IMPLEMENTATION OF PERMANENT MAGNET SYNCHRONOUS MOTOR DRIVES SYSTEM SUITABLE FOR AIR-CONDITIONER APPLICATION

MUHAMMAD ABDUL RAHMAN

FACULTY OF ENGINEERING
UNIVERSITY OF MALAYA
KUALA LUMPUR

2013

IMPLEMENTATION OF PERMANENT MAGNET SYNCHRONOUS MOTOR DRIVES SYSTEM SUITABLE FOR AIR-CONDITIONER APPLICATION

MUHAMMAD ABDUL RAHMAN

DISSERTATION SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF ENGINEERING SCIENCE

FACULTY OF ENGINEERING
UNIVERSITY OF MALAYA
KUALA LUMPUR
2013

UNIVERSITI MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: **Muhammad Abdul Rahman**

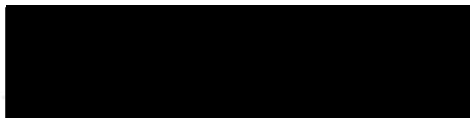Registration/Matric No: **KGA080016**

Name of Degree: **Master of Engineering Science**

Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"):
Implementation of Permanent Magnet Synchronous Motor Drives System Suitable for Air-Conditioner Application.

Field of Study: **Power Electronics and Machines Drives**
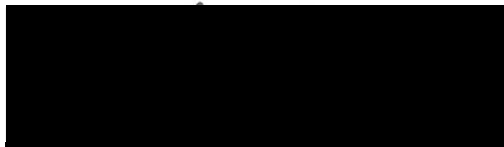
I do solemnly and sincerely declare that:

(1)   I am the sole author/writer of this Work;

(2)   This Work is original;

(3)   Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;

(4)   I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;

(5)   I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;

(6)   I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature   Date 21 / 06 / 2013

Subscribed and solemnly declared before,

Witness's Signature   Date 21/6/2013

Name:
Designation: Dr. Jeyraj a/l Selvaraj
                     Pensyarah Kanan
                     UMPEDAC, Universiti Malaya

ABSTRACT

Air-conditioners are being rapidly used by domestic consumers, industries and offices. Generally, an air-conditioner system consists of two motors in the outdoor unit. It consists of a fan and a compressor motor that always running continuously at full speed with on-off control technique, and wasteful electric power consumption especially during transient period. This work presents an implementation of a high-energy efficiency Permanent Magnet Synchronous Motor (PMSM) with variable speed drive system suitable for air-conditioner application. The motors run continuously with optimum speed operation depend on the load demand.

The motor drive system is implemented using two types of motors: surface-mounted and interior type PMSM suitable for fan and compressor motor, respectively. Sensored closed-loop Field Oriented Control (FOC) drive is used for Surface-mounted Permanent Magnet Synchronous Motor (SPMSM). A simplified Stator Flux Observer (SFO) is proposed for speed sensorless closed-loop FOC drive for an Interior Permanent Magnet Synchronous Motor (IPMSM). The proposed SFO eliminates the use of the extended rotor flux in order to estimate the rotor position.

A 32-bit fixed-point digital signal processor from Texas Instrument, TMS320F2812 board is used as motor controller. The motor drive system was verified and experimentally tested on a 920 W, 4 poles, SPMSM and a 600 W, 4 poles, IPMSM. The step-response of speed and load torque is applied to the motor drive system in order to observe the dynamic behaviour of system and the effectiveness of speed and current controller to sustain from sudden change of speed or load.

ABSTRAK

Penghawa dingin sedang pesat digunakan oleh pengguna industri, domestik dan pejabat. Secara umumnya, sistem penghawa dingin terdiri daripada dua motor di dalam unit bahagian luar. Unit bahagian luar terdiri daripada sebuah motor kipas dan sebuah motor pemampat yang sentiasa berjalan berterusan pada kelajuan penuh dengan teknik kawalan buka-tutup, dan penggunaan kuasa elektrik membazir terutama dalam tempoh fana. Kerja ini membentangkan pelaksanaan kecekapan tenaga-tinggi motor synchronous magnet kekal (*PMSM*) dengan sistem pemandu kelajuan boleh ubahsuai yang bersesuaian untuk penghawa dingin. Motor berjalan secara berterusan dengan operasi kelajuan yang optimum bergantung kepada permintaan beban.

Sistem pemandu motor dilaksanakan menggunakan dua jenis motor: *PMSM* jenis permukaan dan *PMSM* jenis dalaman sesuai untuk motor kipas dan pemampat. Pemandu gelung-tertutup dengan sensor dan kawalan berorientasikan bidang (*FOC*) digunakan untuk *PMSM* jenis permukaan (*SPMSM*). Sebuah pemerhati pemegun fluks yang telah disederhanakan (*SFO*) dicadangkan untuk pemandu kelajuan gelung-tertutup kawalan *FOC* tanpa sensor untuk *PMSM* jenis dalaman (*IPMSM*). *SFO* yang dicadangankan menghapuskan penggunaan fluks-pemutar lanjutan untuk menganggarkan kedudukan pemutar.

Sebuah pemproses isyarat digital 32-bit daripada *Texas Instrument*, TMS320F2812 digunakan sebagai pengawal motor. Sistem pemacu motor telah disahkan dan diuji dengan uji kaji pada motor *SPMSM* dengan kuasa elektrik 920 W, 4 kutub dan motor *IPMSM* dengan kuasa elektrik 600 W, 4 kutub. Langkah tindak balas kelajuan dan tork beban digunakan untuk sistem pemacu motor untuk memerhatikan tingkah laku dinamik sistem dan keberkesanan pengawal kelajuan dan arus untuk mengekalkan daripada perubahan kelajuan atau beban secara tiba-tiba.

ACKNOWLEDGEMENTS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

μ_r	Relative permeability of material
B	Viscous coefficient of the motor
$e_{d,q}$	dq-axis cross-coupling term
f_s	Stator frequency
G	Gain
$i_{a,b}$	Stator currents
$i_{d,q}$	dq-axis stator currents
$i_s = i_{\alpha,\beta}$	$\alpha\beta$-axis stator currents
J	Total moment inertia of the motor
$L_{d,q}$	dq-axis inductances
M	Modulation index
P	Number of poles
p	Number of poles pair
R_s	Stator resistance
s	Derivation (d/dt)
$S_{a,b,c}$	Switching signal of SVM
T_e	Electromagnetic torque
T_L	Load torque

T_m	Permanent magnet torque
T_r	Reluctance torque
T_s	Time sampling
$v_{d,q}$	dq-axis stator voltages
v_{dc}	DC-link voltage
$v_s = v_{\alpha,\beta}$	$\alpha\beta$-axis stator voltages
θ_e*	Estimated electrical theta
θ_m*	Estimated mechanical theta
$\lambda_{d,q}$	dq-axis stator flux linkages
$\lambda_{d,q}$	dq-axis stator flux linkages
λ_{PM}	Permanent magnet flux linkage
$\lambda_s* = \lambda_{\alpha,\beta}*$	Estimated $\alpha\beta$-axis stator flux linkages
$\lambda_{sc}* = \lambda_{sc\alpha,\beta}*$	Compensation $\alpha\beta$-axis flux linkages
ω_e	Electrical angular speed
$\omega_{m\ ref}$	Mechanical angular speed reference
ω_m*	Estimated mechanical angular speed

LIST OF ABBREVIATIONS

AC   Alternating Current

ADC   Analog Digital Converter

CT   Current Transformer

CTA   Constant Torque Angle

DAC   Digital Analog Converter

DC   Direct Current

DMC   Digital Motor Control

DSP   Digital Signal Processor

DTC   Direct Torque Control

emf   electromotive force

EMI   Electromagnetic Interference

FOC   Field Oriented Control

HVAC   Heating Ventilation and Air Conditioning

IGBT   Insulated Gate Bipolar Transistor

iPMSM   inset Permanent Magnet Synchronous Motor

IPMSM   Interior Permanent Magnet Synchronous Motor

MTPA   Maximum Torque Per Ampere

PI   Proportional Integrator

PC   Personal Computer

PMBLDCM   Permanent Magnet Brushless Direct Current Motor

PSU   Power Supply Unit

QEP   Quadrature Encoder Pulse

RFI   Radio Frequency Interference

SCC   Signal Conditioning Circuit

SFO   Stator Flux Observer

SPMSM   Surface - mounted Permanent Magnet Synchronous Motor

SVM   Space Vector Modulator

SVPWM   Space Vector Pulse Width Modulation

THIPWM   Third Harmonic Injection Pulse Width Modulation

VRC   Voltage Reconstruction Calculator

VSD   Variable Speed Drives

VSI   Voltage Source Inverter

CHAPTER 1
INTRODUCTION

1.1 Background

Air-conditioners and evaporator-coolers, used daily by domestic consumers, industries and offices, have become one of the highest power consuming electric appliances. Continuously running air-conditioners, evaporator-coolers, and also other motor-driven applications contribute to about 50% of total power consumption by domestic and industrial sectors (Slemon, 1993). Conventional air-conditioners and evaporator-coolers use Induction Motor (IM) with on/off control technique. It has some drawbacks such as huge fluctuation of temperature (poor temperature regulation), noisy compressor, and compressor motor's high consumption of energy caused by high transient starting current (six times of nominal current for IMs) and poor efficiency (Yu, 2005). Moreover, because the motor always runs at the full speed and the power output of the motor is proportional to the mechanical rotor speed, the power consumption in on/off control technique is still high even in light load condition.

In order to save the energy, power consumption of IMs can be reduced through a Variable Speed Drive (VSD) motor drive system. By using VSD, the speed of motor can be adjusted depending on desired load demand. Implementation of VSD in a compressor motor increases the system efficiency and eliminates periodic losses during the start-up (Singh, Singh, & Kumar, 2003). Moreover, it reduces both the peak and average energy consumption.

VSD can be categorized into two types: scalar control and vector control. The scalar control has poor dynamic performance. The concept of vector control in Alternating Current (AC) motor drives was introduced by Blaschke (1972). In order to

achieve high dynamic performance, VSD drive system with vector control technique is the most recommended.

Recently, with the increased efficiency in motors emerging as an alternative answer for energy saving, IMs are being replaced by Permanent Magnet (PM) brushless motors. The reason the IM's efficiency is lower than brushless PM motors is due to the losses of rotor. During operation, the stator winding of IM induces the squirrel cage bar on the rotor. As a result, the current flows to the bar, magnetizes the rotor and the rotor flux increases gradually to a certain value at rated voltage and vice versa. This current flow circulates in the rotor bar and is dissipated as heat energy. In contrast to the IM, brushless PM motor has constant rotor flux provided by the permanent magnet mounted at the rotor. It has no rotor losses since no current flows in the rotor and this makes brushless PM motors preferable to IM for improvement of the system efficiency.

Although brushless PM motors are more expensive in comparison with IMs, they offer several advantages (Bose, 2002; Fitzgerald A. E., 2003; Vas, 1998) that can be summarized as follows:

- High-efficiency (no rotor losses)

- High-torque

- Higher power/weight ratio

- Robust construction and low-cost maintenance (no commutator or slip ring)

- Safe operation in flammable area (sparkless)

- Lower inertia

- Good dynamic performance

Brushless PM motors have several variants with different rotor construction and different characteristic. They are used widely in many high-performance servo drive applications. Furthermore, they have served energy-saving purposes in industrial

applications; hybrid electric vehicles, elevator and home appliances such as air-conditioners, washing machines, and fans (J. Soleimani, 2011; Melfi, Rogers, Evon, & Martin, 2008; Murray, 2007). Significantly, the brushless PM motor increases the power density of the motor and reduces the size, weight, heat output and energy consumptions by up to one half as compared to induction motor (Pillay & Freere, 1989).

The VSD control technique for brushless PM motors has been investigated by Jahns, Kliman, & Neumann (1986) and Jungreis & Kelley (1995). In a VSD with vector control algorithm, rotor position and speed sensor are required for feedback to the system. The rotor position information is needed to perform coordinate transformations during vector control process. Several types of speed sensor such as resolver, incremental, or absolute encoder can be used to provide rotor position information and speed measurement. In fact, the motor drive system with speed sensor as feedback offers an accurate speed control. However, the use of speed sensor has some drawbacks. It increases the cost of the motor drive (Matsui, 1996). Moreover, the speed sensor cannot be mounted on the motor shaft for certain applications such as air-conditioner compressors and other motor implementations operating in corrosive environments. To solve this problem, motor drives are moving towards speed or position sensorless drives systems. This concept was introduced by Frus (1976). This technique used current waveform analysis. The sensorless drive system offers some advantages such as no mechanical connection on rotor and eliminating the cost of the speed sensor (Acarnley & Watson, 2006). For sensorless control, a stator flux observer (SFO) is needed to estimate the stator flux and it is used for rotor position estimation.

In this work, two configurations: sensor and sensorless vector control are implemented in two motor drive systems, separately. Both systems are operated at constant torque angle control mode. A simplified SFO is proposed for the sensorless drives system where the use of the extended rotor flux is eliminated.

1.2 Research Objective

- To develop a Permanent Magnet Synchronous Motor (PMSM)-driven inverter drives system suitable for air-conditioner system.

- To develop sensor and sensorless vector control algorithm for a variable-speed PMSM drive system.

- To observe the performance of the PMSM drive system.

1.3 Methodology of Research

1. **Design of PMSM inverter drives system.**

The control scheme used vector control algorithm for controlling the motor over a wide range of speed because it fulfils the design criteria of high-performance drive system.

2. **Software development.**

High-speed Digital Signal Processor (DSP) controller board was used to implement the algorithm and generate switching control of the inverter. The DSP board is required because the switching control algorithm needs intensive mathematical calculation to determine the appropriate switching. Furthermore, the DSP board has many input-output ports and analog to digital converters to provide control signal, feedback signal and other sensors or transducer signals.

3. **Hardware development.**

Basically, there are two main blocks for the inverter drive system: high-voltage side block and low-voltage side block. The high-voltage side block consists of a single-phase rectifier and a three-phase inverter, while the low-voltage side block consists of DSP controller, gate drive circuit, current sensor, voltage sensor and speed sensor. Six-

Insulated Bipolar Gate Transistors (IGBTs) were used in three-phase inverter as a switch to convert the Direct Current (DC) voltage from rectifier into three phase AC voltage.

The control signal was provided by DSP across a gate drive circuit for each IGBT. The use of gate drive circuit is important to increase small voltage signal from DSP to the particular voltage of the IGBT, usually around 15 volts to turn-on the IGBT. Moreover, the gate drive circuit provides a protection and insulation between high-voltage side block (three-phase inverter) and low-voltage side block (DSP controller).

4. Software and hardware integration.

The proposed algorithm was programmed into DSP. It produced a control signal for the IGBT according to the algorithm programmed in it. There were more than two control loops for the motor drive. The feedback signal was connected to Analog - Digital Converter (ADC) which was embedded into the DSP unit itself. The programmed was written in the computer and uploaded into the DSP board.

5. Testing and measurement.

The actual operating condition and real-time response can only be obtained through experiments, for which testing and measurement must be carried out. The measurement requires very high speed equipment that can capture the actual response during the transient period as well as isolated channel inputs, which enable every point on the circuit to be tested to confirm that they are functioning properly.

1.4 Dissertation Organization

The dissertation was organized into five chapters. A brief introduction of the research, the objectives and methodology of research are presented in Chapter 1. In Chapter 2, literature reviews, motor analysis, modelling, control scheme of the PMSM and its operation are discussed here. Then, the configurations of the motor drive system and experiment setup diagram are described in Chapter 3. The next chapter presents the experimental results and discussion. Finally, concluding remarks and suggestion for future work are presented in Chapter 5. Additional important information is given in the appendices.

CHAPTER 2

MOTOR ANALYSIS, MODELLING AND CONTROL

2.1 Motor Construction

Generally, according to the back-emf shape of motor, three-phase brushless PM motor can be divided into two types, as shown in Figure 2.1: trapezoidal type or "Permanent Magnet Brushless Direct Current Motor" (PMBLDCM) and sinusoidal type or "Permanent Magnet Synchronous Motor" (PMSM). To generate constant torque, the PMBLDCM requires rectangular stator phase current input for excitation and produces trapezoidal back-emf. Otherwise, the PMSM requires sinusoidal stator phase current input for excitation and produces sinusoidal back-emf. The PMSM also can be divided into two types related to the permanent magnets' arrangement in the rotor: non-salient Surface-Mounted Permanent Magnet Synchronous Motor (SPMSM), salient Surface-Mounted Permanent Magnet Synchronous Motor called inset Permanent Magnet Synchronous Motor (iPMSM) and Interior Permanent Magnet Synchronous Motor (IPMSM).
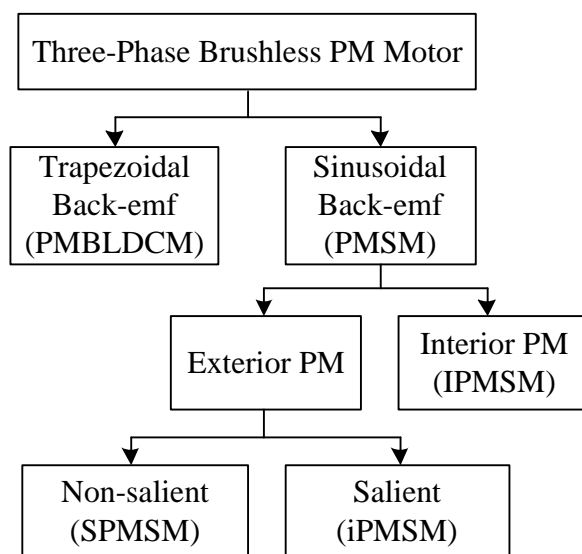
Figure 2.1: Brushless PM motor classification

Figure 2.2 shows cross section of different types of brushless PM motors. PMBLDCM has similarity with SPMSM. The rotor surface is surrounded by the permanent magnets but it has a different stator winding arrangement and permanent magnet shape (Pillay & Krishnan, 1991). This magnet is attached on the surface of rotor and makes an air-gap between the magnets. In iPMSM, the rotor construction is quite similar to the SPMSM but it has rotor saliency. It is created by the teeth between the permanent magnet in iPMSM's rotor. While in IPMSM, permanent magnets are implanted inside the rotor.

Iron

(a)   (b)   (c)   (d)
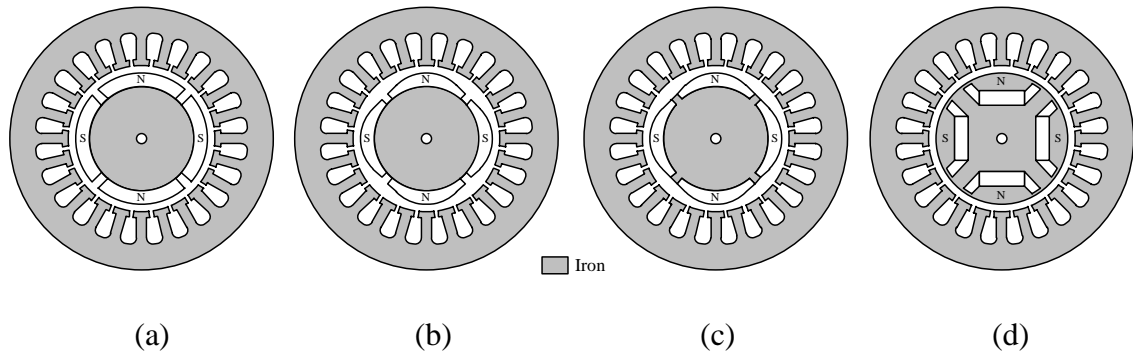
Figure 2.2: Cross section of brushless PM motors
(a)   PMBLDCM (b) SPMSM (c) iPMSM (d) IPMSM

The same VSD control scheme can be used for any rotor type in PMSM group. However, the electromagnetic torque for each PMSM family does not produce the same value. In SPMSM, the electromagnetic torque is only produced by permanent magnet torque, but in IPMSM and iPMSM, the electromagnetic torque is not only produced by permanent magnet torque but also by additional torque called reluctance torque. It occurs due to the different value of direct-inductance (L_d) and quadrature-inductance (L_q) as a function of rotor position, called saliency ratio. The saliency ratio can be expressed as:

$$SR = \frac{L_q}{L_d}$$
(2.1)

The saliency ratio for SPMSM is almost equal to one ($SR \approx 1$). It indicates the d-axis inductance is equal to the q-axis inductance. But, the saliency is larger than one ($SR > 1$) for iPMSM and IPMSM which means that the d-axis inductance is smaller than the q-axis inductance. The saliency ratio of IPMSM is higher than iPMSM. The saliency ratio differences between SPMSM and IPMSM can be illustrated from the Figure 2.3 (Kang, 2010). The magnetic flux path passes along d-axis and q-axis through the different materials with different permeability values, where relative permeability of the air is almost equal to relative permeability of the permanent magnet ($\mu_r = 1$ for the air; $\mu_r = 1.05$ for the permanent magnet) and relative permeability of the iron is the largest value ($\mu_r = 500$).

(a)   (b)
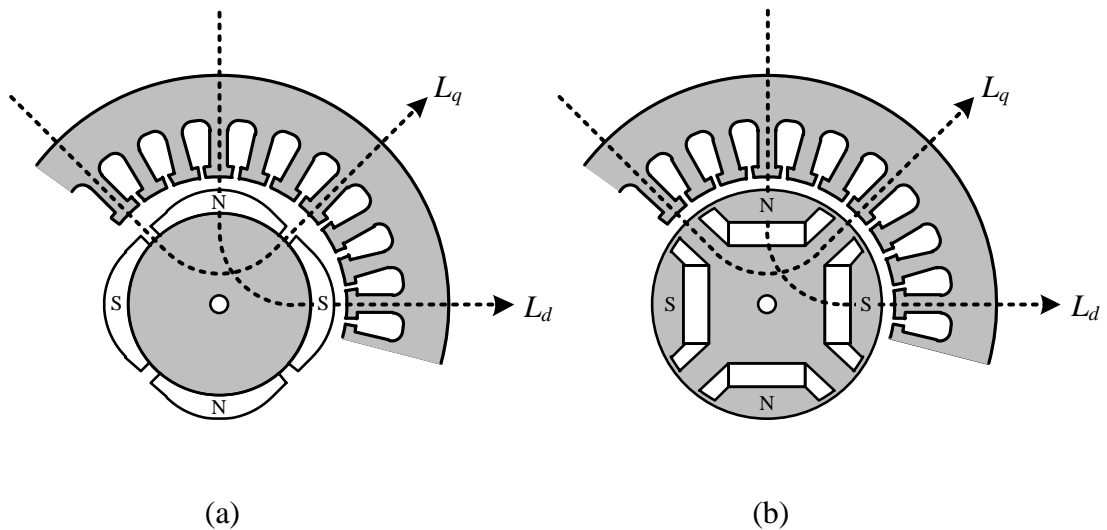
Figure 2.3: Inductance path of Motor
(a) SPMSM (b) IPMSM

The electromagnetic torque as function of torque angle can be seen in Figure 2.4. For iPMSM and IPMSM, the electromagnetic torque is represented in curve 1 which is the sum of curves 2 and 3. Curve 2 shows the electromagnetic torque for SPMSM. In order to achieve maximum torque, several control modes can be chosen, all of which are discussed later in Chapter 2, section 2.4. Before selecting the appropriate mode of control, the PMSM needs to be modelled in mathematic equations as is discussed in the next section.
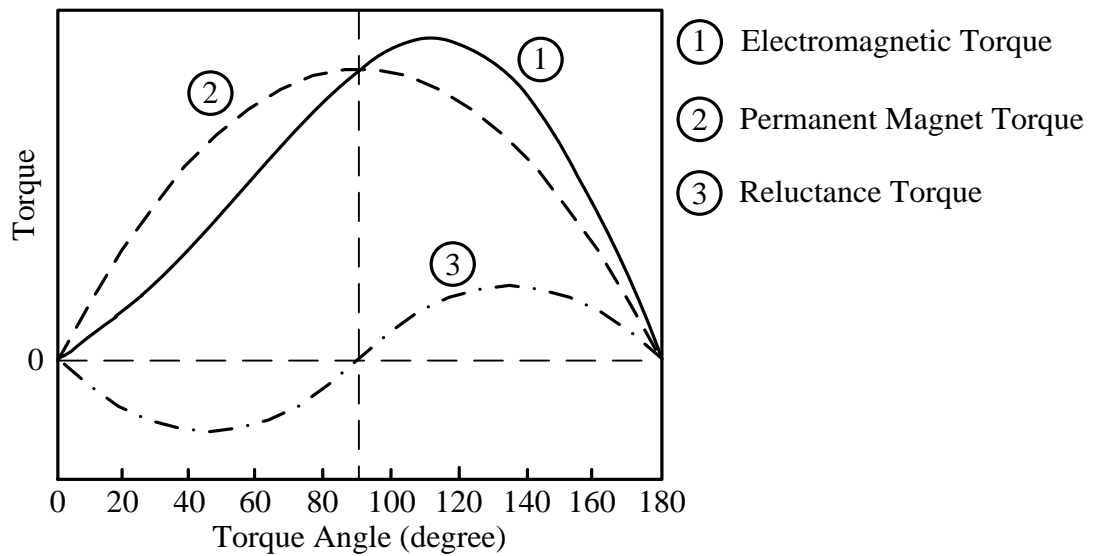
Figure 2.4: Torque curve of PMSM

2.2 Mathematical Model of PMSM

Figure 2.5 illustrates the stationary and rotating reference frame of three-phase brushless PM motor. The purpose of the PMSM modelling is to simplify the three-phase motor's mathematical equation from stationary reference frame (abc-axis) into rotating reference frame (dq-axis). The PMSM modelling can be derived with some assumptions below (Pillay & Krishnan, 1988):

- The back-emf shape is sinusoidal.

- The core loses and eddy current are neglected.

- The three phase supply is in balanced condition.

- Rotor flux is constant.
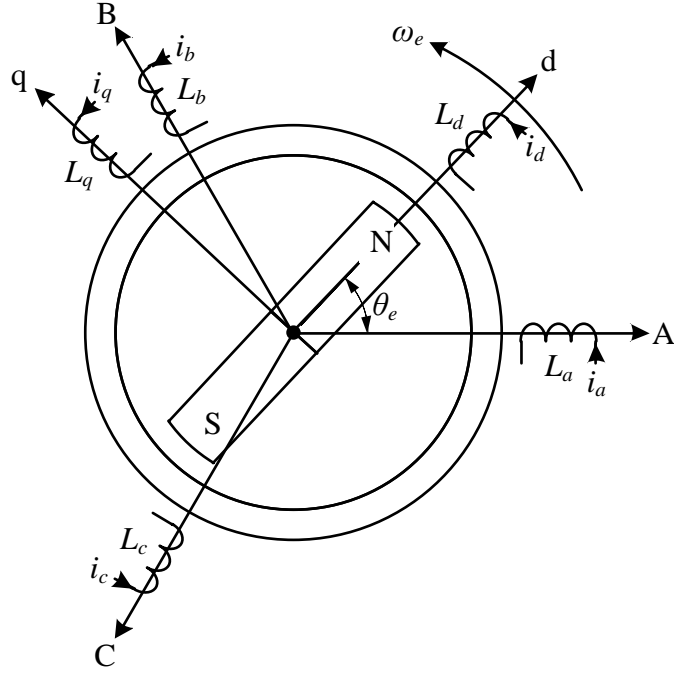
Figure 2.5: Reference frame illustration of three-phase brushless PM motor

The rotating reference frame rotates at the electrical angular speed and the electrical angle is an integration of it, as is expressed below:

$$\omega_e = p\,\omega_m$$
$$\theta_e = \int \omega_e dt \qquad\qquad (2.2)$$

Where, the mechanical angular synchronous speed is formulated as,

$$\omega_m = \frac{4\pi\,f_s}{P} \qquad\qquad (2.3)$$

Where, f_s is stator frequency and P is number of poles.

The three-phase system can be expressed in space vector representation to simplify the system analysis. By assuming the system is in balanced condition, the space vector is given in equation (2.4) and illustrated in Figure 2.6.

$$x_S = \frac{2}{3}\left(x_a + ax_b + a^2x_c\right) \tag{2.4}$$

Where x represents currents, voltages or flux linkages, a is unit vector $a = e^{j2\pi/3}$
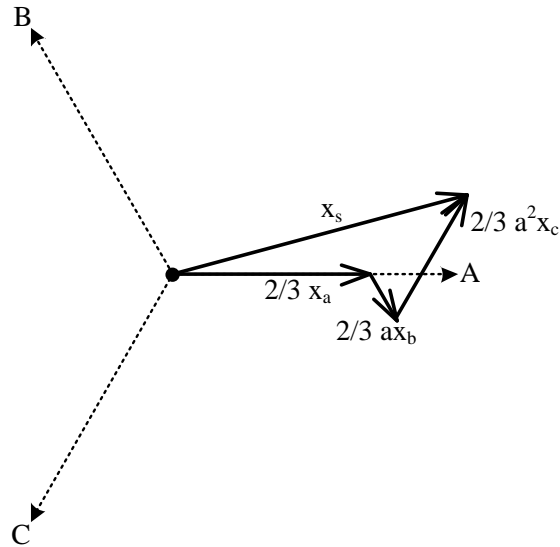
Figure 2.6: Space vector representation

A. Voltage Equations

Figure 2.7 shows equivalent circuit of PMSM in stationary reference frame (three-phase system).
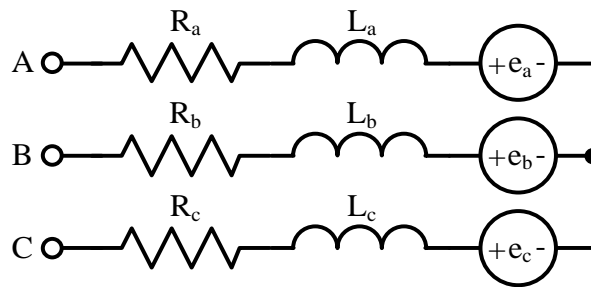
Figure 2.7: Three-phase equivalent circuit of PMSM

The voltage equation of the PMSM is defined as,

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = R_S \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \lambda_a \\ \lambda_b \\ \lambda_c \end{bmatrix} \tag{2.5}$$

And the stator flux is:

$$\begin{bmatrix} \lambda_a \\ \lambda_b \\ \lambda_c \end{bmatrix} = \begin{bmatrix} L_a & M_{ab} & M_{ac} \\ M_{ba} & L_b & M_{ab} \\ M_{ca} & M_{cb} & L_c \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \lambda_{PM} \begin{bmatrix} \cos(\theta_e) \\ \cos\left(\theta_e - \dfrac{2\pi}{3}\right) \\ \cos\left(\theta_e - \dfrac{4\pi}{3}\right) \end{bmatrix} \tag{2.6}$$

The voltage equation of PMSM in dq-axis can be expressed as,

$$V_d = R_S i_d + s\lambda_d - \omega_e \lambda_q$$
$$V_q = R_S i_q + s\lambda_q + \omega_e \lambda_d \tag{2.7}$$

Where stators flux linkages is:

$$\lambda_d = \lambda_{PM} + L_d\, i_d$$
$$\lambda_q = L_q\, i_q \tag{2.8}$$

The last term in equation (2.7) is the cross-coupling component where it is considered as voltage disturbance. It makes the motor become nonlinear since d and q-axis current appear in both voltage equations. The voltage disturbance components for d-axis and q-axis are expressed in the equation below:

$$e_d = -\omega_e \lambda_q$$
$$e_q = \omega_e \lambda_d \tag{2.9}$$

Figure 2.8 shows the entire dq-axis model of PMSM where core losses are not taken into account.

(a)　　　　　　　　　　　　　　　　(b)
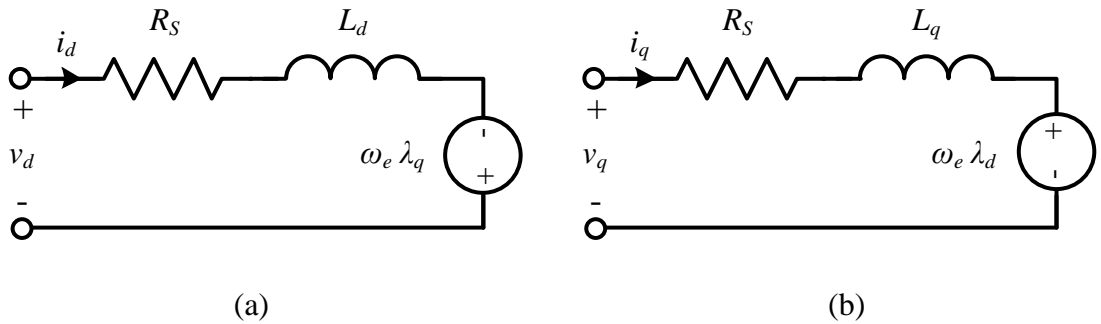
Figure 2.8: Equivalent circuit of PMSM in dq-axis
(a) d-axis (b) q-axis

B. Torque and Power Equation

Electromagnetic torque production is a vector product of stator flux linkage and stator current as given in equation:

$$T_e = \frac{3}{2}\frac{P}{2}\left(\lambda_S \times i_S\right)$$
$$T_e = \frac{3}{2}\frac{P}{2}\left(\lambda_{PM}\ i_q - \left(L_q - L_d\right)i_q\ i_d\right) \tag{2.10}$$

Equation (2.10) contains two components of torque: the permanent magnet torque and the reluctance torque that can be formulated in the following equations below, respectively:

$$T_m = \frac{3}{2}\frac{P}{2}\ \lambda_{PM}\ i_q \tag{2.11}$$

$$T_r = \frac{3}{2}\frac{P}{2}\left(L_q - L_d\right)i_q\ i_d \tag{2.12}$$

Electromagnetic torque also can be expressed as a kinetic motion equation, given by:

$$T_e = T_L + \frac{J}{p}s\omega_m + \frac{B}{p}\omega_m \tag{2.13}$$

Where J is sum of moment of inertia of rotor (J_m) and load (J_L), B is total viscous coefficient of rotor (B_m) and load (B_L):

$$J = J_m + J_L$$
$$B = B_m + B_L \tag{2.14}$$

The mechanical power output is proportional to mechanical angular speed as given by:

$$P_{mech} = T_e\ \omega_m \tag{2.15}$$

Efficiency is obtained from:

$$\eta = \frac{P_{mec}}{P_{in}} \times 100\% \tag{2.16}$$

14

2.3 Control Schemes of PMSM

In variable speed drive of separately excited DC motors, it is easier to control the flux and torque component independently. The torque angle in a DC motor is always maintained at 90 degree and the maximum electromagnetic torque is produced as a result. The variable-speed drive of AC motors has several control schemes. Generally, it can be divided into two main categories: scalar control and vector control. The scalar control scheme is simple to implement. The most popular scalar control in industry is constant Voltage/Frequency control. This is the simplest method, which does not provide a high-dynamic performance as can be obtained in DC motors.

The idea of vector control for AC motors was introduced by Blaschke (1972) in order to imitate the DC motor behaviour, where the flux and the torque component of the AC motors are decoupled so as to easily be controlled separately. The vector control allows not only control of the voltage amplitude and frequency, like in the scalar control schemes, but also it controls the instantaneous position of the voltage, current and flux vectors. As a result, the dynamic performance of the AC motor drive can be improved significantly. To implement the vector control scheme, the AC motor needs to be modelled as has been discussed in section 2.2. The vector control itself is divided into two main groups: Field Oriented Control (FOC) and Direct Torque Control (DTC).

PMSM employed FOC also known as current control to enable the motor to attain quick response similar to DC motor. The system usually employs position information for reference frame coordinate transformation of the current components. The FOC technique decouples the two components of stator current: one providing the air-gap flux and the other producing the torque. It provides independent control of flux and torque, and the control characteristic is linearized. The stator currents are converted to a fictitious synchronously rotating reference frame aligned with the flux vector and are transformed back to the stator frame before feeding back to the motor. The two

components are q-axis analogous to armature current, and d-axis analogous to the field current of separately excited DC motor. The rotor flux linkage vector is aligned along the d-axis of the reference frame.

Implementation of FOC algorithm is only suitable to run a PMSM with sinusoidal back-emf shape. For PMBLDC, this algorithm cannot be applied effectively (Microchip, 2009). It is because PMBLDC cannot produce constant torque when sinusoidal input current is applied. PMBLDC only produces constant torque for rectangular current injection. Therefore, before FOC scheme is implemented, it is important to verify the back-emf shape of motor, whether it has sinusoidal or trapezoidal back-emf.
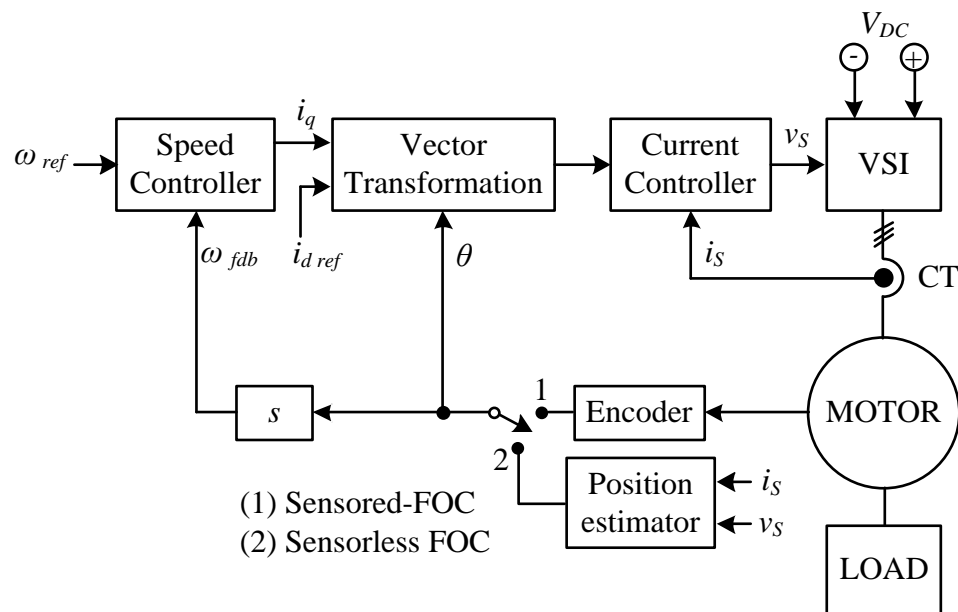
Figure 2.9: Single line diagram of closed-loop FOC

Basically, closed-loop FOC consists of two main loops: inner loop and outer loop as shown in Figure 2.9. Inner loop is the current controller while outer loop is the speed controller. Since inner loop has mathematical calculation for reference transformation, a faster time sampling is required. The inner loops' time sampling

should be faster than the outer loop. The controllers are compared with the reference value to take an action for error elimination in steady-state condition.
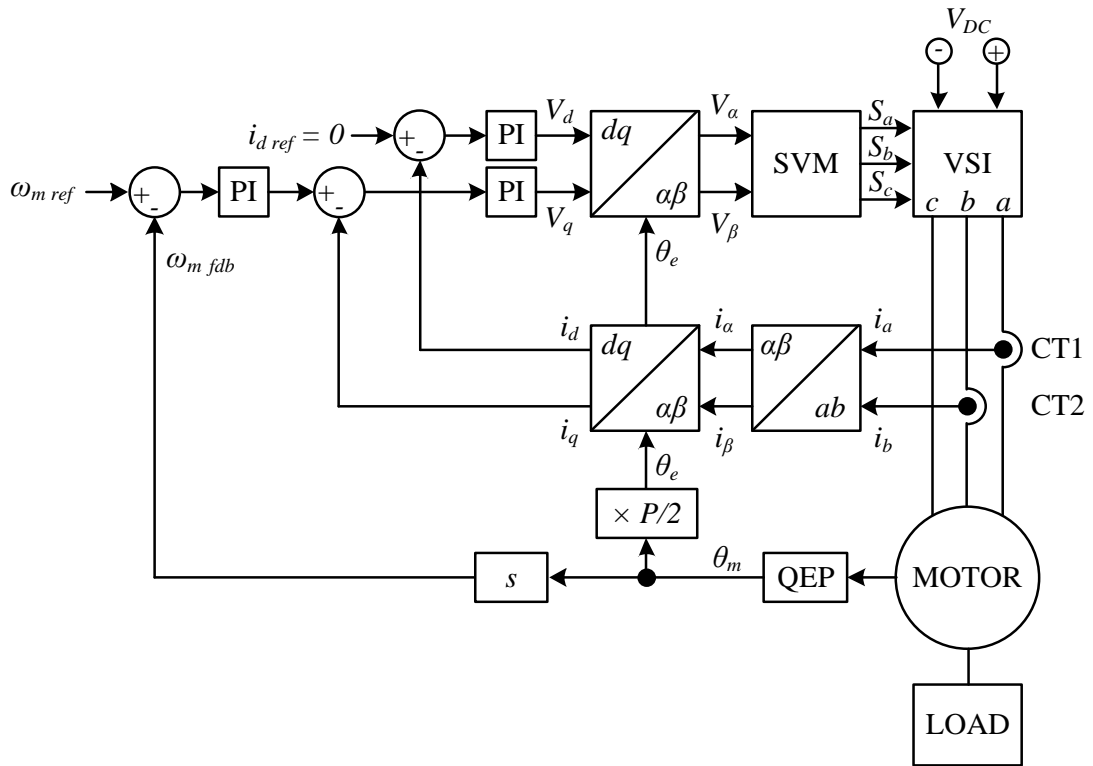
Figure 2.10: Sensored-FOC block diagram
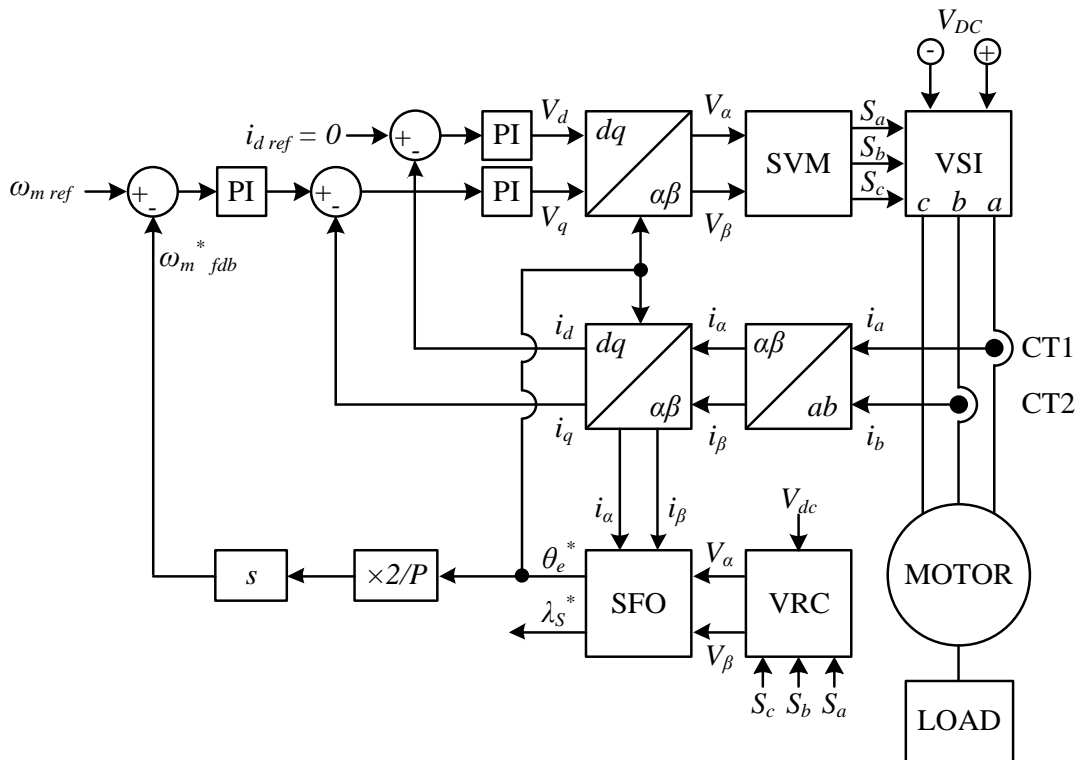
Figure 2.11: Sensorless-FOC block diagram

In this research, the SPMSM uses sensored-FOC where the rotor position information is provided by a speed sensor as described in Figure 2.10; while IPMSM uses sensorless-FOC algorithm and the rotor position is estimated by an observer as shown in Figure 2.11. Both systems used constant torque angle control mode due to its simplicity. It will be explained in section 2.4.

2.3.1 Speed and Current Controller

The robustness of closed-loop FOC drives system depends on the effectiveness of the speed and current controller to keep on tracking any changes of reference value and take an action to follow the desired reference value faster. The speed and current controller has been investigated by Brod & Novotny (1985), Jouve, Rognon, & Roye (1990), Kazmierkowski & Malesani (1998) and Uddin, Radwan, George, & Rahman (2000). Generally, there are two main types of current controllers: stationary current controller and rotating current controller.
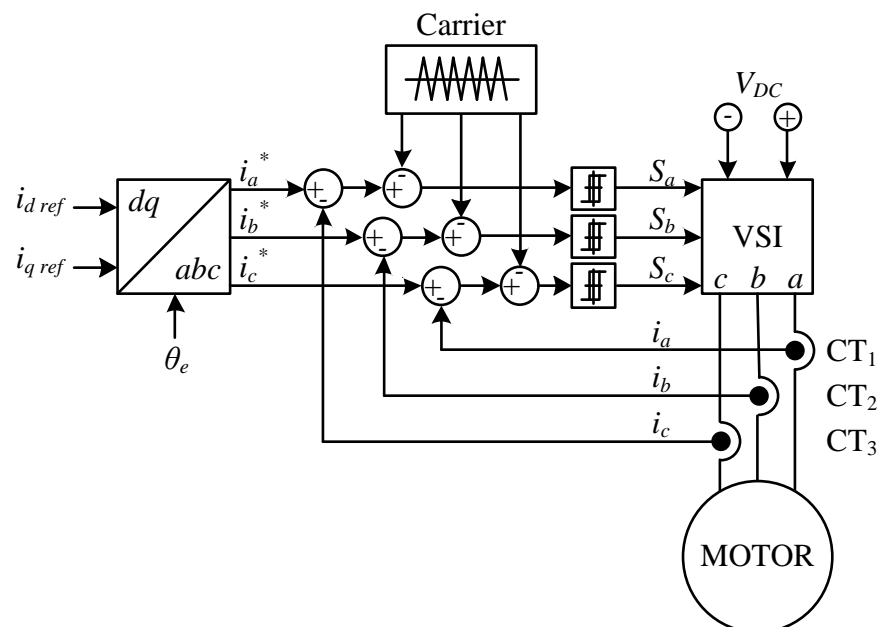
Figure 2.12: Stationary current controller

Figure 2.12 shows the stationary current controller block diagram. In stationary current controller, three current sensors are required to perform the actual stator current measurement. These actual currents then are compared to the reference current before comparing with carrier signal and producing switching signal for the VSI.
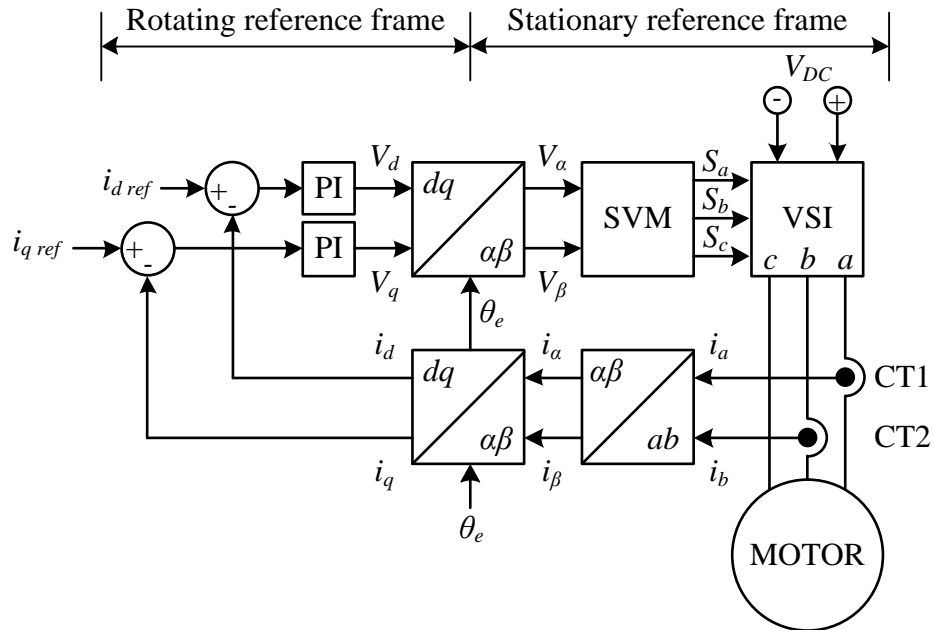
Figure 2.13: Rotating current controller

In rotating current controller, only two current sensors are needed for stator current measurement. As illustrated in Figure 2.13, the current sensor outputs are converted into rotating reference frame by using Clarke and Park transformation. Its result is compared to dq-axis current references and the controller takes the appropriate action to follow the reference current.

2.3.2 Reference Frame Transformation

Reference frame transformation was proposed by Park (1929) for simplification of three-phase system analysis. Generally, there are two main types of reference frames: stationary reference frame and rotating reference frame. There are four transformations:

Clarke, Park, inverse-Park and inverse-Clarke. Clarke and Park transformations are used for stationary (abc-axis) to rotating (dq-axis) reference frame transformation while reverse processing is done by inverse-Clarke and inverse-Park. In Clarke transformation, the abc space vector component is projected perpendicular to $\alpha\beta$-axis as shown in Figure 2.14.
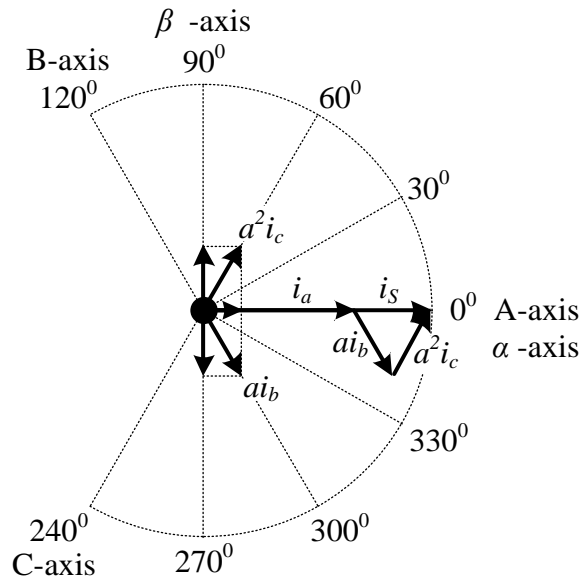
Figure 2.14: Clarke transformation

The Clarke transformation is formulated as (Duesterhoeft, Schulz, & Clarke, 1951),

$$\begin{bmatrix} x_\alpha \\ x_\beta \\ x_0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \\ 1/2 & 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} \qquad (2.18)$$

And inverse-Clarke transformations are given by,

$$\begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ -1/2 & \sqrt{3}/2 & 1 \\ -1/2 & -\sqrt{3}/2 & 1 \end{bmatrix} \begin{bmatrix} x_\alpha \\ x_\beta \\ x_0 \end{bmatrix} \qquad (2.19)$$

To obtain dq-component, $\alpha\beta$-component is projected perpendicular to the dq-axis which rotates at ω_e as is illustrated in Figure 2.15. Park and inverse-Park transformation are expressed in equation (2.20) and (2.21), respectively:

$$\begin{bmatrix} x_d \\ x_q \end{bmatrix} = \begin{bmatrix} \cos\theta_e & \sin\theta_e \\ -\sin\theta_e & \cos\theta_e \end{bmatrix} \begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} \tag{2.20}$$

$$\begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} = \begin{bmatrix} \cos\theta_e & -\sin\theta_e \\ \sin\theta_e & \cos\theta_e \end{bmatrix} \begin{bmatrix} x_d \\ x_q \end{bmatrix} \tag{2.21}$$
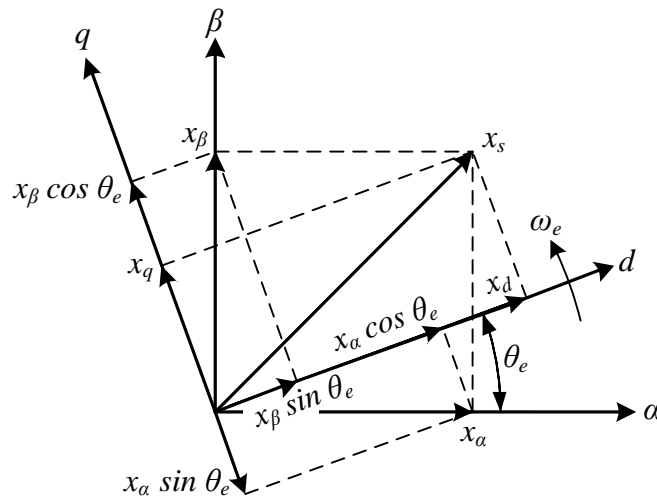
Figure 2.15: Park transformation

2.3.3 Space Vector Pulse Width Modulation

Several Pulse Width Modulation (PWM) techniques such as: Sinusoidal PWM (SPWM), Third Harmonics Injection PWM (THIPWM) and Space Vector PWM (SVPWM) can be implemented in VSI. However, the SVPWM is the most popular modulation technique among the others. It found widespread use in the industry in the early seventies (Neacsu, 2001). SVPWM offers some advantages such as: less-harmonics (van der Broeck, Skudelny, & Stanke, 1988) and effective utilization of the DC link voltage, 15% higher than SPWM.

The space vector diagram consists of six regions as shown in Figure 2.16. There are three steps in the SVPWM process:

- *αβ*-axis voltage, reference voltage vector and angle calculation

- Time duration calculation

- Switching time determination of each switch
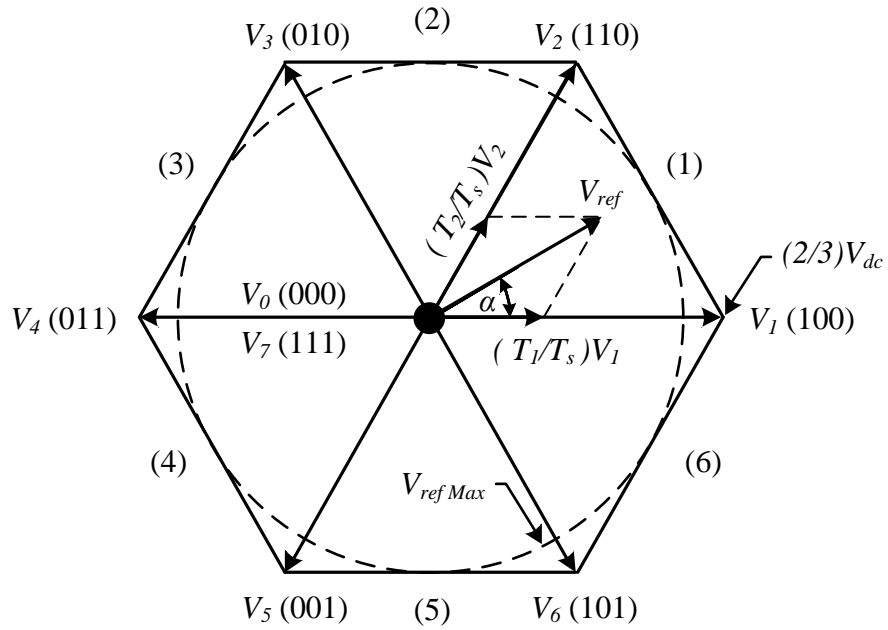
Figure 2.16: Space vectors diagram

A. **Calculation of *αβ*-axis voltage, reference voltage vector and angle**

The *αβ*-axis voltage is obtained from:

$$\begin{bmatrix} V_\alpha \\ V_\beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix} \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} \tag{2.22}$$

The reference voltage vector and angle is calculated as given in (2.23) and (2.24), respectively:

$$\left| V_{ref} \right| = \sqrt{V_\alpha{}^2 + V_\beta{}^2} \tag{2.23}$$

$$\alpha = \tan^{-1}\left(\frac{V_\beta}{V_\alpha}\right) \qquad (2.24)$$

Where maximum $V_{ref} = 1/\sqrt{3}$ that is represented by the dash-circle line in Figure 2.16.

B. Time duration calculation

Referring to Figure 2.17, the relationship between time durations and voltages vector in sector 1 is:

$$\vec{V}_{ref}\, T_S = \vec{V}_1\, T_1 + \vec{V}_2\, T_2 + \vec{V}_0\, T_0 \qquad (2.25)$$

where:

$$
\begin{aligned}
\vec{V}_0 &= 0 \\
\vec{V}_1 &= \frac{2}{3} Vdc\; e^0 \\
\vec{V}_2 &= \frac{2}{3} Vdc\; e^{j\pi/3} \\
\vec{V}_{ref} &= \left|V_{ref}\right| e^{j\alpha}
\end{aligned}
\qquad (2.26)
$$
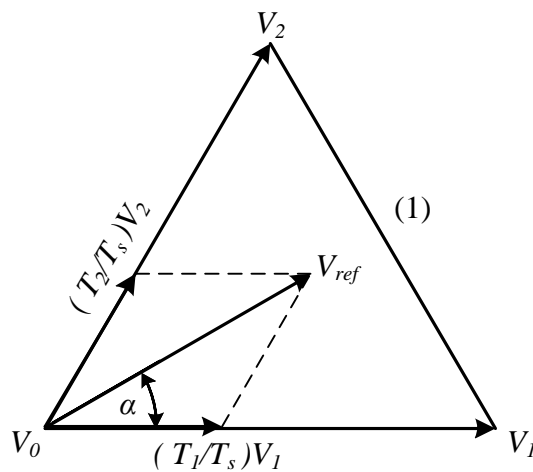
Figure 2.17: Sector 1 of space vector diagram

23

Substitution of the equation (2.25) and (2.26) yields time duration as given in:

$$T_1 = M \; T_S \; \frac{\sin\left(\dfrac{\pi}{3} - \alpha\right)}{\sin\left(\dfrac{\pi}{3}\right)} \tag{2.27}$$

$$T_2 = M \; T_S \; \frac{\sin(\alpha)}{\sin\left(\dfrac{\pi}{3}\right)} \tag{2.28}$$

where:

$$T_S = 1/f_S$$
$$M = \frac{\left|V_{ref}\right|}{2/3 V_{dc}} \tag{2.29}$$

Time duration for any sector n (n = 1 - 6) is formulated as,

$$T_1 = \frac{\sqrt{3} \; T_S \; \left|V_{ref}\right|}{V_{dc}} \left(\sin\frac{n}{3}\pi \cos\alpha - \cos\frac{n}{3}\pi \sin\alpha \right) \tag{2.30}$$

$$T_2 = \frac{\sqrt{3} \; T_S \; \left|V_{ref}\right|}{V_{dc}} \left(-\cos\alpha \sin\frac{n-1}{3}\pi + \sin\alpha \cos\frac{n-1}{3}\pi \right) \tag{2.31}$$

$$T_0 = T_S - \left(T_1 + T_2\right) \tag{2.32}$$

C. Switching time determination of each switches

After obtaining the time duration in appropriate sector, the next step is determination of the switching time for each switch: upper (S1, 3 and 5) and lower (S2, 4 and 6) switches of VSI. The switching time is determined by the following calculation that can be summarized in Table 2.1 and its pattern is illustrated in Figure 2.18.

Sector 1 ($0° \leq \alpha \leq 60°$)

Sector 2 ($60° \leq \alpha \leq 120°$)

Sector 3 ($120° \leq \alpha \leq 180°$)

Sector 4 ($180° \leq \alpha \leq 240°$)
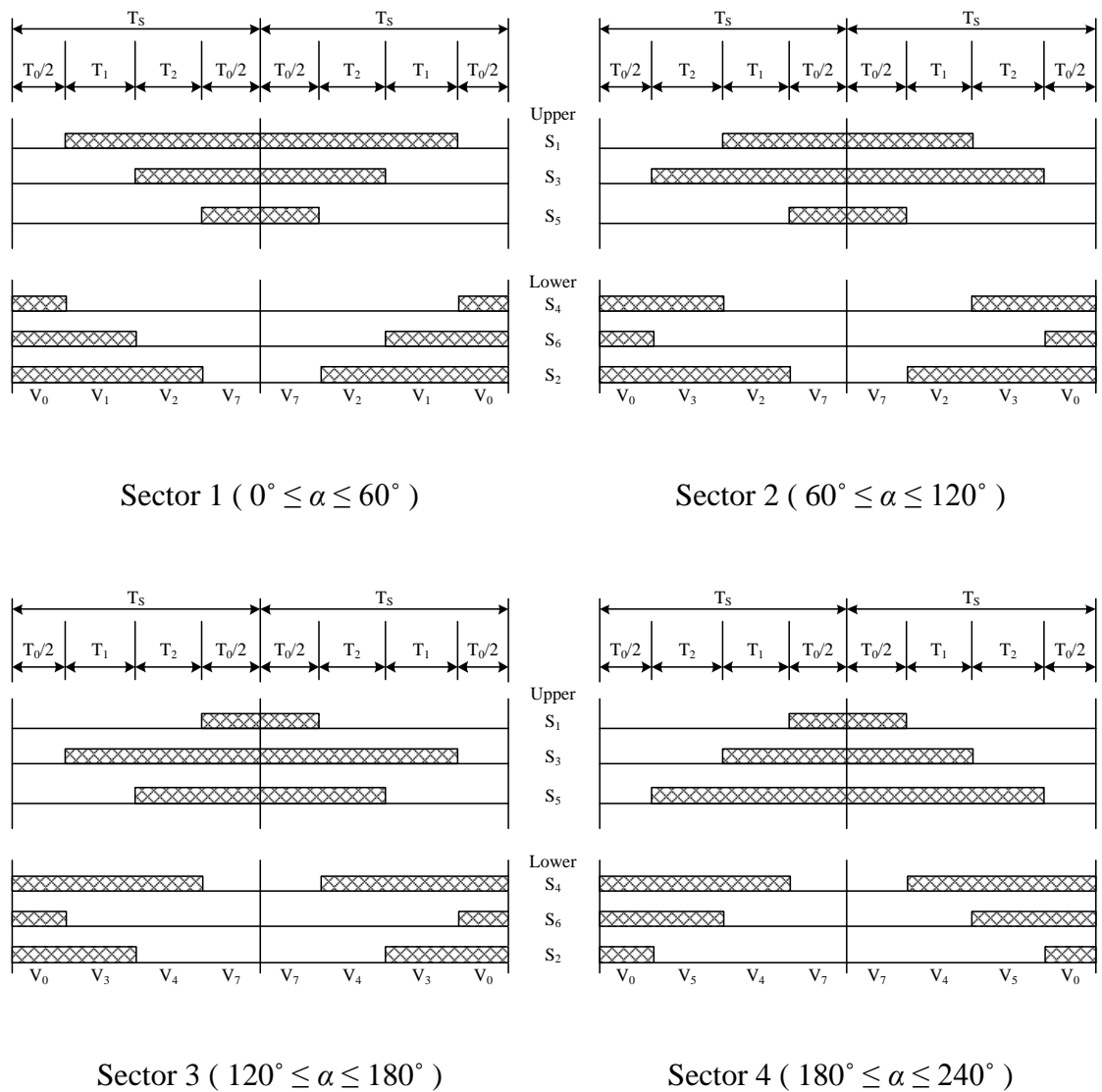
Figure 2.18: Switching pattern for each sectors - n
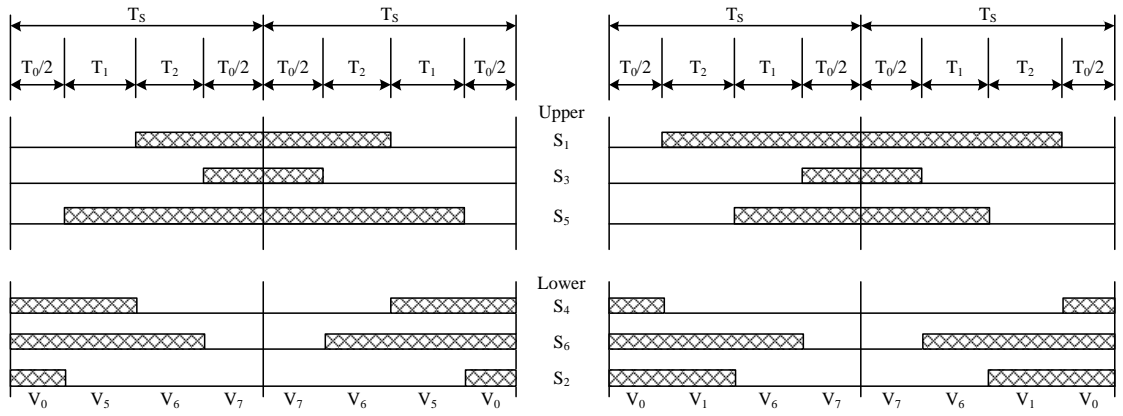
Sector 5 ($240° \leq \alpha \leq 300°$) Sector 6 ($300° \leq \alpha \leq 360°$)

Figure 2.18: Switching pattern for each sectors – n (continued)

Table 2.1: Switching time calculation for different sectors

Sector	Upper switches	Lower switches
1	$S_1 = T_1 + T_2 + T_0/2$ $S_3 = T_2 + T_0/2$ $S_5 = T_0/2$	$S_4 = T_0/2$ $S_6 = T_1 + T_0/2$ $S_2 = T_1 + T_2 + T_0/2$
2	$S_1 = T_1 + T_0/2$ $S_3 = T_1 + T_2 + T_0/2$ $S_5 = T_0/2$	$S_4 = T_2 + T_0/2$ $S_6 = T_0/2$ $S_2 = T_1 + T_2 + T_0/2$
3	$S_1 = T_0/2$ $S_3 = T_1 + T_2 + T_0/2$ $S_5 = T_2 + T_0/2$	$S_4 = T_1 + T_2 + T_0/2$ $S_6 = T_0/2$ $S_2 = T_1 + T_0/2$
4	$S_1 = T_0/2$ $S_3 = T_1 + T_0/2$ $S_5 = T_1 + T_2 + T_0/2$	$S_4 = T_1 + T_2 + T_0/2$ $S_6 = T_2 + T_0/2$ $S_2 = T_0/2$
5	$S_1 = T_2 + T_0/2$ $S_3 = T_0/2$ $S_5 = T_1 + T_2 + T_0/2$	$S_4 = T_1 + T_0/2$ $S_6 = T_1 + T_2 + T_0/2$ $S_2 = T_0/2$
6	$S_1 = T_1 + T_2 + T_0/2$ $S_3 = T_0/2$ $S_5 = T_1 + T_0/2$	$S_4 = T_0/2$ $S_6 = T_1 + T_2 + T_0/2$ $S_2 = T_2 + T_0/2$

2.3.4 Voltage Reconstruction Calculator

For sensorless drives, the stator phase voltage is not obtained directly from the stator terminal. It should be calculated by using Voltage Reconstruction Calculator (VRC). The variable inputs of voltage reconstruction are DC bus voltage and switching signals generated by the space vector modulator. The phase voltage reconstruction is formulated in equation (2.33). The phase voltage is converted into $\alpha\beta$-axis by using Clarke transformation as expressed in equation (2.18).

$$\begin{bmatrix} V_{an} \\ V_{bn} \\ V_{cn} \end{bmatrix} = \frac{V_{DC}}{3} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} S_a \\ S_b \\ S_c \end{bmatrix} \tag{2.33}$$

2.3.5 Stator Flux Observer

For a sensorless drive system, the speed and position of rotor is obtained by stator flux estimation using an observer. In general, Stator Flux Observer (SFO) can be divided into two main types: current model (CM) and voltage model (VM) based observer. As shown in Figure 2.19, the stator flux can be estimated simply by using pure integration (voltage based observer) as expressed in (2.34).

$$\lambda_S^* = \int \left(v_S - i_S R_S \right) dt \tag{2.34}$$

This is a simple conventional method but it encounters DC drift and offset value problem (Foo & Rahman, 2010; Gilsu, Wook-Jin, Junho, & Dalho, 2011; Paicu, Boldea, Andreescu, & Blaabjerg, 2009).
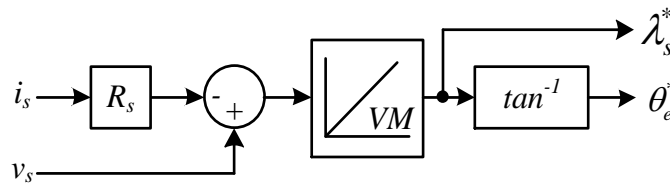
Figure 2.19: Stator flux observer with pure integration

SFO for IPMSM with extended rotor flux using dSpace DS1104 processor was investigated by Foo & Rahman (2010). They proposed an extended rotor flux concept that acts as a virtual permanent magnet flux linkage where the IPMSM is converted into SPMSM. The SFO is formed by the combination of the current model (CM) and voltage model (VM). In 2011, Gilsu (2011) proposed a simple SFO with proportional-integrator controller to reduce the rotor position error.

In this IPMSM drive system, the proposed SFO algorithm is implemented in DSP TMS320F2812 platform. The proposed SFO algorithm without extended rotor flux calculation (Figure 2.20) is used for DSP's internal memory saving. The free-space of DSP's internal memory can be used for other calculation in future system development. The calculation in SFO involves motor parameters such as dq-axis inductances and stator resistance. The permanent magnet flux linkage of motor is used to verify the correctness of flux linkage estimation.
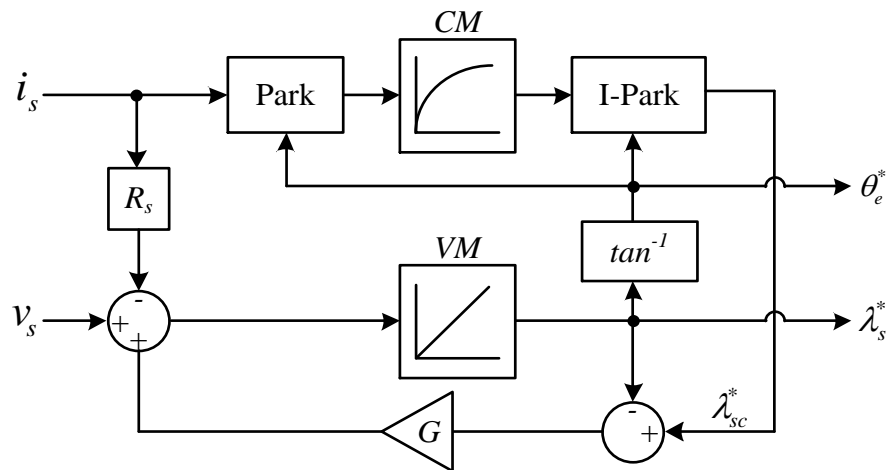
Figure 2.20: The proposed stator flux observer block diagram

In the proposed SFO, the $\alpha\beta$-axis stator currents (i_s) are transformed into dq-axis stator currents by applying Park's transformation as expressed in:

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} \cos\theta_e^* & \sin\theta_e^* \\ -\sin\theta_e^* & \cos\theta_e^* \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \tag{2.35}$$

The current model is calculated from equation 2.8 and 2.35 in order to obtain dq-axis stator flux linkages. They are converted into stationary reference frame by using inverse-Park transformation in the equation below:

$$\begin{bmatrix} \lambda_{SC-\alpha}^* \\ \lambda_{SC-\beta}^* \end{bmatrix} = \begin{bmatrix} \cos\theta_e^* & -\sin\theta_e^* \\ \sin\theta_e^* & \cos\theta_e^* \end{bmatrix} \begin{bmatrix} \lambda_d \\ \lambda_q \end{bmatrix} \tag{2.36}$$

Then, the estimated stator flux linkages are obtained from:

$$\lambda_S^* = \int \left(v_S - i_S R_S + G\Delta\lambda_S^* \right) T_S \ dt \tag{2.37}$$

where, the flux error is defined as,

$$\Delta\lambda_S^* = \lambda_{SC}^* - \lambda_S^* \tag{2.38}$$

From the calculated stator flux, the estimated electrical rotor position is then given by,

$$\theta_e^* = \tan^{-1}\left(\frac{\lambda_\beta^*}{\lambda_\alpha^*} \right) \tag{2.39}$$

The relationship between mechanical and electrical rotor position can be expressed as:

$$\theta_m^* = \frac{2}{P}\theta_e^* \tag{2.40}$$

Then, the estimated mechanical speed can be obtained from the derivation of rotor position as follows:

$$\omega_m^* = s\theta_m^* \tag{2.41}$$

2.3.6 Cross-Coupling and Decoupling

As discussed in the previous section, d-axis and q-axis voltage equation is affected by another axis component. Also, similar nonlinearity can be inferred from the electromagnetic torque equation. As a result, the control techniques are not easy to implement when the motor is operated at d-axis current reference value not equal to zero. Furthermore, an accurate speed control cannot be achieved for low switching frequency time sampling. Therefore, it is necessary to eliminate voltage disturbance by using decoupling component as shown in Figure 2.21. However, decoupling component can be ignored when the motor drive system is operated in high switching frequency (Wilamowski & Irwin, 2011). Operation in high switching frequency makes the current controllers react faster to follow the desired current references. In addition, operation in d-axis current reference equal to zero can also solve the nonlinearity problem of the motor.
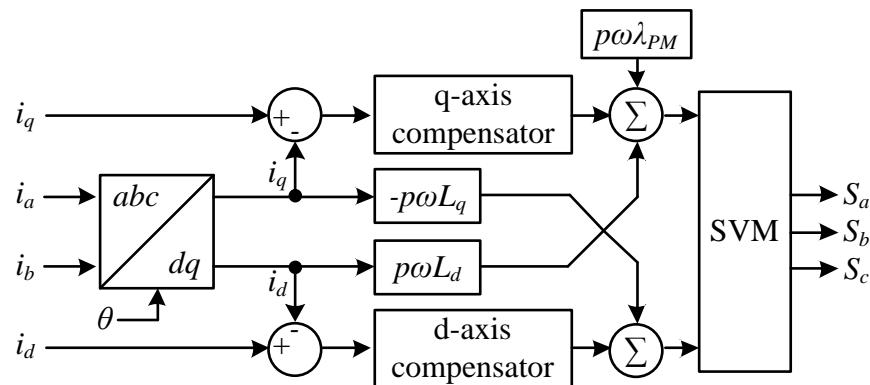
Figure 2.21: Block diagram of FOC with decoupling

2.4 Control Modes of PMSM

The electromagnetic torque production of PMSM depends on the torque angle value (α). Torque angle is defined as the angle between stator current space vector and d-axis, as shown in Figure 2.22.
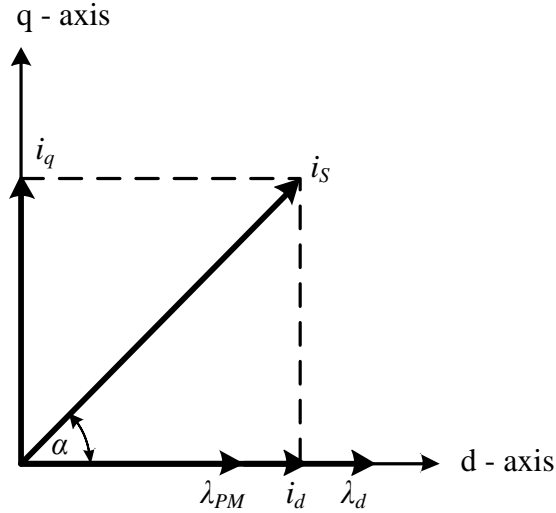
Figure 2.22: Torque angle

where:

$$i_d = i_S \cos \alpha$$
$$i_q = i_S \sin \alpha \tag{2.42}$$
$$|i_S| = \sqrt{i_d{}^2 + i_q{}^2}$$

By substitution of equations (2.10) and (2.42), the electromagnetic torque as function of torque angle becomes:

$$T_e = \frac{3}{2}\frac{P}{2}\left(\lambda_{PM}\,|i_S|\sin\alpha - \left(L_q - L_d\right)|i_S|^2\,\cos\alpha\,\sin\alpha\right) \tag{2.43}$$

Several different types of control modes that can be used in PMSM drives system are (Krishnan, 2001; Monajemy & Krishnan, 2001; Morimoto, Takeda, & Hirasa, 1990; Pillay & Krishnan, 1989): constant torque angle control (CTA), maximum

torque angle per ampere (MTPA), unity power factor (UPF), and constant stator flux control (CSF).   One of these control modes is chosen depending on the type of PMSM as discussed in the previous section. The selection of control modes depends on the motor criteria/characteristic and performance requirements (Morimoto et al., 1990). For maximum torque extraction, CTA and MTPA are commonly used in PMSM drive system.

2.4.1 Constant Torque Angle Control (CTA)

CTA control (or $i_d = 0$ control) is used to keep torque angle value constant at 90 degrees ($\alpha = 90°$). Hence, to achieve and maintain torque angle at 90 degrees, the direct-axis reference current (i_{dref}) should be set to 0 values. Moreover, it can be kept constant due to existence of constant flux that is provided by permanent magnet. As a result, the torque production depends only on the quadrature-axis current (i_q) and it is proportional to the stator current (i_s). Figure 2.23 shows the current vector in CTA control.

Typically, this control mode is used for non-salient SPMSM ($L_d \approx L_q$) where maximum torque can be achieved when the torque angle is equal to 90 degrees. However, it is not suitable for IPMSM due to saliency ($L_q \gg L_d$) where the maximum torque can be obtained when the torque angle is larger than 90 degrees. It means the d-axis current reference should be set to a negative value. However, CTA control still can be used for IPMSM. But it is not recommended when additional torque needs to be extracted from the motor. The reluctance torque is not utilized properly since i_d reference is maintained at zero. Hence, reluctance torque vanishes from the equation (2.10) and only permanent magnet torque component remains.

q - axis

$i_q = i_S$

α

d - axis

$I_d = 0$

λ_{PM}
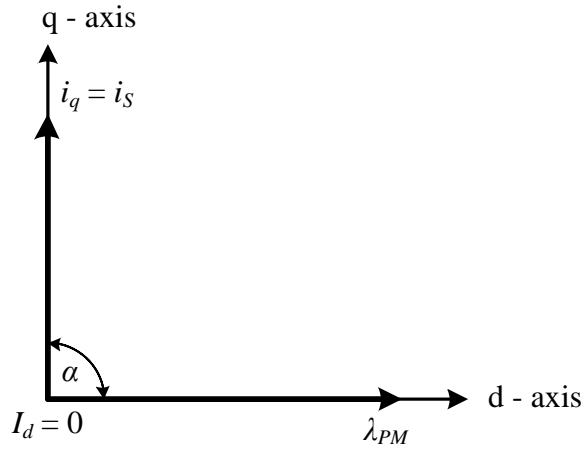
Figure 2.23: CTA control current vector

Because $i_d = 0$ and $\alpha = 90^0$, the reluctance torque component is cancelled out. Then, the electromagnetic torque in (2.43) becomes:

$$T_e = T_m = \frac{3}{2}\frac{P}{2}\,\lambda_{PM}\;i_q = K_T\;i_q \qquad (2.44)$$

where, K_T represents the torque constant.

In SPMSM, however, there are slight differences between d-axis and q-axis inductance. The additional reluctance torque provided is insignificant compared to the total of electromagnetic torque. Hence, it is better to keep the i_d reference zero. It is useful not only to achieve maximum torque but also to eliminate the non-linearity in the electromagnetic torque equation. Furthermore, it avoids demagnetization of permanent magnet. The i_d value can be changed to negative value when the motor needs to be run beyond the rated speed (flux weakening region).

2.4.2 Maximum Torque per Ampere Control (MTPA)

Another torque control mode is maximum torque per ampere control (MTPA). In this control mode, torque angle is controlled more than 90 degrees in order to get maximum torque output with minimum stator current. Thus, the electromagnetic torque is not only controlled by quadrature-axis current (i_q) but also by direct-axis current (i_d). The current vector of MTPA mode is shown in Figure 2.24. The electromagnetic torque in MTPA is:

$$T_e = \frac{3}{2}\frac{P}{2}\left(\lambda_{PM}|i_s|\sin(\alpha) - \frac{1}{2}\left(L_q - L_d\right)|i_s|^2\sin(2\alpha)\right) \qquad (2.45)$$

This control mode is suitable for IPMSM due to the appearance of reluctance torque. Thus, it provides additional torque at a certain torque angle when MTPA control is used.
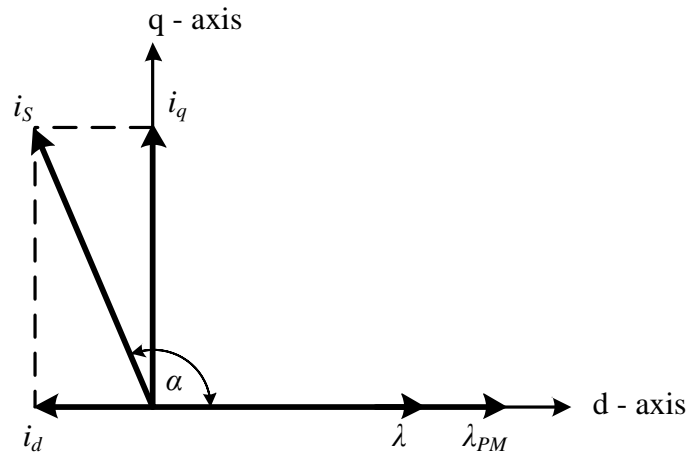
Figure 2.24: MTPA control current vector

CHAPTER 3

HARDWARE AND SOFTWARE CONFIGURATION

3.1 Hardware Configuration

This section describes the overall hardware configuration used for the motor drive system in this research. The complete hardware configuration consists of insulated-transformer (T), auto-transformer (AT), single-phase rectifier, three-phase inverter, gate drives, TMS320F2812 EzDSP development board, current sensors, 920 W, 4-poles SPMSM, 600 W, 4-poles IPMSM, incremental encoder and signal-conditioning-circuit (SCC). The experimental setup diagram for SPMSM and IPMSM is shown in Figure 3.1 and Figure 3.2, respectively. The detailed information and function of each part is explained in the next section.
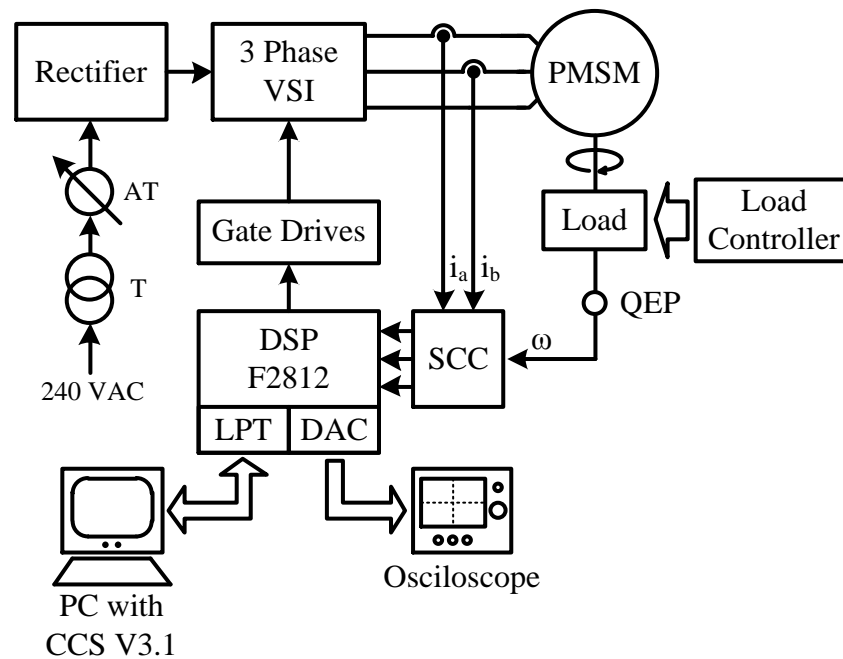
Figure 3.1: SPMSM inverter drive system experimental setup
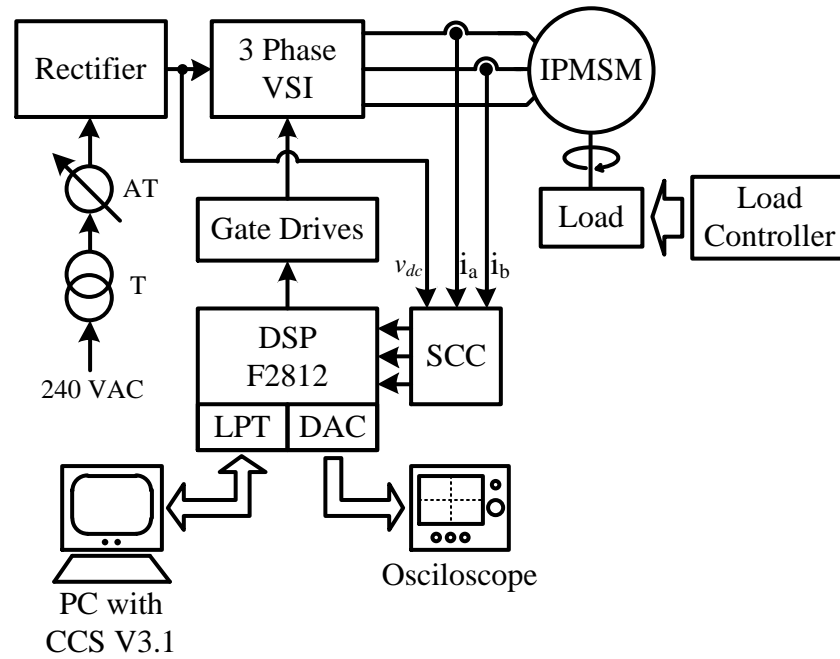
Figure 3.2: IPMSM inverter drive system experimental setup

3.1.1 Rectifier

A single-phase uncontrolled full-bridge rectifier is used as an input DC supply to the three-phase inverter. As shown in Figure 3.3, this rectifier consists of two different transformers: insulated-transformer and auto-transformer. The insulated-transformer with rating up to 1 kVA, 1:1 winding ratio is used as protection and insulation between the inverter drive system and power line from the grid while the auto-transformer is used to obtain variable AC output from 0 to 240 V AC. For rectification, the output of autotransformer is connected to a single-phase full-bridge rectifier and smoothed by a capacitor. As a result, 339 V DC bus voltage (maximum) is produced across the +HVDC and –HVDC terminal when 240 V AC of Vin is applied to the rectifier.

Two inductances are connected in series to perform additional filtering. These inductances are used to smooth the peak current as a result of capacitor switching and suppress line's harmonic current.
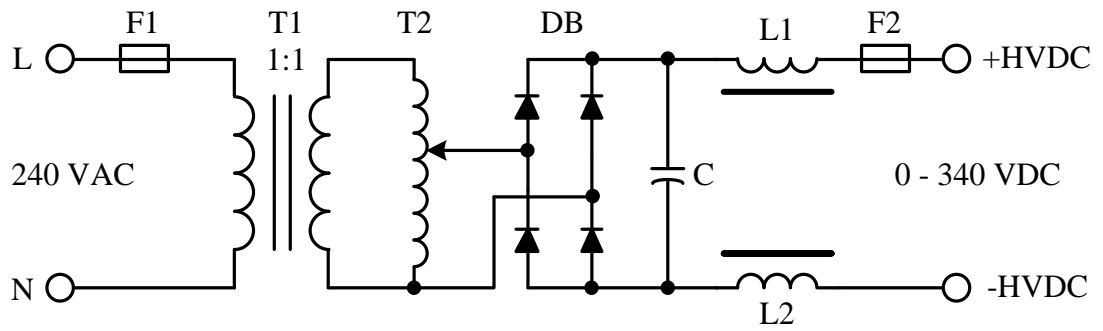
Figure 3.3: DC-link supply circuit

3.1.2 Three phase VSI

As power switches, six Infineon's Insulated Gate Bipolar Transistors (IGBTs) with rating up to 1200 V/15 A (IKW15T12) are used to form three phase voltage source inverter (VSI) as shown in Figure 3.4. It was powered with 300 V of DC-link voltage from a single phase rectifier. 300 VDC of the DC-link voltage was selected to run the PMSM motor up to the nominal speed.
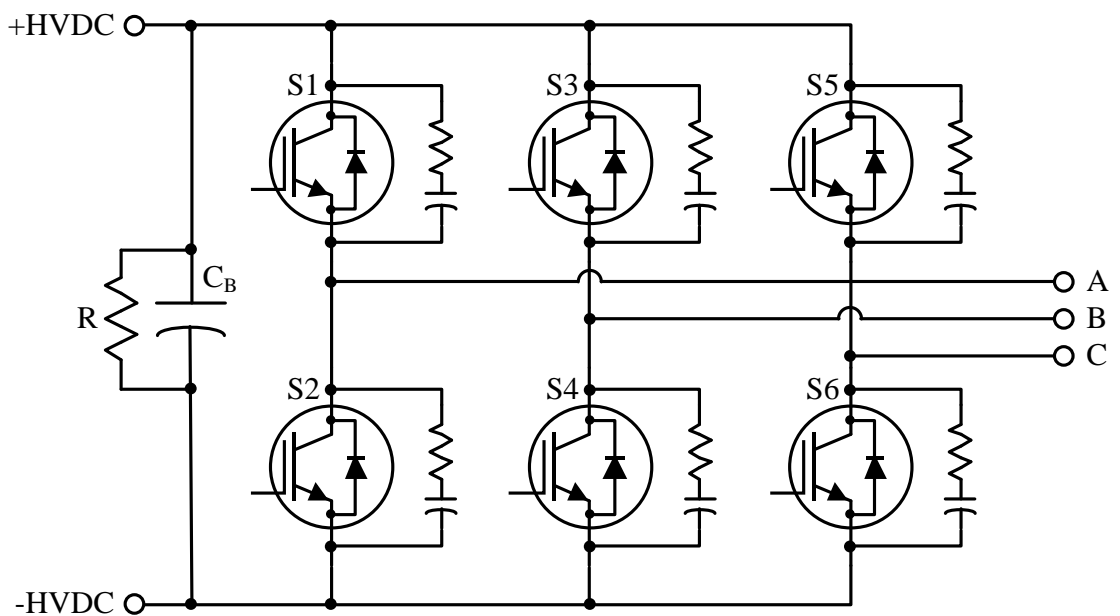
Figure 3.4: Three-phase VSI

The inverter converts DC voltage into AC voltage with variable frequency output by using space vector modulation for IGBT switching. The lower-IGBT signals are complementary to the upper-IGBTs and a dead band time is added to make sure the upper and the lower-IGBTs in the same IGBT legs are not connected at the same time which would make the DC-link become short-circuited. 2 µs of dead band time is used in this drive system. The signals are shifted by 120 degrees for each IGBT leg. To minimize the switching losses, each IGBT was equipped with RC snubber circuit, 10 Ω / 2 W and 0.001 µF / 1000 V for the resistor and the capacitor values, respectively.

A DC link capacitor is placed as close as possible across the inverter to reduce the line's impedance between the capacitor and the IGBTs. Moreover, for safety reasons, 22 k / 10 W resistor is put in parallel across the capacitor to discharge the capacitor's voltage when the system is turned off for a long period.

3.1.3 Gate Drive

Typically, an IGBT requires +-20 V input signal level to turn-on, but the DSP can only generate a PWM signal of +3.3 V output level. Although it still can be used, it would give rise to some problems. The IGBTs cannot deliver high-current and high-power rate to the load by applying +3.3 V input signal for the IGBTs. Another problem is the DSP is exposed to the high-voltage side (inverter) directly without any protection. It can be harmful to the DSP when over-voltage, over-current or short-circuit suddenly occurs at high-voltage side.

Due to the insufficient input signal level to the IGBT and the requirement to protect the DSP, gate driver is needed. The complete block of gate drive circuit is shown in Figure 3.5. One side of the opto-coupler is connected to the DSP and the other side is connected to the IGBT, and both sides are separated optically. The gate drive

circuit consists of two main parts. The first part is voltage increaser circuit called insulated DC-DC converter and the second part is opto-coupler part.
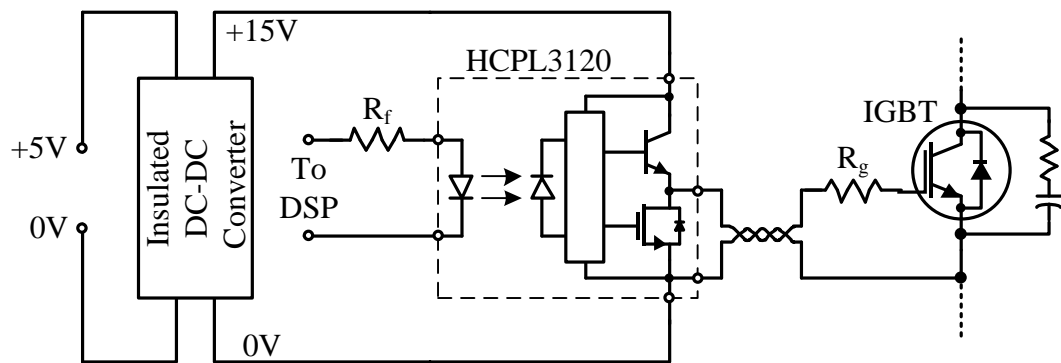
Figure 3.5: Gate drive circuit

The insulated DC-DC converter, as shown in Figure 3.6, increases the DC input voltage from +5 V to +15 V level. First, the oscillator produces high-frequency pulse with +5 V of amplitude and then this pulse amplitude is increased by high-frequency transformer (HFT) with 7:25 winding ratio. The output of transformer is rectified with a full-bridge diode to produce DC output. It has to be maintained at +15 V level by using voltage regulator.

The output of regulator is used to power-up the opto-coupler. In this drive system, HCPL3120 was used as opto-coupler and IGBT gate driver. HCPL3120 can drive the IGBTs with rating up to 1200 V / 100 A, directly. Furthermore, HCPL3120 is equipped with under voltage lockout (UVLO) that has a function to protect the IGBT when the supply of opto-coupler becomes less than 11 V.
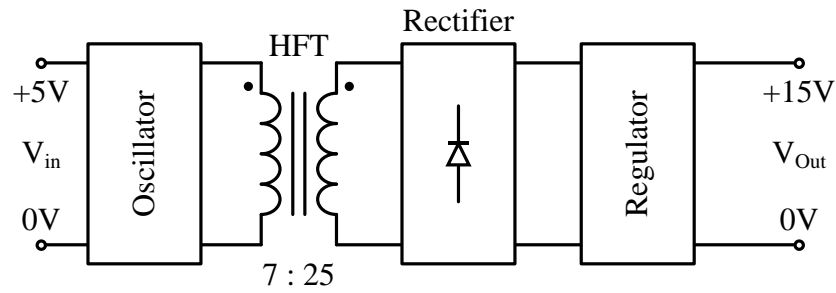
Figure 3.6: Insulated DC-DC converter

The connection between the gate driver and the IGBT should be as short as possible and also twisted or coaxial cables are recommended to prevent the IGBTs from picking up the noises/glitches from inside or outside the system. It is discussed in more detail in section 3.2.

3.1.4 Sensors and Signal Conditioning Circuits

This section describes the sensors and signal conditioning circuit. There are two kinds of sensors that are used in the motor drive system: speed sensor and current sensor. Due to DSP input signals being limited to 3.3 V for maximum value and 0 V for minimum value, these signals must be scaled-down by using the signal conditioning circuit. Moreover, the signal conditioning circuit gives protection to avoid DSP damage.

A. Speed and Position Sensor

In this system, the Quadrature Encoder Pulse (QEP) sensor is proposed because it is commonly used for rotor position and speed measurement of PMSM. Moreover, the QEP is not only cheaper than absolute encoder but also has the same capability as the absolute encoder; it can determine the absolute rotor position as well.

QEP encoder EH60E-5000 series from Autonics (Figure 3.7) is used in this motor drive system. QEP sensor is a kind of incremental encoder and it produces three signals: channel A, channel B, and channel Z or the index signal. Channel A and B are square-wave signal with 90 degrees difference between each other. It can be lagging or leading depending on the direction of rotor rotation. Both signals generate 5000 pulses per one complete rotation. Internally, through the software, these pulses are multiplied by 4 to obtain more accuracy in calculation and avoid missing counting at low speed operation of motor.
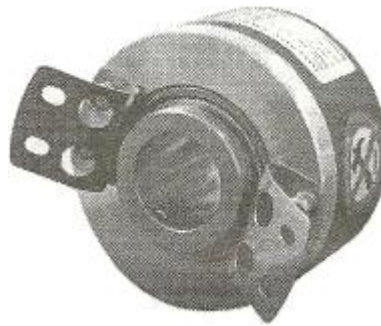
Figure 3.7: Autonics QEP EH60E series

The index signal produces a single pulse for every one complete mechanical revolution, as shown in Figure 3.8. It is used as QEP interrupt signal and resets the timer counter before a new round of counting is started.
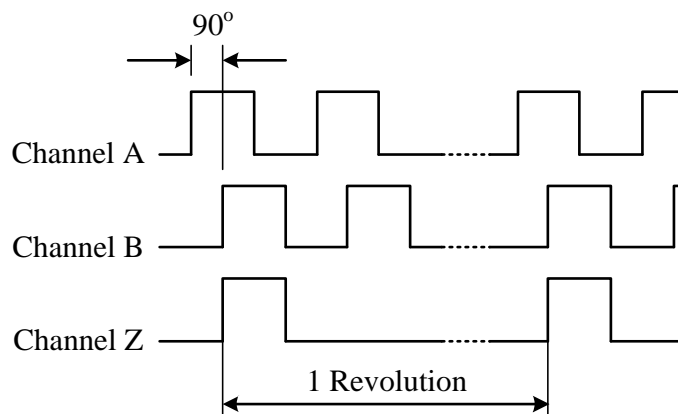
Figure 3.8: Output signal of incremental encoder with index

In implementation, the QEP sensor must be calibrated before it is used in the motor drive system. A simple calibration method is by measuring the time difference between the QEP index signal and the back-emf waveform using an oscilloscope (Konghirun, 2005, 2008). Thus, the time difference would be considered as the offset value of the QEP and it is put inside the algorithm as calibration angle value. The motor would draw more current from the DC-link with an incorrect value of calibration angle and the optimum torque production is not achieved in this condition.

Figure 3.9 shows an illustration of rotor position alignment. The notation "θ_{e1}" indicates the incorrect rotor position angle which is obtained from the measurement of encoder and θ_e is the correct rotor position angle after calibration angle is added to θ_{e1}.
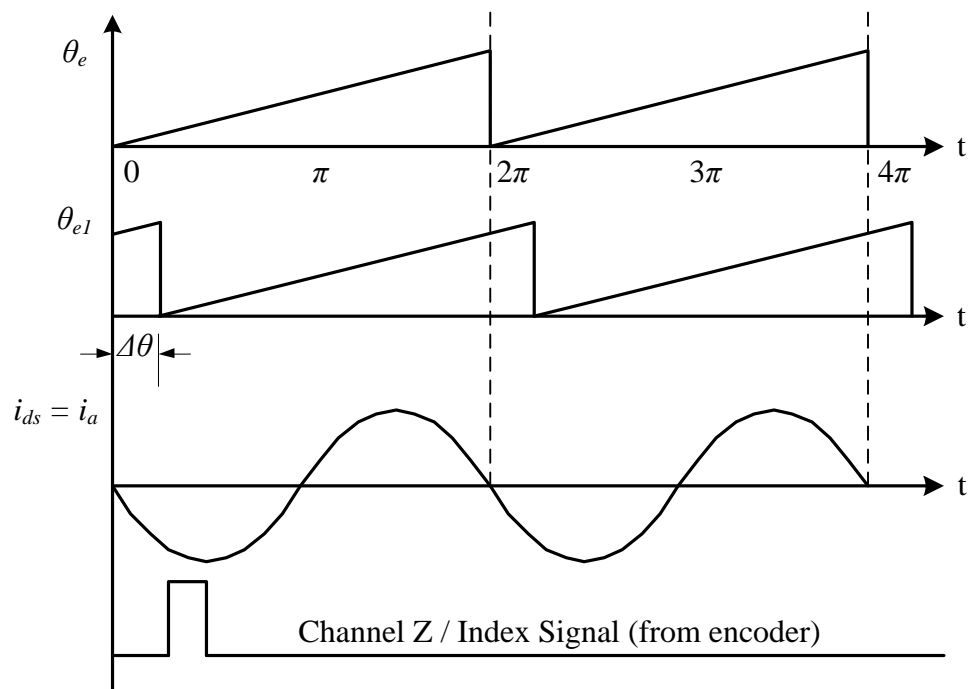
Figure 3.9: Rotor position alignment

The misalignment of rotor position is illustrated in dq-axis current vector as shown in Figure 3.10. The asterisk mark indicates the dq-axis with incorrect rotor position angle.
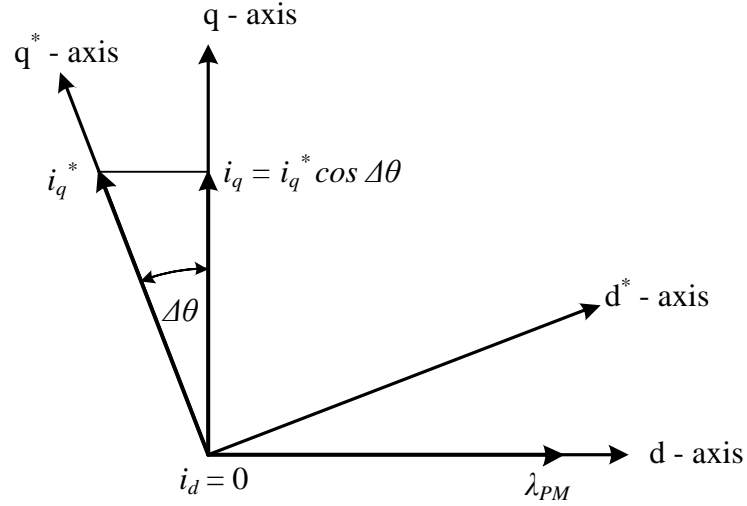
Figure 3.10: dq-axis current vector with misalignment of rotor position

Misalignment of rotor position makes equation 2.44 becomes:

$$T_e = \frac{3}{2} \frac{P}{2} \lambda_{PM} \ i_q^* \cos(\Delta\theta) \tag{3.1}$$

Where the rotor angle deviation is defined as,

$$\Delta\theta = \theta_{e1} - \theta_e \tag{3.2}$$

From equation (3.1), by assuming the value of electromagnetic torque (T_e) is equal to the load torque (T_L), which is a constant value, the q-axis current will increase as a result of misalignment of rotor angle. The optimum torque and current can be achieved when the rotor angle deviation is zero.

The QEP has NPN open-collector outputs and each channel need to pull-up by using a resistor connected to +5 V. Then Schmitt triggers (N1 and N2) are used to suppress the glitch and sharpen the pulse. Because only +3.3 V inputs are acceptable for the DSP, a trimmer-potentiometer (VR) is needed as a voltage divider to reduce the signal output from the Schmitt triggers at 3.3 V of maximum amplitude.

Figure 3.11 shows the interface circuit between the QEP and the DSP. The output of the circuit was connected to a special function pin which is dedicated for QEP encoder. Channel-A, B, and Z is connected to QEP1, QEP2, and QEPI1, respectively. The most important thing is that a shield wire (F.G) has to be connected to 0 V of power supply wire (Gnd) and protective-earth (PE) point together. Significantly, it reduces the noise of QEP output. As a result, the error in the measurement of the rotor position can be minimized.
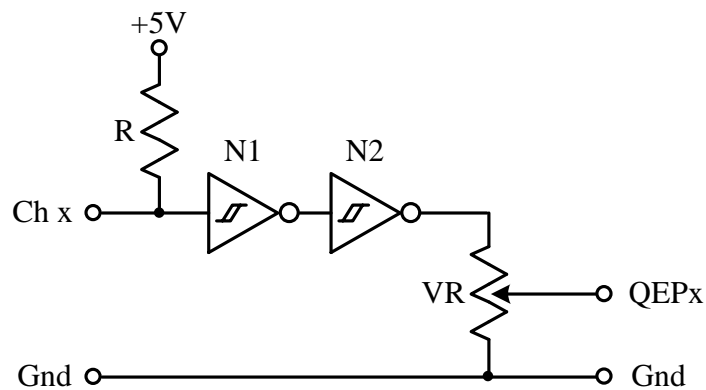
Figure 3.11: QEP's signal conditioning circuit

B. Current Sensor

The actual stator currents are obtained from Hall-effect current sensors. Two LEM Hall-effect current sensors (LA25-NP) are used as current feedback input to the DSP. These current sensors measure the stator current phase-a and phase-b then afterward phase-c can be calculated by using ic = -(ia+ib) equation.

Both of the sensors need +/- 15 VDC supply to operate. However, the current measurement output from these sensors cannot be applied to the DSP's ADC directly. In order to fulfil this ADC voltage range, the current sensors output needs a signal conditioning circuit to scale-down to that particular voltage range of signals. Also, the signal is shifted by 1.5 V of DC offset in order to avoid negative voltage.

Figure 3.12 shows the signal conditioning process. In this system, the actual current 8 A (peak) is set equal to 3 V.

(a)   (b)   (c)
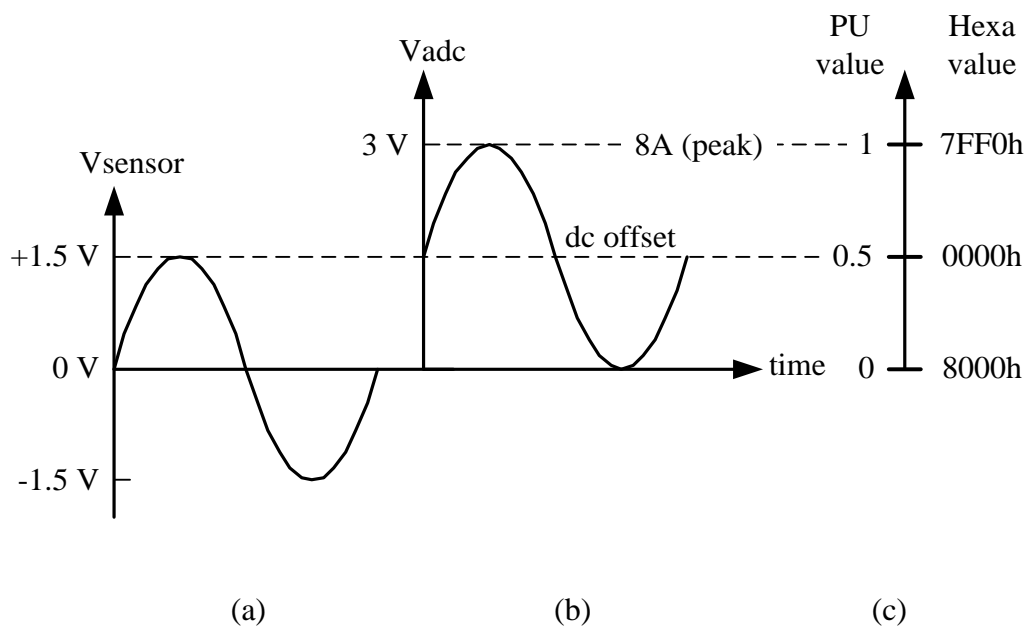
Figure 3.12: Current sensor signal conditioning
(a) scalling-down, (b) shifting-up the signal and (c) hexa number representation

Two clamping-diodes are placed as a backup protection to prevent the ADC's input signal lower or equal to +3.3 V and higher or equal to 0 V (0 V ≤ VADC ≤ +3.3 V) during the motor start-up, acceleration, or deceleration period. Figure 3.13 shows the configuration of current sensors and signal conditioning circuit. In addition, a simple low-pass-filter (LPF) is placed as close as possible to the ADC input terminal to provide signal filtering.
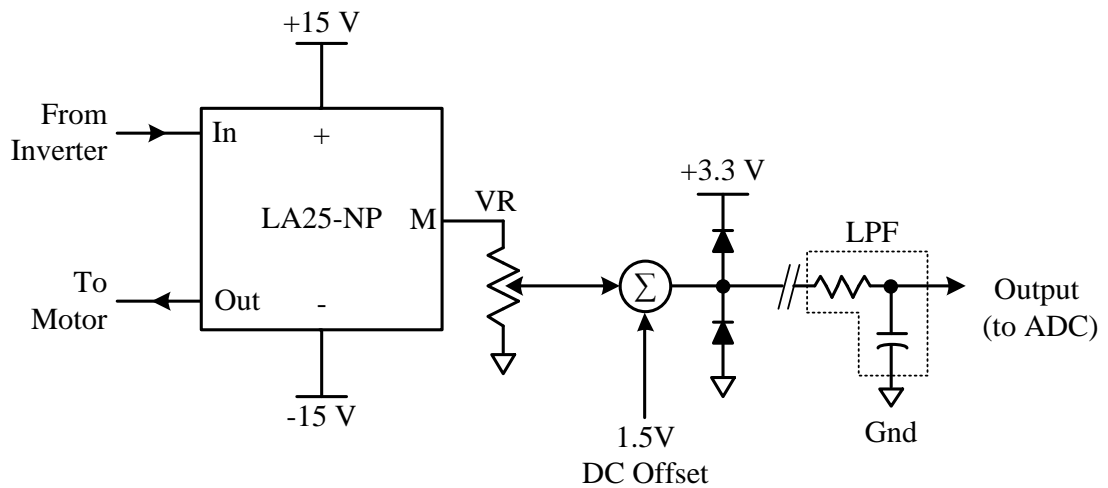
Figure 3.13: Current sensor circuit

C. Voltage Sensor

For voltage reconstruction calculation in sensorless drives system, a voltage sensor LEM LV25P is used to measure the DC-link voltage. The output of the voltage sensor needs to be scaled down from 300 VDC to 3 V as shown in Figure 3.14, before it is applied to the ADC input of DSP. Figure 3.15 shows the circuit diagram of voltage sensor.
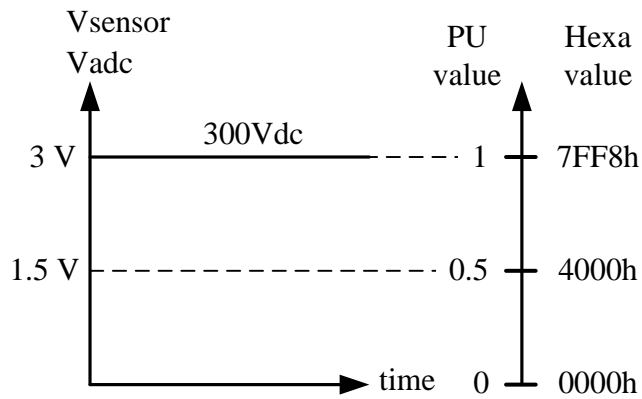
Figure 3.14: Voltage sensor signal conditioning
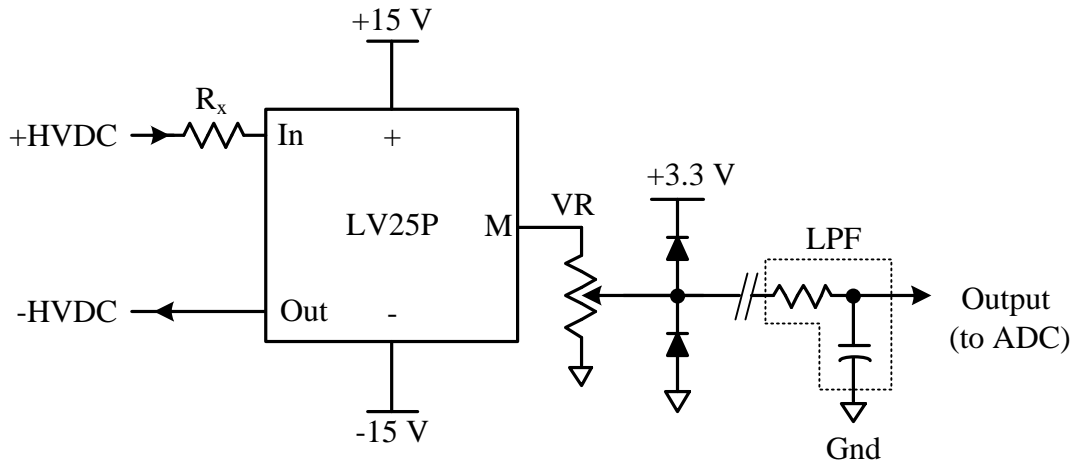
Figure 3.15: Voltage sensor circuit

3.2 Noise / Glitches Interferences

Noise or glitch reduction plays an important role in the motor drive system's stability and reliability. The source of the noises or glitches itself, can come from radio frequency interference (RFI) or electromagnetic interference (EMI). Both of them should be reduced to obtain better system performance. Several parts of the system such as current sensors and encoder are very sensitive. These parts can be distorted and encounter interference by internal or external noises sources. As a result, the measurements will be incorrect. It makes the system become unstable and causes

vibration of the motor, disturbing or even disconnecting the communication between PC and DSP during the real time operation. Figure 3.16 shows the wiring connection of grounding, shielding and protective earth (PE).
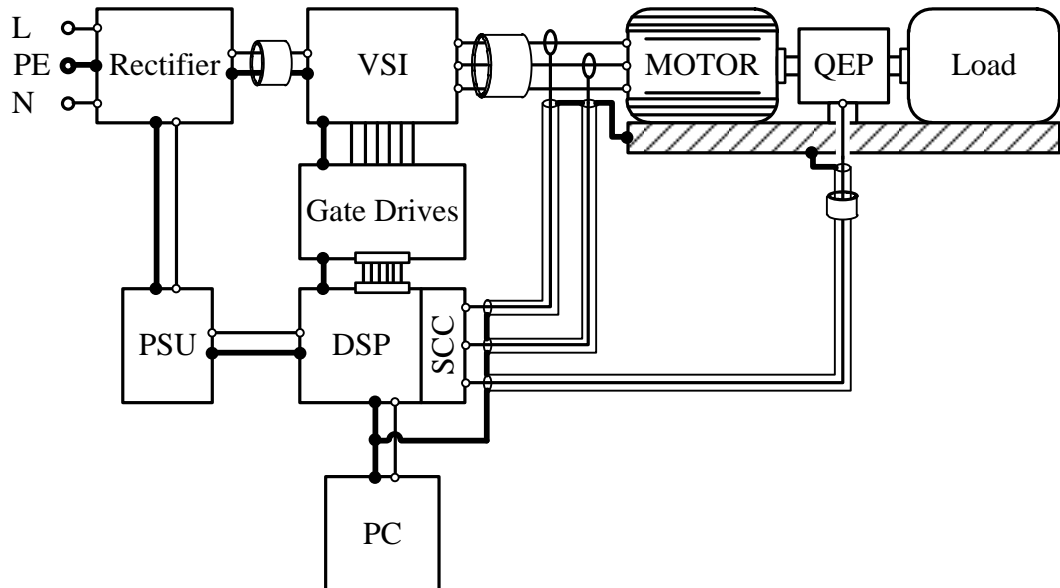
Figure 3.16: Overall system's grounding and shielding diagram

There are two main internal parts of the system which produce noises or glitches. One of the parts which produce noises is inverter. Interference from the inverter is a common issue since it uses high switching frequency for the IGBT command signal. The power cable of the inverter delivers high voltage with high switching frequency to the motor then radiates high frequency glitches to other electronic parts around the power cable (Morrison, 2007). Another part is transformer where the electromagnetic leakage from the transformer can affect the system when it is mounted next to the sensitive electronic circuit.

In order to make the system immune from noises and glitches, there are some considerations to improve the motor drive system operation stability that can be summarized as follows:

- The IGBT signal wire connections have to be as short as possible. In fact, a long wire can act as an antenna that has the ability to collect more radio frequency or electromagnetic interferences in comparison with a short wire.

- Twisted and coaxial cables are more recommended for the feedback signals and the IGBT signals.

- Use snubber circuit across the collector and the emitter terminal of IGBTs.

- For the current sensor, a R-C filter is used as low pass filter and it is put as close as possible to the ADC input terminal of the DSP board.

- The crucial part such as sensor, DSP, SCC is put far away from the electromagnetic sources and protected from electromagnetic exposure by using magnetic material shielding such as metal plate.

- Toroid core can be placed at QEP cable, input and output power line of inverter to suppress EMI and RFI in critical part of the system.

- Make sure the overall system has good wiring connections to avoid arc that can contribute additional noises/glitches to the system.

- Ground and shielding are connected properly together to the protective earth (PE) point.

3.3 Motor Specifications

In this research, two different types of motor are used. The sensored-FOC drive algorithm is tested to run a 920 W Baldor SPMSM (BSM80C-2150). While a 600 W of IPMSM is tested using sensorless-FOC drive algorithm. The operating area of BSM80C-2150, as illustrated in Figure 3.17 and some important motor parameters, as listed in the Table 3.1 have been provided from the datasheet (Baldor, 2003).
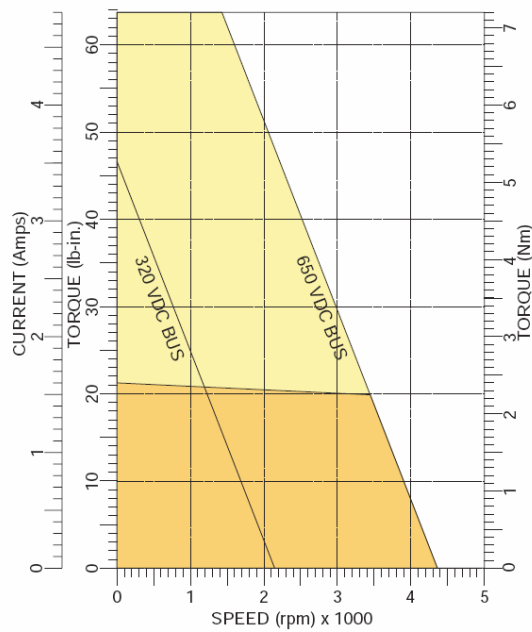
Figure 3.17: Operating area of BSM80C-2150

Table 3.1: Baldor SPMSM Motor specification

Model	: BSM80C-2150
Type of magnet mounting	: Surface
Rated power	: 920 W
Continuous stall torque	: 2.4 Nm
Continuous stall current	: 1.83 Amps
Continuous current	: 2.4 Amps
Rated speed	: 2000 rpm
Number of poles	: 4
Rated DC-bus voltage	: 300 VDC
q-axis inductance	: 40 mH
d-axis inductance	: 40 mH
Resistance	: 14.55 Ω
Voltage constant	: 105.4 $V_{rms\ L-L}$/krpm
Torque constant	: 1.74 Nm/Amp

For the IPMSM, the motor parameters such as dq-axis inductances, stator resistance, and permanent magnet flux linkage are not available from the Daikin website. Because those parameters used in the sensorless algorithm for the stator flux calculation, the IPMSM's parameters has to be determined experimentally, as discussed in several step below.

A. Voltage constant and permanent magnet flux linkage

Voltage constant can be calculated by running the IPMSM as a generator. The motor run at constant speed by using a motor (M) with variable speed drive (VSD) and voltage measurement is taken across the motor terminal. Because stator current is zero, the measured line to line voltage is considered as back-emf voltage. It is proportional to the mechanical angular speed. The measurement is taken under different speed value to get the average value of voltage constant. Figure 3.18 shows the setup diagram and its equivalent circuit.

By using this voltage measurement, the permanent magnet flux linkage can be calculated from equation 2.7 and 2.8. Because, stator current is zero, equation 2.8 becomes:

$$\begin{aligned} \lambda_d &= \lambda_{PM} \\ \lambda_q &= 0 \end{aligned} \tag{3.3}$$

Substitution of equation 3.3 to equation 2.7 yields:

$$\begin{aligned} V_d &= 0 \\ V_q &= \omega_e \lambda_{PM} \end{aligned} \tag{3.4}$$

Because,

$$|v_S| = \sqrt{v_d{}^2 + v_q{}^2} \tag{3.5}$$

Then, the permanent magnet flux linkage becomes:

$$\lambda_{PM} = \frac{|v_S|}{\omega_m \times p} \tag{3.6}$$

Where:

$$|v_S| = \frac{V_{AB}}{\sqrt{3}} \sqrt{2} \tag{3.7}$$
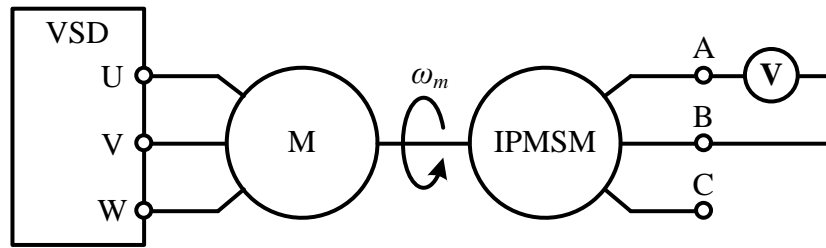
(a)
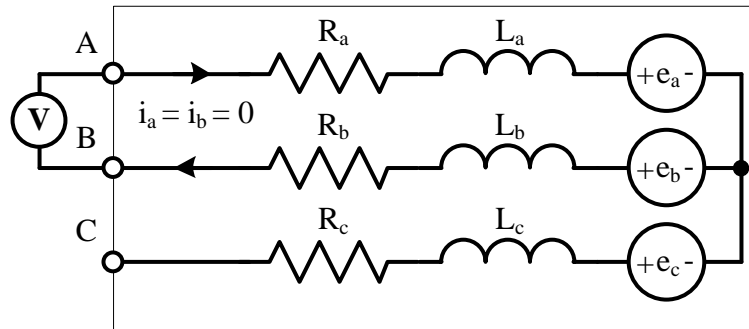
(b)

Figure 3.18: Voltage constant and PM flux measurement
(a) Setup diagram (b) Equivalent circuit

The voltage output of motor is proportional to the speed while the PM flux linkage remains constant as it is obtained from experiments and its result is listed in Table 3.2 below.

Table 3.2: Voltage output and PM flux linkage as function of the rotor speed

Rotor speed (rpm)	Line-line voltage (V_{rms})	PM flux linkage (Wb)
1000	27.24	0.106
2000	54.50	0.106
3000	81.51	0.106

B. Stator resistance

The stator resistance is obtained by using DC current injection. The DC supply is applied across the motor terminal as shown in Figure 3.19 . A voltmeter and ampere meter is needed to measure the voltage and current. Because DC source has zero-frequency, the reactance of stator winding will be zero and the stator resistance can be calculated by using Ohm's law equation and the reading from voltmeter and ampere meter. The calculated resistance is the line-to-line value. For the line-to-neutral value, it should be divided by two. The stator resistance can be directly measured by ohmmeter.
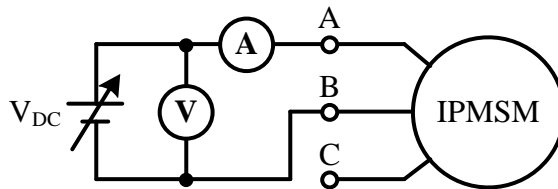
Figure 3.19: Setup for stator resistance measurement

C. dq-axis inductances

The inductance value in PMSM depends on the rotor or magnet position. To measure the motor inductances at different position, a DC power supply and a LCR meter is used. At first, the rotor alignment needs to be done by injecting DC current using DC power supply. For q-axis inductance measurement, the rotor needs to be aligned along the d-axis. Therefore, the B and C point is connected together to the negative terminal of DC power supply while the A point is connected to the positive terminal. The wiring diagram and the flux vector are shown in Figure 3.20. The rotor is locked then the DC power supply is disconnected and the LCR meter is connected across the B and C point as shown in Figure 3.22.
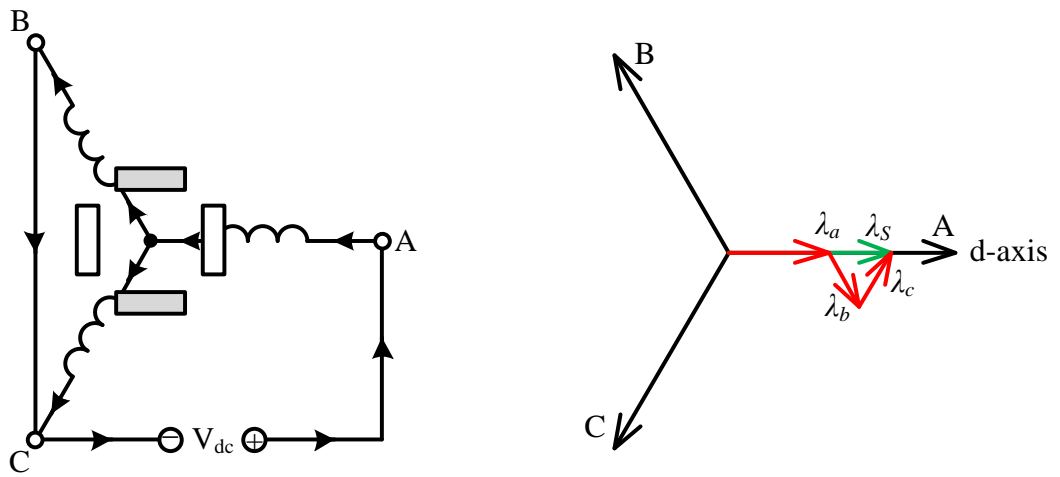
Figure 3.20: d-axis alignment for q-axis inductance measurement
(a) wiring diagram (b) flux vector

For d-axis inductance measurement, the rotor is aligned along the q-axis. The B and C terminal is connected to the positive and negative terminal of power supply respectively, as shown in Figure 3.21. The same procedure in the q-axis inductance measurement is used to extract the value of d-axis inductance by using LCR meter.
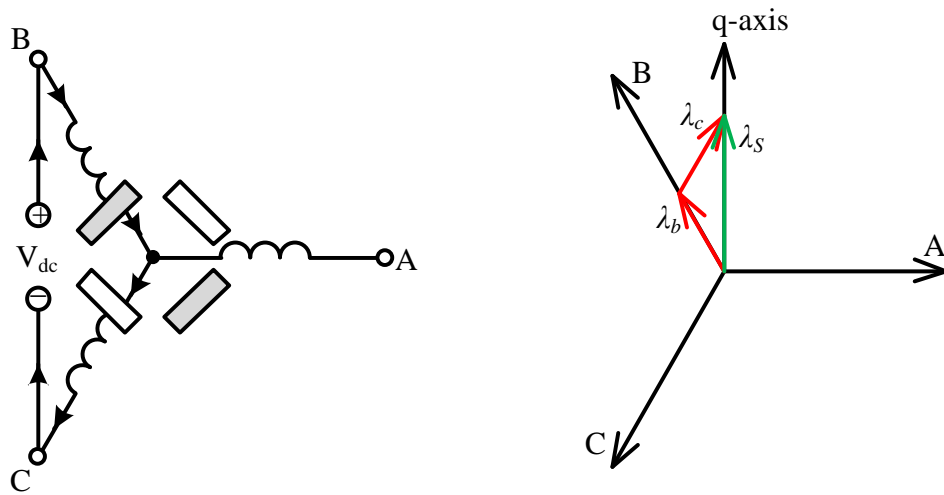
Figure 3.21: q-axis alignment for d-axis inductance measurement
(a) wiring diagram (b) flux vector
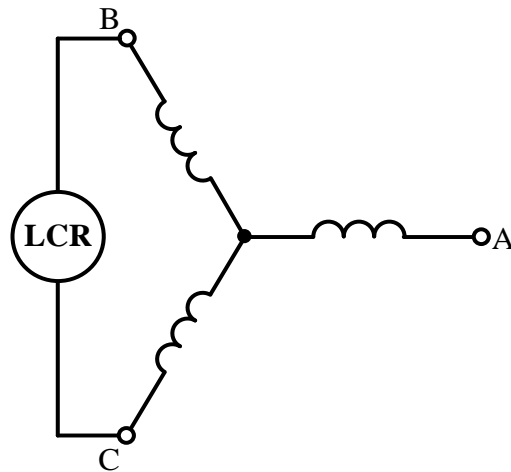
Figure 3.22: LCR meter connection

Overall parameter of IPMSM is summarized in Table 3.3 below:

Table 3.3: IPMSM Motor's specification

Model	: 1YC23
Type of magnet mounting	: Interior
Rated power	: 600 W **
Rated torque	: 2.15 Nm
Rated current	: 4.7 Arms **
Rated speed	: 3000 rpm
Number of poles	: 4
Rated DC-bus voltage	: 300 VDC
q-axis inductance	: 19.12 mH
d-axis inductance	: 12.51 mH
Resistance	: 1.3 Ω
Voltage constant	: 27.24 $V_{rms\ L\text{-}L}$/krpm
PM flux	: 0.106 Wb

Note: ** Technical data is taken from Daikin (2001)

3.4 Hysteresis Brake

For motor loading, a compressed-air hysteresis brake (AHB-3.0) is used to observe the motor dynamic behaviour under loading condition at different load torque input value. AHB-3.0 can handle loading operation up to 3 Nm. The load value of hysteresis brake can be changed by adjusting the value of current from brake control unit; it can be seen in APPENDIX B.

3.5 Software Configuration

Real time calculation is needed to run the vector-control algorithm. Therefore, a high -speed EzDSP TMS320F2812 board is used and it has several features that suit the motor drive applications. The DSP specifications can be summarized as follows:

- 12-bit fixed-point integer ADC output

- 6 capture units

- 16 PWM outputs

- 16 ADC input

Figure 3.23 shows the DSP interfacing of the motor drives system. The phase current signals from the current transducer are analogue signals. These signals have to be injected to the DSP's ADC pin. Then, the DSP converts the analogue signals into digital equivalent values. It is used together with measurement of rotor position to perform reference frame transformation calculation.
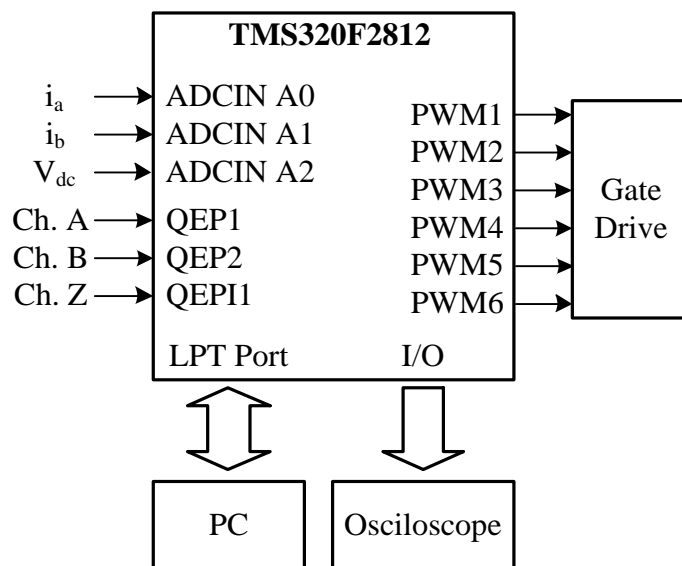
Figure 3.23: DSP interfacing
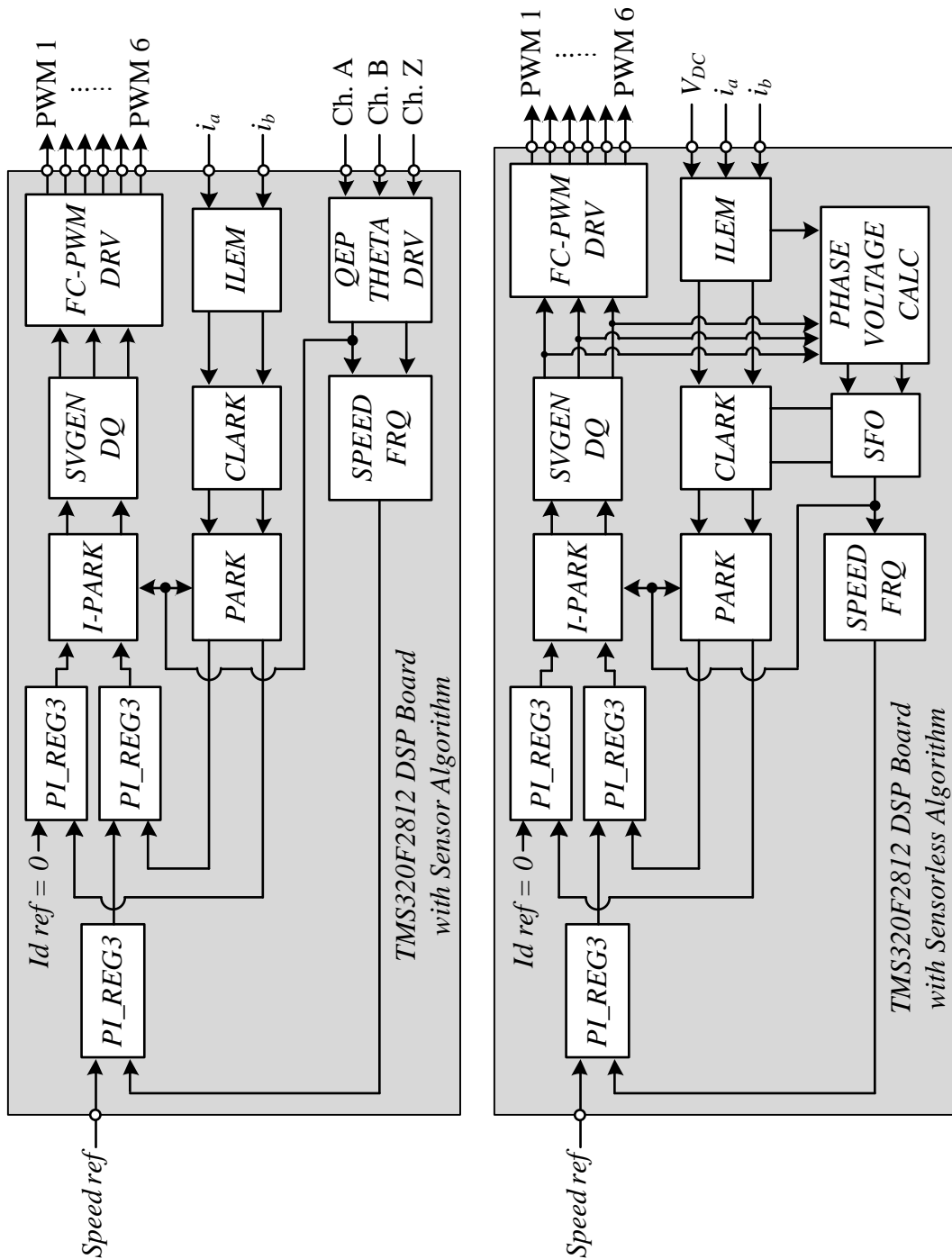
Figure 3.24: Software configurations for sensor and sensorless algorithm

For programming, debugging, compiling and uploading the algorithm into the DSP, a personal computer with Code Composer Studio v3.1, C++ based software, is connected to the DSP through DB25 parallel port. The FOC algorithm uses the Digital Motor control (DMC) library module that is available free and can be downloaded from the Texas Instruments website. Figure 3.24 shows the overall DMC module

arrangement for sensor and sensorless algorithm. The sampling time of inner loop was set to 50 us (20 kHz), equal to the carrier frequency of the PWM. The outer loop's time sampling is ten times slower than the inner loop's time sampling.

CHAPTER 4

EXPERIMENTAL RESULT AND DISCUSSION

4.1 Introduction

The experimental setup diagram, hardware and software configuration that has been discussed in the previous chapter is integrated and implemented for the motor drive system in real time mode. The system is tested and verified experimentally. This chapter presents the experiment result and its discussion. The results were obtained by applying various speed and torque step response in order to observe the speed and current controller effectiveness during transient and steady state condition. The motor is run with initial predetermined rotor position to avoid temporary rotor reversal and limit the current absorption during the start-up period.

For capturing the waveform and measurement of the time response that happened inside the DSP, a digital-to-analogue subroutine was inserted into the main source code in DSP's program algorithm and the output was obtained at PWM 7-9 pin. Each pin is applied to three gate drivers in order to increase the signal amplitude. The outputs of gate drivers have to be filtered by using a simple low-pass-filter circuit. The output of low-pass-filter is an analogue signal and it is connected to the Tektronix oscilloscope TDS2024B. A power quality analyzer FLUKE 43B is used for power consumption measurement.

4.1.1 SPMSM Drive Performance

The experimental setup diagram for SPMSM refers to Figure 3.1. Figure 4.1 shows the line-line voltage of the inverter output at 300 VDC of DC-link voltage in a steady state condition. During system operation, the auto-transformer output voltage is

increased gradually up to 300 V of DC-link is obtained across the single-phase rectifier's output terminal. This step is useful to limit the inrush-current during the charging time of capacitor in the start-up period.
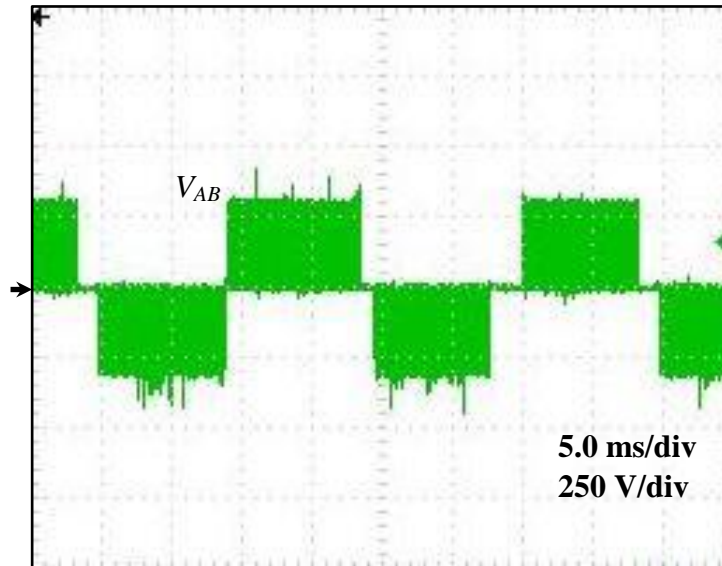
Figure 4.1: Line to line output voltage of three-phase inverter

4.1.1.1 Speed Step-response

This experiment was conducted for the performance observation of the speed and current controller when the step response of speed reference is applied in the motor drive system. The motor drive system is run at loading condition, and the speed reference is changed by keying in the new value of speed reference value in the "watch window" of code composer studio software.

In Figure 4.2 (a), the motor drive system is loaded at 0.5 Nm then the motor is accelerated from standstill to 1200 rpm of speed. During the acceleration, the actual speed increases gradually with 400 rpm overshoot in 200 ms. The steady state condition where the actual speed follows the speed reference value is achieved after 450 ms. The same response is also seen in the phase stator current profile. For deceleration, the motor is decelerated suddenly to standstill. The motor completely stopped after 100 ms without undershoot, as shown in Figure 4.2 (b).

When the actual speed keeps changing as a function of the time, the q-axis current that is proportional to the electromagnetic torque keeps increasing until the speed becomes constant, as shown in Figure 4.3 (a). This can be attributed to the inertia effect of the motor. The q-axis current achieves the steady state condition in 450 ms. In deceleration condition, as shown in Figure 4.3 (b), the q-axis current returns to the 0 Nm of torque after 100 ms.
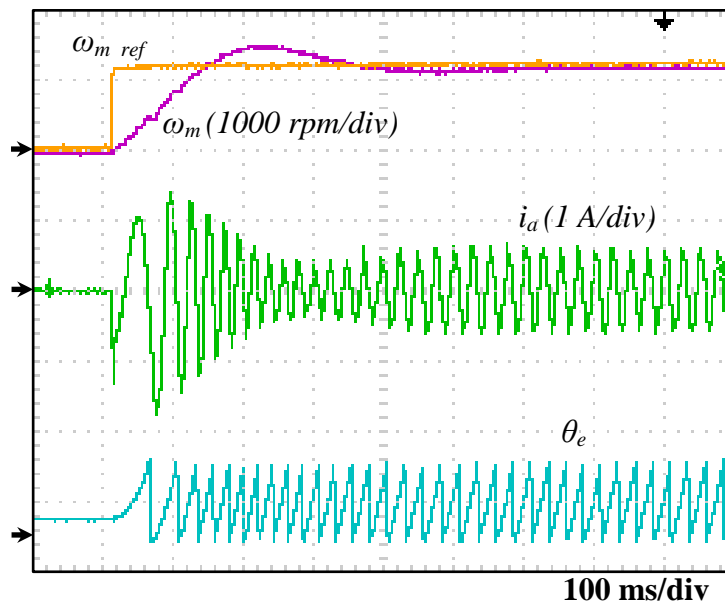
(a)
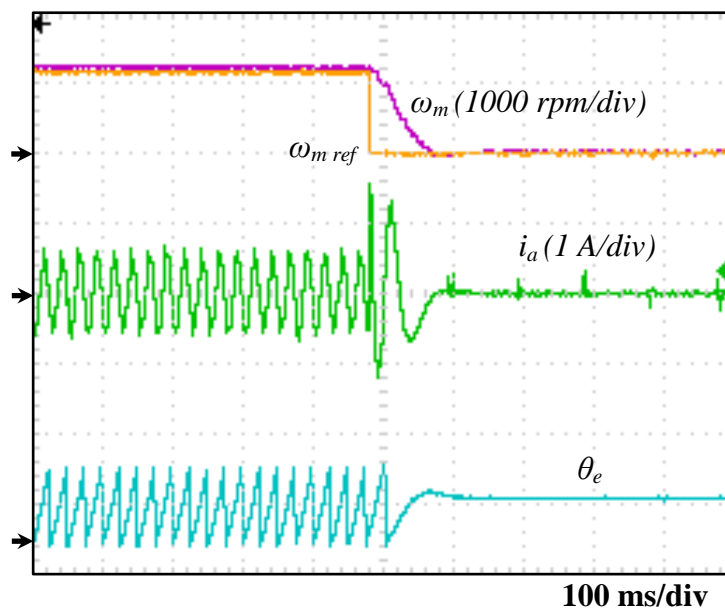
(b)

Figure 4.2: Speed controller action 0 rpm – 1200 rpm at $T_L = 0.5$ Nm
(a) Acceleration (b) Deceleration
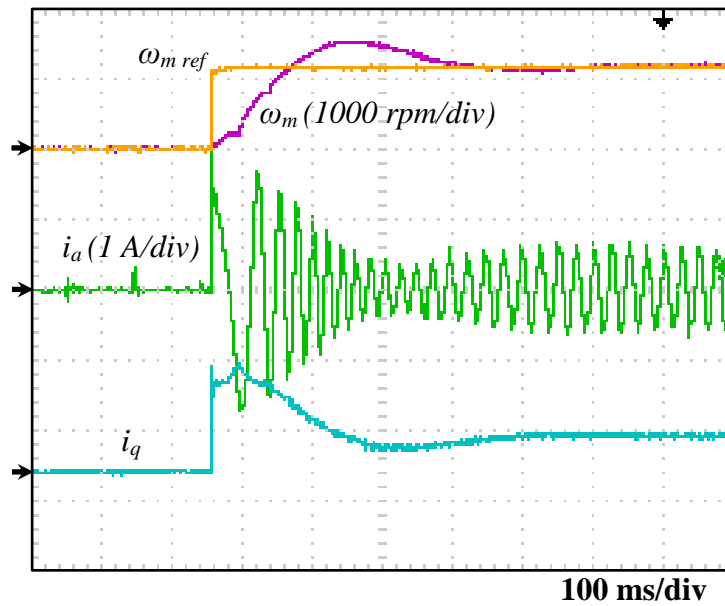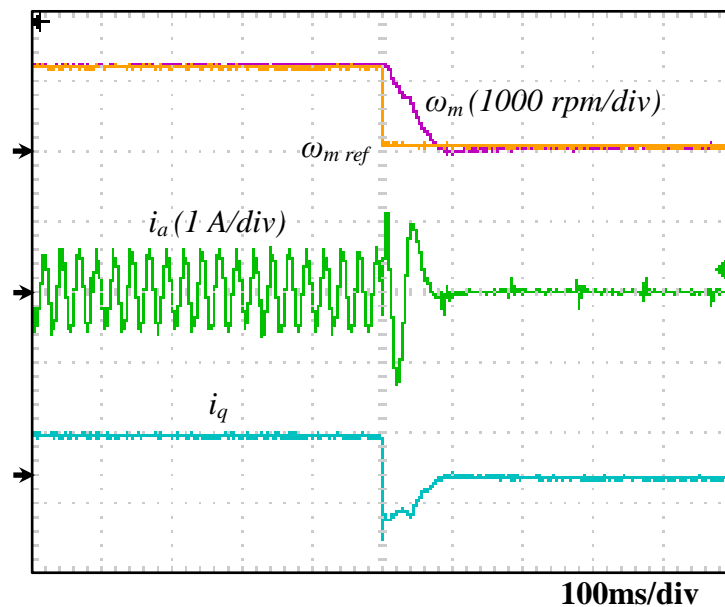
Figure 4.3: Current controller action 0 rpm – 1200 rpm at $T_L = 0.5$ Nm
(a) Acceleration (b) deceleration

Another speed response variation is also taken for acceleration and deceleration as shown in Figure 4.4 (a) and (b), respectively. Figure 4.4 (a) shows the motor is initially running at 600 rpm after which it is accelerated to 1200 rpm at 0.5 Nm of load whereas its opposite is shown in Figure 4.4 (b). Both conditions have the same time response duration. The overshoot of the actual speed is 200 rpm in 250 ms. The transient condition occurs only in 300 ms. At steady state condition, the amplitude of stator current is slightly bigger when the motor is run at 1200 rpm. This can be

happened as the effect of the viscosity component of electromagnetic torque (equation 2.13). The viscosity component's value increases when the speed is increased.
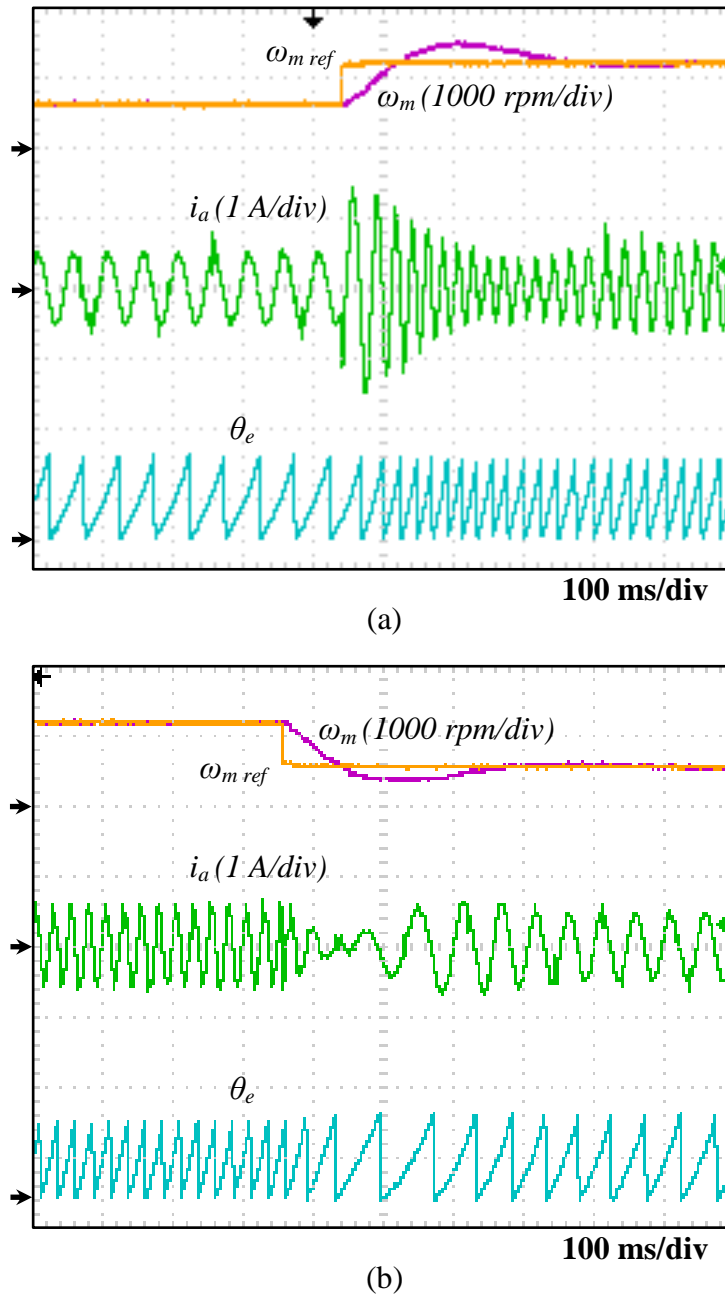
Figure 4.4: Speed controller action 600 rpm – 1200 rpm at $T_L = 0.5$ Nm
(a) Acceleration (b) deceleration

Besides the effect of PI setting, the overshoot of stator current and q-axis current is also affected by the deviation of speed. From Figure 4.3 and Figure 4.5, it can be seen that the current overshoot in Figure 4.3(a) is bigger than in Figure 4.5(a). The inertia component of electromagnetic torque equation increases when the deviation of speed

increases. For deceleration, the q-axis drops to a negative value because the motor acts as a generator in a short period of time before the motor back to the normal condition.
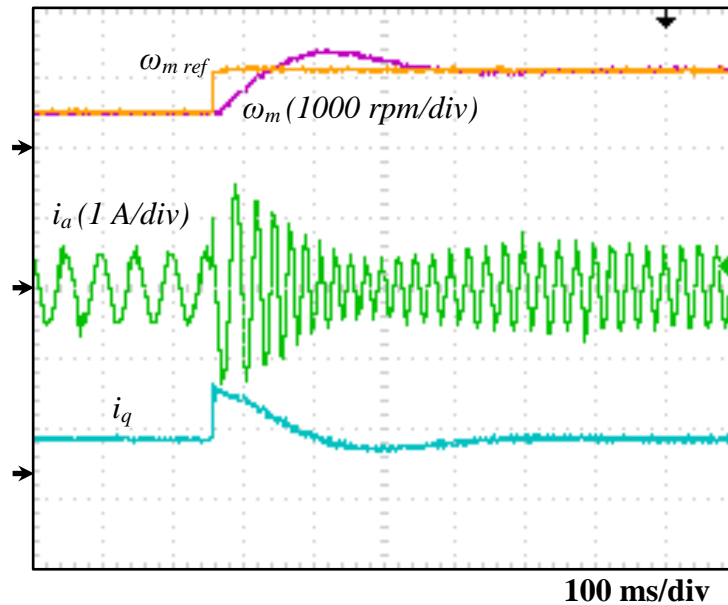
(a)
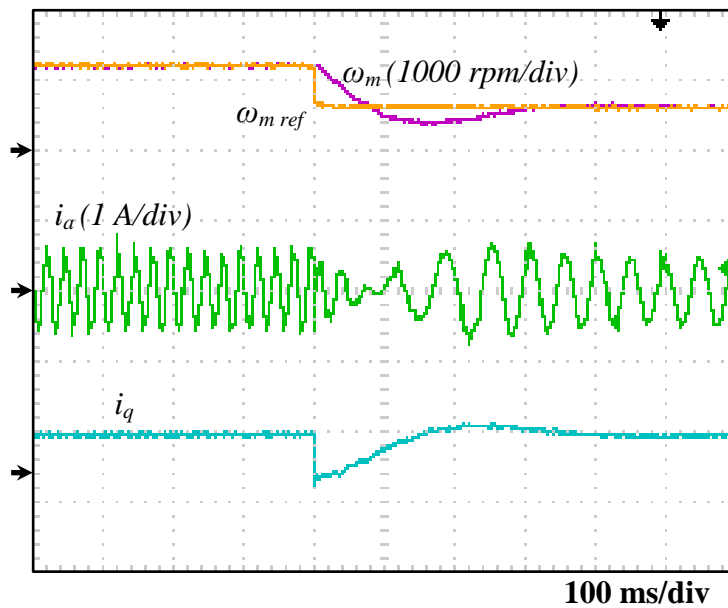
(b)

Figure 4.5: Current controller action 600 rpm – 1200 rpm at T_L = 0.5 Nm
(a) Acceleration (b) deceleration

The observation is also taken when the reverse speed step response is applied to the motor drive system. The motor is run at -900 rpm of speed with 0.5 Nm of load torque. Then instantaneously, it is changed to +900 rpm in a short period of time. The reverse speed step response is performed to observe how fast the motor change the

direction from clockwise to the counter clockwise direction. As shown in Figure 4.6 and Figure 4.7, the motor takes 450 ms to change the rotor rotation direction.
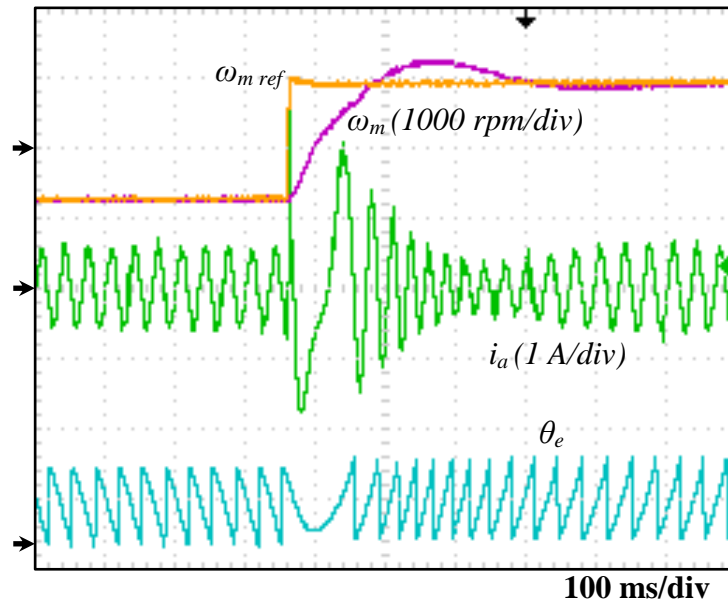
(a)
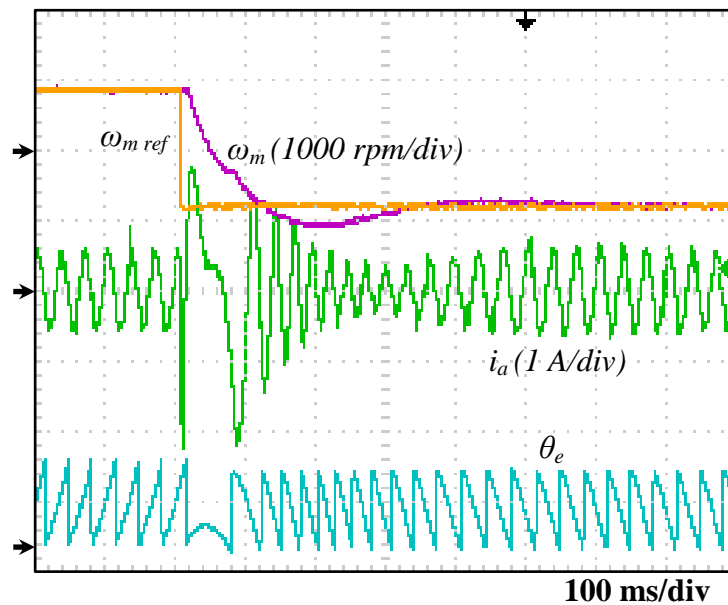
(b)

Figure 4.6: Speed controller action during speed reversal at T_L = 0.5 Nm
(a) -900 rpm to 900 rpm (b) 900 rpm to -900 rpm
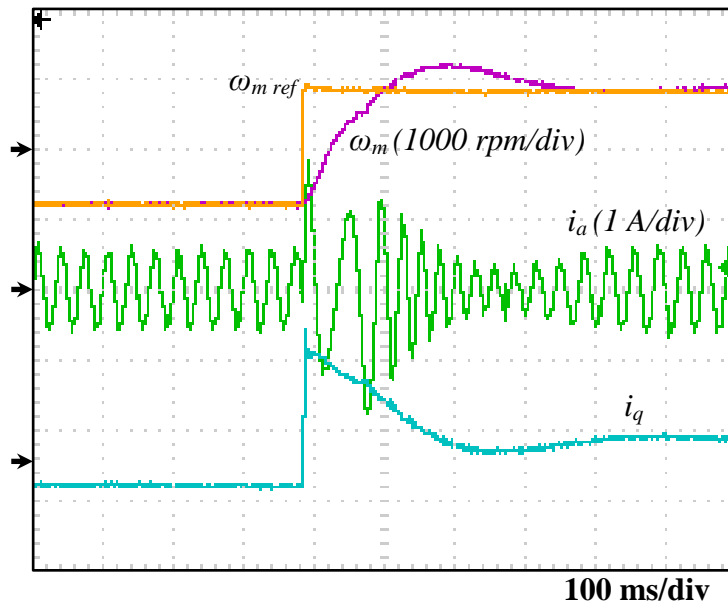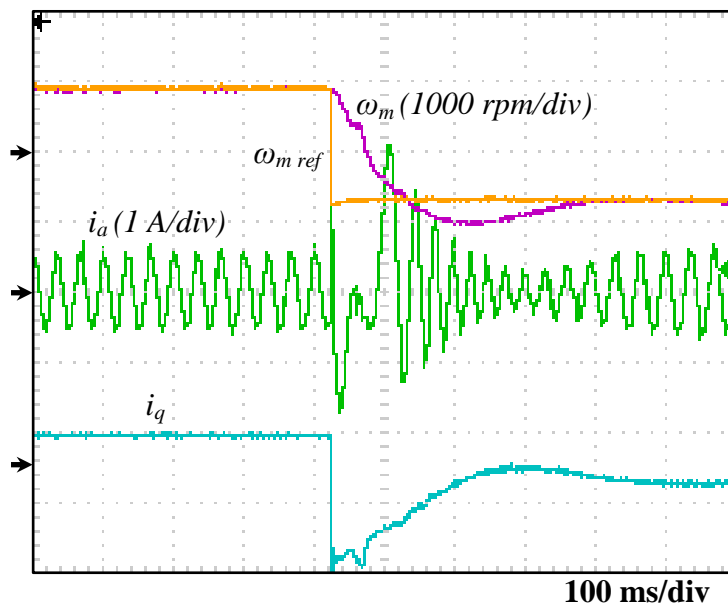
Figure 4.7: Current controller action during speed reversal at $T_L = 0.5$ Nm
(a) -900 rpm to 900 rpm (b) 900 rpm to -900 rpm

The phase stator current has the same amplitude for negative and positive direction that means the motor consumes the same current value in both directions. However, the q-axis current that represents the electromagnetic torque has different value as shown in Figure 4.7. The q-axis current and torque has negative value when the speed is negative. This condition indicates that the motor is driven in the motoring mode

with counter clockwise or reverse direction (quadrant 3). While in the motoring mode with clockwise direction (quadrant 1), the speed and torque have positive value.

4.1.1.2 Torque Step-response

The robustness of any high-performance motor drive system can be verified from the load torque step-response for loading and unloading condition. The load torque step-response is also useful in order to verify the effectiveness of current controller to reduce the error between the current reference and current feedback. A high-performance motor drive system can sustain the rotor speeds despite a sudden change of load. In order to perform this action, a compressed-air hysteresis magnetic brake is coupled to the PMSM, while the brake power supply unit is used to change the load value. The motor is running at constant speed reference value.

The observation of the measured speed, phase-a stator current and the dq-axis current is taken at 1000 rpm of speed reference 1 Nm and 2 Nm of load. Initially, the motor is operated with no-load condition at 1000 rpm and then the load is increased to a certain value by increasing the load value setting on the hysteresis brake controller. To perform the unloading condition action, the brake controller is switched off without decreasing the load value; whereas the brake controller is switched on to perform loading condition action.

Figure 4.8 shows the response of the system at 1 Nm of load. From the figure, it can be seen that the stator current and also the q-axis current start to increase gradually from zero during loading condition then maintain at a fixed value after a few seconds in the steady state condition. Undershoot of the measured speed occurs in 0.45 s. The measured speed of the motor slightly decreases, 300 rpm of the speed reference, before it is stable and follows the speed reference.
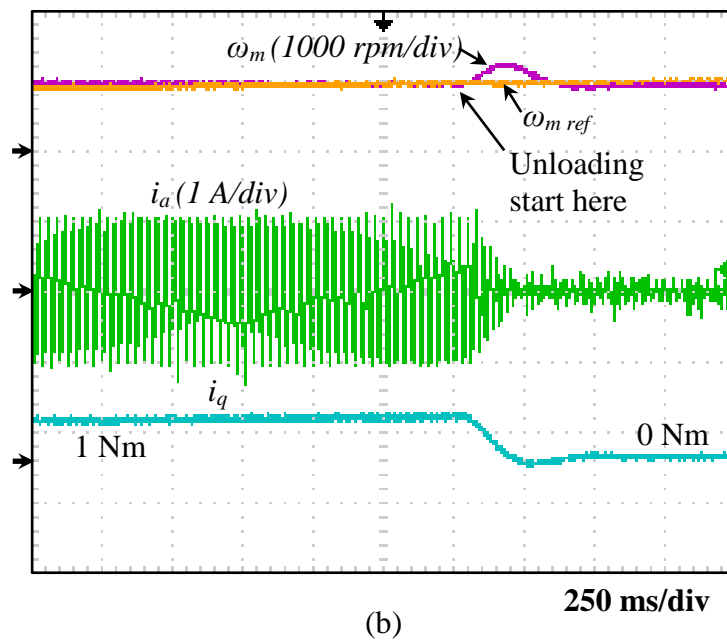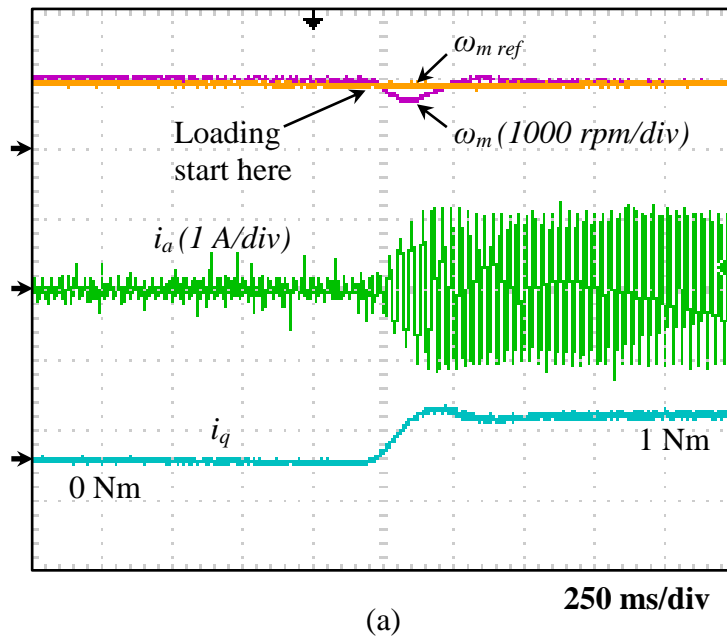
Figure 4.8: Speed controller action during torque step-response 0-1 Nm at 1000 rpm
(a) Loading (b) Unloading
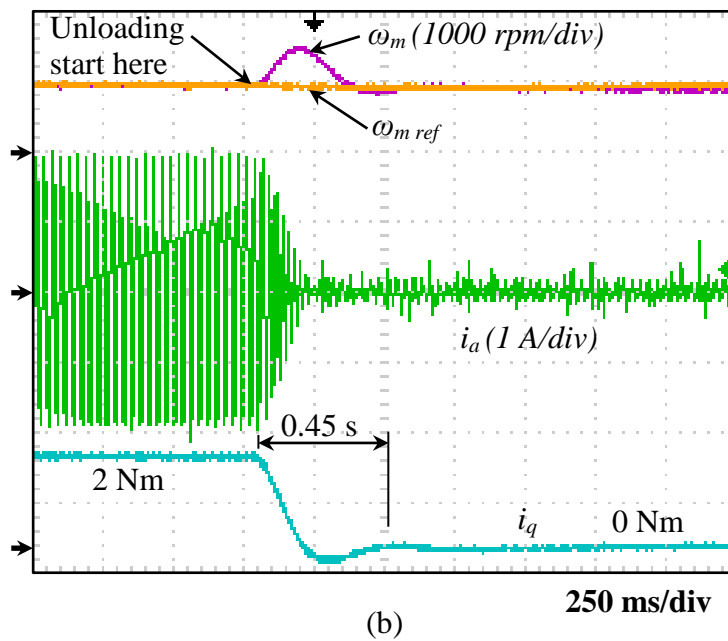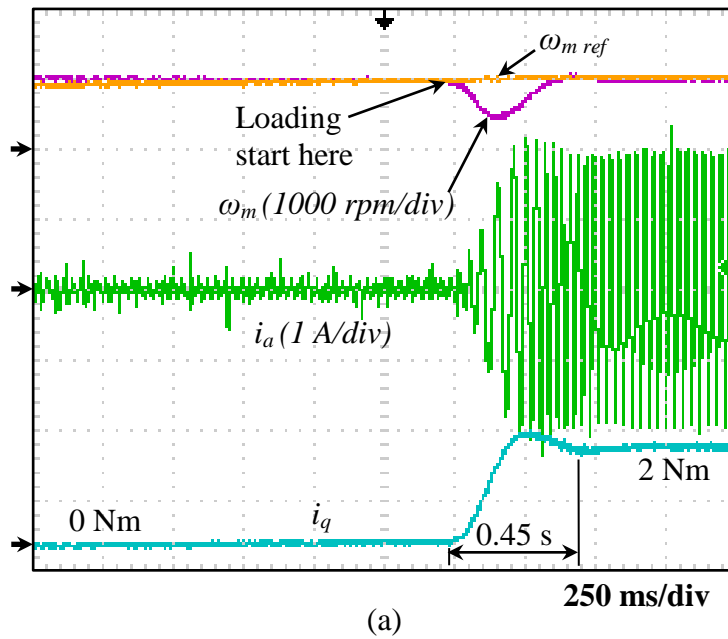
Figure 4.9: Speed controller action during torque step-response 0-2 Nm at 1000 rpm
(a) Loading (b) Unloading

In Figure 4.9, the torque step response 0-2 Nm is applied at the same speed, 1000 rpm. From signal observation, the actual speed dips to 500 rpm in 0.45 s then the actual speed is back to the normal reference speed. The q-axis current and phase stator current increase twice as much as in the previous torque step response. This indicates that the torque is proportional to the stator current.
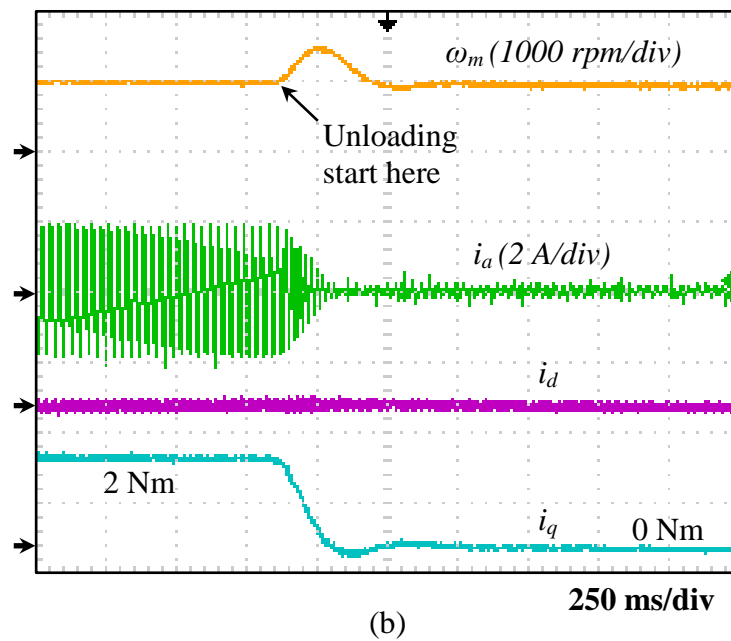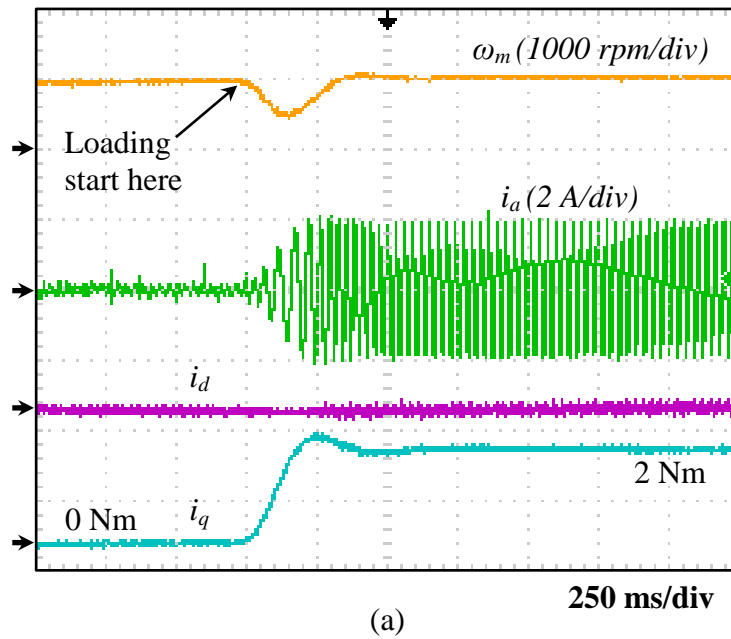
Figure 4.10: dq-axis current controller action during torque step-response 0-2 Nm
at 1000 rpm
(a) Loading (b) Unloading

Figure 4.10 shows the current controller action during torque step response from
0 to 2 Nm at 1000 rpm of speed. From the figure, it can be seen the d-axis current is still
maintained at zero while q-axis current increases during loading until the steady state
condition is achieved and decreases when the load is removed. Figure 4.11 shows the
power input of inverter measurement at 1000 rpm of speed and 2 Nm of load.
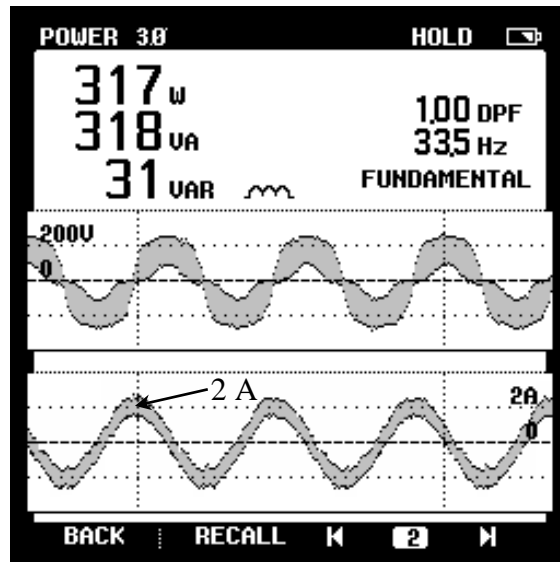
Figure 4.11: Stator voltage, current and electrical power input at 1000 rpm, 2 Nm

By assuming the electromagnetic torque is equal to the load torque, the mechanical power of the motor is calculated equation 2.15:

$$P_{mech} = 2\ Nm \times \left(\frac{2\pi}{60} 1000\ rpm \right)\ rad/s = 209.44\ W$$

Then using equation 2.16, the efficiency of motor is:

$$\eta = \frac{209.44\ W}{317\ W} \times 100\% = 66.07\%$$

4.1.2 IPMSM Drive Performance

The dynamic performance of IPMSM drive system is tested by applying a step change in the speed reference. The experimental setup diagram for IPMSM refers to Figure 3.2. The motor drive system with predetermined initial rotor position is accelerated from standstill to 1200 rpm at 0.5 Nm loading condition, as shown in Figure 4.12 (a). The system reaches the steady state condition in 1.5 s. While running at 1200 rpm, the motor is decelerated to standstill condition Figure 4.12 (b). The motor is completely stopped in 400 ms.

(a)

(b)
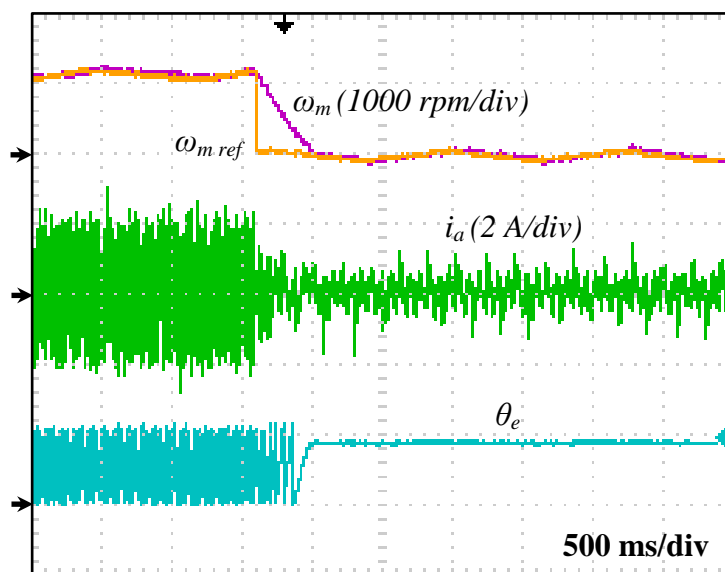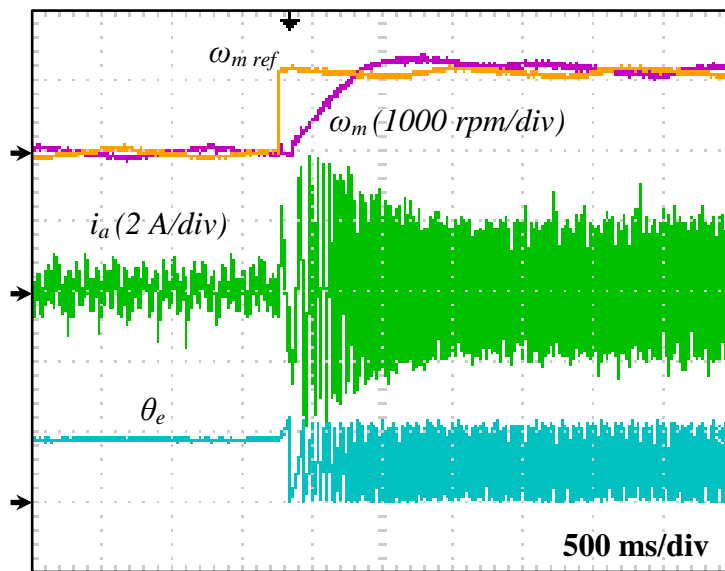
Figure 4.12: Speed step response 0 rpm − 1200 rpm at $T_L = 0.5$ Nm for IPMSM
(a) Acceleration (b) Deceleration

The drive condition where the motor is accelerated from 450 rpm to 1500 rpm at 50% of rated torque is shown in Figure 4.13 (a) whereas Figure 4.13 (b) shows the motor is decelerated from 1500 rpm to 450 rpm at the same load condition.
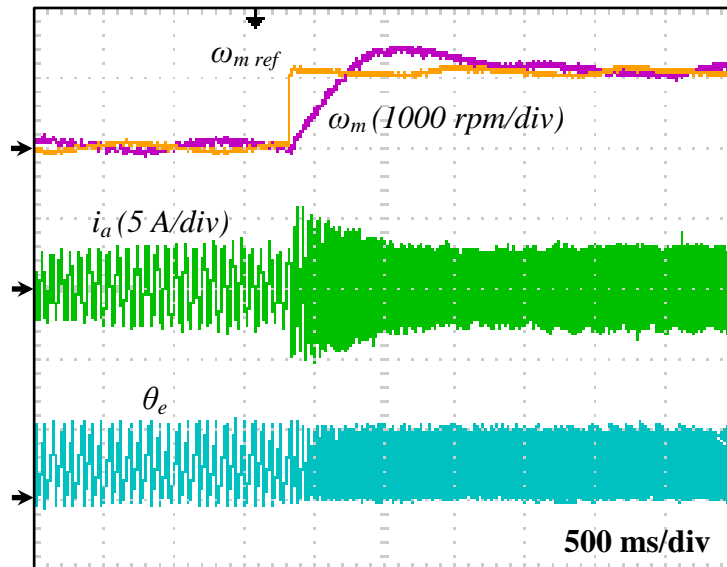
(a)
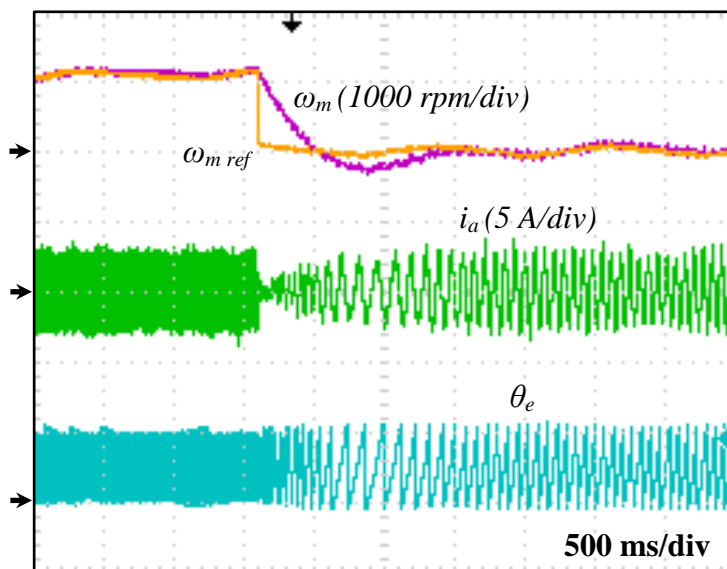
(b)

Figure 4.13: Speed step response 450 rpm – 1500 rpm at $T_L = 1$ Nm for IPMSM
(a) Acceleration (b) Deceleration

Figure 4.12 and Figure 4.13 indicate that the estimated speed follows the speed reference. It means that the SFO is working properly to estimate the speed as well as the

rotor position. The oscillation could be happening due to the improper value of the PI controller and the effect of the time sampling setting in oscilloscope.

At steady state condition, the estimated rotor position is exactly aligned to the stator phase current, as shown in Figure 4.14. Misalignment and inaccurate estimation of the rotor position would cause the motor current to draw more current than its rated value.
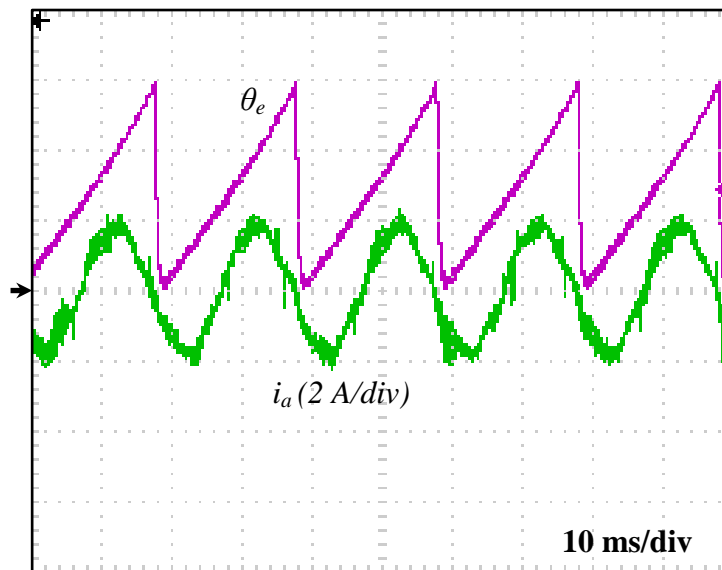
Figure 4.14: Steady state condition at 1500 rpm, 0.5 Nm.
3 - Estimated rotor position, 4 - stator phase current

In this IPMSM drive system, the estimated stator flux is only utilized for rotor position and speed calculation. The estimated stator flux and its trajectory were captured with the help of the data logger and the graph tool from code composer studio software in PC.

Figure 4.15 shows the estimated $\alpha\beta$-axis stator flux when the motor run at 1500 rpm in the steady state condition. The α-axis and β-axis stator flux has 90 degrees of angle different each other. The amplitude of estimated $\alpha\beta$-axis stator flux is 0.11 Wb. It is close to the actual permanent magnet flux value (0.106 Wb), as obtained in parameter determination of IPMSM. The estimated stator flux trajectory shape at steady state

condition is shown in Figure 4.16. The flux trajectory has uniform radius (r) and creates a circle shape with 0.11 of radius that represent the peak value of estimated flux linkages. The DC drift of current sensor is successfully eliminated by the SFO and make the trajectory stable at the centre point in the steady state condition.
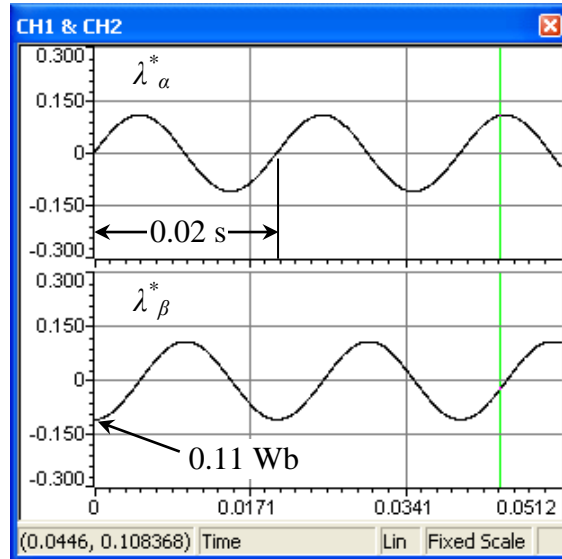
Figure 4.15: Estimated stator flux at 1500 rpm
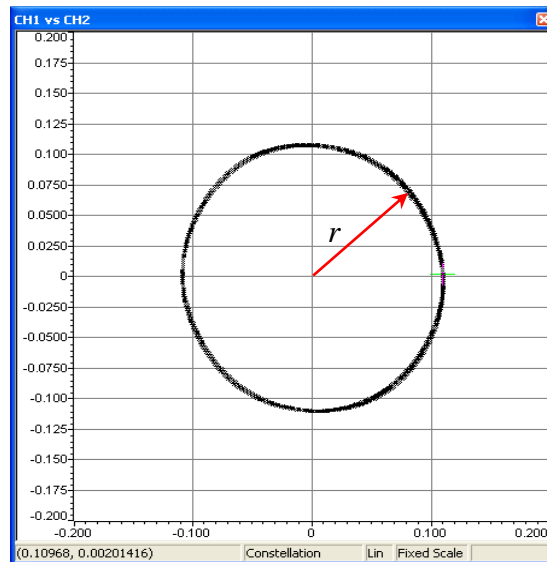
Figure 4.16: Stator flux trajectory at 1500 rpm

Figure 4.17 shown the power input consumption measurement of the inverter when the IPMSM run at 2550 rpm, 0.9 Nm. The calculated mechanical power output is,

$$P_{mech} = 0.9\,Nm \times \left(\frac{2\pi}{60} 2550 \right)\ rad/s = 240.33\,W$$

And efficiency is:

$$\eta = \frac{240.33\,W}{252\,W} \times 100\% = 95.40\%$$
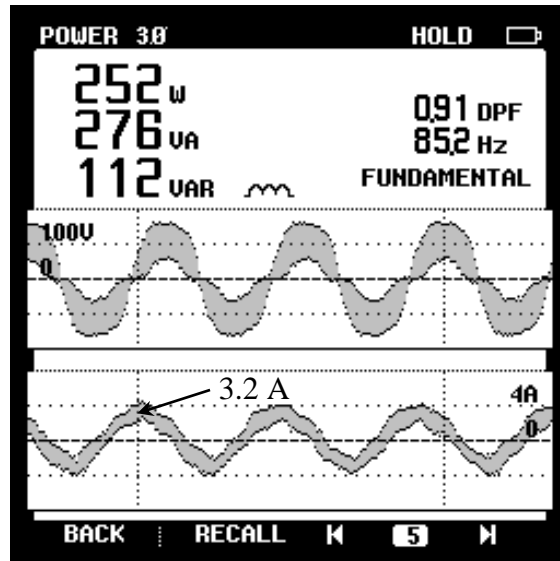
Figure 4.17: Stator voltage, current and electrical power input at 2550 rpm, 0.9 Nm

4.2 Summary

The experimental results of the motor drive system and its discussion have been presented in this chapter. All the experiment result for PMSM and IPMSM drive system is taken under closed-loop FOC operating condition. The rotor position is pre-determined and aligned manually before the DC-link is applied to the inverter.

In SPMSM drive system, the effectiveness of speed and current controller has been tested when the speed reference is changing and the load interruption appears instantaneously (loading or unloading). While in IPMSM drive system, the load interruption test cannot be performed to verify the effectiveness of speed and current controller. For the speed step response in sensored FOC scheme, the speed can be

controlled accurately and faster than in the sensorless FOC scheme. The sensorless FOC scheme needs additional time for speed calculation in order to respond to the speed change. Nevertheless, the proposed SFO in IPMSM sensorless drive system is working properly to eliminate the DC drift that appears on the current sensor output and the estimated flux trajectory is stable in circle shape with uniform radius at steady state condition. From the experiment, 95% of efficiency can be achieved in IPMSM while in SPMSM the efficiency is only 66.1%.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

In this research, the DSP-based PMSM drive system with sensor and sensorless FOC control algorithm has been developed successfully. The motor drive system has good dynamic performance and accurate speed tracking. The inverter runs continuously even when the motor is in light-load condition. This is suitable for variable-speed motor drive system especially in compressor and fan control system for an air-conditioner. Moreover, the motor drive system facilitates energy saving for compressor and fan control system by consuming less energy with better performance and response.

The proposed SFO is simpler than the previous SFO. This simplified SFO requires no extended rotor flux in the estimation of the motor rotor position. The implementation of simplified SFO for speed sensorless FOC of IPMSM has been performed successfully in TMS320F2812 DSP platform. The effectiveness of the proposed SFO in drive system has been tested and verified through experiments. But, the system has a problem when the motor run at particular speed in no-load condition then suddenly the motor is loaded.

In heating, ventilation and air conditioning (HVAC) system, the compressor is not only rotating in the positive direction but also in negative or reverse direction. The reverse speed is performed to change the refrigerant gas flow for heating purpose. Unlike the HVAC, the function of compressor in an air-conditioner is compressing the refrigerant gas in one direction only. Hence, the reverse speed becomes not necessary to perform in an air-conditioner's compressor operation.

5.2 Future Work

Both systems used conventional PI controller where the overshoot is still high. It can be reduced by using fuzzy logic controller. In sensorless drives, the proposed SFO without extended rotor flux calculation is working properly for speed step response. The torque step response cannot be performed due to the non-linearity of the motor and the power switches (IGBTs) in inverter. It affects the stator flux linkage estimation and further an error appears on the torque estimation. Therefore, the system needs to be modelled accurately, possibly by using a DTC control scheme for better dynamic performance. In IPMSM drive system, the electromagnetic torque is not optimal since the system uses CTA control modes. MTPA in IPMSM drive is recommended to utilize the reluctance torque as additional torque. To avoid high starting current for the first time running of the motor, a start-up algorithm can be added to determine the rotor position at standstill condition.

REFERENCES

Acarnley, P. P., & Watson, J. F. (2006). Review of position-sensorless operation of brushless permanent-magnet machines. *IEEE Transactions on Industrial Electronics, 53*(2), 352-362.

Baldor. (2003). Baldor AC Servo Motors and Servo Rated Gearheads. from http://www.baldor.com/

Blaschke, F. (1972). The principle of field orientation as applied to the new transvector closed-loop control system for rotating field machines. Siemen's Review, 34, 217-221.

Bose, Bimal K. (2002). *Modern Power Electronics And AC Drives*: Prentice Hall.

Brod, D. M., & Novotny, D. W. (1985). Current Control of VSI-PWM Inverters. IEEE Transactions on Industry Applications, IA-21(3), 562-570.

Daikin. (2001). Split Type Air Conditioners. from http://www.daikin.com.my/

Duesterhoeft, W. C., Schulz, Max W., & Clarke, Edith. (1951). Determination of Instantaneous Currents and Voltages by Means of Alpha, Beta, and Zero Components. *Transactions of the American Institute of Electrical Engineers, 70*(2), 1248-1255. doi: 10.1109/t-aiee.1951.5060554

Fitzgerald A. E., Kingsley C., Umans S. D. (2003). *Electric Machinery*: McGraw-Hill.

Foo, G., & Rahman, M. F. (2010). Sensorless Direct Torque and Flux-Controlled IPM Synchronous Motor Drive at Very Low Speed Without Signal Injection. *IEEE Transactions on Industrial Electronics, 57*(1), 395-403.

Frus, J. R. (1976). Closed-loop control of step motors using waveform-detection feedback: University of Illinois.

Gilsu, Lee, Wook-Jin, Lee, Junho, Ahn, & Dalho, Cheong. (2011). *A simple sensorless algorithm for an interior permanent magnet synchronous motor using a flux observer.* Paper presented at the IEEE 8th International Conference on Power Electronics.

J. Soleimani, A. Vahedi, A. Ejlali. (2011). New Structures of IPM Machine for HEV Traction: Investigation on Field Weakening Performance and Harmonic Analyzing. *IREE, 6*(5).

Jahns, T. M., Kliman, G. B., & Neumann, T. W. (1986). Interior Permanent-Magnet Synchronous Motors for Adjustable-Speed Drives. Industry Applications, IEEE Transactions on, IA-22(4), 738-747.

Jouve, D., Rognon, J. P., & Roye, D. (1990, 11-16 Mar 1990). Effective current and speed controllers for permanent magnet machines: a survey. Paper presented at the Applied Power Electronics Conference and Exposition, 1990. APEC '90, Conference Proceedings 1990., Fifth Annual.

Jungreis, A. M., & Kelley, A. W. (1995). Adjustable speed drive for residential applications. Industry Applications, IEEE Transactions on, 31(6), 1315-1322.

Kang, Jun. (2010). Sensorless Control of Permanent Magnet Motors. In Y. E. America (Ed.).

Kazmierkowski, M. P., & Malesani, L. (1998). Current control techniques for three-phase voltage-source PWM converters: a survey. Industrial Electronics, IEEE Transactions on, 45(5), 691-703.

Konghirun, M. (2005). *Automatic Offset Calibration of Quadrature Encoder Pulse Sensor for Vector Controlled Drive of Permanent Magnet Synchronous Motors.* Paper presented at the TENCON 2005.

Konghirun, M. (2008). *Effects and detection of misaligned position sensor in AC servo drive system with filtered current delay.* Paper presented at the International Conference on Electrical Machines and Systems (ICEMS 2008).

Krishnan, R. (2001). *Electric Motor Drive, Modeling Analysis and Control* Prentice Hall.

Matsui, N. (1996). Sensorless PM brushless DC motor drives. *IEEE Transactions on Industrial Electronics, 43*(2), 300-308. doi: 10.1109/41.491354

Melfi, M. J., Rogers, S. D., Evon, S., & Martin, B. (2008). Permanent-Magnet Motors for Energy Savings in Industrial Applications. *IEEE Transactions on Industry Applications, 44*(5), 1360-1366.

Microchip. (2009). AN1078 Tuning Procedure ReadMe.

Monajemy, R., & Krishnan, R. (2001). Control and dynamics of constant-power-loss-based operation of permanent-magnet synchronous motor drive system. *IEEE Transactions on Industrial Electronics, 48*(4), 839-844.

Morimoto, S., Takeda, Y., & Hirasa, T. (1990). Current phase control methods for permanent magnet synchronous motors. *IEEE Transactions on Power Electronics, 5*(2), 133-139.

Morrison, Ralph. (2007). *Grounding and Shielding Circuits and Interference 5th Edition* (5 ed.): John Wiley & Sons Inc.

Murray, Aengus. (2007). Design Platforms for Energy Efficient Variable-Speed Motors. *EE Times-India*.

Neacsu, Dorin O. (2001). *Space Vector Modulation –An Introduction*. Paper presented at the IECON 2001.

Paicu, M. C., Boldea, I., Andreescu, G. D., & Blaabjerg, F. (2009). Very low speed performance of active flux based sensorless control: interior permanent magnet synchronous motor vector control versus direct torque and flux control. *Electric Power Applications, IET, 3*(6), 551-561.

Park, R. H. (1929). Two-reaction theory of synchronous machines generalized method of analysis-part I. American Institute of Electrical Engineers, Transactions of the, 48(3), 716-727.

Pillay, P., & Freere, P. (1989, 1-5 Oct. 1989). *Literature survey of permanent magnet AC motors and drives.* Paper presented at the Industry Applications Society Annual Meeting, 1989., Conference Record of the 1989 IEEE.

Pillay, P., & Krishnan, R. (1988). Modeling of permanent magnet motor drives. *IEEE Transactions on Industrial Electronics, 35*(4), 537-541.

Pillay, P., & Krishnan, R. (1989). Modeling, simulation, and analysis of permanent-magnet motor drives. I. The permanent-magnet synchronous motor drive. *IEEE Transactions on Industry Applications, 25*(2), 265-273.

Pillay, P., & Krishnan, R. (1991). Application characteristics of permanent magnet synchronous and brushless DC motors for servo drives. *IEEE Transactions on Industry Applications, 27*(5), 986-996. doi: 10.1109/28.90357

Singh, Bhim, Singh, B P, & Kumar, M. (2003). PFC Converter fed PMBLDC Motor Drive for Air Conditioning. *IE(I) Journal-EL, 84*.

Slemon, G. R. (1993, 15-19 Nov 1993). *High-efficiency drives using permanent-magnet motors.* Paper presented at the International Conference on Industrial Electronics, Control, and Instrumentation, 1993. Proceedings of the IECON '93.

Uddin, M. N., Radwan, T. S., George, G. H., & Rahman, M. A. (2000). Performance of current controllers for VSI-fed IPMSM drive. IEEE Transactions on Industry Applications, 36(6), 1531-1538.

van der Broeck, H. W., Skudelny, H. C., & Stanke, G. V. (1988). Analysis and realization of a pulsewidth modulator based on voltage space vectors. *IEEE Transactions on Industry Applications, 24*(1), 142-150. doi: 10.1109/28.87265

Vas, Peter. (1998). *Sensorless Vector and Direct Torque Control*: Oxford University Press.

Wilamowski, Bogdan M., & Irwin, J. david. (2011). *The Industrial Electronics Handbook Power Electronics and Motor Drives* (2 ed.): CRC Press.

Yu, Zhenyu. (2005). *Variable Frequency Air-Conditioner Control Solution Based on C2000 DSP*. Paper presented at the 8th Texas Instruments Developer Conference.

APPENDIX A Hardware Pictures
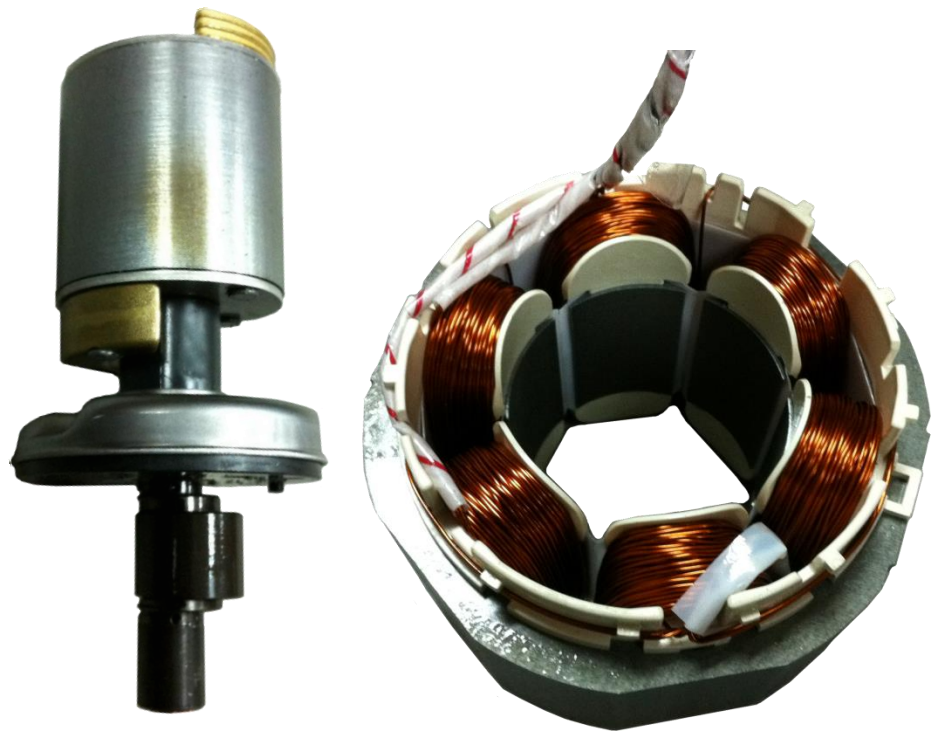
Rotor (left side) and stator (right side) of IPMSM

Exploded view of SPMSM (Baldor, 2003)
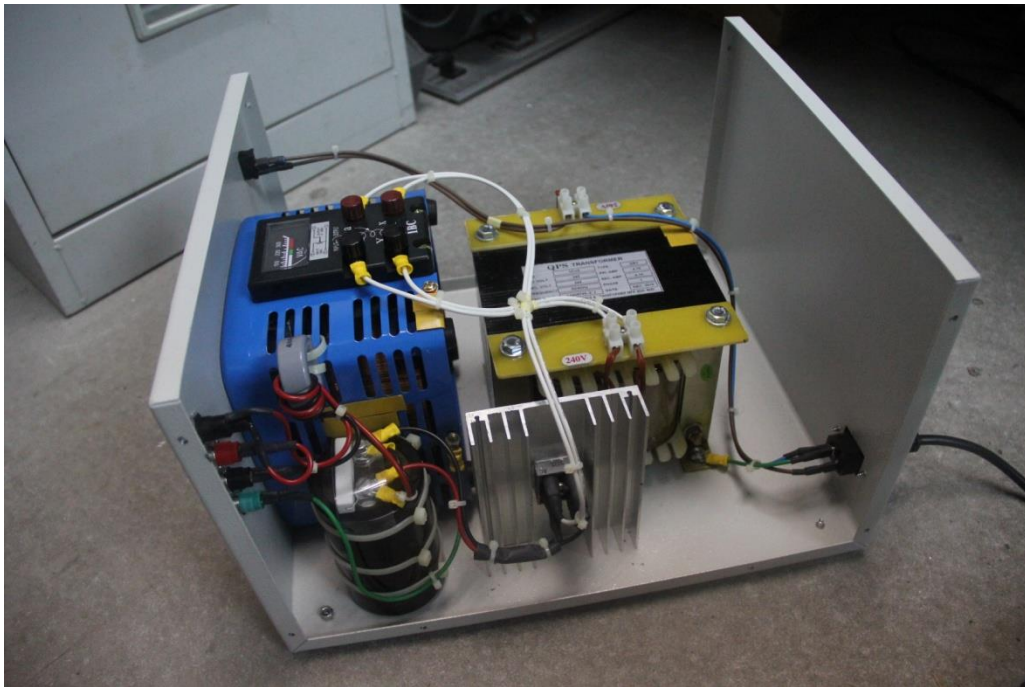
Single phase 300VDC power supply
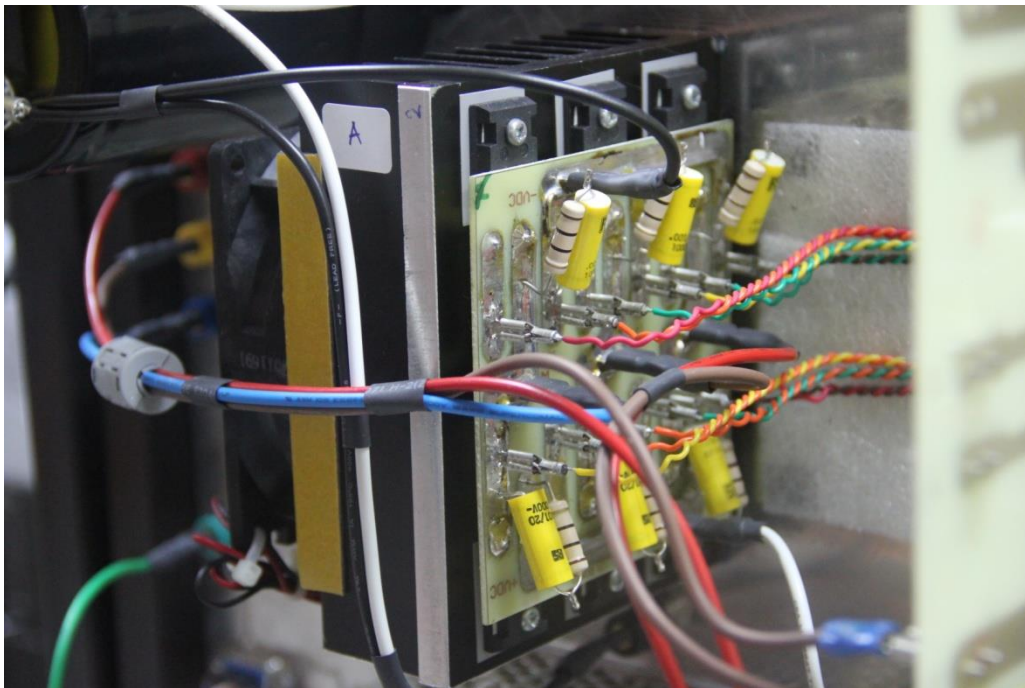
Three-phase VSI

Complete variable speed drive controller for SPMSM
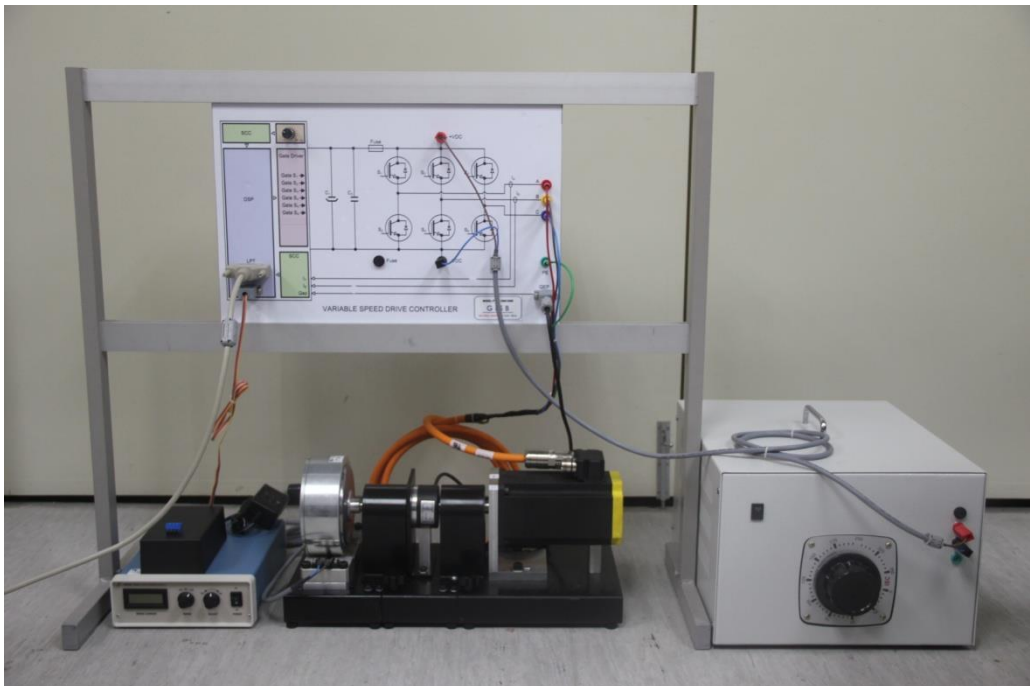
Complete setup of SPMSM drive system
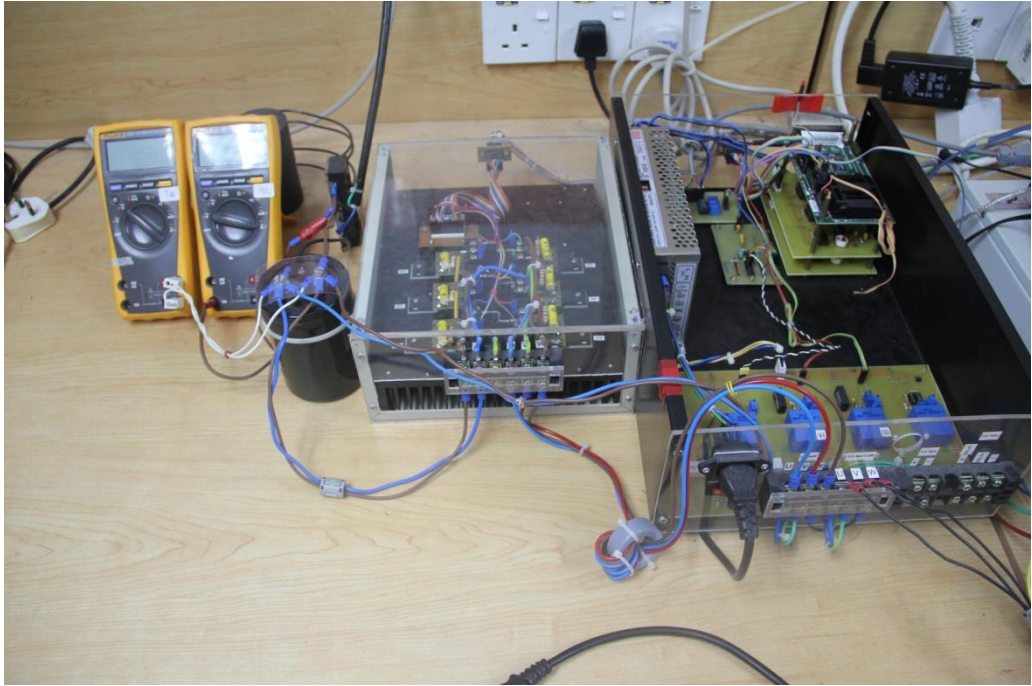
Complete variable speed drive controller for IPMSM
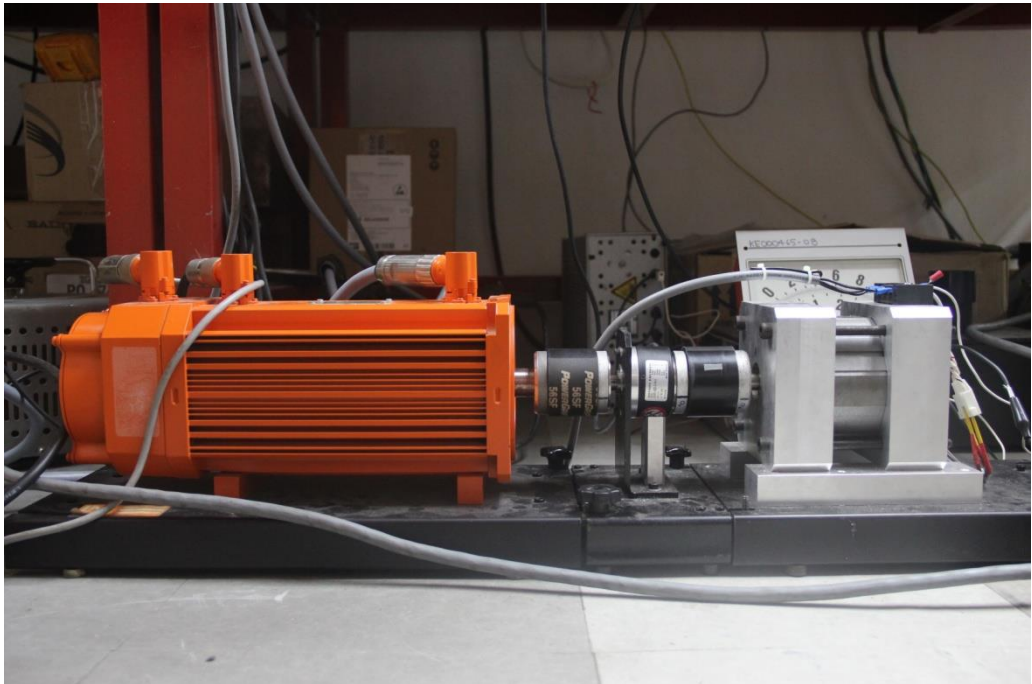
Motor setup

APPENDIX B Hysteresis Brake Curve and Table
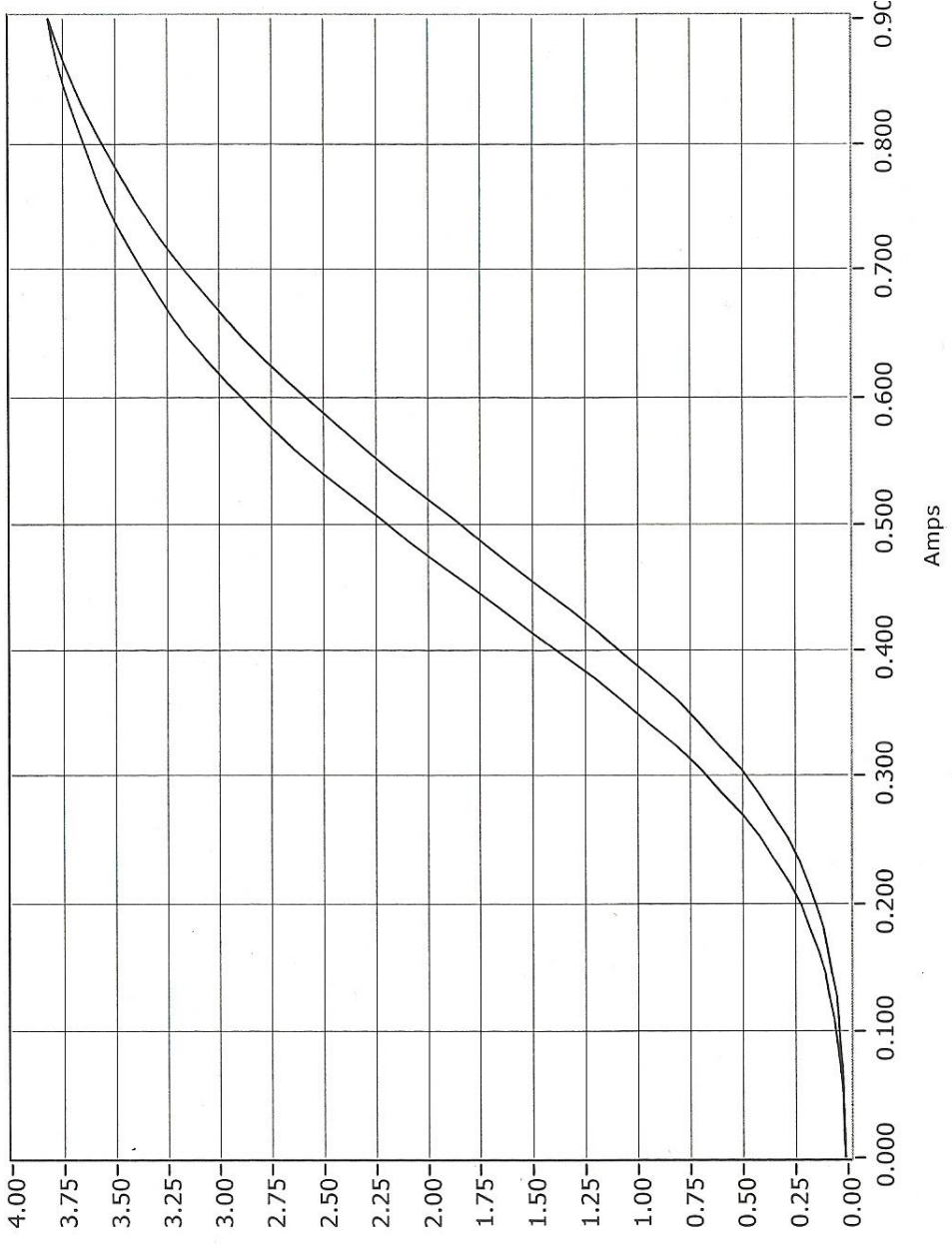
Model Number
AHB-3

Serial Number
11H3484

Comments
SHANGHAI
S/O#40610

N m

MAGTROL

Model
AHB-3

Serial Number
11H3484

Comments
SHANGHAI
S/O#40610

MAGTROL

Amps	N m	Hysteresis
0.000	0.014	0.003
0.018	0.013	0.005
0.036	0.022	0.005
0.054	0.022	0.002
0.072	0.026	0.012
0.090	0.037	0.012
0.108	0.044	0.020
0.126	0.053	0.034
0.144	0.075	0.031
0.162	0.094	0.045
0.180	0.116	0.065
0.198	0.152	0.069
0.216	0.191	0.086
0.234	0.234	0.115
0.252	0.288	0.128
0.270	0.358	0.144
0.288	0.428	0.176
0.306	0.507	0.194
0.324	0.608	0.207
0.342	0.707	0.241
0.360	0.808	0.265
0.378	0.931	0.273
0.396	1.061	0.291
0.414	1.184	0.316
0.432	1.316	0.327
0.450	1.460	0.330
0.468	1.603	0.339
0.486	1.741	0.350
0.504	1.877	0.350
0.522	2.020	0.345
0.540	2.165	0.340
0.558	2.296	0.339
0.576	2.421	0.329
0.594	2.545	0.314
0.612	2.670	0.296
0.630	2.787	0.279
0.648	2.895	0.265
0.666	2.991	0.248
0.684	3.086	0.224
0.702	3.177	0.201
0.720	3.265	0.177
0.738	3.340	0.161
0.756	3.409	0.142
0.774	3.472	0.121
0.792	3.533	0.099
0.810	3.594	0.077
0.828	3.649	0.060
0.846	3.698	0.048
0.864	3.741	0.036
0.882	3.781	0.019
0.900	3.820	0.000

Amps	N m	Hysteresis
0.882	3.801	0.000
0.864	3.777	0.000
0.846	3.746	0.000
0.828	3.710	0.000
0.810	3.671	0.000
0.792	3.632	0.000
0.774	3.593	0.000
0.756	3.551	0.000
0.738	3.501	0.000
0.720	3.442	0.000
0.702	3.378	0.000
0.684	3.309	0.000
0.666	3.239	0.000
0.648	3.160	0.000
0.630	3.066	0.000
0.612	2.966	0.000
0.594	2.859	0.000
0.576	2.750	0.000
0.558	2.634	0.000
0.540	2.505	0.000
0.522	2.365	0.000
0.504	2.227	0.000
0.486	2.091	0.000
0.468	1.942	0.000
0.450	1.790	0.000
0.432	1.643	0.000
0.414	1.501	0.000
0.396	1.351	0.000
0.378	1.204	0.000
0.360	1.073	0.000
0.342	0.947	0.000
0.324	0.815	0.000
0.306	0.701	0.000
0.288	0.604	0.000
0.270	0.502	0.000
0.252	0.416	0.000
0.234	0.348	0.000
0.216	0.278	0.000
0.198	0.221	0.000
0.180	0.181	0.000
0.162	0.139	0.000
0.144	0.106	0.000
0.126	0.087	0.000
0.108	0.064	0.000
0.090	0.049	0.000
0.072	0.037	0.000
0.054	0.024	0.000
0.036	0.017	0.000
0.018	0.018	0.000
0.000	0.011	0.000

APPENDIX C Sensor-FOC SPMSM source code

```
/*============================================================
System Name    :    SPMSM
File Name      :    SPMSM.C
Description    :
Sensor Field Orientation Control for a Three Phase Surface-
mounted Permanent-Magnet Synchronous Motor (SPMSM)
Date           :    23 Jan 2010
Update         :    24 May 2011
Sampling time of inner loop  = 50us
Sampling time of outer loop  = 500us
============================================================*/


// Include header files used in the main function
#include "DSP281x_Device.h"
#include "IQmathLib.h"
#include <math.h>

// DMC header files
#include "park.h"
#include "ipark.h"
#include "pid_reg3.h"
#include "clarke.h"
#include "svgen_dq.h"
#include "speed_fr.h"
#include "f281xqep.h"
#include "xilem_vdc.h"
#include "f281xpwm.h"
#include "volt_calc.h"
#include "dlog4ch.h"
#include "f281xpwmdac.h"

// Parameter
#define PI 3.14159265358979
#define SYSTEM_FREQUENCY 150
#define ISR_FREQUENCY 20
#define P 4
#define BASE_FREQ 100

// Prototype statements for functions
void active_update(void);
interrupt void MainISR(void);
interrupt void QepISR(void);

// Global variables used in this system
float32 IdRef = 0;
float32 SpeedRef = 0.2;
float32 T = 0.001/ISR_FREQUENCY;
Uint16 LoopIsr = 0;
Uint16 Loop1 = 0;
```

```c
Uint16 Loop2 = 0;
int16 PwmDacCh1=0;
int16 PwmDacCh2=0;
int16 PwmDacCh3=0;
int16 DlogCh1 = 0;
int16 DlogCh2 = 0;
int16 DlogCh3 = 0;
int16 DlogCh4 = 0;
volatile Uint16 Enable = FALSE;
volatile Uint16 Start_Stop = FALSE;
Uint16 Mode = FALSE;
Uint16 SpeedLoopPrescaler = 10;
Uint16 SpeedLoopCount = 1;
_iq15 spd;

// Initialize DMC modules with default setting
ILEM2DCBUSMEAS ilm2_vdc1 = ILEM2DCBUSMEAS_DEFAULTS;
CLARKE clarke1 = CLARKE_DEFAULTS;
PARK park1 = PARK_DEFAULTS;
IPARK ipark1 = IPARK_DEFAULTS;
PIDREG3 pid1_id = PIDREG3_DEFAULTS;
PIDREG3 pid1_iq = PIDREG3_DEFAULTS;
PIDREG3 pid1_spd = PIDREG3_DEFAULTS;
PWMGEN pwm1 = PWMGEN_DEFAULTS;
SVGENDQ svgen_dq1 = SVGENDQ_DEFAULTS;
QEP qep1 = QEP_DEFAULTS;
SPEED_MEAS_QEP speed1 = SPEED_MEAS_QEP_DEFAULTS;
RMPCNTL rc1 = RMPCNTL_DEFAULTS;
DLOG_4CH dlog = DLOG_4CH_DEFAULTS;
PWMDAC pwmdac1 = PWMDAC_DEFAULTS;

void main(void)
 {
// Initialization code for DSP_TARGET = F2812
     InitSysCtrl();

    EALLOW;
    SysCtrlRegs.HISPCP.all = 0x0000;
    EDIS;

     DINT;
     IER = 0x0000;
     IFR = 0x0000;

     InitPieCtrl();
     InitPieVectTable();
     EvaRegs.GPTCONA.all = 0;

while (Enable==FALSE)
    {
      Loop1++;
    }
// Enable T1 Underflow interrupt bits for GP timer 1
```

```
        EvaRegs.EVAIMRA.bit.T1UFINT = 1;
        EvaRegs.EVAIFRA.bit.T1UFINT = 1;


// Enable CAP3 interrupt bits for GP timer 2
        EvaRegs.EVAIMRC.bit.CAP3INT = 1;
        EvaRegs.EVAIFRC.bit.CAP3INT = 1;


         EALLOW;
         PieVectTable.T1UFINT = &MainISR;
         PieVectTable.CAPINT3 = &QepISR;
         EDIS;


// Enable PIE group 2 interrupt 6 for T1UFINT &
// group 3 interrupt 7 for CAP3INT
        PieCtrlRegs.PIEIER2.all = M_INT6;
        PieCtrlRegs.PIEIER3.all = M_INT7;


// Enable CPU INT2 for T1UFINT and INT3 for CAP3INT:
        IER |= (M_INT2 | M_INT3);


// Initialize PWM module
        pwm1.PeriodMax = SYSTEM_FREQUENCY*1000000*T/2;
        pwm1.init(&pwm1);


// Initialize PWMDAC module
        pwmdac1.PeriodMax = (SYSTEM_FREQUENCY*200/(30*2))*5;
        pwmdac1.PwmDacInPointer0 = &PwmDacCh1;
        pwmdac1.PwmDacInPointer1 = &PwmDacCh2;
        pwmdac1.PwmDacInPointer2 = &PwmDacCh3;
        pwmdac1.init(&pwmdac1);


// Initialize DATALOG module
        dlog.iptr1 = &DlogCh1;
        dlog.iptr2 = &DlogCh2;
        dlog.iptr3 = &DlogCh3;
        dlog.iptr4 = &DlogCh4;
        dlog.trig_value = 0x1;
        dlog.size = 0x400;
        dlog.prescalar = 1;
        dlog.init(&dlog);


// Initialize QEP module
        qep1.LineEncoder = 5000;
        qep1.MechScaler = _IQ30(0.25/qep1.LineEncoder);
        qep1.PolePairs = P/2;
        qep1.CalibratedAngle = 1800;
        qep1.init(&qep1);


// Initialize ADC module
        ilm2_vdc1.init(&ilm2_vdc1);
        ilm2_vdc1.ImeasAOffset = 0x0000;
        ilm2_vdc1.ImeasBOffset = 0x0000;
// Initialize the Speed module for
```

```c
// QEP based speed calculation
    speed1.K1 = _IQ21(1/(BASE_FREQ*T));
    speed1.K2 = _IQ(1/(1+T*2*PI*30));
    speed1.K3 = _IQ(1)-speed1.K2;
    speed1.BaseRpm = 120*(BASE_FREQ/P);

//Initialize PID Module for speed
    pid1_spd.Kp = _IQ(1);
    pid1_spd.Ki = _IQ(0.007);
    pid1_spd.Kd = _IQ(0);
    pid1_spd.Kc = _IQ(0.007);
    pid1_spd.OutMax = _IQ(1);
    pid1_spd.OutMin = _IQ(-1);

//Initialize PID Module for id
    pid1_id.Kp = _IQ(0.3);
    pid1_id.Ki = _IQ(0.15);
    pid1_id.Kd = _IQ(0);
    pid1_id.Kc = _IQ(0.15);
    pid1_id.OutMax = _IQ(0.3);
    pid1_id.OutMin = _IQ(-0.3);

//Initialize PID Module for iq
    pid1_iq.Kp = _IQ(0.7);
    pid1_iq.Ki = _IQ(0.15);
    pid1_iq.Kd = _IQ(0);
    pid1_iq.Kc = _IQ(0.15);
    pid1_iq.OutMax = _IQ(0.85);
    pid1_iq.OutMin = _IQ(-0.85);

// Enable global Interrupts and higher priority
// real-time debug events:
    EINT;   // Enable Global interrupt INTM
    ERTM;   // Enable Global realtime interrupt DBGM

// IDLE loop. Just sit and loop forever:
    for(;;) Loop2++;

}


interrupt void MainISR(void)
{

// Verifying the ISR
    LoopIsr++;

// Call the ILEG2_VDC read function.
    ilm2_vdc1.read(&ilm2_vdc1);

// Call the QEP calculation function.
    qep1.calc(&qep1);
// Connect inputs of the SPEED_FR module
```

```c
// and call the speed calculation function
    speed1.ElecTheta = _IQ15toIQ((int32)qep1.ElecTheta);
    speed1.DirectionQep = (int32)(qep1.DirectionQep);
    speed1.calc(&speed1);

// Connect inputs of the CLARKE module
// and call the clarke transformation
    clarke1.As = _IQ15toIQ((int32)ilm2_vdc1.ImeasA);
    clarke1.Bs = _IQ15toIQ((int32)ilm2_vdc1.ImeasB);
    clarke1.calc(&clarke1);

// Connect inputs of the PARK module
// and call the park transformation
    park1.Alpha = clarke1.Alpha;
    park1.Beta = clarke1.Beta;
    park1.Angle = speed1.ElecTheta;
    park1.calc(&park1);

// Connect inputs of the PID_REG3 module
// and call the PID speed controller
   if (SpeedLoopCount==SpeedLoopPrescaler)
    {

    if(Mode==FALSE      //Internal SpeedRef input
    {
     pid1_spd.Ref = _IQ(SpeedRef);
    }

    else if(Mode==TRUE  //External SpeedRef input
    {
    spd = ilm2_vdc1.VdcMeas - _IQ15(0.015);
    pid1_spd.Ref = _IQ15toIQ((int32)spd);
    }


     pid1_spd.Fdb = speed1.Speed;
     pid1_spd.calc(&pid1_spd);
     SpeedLoopCount=1;
    }
    else SpeedLoopCount++;

// Connect inputs of the PID_REG3 module
// and call the PID IQ controller
    pid1_iq.Ref = pid1_spd.Out;
    pid1_iq.Fdb = park1.Qs;
    pid1_iq.calc(&pid1_iq);
// Connect inputs of the PID_REG3 module
// and call the PID ID controller
    pid1_id.Ref = _IQ(IdRef);
    pid1_id.Fdb = park1.Ds;
    pid1_id.calc(&pid1_id);

// Connect inputs of the INV_PARK module
```

```
    // and call the inverse park transformation
     ipark1.Ds = pid1_id.Out;
     ipark1.Qs = pid1_iq.Out;
     ipark1.Angle = speed1.ElecTheta;
     ipark1.calc(&ipark1);

// Connect inputs of the SVGEN_DQ module
// and call the space-vector gen.
     svgen_dq1.Ualpha = ipark1.Alpha;
     svgen_dq1.Ubeta = ipark1.Beta;
     svgen_dq1.calc(&svgen_dq1);

// Connect inputs of the PWM_DRV module
// and call the PWM signal generation
     pwm1.MfuncC1 = (int16)_IQtoIQ15(svgen_dq1.Ta);
     pwm1.MfuncC2 = (int16)_IQtoIQ15(svgen_dq1.Tb);
     pwm1.MfuncC3 = (int16)_IQtoIQ15(svgen_dq1.Tc);
     pwm1.update(&pwm1);

// Connect inputs of the PWMDAC and DLOG module
     PwmDacCh1 =  ilm2_vdc1.ImeasA;
     PwmDacCh2 =  ilm2_vdc1.ImeasB;
     PwmDacCh3 = (int16)_IQtoIQ15(speed1.ElecTheta);
     DlogCh1 = ilm2_vdc1.ImeasA;
     DlogCh2 = (int16)_IQtoIQ15(speed1.ElecTheta);
     DlogCh3 = ilm2_vdc1.VdcMeas;
     DlogCh4 = spd;

     active_update();

// Call the PWMDAC update function.
     pwmdac1.update(&pwmdac1);

// Call the DATALOG update function.
     dlog.update(&dlog);

// Enable more interrupts from this timer
     EvaRegs.EVAIMRA.bit.T1UFINT = 1;

     EvaRegs.EVAIFRA.all = BIT9;

// Acknowledge interrupt to recieve
// more interrupts from PIE group 2
     PieCtrlRegs.PIEACK.all |= PIEACK_GROUP2;
}

interrupt void QepISR(void)
{

// ------------------------------------------------------
// ----------------------
//    Call the QEP_DRV isr function.
```

```
// ----------------------------------------------------------------------------
    qep1.isr(&qep1);

// Enable more interrupts from this timer
    EvaRegs.EVAIMRC.bit.CAP3INT = 1;

     EvaRegs.EVAIFRC.all = BIT2;

// Acknowledge interrupt to recieve more interrupts from
PIE group 3
    PieCtrlRegs.PIEACK.all |= PIEACK_GROUP3;

}


void active_update()
{
    if (Start_Stop==TRUE)
    {
    // Enable PWM output
        EvaRegs.COMCONA.all = 0xA300;
    }
    else
    if (Start_Stop==FALSE)
    {
    // Disable PWM output
        EvaRegs.COMCONA.all = 0xA100;
    }
}

//==========================================================
// End of Code
//==========================================================
```

APPENDIX D Sensorless-FOC IPMSM source code

```c
/*==============================================================
System Name    :    IPMSM
File Name      :    IPMSM.C
Description    :
Sensorless Field Orientation Control for a Three Phase
Interior Permanent-Magnet Synchronous Motor (IPMSM)
Date           :    26 April 2011  09:00:00 A.M.
Update         :    29 April 2011  10:27:00 P.M.
Sampling time of inner loop  = 50us
Sampling time of outer loop  = 500us
==============================================================*/

// Include header files used in the main function
#include "DSP281x_Device.h"
#include "IQmathLib.h"
#include <math.h>

// DMC header files
#include "park.h"
#include "ipark.h"
#include "pid_reg3.h"
#include "clarke.h"
#include "svgen_dq.h"
#include "speed_fr.h"
#include "f281xqep.h"
#include "xilem_vdc.h"
#include "f281xpwm.h"
#include "volt_calc.h"
#include "dlog4ch.h"
#include "f281xpwmdac.h"

// Parameter
#define PI 3.14159265358979
#define SYSTEM_FREQUENCY 150
#define ISR_FREQUENCY 20
#define P 4
#define BASE_FREQ 100

// Prototype statements for functions
void active_update(void);
interrupt void MainISR(void);

// Global variables used in this system
float32 IdRef = 0;                          // Id reference
float32 SpeedRef = 0.2;                      // Speed reference
float32 T = 0.001/ISR_FREQUENCY;        // Samping period
Uint16 LoopIsr = 0;
Uint16 Loop1 = 0;
Uint16 Loop2 = 0;
```

```
int16 PwmDacCh1=0;
int16 PwmDacCh2=0;
int16 PwmDacCh3=0;
int16 DlogCh1 = 0;
int16 DlogCh2 = 0;
int16 DlogCh3 = 0;
int16 DlogCh4 = 0;
volatile Uint16 Enable = FALSE;
volatile Uint16 Start_Stop = FALSE;
Uint16 LockRotorFlag = FALSE;
Uint16 SpeedLoopPrescaler = 10
Uint16 SpeedLoopCount = 1

// Actual value variable
_iq vx=0;        // VAlpha
_iq vy=0;        // VBeta
_iq ix=0;        // iAlpha
_iq iy=0;        // iBeta
_iq ldx=0;       // FluxAlpha
_iq ldy=0;       // FluxBeta
_iq ldm;         // FluxMagnitude
_iq Ldd;
_iq Lqq;
_iq ixc;         // iAlpha correction
_iq iyc;         // iBeta correction
_iq ThetaEst2;   // Estimated theta

// Initialize DMC modules with default setting
ILEM2DCBUSMEAS ilm2_vdc1 = ILEM2DCBUSMEAS_DEFAULTS;
CLARKE clarke1 = CLARKE_DEFAULTS;
PARK park1 = PARK_DEFAULTS;
PARK park2 = PARK_DEFAULTS;
IPARK ipark1 = IPARK_DEFAULTS;
IPARK ipark2 = IPARK_DEFAULTS;
PIDREG3 pid1_id = PIDREG3_DEFAULTS;
PIDREG3 pid1_iq = PIDREG3_DEFAULTS;
PIDREG3 pid1_spd = PIDREG3_DEFAULTS;
PWMGEN pwm1 = PWMGEN_DEFAULTS;
SVGENDQ svgen_dq1 = SVGENDQ_DEFAULTS;
SPEED_MEAS_QEP speed1 = SPEED_MEAS_QEP_DEFAULTS;
DLOG_4CH dlog = DLOG_4CH_DEFAULTS;
PWMDAC pwmdac1 = PWMDAC_DEFAULTS;
PHASEVOLTAGE volt1 = PHASEVOLTAGE_DEFAULTS;

void main(void)
 {
// Initialization code for DSP_TARGET = F2812
     InitSysCtrl();

     EALLOW;
     SysCtrlRegs.HISPCP.all = 0x0000;     // SYSCLKOUT/1
     EDIS;
```

```
     DINT;
     IER = 0x0000;
     IFR = 0x0000;

     InitPieCtrl();
     InitPieVectTable();
    EvaRegs.GPTCONA.all = 0;

   while (Enable==FALSE)
    {
      Loop1++;
    }

// Enable T1 Underflow interrupt bits for GP timer 1
    EvaRegs.EVAIMRA.bit.T1UFINT = 1;
    EvaRegs.EVAIFRA.bit.T1UFINT = 1;

     EALLOW;
     PieVectTable.T1UFINT = &MainISR;
     EDIS;

// Enable PIE group 2 interrupt 6 for T1UFINT
    PieCtrlRegs.PIEIER2.all = M_INT6;

// Enable CPU INT2 for T1UFINT
     IER |= (M_INT2);

// Initialize PWM module
    pwm1.PeriodMax = SYSTEM_FREQUENCY*1000000*T/2;
    pwm1.init(&pwm1);

// Initialize PWMDAC module
    pwmdac1.PeriodMax = (SYSTEM_FREQUENCY*200/(30*2))*5;
    pwmdac1.PwmDacInPointer0 = &PwmDacCh1;
    pwmdac1.PwmDacInPointer1 = &PwmDacCh2;
    pwmdac1.PwmDacInPointer2 = &PwmDacCh3;
    pwmdac1.init(&pwmdac1);

// Initialize DATALOG module
    dlog.iptr1 = &DlogCh1;
    dlog.iptr2 = &DlogCh2;
    dlog.iptr3 = &DlogCh3;
    dlog.iptr4 = &DlogCh4;
    dlog.trig_value = 0x1;
    dlog.size = 0x400;
    dlog.prescalar = 1;
    dlog.init(&dlog);

// Initialize ADC module
    ilm2_vdc1.init(&ilm2_vdc1);
    ilm2_vdc1.ImeasAOffset = 0xFE00;
    ilm2_vdc1.ImeasBOffset = 0x0000;
// Initialize the Speed module for
```

```c
// QEP based speed calculation
    speed1.K1 = _IQ21(1/(BASE_FREQ*T));
    speed1.K2 = _IQ(1/(1+T*2*PI*30));
    speed1.K3 = _IQ(1)-speed1.K2;
    speed1.BaseRpm = 120*(BASE_FREQ/P);

//Initialize PID Module for speed
    pid1_spd.Kp = _IQ(0.7);
    pid1_spd.Ki = _IQ(0.003);
    pid1_spd.Kd = _IQ(0);
    pid1_spd.Kc = _IQ(0.003);
    pid1_spd.OutMax = _IQ(1);
    pid1_spd.OutMin = _IQ(-1);

//Initialize PID Module for id
    pid1_id.Kp = _IQ(0.5);
    pid1_id.Ki = _IQ(0.004);
    pid1_id.Kd = _IQ(0);
    pid1_id.Kc = _IQ(0.008);
    pid1_id.OutMax = _IQ(0.5);
    pid1_id.OutMin = _IQ(-0.5);

//Initialize PID Module for iq
    pid1_iq.Kp = _IQ(0.5);
    pid1_iq.Ki = _IQ(0.0045);
    pid1_iq.Kd = _IQ(0);
    pid1_iq.Kc = _IQ(0.009);
    pid1_iq.OutMax = _IQ(0.85);
    pid1_iq.OutMin = _IQ(-0.85);

// Enable global Interrupts and higher priority
// real-time debug events:
    EINT;   // Enable Global interrupt INTM
    ERTM;   // Enable Global realtime interrupt DBGM

// IDLE loop. Just sit and loop forever:
    for(;;) Loop2++;

}


interrupt void MainISR(void)
{

// Verifying the ISR
    LoopIsr++;

// Call the ILEG2_VDC read function.
    ilm2_vdc1.read(&ilm2_vdc1);

// Connect inputs of the SPEED_FR module
// and call the speed calculation function
    speed1.ElecTheta = ThetaEst2;
```

```
        speed1.calc(&speed1);

// Connect inputs of the CLARKE module
// and call the clarke transformation
        clarke1.As = _IQ15toIQ((int32)ilm2_vdc1.ImeasA);
        clarke1.Bs = _IQ15toIQ((int32)ilm2_vdc1.ImeasB);
        clarke1.calc(&clarke1);

// Connect inputs of the PARK module
// and call the park transformation
        park1.Alpha = clarke1.Alpha;
        park1.Beta = clarke1.Beta;
        park1.Angle = ThetaEst2;
        park1.calc(&park1);

// Connect inputs of the PID_REG3 module
// and call the PID speed controller
        if (SpeedLoopCount==SpeedLoopPrescaler)
         {
          pid1_spd.Ref =_IQ(SpeedRef);
         pid1_spd.Fdb = speed1.Speed;
          pid1_spd.calc(&pid1_spd);
         SpeedLoopCount=1;
         }
        else SpeedLoopCount++;

// Connect inputs of the PID_REG3 module
// and call the PID IQ controller
        pid1_iq.Ref = pid1_spd.Out;
        pid1_iq.Fdb = park1.Qs;
        pid1_iq.calc(&pid1_iq);

// Connect inputs of the PID_REG3 module
// and call the PID ID controller
        pid1_id.Ref = _IQ(IdRef);
        pid1_id.Fdb = park1.Ds;
        pid1_id.calc(&pid1_id);

// Connect inputs of the INV_PARK module
// and call the inverse park transformation
        ipark1.Ds = pid1_id.Out;
        ipark1.Qs = pid1_iq.Out;
        ipark1.Angle = ThetaEst2;
        ipark1.calc(&ipark1);

// Connect inputs of the SVGEN_DQ module
// and call the space-vector gen.
        svgen_dq1.Ualpha = ipark1.Alpha;
        svgen_dq1.Ubeta = ipark1.Beta;
        svgen_dq1.calc(&svgen_dq1);


// Connect inputs of the PWM_DRV module
```

```
// and call the PWM signal generation
    pwm1.MfuncC1 = (int16)_IQtoIQ15(svgen_dq1.Ta);
    pwm1.MfuncC2 = (int16)_IQtoIQ15(svgen_dq1.Tb);
    pwm1.MfuncC3 = (int16)_IQtoIQ15(svgen_dq1.Tc);
    pwm1.update(&pwm1);


// Connect inputs of the VOLTCALC module
    volt1.DcBusVolt = _IQ15toIQ((int32)ilm2_vdc1.VdcMeas);
    volt1.MfuncV1 = svgen_dq1.Ta;
    volt1.MfuncV2 = svgen_dq1.Tb;
    volt1.MfuncV3 = svgen_dq1.Tc;
    volt1.calc(&volt1);


// Flux calculation with actual value conversion
    vx = _IQmpy(_IQ(100),volt1.Valpha);
    ix = _IQmpy(_IQ(8.49),clarke1.Alpha);
    vy = _IQmpy(_IQ(100),volt1.Vbeta);
    iy = _IQmpy(_IQ(8.49),clarke1.Beta);
ldx = ldx + IQmpy(_IQ(T),(vx - _IQmpy(_IQ(1.3),ix) + ixc));
ldy = ldy + IQmpy(_IQ(T),(vy - _IQmpy(_IQ(1.3),iy)+ iyc));
    ldm = _IQmag(ldx,ldy);
    ThetaEst2 = _IQatan2PU(ldy,ldx);


// Stator flux observer
    park2.Alpha = ix;
    park2.Beta = iy;
    park2.Angle = ThetaEst2;
    park2.calc(&park2);
    Ldd = _IQmpy(park2.Ds,_IQ(0.0125)) + _IQ(0.11);
    Lqq = _IQmpy(park2.Qs,_IQ(0.0191));

    ipark2.Ds = Ldd;
    ipark2.Qs = Lqq;
    ipark2.Angle = ThetaEst2;
    ipark2.calc(&ipark2);
    ixc = _IQmpy(_IQ(50), (ipark2.Alpha - ldx));
    iyc = _IQmpy(_IQ(50), (ipark2.Beta - ldy));

// Connect inputs of the PWMDAC and DLOG module
    PwmDacCh1 = (int16)_IQtoIQ15(pid1_spd.Ref);
    PwmDacCh2 = (int16)_IQtoIQ15(pid1_spd.Fdb);
    PwmDacCh3 = (int16)_IQtoIQ15(speed1.ElecTheta);
    DlogCh1 = (int16)_IQtoIQ15(ldx);
    DlogCh2 = (int16)_IQtoIQ15(ldy);
    DlogCh3 = (int16)_IQtoIQ15(volt1.Valpha);
    DlogCh4 = (int16)_IQtoIQ15(ThetaEst2);
    active_update();

// Call the PWMDAC update function.
    pwmdac1.update(&pwmdac1);


// Call the DATALOG update function.
```

```
        dlog.update(&dlog);

// Enable more interrupts from this timer
        EvaRegs.EVAIMRA.bit.T1UFINT = 1;

        EvaRegs.EVAIFRA.all = BIT9;

// Acknowledge interrupt to recieve
// more interrupts from PIE group 2
        PieCtrlRegs.PIEACK.all |= PIEACK_GROUP2;
}


void active_update()
{
        if (Start_Stop==TRUE)
        {
        // Enable PWM output
            EvaRegs.COMCONA.all = 0xA300;
        }
        else
        if (Start_Stop==FALSE)
        {
        // Disable PWM output
            EvaRegs.COMCONA.all = 0xA100;
        }
}

//=========================================================
// End of Code
//=========================================================
```

PUBLICATION

N. A. Rahim, M. A. Rahman, M. F. M. Elias, H. W. Ping. (2012). Implementation of Speed Sensorless Drives for IPMSM Based on Simplified Stator Flux Observer. *IREE, 7*(4). (*ISI-Cited Publication*)