Power Hardware-In-The-Loop Emulation of a Brushless DC Motor

Tshamala Kalonji Gael

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Electrical and Computer Engineering) at

Concordia University

Montréal, Québec, Canada

April 2023

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Tshamala Kalonji Gael

Entitled: Power Hardware-In-The-Loop Emulation of a Brushless DC Motor

and submitted in partial fulfilment of the requirements for the degree of

Master of Applied Science (Electrical and Computer Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair
Dr Luiz Lopes

_____ Internal Examiner
Dr Chunyan Lai

_____ Supervisor
Dr Pragasen Pillay

Approved by: _____
Dr Yousef R. Shayan, Chair
Department of Electrical and Computer Engineering

_____
Dr Mourad Debbabi, Dean,
Faculty of Engineering and Computer Science

Abstract

Power Hardware-In-The-Loop Emulation of a Brushless DC Motor

Tshamala Kalonji Gael, MASc.

Concordia University, 2023.

The current awareness towards global warming, green energy and energy saving is the driving force behind the development of power electronics. Power electronics plays a significant role in many industrial, commercial and military applications such as electric vehicles (EVs), photovoltaic (PV) inverters, transportation, distribution, home appliances, electric drive systems, etc. Many of the technological advancements recorded nowadays are made possible because of the rapid development in the field of power electronics over the last few decades. Therefore, more energy-savvy, reliable, intelligent, and efficient devices are currently being developed and deployed because of the rapid development of power electronics. Hence power electronics is, to a great extent, one of the biggest enablers of environmental clean energy.

Electric motors are the workhorse of the industry and it is estimated that 90 % of industrial electricity consumption is used to power electric drive systems. Electric drive systems are made up of various parts among which the most important are the electric motor, the drive inverter, the load and the drive controller. In conventional electric motor testing, the electric motor of the drive system to be tested is mounted on a test bench and coupled to a dynamometer for testing the drive system for various speeds and loading conditions. However, the conventional dynamometer-based drive testing method has many limitations., An alternative solution which is currently gaining popularity in the industry is the use of power-hardware-in-the-loop (PHIL) emulation for performing electric machine testing.

In this research work, a PHIL emulation of a brushless DC (BLDC) motor for a BLDC drive utilizing sensored trapezoidal control is presented. First, an improved BLDC model is used for simulating the drive. Then a BLDC drive setup is built and tested. The simulation results obtained are validated using the results from this physical BLDC motor drive. Next, the simulations of PHIL emulators for permanent magnet synchronous machine (PMSM) and BLDC are performed. Then the emulator hardware for BLDC emulation is built and tested. Open-loop emulator control is employed for emulating the BLDC motor. The results from the emulator are validated for steady-state and transient operations using results from the physical BLDC motor drive.

Acknowledgement

For God so loved the world, that he gave his only begotten Son, that whosoever believeth in him should not perish, but have everlasting life. Thanks be unto my God Jesus Christ who always causeth me to triumph.

I would like to express my heartfelt appreciation to my parents Pierre Kalonji Kongolo and Modestine Meta Manga, and to my brothers and sisters Patrick, Philip, Franck, Arthur, Mike, Nadine, Christelle and Pascal, Leticia and Gloria, Tonton Ghyslain.

I would like to express my genuine gratitude to my supervisor Prof. Pragasen Pillay, who gave me an opportunity to pursue my MASc under his guidance. Prof. Pillay has been a role model, Boss, luminary, and above all a father. I hope to never discredit the investment he made in me.

My sincere thanks go to my committee members, Prof. Chunyan Lai and Prof. Luiz A.C. Lopes for being committee members even in hardship and for the valuable comments, suggestions, and valuable time.

I would like to sincerely acknowledge the help of Dr Mathews Boby throughout this research.

I would also like to cordially thank my colleagues who made a home-like atmosphere in the PEER group lab. Nazanin, Amir, Sumeet, Yupeng, Gayathri, Mohanraj, Tamanwe, Neetusha, Bassam, Koteswara, Tolga, Talha, Ying, Dr Akrem, Ahmad, Paul, Sahil, Neil, Shiva, Zixhe.

Finally, I would like to express my gratitude to my friends and family Handy, Prince, Hans, Idie, Enoch, Papy, Rati, Claude, Jean-Pierre, Mr Mubanga.

This thesis is wholeheartedly dedicated to my wife Gabriella Tshika, thanks for your love and patience.

Table of contents

List of Figures

List of tables

Chapter 1: Introduction

1.1 General

An electric motor is an electromechanical device which converts electrical power into mechanical power. It plays a major role in industrial applications as the workhorse of a drive system. Most of the electricity used in the industry sector is for motor-driven systems (including pumps, fans and compressors). Industry and buildings account for over 90% of global electricity demand today, while transport makes up less than 2%. The leading drivers of global electricity demand growth are industrial motors (over 30% of the total growth to 2040), space cooling (17%), and large appliances, small appliances and electric vehicles (EVs) (10% each) [1]. A motor drive is a system that provides a mechanical load with continuous range process speed and torque control; thus, it is capable of adjusting both the speed and torque of a DC or an AC motor [2]. Some of the advantages of electric motor drive systems are [2]:

- Starting and braking are easy and simple.
- Provides a wide range of torque over a wide range of speed (both AC and DC motor).
- Can be started and accelerated within a very short time.
- Flexible control characteristics.
- Capable of operating in all four quadrants of torque and speed plane.

Fig.1-1 shows a typical block diagram of an electric motor drive system. The main components of an electric drive system are: the electrical power supply which provides either DC or AC voltage;

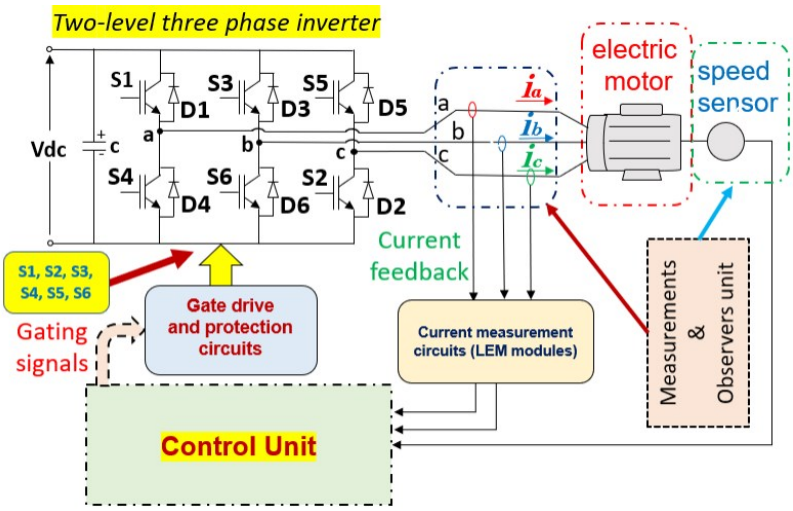


Figure 1-1: Electric drive system block diagram

the power electronics converter which serves as an interface between the voltage source and the motor; the electric motor which is coupled to the mechanical load; the control unit which includes the hardware and control software algorithm necessary to monitor and control the required electric motor drive system; the control observer and measuring unit which is used to measure electrical quantities (voltage and/or current) or mechanical quantities (torque and/or speed); and finally the mechanical load which is connected to the motor shaft[2]. The design of a motor drive depends on the type of mechanical load connected to it whether a constant power load, a fixed load torque, a linear load torque or a parabolic load torque.

Industrial drive applications are generally classified into constant-speed and variable-speed drives. Traditionally, AC machines with a constant frequency sinusoidal power supply have been used in constant-speed applications, whereas DC machines were preferred for variable-speed drives [3]. Although DC machines have the advantage of simple control of the speed over their AC counterpart, the use of the commutator inside these machines leads to high maintenance costs and lower efficiency, hence their use in the industry is gradually being limited to low-power applications. With the current global warming and its impacts on the environment, the need to lower the global energy demand is driving current and future research in power electronics, motor control and machine design to increase the overall efficiency of electric drive systems, and contribute to the global call for energy saving and fight against the current climate crisis the planet is facing.

1.2 Problem statement

The widely accepted and used industrial method for testing electric drive systems uses a dynamometer test bench. The purpose of using the dynamometer is to apply different test conditions (speed or torque) to the drive system to validate its operation. For a typical drive system application, the design and prototyping of the motor usually have a long lead time. The power converters on the other hand are usually readily available. Due to this, in a drive system manufacturing industry, the drive engineer must wait a long time to test the controller and power converter of the drive system until the motor is available. This is one of the shortcomings of the conventional drive testing method. Another problem arises when the ratings of the available mechanical test bench do not meet the specifications of the drive system, therefore the number of tests which can be performed in such cases is limited. A third scenario can be in an (EV)

application when there is a certain type of fault in the drive system (motor or inverter fault) while the EV is cruising on the road. EV manufacturers are currently developing new motor control strategies to keep the vehicle running for few kilometres under this condition, to ensure that the vehicle reaches the nearest mechanic shop or it is safely driven out of the road. Testing faulty conditions on a physical machine is usually not recommended as it may lead to insulation failure, demagnetization of the magnets in the case of a permanent magnet synchronous motor (PMSM), or complete damage to the machine.

Power-Hardware-In-the-Loop (PHIL) testing method is increasingly gaining momentum as a new testing method for applications such as EVs, electric drive systems, photovoltaic (PV) inverters, etc. In electric drive applications, PHIL provides a way to replace the conventional mechanical test bench with an electrical test bench. In other words, the real machine is replaced by a so-called virtual machine [4]. In drive applications, the PHIL emulation uses power converters to behave as a motor (virtual motor). The emulator controller controls the power electronics converter in such a way that it draws the same currents which would be drawn by the real machine if it was connected to the same drive inverter and subjected to the same loading and speed conditions.

1.3 Motivation

The work presented in this thesis proposes an open-loop PHIL emulation method of a BLDC motor for use in a drive with sensored trapezoidal control. The focus of the work presented here is on the design and development of a PHIL emulation platform for a three-phase brushless DC motor (BLDC) drive with a sensored trapezoidal control. The emulator setup is validated by comparing the emulator current with the real machine current at various speeds and loading conditions. The steady-state operation of the emulator and the real machine currents at different speeds and torque conditions are analysed and compared. The transient operation of the emulator and real machine currents at different speeds and torque conditions are analysed and compared. Software simulation of the BLDC emulator, emulator hardware and real machine experimental results are analysed and compared for different speeds and torque conditions.

1.4 Objectives

The main objective of this work is to simulate, implement and analyse the open-loop emulation of a BLDC motor with sensored trapezoidal control. This is achieved by:

- Proposing an improved BLDC motor model, which is validated by comparing it to the experimental BLDC drive results and once validated the model is used throughout the rest of the thesis for simulation purposes.
- Developing and testing the BLDC drive setup and testing the motor at various speeds and load conditions using a load bank.
- Developing and implementing the software simulation of closed-loop and open-loop emulation of the BLDC motor.
- Prototyping and testing a three-phase two-level MOSFET-based drive inverter which is used in the drive system.
- Developing, testing and validating the open-loop emulator test bench of the BLDC with sensored trapezoidal control.

1.5 Limitations

The hardware implementation of the closed-loop emulation of BLDC is presented in [5], in this thesis only the software simulation of the closed-loop emulation of the BLDC with sensored trapezoidal control is presented.

The prototyping and testing of the drive inverter are explained in this thesis. Design aspects of this inverter are not considered in this thesis.

The BLDC motor hardware used in this thesis has a rated power of 200 watts and a maximum speed of 3000 rpm. Therefore, the tests performed in this thesis are limited to these ratings.

1.6 Literature review

The concept of the hardware-in-the-loop (HIL) testing method is not new, it has been used in the industry for many decades. HIL testing methods can be classified as control hardware-in-the-loop (CHIL) and power hardware-in-the-loop (PHIL). CHIL is used in the industry to test the control, communication or operation logic of controllers such as PLCs or intelligent electric devices (IEDs) such as protection relays. A typical example of the use of the CHIL testing method is during a factory acceptance test (FAT) where a secondary injection set is used to test the protective relaying control of a switchboard before being deployed on-site [6]-[8]. PHIL on the other hand is a relatively new concept. It uses power electronics converters to allow the hardware under test (HUT) to be tested with real power flow (electric power is absorbed or supplied). In other words, as opposed to CHIL where there is no flow of power, PHIL allows the HUT to be tested with real power flow.

In electric drive applications, PHIL testing is now being considered as an alternative testing solution for drive systems when the drive inverter (HUT) and its control algorithm need to be tested for various speed and load conditions. The design of PHIL test benches for induction motors and permanent magnet motors is discussed in different literature such as [9]-[15]. In this thesis, the focus is on the PHIL test bench for permanent magnet (PM) motors, more specifically (BLDC) motors. The closed-loop emulation of the BLDC motor is introduced in [5]. The real-time simulation of a BLDC-based wind turbine emulator is discussed in [16]. However, not much work is done in the literature on the open-loop emulation of the BLDC motor. Therefore, this thesis work develops, implements and tests a PHIL open-loop emulation of a BLDC motor.

1.7 Thesis outline

This thesis is organized as follows:

Chapter 1 gives a summary of the research work presented in this thesis. It gives background information on the electric drive systems and their purpose in the industry. It also states the problem, motivation, objectives, and limitations related to this research.

Chapter 2 discusses and develops an improved BLDC motor model. This motor model correctly reproduces some of the salient features of the BLDC drive waveform. The model is validated by comparing the pole voltages and motor currents of the hardware versus simulation results.

Chapter 3 discusses the implementation of the BLDC drive with rotor position sensor and trapezoidal control and the hardware setup used for the BLDC drive. It also gives background information on machine emulation using PHIL. The simulation of the closed-loop emulation of PMSM and the simulation of the closed-loop and open-loop emulation of the BLDC motor with sensored trapezoidal control are discussed in the same chapter.

Chapter 4 presents the prototyping and testing of a MOSFET-based three-phase two-level power converter which is later used as a drive inverter in the BLDC motor drive system.

Chapter 5 discusses the implementation of the open-loop emulation of the BLDC motor with sensored trapezoidal drive control. The emulator setup is built, tested and validated using the results of the emulator currents and the experimental BLDC motor currents.

Chapter 6 presents the conclusions and future works.

Chapter 2: BLDC modeling and simulation

2.1 Introduction

In this chapter, a quick review of the brushless DC motor (BLDC) modeling is done. A conventional BLDC model found in literature is simulated using MATLAB. An improved BLDC motor model is later introduced. It adds additional features to the conventional motor model. The improved BLDC motor model is then validated by comparing it to the physical BLDC motor.

A BLDC motor is built with armature windings on the stator, with the field in the machine being produced by the permanent magnet (PM) mounted on the rotor as shown in Fig.2-1. A brief comparison between the BLDC motor with other type of motors is given below [17]-[20]:

- **Brushless vs Brush DC motor:** BLDC has higher torque and speed bandwidths, lower maintenance, and robustness. The use of a mechanical commutator in a brushed DC motor imposes speed and current limitations, thus resulting in design limitations.
- **Brushless DC motor vs Induction motor:** BLDC has higher torque to inertia ratio, higher efficiency, smaller size for the same capacity, higher power density, etc. The induction motor on the other hand has a lower cost, lower cogging torques, can operate at higher temperatures, etc.
- **Brushless DC motor vs PMSM:** In the case of the trapezoidal brushless DC machine, simple position sensing devices like the Hall effect sensor can be used to give rotor position feedback, while costly high-resolution optical encoder or resolver are required in permanent magnet synchronous motor (PMSM) drive. Additional information on the selection criteria between these two motors for a specific application is discussed in [20].

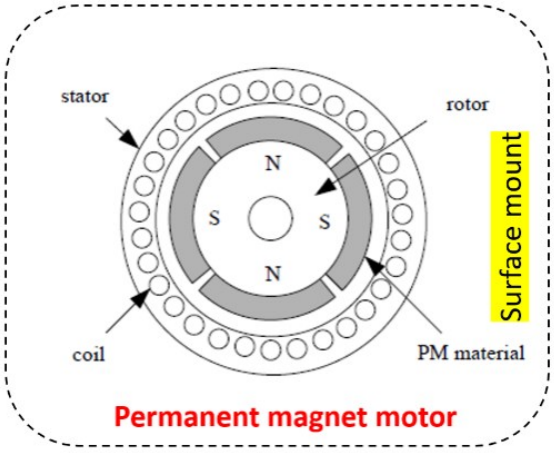


Figure 2-1: Brushless DC motor construction [21]

2.1.1 Applications and future of BLDC motors

Because of its low noise, control simplicity, robustness, high efficiency, and linear speed-torque characteristics in its operating range, the BLDC has made remarkable expansion in various applications such as household appliances, automotive, aerospace, civil, and military robots, etc. [21]. The future of BLDC motor drives depends on the development and miniaturization of power electronics and microprocessors, the development of PMs, optimization and implementation of newer design algorithms such as genetic algorithms, niche algorithms, etc. Performance comparison of different types of motors for specific applications are given in [21] [22].

2.2 Mathematical model of the BLDC motor

The mathematical modeling of BLDC machine is explained in this section. First a general model which is found in literature is explained and simulated. The short-comings of the conventional model are pin-pointed. An improved BLDC model which is capable of representing the salient waveform features of a BLDC motor drive is presented.

2.2.1 Conventional BLDC motor model

2.2.1.1 BLDC motor equations

The constructional feature of the BLDC stator is the same as that of the induction motor and PMSM, with the only difference being that the stator windings of the BLDC motor are wound with full pitch, hence ensuring that the back-emfs have trapezoidal shapes. Whereas in induction and PMSM motors, the back-emfs are sinusoidal, allowing the motor *abc* model to be transformed to the *dq* reference frame, which has the advantage of reducing the complexity of the machine model. However, for the BLDC motor, this trapezoidal back-emf makes the transformation to a *dq* reference frame difficult to accomplish. Hence, for BLDC motors, only the *abc* model is used [22]. Since the BLDC has three stator winds and PMs on the rotor, equations 2.1 to 2.6 below are the circuit equations of the three-phase BDLC motor as given in [22].

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} L_{aa} & L_{ba} & L_{ca} \\ L_{ba} & L_{bb} & L_{bc} \\ L_{ca} & L_{cb} & L_{cc} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} \qquad (2.1)$$

Assuming that $L_{aa} = L_{bb} = L_{cc} = L$, and $L_{ab} = L_{ca} = L_{cb} = M$, then

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} L & M & M \\ M & L & M \\ M & M & L \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} \qquad (2.2)$$

For windings connected in star, $i_a + i_b + i_c = 0$. Since, $Mi_a = -(Mi_b + Mi_c)$. Equation (2.2) becomes:

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} L-M & 0 & 0 \\ 0 & L-M & 0 \\ 0 & 0 & L-M \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} \qquad (2.3)$$

The steady-state equations are:

$$\frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \begin{bmatrix} 1/(L-M) & 0 & 0 \\ 0 & 1/(L-M) & 0 \\ 0 & 0 & 1/(L-M) \end{bmatrix} . \left[\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} - \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} - \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} \right] \qquad (2.4)$$

and the electromagnetic torque is

$$T_e = \frac{(e_a i_a + e_b i_b + e_c i_c)}{\omega_r} \qquad (2.5)$$

The model for the mechanical subsystem is

$$\frac{d}{dt} \omega r = \frac{(T_e - T_L - B\omega_r)}{J} \qquad (2.6)$$

Fig.2-2 shows the conventional BLDC motor drive system using a two-level three-phase inverter configuration. The BLDC motor model shows the internal stator star winding configuration. The mechanical subsystem represents the torque and motion equations.

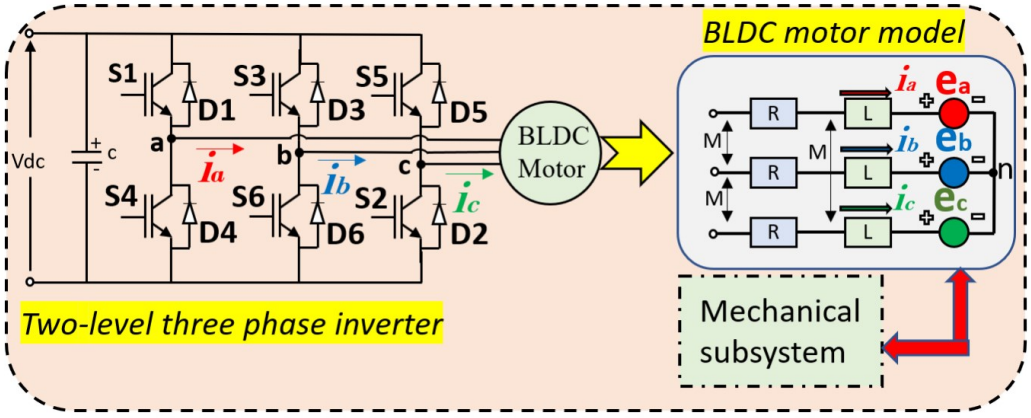


Figure 2-2: Conventional BLDC motor drive system

In a BLDC motor drive with sensored trapezoidal control, the electronic commutation is achieved by energizing only two of the three motor windings at any time. The most common BLDC winding

configuration is star connected, hence in this thesis, only star-connected windings are discussed. Fig.2-3(a) shows the two-level inverter pole terminals and Fig.2-3(b) shows the BLDC motor star winding connection, the voltages, back emf, currents, the winding resistances, and inductances inside the machine. In the actual physical machine, the effect of demagnetization of the commutated phase is visible in the pole voltage as can be observed from Fig.2-5(c) event number (1), where the pole voltage is clamped to Vdc or 0 depending on the direction of the current before the floating phase. From Fig.2-3(a, b) the following equations can be derived:

$$v_{ab} = R_{LL} i_{ab} + (L - M)_{LL} \frac{di_{ab}}{dt} + e_{ab} \qquad (2.7)$$

$$v_{bc} = R_{LL} i_{bc} + (L - M)_{LL} \frac{di_{bc}}{dt} + e_{bc} \qquad (2.8)$$

$$v_{ca} = R_{LL} i_{ca} + (L - M)_{LL} \frac{di_{ca}}{dt} + e_{ca} \qquad (2.9)$$

using Kirchhoff's current law,

$$i_a + i_b + i_c = 0 \qquad (2.10)$$

therefore,

$$i_a = \frac{i_{ab} - i_{ca}}{3} \qquad (2.11)$$

$$i_b = \frac{i_{bc} - i_{ab}}{3} \qquad (2.12)$$

$$i_c = \frac{i_{ca} - i_{bc}}{3} \qquad (2.13)$$

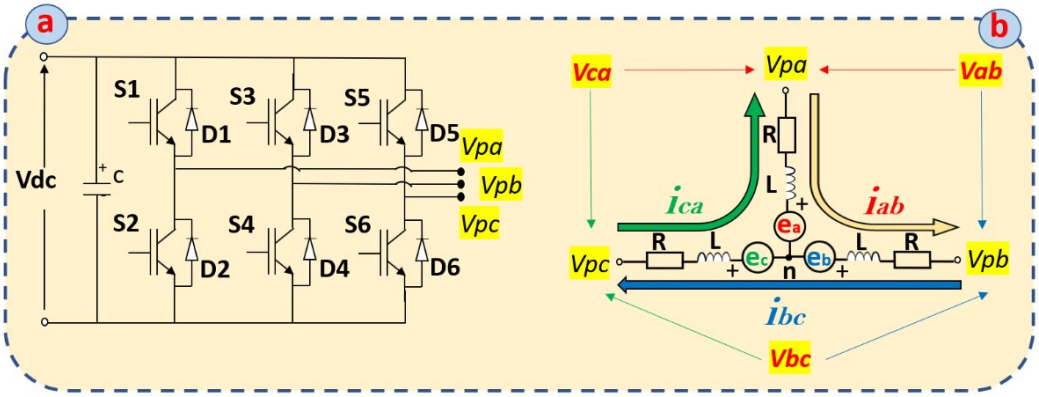


Figure 2-3: Pole voltage during the floating phase with the positive current. (b) Pole voltage during the floating phase with the negative current

2.2.1.2 Conventional BLDC motor model simulation

BLDC motors that are used in sensored trapezoidal control applications are fit with sensors to provide rotor position information. Hall sensors are the most common type among them [23]. Hall-effect position sensors are placed at every 60° electrical angle increments [22]. Whenever the rotor magnetic poles pass near the Hall sensors, they give a high or low signal, indicating that a North or South pole is passing near the sensor [23]. These sensor signals are used inside the controller to determine the rotor position and send switching signals to the drive inverter (two-level three-phase inverter in this case) to run the motor. When using a two-level three-phase inverter as a drive inverter, considering only one leg of the inverter as shown in Fig.2-4, the voltage measured between the inverter terminal 'a' and the DC bus ground (GND) is called the pole voltage. The switching sequence of the switches is also given in Fig.2-4. It is important to notice here that when no switching signal is sent to the switches (S1, S4), the pole terminal 'a' is said to be floating. The pole voltage depends on the back emf, the current direction, and the neutral voltage as it be discussed in the next section.

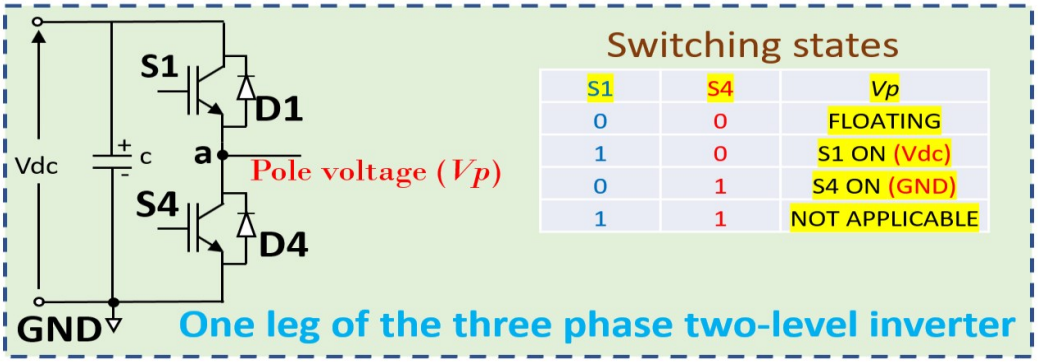


Figure 2-4: BLDC drive inverter switching states

A simulation of the BLDC motor drive has been presented in [24]. As can be observed from the simulation results presented in [22] and [24], the pole voltage is assumed to be equal to the back-emf of the motor during the floating phase of the pole terminal. The approach given in [24] is used to simulate the conventional motor model in Simulink. The model is simulated with fixed-step discrete solver with a time-step of 1 us, the drive switching frequency is 10 kHz and the drive control works at a time-step of 100 us. The drive employs conventional speed control using an inner current PI controller and an outer PI controller.

Fig.2-5 (a) is the simulation result for the current and pole voltage for phase A with maximum duty cycle (duty =1). Fig.2-5 (b) is the simulation result for the current and pole voltage for phase A with a duty cycle less than the maximum duty cycle (duty < 1). Fig.2-5 (c) is the physical BLDC hardware result for the current and pole voltage for phase A with maximum duty cycle (duty = 1). Fig.2-5 (d) is the physical BLDC hardware result for the current and pole voltage for phase A with a duty cycle less than the maximum duty cycle (duty < 1). The simulation and experimental results do not match, the reason being that with the approach given in [24], the floating phase of the inverter terminal is not modeled correctly.

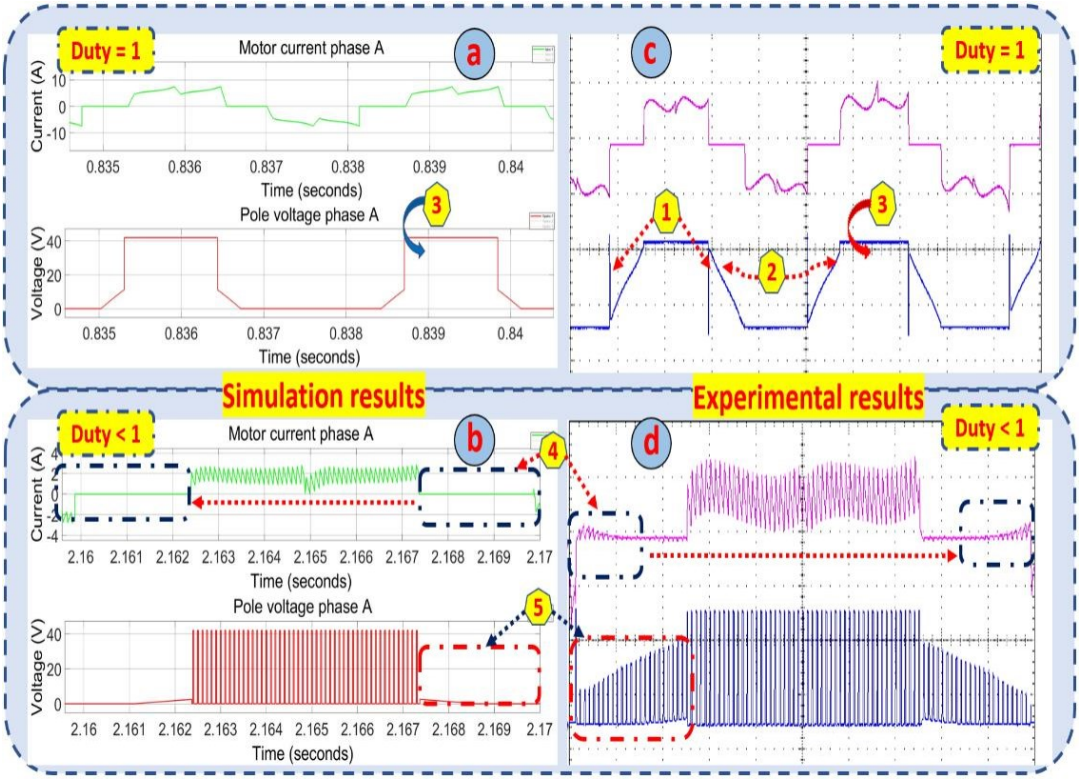


Figure 2-5: (a & b) Simulation results: Green trace: phase current ((10 A/div) (2 A/div)), Red trace: pole voltage (20 V/div). (c & d) Experimental results: Purple trace: phase current ((10 A/div) (2 A/div)), Blue trace: pole voltage (30 V/div).

Although, Simulink Simpower simulations can take care of this floating phase during simulation when an inverter block is used, the simulation is usually very slow, and the user usually has no idea of how the implementation is done. Therefore, the choice to implement the entire drive using a discrete solver was selected. To enhance the features of the conventional BLDC motor model presented in [22] and [24], an improved BLDC motor model was derived using the approach given in [21].

2.2.2 BLDC motor waveform features

2.2.2.1 Demagnetization of the commutated phase

As previously stated, in the conventional BLDC motor drive simulation in [22] and [24], the floating state is not considered. Thus, the pole voltage is assumed to be equal only to the back-emf during the floating phase of the switches. Fig.2-6 below is used to study the pole voltage (*Vpa* in the figure) during the floating phase of the inverter switches. To simplify the circuit the equivalent circuit of a DC motor which consists of the winding resistance and inductance (R and L) and the back-emf voltage (e_a) is used in Fig.2-6. Consider the case at the beginning of the floating phase when the winding current is still positive (Ia^+) (typically when S1 was ON and S4 OFF before going into the floating phase). As it can be observed, the bottom diode D4 starts conducting, and the pole voltage becomes equal to 0 V (neglecting the diode voltage drop). When the winding current is negative (Ia^-) at the beginning of the floating phase (typically when S4 was ON and S1 OFF before going into the floating phase), the top diode D1 starts conducting, and the pole voltage becomes equal to Vdc (neglecting the diode voltage drop).

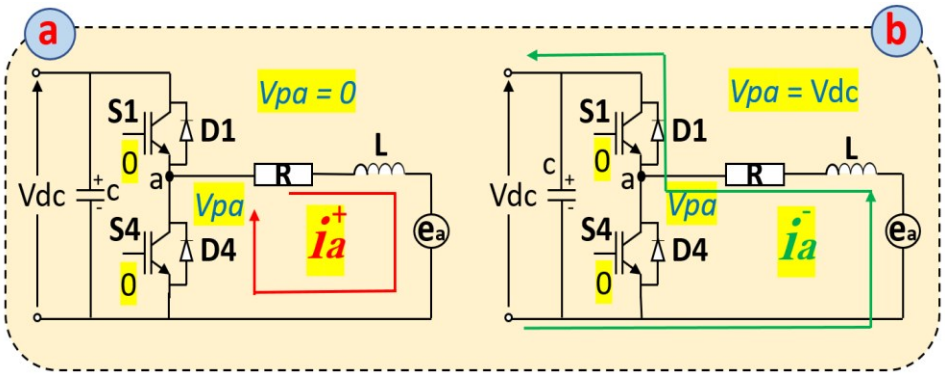


Figure 2-6: (a) Pole voltage during the floating phase with the positive current. (b) Pole voltage during the floating phase with the negative current

2.2.2.2 Neutral voltage

During the floating phase of the BLDC drive inverter leg as shown in Fig.2-7 (c phase), the pole voltage V_{pc} which will appear at pole 'c' terminal of the drive inverter is equal to the back-emf voltage of phase-c plus the neutral voltage (*Vn*). The neutral voltage can be obtained using Kirchhoff's voltage law as described in [25]. A brief explanation of the neutral voltage is given below using the scenarios described in Fig.2-7:

- **C-phase floating with S1 ON and S6 ON:** positive current (Ia^{+}) *flows* in the winding of phases A and B. The voltage loop can be derived to be:

$$V_n = v_a - (R * i_a) - (L * \frac{d\, i_a}{dt}) - e_a \quad (2.14a)$$

$$V_n = v_b + (R * i_a) + (L * \frac{d\, i_a}{dt}) - e_b \quad (2.14b)$$

Vn can be calculated to be

$$V_n = [(\frac{v_a + v_b}{2}) - (\frac{e_a + e_b}{2})] \quad (2.15)$$

Therefore, the pole voltage (*Vpc*) is equal to

$$V_{pc} = e_c + [(\frac{v_a + v_b}{2}) - (\frac{e_a + e_b}{2})] \quad (2.16)$$

$$(\frac{v_a + v_b}{2}) = \frac{V_{dc}}{2} \quad (2.17).$$

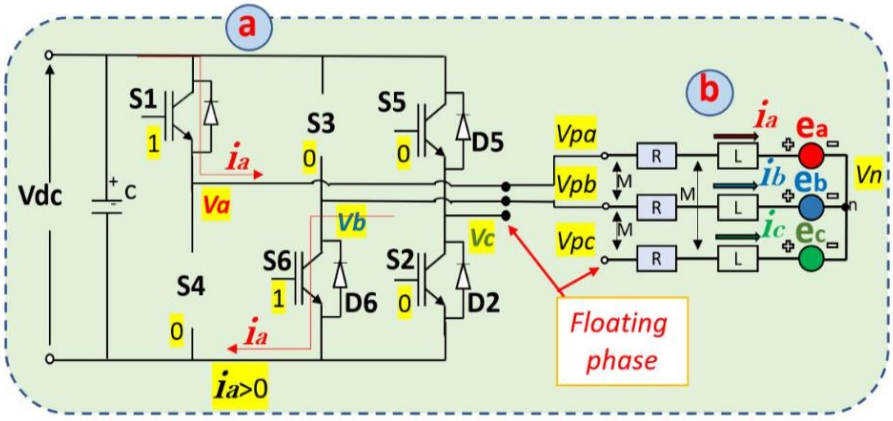


Figure 2-7: (a) Drive inverter with c phase floating (S1 ON, S6 ON), (b) BLDC motor windings

- **C phase floating with S4 ON and S6 ON:** positive current (Ia^{+}) *flows* in the winding of phases A and B as can be observed in Fig.2-8. The same derivation can be used to find the neutral voltage (*Vn*) and the pole voltage (*Vpc*) with the only difference being that

$$(\frac{v_a + v_b}{2}) = 0 \quad (2.18).$$

Equation (2.16) has proven that the pole voltage is essentially a combination of two voltages, the back-emf of the floating phase superimposed on the neutral voltage which appears at the star point of the machine winding. This can be observed in the physical machine in Fig.2-4 (c, d) by the event number (5) and (2).

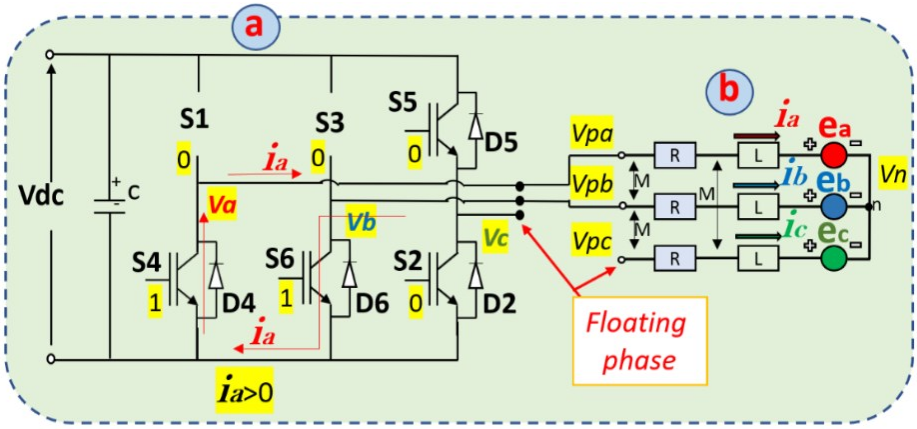


Figure 2-8: (a) Drive inverter with c phase floating (S4 ON, S6 ON), (b) BLDC motor windings

2.2.2.3 Ripple current during the floating phase

During the floating phase of the drive inverter leg, a very interesting phenomenon is happening in the machine. The scenario described in Fig.2-9 can be used to illustrate the concept. When the back-emf (e_c) of the c-phase is positive, the pole voltage will always be positive with respect to the DC ground irrespective of the neutral voltage. However, when this back-emf voltage becomes negative the pole voltage (Vpc) can become negative with respect to the DC ground of the inverter, and this causes the diode D_2 to conduct causing the phase current in the c phase (I_c) to flow. Depending on the neutral voltage value, the (I_c) current increases and decreases, creating a very small ripple which can be observed in the physical machine in Fig.2-5(d) event number (5).

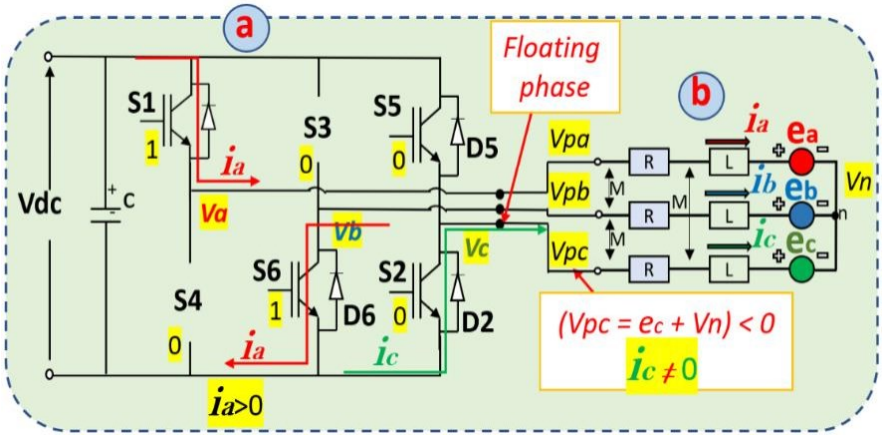


Figure 2-9: (a) Drive inverter with c phase floating (S1 ON, S6 ON), (b) BLDC motor windings

2.3 Improved BLDC motor model

Several BLDC motor models such as developed in [22] [24], are used for the simulation of BLDC motor drives. These BLDC conventional motor models are approximations of the actual machine in which the pole voltage of the floating phase of the BLDC motor winding during the switching

phase of the other phases (winding) is assumed to be equal to the back-emf of the floating phase. This approximation is sufficient in BLDC motor drive simulation, where at any given time knowledge is needed of the two phases or windings to be switched while ignoring the floating phase. However, for simulation purposes where the exact motor pole voltage characteristics are important, the conventional BLDC motor model is not sufficient. For example, for simulation of a sensorless BLDC drive scheme which utilizes back-emf zero cross time detection, reproducing the exact pole voltage is necessary. Therefore, the approach given in [21] is used in this work to develop an improved BLDC motor model which is used for simulation purposes of the BLDC motor drive throughout the rest of this work.

Fig.2-10 shows the block diagram of the improved BLDC motor model. This motor model makes use of a floating signal which is generated by the switching sequence logic and the phase currents to determine the correct pole voltages at the inverter terminals. These pole voltages are then used in the BLDC mathematical equations [2.1-2.18] to find the new currents, torque, and speed. The improved motor model code is given in Appendix 1.

The need for an external (floating) signal in simulation can be explained by the fact that the model needs to know the instant of time when the floating phase starts, hence from the switching sequence block (described in the next chapter), the floating signal is generated then sent to the motor model. As can be observed from Fig 2-10 when the floating signal is detected by the motor model, the previous phase current before the floating phase is checked. If this previous current value is positive, then the pole voltage is clamped to 0, as it can be considered that the bottom diode conducts during that time until the current becomes equal to zero. If on the other hand, the previous

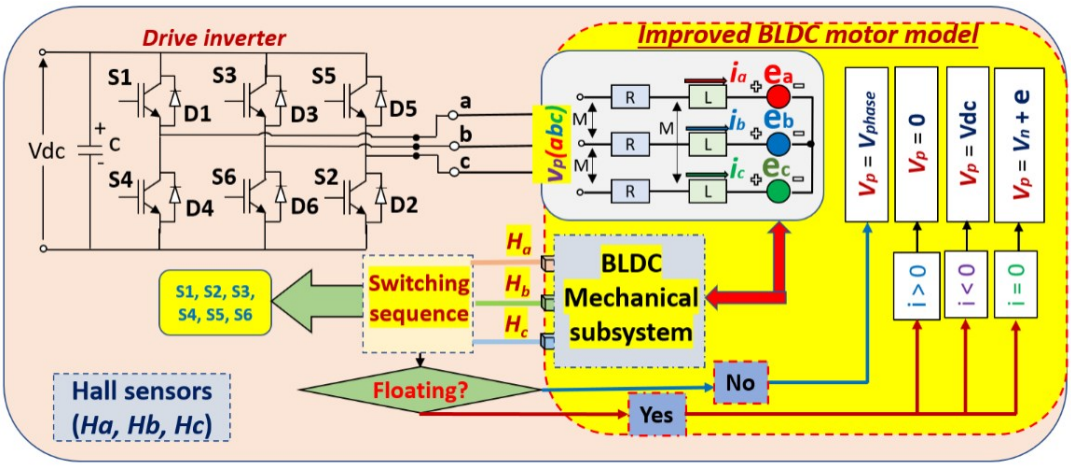


Figure 2-10: Improved BLDC model logic diagram (e is back-emf, v_n is neutral voltage)

value of the current is negative, then the pole voltage is clamped to Vdc, as it can be considered that the top diode conducts during that time until the current is equal to zero. The same logic implemented in Fig.2-6 is used here for the positive and negative currents. Hence, by knowing the previous current value and the floating state, the demagnetization of the commutated phase described previously is implemented in this improved BLDC model.

Zero current detection in the model is achieved by comparing the previous value of current to the current value of current, once a change of sign or zero current is detected, then the demagnetization is considered complete. With zero phase current, the pole voltage becomes equal to the back-emf superimposed on the neutral voltage. Hence, the switching voltage which affects the neutral voltage also appears at the terminal of the floating phase of the BLDC motor. As explained in the previous section, when the floating phase back-emf voltage is positive, no phase current will flow. Once the back-emf becomes negative, a small ripple current flows in the motor as explained in the previous section. Hence, once zero current is detected, the correct pole voltages are assigned and the new phase currents are computed. Fig.2-11 shows the simulated versus the measured pole

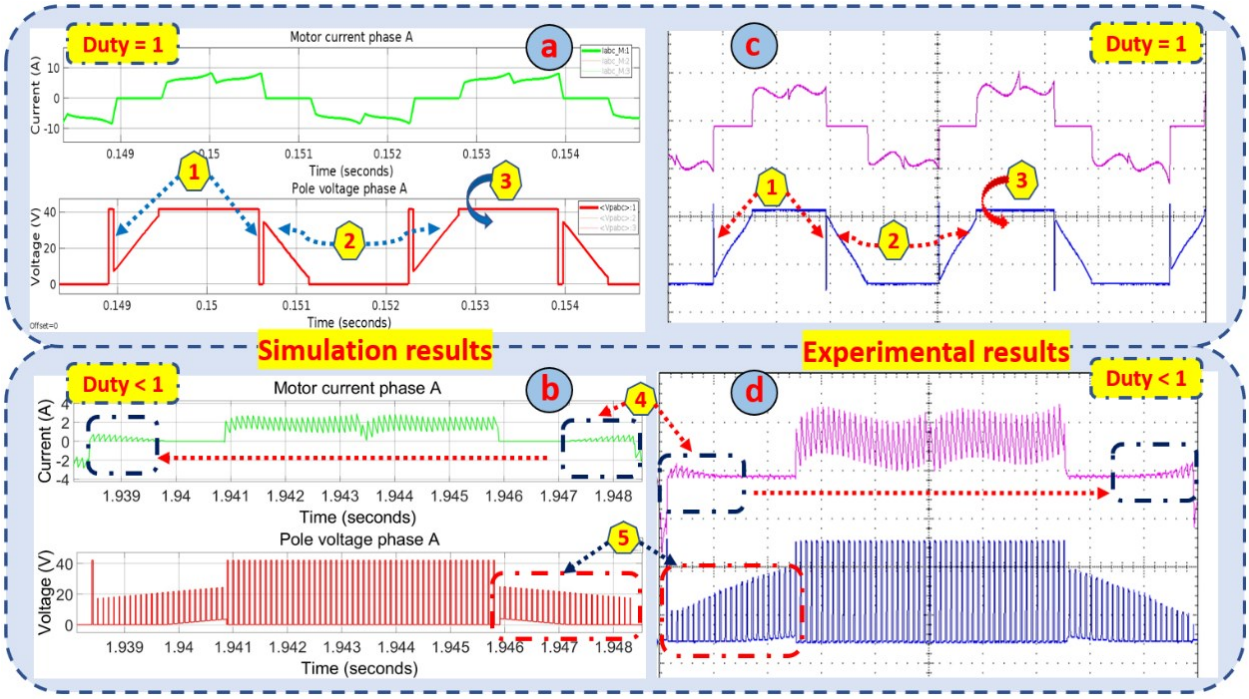


Figure 2-11: (a & b) Simulation results: Green trace: phase current ((10 A/div) (2 A/div)), Red trace: pole voltage (20 V/div). (c & d) Experimental results: Purple trace: phase current ((10 A/div) (2 A/div)), Blue trace: pole voltage (30 V/div).

voltage and current for one of the phases. The salient features of the waveforms in a BLDC drive utilizing trapezoidal control are:

1. Pole voltage during demagnetization of a commutated phase (Clamped to Vdc or 0 volt if the voltage drop of the bottom diode is neglected).
2. Pole voltage during floating condition when duty cycle = 1 (back-emf offset with Vdc/2).
3. Pole voltage when duty cycle = 1 (equals to Vdc).
4. Current ripple in the floating phase (duty cycle <1).
5. Pole voltage of the floating phase when duty cycle < 1 (back-emf superimposed with neutral point voltage).

The simulation results of the improved BLDC motor model are matching with the experimental results, thus validating the model and its implementation. Additional information on the BLDC drive and its implementation is given in the following chapter.

Chapter 3: Simulation and implementation of BLDC motor drive and simulation of PHIL emulation

3.1 Introduction

In this chapter, the implementation of the BLDC motor drive is presented. The simulation of the BLDC drive using the improved BLDC motor model is implemented and validated by comparing the simulation results to the physical BLDC drive at various speed and loading conditions. The PHIL simulation of emulation of permanent magnets motor is introduced and validated.

3.2 BLDC motor drive scheme

PMSM motors can provide a large torque for a wide range of speeds, and is compact in size. Thus, it is the most preferred machine for automotive applications and has gained more popularity over a few decades [1]. Permanent magnet (PM) machines with sinusoidal back-emf are known as PMSM while those with trapezoidal back-emf are known as BLDC machines. Sinusoidal currents are required in PMSM while rectangular currents are required in BLDC to produce constant torque in the machine. BLDC machines are simple to control over the PMSM. To initiate the onset and commutation of current in the phase of a BLDC machine, the beginning and the end of the flat portion of the back-emf has to be tracked, that amounts to only six discrete positions for a three-phase machine in each of the electrical cycle [26]. Hall effect sensors mounted inside the machine facing a small magnet wheel fixed to the rotor having the same number of poles as in the rotor of the BLDC machine can be used to generate signals corresponding to the six discrete rotor sectors. Fig.3-1 shows the six discrete sectors, the back emf, and phase current waveforms for phase-A of an ideal BLDC motor drive.

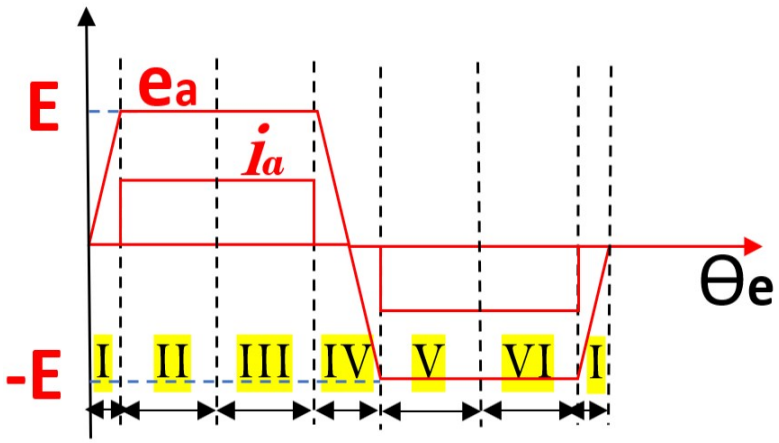


Figure 3-1: Back emf and phase current waveforms of BLDC motor drive

There are various methods of controlling PM motors, the most popular of them being the field orientated control (FOC) and the trapezoidal control methods. FOC is mostly used in PMSM drives, and in general it requires the use of high-resolution encoders which is costly in most cases. Trapezoidal control is preferred for drive applications that do not require high dynamic performance. The trapezoidal control method is simpler and cheaper to implement than FOC. Fig.3-2 shows the switching signals to be sent to the BLDC two-level three-phase drive inverter for each sector of operation of the motor.

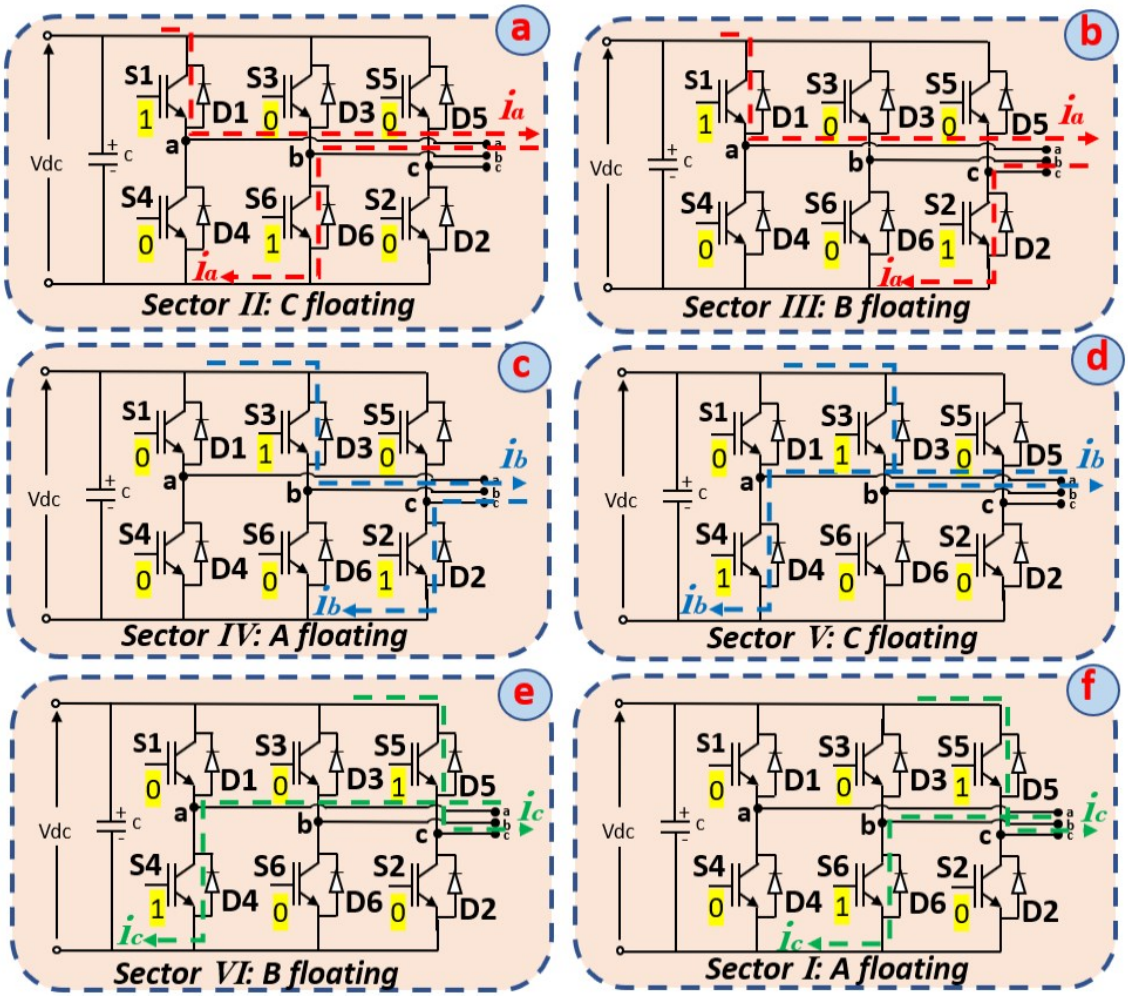


Figure 3-2: (a) Sector II with A phase switching (S1, S4) C phase floating, (b) Sector III with A phase switching (S1, S4) B floating, (c) Sector IV with B phase switching (S3, S6) A phase floating, (d) Sector V with B phase switching (S3, S6) C phase floating, (e) Sector VI with C phase switching (S5, S2) B phase floating, (f) Sector V with C phase switching (S5, S2) A phase floating.

The work presented in this thesis implements sensored trapezoidal control of the BLDC motor using Hall effect sensors. Fig.3-3 shows the block diagram of the entire BLDC drive and the trapezoidal control method implemented in this work.

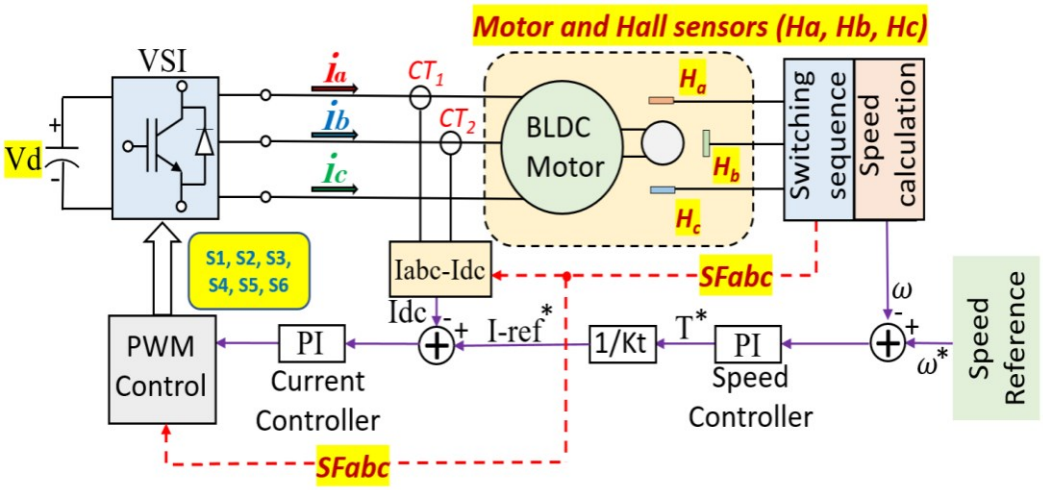


Figure 3-3: Sensored trapezoidal control of BLDC motor

In Fig.3-3 the functions of each block are explained below:

- **VSI block:** This block represents the two-level three phase voltage source inverter (VSI) presented earlier.
- **CT_1 and CT_2:** These are the current transducers (CTs). They are used to measure the phase currents Ia and Ib respectively, with current in phase C being derived from these two currents. These currents are used for the closed-loop current control.
- **Iabc-Idc block:** This block uses the switching function signals SFabc from the switching sequence block to convert the motor phase currents (Iabc) into the Idc which is a current with only the positive side of the Iabc current waveforms. This is done to reduce the number of PI controllers in the current loop. Thus, eliminating the need to have three separate PI controllers for the three motor currents, thus simplifying the control implementation.
- **Speed calculation block:** The speed calculation block computes the actual motor speed using the Hall effect sensor signals and a counter. The calculated speed is used as the speed feedback for closed loop speed control. In simulation, this is not required as the speed is obtained directly from the motor model.
- **The rest of the circuit:** Two proportional-integral (PI) controllers are used, one for the speed loop (outer loop) and the other for the current loop (inner loop). The current PI controller output is the duty cycle command which is used in the PWM block along with the switching function signals SFabc.

The same control logic block diagram as the one shown in Fig.3-3 is used for the simulation and hardware implementation of the BLDC motor drive. The only difference is that an additional signal ("float signal") generated by the switching sequence block is sent to the improved BLDC motor model as discussed in chapter 2.

3.2.1 Voltage Source Inverter (VSI) Simulink implementation

The two-level three phase inverter in Fig.3-2 is implemented in Simulink using multiplication blocks. The input PWM signals (PWMabc) from the PWM control block are multiplied by the DC bus voltage Vdc. This inverter switching voltage is output as (Vabc) and sent to the BLDC motor model. Fig.3-4 shows the Simulink implementation.

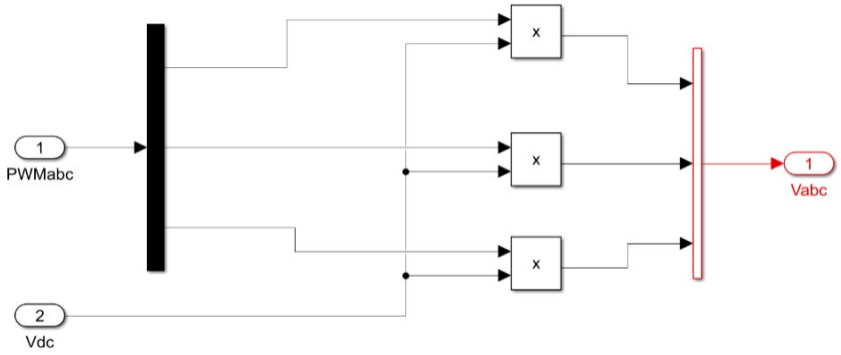


Figure 3-4: Two-level three phase inverter

3.2.2 Switching sequence block

As previously stated, in sensored trapezoidal control of the BLDC motor, the electronic commutation is achieved by using inputs from sensor switches (Hall effect sensors) to determine the position of the rotor. These signals are used to select the motor phase to be switched. To generate the correct switching sequence for the drive inverter, the Hall effect sensor outputs must be synchronized with the back-emf voltage waveforms. Fig.3-5(a) shows the BLDC motor test bench used in this thesis work. In the figure, (1) represents the BLDC motor, (2) the BLDC motor winding terminal outputs, and (3) the Hall sensor output cable. For checking the back-emf voltage and Hall effect sensor outputs, the DC generator is powered and used to drive the BLDC motor (with the BLDC motor terminals open). The BLDC back-emf voltages are measured and displayed on an oscilloscope. The Hall sensors signals are also displayed on the same oscilloscope. Fig.3-5(b) shows the oscilloscope reading of the back-emf voltage of phase-A (e_a) and the Hall sensor outputs (H_a, H_b, H_c). It can be observed that initially, the sensor outputs do not match with the

back-emf voltage, and also the angle between two consecutive Hall sensors signal transitions (Θ_1, Θ_2) are different with one being greater than the other.

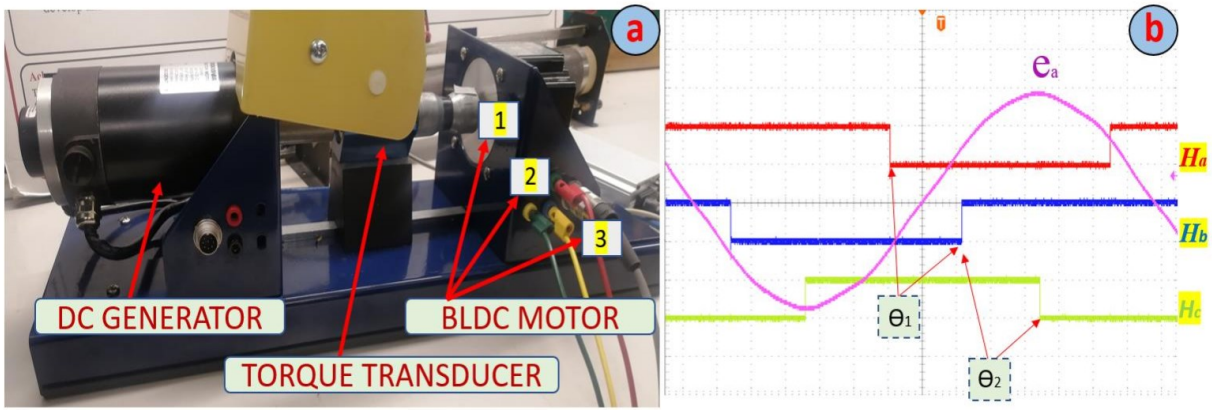


Figure 3-5: (a) BLDC motor testbench MOTORSOLVER, (b) Back-emf voltage and Hall sensor signals

The back-emf voltage waveforms of the BLDC motor used in this thesis are sinusoidal and not trapezoidal as one would expect as shown in Fig.3-6(b). It is common practice in the industry to use permanent magnet machines with sinusoidal back-emf voltage waveforms for testing BLDC motor drives. This is because in practice it is challenging to design a machine with pure trapezoidal back-emf voltage waveforms. Therefore, in this thesis the BLDC motor with sinusoidal back-emf is adopted and used throughout the rest of this thesis for simulation and experimental tests. However, sinusoidal back-emf waveform renders the task of positioning the Hall sensors of the BLDC motor challenging. Using the facility provided by the motor manufacturer as shown in Fig.3-6 the position of the Hall effect sensors is adjusted to provide sensor output signal transition

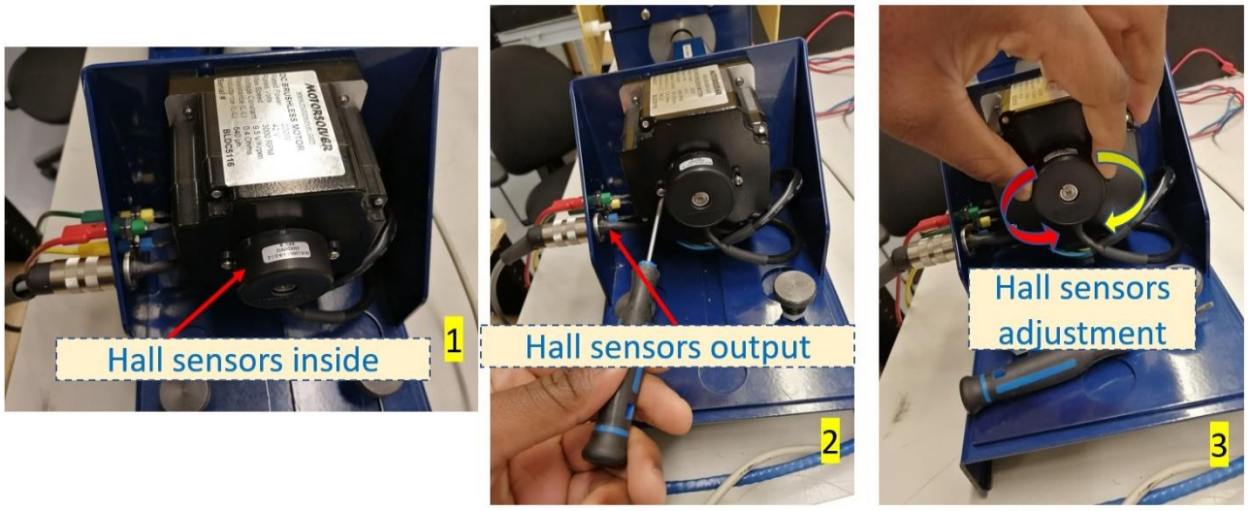


Figure 3-6: Hall sensors adjustment process

for every 60 degrees (electrical) of rotation of the rotor and also matching with the back-emf voltages as shown in Fig.3-7. The steps required to position the Hall sensors are (1) making the necessary connections to the oscilloscope (sensor outputs and back-emf voltage), (2) unscrewing the Hall sensor box, (3) positioning the Hall sensors while checking their output signals and that of the back-emf voltage on the oscilloscope.

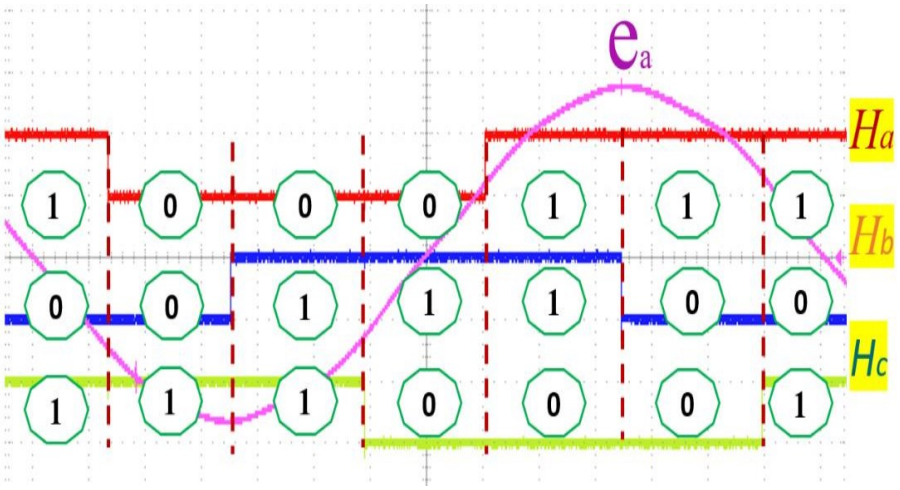


Figure 3-7: Hall sensors output vs BLDC motor back-emf

Using the information provided in Fig.3-11, Table 3-1 can be derived. The switching sequence of the drive inverter switches, the back-emf voltage angle, the floating phases ("F" is used to indicate the floating phase in Table 3-1), the Hall sensor outputs, and the sector of switching corresponds to the explanation given for Fig.3-1, Fig.3-2 and Fig.3-7.

Table 3-1: Switching sequence table (Hardware implementation)

Hall Pattern [*Hc Hb Ha*]	**0 1 1**	**0 0 1**	**1 0 1**	**1 0 0**	**1 1 0**	**0 1 0**
S1	ON	ON	OFF	OFF	OFF	OFF
S2	OFF	ON	ON	OFF	OFF	OFF
S3	OFF	OFF	ON	ON	OFF	OFF
S4	OFF	OFF	OFF	ON	ON	OFF
S5	OFF	OFF	OFF	OFF	ON	ON
S6	ON	OFF	OFF	OFF	OFF	ON
Rotor electrical angle (Θe)	**30°-90°**	**90°-150°**	**150°-210°**	**210°-270°**	**270°-330°**	**330°-30°**
C phase	**F**			**F**		
B phase		**F**			**F**	
A phase			**F**			**F**
SECTOR	**II**	**III**	**IV**	**V**	**VI**	**I**

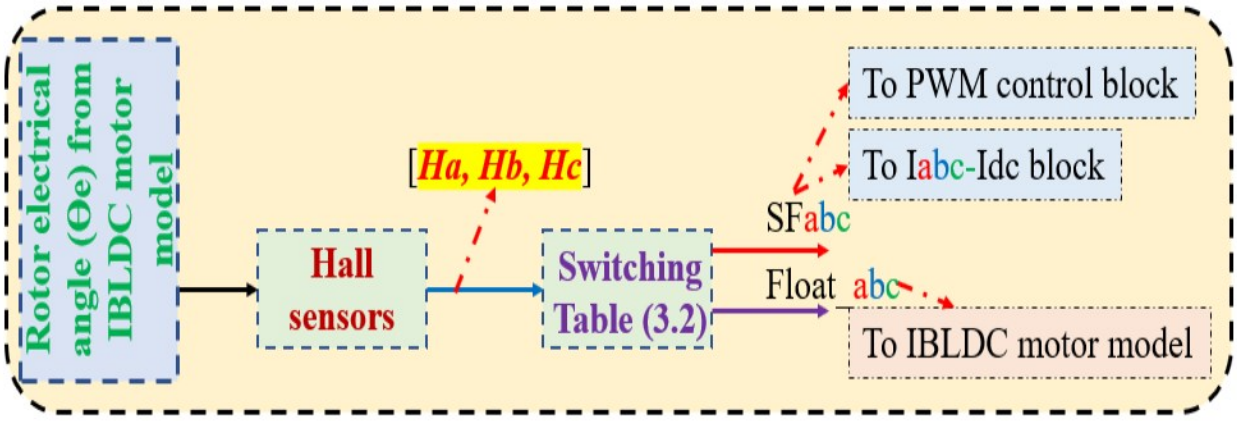


Figure 3-8: Simulink implementation logic of Switching Function block

Fig.3-8 shows the Simulink logic implementation of the switching sequence block described in Fig.3-3. The rotor electrical angle from the improved BLDC motor model is given to the Hall sensors block which senses the input angle and output logic signals as given in Table 3-2. These signals are then used inside the switching sequence block to output the switching signals (SFabc) that go to the current block and the PWM control block as shown in Fig.3-3.

Table 3-2: Switching sequence table (Simulink implementation)

Hall Pattern [*Hc Hb Ha*]	0 1 1			0 0 1			1 0 1			1 0 0			1 1 0			0 1 0		
Rotor electrical angle (Θe)	30°-90°			90°-150°			150°-210°			210°-270°			270°-330°			330°-30°		
SECTOR	II			III			IV			V			VI			I		
SFabc	c	b	a	c	b	a	c	b	a	c	b	a	c	b	a	c	b	a
	0	-1	1	-1	0	1	-1	1	0	0	1	-1	1	0	-1	1	-1	0
Float_abc	Fc	Fb	Fa	Fc	Fb	Fa	Fc	Fb	Fa	Fc	Fb	Fa	Fc	Fb	Fa	Fc	Fb	Fa
	1	0	0	0	1	0	0	0	1	1	0	0	0	1	0	0	0	1

3.2.3 Iadc-Idc block

This block is used to simplify the control algorithm by reducing the number of PI controllers for the current control loop. This is achieved using the logic circuit in Fig.3-9. In trapezoidal control of the BLDC motor, only two motor windings are energized at any time, hence only one current will be positive at any time in the machine. The positive direction of current is considered as the

current leaving the terminal of one of the three legs of the two-level three-phase inverter. The reference current is compared only to this positive current, thus reducing the level of complexity of the control topology. The simulation result of the circuit in Fig.3-9 is shown in Fig.3-10.

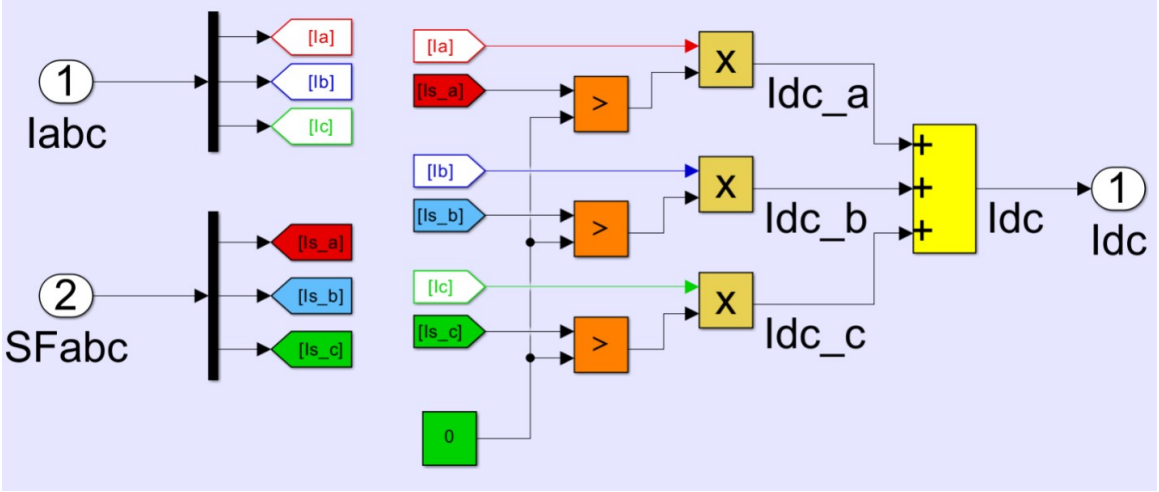


Figure 3-9: Iabc-Idc block Simulink implementation

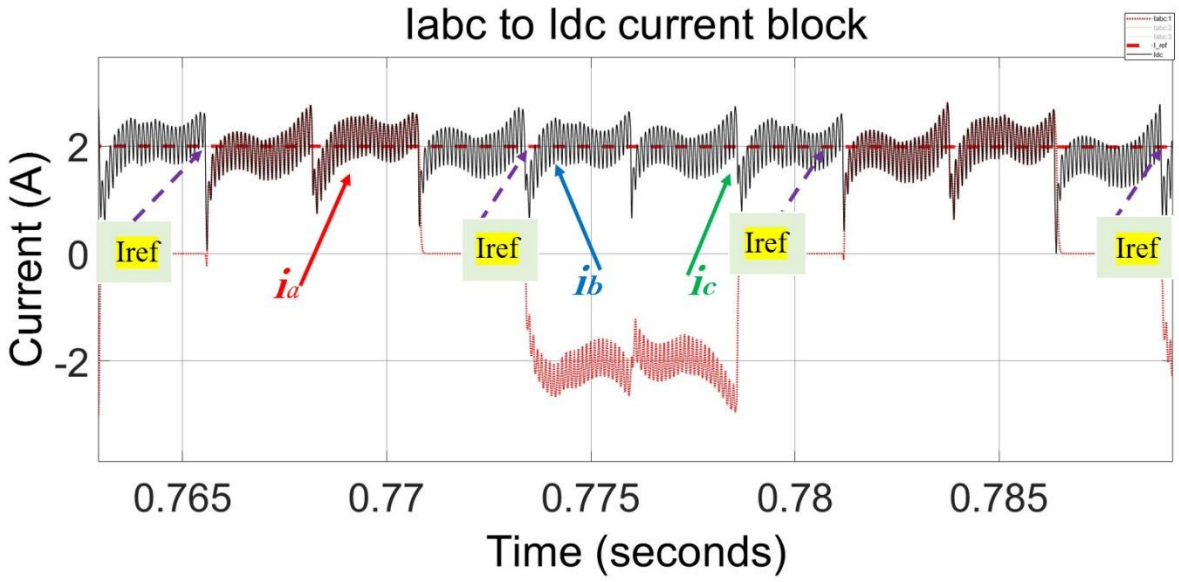


Figure 3-10: Iabc-Idc block simulation result

3.2.4 PWM control block

A pulse-width-modulated (PWM) current control as the one described in [22] is implemented in this work. The PI controller output is multiplied by the switching function signals from the switching sequence generator block (SFabc) to generate three separate signals which are then

compared with a triangular carrier signal to generate the PWM switching signals that are sent to the BLDC drive inverter. Fig.3-11 is the Simulink implementation of the PWM block.

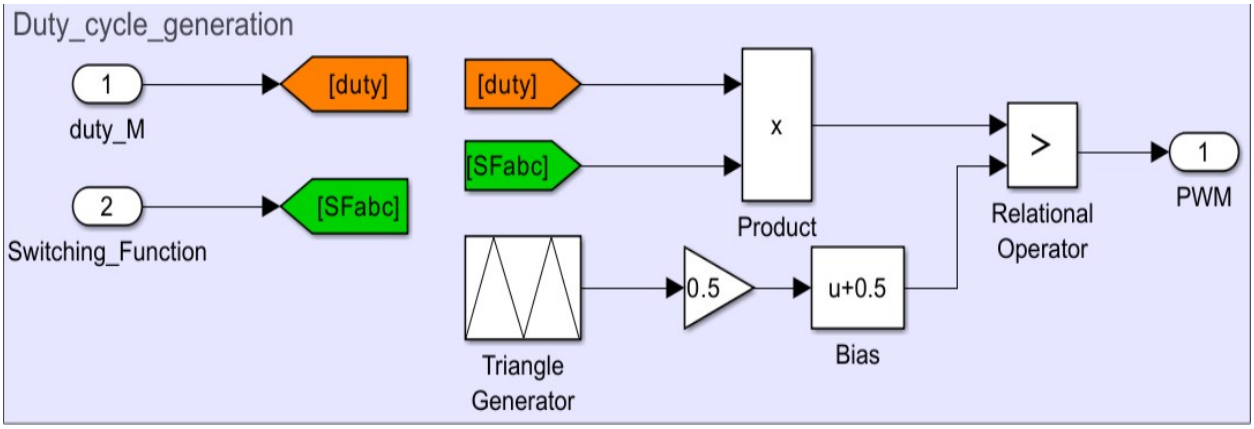


Figure 3-11: PWM control block

3.2.5 PI controller

Fig.3-12 shows the Simulink logic diagram of the PI controller implemented in this work. The approach given in [27] is used to tune the series PI controller topology for the current (inner) and the speed (outer) loop in a cascaded speed control loop as shown in Fig.3-3. In equation 3.1 the PI controller output (Out) is equal to the error times the proportional and integral action (Eout) when (Eout) is less than the saturation value. Once (Eout) exceeds the saturation value, the integral action is disabled through the enable signal (EN) to avoid the integral windup effect, and the output is clamped to the saturation value.

$$Out = Kp\ e(t) + Ki \int e(\tau)d\tau \qquad (3.1)$$

Where Kp is the proportional gain term), Ki integral gain term, e = Error ((Sp ($Set\ Point$) − Pv ($Process\ Variable$)) and τ is the Time.

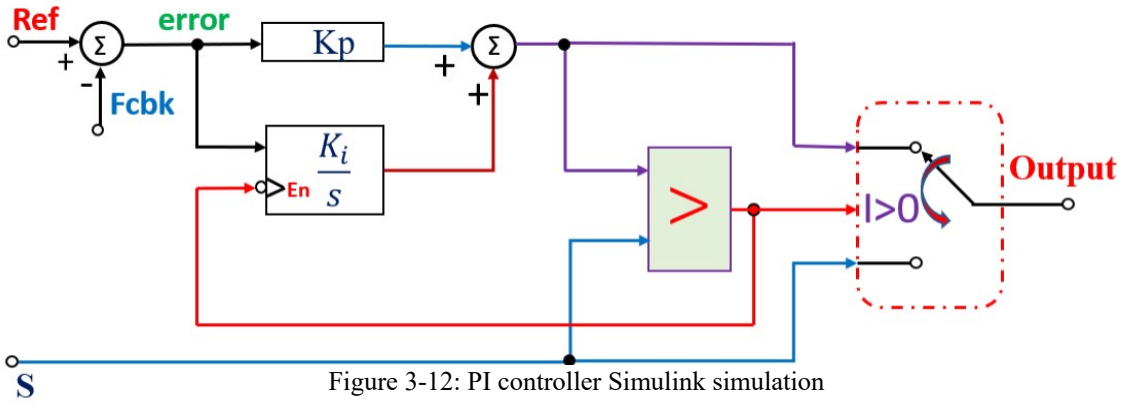


Figure 3-12: PI controller Simulink simulation

3.3 BLDC drive simulation

3.3.1 BLDC motor parameters

BLDC5116 from Motorsolver is the BLDC motor used throughout this thesis. The motor winding inductances, resistances, and back-emf constant were measured experientially and are given in table 3-3.

Table 3-3: BLDC 5116 (Motorsolver)

Rated Power	200 W
Rated voltage	42 V
Rated speed	3000 RPM
Pole	8
R	0.15 Ω
L-M	164 µH
λaf_{L-L}	0.097 V/rad/s

The BLDC motor used in this thesis is connected to a DC generator as shown in Fig.3.6(a), hence the inertia *(J)* and viscous friction constant *(B)* of the entire system must be determined. Using the torque transducer shown in Fig.3-6(a) the motor is run at various speeds on no-load and the torque measurements are recorded to determine the viscous friction constant. A rundown test is performed on the entire system to determine its inertia as shown in Fig.3-13, the average value of the entire system inertia *(J)* is given below. Using the curve fitting technic as shown in Fig.3-14 a relationship between the motor speed (in RPM) and the viscous friction constant *(B)* of the entire is given in equation 3.2. Equation 3.2 is stable for speeds below 1000 rpm, any speed above this speed results in a wrong value of B, equation 3.2 is extrapolated for speeds greater than 1000 rpm to force B to follow a decremental curve as shown in Fig.3-14.

Speed		Te (Nm)	delta_t	J
RPM	Rad/Sec	Nm	seconds	kg-m2
1000	104.7333	0.082	0.4	0.000313
1500	157.1	0.077	0.6	0.000294
2000	209.4667	0.071	0.75	0.000254

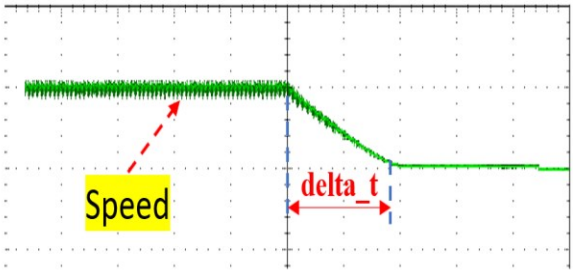


Figure 3-13: Rundown test: Green trace: motor speed (2000 rpm/div)

$J = 0.0003$ kg-m^2.

$$B = (3.437e^{-27} * (speed^8)) - (3.273e^{-23} * (speed^7)) + (1.318e^{-19} * (speed^6)) - (2.925e^{-16} * (speed^5)) + (3.896e^{-13} * (speed^4)) - (3.191e^{-10} * (speed^3)) + (1.58e^{-7} * (speed^2)) - (4.48e^{-5} * (speed)) + 0.006707 \quad (3.2)$$

No-load test

RPM	Rad/Sec	Torque	B
200	20.944	0.043	0.00205
300	31.416	0.044	0.0014
400	41.888	0.047	0.00112
500	52.36	0.05	0.00095
600	62.832	0.052	0.00083
700	73.304	0.054	0.00074
800	83.776	0.055	0.00066
900	94.248	0.056	0.00059
1000	104.72	0.058	0.00055
1100	115.192	0.06	0.00052
1200	125.664	0.061	0.00049
1300	136.136	0.062	0.00046
1400	146.608	0.063	0.00043
1500	157.08	0.064	0.00041
1600	167.552	0.065	0.00039
1700	178.024	0.066	0.00037
1800	188.496	0.066	0.00035
1900	198.968	0.0665	0.00033
2000	209.44	0.067	0.00032

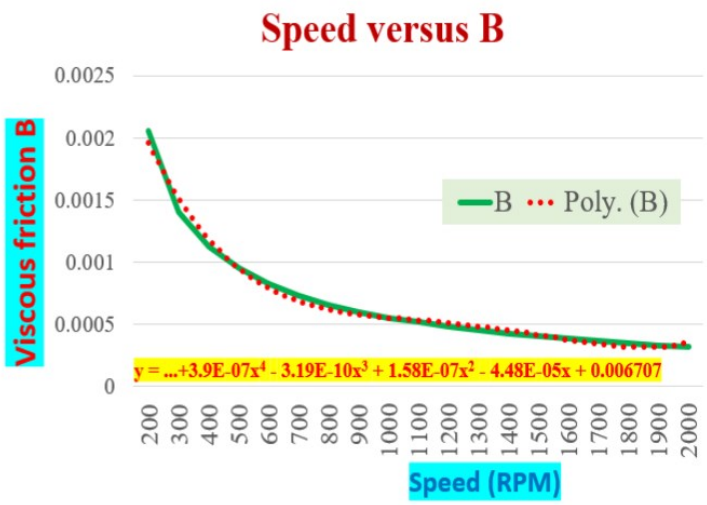


Figure 3-14: Speed to viscous friction B relationship

3.3.2 Speed and current PI controller gains

The speed and current loop PI controller parameters are given in Table 3-4. The tuning process of the PI controller will be discussed in the next section. Because of uncertainty in the values of the inertia *(J)* and viscous friction constant *(B)* of the entire system, an online tuning method was used to obtain these PI controller gains for the speed loop. These controller gains values were not chosen to meet specific drive control requirements, they were set based on observing the motor response to step changes in speed.

Table 3-4: PI current and speed controller gains

Current loop PI controller	
Kp	**0.025**
Ki	**0.08**
Speed loop PI controller	
Kp	**0.02**
Ki	**5**

3.3.3 BLDC drive hardware and simulation test

In this section, all the Simulink blocks described before are put together and all the connections are made as per Fig.3-3. The results of the BLDC drive simulation and the physical machine BLDC drive measurements are compared for different test conditions. Fig.3-15 shows the BLDC drive hardware block diagram.

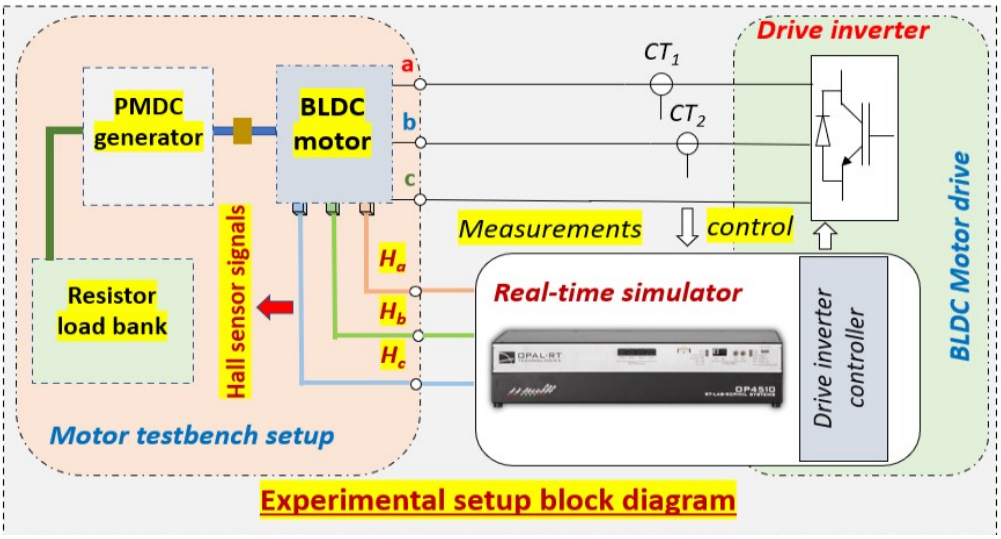


Figure 3-15: Hardware experimental setup block diagram

In the hardware implementation, the BLDC drive control is implemented using the real-time simulator hardware OP4510 from Opal-RT. The time-step used in the real-time simulator is 20 us. The drive switching frequency is 10 kHz and the drive control works at a time-step of 100 us. The drive phase currents are sensed using current transducers (CTs) and are used to close the current loop. The BLDC motor Hall sensors (*Ha, Hb, Hc*) are wired to the OP4510 and are used in the switching sequence block and the speed calculation block. The speed calculation block is implemented inside the OP4510, it uses a counter which is reset at every Hall sensor signal change (which is every 60 electrical degrees) and computes the motor speed. This speed is used as the feedback to close the speed loop. A resistor load bank is connected to the PMDC generator output terminals to apply different loading to the BLDC motor. The discrete fixed-step solver in Simulink is used for simulation implementation in Matlab. The time-step used in the simulation is 20 us, the drive switching frequency is 10 kHz and the drive control works at a time-step of 100 us.

3.3.3.1 BLDC drive simulation results

The drive scheme is simulated for various test conditions and the results are presented in this section:

a. **Speed reversal:** The motor speed is reversed from 1000 rpm to -1000 rpm on no-load, and the motor phase-A current is recorded during the entire test. Fig.3-16 shows the simulation results. Violet trace is the improved BLDC motor current and the green trace is the motor speed. Up on providing the speed reversal command, the motor current reduces to zero and reverses direction. For a short duration, the machine operates in regenerative mode (positive current/torque and negative speed). Once the speed crosses zero and becomes positive, the operation goes back to motoring mode.

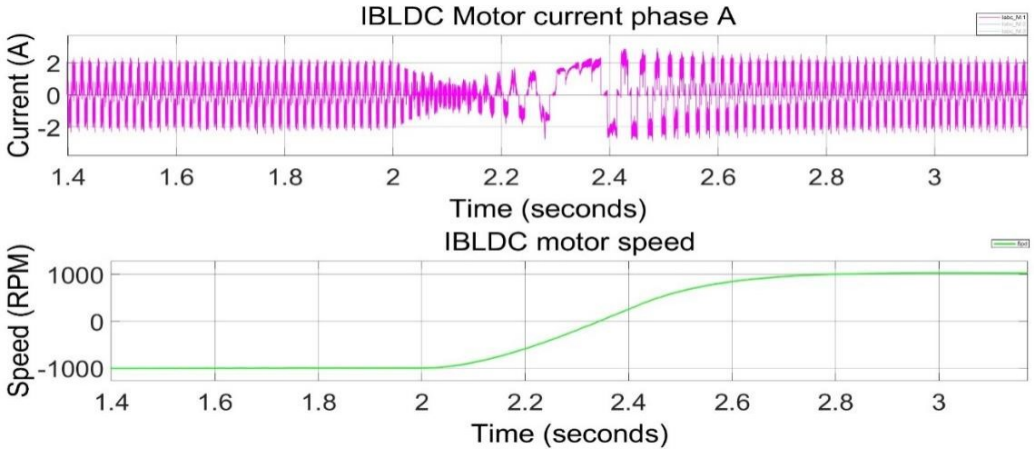


Figure 3-16: Simulation results: Pink trace: phase current (2 A/div), Green trace: speed (1000 rpm/div).

b. **Step speed change:** The motor speed is changed from 500 to 1500 rpm on no-load, and the motor phase-A current is recorded during the entire test. Fig.3-17 shows the simulation results. Violet trace is the improved BLDC motor current and the green trace is the motor

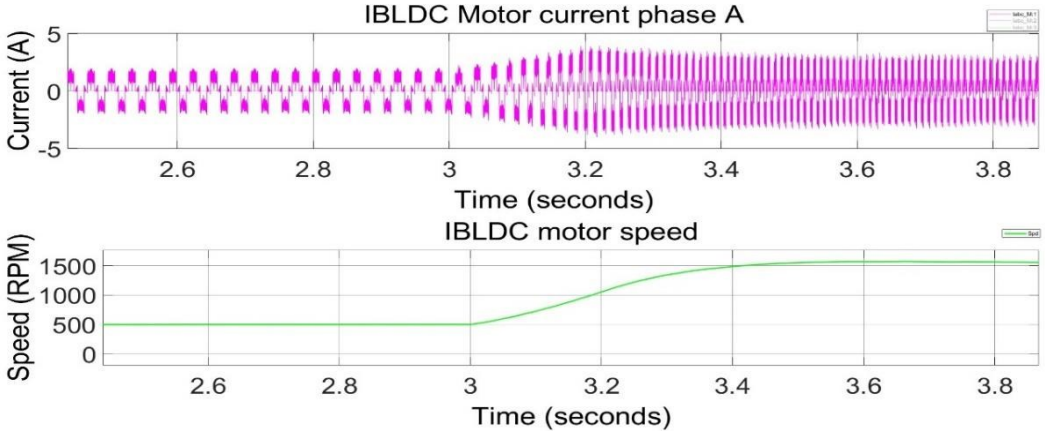


Figure 3-17: Simulation results: Pink trace: phase current (5 A/div), Green trace: speed (500 rpm/div).

speed. The motor current increases to accelerate the motor. Once the speed reaches the set reference (1500 rpm), the motor current reduces to a lower value.

c. **Step load change**: A load torque step is from 0 to 0.5 p.u is applied to the motor while keeping the speed constant at 1500 rpm. The motor phase-A current is recorded during the entire test. Fig.3-18 shows the simulation results. Violet trace is the improved BLDC motor current and the green trace is the motor speed. The motor speed reduces momentarily because of the load. The speed controller increases the current reference to bring back the speed to the set reference.

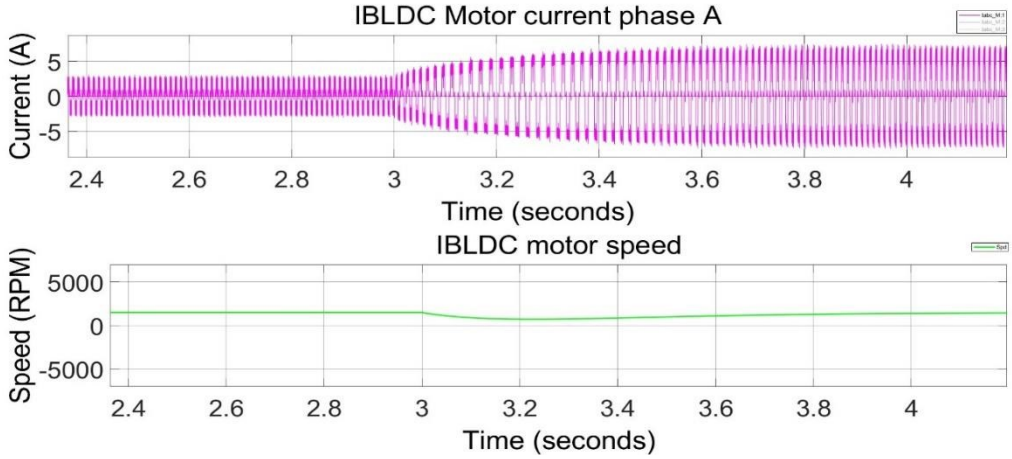


Figure 3-18: Simulation results: Pink trace: phase current (5 A/div), Green trace: speed (5000 rpm/div).

d. **Pole voltage:** One of the important features of the improved BLDC is the ability to reproduce the pole voltage of the waveform in a BLDC motor drive application. Fig.3-19 shows the simulation results for the steady state operation of the drive, when a constant

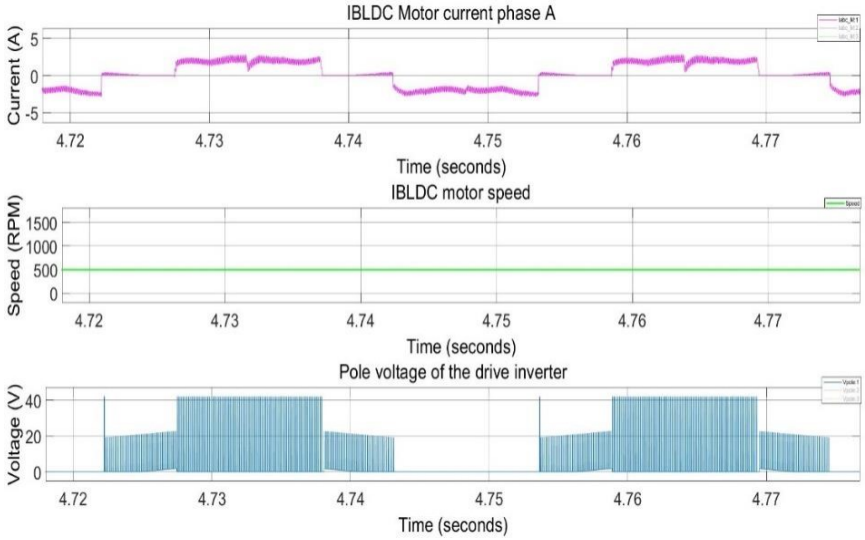


Figure 3-19: Simulation results: Pink trace: phase current (5 A/div), Green trace: motor speed (500 rpm/div), Blue trace: pole voltage (20 V/div).

speed of 500 rpm is maintained at no-load. The effect of the demagnetization of the commutated phase as well as the effect of the PWM signal superimposed on the back emf voltage area clearly visible in the pole voltage. In Fig.3-20, the transient operation of the drive is simulated with a speed change from 500 to 1500 rpm. The motor current increases to accelerate the motor. Once the speed is reached (1500 rpm) the current gradually decreases. The drive controller increases the frequency of the inverter; hence the pole voltage frequency increases accordingly as depicted in Fig.3-20.

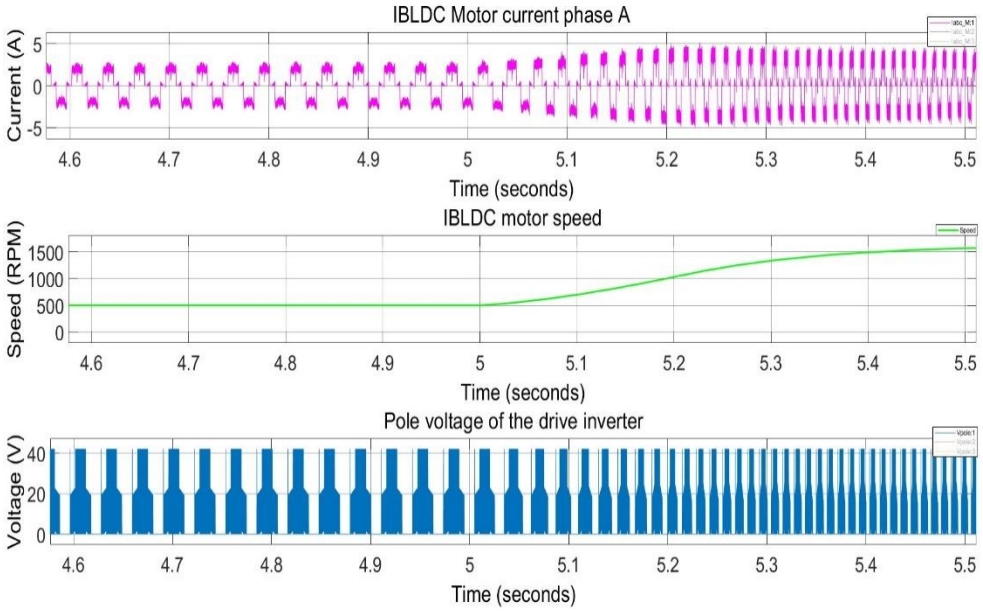


Figure 3-20: Simulation results: Pink trace: phase current (5 A/div), Green trace: motor speed (500 rpm/div), Blue trace: pole voltage (20 V/div).

3.3.3.2 BLDC drive hardware results

Using all the required blocks as described in the previous sections, the BLDC motor drive is implemented and tested. The drive is tested for various test conditions such as:

a. **Speed reversal:** The motor speed is reversed from 1000 rpm to -1000 rpm on no-load, and the BLDC motor phase-A current is recorded during the entire test. Fig.3-21 is the BLDC drive experimental results. Violet trace is the motor current and the green trace is the motor speed.

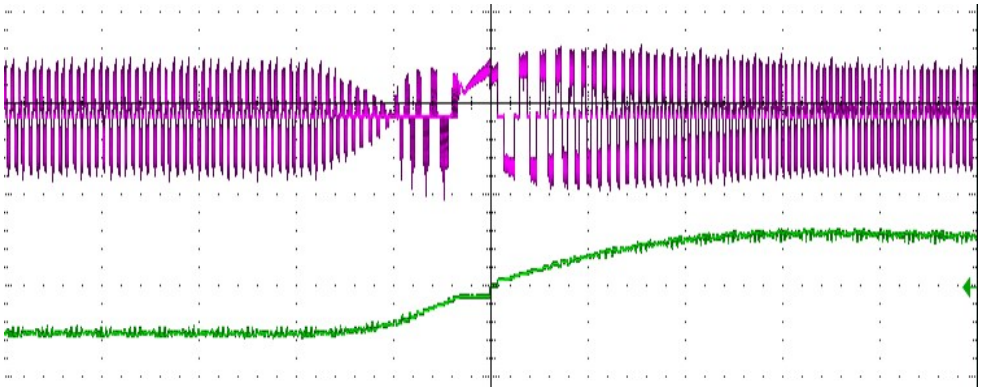


Figure 3-21: Hardware results: Violet trace: phase current (2 A/div), Green trace: speed (2000 rpm/div), time scale: 200 ms/div.

b. **Step speed change:** The motor speed is changed from 500 to 1500 rpm in on-load, and the BLDC motor phase-A current is recorded during the entire test. Fig.3-22 is the BLDC drive experimental results.

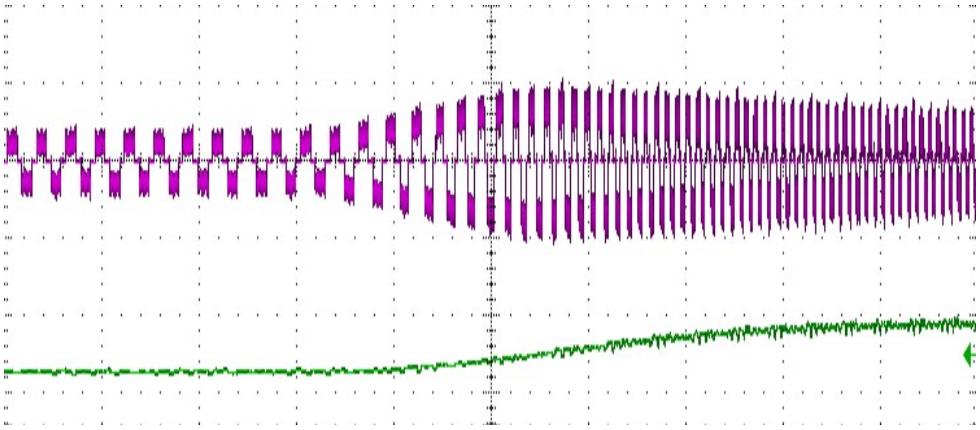


Figure 3-22: Hardware results: Violet trace: phase current (5 A/div), Green trace: speed (2000 rpm/div), time scale: 100 ms/div.

c. **Step load change:** The loading on the motor is changed from 0 to 0.5 p.u using the resistor load bank shown in Fig.3-15 while keeping the speed constant at 1500 rpm. The BLDC motor phase-A current is recorded during the entire test. Fig.3-23 is the BLDC drive experimental results. The speed dips at the moment of applying the load and the controller increases the current to bring back the set speed to the reference speed.

d. **Pole voltage:** Fig.3-24(a) shows the steady state operation of the physical BLDC motor drive while running at 1000 rpm on no-load. Fig.3-24(b) shows the transient operation of the drive with a change of speed on no-load from 500 – 1500 rpm on no-load.

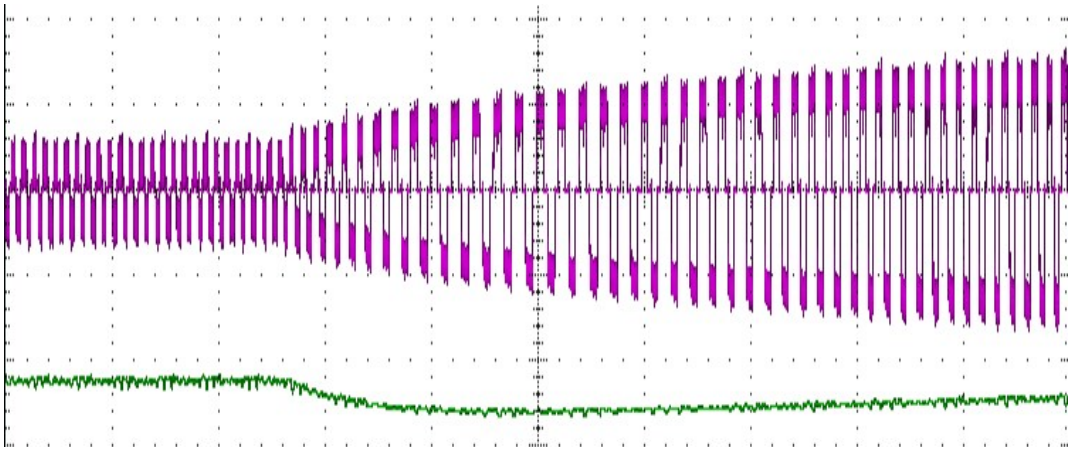


Figure 3-23: Hardware results: Violet trace: phase current (5 A/div), Green trace: speed (2000 rpm/div), time scale: 100 ms/div.

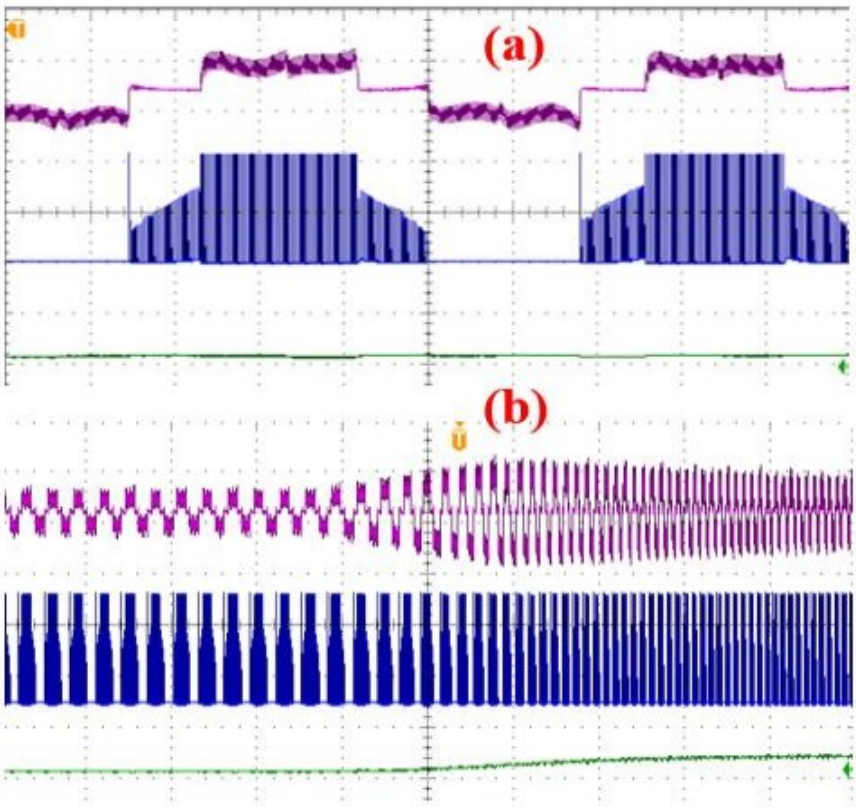


Figure 3-24: Hardware results: (a) Steady state: Violet trace: phase current (5 A/div), Green trace: speed (5000 rpm/div),), Blue trace: pole voltage (20 V/div), time scale: 4 ms/div. (b) Transient state: Violet trace: phase current (5 A/div), Green trace: speed (5000 rpm/div),), Blue trace: pole voltage (20 V/div).time scale: 100 ms/div

The close match between the simulation results with the improved BLDC and the experimental results obtained with the physical BLDC motor drive validates the simulation implementation of the drive.

3.4 PHIL testing method

In a conventional electric drive system testing, the machine to be tested is coupled to a dynamometer test bench. Fig.3-25 shows the conventional drive testing method. The test machine is controlled by the drive controller through the inverter; thus, the machine speed and torque can be controlled by controlling the voltage of the inverter and its frequency. The dynamometer is controlled by its own controller through the dyno inverter. The test machine and the dynamometer are mechanically coupled, this allows the test machine to be tested at various speed and torque conditions by using the dynamometer. Fig.3-25(a) shows the block diagram of the conventional testing method, Fig.3-25(b) shows a picture of an experimental setup in which the test machine (PMSM) is mechanically coupled to the dynamometer.

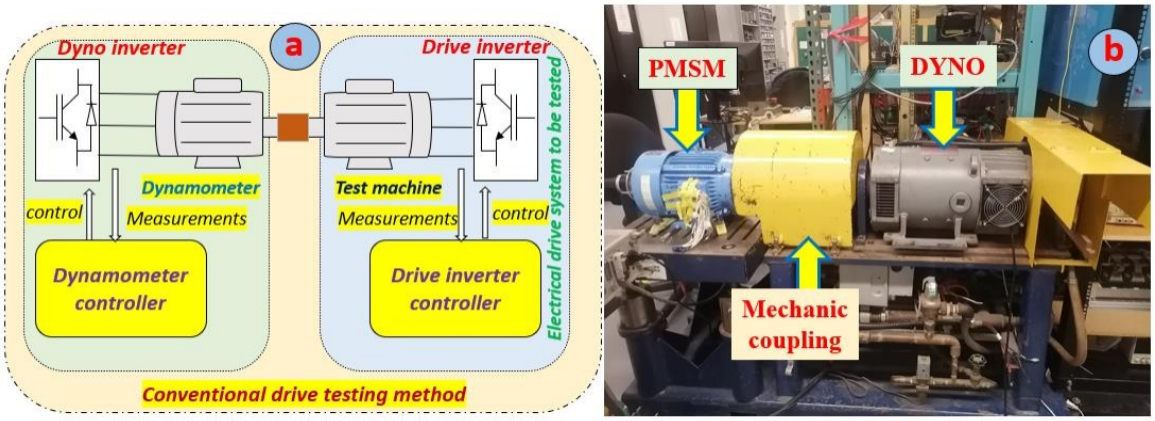


Figure 3-25: (a) Conventional drive testing block diagram, (b) a photo of the conventional testing setup

Power-hardware-in-the-loop (PHIL) drive testing on the other hand utilizes real-time simulation to test a drive system at its rated voltage, current, and power, by mimicking the machine under test using a power electronic converter [10]. This offers the advantage that the drive inverter and its control algorithm can be tested even before the machine used in the drive system is built. PHIL testing method also removes the need for a dynamometer test bench which is costly, requires a lot of space, has ratings and capacity limitations, and care must be exercised with safety because of rotating parts at high speed. A mathematical model of the machine to be emulated is implemented in a real-time emulator control hardware. The model generates the current or voltage reference for the emulator control, therefore the same emulator setup can be used to test various types of machines simply by changing the same emulator setup can be used to test various types of machines simply by changing the models. Fig.3-26(a) shows the conventional versus the PHIL Fig.3-26(b) drive testing method. Therefore, the mechanical testbench is essentially replaced by

an electrical testbench, thus unleashing newer testing possibilities which are not possible with the conventional testing method.

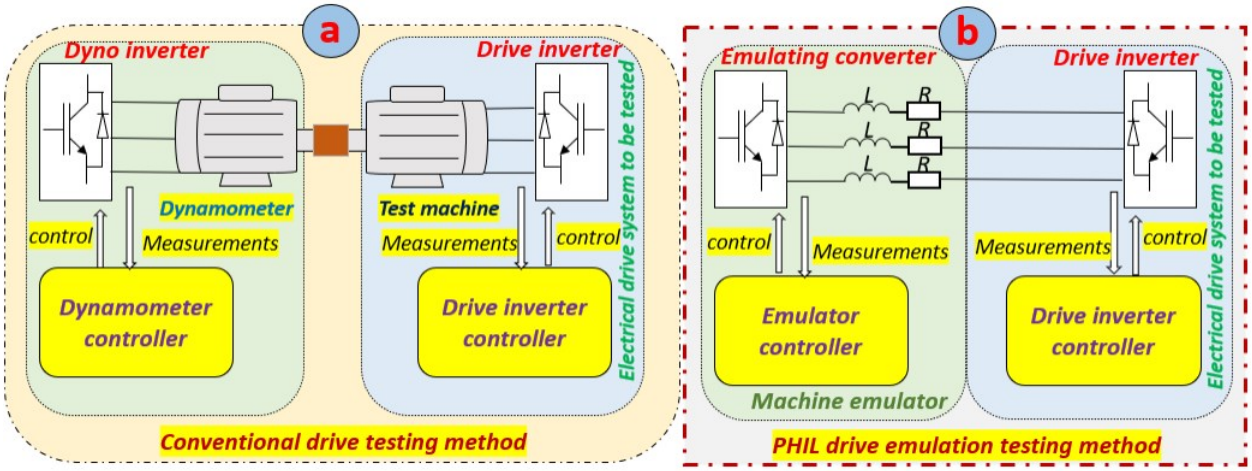


Figure 3-26: (a) Conventional and (b) PHIL drive testing methods

Depending on how the emulator control is implemented, emulator control techniques can be classified in to closed-loop and open-loop control. Closed-loop emulation control method uses voltage-in current-out models for the machine being emulated. The drive inverter output voltage is sensed and given to a machine model. The machine model then generates reference currents which are used by PI current controllers to generate a control signal to control the emulator amplifier. The feedback currents are the drive inverter phase currents.

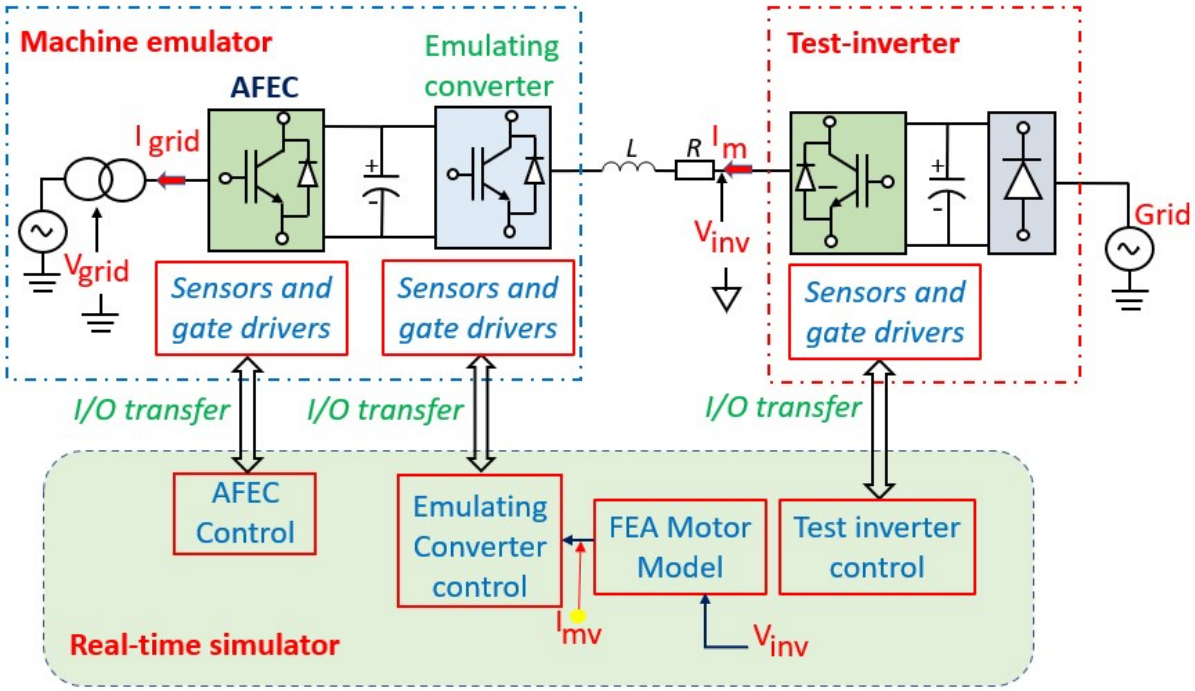


Figure 3-27: Closed-loop emulation control voltage-in current-out model

Fig.3-27 shows a closed-loop emulation block diagram, in this block diagram the voltage from the test inverter is measured (Vinv) and then given to the FEA motor model, this generates the reference currents (Imv). The emulating converter control uses these reference currents to execute a control logic to force the emulating converter to produce voltages (Vemu) that will cause the same currents (Im) as the reference currents to flow in each phase. An active front-end converter (AFEC) is required to regulate the DC-bus voltage of the emulating converter. The isolation transformer is required to avoid circulating current flowing during the test. A detailed discussion on the machine emulator shown in Fig.3.27 is presented in [10].

In the open-loop emulation control method, current-in voltage-out (CIVO) machine models are used. Therefore, the drive inverter phase currents are sensed and provided to the machine model, which generates voltage references that are used to control the emulator amplifier. Open-loop emulation control method mimics the back-emf of the machine using the feedback from the sensed phase currents. Therefore, to accurately emulate a machine, the open-loop emulation method requires that the coupling element inductance and resistance match the winding inductance and resistance of the machine to be emulated.

Fig.3-28 shows an open-loop emulation block diagram, in this block diagram the phase currents (Im) from the test inverter are measured using the current transducers and then given to the CIVO

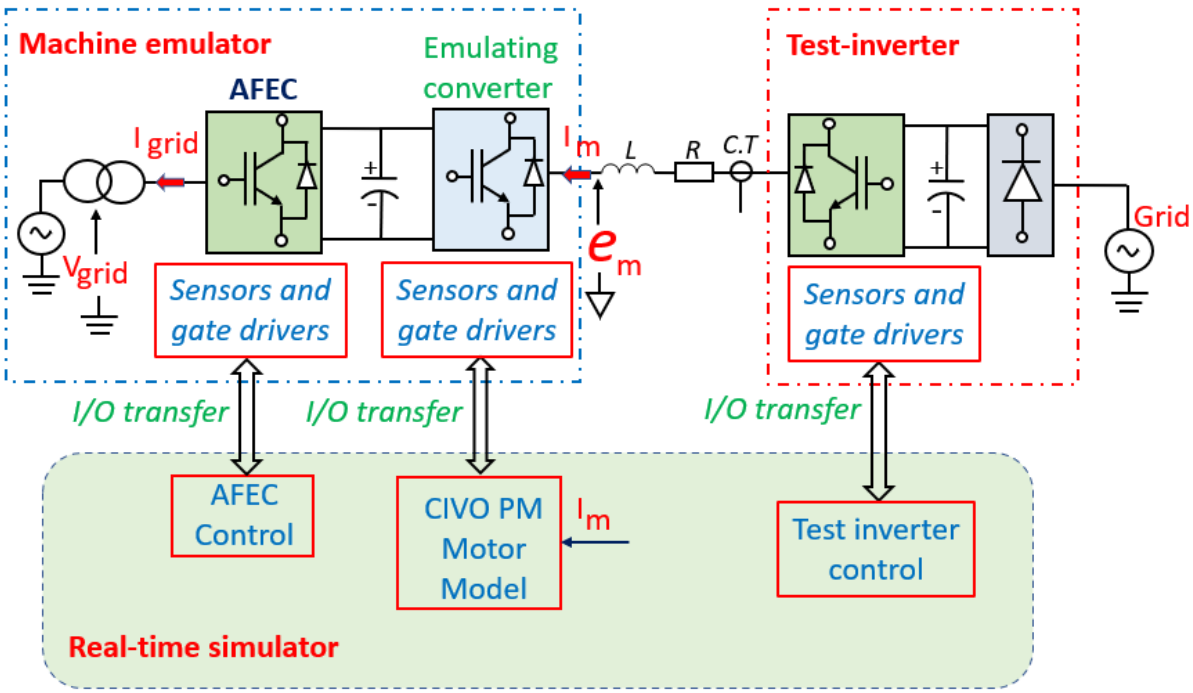


Figure 3-28: Open-loop emulation control current-in voltage-out model

motor model, this generates the reference voltage. The emulating converter is used to amplify these reference voltages which are essentially the back-emf reference voltages of the motor model. Thus, as opposed to the closed-loop in which PI controller are used to control the emulator, in open-loop emulation control the emulating converter output voltages (e_m) which are mimicking the back-emf voltages of the motor are used to control the emulator current. An active front-end converter (AFEC) is required to regulate the DC-bus voltage of the emulating converter. The isolation transformer is required to avoid circulating current flowing during the test.

In general, the accuracy of a PHIL motor emulation depends on [10]:

- **the machine model used:** whether lumped parameter model or FEA-based behavioural motor model. FEA-based behavioural motor model is mostly used when transient behaviour such as slot harmonics, saturation, torque ripple, etc. of the motor must be accurately emulated by the machine emulator. When all the motor design parameters are not available lumped parameter model is preferred.
- **the emulator control strategy used:** whether closed loop or open loop emulation is implemented. For very accurate machine emulation closed loop emulation is preferred, it has the advantage that when a coupling element is used between the machine drive inverter and the emulator, it is immune to machine parameter variation. Open loop emulation has the disadvantage that the coupling element used between the machine drive inverter and the emulator must have the same resistance and inductance value as the machine winding, which is not easy to achieve in practice. However, open-loop emulation is easier to implement than closed-loop, thus for applications that do not require very accurate emulation, open-loop control is the preferred choice.
- **the power amplifier used and its characteristics**: whether switched mode amplifier or linear amplifier is used. To achieve stable and accurate PHIL emulation the choice of the power amplifier for a motor emulation application is of great importance. The choice is based on the required closed-loop motor emulation bandwidth, the accuracy required for motor emulation, and the characteristics of the power amplifier and the test machine. Linear amplifiers have high bandwidth, short delay time, low output noise, and medium efficiency, and are considered to be the best option for machine emulation. However, for

applications where high efficiency, high power, and lower cost amplifiers are required, switched-mode amplifiers are preferred.

- **the coupling element used:** whether L coupling, LCL coupling or transformer coupling, etc. is used. The simplest coupling element which can be used for machine emulation is an L coupling. It has the advantage of the simplicity of control.

This work implements open-loop emulation of a BLDC motor drive employing sensored trapezoidal control. Therefore, a current-in voltage-out motor model is required. Fig.3-29 shows the current-in voltage-out BLDC motor model used in this work. The inputs to this model are the current measurements from the current transducers (CTs). These currents (I_{abc}) are multiplied by the back-emf voltages (e_{abc}) to give the electromagnetic power (Pe) of the machine as per equation (2.6). The electromagnetic torque (Te) is obtained by dividing the electromagnetic power by the previous rotor speed (ω_r) of the machine. Equation (2.7) is then used to find the new rotor speed. Fig.3-30 shows the simulation result of the CIVO BLDC motor back-emf versus the improved BLDC motor model discussed and validated in chapter 2. The two back-emfs voltage are matching, thus validating the CIVO BLDC motor model. It is important to note here that at the beginning of the simulation of the CIVO BLDC motor model, a small value of rotor speed (ω_r = 0.0001, for example) is assigned to it. This is required to allow the back-emf voltages to build-up.

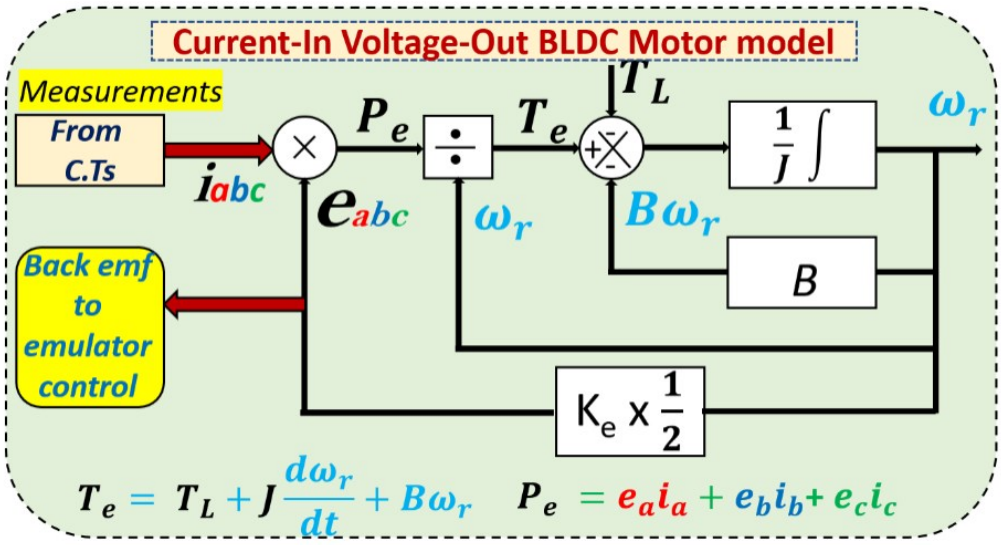


Figure 3-29: Current-in voltage-out BLDC motor model

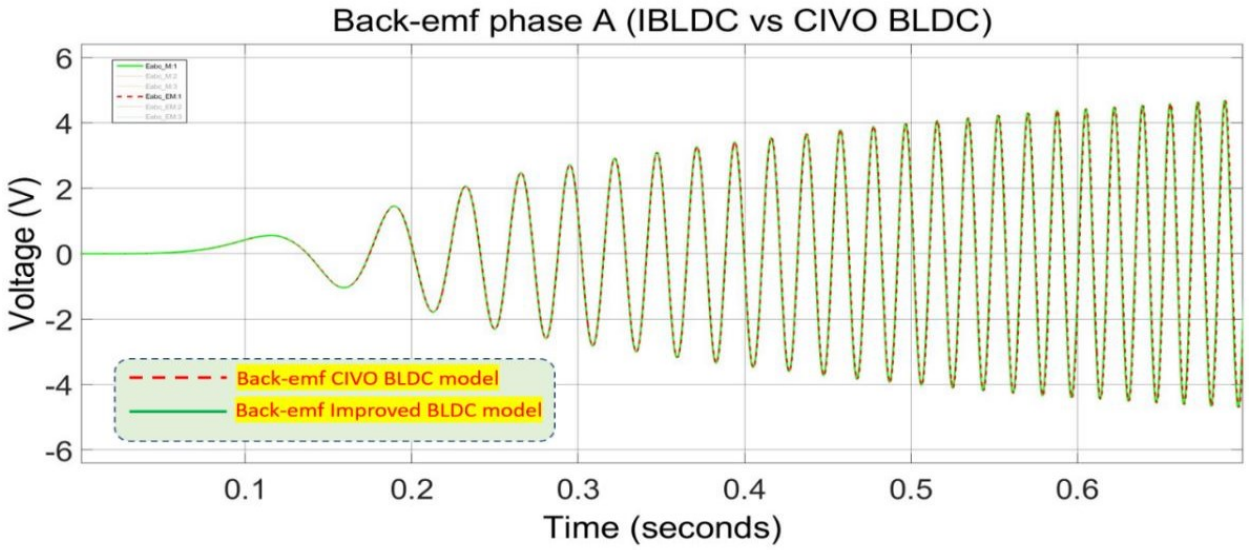


Figure 3-30: CIVO versus improved BLDC motor model back-emfs

3.4.1 PHIL Emulation of Permanent Magnet Motors

There are two types of permanent magnet (PM) motors namely the permanent magnet synchronous motor (PMSM) and the Brushless DC motor (BLDC). FOC is often used in PMSM to control the motor while trapezoidal control is used in BLDC. With current trends towards safer, cost-effective, space-saving, ease of implementation drive testing methods, PHIL motor emulation is quickly growing and increasing its footprint in electric drive testing applications and it is slowly being approved in the industry as an alternative drive testing solution to the conventional dynamometer test method for drive applications, because of its many advantages. The need to accurately mimic the behaviour of a physical electric machine by using a power electronic converter is one of the requirements of PHIL emulation. Much work in the literature has been done on the emulation of permanent magnet machines for closed-loop and open-loop control [10]-[13], however not much work has been done on PHIL emulation of Brushless DC motor. A closed-loop emulation of the brushless DC (BLDC) motor is introduced in [28]. To have a thorough understanding of the PHIL emulation control method, closed-loop simulation of the emulation of PMSM and open-loop simulation of the emulation of BLDC is presented in this section.

3.4.2 Simulation Study of PHIL Emulation of PMSM

The emulation of PMSM in closed-loop control is shown as a block diagram in Fig.3-31. Fig.3-31(a) is the PHIL emulation setup with the drive inverter and the emulating converter which is mimicking the PMSM machine. Since closed-loop control is implemented, a voltage-in current-

out model is used in the emulator. The drive inverter voltages are sensed and given to the PMSM model which generates the current references that are used to control through the PI controller the emulating converter as shown in Fig.3-31(b). Conventional drive control with an outer speed loop and inner current loop with PI controllers is implemented in Fig.3-31(c). The design requirements of the closed-loop control and its control requirements are discussed in [29], therefore this section focuses on the implementation and simulation of the closed-loop emulation of PMSM. DQ model of the PMSM derived in [22] is implemented in this simulation.

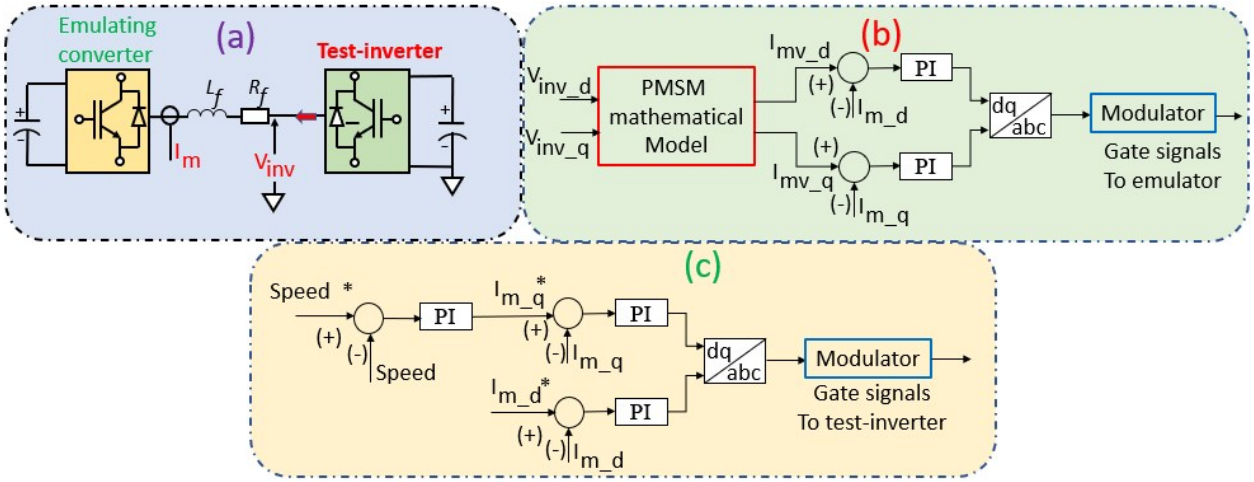


Figure 3-31: (a) PHIL closed-loop motor emulation method with L filter, (b) Emulator PMSM machine model, (c) Drive inverter control

- **Emulating converter PI controller design:**

The first step of the control design exercise is to design the emulating converter current loop; this current controller is designed to have the highest possible bandwidth for the designed emulation system [29]. An RL (R_f, L_f in table 3-5) coupling element is used between the drive inverter and the emulating converter. Table 3-5 gives the PMSM motor parameters and the emulator converter data. Since the back-emf of the Motorsolver BLDC5116 motor is sinusoidal, it is emulated in this section as a PMSM with FOC. Equations 3.3 and 3.4 are the *dq* voltage loop equations of the emulator, and Fig.3-32 is the block diagram of the emulating converter current loop

$$-V_{emu\_q} = R\, i_q + L\frac{di_q}{dt} + \omega L i_d - V_{inv_q} \qquad (3.3)$$

$$-V_{emu\_d} = \mathrm{R} i_d + L\frac{di_d}{dt} - \omega L i_q - V_{inv_d} \qquad (3.4)$$

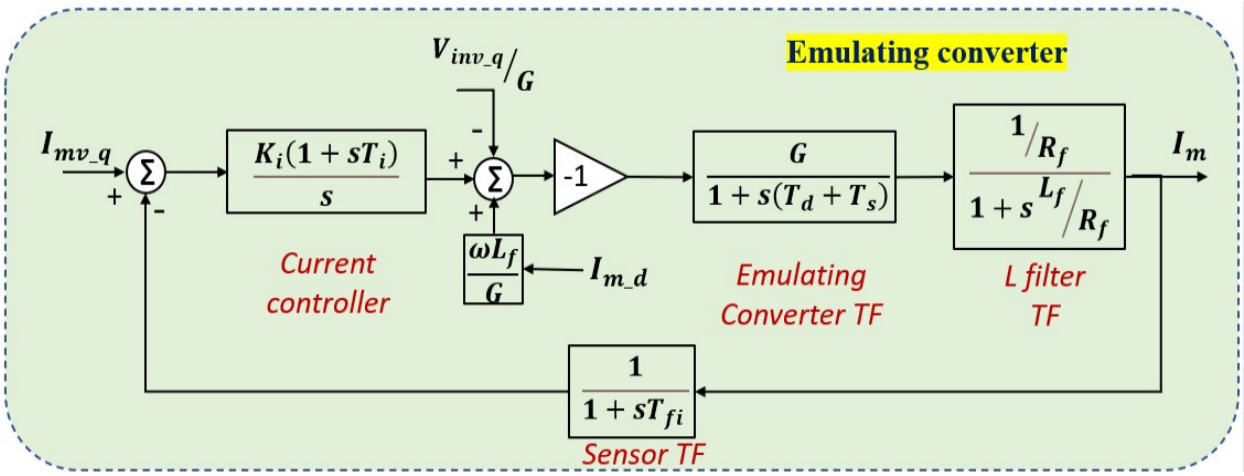


Figure 3-32: Block diagram of the emulating converter current loop

Table 3-5: PMSM motor parameters and Emulator converter parameters

PMSM Motor parameters		Emulator converter	
Rs = 0.15	stator resistance[ohms]	Vdc	350 V
Ld = 0.000164	d axis inductance [H]	Ginv	175
Lq = 0.000164	q axis inductance [H]	Fsw	10 kHz
λafL-L = 0.097	mutual flux linkage line to line due to rotor magnets [Wb.turns]	Rf	0.3
J = 0.0003	rotational inertia [kg.m2]	Lf	0.003
B = 0.00028	friction coefficient [Nm/rad/s]	Tfi	20uS
P = 4	number of poles		
Ginv = 21	Drive inverter		
ω_r= 3000 rpm	Rated speed		
V_r = 42 V	Rate voltage		

From the information given in table 3-5, the bandwidth of the emulating can be calculated to be:

$$T_d = \frac{1}{(2\ X\ Fsw)} = 50\ uS,$$

$$2\zeta\omega_n = \frac{1}{(T_d + T_{fi} + T_s)} = 11\ 111.11,\ with\ \zeta = 0.707,\ \omega_n = 7857.93\ rad/\text{sec}\ = 1250\ Hz\ .$$

$$\omega_n{}^2 = \frac{Ki * Ginv_{emu}}{Rf(Td + Ts + Tfi)} = \frac{175 * Ki}{0.3 * 90 * 10^{-6}}\ ,hence\ Ki = 9.53.$$

$$T_i = \frac{Lf}{Rf} = 0.01 \ , hence \ K_p = \ T_i * K_i = 0.095.$$

Where: **Td** is the time constant of the emulating converter,

Ts is the sampling time (20 us) and Tfi is the current sensor time constant,

Ki and **Kp** are the PI current controller gains for the emulating converter current loop.

With ζ=0.707, the system desired bandwidth is 7876 rad/sec. Using the calculated Kp and Ki values, the bode plot of the system achieves an emulating converter system bandwidth of 6970 rad/sec. The values of Ki and Kp are adjusted to increase the bandwidth, hence Ki = 11 and Kp = 0.11 are chosen. This results in an emulating converter bandwidth of approximately 7650 rad/sec which is close enough to the desired bandwidth value as shown in Fig.3-33. The drive inverter bandwidth is selected to be 4 times lower than the emulating converter bandwidth. This corresponds to the bandwidth shown in Fig.3-33.

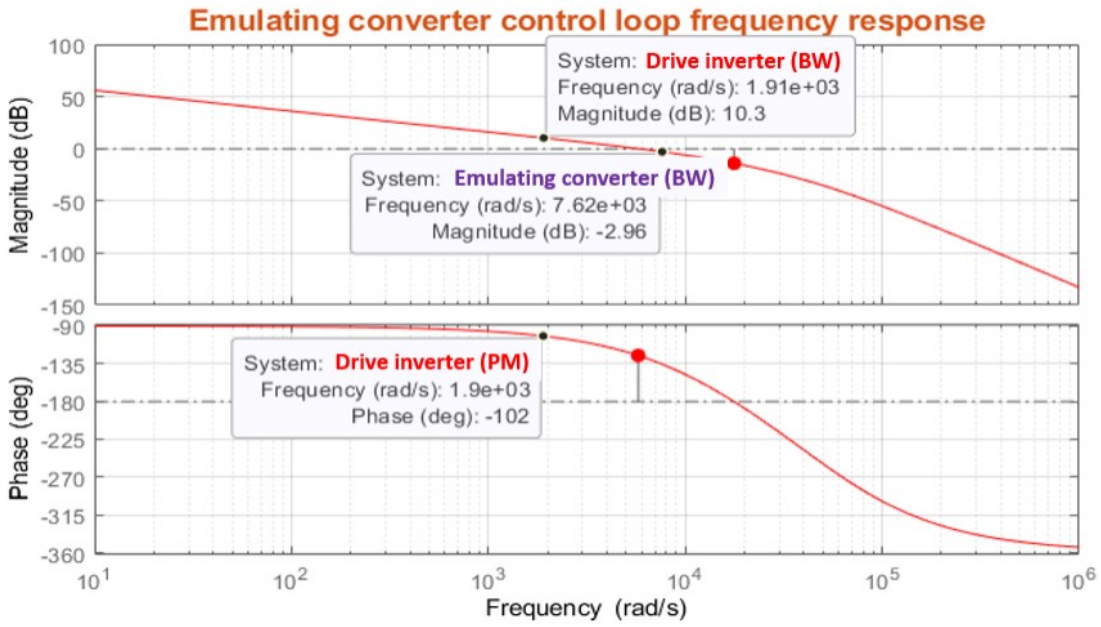


Figure 3-33: Emulating converter closed loop frequency response

- **Drive Inverter PI controller design**

The tuning process of the speed and current controller gains of the drive inverter is done using the magnitude and phase plots of the emulating converter current control loop frequency response. To ensure control stability between the drive inverter control loop and the emulating converter control

loop, the emulating inverter current bandwidth must be at least 4 times higher than the drive inverter current bandwidth. Hence, the tuning of the speed and current loop PI controller gains is done to meet the aforementioned control requirement. Equations 3.5 and 3.6 below are the voltage loop equations of the PMSM *dq* model. Fig.3-34 shows the block diagram of the drive inverter current loop.

$$V_{drive\_d} = R\, i_d + Ld\frac{di_d}{dt} - \omega_s L_q i_q \qquad (3.5)$$

$$V_{drive\_q} = R\, i_q + Lq\frac{di_q}{dt} + \omega_s L_d i_d + \omega_s \lambda_{af} \qquad (3.6)$$

The tuning process of speed and current PI controllers of PMSM is extensively discussed in [28][29], and the same approach is used in this thesis to tune the speed and current PI controllers of all permanent magnet motors. Using the motor parameters data from table 3-5, the design is as follows:

$$Drive_{inv\_BW} = \frac{Emu_{bw}}{4} = \frac{7650}{4} = 1912.5\ rad/\text{sec} = 304.38\ Hz.$$

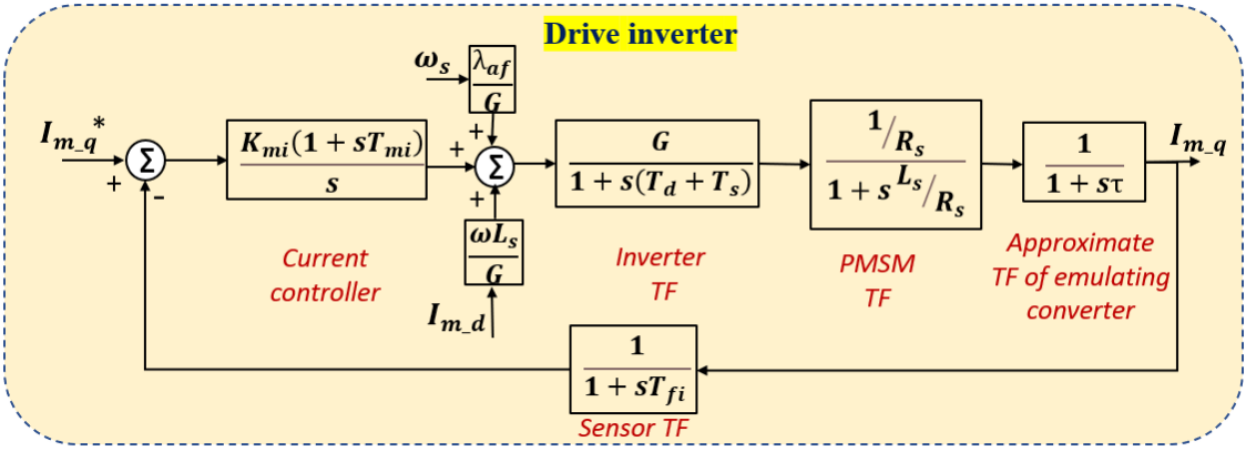


Figure 3-34: Block diagram of the drive inverter current loop

For the speed loop, δ is selected to be equal to 4, where " δ " is defined as the damping factor. The larger δ is, the further apart the zero-corner frequency and the current loop pole will be and the further apart they are, the phase margin is allowed to peak to a higher value in between these frequencies, this improves stability at the expense of speed loop bandwidth [28]. The PI controller gains can be calculated to be:

- **Current loop**

$$T_q K_p = \frac{Lq * Drive_{inv_{BW}}}{Ginv} = 0.015$$

$$T_q K_i = \frac{Rs}{Ls} = \frac{0.15}{0.000164} = 914.63$$

- **Speed loop**

$$K = \frac{3 * P * \lambda_r}{4 * J} = 1940$$

$$Spd\_K_p = \frac{TqKp}{K * Ls * \delta} = 0.012$$

$$Spd\_K_i = \frac{T_q K_p}{(\delta^2 * Ls)} = 5.63$$

3.4.3 Simulation Results for Emulation of PMSM

The results for closed-loop emulation of a PMSM are presented here. The simulation is performed in Simulink. A discrete fixed-step solver in Simulink is used and the time-step used in the simulation is 20 us. The PMSM drive inverter (inverter gain Ginv = 21) and the emulator drive converters (inverter gain Ginv = 175) are implemented as linear amplifiers (only the inverter gain is used), to simplify the implementation. The emulator voltage is set as a constant and hence an active front-end converter is not included in the simulation. Fig.3-35 is an incremental speed change from 500 to 1500 rpm on no-load, the machine current (red trace) and the emulator current

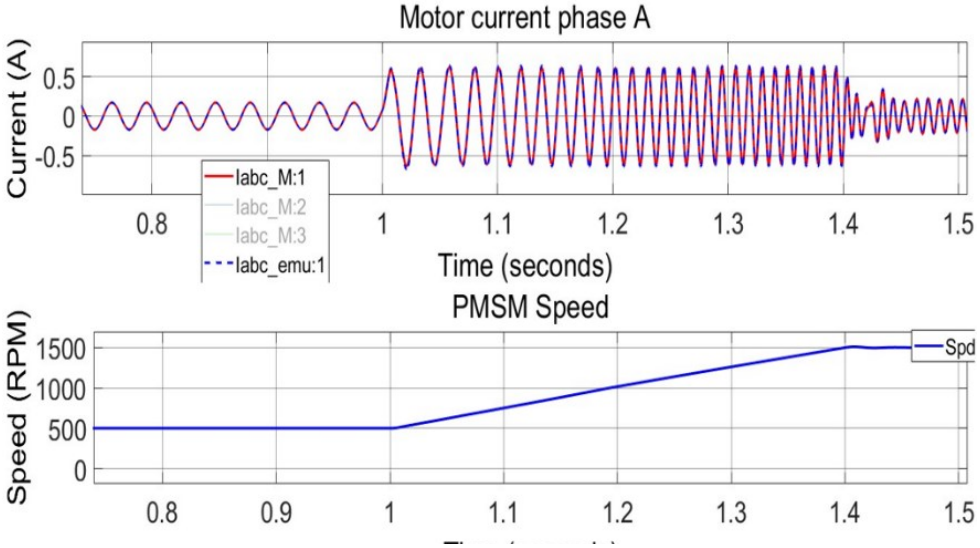


Figure 3-35: Speed change test from 500 – 1500 rpm on no-load. Red trace: PMSM current (0.5 A/div), dashed blue trace: Emulator current (0.5 A/div), solid blue trace: machine model speed feedback (500 rpm/div).

(dashed blue trace) are matching. Fig.3.36 is the starting transient of the PMSM motor and the emulator for a ramp speed input, the machine current (red trace) and the emulator current (blue trace) are matching. Fig.3-37 is a step load input from 0 to 0.5 p.u, the machine current (red trace) and the emulator current (blue trace) are matching.

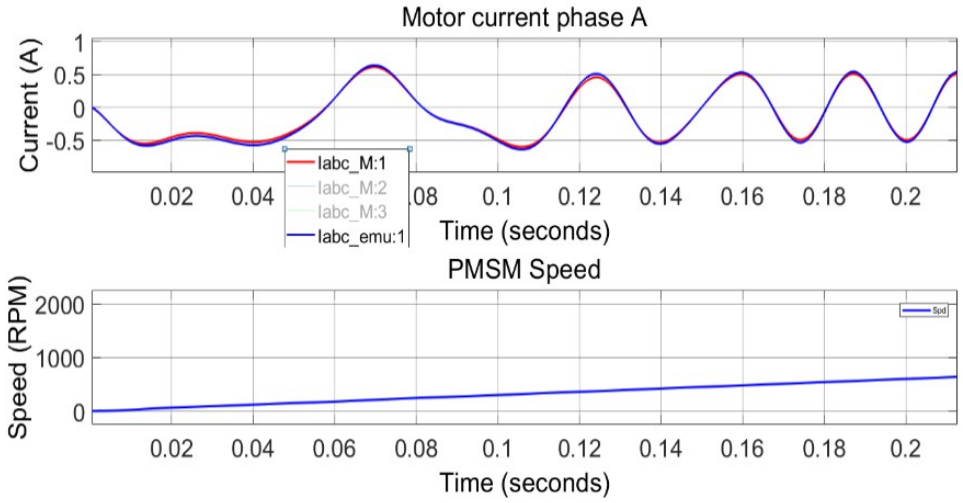


Figure 3-36: Starting transient on no-load. Red trace: PMSM current (0.5 A/div), blue trace: Emulator current (0.5 A/div), solid blue trace: machine model speed feedback (1000 rpm/div).

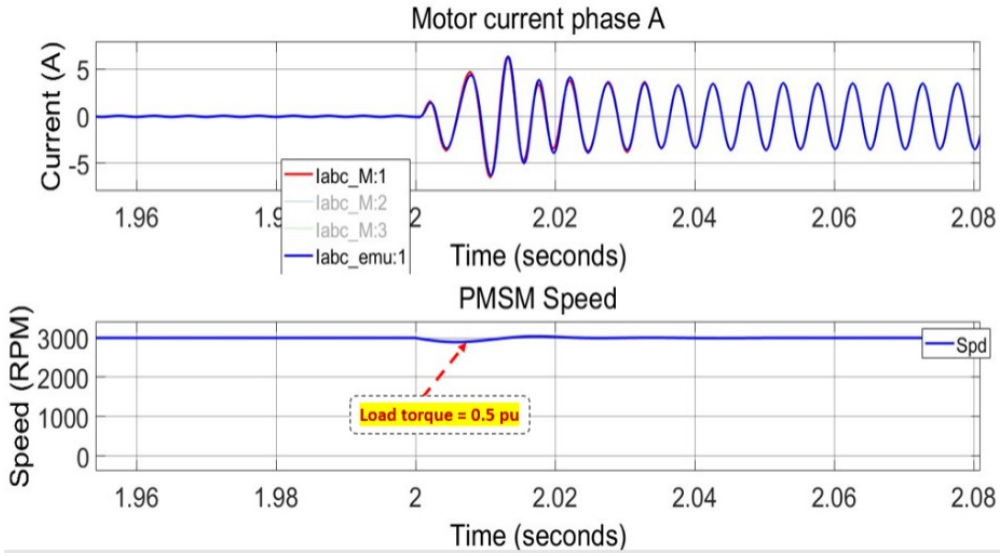


Figure 3-37: Load change test from 0 – 0.5 p.u at 1500 rpm. Red trace: PMSM current (5 A/div), dashed blue trace: Emulator current (0.5 A/div), solid blue trace: machine model speed feedback (1000 rpm/div).

3.4.4 Simulation of Open-Loop Emulation of BLDC

3.4.2.1 PI controller tuning

In the closed-loop emulation method, the bandwidth for the emulator control is chosen to be 4 times higher than that of the drive converter. This control constraint between the emulator

converter controller and the drive inverter controller does not exist in open-loop emulation as the emulating converter is only used to emulate the back-emf of the physical motor. A similar approach as the one described in the section on drive inverter PI controller design is used to find the PI controller gain values, the only difference being that the requirement to meet the emulator controller bandwidth is ignored. Depending on the drive inverter speed-loop bandwidth (BW_s) requirements, equation 3.7 can be used to find the current-loop bandwidth (BW_c), and the PI controller gains can then be computed as discussed in [28].

$$BW_c = \frac{K_p{}^{series}}{L} = BW_s\,(\delta + 2.16 \times e^{-\frac{\delta}{2.8}} - 1.86)\ (\frac{rad}{sec}) \qquad (3.7)$$

3.4.2.2 Simulation Results For Emulation of BLDC Motor

The improved BLDC motor (voltage-in current-out) model with its drive is simulated concurrently with the emulated BLDC motor (current-in voltage-out) model and its drive. The two motor drives receive the same speed command and load torque command and the two-phase currents are compared. Fig.3-38 shows the block diagram of the simulation of the emulation. The RL coupling between the drive inverter and the CIVO BLDC motor model (Fig.3-38(b)) is implemented with floating state inputs. The same technique described for the IBLDC is used to ensure that the correct phase currents are output from the RL coupling current output.

A discrete fixed-step solver in Simulink is used for simulation implementation in Matlab, the time-step used in the simulation is 1 us. The improved BLDC motor model implemented in the simulation works at a time-step of 1 us, and the drive control of the BLDC motor drive in Fig.3-38(a) works at a time-step of 100 us. The CIVO BLDC motor model implemented in the simulation works at a time-step of 20 us (to match the real-time hardware simulator time-step), and the motor emulator control works at a time-step of 100 us.

Fig.3-39 shows the simulation results of the phase-A current of the IBLDC and emulator current for a speed reversal command from -1000 rpm to 1000 rpm at no-load. Fig.3-40 shows the simulation results of the A phase current of the IBLDC and emulator current for a speed change command from 500 rpm to 1500 rpm at no-load. Fig.3-41 shows the simulation results of the A phase current of the IBLDC and emulator current for a step load from 0 to 0.5 p.u at a constant speed of 1500 rpm. It is evident that the open-loop emulator currents are matching the improved BLDC machine currents, hence validating the results of the simulation of the emulation.

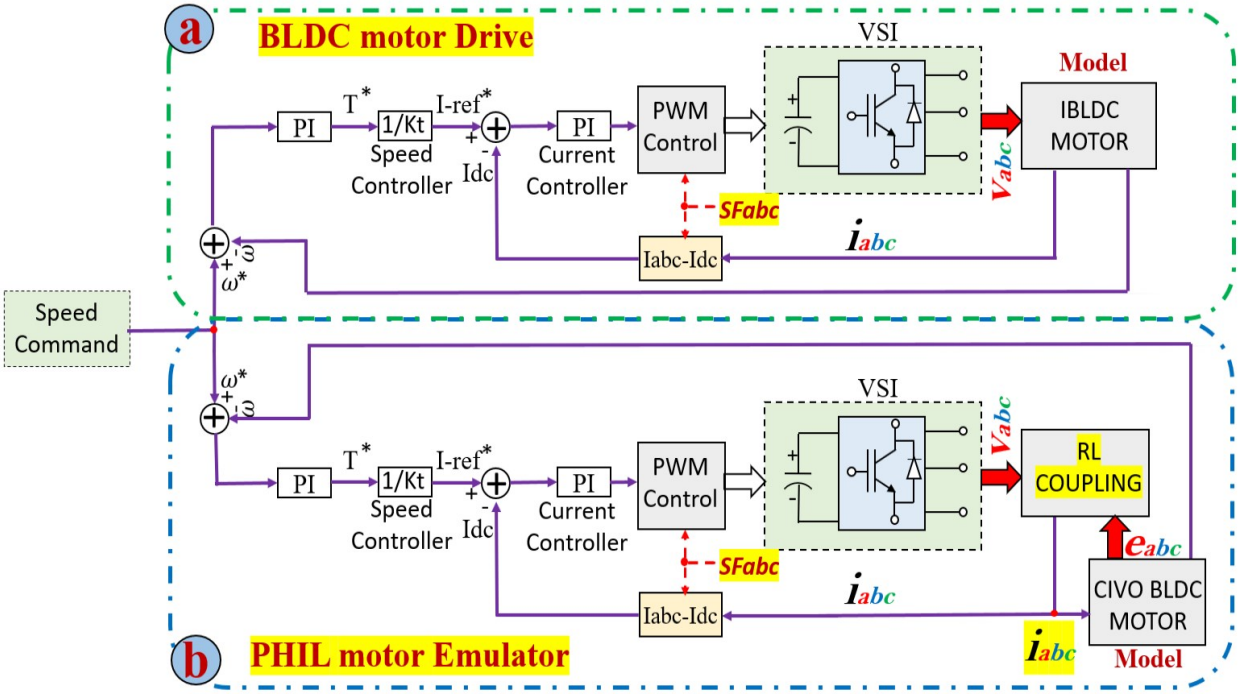


Figure 3-38: Open-loop simulation of emulation of BLDC: (a) Improved BLDC motor model with its drive inverter and control (green box), (b) BLDC current-in voltage-out motor model with its drive inverter and control (blue box)

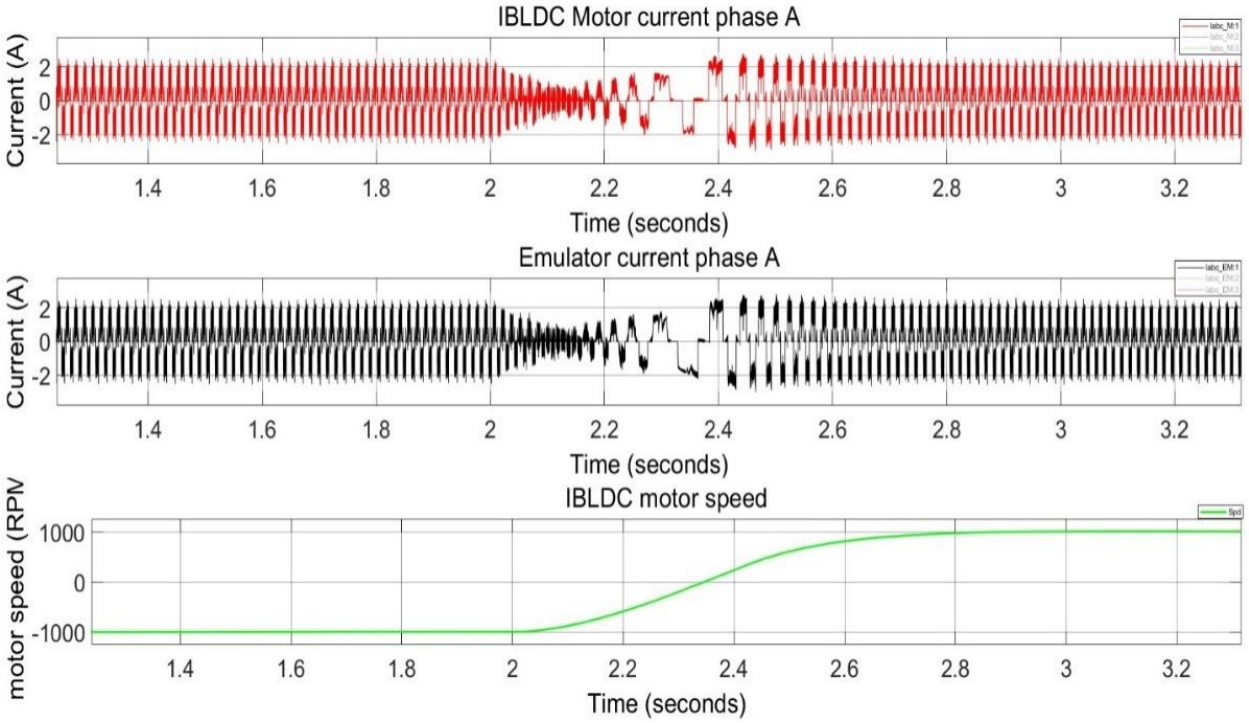


Figure 3-39: Simulation results: Red trace: IBLDC current (2A/div), Black trace: emulator current (2A/div), Green trace: motor speed (1000 rpm/div).

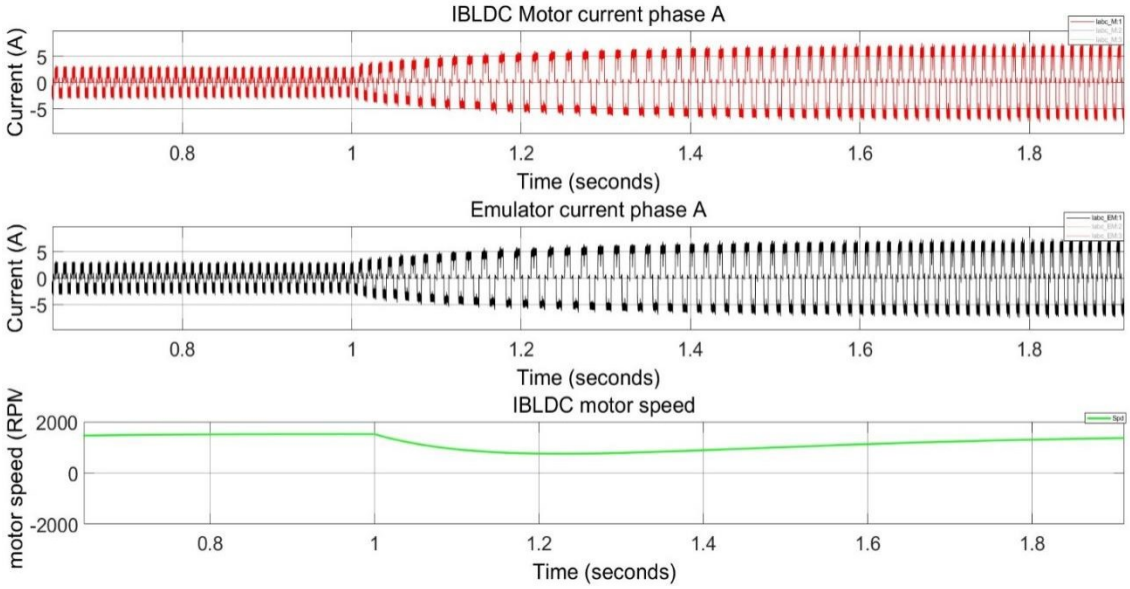


Figure 3-40: Simulation results: Red trace: IBLDC current (5A/div), Black trace: emulator current (5A/div), Green trace: motor speed (2000 rpm/div)

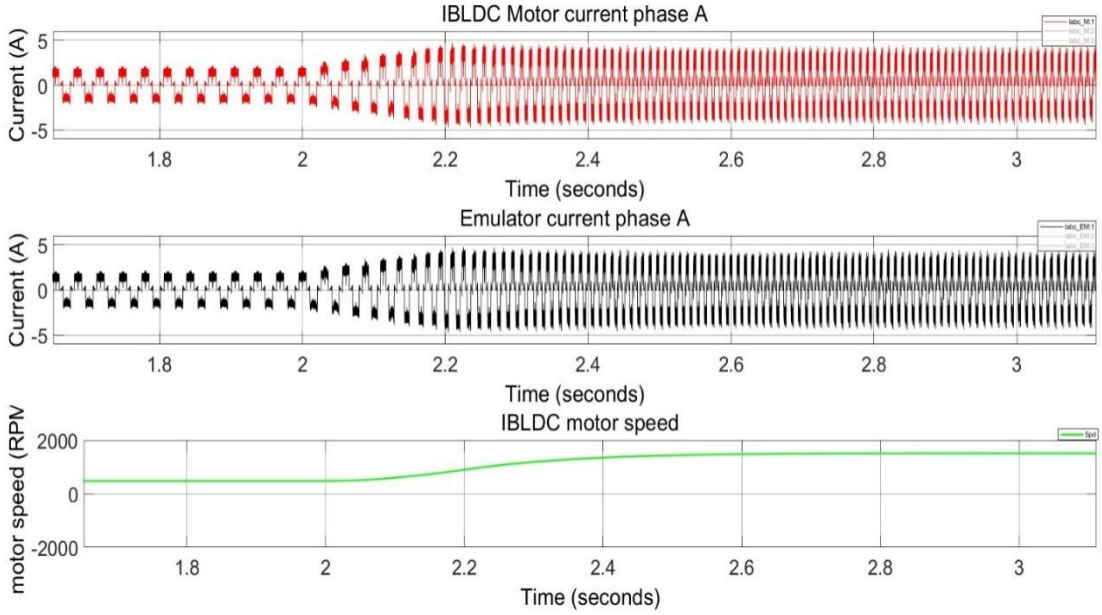


Figure 3-41: Simulation results: Red trace: IBLDC current (5A/div), black trace: emulator current (5A/div), Green trace: motor speed (2000 rpm/div)

Chapter 4: Prototyping and testing of a MOSFET inverter

4.1 Introduction

Power electronic converters can be classified into two categories with respect to the type of their input source on the DC side: voltage-source converters (VSCs) or voltage-source inverters (VSIs) for which the DC bus input is a voltage source (typically a capacitor) and current-source converters (CSCs) or current-source inverter (CSIs) for which the DC bus input is a current source [30]. Power electronic converters allow bidirectional power flow between the AC and DC sides. VSIs are used quite extensively in various applications such as AC motor drives, AC uninterruptible power supplies (UPSs), Static VAR generators (SVG) or compensators (SVC), AC power supplies from batteries, PV arrays, or fuel cells [3]. In chapter 3 of this thesis the BLDC motor drive inverter was introduced and its operation was discussed. In this chapter, prototyping and testing of a MOSFET based three-phase two-level VSI which is used as the BLDC motor drive inverter is explained.

4.1.1 MOSFET and Gate Driver

The drive inverter presented in this thesis work is designed to match the following specifications given in Table 4.1.

Table 4-1: Drive specifications (using Sine PWM modulation)

Specifications	Values
Battery voltage	96 V
Battery Current	81 A
Phase RMS current	112 A
Switching frequency	20 kHz

MOSFET part number HY1920P is selected and used in this work. The MOSFET is rated for 200 V, 90 A. Four of these MOSFETs are paralleled to meet the current carrying capability and to reduce the conduction losses. Ample information on the selection process of MOSFET and explanation on the MOSFET datasheet is given in [31]-[35].

The gate drive circuit of the power MOSFET structure must deliver enough current to rapidly charge and discharge the input capacitance for the device during each switching cycle [36]. In this work, a transformer-coupled gate driver, 2SC0115T SCALE™-2+, is used. Some of the important features are ±15A peak output gate current, +15 V/-6 V gate output voltage, on-board regulated power supply, short circuit protection, under voltage lockout, delay time <100 ns. It combines a complete two-channel driver core with all components required for driving, such as an isolated DC/DC converter, short circuit protection as well as supply voltage monitoring; each of the two output channels is electrically isolated from the primary side and the other secondary channel [37]. Fig.4.1 shows a picture of the 2SC0115T, sample additional circuitry such as hardware dead time circuit, shoot through current protection circuit, gate driver speed enhancement circuit, and short circuit protection circuits are shown in Fig.4-2(a, b, c, d) respectively. Hardware dead-time circuit is set using the time constant of the gate circuit resistor capacitor configuration. Shoot-through fault protection is achieved using a NAND gate at the input of the 2SC0115T gate driver hence preventing both MOSFETs to turn on at the same time. Separate paths for turn-ON and turn-OFF preventing both MOSFETs to turn on at the same time. Separate paths for turn-ON and turn-OFF, different gate resistors are provided to program the turn-ON and turn-OFF times.

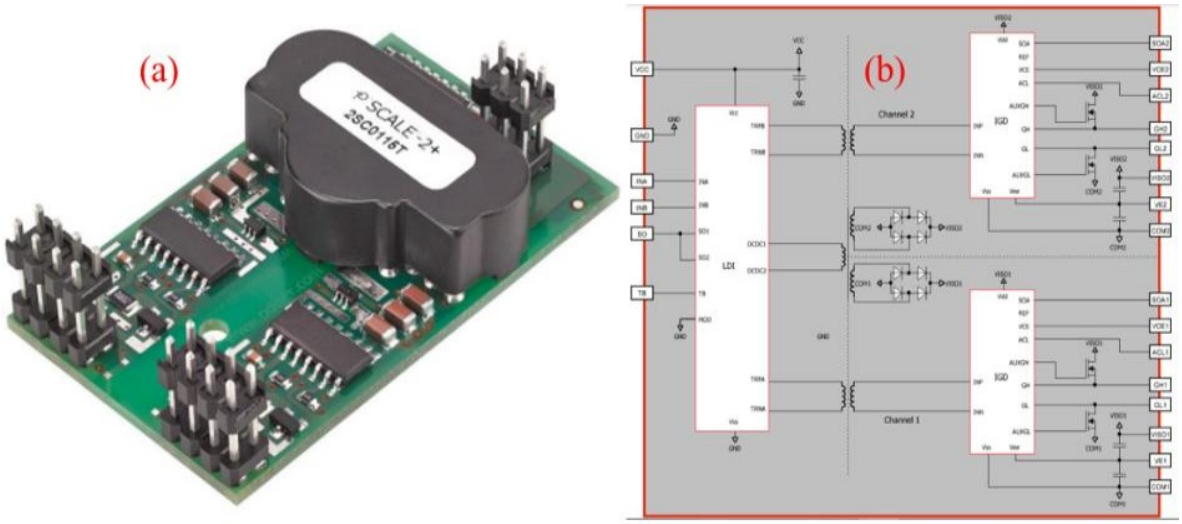


Figure 4-1: (a) 2SC0115T gate driver image [37], (b) 2SC0115T gate driver internal block diagram [37]

MOSFETs are usually used in low voltage low current applications. In low voltage high current applications, MOSFETs are usually connected in parallel to be able to handle the required load current without exceeding each MOSFET's current carrying capacity. High-current MOSFET

devices are usually expensive and hence paralleling low-current MOSFET is sometimes preferred. There are two ways to drive MOSFETs which are connected in parallel [38]:

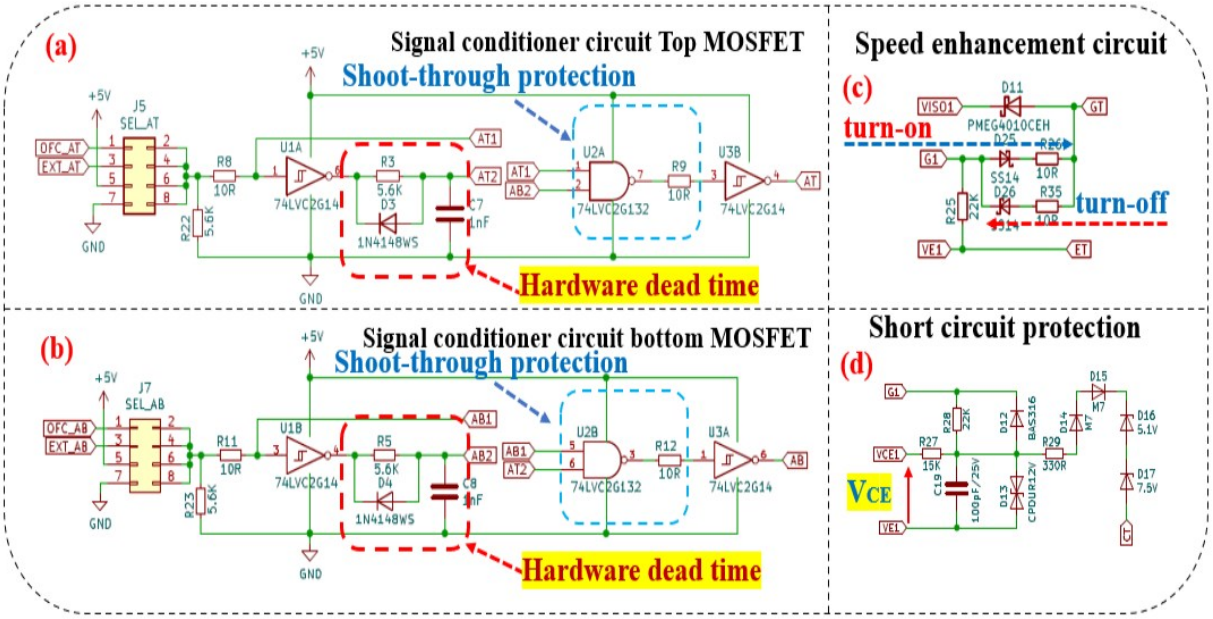


Figure 4-2: (a) Signal conditioner circuit top Mosfet, (b) Signal conditioner circuit bottom MOSFET, (c) Speed enhancement circuit, overcurrent protection

- **Common gate resistor topology:** In this MOSFET driving scheme all the MOSFET gates are connected to the same gate driver through a single gate resistor (R_G). This gate driving scheme is never used in practice as the threshold voltages of MOSFETs connected in parallel never match. This topology results into the switch with the lowest threshold voltage to always conduct first and carry all the current while the other will not conduct at all.

- **Individual gate resistor topology:** This is the preferred parallel driving scheme since each MOSFET has its gate resistor. The gate resistor is chosen to ensure that a small deviation in the threshold voltage of each MOSFET does not affect the switching process of the MOSFETs, thus ensuring that all MOSFETs in the parallel group turn on concurrently. The VSI designed in this thesis uses a similar approach as shown in Fig.4-3 where phase A MOSFETs are connected as a group of 4 top MOSFETs and 4 bottom MOSFETs in parallel each with its gate resistor (R_G).

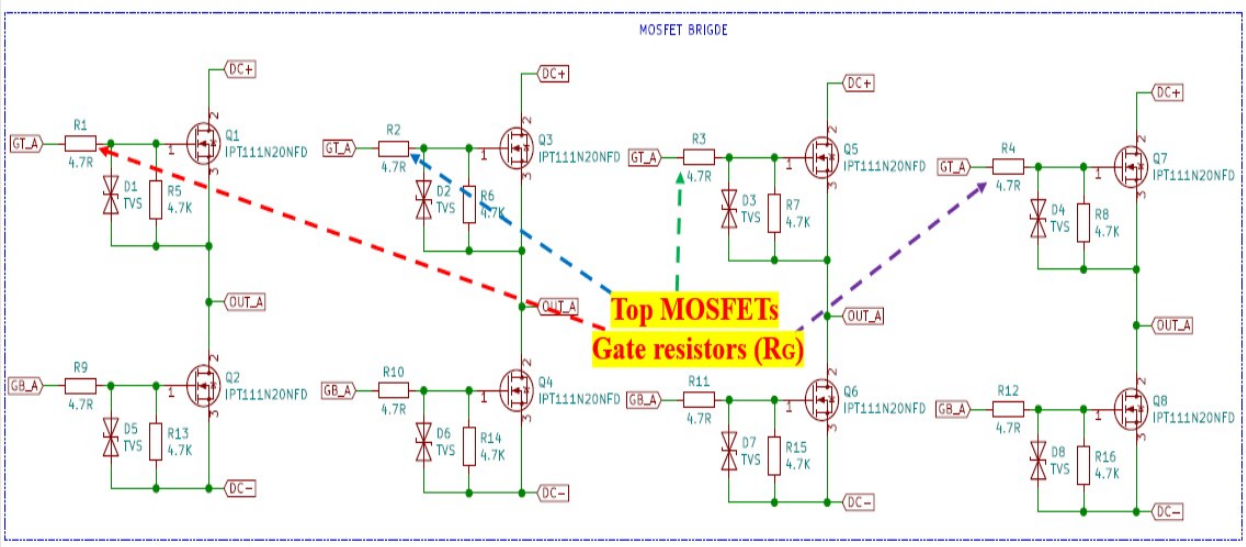


Figure 4-3: VSI phase A top and bottom MOSFETs connected in parallel each with individual gate resistors

4.2 VSI prototyping

The design requirements of the VSI presented in this work are: an operating DC bus voltage of 96 volts, a maximum load peak current of 200 A, and a maximum MOSFET junction temperature of 150°C at an ambient temperature of 27°C. Since the design presented in this thesis is a two-level three-phase voltage source inverter a total of 24 MOSFETs arranged as 8 MOSFETs per leg of the inverter is used to match the load peak current requirements. Additionally, to ensure that the maximum junction temperature of the devices is never exceeded heat transfer with heatsink and forced convection (fans) is used in this work.

4.2.1 MOSFET Mounting

Fig.4-4 (a) shows the heatsink with 24 mounting holes for the 24 MOSFETs devices as described previously. The thermal electric insulator (thermal pad TG-A4500F Fiberglass Mesh Series Thermal Pad) with Dielectric Breakdown Voltage of 6 kV/mm is used to insulate the MOSFET drain terminal from the heatsink as shown in Fig.4-4(b) and Fig.4-4(c). Fig.4-4(d) shows the top view of the inverter PCB. 24 holes are prepared for mounting the MOSFET devices to the heatsink. Proper torque needs to be applied to the MOSFET screw to ensure good thermal conductivity. Fig.4.4(e) shows the bottom view of the inverter PCB, the MOSFET devices are soldered to the PCB then mounted on the heatsink as shown in Fig.4-4(f).

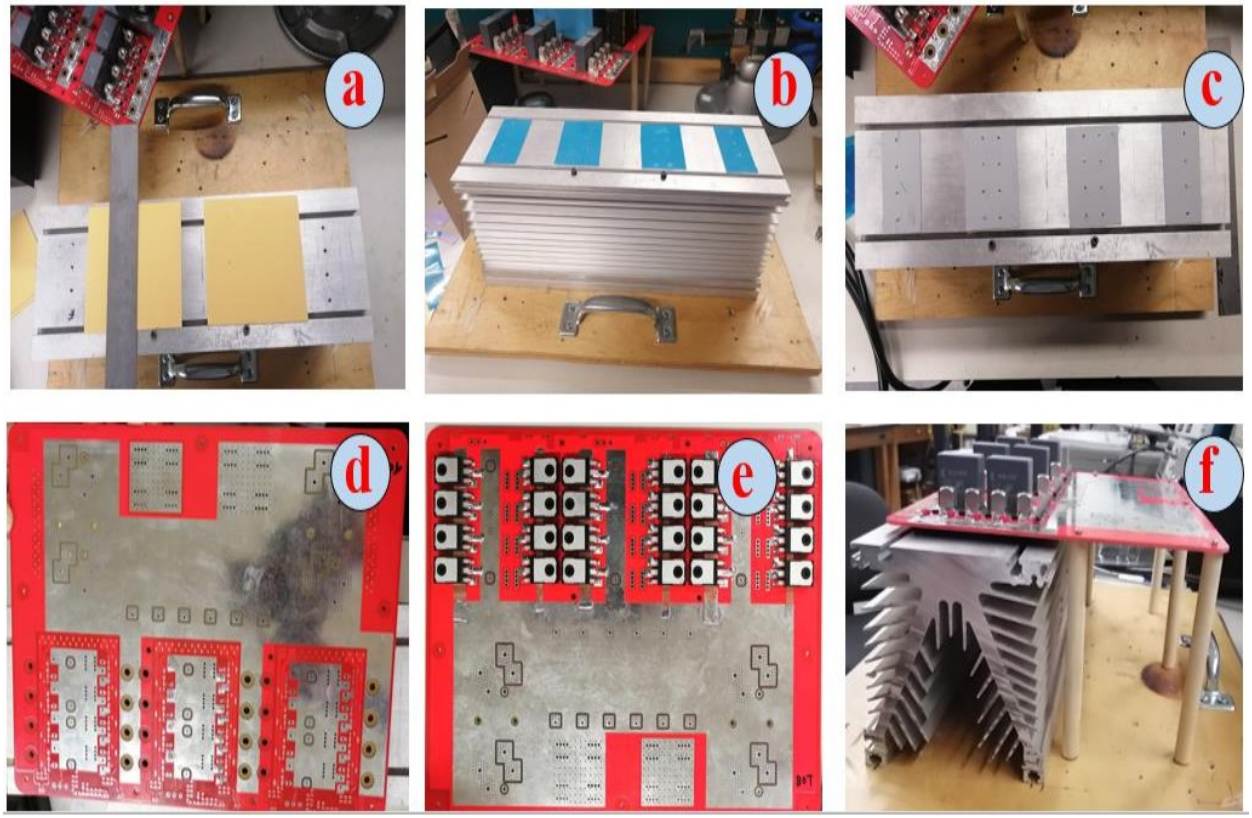


Figure 4-4: (a) Thermal electric insulator (thermal pad) mounted on the heatsink, (b & c) Heatsink preparation before mounting the MOSFETs, (d & e) Inverter PCB top and bottom view with mounting holes and MOSFET devices, (f) Inverter PCB mounted unto the heatsink.

4.2.2 Gate Driver Board Installation

Fig.4-5(a) shows the top view of the MOSFET gate driver board which uses the stand-alone gate driver 2SC0115T and has additional circuitry for protection and control. The fibre optic transmitter (number 2) and fibre optic receiver (number 1) are used to: communicate with the signal conditioner board as shown in Fig.4-5(a), receive gate switching signals (number 2) and send fault signal (number 1). Fig.4.5(b) shows the bottom view of the gate driver board with the 2SC0115T board mounted onto the PCB. Fig.4-5(c) shows the top view of the signal conditioner board, this board receives gate switching signals from the controller and sends them via fibre optic transmitters (number 1) to the corresponding gate drivers. It receives also fault signals from the gate driver boards which are fed back to the controller (the OP4510 controller is not shown here). Fig.4-5(d) shows the bottom view of the signal conditioner board. Fig.4-5(e) shows the three-gate drive boards mounted on the inverter PCB. Each gate driver interfaces between the switches and the controller of the VSI.

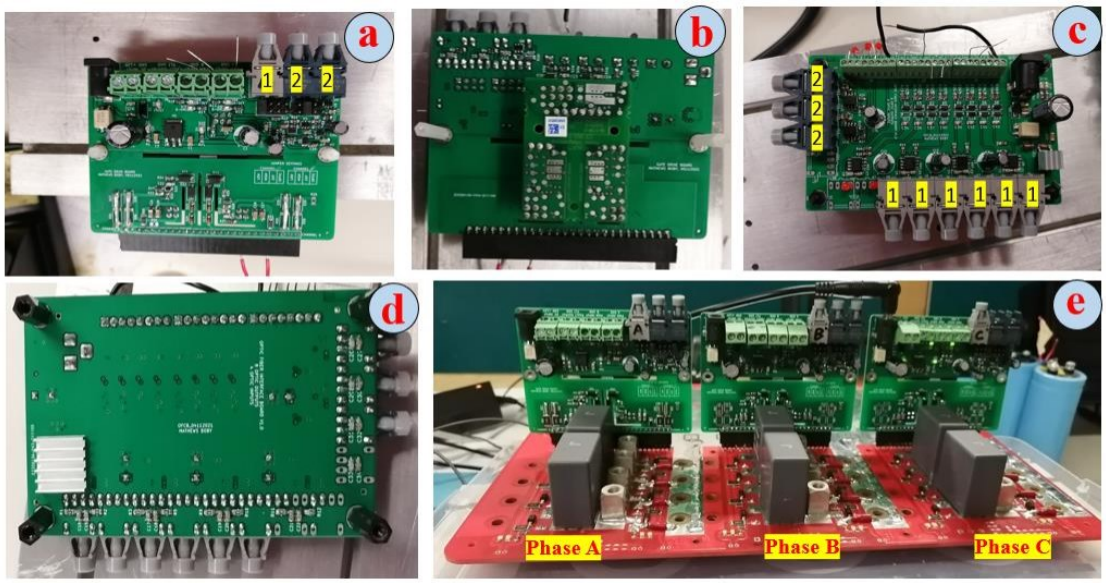


Figure 4-5: (a) Gate driver board top view, (b) Gate driver board bottom view, (c) Signal conditioner board top view, (d) Signal conditioner board bottom view, (e) Gate driver board mounted unto the inverter board

As mentioned previously MOSFET devices connected in parallel, must have separate gate resistor (R_G). Fig.4-6(a) shows individual gate resistors (R_G) of one of the legs of the VSI. Fig.4-6(b) shows the inverter board with snubber capacitors fitted and soldered. Fig.4-6(c) shows the inverter board with the DC link decoupling capacitors fitted and soldered to the board. Fig.4-6(d) shows the inverter board mounted on the heatsink. Additional photos and schematic diagrams of the inverter

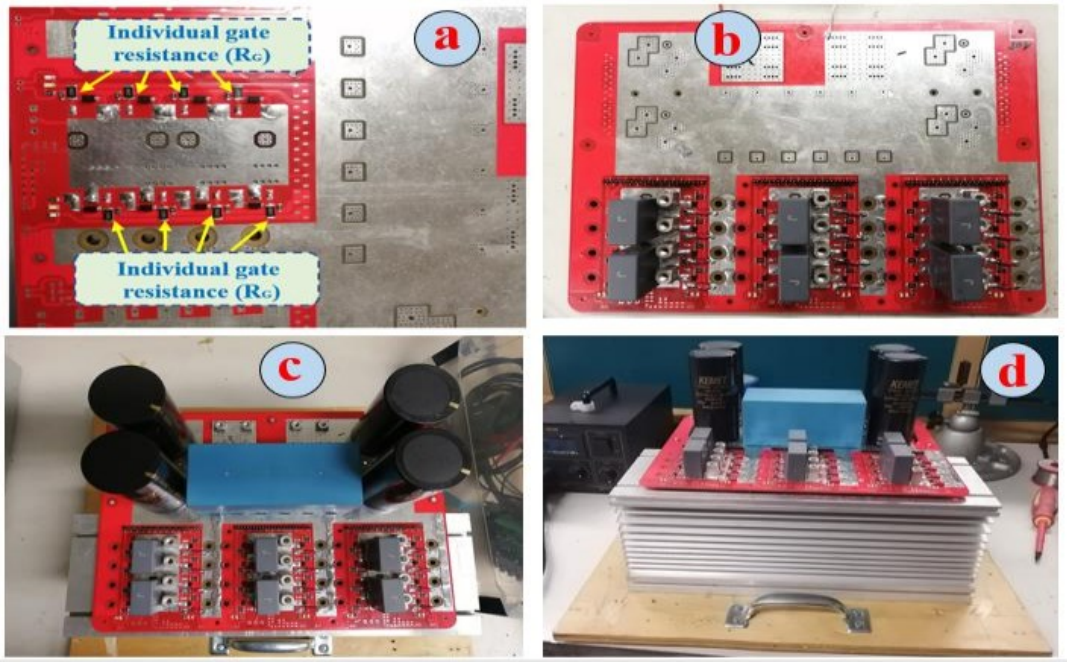


Figure 4-6: (a) One leg of the inverter with MOSFETs connected in parallel with each one having an individual gate resistor (R_G), (b) Inverter board with snubber capacitors, (c) Inverter board with dc link decoupling capacitors, (d) Inverter board with dc link decoupling and snubber capacitors.

are provided in Appendix 2,3, and 4. Sizing of the DC link decoupling capacitors is not discussed in this thesis, however, there is ample information in the literature on that topic.

4.3 Testing

4.3.1 Insulation resistance test

Before soldering or fitting any components to the PCB, the first step is to perform insulation resistance test to the board to ensure that the PCB does not have a short circuit path between the positive and the negative terminals. Additionally point to point test is also performed to ensure that all copper routing and connections are tested. These tests save time and material in case the PCB is defective.

4.3.2 Double pulse test (DPT)

In this thesis work, no provision was made in the inverter board (PCB) to have sense resistors as depicted in Fig.4-7(a) where only one leg of the VSI is shown, and to measure the current of the device under test as depicted in Fig.4-7(b). Consequently, only the inductor current is measured and recorded during the DPT performed in this work. Therefore, the shape of the current recorded by the oscilloscope during the DPT performed on the VSI presented in this chapter is different from the shape of the current of the device under test depicted in Fig.4-7(a). Due to this, the analyses related to switching losses cannot be accurately performed in this case. Hence, the DPT performed in this work is used only to evaluate the switching performance of the MOSFET devices. Fig.4-8(a) shows the DPT setup circuit topology for the top MOSFETs of one of the legs of the inverter while Fig.4-8(b) shows the DPT setup circuit topology for the bottom MOSFETs of one of the legs of the inverter. As shown in Fig.4-8 the gate signal (V_G), the voltage between

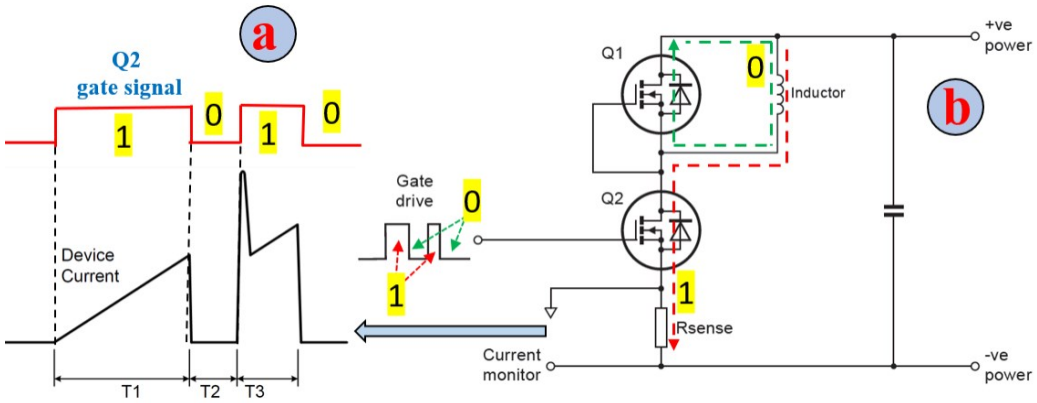


Figure 4-7: (a) Bottom MOSFET Q_2 current, (b)Double pulse test of a half-bridge MOSFET

the drain and the source (V_{DS}) of the device under test, and the inductor current (i_p) waveforms are measured and recorded in the oscilloscope. Fig.4-9 (a and b) shows the hardware setup of the DPT used in this work. Two inductors rated at 100 A each are connected in parallel to increase the current carrying capacity of the inductor to reach the 200 A peak current as per design requirements.

In Fig.4-10(a) the DPT is performed for the top MOSFETs of phase A of the inverter. Output switching pole voltage oscillations, and ringing are observed as shown in Fig.4-10(b), a gate resistance of 5 Ω is used in this case. Fig.4-10(c) shows the DPT performed on the top MOSFETs and Fig.4-10(d) is the DPT performed on the bottom MOSFETs of phase A of the VSI. It can be observed that the oscillations and ringing in the output pole voltage are reduced considerably when the gate resistance is increased to 100 Ω. Fig.4-11(a) shows the oscillations and ringing of the gate voltage when using a gate resistance of 5 Ω, these oscillations are damped when the gate resistance is increased to 100 Ω as shown in Fig.4-11(b).

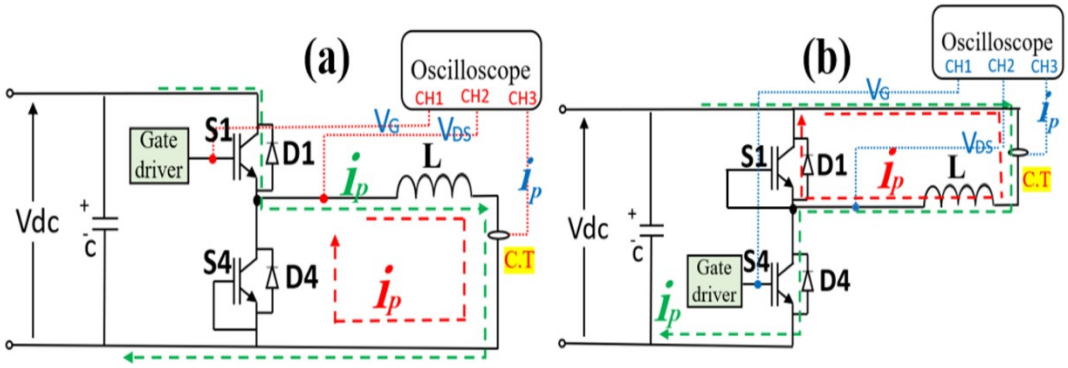


Figure 4-8: (a) Double pulse test for the top MOSFET, (b) Double pulse test for the bottom MOSFET

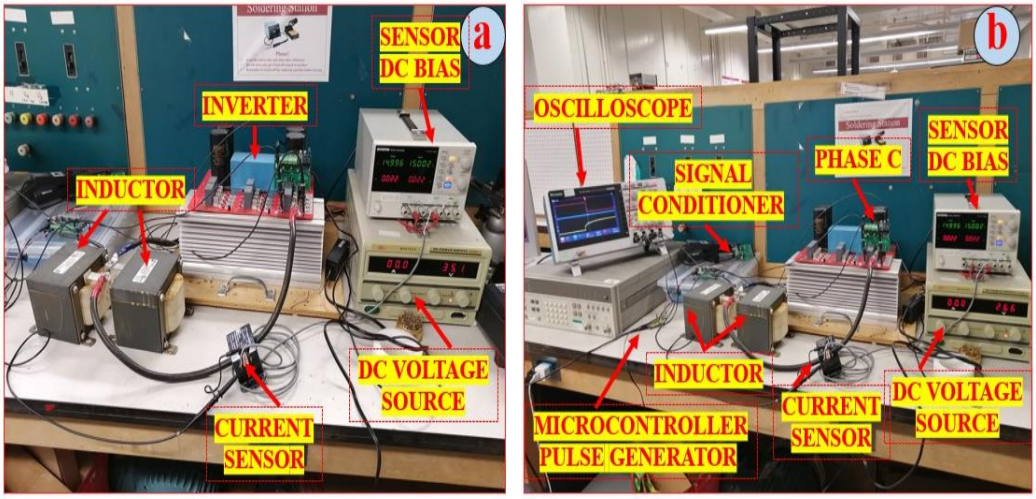


Figure 4-9: Double pulse test setup

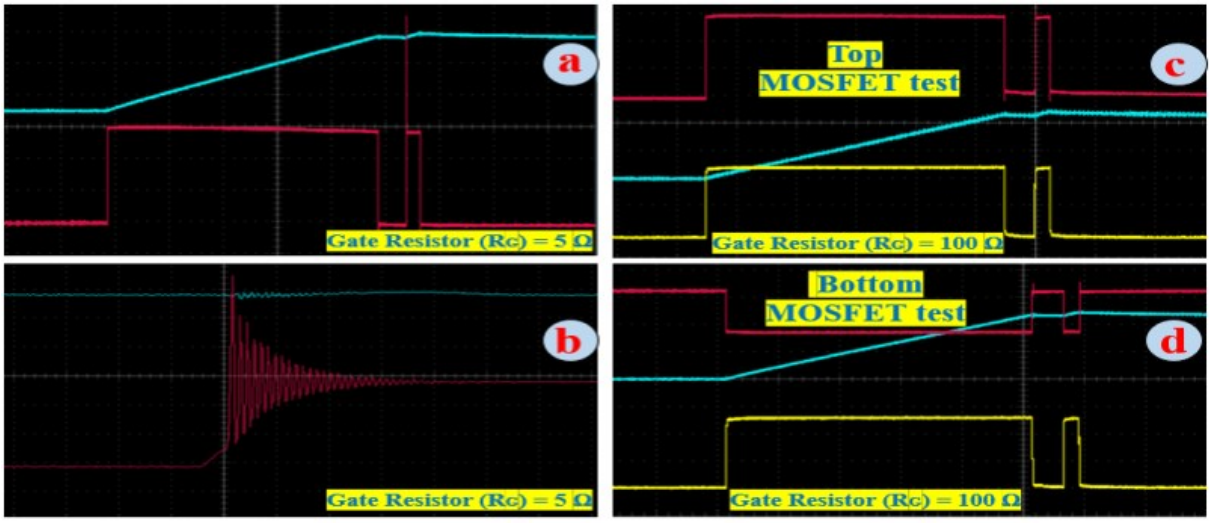


Figure 4-10: Double Pulse Test: (a and b) Red trace: output voltage (20 V/div), Blue trace: inductor current (40 A/div), (c) Red trace: output voltage (20 V/div), Blue trace: inductor current (40 A/div), Yellow trace: MOSFET gate voltage (10 V/div), (d) Red trace: output voltage (40 V/div), Blue trace: inductor current (40 A/div), Yellow trace: MOSFET gate voltage (10 V/div).

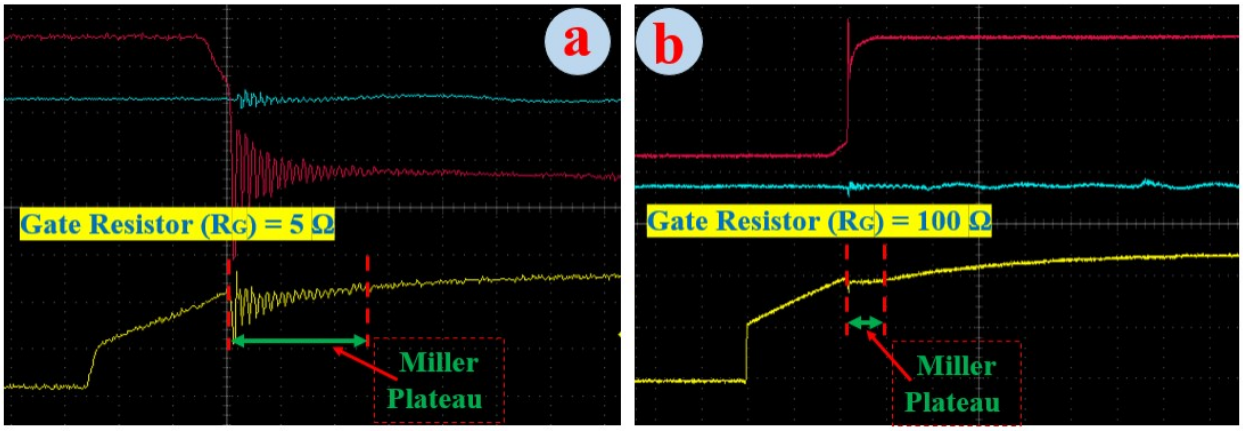


Figure 4-11: Double Pulse Test: (a and b) Red trace: output voltage (20 V/div), Blue trace: inductor current (40 A/div), Yellow trace: MOSFET gate voltage (10 V/div).

4.3.3 Passive load test of the VSI

Before connecting the VSI to an active load such as a motor, the inverter is connected to a passive load (inductors) and tested up to the rated current. Fig.4-12(a) shows the experimental setup block diagram used during this test. Sinusoidal PWM (SPWM) with a switching frequency of 10 kHz and a fundamental frequency of 60 Hz is implemented inside the real-time simulator (OP4510) from OPAL-RT, the time-step used is 20 us. The gate switching signals from the SPWM block implemented in the real-time simulator OP4510 are sent to the gate drivers of the VSI to control it. Sinusoidal voltage reference waveforms (V_{ref}) which are used in the SPWM block in the OP4510 are measured in the oscilloscope as shown in Fig.4-12(a). The line currents output by the

VSI are measured in the oscilloscope using the current transducer (CT_1). Fig.4-12(b) shows the experimental test results. Due to hardware limitations, the VSI was only loaded up to 100 A.

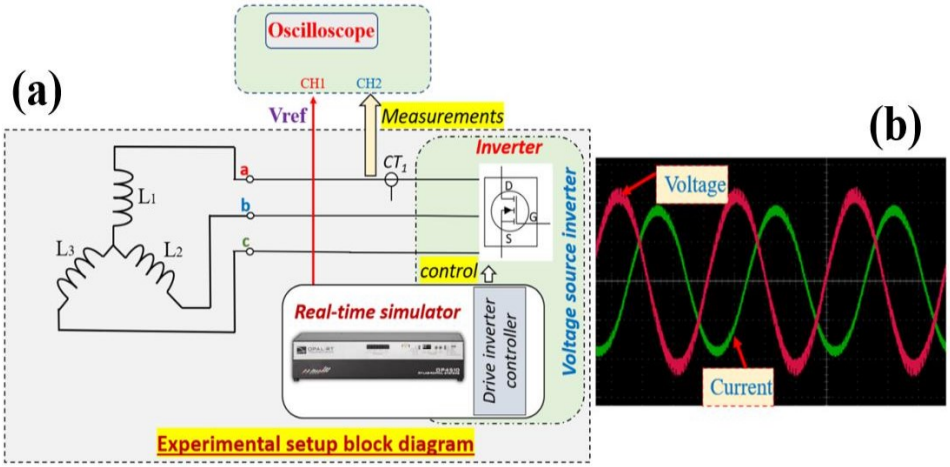


Figure 4-12: Inverter test with passive load : (a) experimental setup block diagram, (b) Red trace: voltage (0.25 V/div), Green trace: current (50 A/div).

4.3.4 Inverter thermal test and monitoring

Due to hardware limitations, the DPT performed in this thesis cannot be used to evaluate the power loss of VSI. However, software simulation is performed to evaluate the power loss in the VSI when loaded at its rated current under SPWM control with a passive inductive load. The simulation is performed in Powersim (PSIM) simulation software using the MOSFET thermal model. Tables 4-2 to 4-4 are the average power losses of each inverter leg. Phase A, B, and C average power losses and junction temperatures for each group of 4 parallel connected MOSFETs are given in their respective tables where P_{cond_D} is the MOSFET body diode conduction loss. P_{cond_Q} is the MOSFET on conduction loss. P_{sw_D} is the MOSFET body diode switching loss. P_{sw_Q} is the MOSFET switching loss. P_{loss_total} is the overall power loss of the 4-parallel connected MOSFETs and finally T_j is the junction temperature of the group of MOSFETs. Sinusoidal PWM with a fundamental frequency of 500 Hz (fundamental frequency of the motor at maximum speed) and switching frequency of 10 kHz is used in this simulation. A passive (RL) load is connected at the output of the inverter as shown in Fig.4-13, its value is chosen to ensure that maximum peak current of 200 A flows in the circuit. Table 4-5 is the simulation results of the average of all the power losses, junction temperatures, and case temperatures of all MOSFET devices. The thermal resistance of the thermal pad as given in the datasheet of the thermal pad TG-A4500F is 0.18 °C/W. There is no available datasheet for the heatsink used in this thesis, a thermal value of 0.2 °C/W which is an average value for heatsink using forced convection (fans) as given in [40]. Thermal monitoring

and protection of the VSI presented in this work is achieved by sensing the heatsink temperature with a thermocouple sensor. Using a temperature monitoring microcontroller an LED and a buzzer is controlled by an interposing relay as shown in Fig.4-14(b). Fig.4-14(a) shows the cooling fans and the thermocouple mounted to the heatsink.

Additionally, thermal temperature check of the VSI on-load is performed using the FLIR C2 thermal camera. The MOSFET device, the busbar and PCB copper layers temperatures are measured using the thermal camera while the VSI is loaded using a passive load at a current of 100 A. This test is very useful to ensure that there are proper thermal contacts between the MOSFETs and the heatsink, and that all the MOSFETs and busbar screws are well torqued and there is no component overheating during the test. This loading test is performed over a duration of 6 hours, the temperature measurements are given in appendix 5.

Table 4-2: MOSFETs (S3) and (S6)

S1		S4	
Type	Average	Type	Average
S1_Pcond_D	4.14 W	S4_Pcond_D	4.32 W
S1_Pcond_Q	39.9 W	S4_Pcond_Q	34.8 W
S1_Ploss_total	55.9 W	S4_Ploss_total	50.1 W
S1_Psw_D	0.041 W	S4_Psw_D	0.044 W
S1_Psw_Q	11.8 W	S4_Psw_Q	11 W
S1_Tj	155 °C	S4_Tj	155 W

Table 4-3: MOSFETs (S1) and (S4)

S3		S6	
Type	Average	Type	Average
S3_Pcond_D	4.3 W	S6_Pcond_D	4.05 W
S3_Pcond_Q	38.4 W	S6_Pcond_Q	37 W
S3_Ploss_total	54 W	S6_Ploss_total	52.5 W
S3_Psw_D	0.043 W	S6_Psw_D	0.042 W
S3_Psw_Q	11.3 W	S6_Psw_Q	11.4 W
S3_Tj	155 °C	S6_Tj	155 °C

Table 4-4: MOSFETs (S5) and (S2) phase C

S5		S2	
Type	Average	Type	Average
S5_Pcond_D	4.02 W	S2_Pcond_D	4.17 W
S5_Pcond_Q	34.5 W	S2_Pcond_Q	38.4 W
S5_Ploss_total	49.6 W	S2_Ploss_total	76.4 W
S5_Psw_D	0.044 W	S2_Psw_D	0.041 W
S5_Psw_Q	11.1 W	S2_Psw_Q	31.2 W
S5_Tj	155 °C	S2_Tj	155 °C

Table 4-5: Heatsink temperature and power dissipated

Heatsink	
Type	Average
average junction temp	155 °C
Total power loss	339 W
Case temperature	94.3 °C

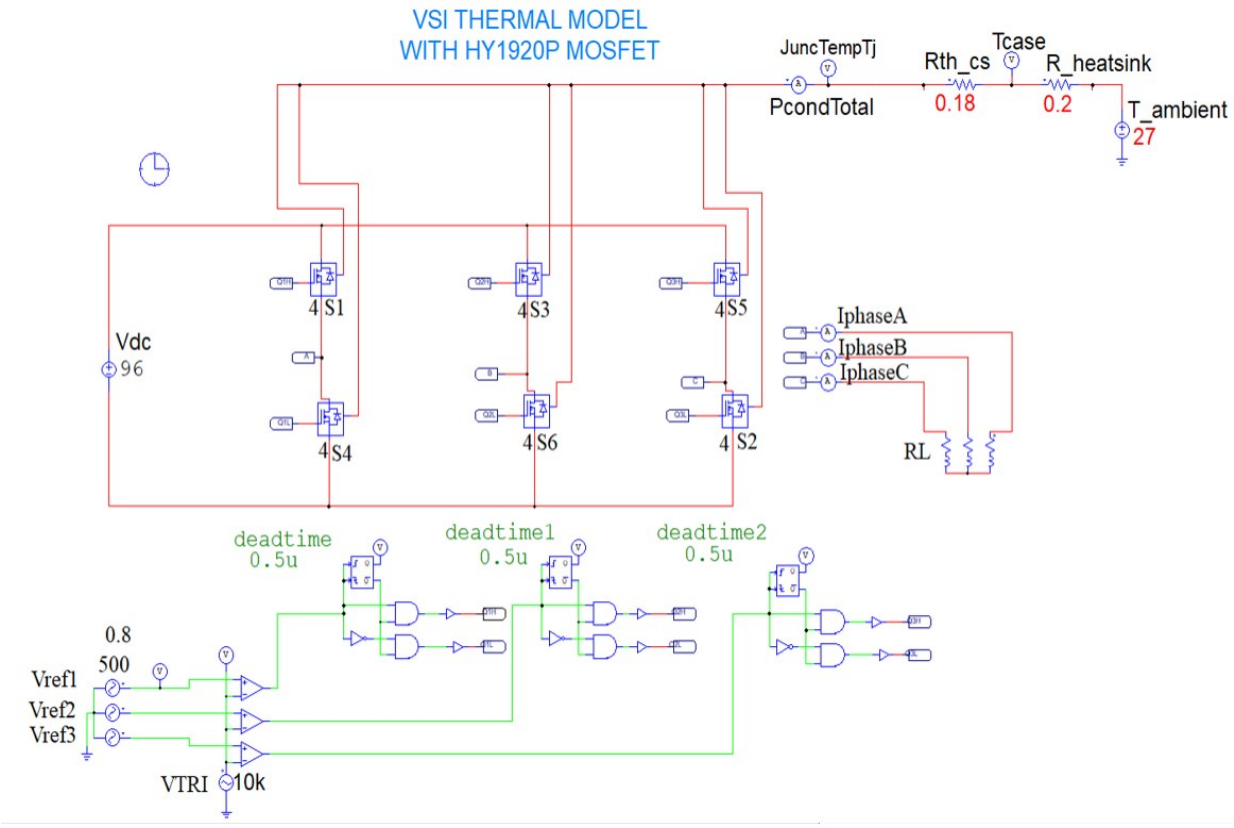


Figure 4-13: PSIM thermal mode simulation of the VSI with HY1920P MOSFETs

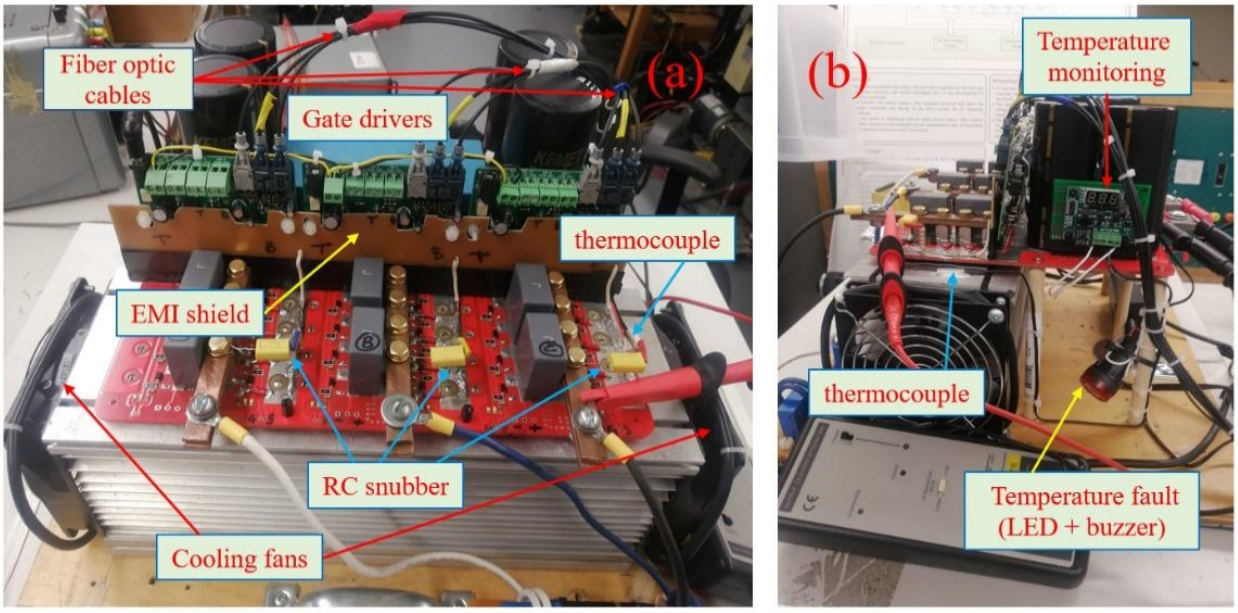


Figure 4-14: (a)VSI fully assembled, (b)heatsink temperature monitoring

4.3.5 Short circuit protection

Each channel of the 2SC0115T driver is equipped with a V_{CE} monitoring circuit, the turn-off threshold is internally set to V_{thx} = 9.3 V. If the voltage at the voltage monitoring input (VCEx) exceeds 9.30 Volts then the module outputs a short-circuit fault status and blocks the output gate switching signals to the MOSFET. In the VSI presented here, the voltage across $R_{ds(on)}$ is monitored if it exceeds 4.5 V then a short circuit current fault condition is detected. As recommended in the datasheet of the 2SC0115T gate driver, additional circuitry made up of zener diodes is used to set the short circuit current level as shown in Fig.4-2(d). The gate driver fault status outputs are wired to the real-time simulator (OP4510) for hardware fault reporting.

4.3.6 Gate drive board protection against EMI

Due to the layout of the PCB and the proximity between the gate driver and the output busbar, EMI due to the output switching voltages can cause disturbances to the operation of the VSI gate drive boards. Therefore, an EMI shield board made up of single sided copper PCB is used to protect the gate driver boards from the EMI by grounding the copper layer to the DC ground, thus forming an EMI shield.

Chapter 5: Implementation and testing of PHIL emulator

5.1 Introduction

In this chapter, the hardware implementation of the open-loop emulation of the BLDC using the PHIL testing method is developed, implemented and validated. Firstly, the drive inverter is connected to the physical motor and subjected to various speed and loading conditions, the motor current is recorded during the entire test using the oscilloscope. Then, the emulator setup is connected to the same drive inverter, the emulator is subjected to the same test conditions as the physical motor for various speed and loading conditions, the emulator current is recorded during the entire test using the oscilloscope. The two currents are then compared to validate the emulation implementation.

5.2 System simulation techniques and applications

A system is defined as a set of elements that are interacting and interrelated and working together as part of a mechanism. The interrelationships between the input and output of the system should be analysed in order to generate a model of the system and to quantitatively examine its behaviour. System models may be divided into physical models and mathematical models [39]. The application of mathematical models of systems has been gaining momentum in the industry in past years, and it is still expanding as computer technology evolves. Systems are classified as continuous-time systems, discrete-time systems, discrete event systems and hybrid systems, they are also classified as subclasses of linear, nonlinear, time-invariant, time-varying, lumped parameters, distributed parameters, deterministic and stochastic systems [39].

Systems simulation is a subject that studies and analyse the response or behaviour of systems [40]. Ideally, the mathematical model should model all the characteristics of the system to ensure high fidelity of simulation. However, in practice this is usually not achievable because of modelling limitations that arise as a result of the complexity nature of systems. Therefore, the models employed in most cases captures the most important features of the system while neglecting some others.

A distinction between two sorts of models for both real-time and non-real-time (offline) simulations soon emerged in the early days of simulation [40]. A continuous type model is one in which the system state variables are continuously changing as determined by the model differential equations and the inputs to the system. Discrete type model on the other hand is one in which the

state variables of the system are assumed to be changing at specific discrete instants of time and remains relatively constant in between. Discrete-time steps are solving techniques in which the time to solve mathematical model or equations is fixed (fixed-time steps), variable time-steps as the name implies are solving techniques in which the time to solve mathematical models or equations are variable. Variable time-steps solving techniques are used for solving high-frequency dynamics and non-linear systems but are unsuitable for real-time simulation [41]. Real-time simulation of systems uses discrete time-steps. A simulation is said to be running in real time when the system variables output by the real-time simulator are produced within the same length of time it would take the physical system to output the its variables while subjected to the same input conditions. In order words, the response time of the real-time simulator to an input to the system model implemented inside it should be ideally equal to the response time of the physical system when subjected to the same input. To achieve this, real-time simulators are normally built to run with very high computational speed. A Real-time (RT) simulator belongs to a class of computer systems that receive inputs from the external world, process them, and generate outputs to influence the external world, within a finite time [4].

In this thesis, the system to be emulated is the BLDC motor, real-time simulation using a RT simulator (OP4510) is used in the implementation of the emulator. A fixed-time discrete solver is used in the offline simulation first to simulate the BLDC drive offline. This explains the need to have a discrete BLDC motor model (improved BLDC motor model), and secondly to run the simulation of the emulation offline before developing the emulator setup.

5.3 Power Hardware-In-the-Loop (PHIL)

In electric drive applications, traditionally there are two possibilities for performing system tests with medium to high power ratings [42]:

- Performing and experimenting on real hardware.
- Performing a pure software simulation.
- A third possibility is now being developed which is called hardware in the loop testing (HIL), this is an approach of combining simulation with hardware experimentation.

The alternative to hardware testing has traditionally always been software simulation [42], software simulation has some advantages over hardware experiments such as lower cost, the same

software can be used to analyse and test various machines, various types of faults can be simulated on a machine with no damage to the equipment, and it is safer, faster, easier to perform. However, system modelling has limitations, therefore software simulation alone cannot be used to validate the operation and implementation of a physical system as experimental results must be used to validate the simulation results obtained.

For the best accuracy of results, performing tests on real hardware is the best approach in most applications. In cases where the whole system is not accessible, some parts of the system can be simulated in a RT simulator while the rest of the system can be tested at the rated power. The approach of combining simulation with hardware experimentation is known as hardware-in-the-loop (HIL), and the addition of power components distinguishes power HIL from control HIL [42]. In hardware-in-the-loop simulation systems, part of the simulation loop is composed of computer software, while the rest is the actual hardware systems [39]. Fig.5-1 depicts a clear difference between PHIL and CHIL.

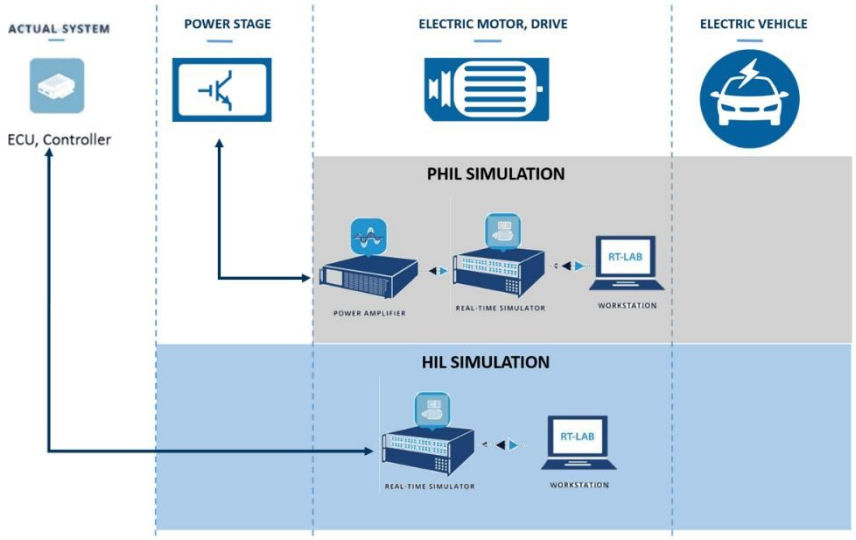


Figure 5-1: PHIL implementation block diagram [43]

Power Hardware in the loop plays an important role in developing, testing and validating equipment as it combines the benefits of pure simulation and laboratory testing, it is used for testing wind energy systems, microgrids, electric vehicles, photovoltaic inverters, electric drive applications etc. [42]. In electric drive applications the electric machine model is implemented in the RT simulator, the RT simulator controls the power converter to behave as a virtual machine to the drive inverter (Hardware Under Test), thus allowing the drive inverter controller and hardware to be tested at different loading and running conditions.

Chapter 3 of this thesis discusses the BLDC drive setup and its implementation, the same control logic and controller are implemented in this section for the BLDC drive inverter control, hence it will not be discussed in this chapter. The only difference between the BLDC drive inverter presented in chapter 3 and the emulation setup discussed in this section is that the BLDC motor is now replaced by a virtual BLDC machine which uses a power electronic converter to mimic the behaviour of the physical BLDC motor. Fig.5-2 below shows the block diagram of the motor emulator setup. Open-loop emulation control (voltage control mode) is implemented in this chapter, hence the values of the coupling element (inductor L and resistor R) as shown in Fig.5-2 are chosen to match the winding inductance and resistance of the physical BLDC motor. The inverter designed and prototyped in chapter 4 of this thesis is used as a drive inverter.

5.3.1 BLDC virtual machine

Fig.5-3 is the BLDC motor model which is implemented inside the RT simulator. Open-loop emulation control requires a current-in voltage-out model, the currents measured by the current transducers as shown in Fig.5-2 are then used in the current-in voltage-out model to generate the reference voltages which are amplified by the linear amplifier, thus controlling the amount of current which flows in each line. The current-in voltage-out BLDC motor model is discussed and validated in chapter 3; therefore, no further explanation is given about the model.

5.3.2 Hardware setup components

In the open-loop emulation scheme used in this work, the drive phase currents are sensed and provided as feedback to the emulator controller. The emulator controller is implemented in an Opal-RT real-time simulator hardware OP4510. The time-step used in the real-time simulator is 20 us. The BLDC drive control is also implemented in the same real-time hardware for reducing hardware complexity. The drive switching frequency is 10 kHz and the drive control works at a time-step of 100 us. The drive phase currents are sensed using current transducers (CTs) and are used in the current-in voltage-out BLDC machine model to generate the back-emf voltages. These back-emf voltages are amplified using a linear amplifier LVC5050 (*AE Techron*). The outputs of the linear amplifier are connected to the drive inverter through link inductors and resistors which are chosen such that they are equal to the machine inductance and resistance. This circuit arrangement physically replicates a voltage-behind-reactance model of a BLDC machine. Fig.5-3 shows the hardware setup used in this thesis.

In Fig.5-3, the number on the picture of the hardware setup represents:

1. The workstation which is the interface between Simulink, RT-Lab and the RT simulator.
2. Linear amplifier LVC 5050 from *AE Techron*.
3. Fuses which are used as an additional short circuit protection.
4. The RL link filter between the linear amplifier and the drive inverter.
5. The drive inverter.
6. The BLDC motor testbench.
7. The current transducers (CT_1 and CT_2).
8. The DC voltage source.
9. The RT simulator OP4510 from opal-RT.
10. The oscilloscope.

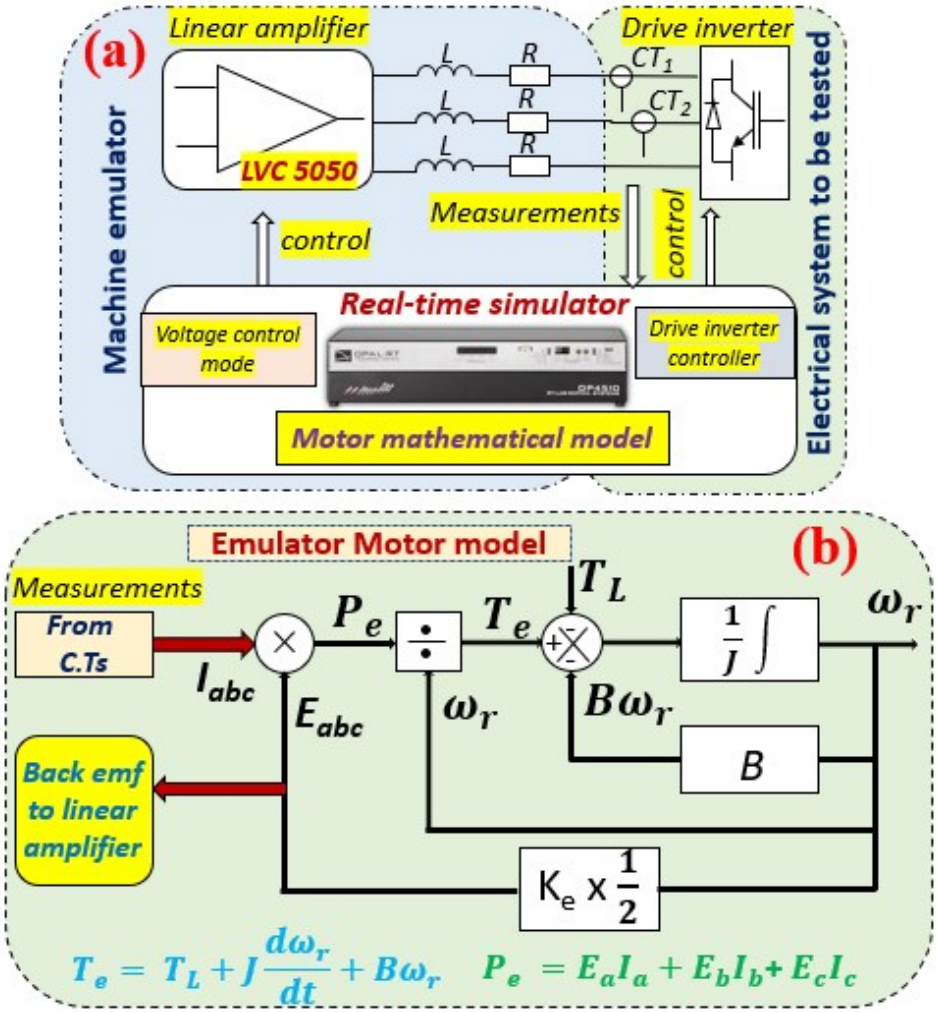


Figure 5-2: (a) Emulator setup of the BLDC motor, (b) Current-In Voltage-Out BLDC motor model

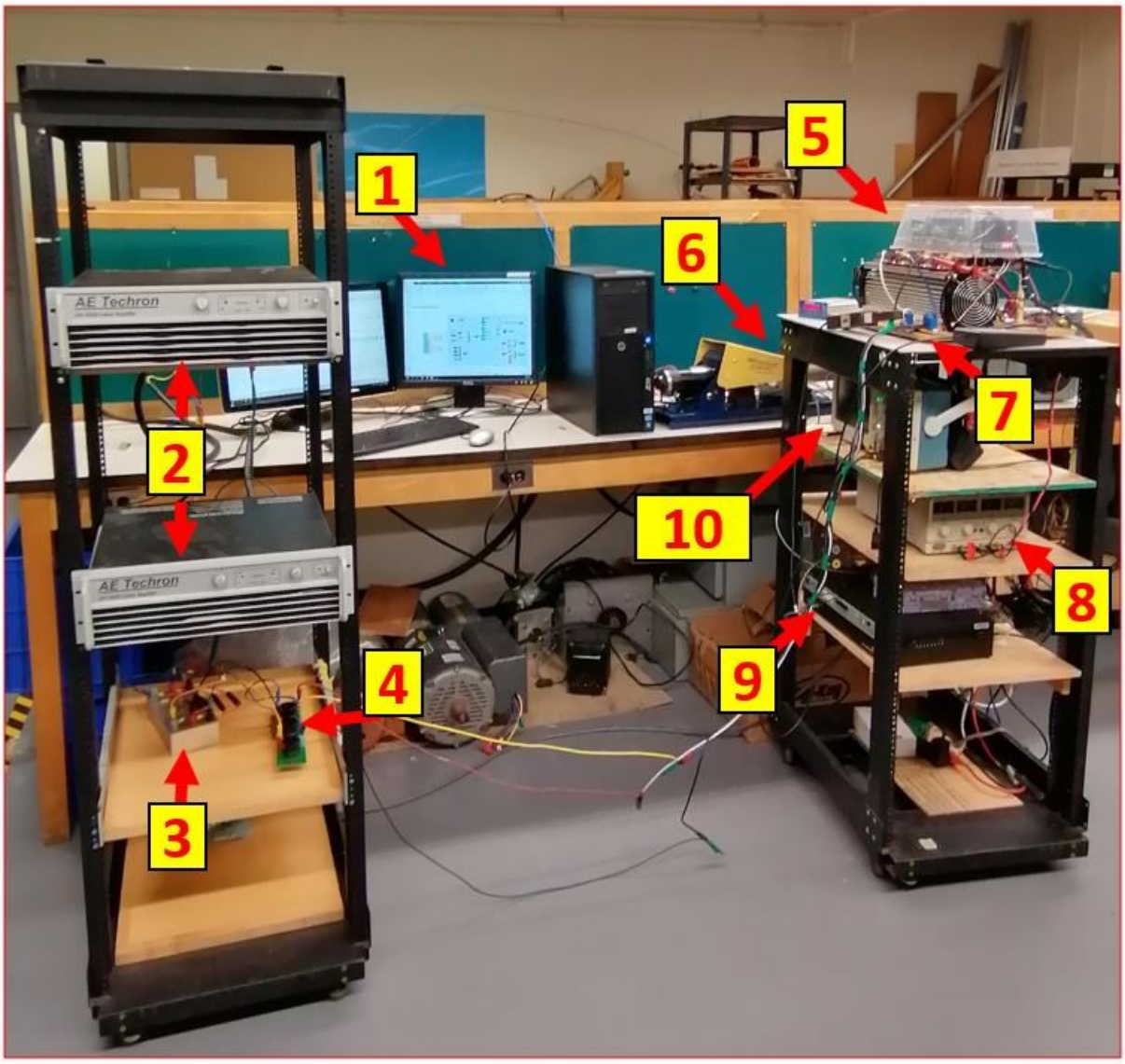


Figure 5-3: BLDC emulation hardware setup

- **RL link filter:** The inductor part number 1140-151K from Bourns is used as the RL filter between the drive inverter and the emulating converter. The inductor DCR is 0.04 Ω, however additional resistance from the wires and the fuses add up to 0.15 Ω. The specifications of the inductor are given in table 5-1.

Table 5-1: RL link filter specifications

RL link	
Part- number	1140-151K-RC
L (μH) ±20% at 1 kHz	150
DCR	0.04 Ω
I(rms)	8.3 A
I(sat)	16.9 A

- **Linear amplifier LVC 5050**: The *AE TECHRON* Model LVC 5050 is a dual channel power supply amplifier designed for use in the most demanding high-power systems, it can deliver up to 2,500 watts per channel into 2-ohm loads**.** It can source and sink power, thus eliminating the need to have an AFE in this application since the power rating of the motor is low. The specifications of the amplifier are given in table 5-2.

Table 5-2: Linear amplifier LVC 5050 specifications

LVC 5050	
Gain (used)	30
Frequency	± 0.1 dB from 20 Hz to 20 kHz at 1 watt
Phase response	±10° from 10 Hz to 20 kHz at 1 watt.
Slew rate	>30 V per microsecond
THD	<0.05% from 20 Hz to 1 kHz

5.4 Emulation test results

Using the emulator setup depicted in Fig.5-2(a) the emulator current is recorded using the oscilloscope. The BLDC motor drive is first connected to the physical motor and tested for various speed and loading conditions. Then the emulator is subjected to the same testing conditions. The following tests are performed:

5.4.1 Steady state operation:

a) **Motor speed 1000 rpm:** The currents in phase A of the emulator (violet trace) and the physical BLDC motor (black trace) during steady-state operation are shown in the figures below. In Fig.5-4 the motor and the emulator are running on no-load. In Fig.5-5 a load

torque of 0.8 p.u is applied to the motor and the emulator. In Fig.5-6 a load torque of 0.5 p.u is applied to the motor and the emulator. In Fig.5-7 speed reversal from -1000 rpm to 1000 rpm is applied to both the emulator and the motor on no-load.

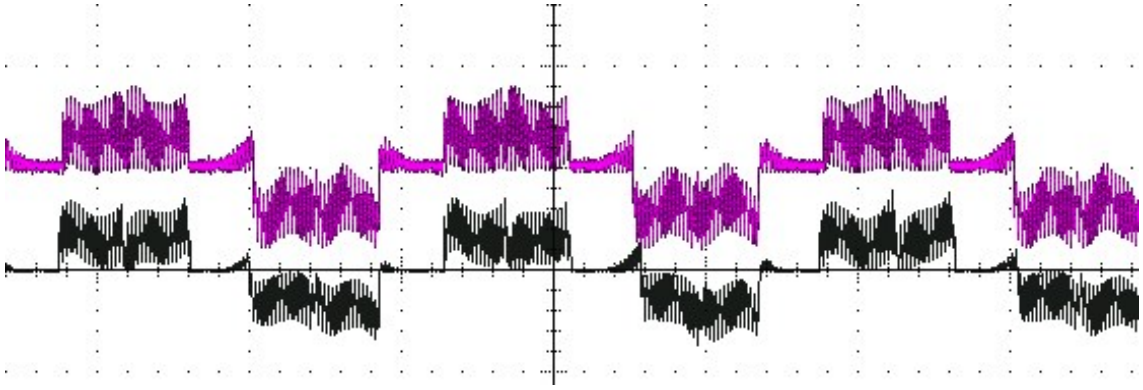


Figure 5-4: Steady-state operation: (0 p.u): Violet trace: emulator phase A current (2A/div), Black trace: motor phase A current (2A/div), time scale: 4 ms/div.

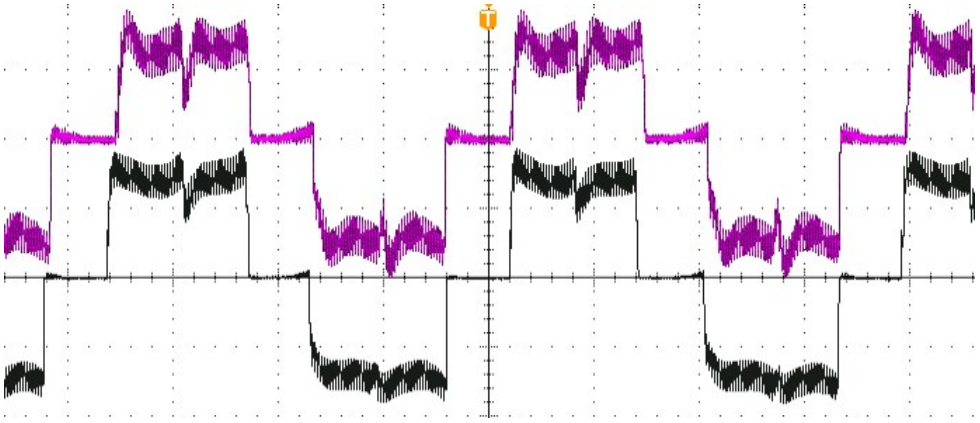


Figure 5-5: Steady-state operation: (0.8 p.u): Violet trace: emulator phase A current (5A/div), Black trace: motor phase A current (5A/div), time scale: 4 ms/div.

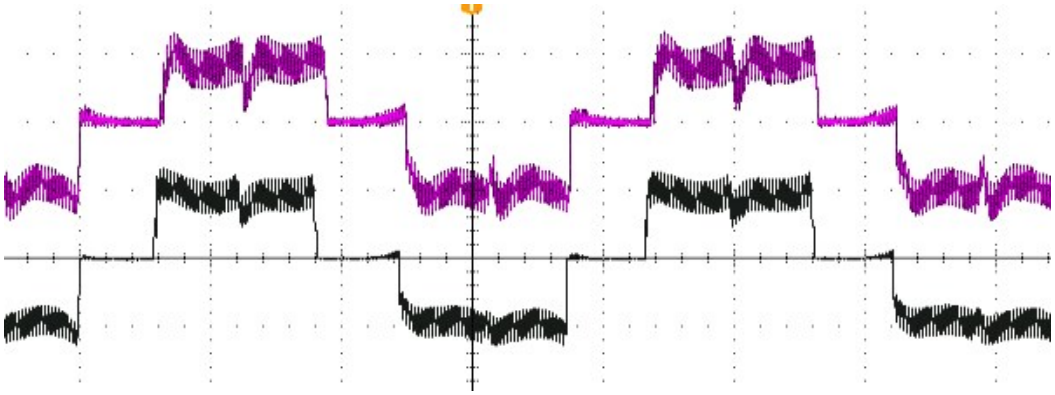


Figure 5-6: Steady-state operation: (0.5 p.u): Violet trace: emulator phase A current (5A/div), Black trace: motor phase A current (5A/div), time scale: 4 ms/div.

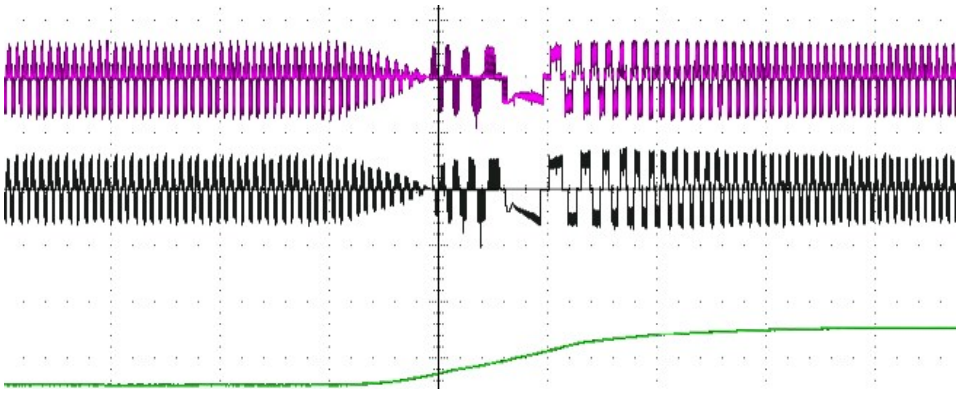


Figure 5-7: Speed reversal operation: (0 p.u): Green trace: (emulator, motor) Speed (2000 rpm/div), Violet trace: emulator phase A current (5A/div), Black trace: motor phase A current (5A/div), time scale: 200 ms/div.

b) **motor speed 1500 rpm:** the current in phase A of the emulator (violet trace) and the physical motor (black trace) during steady-state operation are shown in the figures below. In Fig.5-8 the motor and the emulator are running on no-load. In Fig.5-9 a load torque of 0.8 p.u is applied to the motor and the emulator. In Fig.5-10 a load torque of 0.5 p.u is applied to the motor and the emulator. In Fig.5-11 speed reversal from -1500 rpm to 1500 rpm is applied to both the emulator and the motor on no-load.

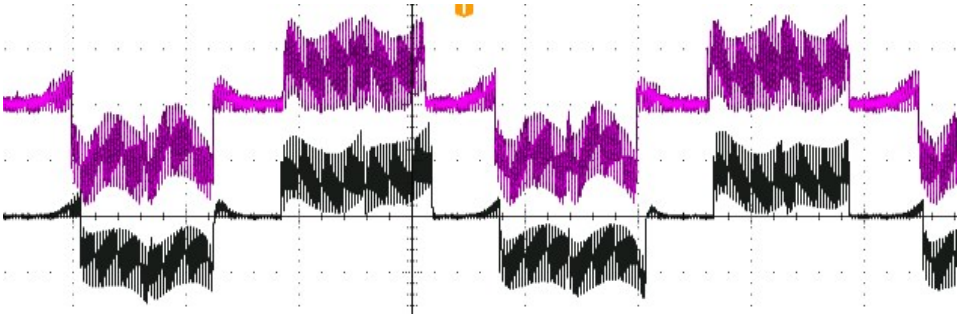


Figure 5-8: Steady-state operation: (0 p.u): Violet trace: emulator phase A current (2A/div), Black trace: motor phase A current (2A/div), time scale: 4 ms/div.

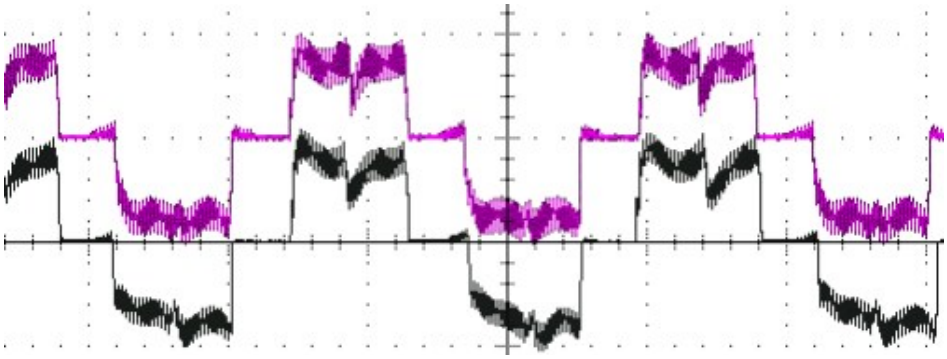


Figure 5-9: Steady-state operation: (0.8 p.u): Violet trace: emulator phase A current (10A/div), Black trace: motor phase A current (10A/div), time scale: 4 ms/div.

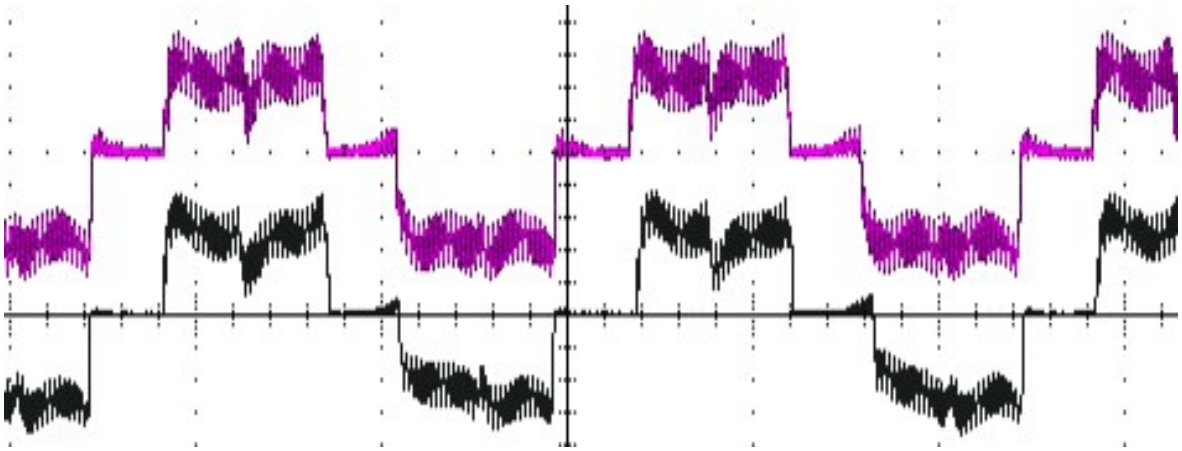


Figure 5-10: Steady-state operation: (0.5 p.u): Violet trace: emulator phase A current (10A/div), Black trace: motor phase A current (10A/div), time scale: 4 ms/div.

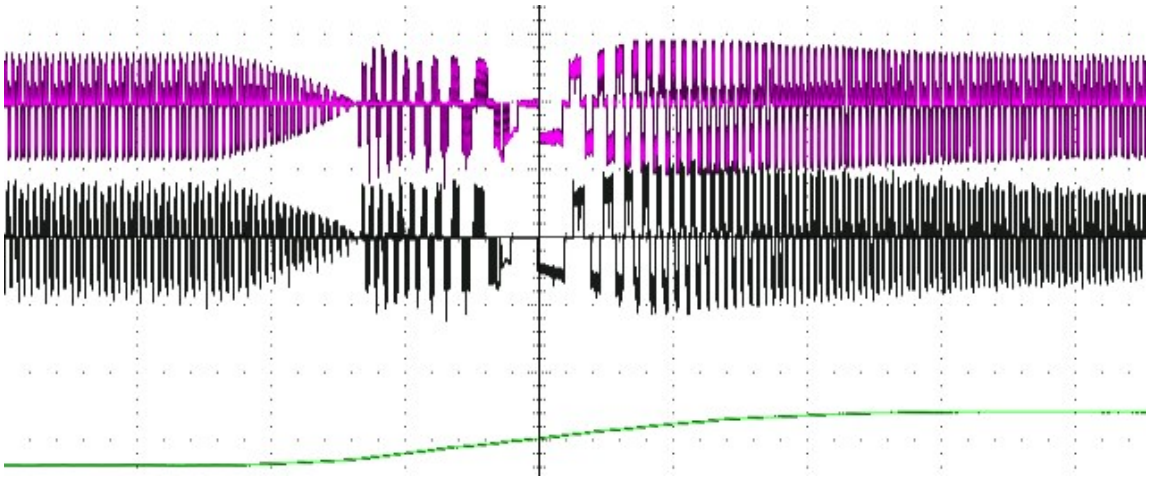


Figure 5-11: Speed reversal operation: (0 p.u, -1500 rpm to 1500 rpm): Green trace: (emulator, motor) Speed (4000 rpm/div), Violet trace: emulator phase A current (5A/div), Black trace: motor phase A current (5A/div), time scale: 200 ms/div.

c) **motor speed 2000 rpm:** the current in phase A of the emulator (violet trace) and the physical motor (black trace) during the steady state are shown in the figures below. In Fig.5-12 the motor and the emulator are running on no-load. In Fig.5-13 a load torque of 0.8 p.u is applied to the motor and the emulator. In Fig.5-14 speed reversal from -2000 rpm to 2000 rpm is applied to both the emulator and the motor on no-load.

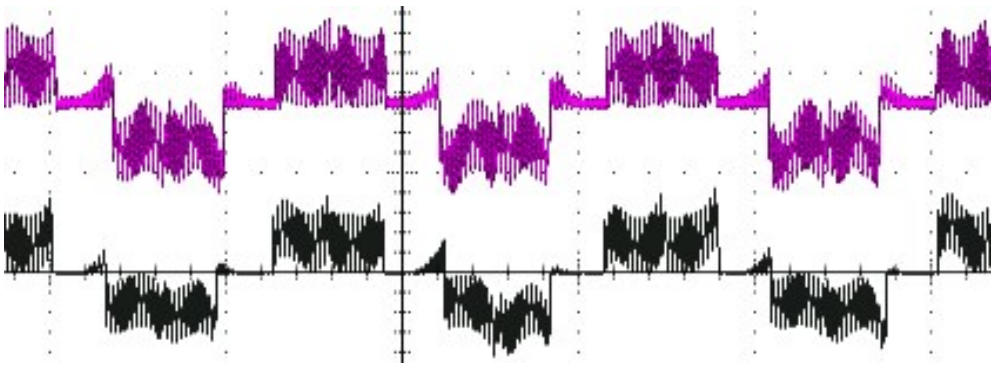


Figure 5-12: Steady-state operation: (0 p.u): Violet trace: emulator phase A current (2A/div), Black trace: motor phase A current (2A/div), time scale: 4 ms/div.

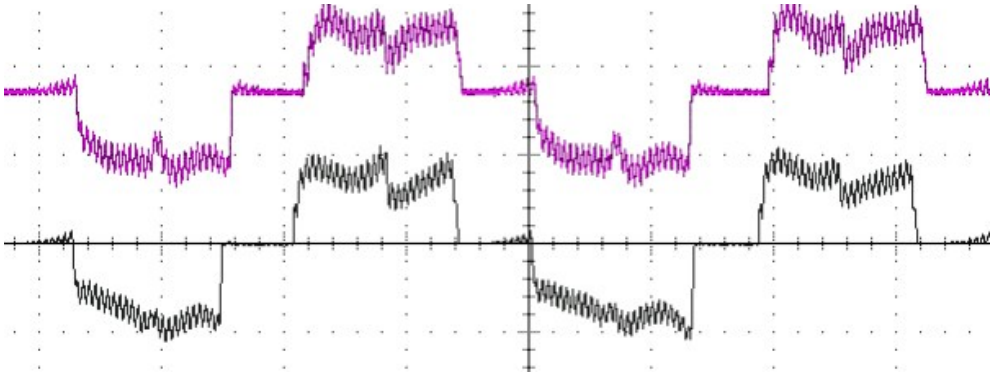


Figure 5-13: Steady-state operation: (0.8 p.u): Violet trace: emulator phase A current (10A/div), Black trace: motor phase A current (10A/div), time scale: 4 ms/div.

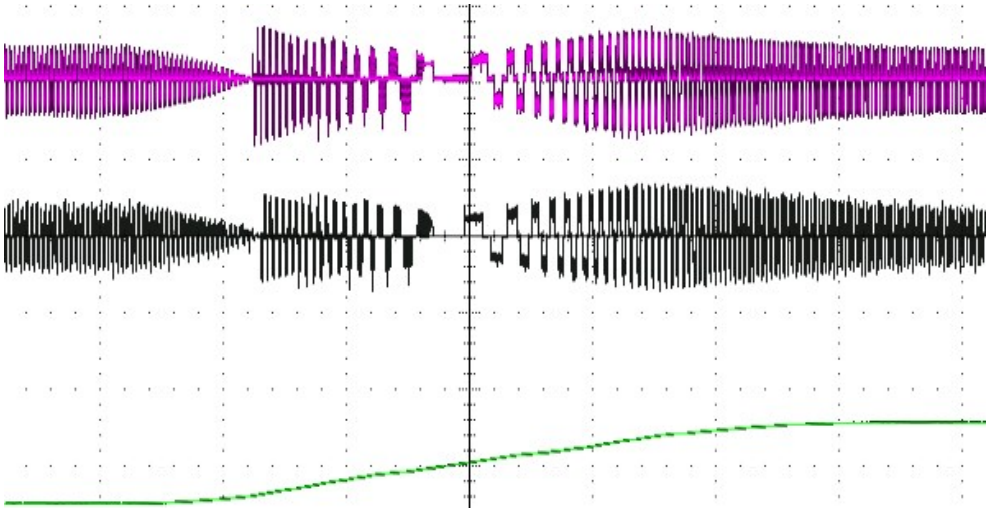


Figure 5-14: Speed reversal operation: (0 p.u, -2000 rpm to 2000 rpm): Green trace: (emulator, motor) Speed (4000 rpm/div), Violet trace: emulator phase A current (10A/div), Black trace: motor phase A current (10A/div), time scale: 200 ms/div.

5.4.2 Transient operation:

a) **Load change (0 – 0.5) p.u at 1000 rpm:** Fig.5-15 shows the operation of the BLDC emulator when a load change is applied at a constant motor model speed of 1000 rpm. The current in phase A of the emulator (violet trace) and the current-in voltage-out BLDC motor model speed (green trace) which is implemented inside the RT simulator are recorded using the oscilloscope. The emulator motor runs at 1000 rpm on no-load (0 p.u), and then a load torque of 0.5 p.u is applied to the motor, this results in an increase in the emulator current

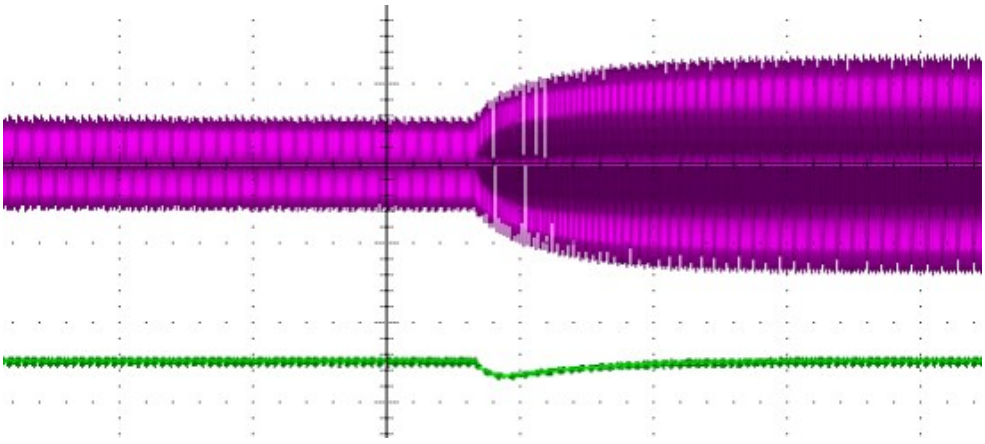


Figure 5-15: Load change from (0-0.5) p.u at 1000 rpm: Green trace: Speed (2000 rpm/div), Violet trace: phase A current (5A/div), time scale: 1 s/div.

while the speed controller regulates the speed.

b) **Load change (0 – 0.5) p.u at 2000 rpm:** Fig.5-16 shows the operation of the BLDC emulator when a load change is applied at a constant motor model speed of 2000 rpm. The current in phase A of the emulator (violet trace) and the current-in voltage-out BLDC motor model speed (green trace) which is implemented inside the RT simulator are recorded using

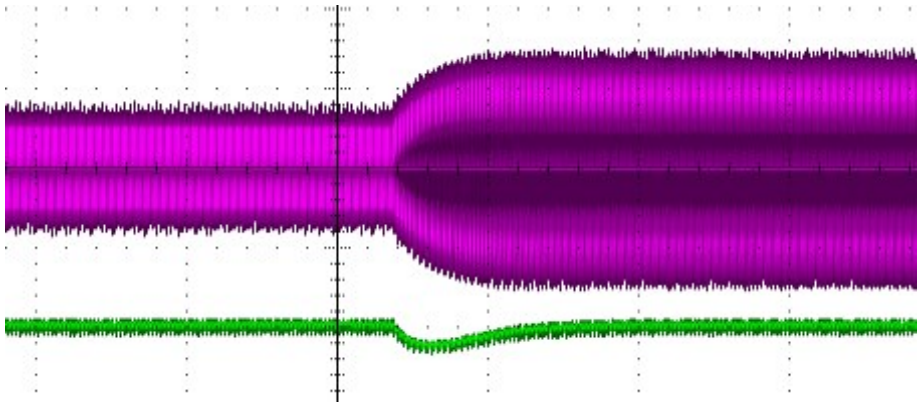


Figure 5-16: Load change from 0-0.5 p.u at 2000 rpm: green trace: Speed (4000 rpm/div), violet trace: phase A current (5A/div), time scale: 1 s/div.

the oscilloscope. The emulator motor runs at 2000 rpm on no-load (0 p.u), and then a load torque of 0.5 p.u is applied to the motor, this results in an increase in the emulator current while the speed controller regulates the speed.

c) **Speed change (500 – 1500) rpm at 0 p.u:** Fig.5-17 shows the operation of the BLDC emulator and physical motor during speed change condition on no-load. The current in phase A of the emulator (violet trace), the reference BLDC current (black trace) and the current-in voltage-out BLDC motor model speed (green trace) which is implemented inside the RT simulator are recorded using the oscilloscope. The emulator motor speed changes command from 500 rpm to 1500 rpm on no-load (0 p.u) is applied. The physical machine speed is superimposed on the emulator motor speed (green trace).

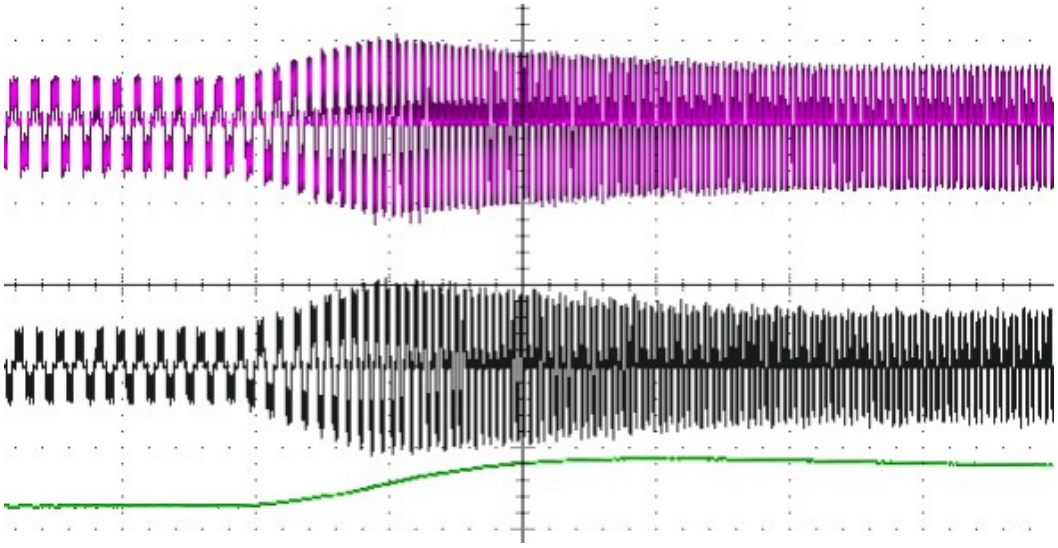


Figure 5-17: Speed change from (500 – 1500) rpm at 0 p.u: Green trace: Speed (2000 rpm/div), Violet trace: emulator phase A current (5A/div), Black trace: motor phase A current (5A/div), time scale: 200 ms/div.

d) **Speed change (1000 – 2000) rpm at 0 p.u:** Fig.5-18 shows the operation of the BLDC emulator during speed change on no load. The current in phase A of the emulator (violet trace), the reference BLDC current (black trace) and the current-in voltage-out BLDC motor model speed (green trace) which is implemented inside the RT simulator are recorded using the oscilloscope. The emulator motor model speed change command from 1000 rpm to 2000 rpm on no-load (0 p.u) is applied. The physical machine speed is superimposed on the emulator motor speed (green trace).

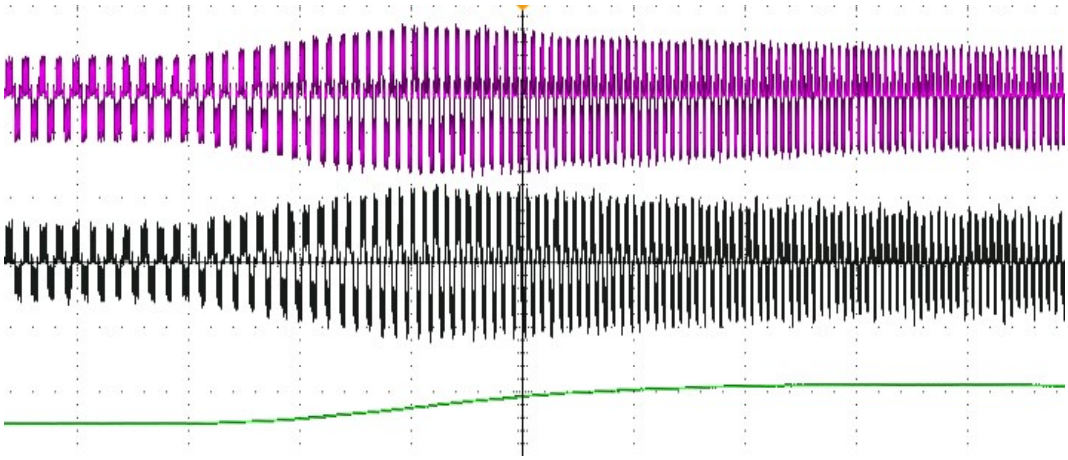


Figure 5-18: Speed change from (1000 – 2000) rpm at 0 p.u: Green trace: (emulator, motor) Speed (2000 rpm/div), Violet trace: emulator phase A current (5A/div), Black trace: motor phase A current (5A/div), time scale: 200 ms/div.

e) **Load change (0 – 0.5) p.u at 1000 rpm current comparison:** Fig.5-19 shows the operation of the BLDC emulator and physical motor during load change when running at 1000 rpm. The current in phase A of the emulator (violet trace), the reference BLDC current (black trace) and the current-in voltage-out BLDC motor model speed (green trace) which is implemented inside the RT simulator are recorded using the oscilloscope. The emulator motor runs at 1000 rpm on no-load (0 p.u), and then a load torque of 0.5 p.u is applied to the emulator motor, this results in an increase in the emulator current while the speed controller regulates the speed. The physical machine speed is superimposed on the emulator speed (green trace). The physical machine current (black trace) is subject to the same loading condition, resulting in a similar increase in current as shown in Fig.5-19.

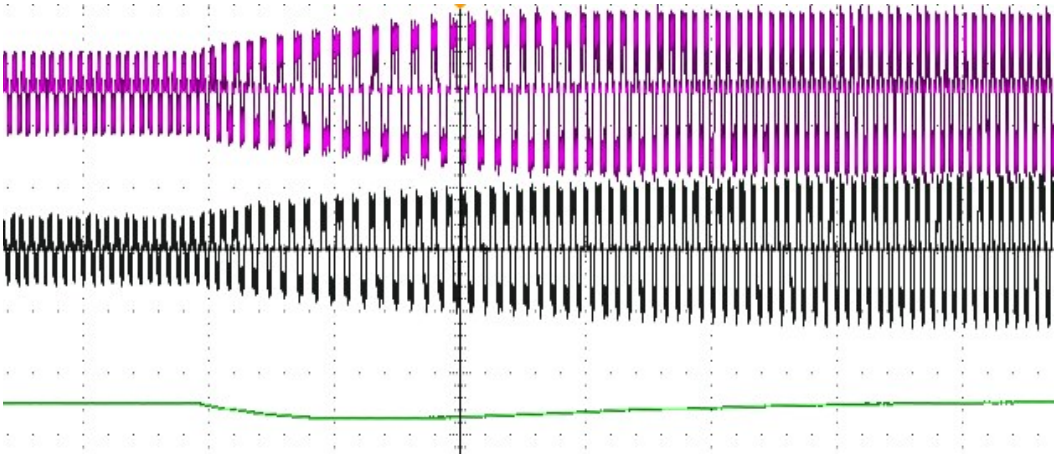


Figure 5-19: Load change from 0-0.5 p.u rpm at 1000 rpm: Green trace: (emulator, motor) Speed (2000 rpm/div), Violet trace: emulator phase A current (5A/div), Black trace: motor phase A current (5A/div), time scale: 200 ms/div.

f) **Load change (0 – 0.5) p.u at 2000 rpm current comparison:** Fig.5-20 shows the operation of the BLDC emulator and physical motor during load change at a constant speed of 2000 rpm. The current in phase A of the emulator (violet trace), the reference BLDC current (black trace) and the current-in voltage-out BLDC motor model speed (green trace) which is implemented inside the RT simulator are recorded using the oscilloscope. The emulator motor model runs at 2000 rpm on no-load (0 p.u), and then a load torque of 0.5 p.u is applied to the motor, this results in an increase in the emulator current while the speed controller regulates the speed. The physical machine speed is superimposed on the emulator speed (green trace). The physical machine current (black trace) is subject to the same loading condition, resulting in a similar increase in current as shown in Fig.5-20.

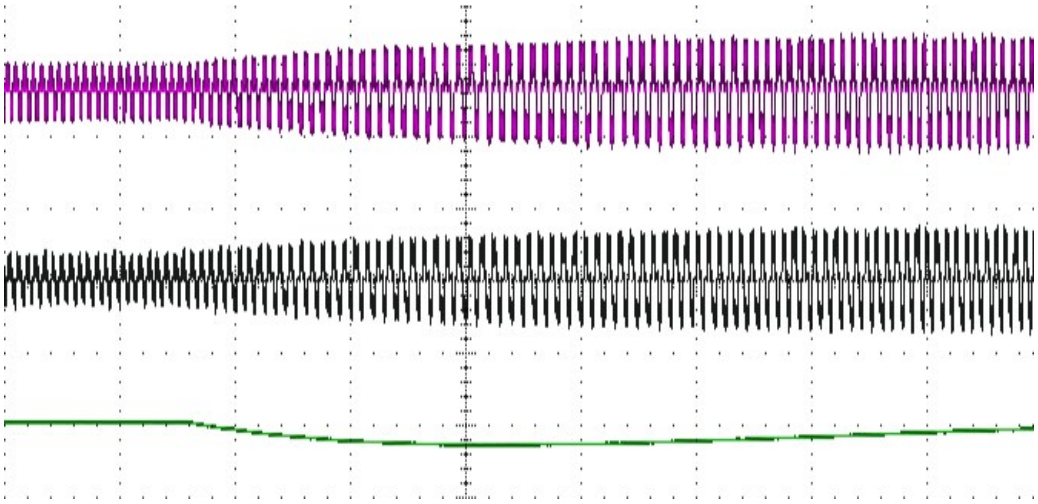


Figure 5-20: Load change from 0-0.5 p.u rpm at 2000 rpm: Green trace: (emulator, motor) Speed (2000 rpm/div), Violet trace: emulator phase A current (5A/div), Black trace: motor phase A current (5A/div), time scale: 100 ms/div.

5.4.3 Pole voltage

Fig.5-21 shows the emulator current, the motor model speed and the pole voltage measured at the drive inverter pole terminals during the steady-state operation of the emulator with a constant speed of 1500 rpm. Fig.5-22 shows the emulator current, the motor model speed and the pole voltage measured during transient operation with a speed change from 500 to 1500 rpm. These pole voltages results are similar to the physical BLDC motor drive.

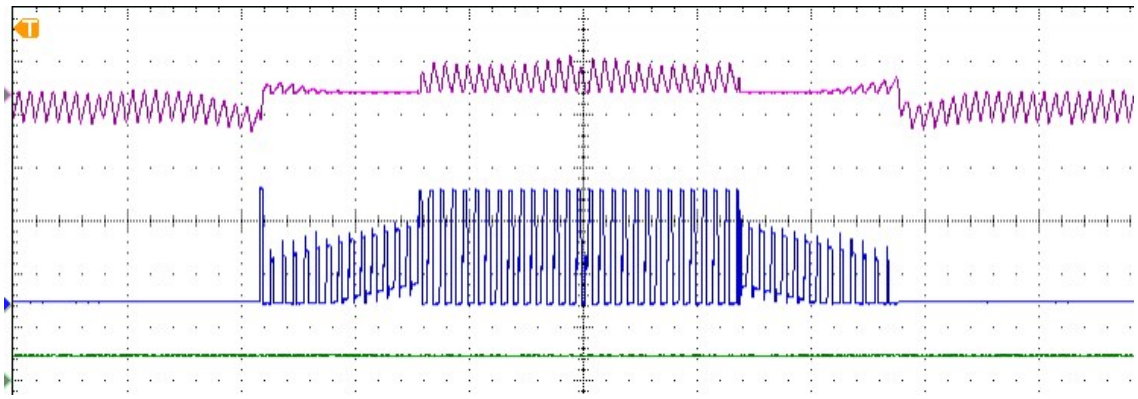


Figure 5-21: Hardware results: Steady state: Violet trace: phase current (5 A/div), Green trace: speed (3000 rpm/div), Blue trace: pole voltage (20 V/div), time scale: 1 ms/div.

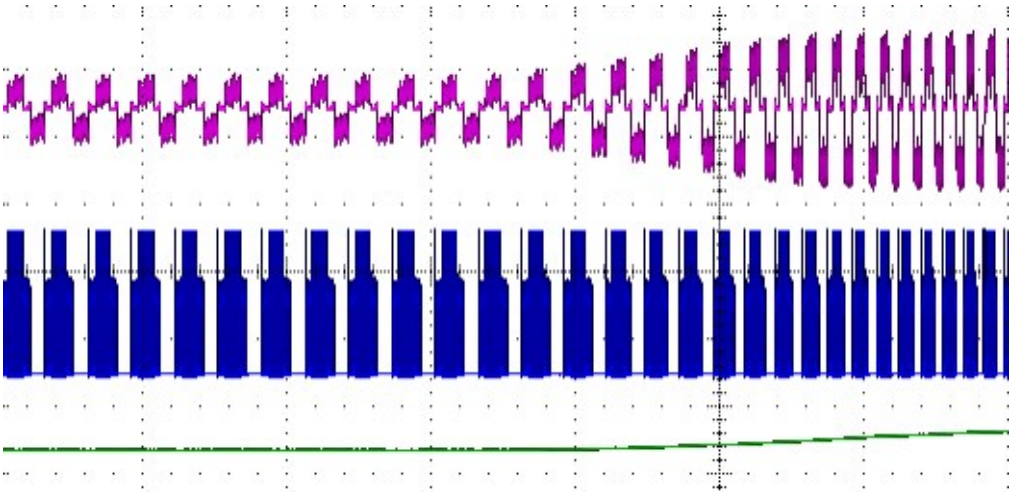


Figure 5-22: Hardware results: Transient state: Violet trace: phase current (5 A/div), Green trace: speed (2000 rpm/div), Blue trace: pole voltage (20 V/div), time scale: 100 ms/div.

5.4.4 Simulation, emulator and physical machine current waveforms

In this section, the simulation results for the steady-state and transient operation of the improved BLDC motor are compared to the emulator and physical BLDC motor currents under the same loading and speed conditions. Fig.5-23(a) shows the emulator phase A current (violet trace) and the physical BLDC motor current in phase A (black trace), Fig.5-23(b) shows the improved BLDC motor simulation current in phase A (green trace). All the results are at 500 rpm and a load torque of 0.25 p.u. Fig.5-23(c) shows the emulator phase A current (violet trace) and the physical BLDC motor current in phase A (black trace), Fig.5-23(d) shows the improved BLDC motor simulation

current in phase A (green trace), a speed change from 500 rpm to 1000 rpm at no-load is applied to all the motors for transient operation.

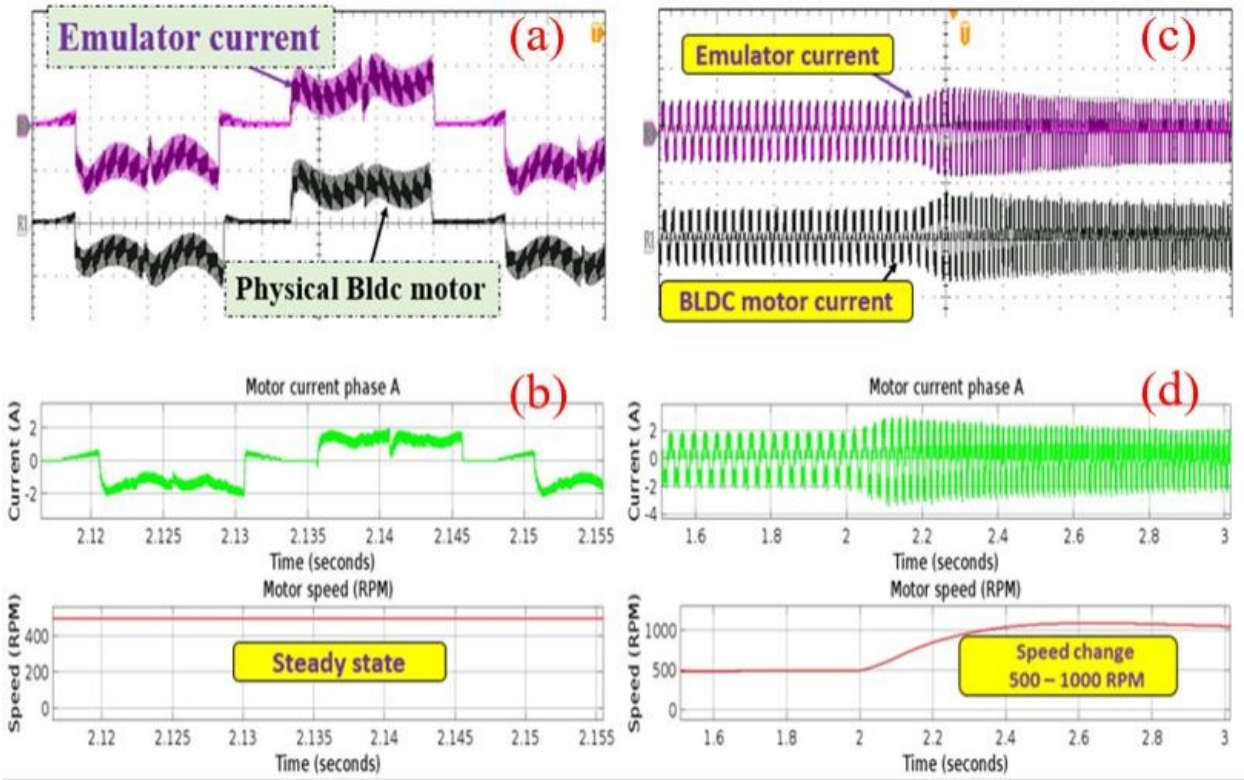


Figure 5-23: Steady-state operation: (a) and (c) Emulator BLDC motor current phase A (violet trace), physical BLDC motor current phase A (black trace), (b) &(d)Improved BLDC motor simulation current (green trace), speed (red trace). Transient operation: (500 rpm to 1000 rpm, 0 p.u).

Chapter 6: Conclusion and future works

6.1 Conclusion

A BLDC motor emulator setup using an open-loop emulation scheme is built. The results obtained from the emulator are compared against the results obtained from a physical motor. The close match of the results proves that the emulator system motor is a viable alternative to dynamometer testing of BLDC drive systems.

The conclusion on the work presented in each chapter is given below:

Chapter 1: An electric drive system is made up of various parts among which the motor and the power converters are the most critical components. In electric drive applications, the use of a dynamometer testbench is a well-established conventional drive testing method used for testing electric drive systems. However, the use of this mechanical testbench suffers from various problems that have led to the development of an electrical testbench instead which uses PHIL machine emulation (virtual machine) as an alternative to the conventional testing method.

Chapter 2: The accuracy of a software simulation depends on the system model used. The conventional mathematical BLDC motor model does not correctly model some of the salient features of the waveforms of a BLDC drive utilizing trapezoidal control. The improved BLDC motor model presented in this chapter modelled correctly all the features of the waveform of a BLDC drive utilizing trapezoidal control. The model was validated by comparing the simulation results against the experimental hardware results of the drive for the same speed and loading conditions, both during steady-state and transient conditions.

Chapter 3: The BLDC drive hardware setup was developed, implemented and tested at various speeds and load conditions. The simulation of the emulation of permanent magnets (PMSM and BLDC) was presented in this chapter, for PMSM the simulation of the closed-loop emulation of the PMSM was developed and implemented successfully by comparing the simulation results of the PMSM and the emulator currents for the same speeds and load conditions. The simulation of the open-loop emulation of the BLDC motor drive with trapezoidal control was developed and implemented, a current-in voltage-out BLDC motor model was developed and validated using the improved BLDC motor simulation results. The current-in voltage-out model was then used in the simulation of the open-loop emulation of the BLDC drive, and the results of the emulator currents were compared to the improved BLDC motor model, thus validating the implementation.

Chapter 4: A brief introduction to the three-phase two-level inverter prototyping and testing was given in this chapter. A double pulse test was used to analyse the switching waveforms of the gate driver and the inverter output voltages at different load currents. The inverter was loaded using a passive load, and the temperature test was performed on the inverter. The inverter was later used as the power electronic converter (drive inverter) in the BLDC drive system.

Chapter 5: A BLDC motor emulator hardware setup was developed, implemented and then tested. The open-loop emulation control mode was implemented; hence the coupling element values were chosen to match the values of the real machine. The coupling element resistance and inductance values chosen in this implementation have a resistance of 0.15 Ω and inductance of 0.155 uH which is different from the mean motor inductance value of 0.164 uH. Because the hardware setup used in this work has the BLDC motor coupled to the dc generator, the overall values of the system frictional torque constant (B) and inertia (J) vary as a function of the speed, hence only estimated values are used for the emulator BLDC motor model B and J. Open-loop emulation control is very sensitive to motor parameter variations, any mismatch between the emulator and the physical motor inductance, resistance, frictional torque constant and inertia value results into inaccuracy in the emulator currents values. However, open-loop emulation control is easy to implement and does not require a complex motor model in cases where high accuracy of emulation is not required, open-loop emulation can be used. The close match between the physical BLDC motor and the emulator currents results for steady-state and transient at various speeds and load conditions validate the emulation presented in this thesis.

6.2 Future work

Some of the future work are:

- The correct pole voltages output by the improved BLDC motor model presented in this thesis can be used to simulate and implement sensorless trapezoidal control of the BLDC.
- PHIL emulation of a sensorless trapezoidal drive using open-loop emulation control.
- Further research related to the emulation of BLDC motors with torque ripple compensation algorithm of the emulation of machine or drive inverter faults can be investigated in future work.

References:

[1] IEA. World Energy Outlook 2019; IEA Publications: Paris, France, 2019.

[2] S. N. Manias, "Introduction to Motor Drive Systems," in Power Electronics and Motor Drive Systems, Amsterdam: Elsevier, 2017, pp. 853–860.

[3] B. K. Bose, *Modern Power Electronics and AC drives*, New Delhi: PHI Learning, 2012.

[4] H. Abu-Rub, M. Malinowski, and K. Al-Haddad, *Power Electronics for Renewable Energy Systems, transportation, and Industrial Applications*, Chichester, West Sussex, United Kingdom: Wiley/IEEE, 2014.

[5] T. Schulte and J. Bracker, "Real-time simulation of BLDC Motors for hardware-in-the-loop applications incorporating sensorless control," *2008 IEEE International Symposium on Industrial Electronics*, 2008.

[6] D. Martinez, D. Celeita, D. Clavijo, and G. Ramos, "Hardware and software integration as a realist SCADA environment to test protective relaying control," *2017 IEEE Industry Applications Society Annual Meeting*, 2017.

[7] E. Tremblay, A. Kachurowski, D. Leschert, and K. Busby, "Protective relay upgrade utilizing real-time simulator technology: Application in an Industrial Facility," *IEEE Industry Applications Magazine*, vol. 27, no. 4, pp. 57–68, 2021.

[8] A. L. Coelho, C. E. Carrer, C. A. Guerrero, and P. M. Silveira, "Loss-of-excitation protection and underexcitation controls correlation for synchronous generators in a real time Digital Simulator," *2014 IEEE Industry Application Society Annual Meeting*, 2014.

[9] K. A. Sashidhar and P. Pillay, "Emulation of Permanent Magnet Machines," thesis, Concordia University, Montreal, Quebec, 2020.

[10] K. S. Amitkumar, R. S. Kaarthik, and P. Pillay, "A versatile power-hardware-in-the-loop-based emulator for rapid testing of Transportation Electric drives," *IEEE Transactions on Transportation Electrification*, vol. 4, no. 4, pp. 901–911, 2018.

[11] N. Sharma, Y. Liu, G. Mademlis, and X. Huang, "Design of a power hardware-in-the-loop test bench for a traction permanent magnet synchronous machine drive," *2020 International Conference on Electrical Machines (ICEM)*, 2020.

[12] J. Noon, H. Song, B. Wen, I. Cvetkovic, S. Srdic, G. Pammer, and R. Burgos, "Design and evaluation of a power hardware-in-the-loop machine emulator," *2020 IEEE Energy Conversion Congress and Exposition (ECCE)*, 2020.

[13] R. S. Kaarthik and P. Pillay, "Emulation of a permanent magnet synchronous generator in real-time using power hardware-in-the-loop," *2016 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*, 2016.

[14] J. Poon, E. Chai, I. Celanovic, A. Genic, and E. Adzic, "High-fidelity real-time hardware-in-the-loop emulation of PMSM inverter drives," *2013 IEEE Energy Conversion Congress and Exposition*, 2013.

[15] N. Sharma, Y. Liu, G. Mademlis, and X. Huang, "Design of a power hardware-in-the-loop test bench for a traction permanent magnet synchronous machine drive," *2020 International Conference on Electrical Machines (ICEM)*, 2020.

[16] H. Guo, B. Zhou, J. Li, F. Cheng, and L. Zhang, "Real-time simulation of BLDC-based wind turbine emulator using RT-Lab," *2009 International Conference on Electrical Machines and Systems*, 2009.

[17] V. R. Moorthi, Power Electronics: Devices, Circuits, and industrial applications. New Delhi: Oxford Univ. Press, 2010.

[18] J. A. N. A. MELKEBEEK, Electrical Machines and drives: Fundamentals and Advanced Modelling. SPRINGER INTERNATIONAL PU, 2019.
[19] S. Jeevananthan, V. Kamaraj, J. Ravishankar, and Kamaraj, Emerging solutions for e-mobility and smart grids. Springer, 2021.
[20] P. Pillay and R. Krishnan, "Application characteristics of permanent magnet synchronous and brushless DC motors for servo drives," IEEE Transactions on Industry Applications, vol. 27, no. 5, pp. 986–996, 1991.
[21] C.-liang Xia, *Permanent magnet brushless DC motor drives and controls*. New Jersey: Wiley, 2012.
[22] P. Pillay and R. Krishnan, "Modeling, simulation, and analysis of permanent-magnet motor drives. II. the brushless DC Motor Drive," IEEE Transactions on Industry Applications, vol. 25, no. 2, pp. 274–279, 1989.
[23] Infineon AP16117, "Speed control of BLDC motor with Hall Sensor using DAvE Drive." Infineon, Jul-2007.
[24] B. Y. O. U. N. G.-K. U. K. LEE and M. E. H. R. D. A. D. EHSANI, "Advanced simulation model for brushless DC motor drives," *Electric Power Components and Systems*, vol. 31, no. 9, pp. 841–868, 2003.
[25] Jianwen Shao, D. Nolan, and T. Hopkins, "A novel direct back EMF detection for sensorless brushless DC (BLDC) motor drives," *APEC. Seventeenth Annual IEEE Applied Power Electronics Conference and Exposition (Cat. No.02CH37335)*.
[26] R. Krishnan, *Permanent magnet synchronous and brushless DC motor drives*. CRC Press, 2017.
[27] (2008) Hands-on Workshop: Motor Control Part 4 - Brushless DC Motors Made Easy. NXP. Available at: https://www.nxp.com/files-static/training_presentation/TP_FTF2008_MOTORCTRL_PART4.pdf (Accessed: December 3, 2022).
[28] (2008) Hands-on Workshop: Motor Control Part 4 - Brushless DC Motors Made Easy. NXP. Available at: https://www.nxp.com/files-static/training_presentation/TP_FTF2008_MOTORCTRL_PART4.pdf (Accessed: December 3, 2022).
[29] K. A. Sashidhar and P. Pillay, "Emulation of Permanent Magnet Machines," thesis, Concordia University, Montreal, Quebec, 2020.
[30] Bausiè re R., F. Labrique, and G. Seguier, "Switching Power Semiconductor Devices," in *Power Electronic Converters*, Berlin, New York: Springer-Verlag, 1993, pp. 58–68.
[31] S. Ben-Yaakov, *MOSFET datasheet – Part I*. YouTube, 2018.
[32] S. Ben-Yaakov, *Continuing education lectures: MOSFET datasheet - Part 2*. YouTube, 2018.
[33] A. Huang, "Application Note OptiMOS™ Power MOSFET Datasheet." Infineon Technologies Austria AG, Villach, 26-Mar-2012.
[34] A. Sattar, "IXYS Power MOSFET Datasheet Parameters Definition Essential Ratings and Characteristics." IXYS Corporation, 2008.
[35] P. S. Bimbhra, "POWER SEMICONDUCTORS DIODES AND TRANSISTORS, INVERTERS," in *Power Electronics*, Delhi, India: Khanna Pub., 2010, pp. 7, 309–23, 311.

[36] L. Balogh, “Fundamentals of MOSFET and IGBT Gate Driver Circuits.” Texas Instruments Incorporated, Dallas, 2018.
[37] P. Integrations, “2+ 2SC0115T Preliminary Description & Application Manual.” Power Integrations, Switzerland, 2016.
[38] S. Ben-Yaakov, *Power MOSFET drivers*. YouTube, 2018.
[39] Xue Dingyu¨ and Y. Chen, “Introduction to System Simulation Techniques and Applications,” in *System simulation techniques with MATLAB and simulink*, Chichester (UK), West Sussex: John Wiley & Sons, 2014, pp. 1–9.
[40] K. Popovici and P. J. Mosterman, “Real-Time Simulation Using Hybrid Models,” in *Real-time simulation technologies: Principles, methodologies, and applications*, Boca Raton, FL: CRC Press, 2017, pp. 4–8.
[41] J. Bélanger, P. Venne, and J.-N. Paquin, *The What, Where and Why of Real-Time Simulation*, pp. 37–49, 2010.
[42] Erik de Jong, Roald de Graaff, Peter Vaessen, Paul Crolla, Andrew Roscoe, Felix Lehfuß, Georg Lauss, Panos Kotsampopoulos, and Francisco Gafaro, *European White Book on Real-Time Powerhardware-in-the-Loop testing*, pp. 2–15, Dec. 2011.
[43] “Power hardware in the Loop │ HIL simulation │ opal-RT,” *OPAL*, 18-Jul-2022. [Online]. Available: https://www.opal-rt.com/power-hardware-in-the-loop/. [Accessed: 12-Mar-2023].

Appendix 1: Improved BLDC motor model

```
function [Vpabc, theta, We, Wr, RPM, Eabc, Iabc, Te] = fcn(Vdc, Vabc, float, Tl, J, Pole, B, Ke, R, L_M,
Ts)

% Variable declaration
persistent theta_rad;   if      isempty(theta_rad)  theta_rad = 0;   end

persistent Wrm;         if      isempty(Wrm)        Wrm = 0.01;         end

persistent Ea;         if       isempty(Ea)          Ea = 0;         end

persistent Eb;         if       isempty(Eb)          Eb = 0;         end

persistent Ec;         if       isempty(Ec)          Ec = 0;         end

persistent Iab;         if       isempty(Iab)          Iab = 0;         end

persistent Ibc;         if       isempty(Ibc)          Ibc = 0;         end

persistent Ica;         if       isempty(Ica)          Ica = 0;         end

persistent Ia;         if       isempty(Ia)          Ia = 0;         end

persistent Ib;         if       isempty(Ib)          Ib = 0;         end

persistent Ic;         if       isempty(Ic)          Ic = 0;         end

persistent Ipa;         if      isempty(Ipa)         Ipa = 0;         end

persistent Ipb;         if      isempty(Ipb)         Ipb = 0;         end

persistent Ipc;         if      isempty(Ipc)         Ipc = 0;         end

Va = Vabc(1);
Vb = Vabc(2);
Vc = Vabc(3);

Vpa = 0;
Vpb = 0;
Vpc = 0;
Vn = 0;

% Speed calculation

% if (Wrm <= 0.0001 && Wrm >= 0)
%     Wrm = -1.00001;
% end

if theta_rad < 0
    theta_rad = ((2*pi) + theta_rad);
end

if theta_rad > (2 * pi)
    theta_rad = (theta_rad - (2*pi));
end

theta_rad = theta_rad + (Wrm * (Pole/2) * Ts);

% SHOULD YOU USE ELECTRICAL OR MECHANICAL DEGREES

theta = theta_rad;

%  Back emf constant because Ke is line to line, hence Ke/2
E = Wrm * Ke *0.5;

% % %%Trapezoidal back-emf
% if (0<=theta) &&(theta<(pi/6))                Ea = ((6*E/pi)*theta);
```

```
% elseif ((pi/6)<=theta) && (theta<(5*pi/6))     Ea = E;
% elseif ((5*pi/6)<=theta) && (theta<(7*pi/6))  Ea = -((6*E/pi)*theta)+(6*E);
% elseif ((7*pi/6)<=theta) && (theta<(11*pi/6)) Ea = -E;
% elseif ((11*pi/6)<=theta) &&(theta<(2*pi))     Ea = ((6*E/pi)*theta)-(12*E);
% end
%
% if ((0<=theta) &&(theta<(pi/2)))              Eb = -E;
% elseif ((pi/2)<=theta) && (theta<(5*pi/6))     Eb = ((6*E/pi)*theta)-(4*E);
% elseif ((5*pi/6)<=theta) && (theta<(9*pi/6))  Eb = E;
% elseif ((9*pi/6)<=theta) && (theta<(11*pi/6)) Eb = -((6*E/pi)*theta)+(10*E);
% elseif ((11*pi/6)<=theta) &&(theta<(2*pi))     Eb = -E;
% end
%
% if (0<=theta) &&(theta<(pi/6))                Ec = E;
% elseif ((pi/6)<=theta) && (theta<(pi/2))       Ec = -((6*E/pi)*theta)+(2*E);
% elseif ((pi/2)<=theta) && (theta<(7*pi/6))     Ec = -E;
% elseif ((7*pi/6)<=theta) && (theta<(9*pi/6))  Ec = ((6*E/pi)*theta)-(8*E);
% elseif ((9*pi/6)<=theta) &&(theta<(2*pi))      Ec = E;
% end

%%Quasi-trapezoidal back-emf
%Ea = E * (8* sqrt(2) / (pi^2) )*( sin (theta) +((1/9) * sin(3*theta) )-( (1/25) * sin(5*theta) )-( (1/49)
* sin(7*theta) )+( (1/81) * sin(9*theta) )+( (1/121) * sin(11*theta) ));
%Eb = E * (8* sqrt(2) / (pi^2) )*( sin (theta - (2*pi/3) ) +((1/9) * sin(3*( theta - (2*pi/3))) )...  %
                -( (1/25) * sin(5*( theta - (2*pi/3))) )-( (1/49) * sin(7*( theta - (2*pi/3))) )...   %
                +( (1/81) * sin(9*( theta - (2*pi/3))) )+( (1/121) * sin(11*( theta - (2*pi/3))) ));
%Ec = E * (8* sqrt(2) / (pi^2) )*( sin ( theta + (2*pi/3) ) +((1/9) * sin(3*( theta + (2*pi/3))) )...
%               -( (1/25) * sin(5*( theta + (2*pi/3))) )-( (1/49) * sin(7*( theta + (2*pi/3))) )...   %
                +( (1/81) * sin(9*( theta + (2*pi/3))) )+( (1/121) * sin(11*( theta + (2*pi/3))) ));

%%Sinusoidal back-emf
Ea = E * (sin (theta));
 Eb = E * (sin (theta - (2*pi/3)));
 Ec = E * (sin (theta + (2*pi/3)));


%Curent determination
Eab = Ea - Eb;
Ebc = Eb - Ec;
Eca = Ec - Ea;

%Previous currents
Ipa = Ia;
Ipb = Ib;
Ipc = Ic;

%A-phase floating
fa = float(1);
fb = float(2);
fc = float(3);

%All floating or all switching####invalid
if ((fa && fb && fc) || (~fa && ~fb && ~fc))

    Vpa = Vdc/2;
    Vpb = Vdc/2;
    Vpc = Vdc/2;

else

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %A-phase floating
    if(fa && ~fb && ~fc)

        if Ia > 0

            Vpa = 0;
            Vpb = Vb;
```

```
    Vpc = Vc;

    Vab = Vpa - Vpb;
    Vbc = Vpb - Vpc;
    Vca = Vpc - Vpa;

    Iab = Iab + ((1/(L_M)) * (Vab - Eab - R * Iab) * Ts);
    Ibc = Ibc + ((1/(L_M)) * (Vbc - Ebc - R * Ibc) * Ts);
    Ica = Ica + ((1/(L_M)) * (Vca - Eca - R * Ica) * Ts);

    Ia = (Iab - Ica) / 3;
    Ib = (Ibc - Iab) / 3;
    Ic = (Ica - Ibc) / 3;

elseif Ia < 0

    Vpa = Vdc;
    Vpb = Vb;
    Vpc = Vc;

    Vab = Vpa - Vpb;
    Vbc = Vpb - Vpc;
    Vca = Vpc - Vpa;

    Iab = Iab + ((1/(L_M)) * (Vab - Eab - R * Iab) * Ts);
    Ibc = Ibc + ((1/(L_M)) * (Vbc - Ebc - R * Ibc) * Ts);
    Ica = Ica + ((1/(L_M)) * (Vca - Eca - R * Ica) * Ts);

    Ia = (Iab - Ica) / 3;
    Ib = (Ibc - Iab) / 3;
    Ic = (Ica - Ibc) / 3;

end

if (Ipa * Ia <= 0)

    Vpa = ((Vb + Vc) / 2) - ((Eb + Ec) / 2) + Ea;
    Vpb = Vb;
    Vpc = Vc;

    if(Vpa < 0)

        Vpa = 0;

        Vab = Vpa - Vpb;
        Vbc = Vpb - Vpc;
        Vca = Vpc - Vpa;

        Iab = Iab + ((1/(L_M)) * (Vab - Eab - R * Iab) * Ts);
        Ibc = Ibc + ((1/(L_M)) * (Vbc - Ebc - R * Ibc) * Ts);
        Ica = Ica + ((1/(L_M)) * (Vca - Eca - R * Ica) * Ts);

        Ia = (Iab - Ica) / 3;
        Ib = (Ibc - Iab) / 3;
        Ic = (Ica - Ibc) / 3;

    else

        Vab = Vpa - Vpb;
        Vbc = Vpb - Vpc;
        Vca = Vpc - Vpa;

        Iab = Iab + ((1/(L_M)) * (Vab - Eab - R * Iab) * Ts);
        Ibc = Ibc + ((1/(L_M)) * (Vbc - Ebc - R * Ibc) * Ts);
        Ica = Ica + ((1/(L_M)) * (Vca - Eca - R * Ica) * Ts);

        Ia = (Iab - Ica) / 3;
        Ia = 0;
```

```
            Ib = (Ibc - Iab) / 3;
            Ic = (Ica - Ibc) / 3;

        end

     end

%B-phase floating
elseif (fb && ~fc && ~fa)

    if Ib > 0

        Vpa = Va;
        Vpb = 0;
        Vpc = Vc;

        Vab = Vpa - Vpb;
        Vbc = Vpb - Vpc;
        Vca = Vpc - Vpa;

        Iab = Iab + ((1/(L_M)) * (Vab - Eab - R * Iab) * Ts);
        Ibc = Ibc + ((1/(L_M)) * (Vbc - Ebc - R * Ibc) * Ts);
        Ica = Ica + ((1/(L_M)) * (Vca - Eca - R * Ica) * Ts);

        Ia = (Iab - Ica) / 3;
        Ib = (Ibc - Iab) / 3;
        Ic = (Ica - Ibc) / 3;

    elseif Ib < 0

        Vpa = Va;
        Vpb = Vdc;
        Vpc = Vc;

        Vab = Vpa - Vpb;
        Vbc = Vpb - Vpc;
        Vca = Vpc - Vpa;

        Iab = Iab + ((1/(L_M)) * (Vab - Eab - R * Iab) * Ts);
        Ibc = Ibc + ((1/(L_M)) * (Vbc - Ebc - R * Ibc) * Ts);
        Ica = Ica + ((1/(L_M)) * (Vca - Eca - R * Ica) * Ts);

        Ia = (Iab - Ica) / 3;
        Ib = (Ibc - Iab) / 3;
        Ic = (Ica - Ibc) / 3;

    end

     if (Ipb * Ib <= 0)

        Vpa = Va;
        Vpb = ((Va + Vc) / 2) - ((Ea + Ec) / 2) + Eb;
        Vpc = Vc;

        if(Vpb < 0)

            Vpb = 0;

            Vab = Vpa - Vpb;
            Vbc = Vpb - Vpc;
            Vca = Vpc - Vpa;

            Iab = Iab + ((1/(L_M)) * (Vab - Eab - R * Iab) * Ts);
            Ibc = Ibc + ((1/(L_M)) * (Vbc - Ebc - R * Ibc) * Ts);
            Ica = Ica + ((1/(L_M)) * (Vca - Eca - R * Ica) * Ts);

            Ia = (Iab - Ica) / 3;
            Ib = (Ibc - Iab) / 3;
```

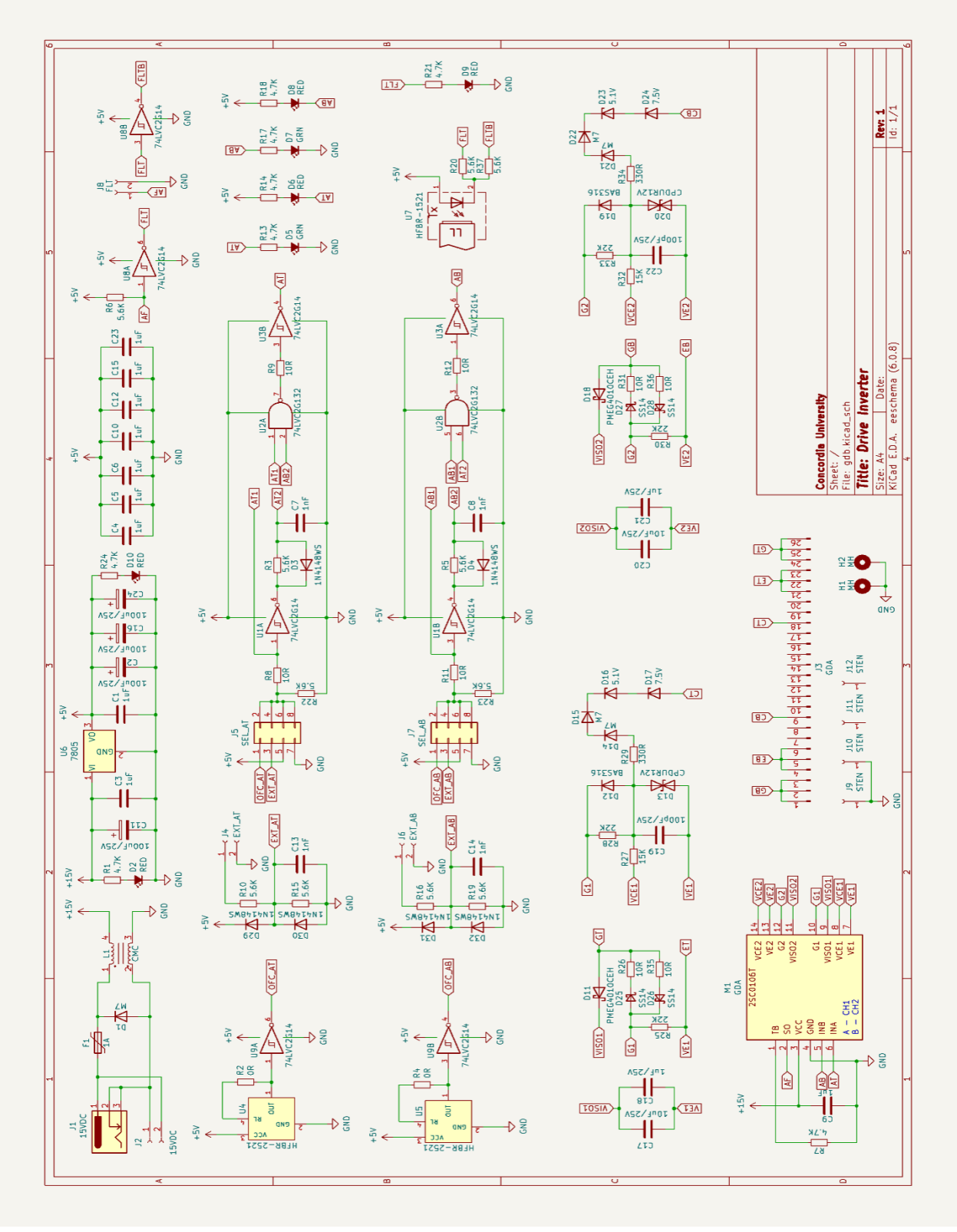
```
            Ic = (Ica - Ibc) / 3;

        else

            Vab = Vpa - Vpb;
            Vbc = Vpb - Vpc;
            Vca = Vpc - Vpa;

            Iab = Iab + ((1/(L_M)) * (Vab - Eab - R * Iab) * Ts);
            Ibc = Ibc + ((1/(L_M)) * (Vbc - Ebc - R * Ibc) * Ts);
            Ica = Ica + ((1/(L_M)) * (Vca - Eca - R * Ica) * Ts);

            Ia = (Iab - Ica) / 3;
            Ib = (Ibc - Iab) / 3;
            Ib = 0;
            Ic = (Ica - Ibc) / 3;

        end

     end

%C-phase floating
elseif(fc && ~fa && ~fb)

    if Ic > 0

        Vpa = Va;
        Vpb = Vb;
        Vpc = 0;

        Vab = Vpa - Vpb;
        Vbc = Vpb - Vpc;
        Vca = Vpc - Vpa;

        Iab = Iab + ((1/(L_M)) * (Vab - Eab - R * Iab) * Ts);
        Ibc = Ibc + ((1/(L_M)) * (Vbc - Ebc - R * Ibc) * Ts);
        Ica = Ica + ((1/(L_M)) * (Vca - Eca - R * Ica) * Ts);

        Ia = (Iab - Ica) / 3;
        Ib = (Ibc - Iab) / 3;
        Ic = (Ica - Ibc) / 3;

    elseif Ic < 0

        Vpa  =  Va;
        Vpb  =  Vb;
        Vpc = Vdc;

        Vab = Vpa - Vpb;
        Vbc = Vpb - Vpc;
        Vca = Vpc - Vpa;

        Iab = Iab + ((1/(L_M)) * (Vab - Eab - R * Iab) * Ts);
        Ibc = Ibc + ((1/(L_M)) * (Vbc - Ebc - R * Ibc) * Ts);
        Ica = Ica + ((1/(L_M)) * (Vca - Eca - R * Ica) * Ts);

        Ia = (Iab - Ica) / 3;
        Ib = (Ibc - Iab) / 3;
        Ic = (Ica - Ibc) / 3;

    end

     if (Ipc * Ic <= 0)

        Vpa = Va;
        Vpb = Vb;
        Vpc = ((Va + Vb) / 2) - ((Ea + Eb) / 2) + Ec;
```

```
            if(Vpc < 0)

                Vpc = 0;

                Vab = Vpa - Vpb;
                Vbc = Vpb - Vpc;
                Vca = Vpc - Vpa;

                Iab = Iab + ((1/(L_M)) * (Vab - Eab - R * Iab) * Ts);
                Ibc = Ibc + ((1/(L_M)) * (Vbc - Ebc - R * Ibc) * Ts);
                Ica = Ica + ((1/(L_M)) * (Vca - Eca - R * Ica) * Ts);

                Ia = (Iab - Ica) / 3;
                Ib = (Ibc - Iab) / 3;
                Ic = (Ica - Ibc) / 3;

            else

                Vab = Vpa - Vpb;
                Vbc = Vpb - Vpc;
                Vca = Vpc - Vpa;

                Iab = Iab + ((1/(L_M)) * (Vab - Eab - R * Iab) * Ts);
                Ibc = Ibc + ((1/(L_M)) * (Vbc - Ebc - R * Ibc) * Ts);
                Ica = Ica + ((1/(L_M)) * (Vca - Eca - R * Ica) * Ts);

                Ia = (Iab - Ica) / 3;
                Ib = (Ibc - Iab) / 3;
                Ic = (Ica - Ibc) / 3;
                Ic = 0;

            end

        end

    end

end


% Current Calculation

Iabc = [Ia , Ib, Ic];

% Torque Calculation

Te = (1/(Wrm))*((Ea * Ia) + (Eb * Ib) + (Ec * Ic)) ;


% Speed Calculation

if abs(Wrm) > 50
    Tl = Tl * sign(Wrm);
else
    Tl = 0;
end


Wrm = Wrm + ((1/J) * (Te - Tl - B*Wrm) * Ts );

% Output Assignment

Wr = Wrm;

We = ((Pole/2) * Wrm);

RPM = (60 /(2*pi) ) * Wrm;
```

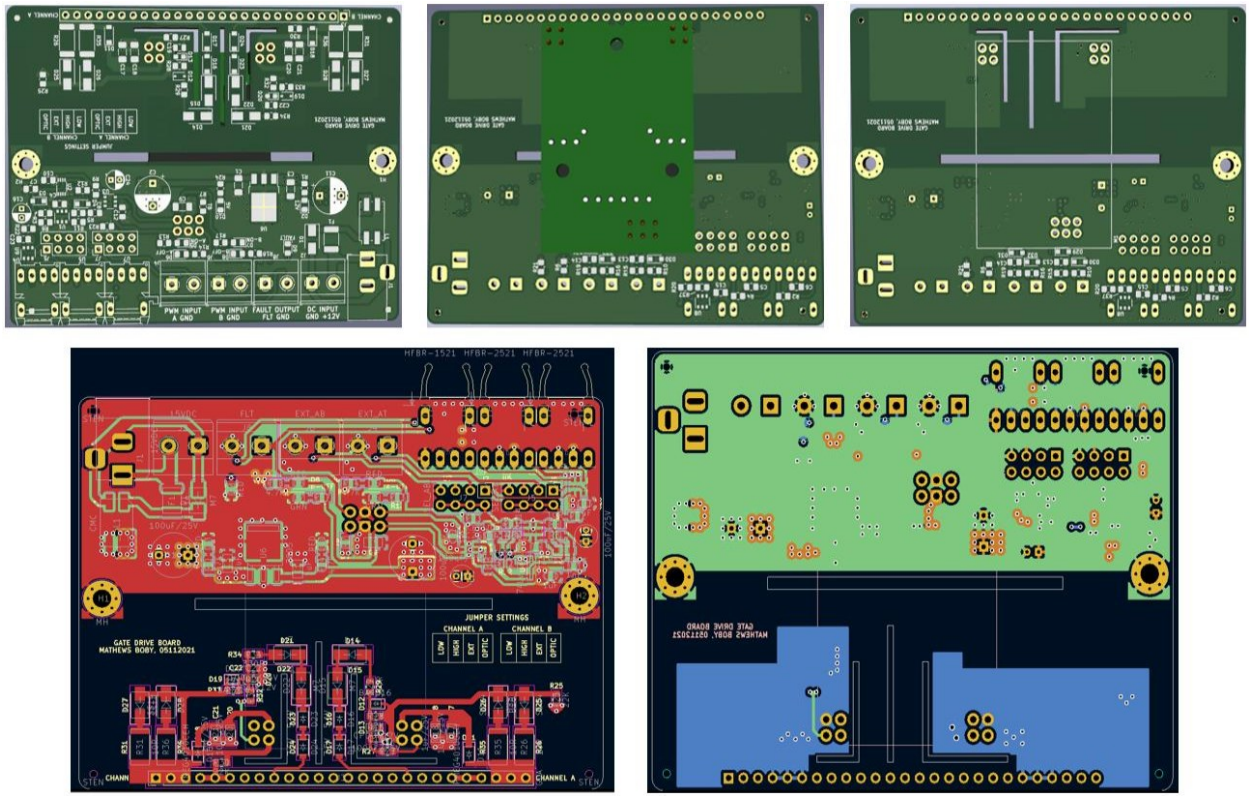
```
Eabc = [Ea, Eb, Ec];

Vpabc = [Vpa, Vpb, Vpc];
```

Appendix 2: Gate driver board

a) Schematic diagram

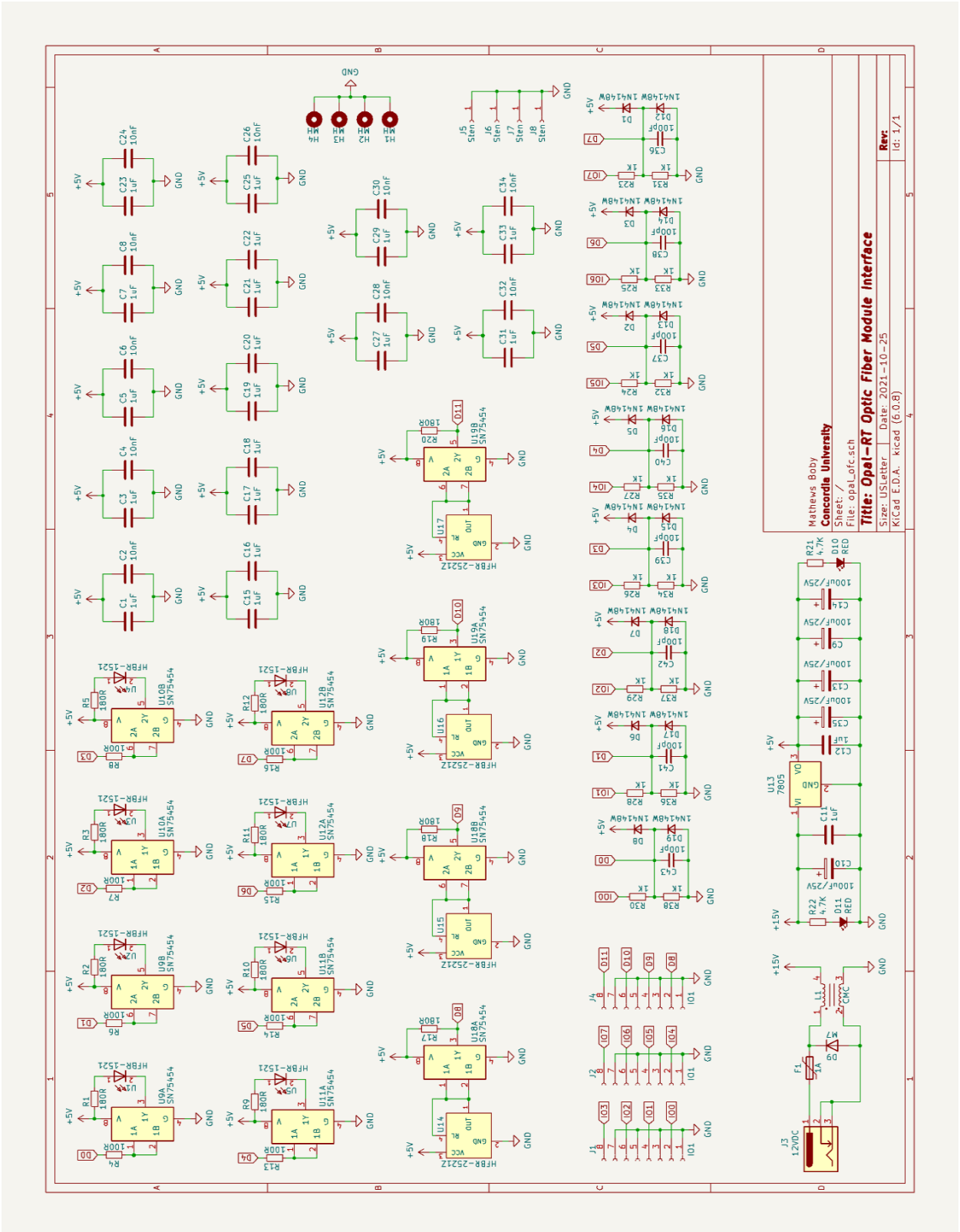


b) PCB layout

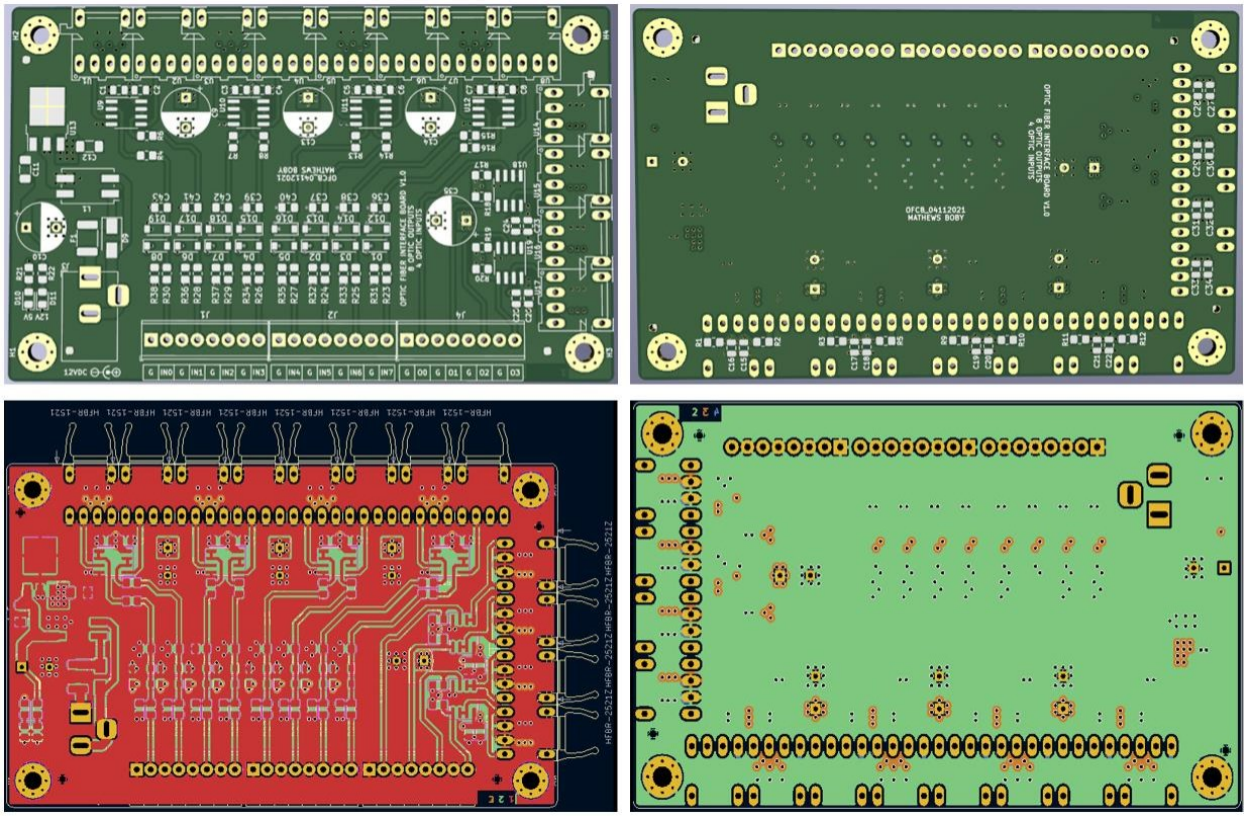

PWM INPUT A GND
PWM INPUT B GND
FAULT OUTPUT FLT GND
DC INPUT GND +12V
HFBR-1521 HFBR-2521 HFBR-2521
JUMPER SETTINGS
CHANNEL A
CHANNEL B
GATE DRIVE BOARD
MATHEWS BOBY. 05112021

Appendix 3: Interface board (signal conditioner board)

a) Schematic diagram

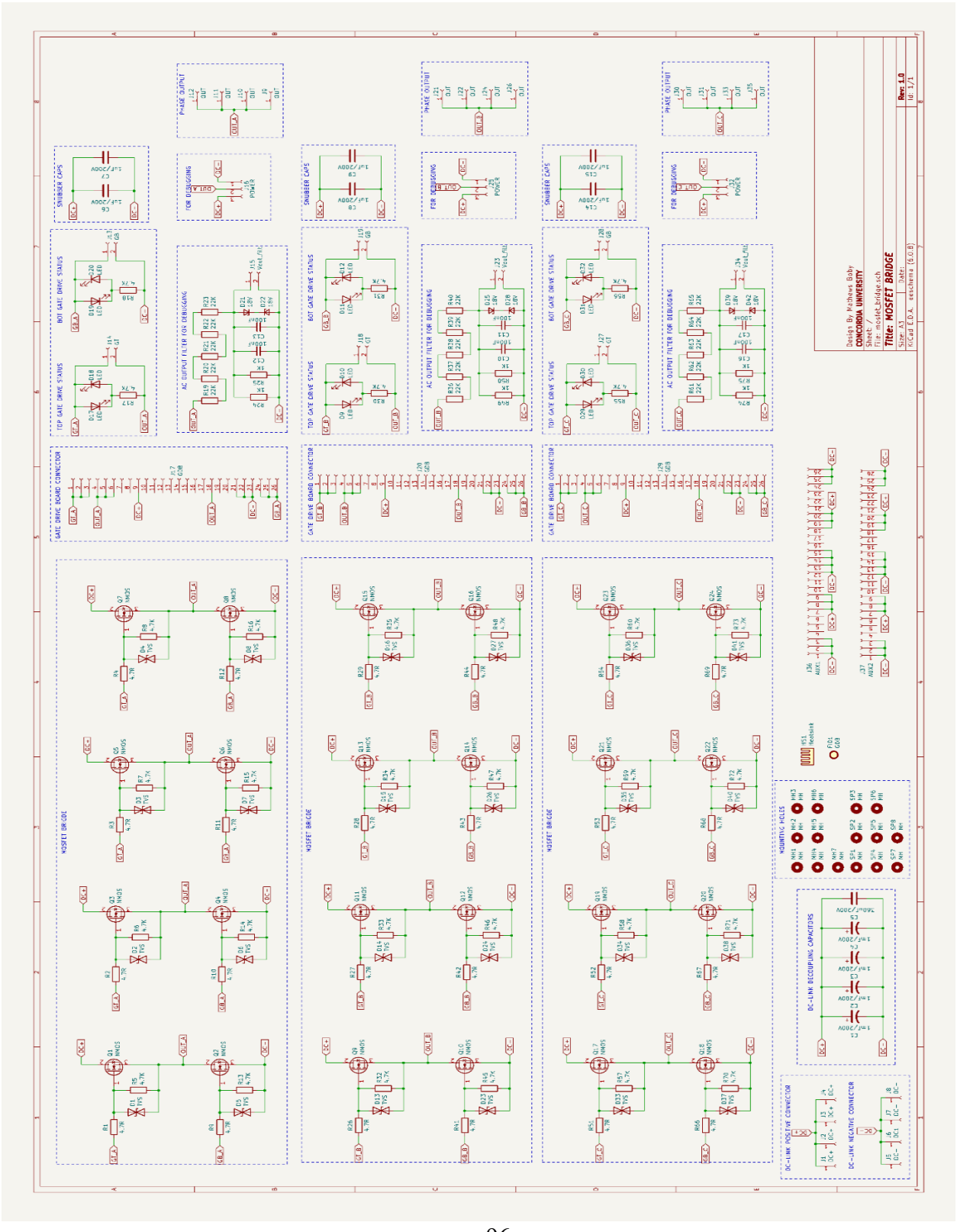


b) PCB layout

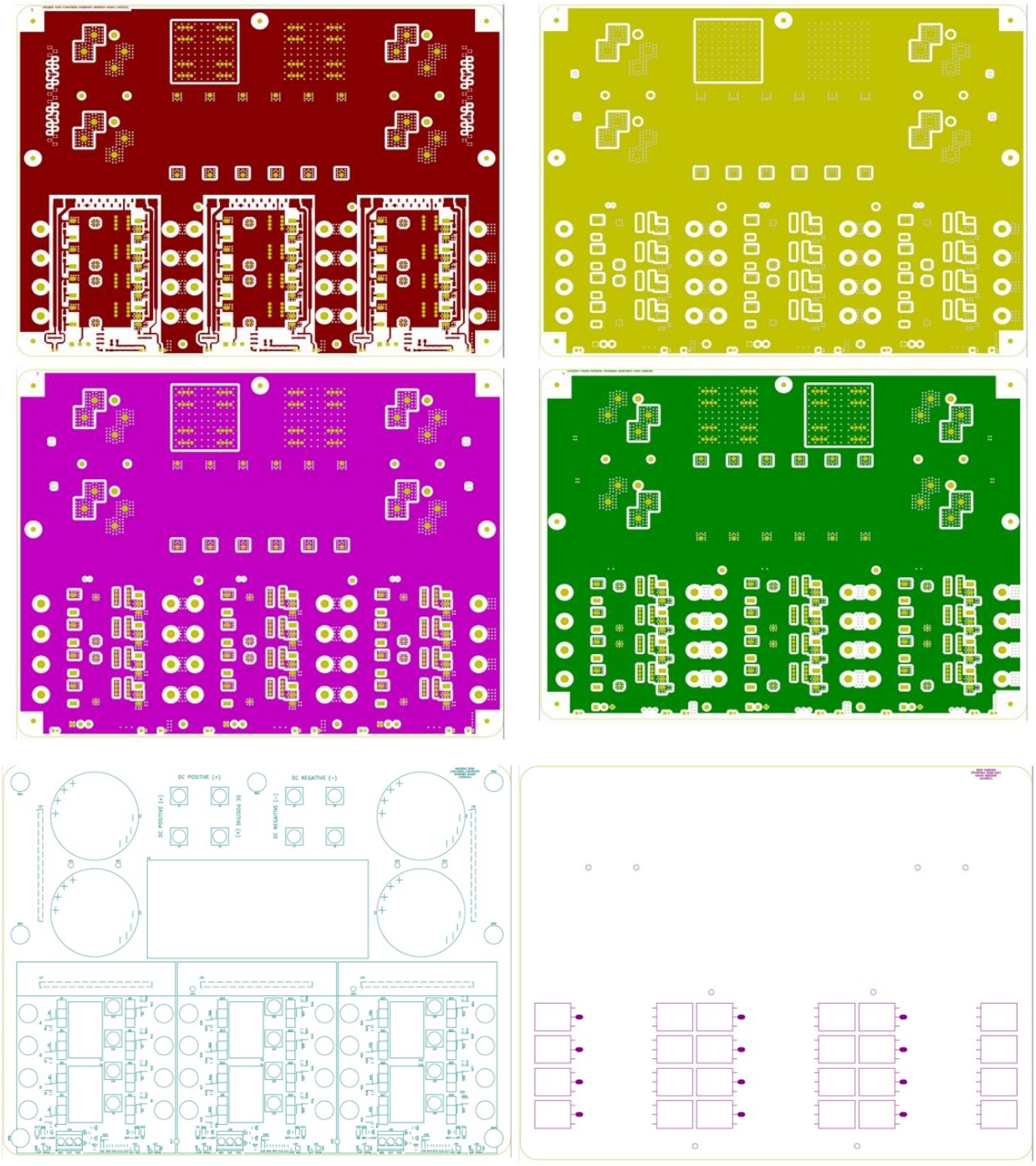


Appendix 4: Voltage source inverter board

a) Schematic diagram



b) PCB layout



Appendix 5: Inverter temperature test with passive load 100 A peak current

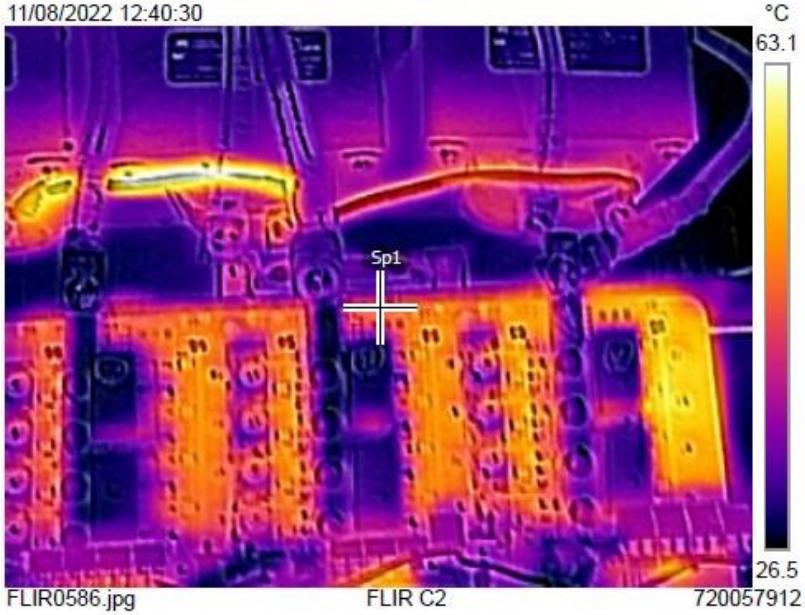
FLIR

Voltage_Source_Inverter

Measurements	
Sp1	36.7 °C

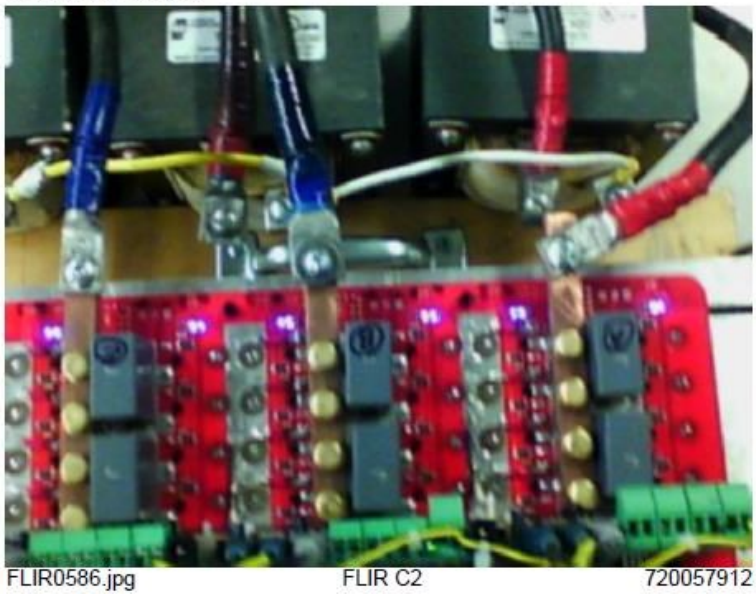
Parameters	
Emissivity	0.95
Refl. temp.	20 °C

11/08/2022 12:40:30



FLIR0586.jpg FLIR C2 720057912

11/08/2022 12:40:30



FLIR0586.jpg FLIR C2 720057912

PCB copper trace temperature test

Measurements	
Sp1	30.9 °C

Parameters	
Emissivity	0.95
Refl. temp.	20 °C

12/08/2022 16:09:49

FLIR0597.jpg FLIR C2 720057912

12/08/2022 16:09:49

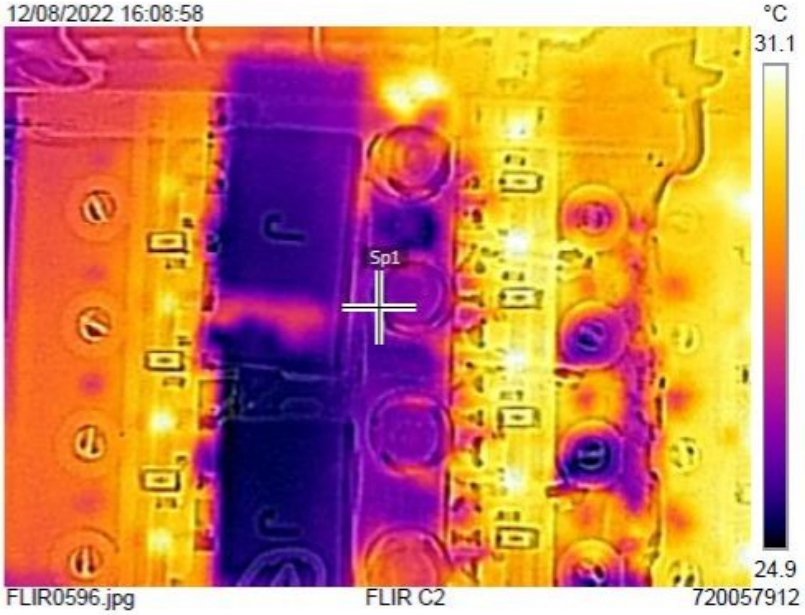
FLIR0597.jpg FLIR C2 720057912

Busbar temperature test

Measurements	
Sp1	26.3 °C

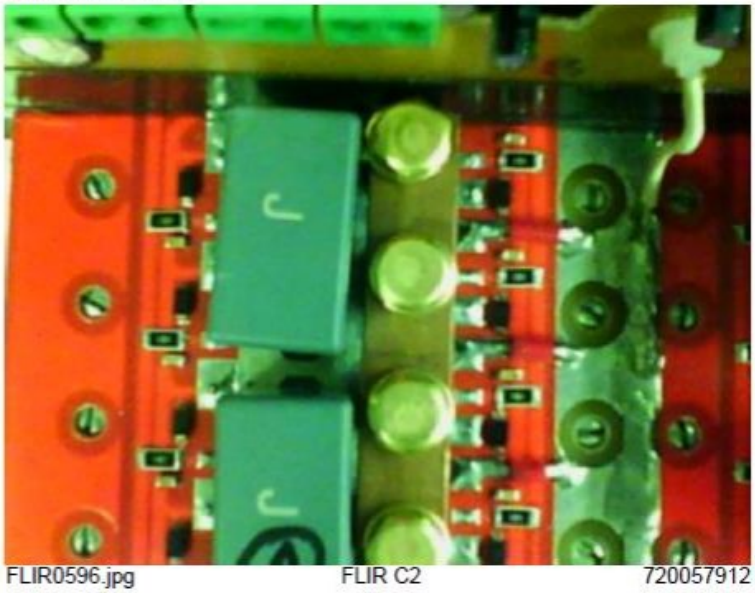
Parameters	
Emissivity	0.95
Refl. temp.	20 °C

12/08/2022 16:08:58



FLIR0596.jpg FLIR C2 720057912

12/08/2022 16:08:58



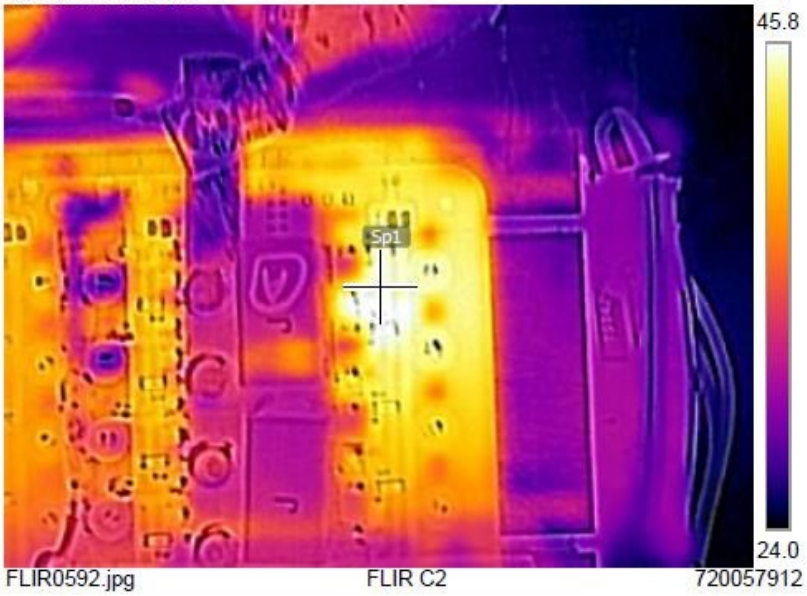
FLIR0596.jpg FLIR C2 720057912

Overheating of the MOSFET with poor thermal contact with the heatsink

Measurements	
Sp1	45.5 °C

Parameters	
Emissivity	0.95
Refl. temp.	20 °C

11/08/2022 12:44:35



FLIR0592.jpg FLIR C2 720057912

11/08/2022 12:44:35

FLIR0592.jpg FLIR C2 720057912

Overheating of the MOSFET with poor thermal conduct with the heatsink

Measurements	
Sp1	48.2 °C

Parameters	
Emissivity	0.95
Refl. temp.	20 °C

11/08/2022 12:40:58

FLIR0591.jpg FLIR C2 720057912

11/08/2022 12:40:58

FLIR0591.jpg FLIR C2 720057912