(j)+((werror(j)*(K*T1/T2)));

if (min(SpeedGovOut(j),Alpha(j)) > FuelUpperLimit)
    Fuel = FuelUpperLimit;
elseif (min(SpeedGovOut(j),Alpha(j)) < FuelLowerLimit)
    Fuel = FuelLowerLimit;
else
    Fuel = min(SpeedGovOut(j),Alpha(j));
end
ValveIn(j) = (Fuel*win(j)*K3)+K6;
ValveOut(j) = (Kv/Tv)*ValvePosition(j);

FuelOut(j) = (Kf/Tf)*FuelSystem(j);

if j < (Tcr/h);
    CompressorIn(j) = 0;
else
    CompressorIn(j) = FuelOut(round(j-(Tcr/h)+1));
end
CompressorOut(j) = (1/Tcd)*CompressorDischarge(j);

Torque(j) = (Khhv)*(CompressorOut(j)-0.23)+(0.5*(1-win(j)));

k(j) = k(j-1)+h;

end

 plot(k)
```

REFERENCES

[1] "IEEE Recommended Practice for 1 kV to 35 kV Medium-Voltage DC Power Systems on Ships," IEEE Std 1709-2010, pp.1-54, Nov. 2 2010

[2] C. Washington, and J. Dolman, "Creating next generation HIL simulators with FPGA technology," IEEE *AUTOTESTCON, Conference,* pp.1-6, Sept. 2010

[3] Z. Weidong, S. Pekarek, J. Jatskevich, O. Wasynczuk, and D. Delisle, "A model-in-the-loop interface to emulate source dynamics in a zonal DC distribution system," *IEEE Transactions on Power Electronics,* vol.20, no.2, pp. 438- 445, March 2005.

[4] M. Matar, and R. Iravani, "Massively parallel implementation of AC machine models for FPGA-based real-time simulation of electromagnetic transients," *IEEE Transactions on Power Delivery,* vol.26, no.2, pp.830-840, April 2011.

[5] Bash, M.; Pekarek, S.; Sudhoff, S.; Whitmore, J.; Frantzen, M.; , "A comparison of permanent magnet and wound rotor synchronous machines for portable power generation," *Power and Energy Conference at Illinois (PECI), 2010* , vol., no., pp.1-6, 12-13 Feb. 2010

[6] P. C. Krause, O. Wasynuczuk and S. D. Sudhoff , *Analysis of Electric Machinery and Drive Systems*, 2nd ed. West Lafayette, IN, Wiley-Interscience, pp. 261-281.

[7] R . H. Park, "Two-Reaction Theory of Synchronous Machines-Generalized Method of Analysis Part I", *AIEE Transactions,* Vol. 48, July 1929, pp. 716-727.

[8] H. C. Stanley, "An Analysis of the Induction Motor" , *AIEE Transactions,* Vol. 57 (Supplement), 1938, pp. 751-755.

[9] G. Kron , *Equivalent Circuits of Electric Machinery*, John Wiley and Sons, New York, 1951

[10] P. C. Krause, F. Nozari, "Simulation of Symmetrical Induction Machinery", *IEEE Transactions on Power Apparatus and Systems"*, Vol.84, November 1965, pp. 1038-1053.

[11] A. Al-Hinai, and A. Feliachi, "Dynamic model of a microturbine used as a distributed generator," IEEE 34th Southeastern Symposium on System Theory, Huntsville, Alabama, 2002.

[12] Y. Zhu, and K. Tomsovic, "Development of models for analyzing the load-following performance of microturbines and fuel cells", in Electric Power Systems Research, vol. 62, pp. 1-11, 2002.

[13] P.A. Pilavachi, "mini- and micro-gas turbines for combined heat and power," in Applied Thermal Engineering, vol. 22, pp. 2003-2014, 2002.

[14] H. Nikkhajoei and M. R. Iravani, “A Matrix converter based micro-turbine distributed generation system,” IEEE Transaction on Power Delivery, vol. 20, no. 3, pp. 2182-2192, Jul. 2005.

[15] Gas Turbine Systems Technician (Mechanical) 3 & 2, NAVEDTRA 10548-2 Chapter 6

[16] W. I. Rowen, “Simplified mathematical representations of heavy-duty gas turbines,” ASME J. End. Power, vol. 105, pp. 865-869, 1983.

[17] Power System Simulator – PSS/E, Program Operation Manual, Power Technologies, Inc., Schenectady, New York, USA.

[18] K. Y. Soon, J. V. Milanovic and F. M. Hughes, "Overview and Comparative Analysis of Gas Turbine Models for System Stability Studies", IEEE Transactions on Power Systems, vol.23, no.1, pp.108-118, 2008.

[19] M. Nagpal, A. Moshref, G.K. Morison, and P. Kundur, “Experience with testing and modeling of gas turbines,” presented at the IEEE Power Engineering Society General Meeting, Columbus, OH, vol. 2, 2001.

[20] G. J. Kish, and P. W. Lehn, "A micro-turbine model for system studies incorporating validated thermodynamic data," IEEE Power and Energy Society General Meeting, pp.1-6, 24-29, 2011.

[21] S. R. Guda, C. Wang, and M. H. Nehrir, Modeling of Micro turbine Power Generation Systems, Electric Power Components and Systems. London, U.K.: Taylor & Francis, Sep. 2006.

[22] C. Dufour, J. Belanger, V. Lapointe and S. Abourida, “Real-time simulation on FPGA of a permanent magnet synchronous machine drive using a finite-element based model," *International Symposium on Power Electronics, Electrical Drives, Automation and Motion, 2008. SPEEDAM 2008*, pp.19-25, June 2008.

[23] S. Kallio, E. Hartikainen, J. Luukko, and O. Pyrhonen, “Emulation of PMSG/half-controlled converter combination with digital signal processor," *IEEE International Conference on Industrial Technology (ICIT),* pp.159-164, March 2011.

[24] open cores SPI Master/Slave Interface. Available: http://www.opencores.org

[25] K. Jayalakshmi, and V. Ramanarayanan, “Real-time simulation of electrical machines on FPGA platform," *India International Conference on Power Electronics (IICPE)*, pp.259-263, Dec. 2006.

[26] C. Hao, S. Song, D. C. Aliprantis, and J. Zambreno, “Dynamic simulation of electric machines on FPGA boards," *IEEE International Conference on Electric Machines and Drives Conference (IEMDC)*, pp.1523-1528, May 2009.

[27] Jeffery J. Leader, *Numerical Analysis and Scientific Computation*. Pearson Education 2004. pp 381-428.

[28] Fixed Point Package User's Guide. Available: http://www.vhdl.org/fphdl/Fixed_ug.pdf.

[29] MATLAB/SIMULINK Documentation. Available: http://www.mathworks.com.

[30] M. bash, R. R. Chan, J. Crider, C. Harianto, J. Lian, J. Neely, S. D. Pekarek, S. D. Sudhoff, and N. Vaks, "A medium voltage DC testbed for ship power system research," *IEEE Electric Ship Technologies Symposium (ESTS),* pp.560-567, April 2009.

[31] N. S. Nise, Control Systems Engineering 5th ed. John Wiley & Sons, pp 126-132, 2007.

BIOGRAPHICAL SKETCH

Nicholas Stroupe was born in Tampa, Florida on August 20, 1987. He graduated high school in Tampa in 2005. He received his B.S. degree Cum Laude in Electrical Engineering at the Florida State University in 2010. During his time as an undergraduate student he served as Vice President of Tau Beta Pi - Florida Eta Chapter in the 2009-2010 academic year and also received the Tau Beta Pi Engineering Honor Society Scholarship. He also participated as a charter member of Eta Kappa Nu – Lambda Delta Chapter Electrical and Computer Engineering Honor Society starting in Spring of 2008. He is currently completing his Master's Degree in Electrical Engineering at the Florida State University with the expected graduation date of August 2012. Currently, he is a research associate for the Energy Conversion and Integration Thrust at the Center for Advanced Power Systems at Florida State University under the supervision of Dr. Chris S. Edrington. During his time as a research associate, Nicholas has achieved and authored the following works:

- N. Stroupe, F. Fleming, A. Hasanzadeh, C.S. Edrington, "Real-time High Speed Generator Emulation for Hardware-in-the-Loop Applications," *ASNE Day 2012*
  1st Place *Graduate Category Student Poster Competition.*

- N. Stroupe, A. Hasanzadeh, C.S. Edrington and T. Bevis, "Real-Time High Speed Generator System Emulation with Hardware-in-the-Loop Application," *ASNE Electric Machine Technology Symposium, 2012.*

- N. Stroupe, A. Hasanzadeh, C.S. Edrington and T. Bevis, "Multi-Platform Real-Time Realization of Isolated Permanent Magnet Synchronous Generator," *Industrial Electronics Society, 2012. IECON 2012. Annual Conference of IEEE.*