ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL and COMPUTER ENGINEERING

DSP BASED IMPELEMENTATION OF FIELD-WEAKENING ON SYNCHRONOUS MOTOR FOR HIGH SPEED OPERATION

Thesis

Submitted in partial fulfillment of the requirement for the degree of

MASTER OF SCIENCE
In
ELECTRICAL ENGINEERING

By

HENOK BEREHANU

ADVISOR:

DR. MENGESHA MAMO

January 2011

ADDIS ABABA, ETHIOPIA

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL and COMPUTER ENGINEERING

DSP BASED IMPELEMENTATION OF FIELD-WEAKENING ON SYNCHRONOUS MOTOR FOR HIGH SPEED OPERATION

By

HENOK BEREHANU

Dr. Mengesha Mamo_____	____________________	_________________
Advisor	Signature	Date
Dr. _________________	_____________________	_________________
Chairman, Department	Signature	Date
Dr. ________________	_______________________	__________________
Internal Examiner	Signature	Date
Prof. ________________	__________________________	__________________
External examiner	Signature	Date

Table of Contents

ABSTRACT

Permanent magnet motors have high efficiency and torque density compared to induction and conventional synchronous motors in general. However, in applications requiring wide speed range, the air-gap field, which is produced by the permanent magnets, can't be controlled directly to control the back-emf of the motor for field weakening operation. In conventional synchronous machines the excitation current can be controlled to control the field and hence the speed.

This thesis presents development of field weakening control for Permanent Magnet Synchronous Motor (PMSM) using demagnetizing component of stator current. The control is implemented using Texas Instruments' TMS320LF2401A Digital Signal Processor (DSP). The control method presented relies on the field orientated control (FOC) together with a field-weakening operation. The field-weakening operation is required to extend the speed in the constant power range of operation, where there is no more voltage adjustment possible to increase the speed.

The field oriented control of the motors has been modeled, simulated using MATLAB and Some of the modules of field oriented control are tested practically and results are given in this thesis. The results demonstrate that the speed of the PMSM can be extended in the constant power range for applications requiring wide speed range like in electric vehicle.

ACKNOWLEDGMENT

During my graduate studies, several persons collaborated directly and indirectly with my research. Without their support it would be impossible for me to finish my work. That is why I wish to dedicate this section to recognize their support.

First, I would like to give thanks and praise to the Almighty God for His grace and blessings throughout the entire project.

I want to start expressing a sincere acknowledgement to my advisor, DR. MENGESHA MAMO because he gave me the opportunity to research under his guidance and supervision. I received motivation; encouragement and caring support from him during my studies and for the completion of my work.

Finally, I would like to thank my wife, my family and friends for their continuous support.

List of Figures

List of Tables

List of Symbols

Cos	Cosine wave value
e	Back emf
$e\alpha, e\beta$	α and β axis back emf
encincr	encoder pulse increment
encpulses	no. of encoder pulses per mechanical revolution
epiq,epid	d- and q- axis current regulator errors
epispeed	speed errors
epvr	PI error for field weakening
f_s	Back emf
ia , ib ,ic	Phase a, b, c stator current.
id , iq	d- and q- axis components of stator current
idfmin	Current output limitation
initphase	initial phase
Is	Stator current vector
isdref	d-axis stator reference current
Ismax	Stator current limit
isqref	q-axis stator reference current
iSqrefmin, iSqrefmax	q-axis current limitations
isalfa , isbeta	alfa and beta axis stator currents
J	Moment of inertia

Kcurrent	current constant
Kencoder	encoder constant
Ki ,Kpi,Kcor	PI constant
Kspeed	speed constant
Ld , Lq	d- and q- axis stator self inductance
Nbase	base speed value
p	Number of pole pairs.
Pi	Input power
Po	Output power
PWMPRD	pulse width modulation period
Rs	Stator resistance
S	Arbitrary assigned variable
sin	sine wave value
T	Developed (motor) torque
T1, T2	Switching time
t1, t2	Time vector application in SVPWM
taon ,tbon tcon	PWM commutation instant
TL	Load Torque
Tm	Electromagnetic Torque
Va ,Vb ,Vc	Phase voltage
vd ,vq	d and q- axis components of stator voltage
Vsdref, Vsqref	d and q components of the reference stator voltage
Vdc	DC- link voltage

Vref	Reference space vector voltage
Vs	Stator voltage magnitude (steady state)
Vsmax	Stator voltage limit
Vαref ,Vβref	α and β axis reference voltage
X, Y, Z	SVPWM variables
θ	Electrical rotor angle between the phase a-axis and the north pole
ψa ,ψb ,ψc	Phase stator flux
ψd ,ψq	d ,q component of stator flux
ΨF	Opposing field
Ψm	Flux linkage due to permanent magnet.
Φ	Flux in the airgap
$\omega, \omega ref$	Speed and reference speed of rotation (in electrical rad/Sec.)
ω_s	Synchronous speed

Acronyms

AC	Alternating current
BLDC	Brushless DC machine
CSI	Current Source Inverter
DC	Direct current
emf	Electromagnetic force
FOC	Field oriented control
FW	Field/Flux weakening
mmf	Magneto-motive force
PI	Proportional integral
PMSM	Permanent magnet synchronous machine
PWM	Pulse width modulation
SVPWM	Space vector PWM
VSI	Voltage Source Inverter

Chapter 1

Introduction

The synchronous machine is a constant-speed machine which always rotates at a synchronous speed, which depends on the supply frequency and on the number of poles. The availability of modern permanent magnets (PM) with considerable energy density led to the development of dc machines with PM filed excitation in the 1950s. Introduction of PM to replace electromagnets, which have windings and require an external electric energy source, resulted in compact dc machines. The synchronous machine, with its conventional field excitation in the rotor, is replaced by the PM excitation; the slip rings and brush assembly are dispensed with. With the advent of switching power transistor and silicon – controlled-rectifier devices in later part of 1950s, the replacement of the mechanical commutator with an electronic commutator in the form of an inverter was achieved. These two developments contributed to the development of PM synchronous and brushless dc machines. The armature of the dc machine need not be on the rotor if the machine commutator is replaced by its electronic version. Therefore, the armature of the machine can be on the stator, enabling better cooling and allowing higher voltages to be achieved: significant clearance space is available for insulation in the stator. The excitation field that used nothing but 'an inside out dc machine' with the field and armature interchanged from the stator to rotor and rotor to stator, respectively.

For many years d.c motors were used extensively in areas where variable speed operation was required since their flux and torque could be controlled independently and easily by the field and armature current. Particularly, the separately excited d.c motor has been used mainly for applications where there was a requirement of fast response and four-quadrant operation with high performance near zero speed. However, due to the existence of the commutator and the brushes d.c motors have certain disadvantage. That is, they required periodic maintenance; they cannot be used in explosive or corrosive environments. These problems can be overcome by the application of a.c motors, because of its simplicity, ruggedness, efficiency, low cost, compactness, and economical and volume manufacturing advantages.

Some of the reasons for dealing with AC motor drives rather than DC motor drives are:

_ Commutator and Brushes do not exist in Permanent Magnet Synchronous motors, which can result in higher speed operations.
_ PMSM motors have comparable and frequently better efficiency than the equivalent dc motor.
_ A dc motor must be regularly taken out of service to check or replace brushes and at less frequent interval to resurface the commutator.

Hence PMSM is superior to d.c machine with respect to size, weight, rotor inertia, efficiency, maximum speed, reliability, cost e.t.c.

With suitable control, PMSM motor drives have been shown to be more than a match for DC drives in high-performance applications. While control of the PMSM machine is considerably more complicated than its DC motor counterpart, with continual advancement of microelectronics, these control complexities have essentially been overcome.

PMSM can be categorized into two main groups:

1. The brushless DC machines having trapezoidal flux distribution; and
2. Synchronous machines having approximately sinusoidal air-gap flux density distribution.

In recent years, permanent magnet synchronous motors (PMSM) are widely used in low and mid power applications such as computer peripheral equipments, robotics, adjustable speed drives and electric vehicles [29].

Permanent magnet machines are usually designed to provide constant torque up to a base speed, which corresponds to the rated armature voltage, rated field current and rated armature current and constant power from the base speed up to the maximum speed. To perform this constant power operation field weakening has to be considered in order to respect the current and voltage limitation.

Field-weakening is the process of reducing the air-gap flux in the motor which results in an increased speed range. **Now-a-days high speed is demanded specially in electric vehicles**. For example when motor speed is lower than base speed, the voltage supply can provide enough voltage to support motor back EMF, so that field weakening is not required. When motor speed is higher than base speed, however, motor back EMF will exceed the supply voltage capability unless field weakening is applied [3] [26]. So, the main emphasis of this project will be to implement field weakening in PMSM.

PM motor drives for constant torque and field weakening operation have been a topic of interest for the last twenty years. Different authors have carried out modeling and simulation of such drives.

Chan et al. introduced the use of transformer electromotive force (emf) to oppose the rotational emf by purposely producing a slope on the armature current. Certain torque ripples would be generated [6].

Sebastian and Slemon [1] proposed that with optimum alignment of the stator and magnet fields, maximum torque per ampere is achieved up to a break-point speed. Operation at higher speeds with reduced torque is achieved by adjustment of the current angle to reduce the effective magnet flux (i.e., the equivalent of field weakening).

Sozer and Torrey presented an approach for adaptive control of the surface mount PM motor over its entire speed range. The adaptive flux weakening scheme is able to determine the right amount of direct-axis current at any operating condition.

Namuduri and Murty focused their discussion on the pulse-width-modulation (PWM) strategy and the microcontroller system that implements the phase-angle control scheme. The authors concluded that though the PM motors were not designed for field weakening by phase-angle control, experiments showed that the torque at no-load speed can be increased significantly with phase advance, at the expense of increased motor losses.

Morimoto et al. stated that in the PM motor, direct control of the magnet flux is not available. However, the direct-axis armature current can weaken the air-gap flux. In their operation, magnet demagnetization due to the direct axis armature reaction must be prevented because the magnet torque decreases irreversibly if this demagnetization is very large.

In 1988 Pillay and Krishnan, R. [2] presented PM motor drives and classified them into two types such as permanent magnet synchronous motor drives (PMSM) and brushless dc motor (BLDC) drives. The PMSM has a sinusoidal back emf and requires sinusoidal stator currents to produce constant torque while the BLDC has a trapezoidal back emf and requires rectangular stator currents to produce constant torque. The PMSM is very similar to the wound rotor synchronous machine except that the PMSM that is used for servo applications tends not to have any damper windings and excitation is provided by a permanent magnet instead of a field winding. Hence the d, q model of the PMSM can be derived from the well known model of the synchronous machine with the equations of the damper windings and field current dynamics removed. Equations of the PMSM are derived in rotor reference frame and the equivalent circuit is presented without dampers. The damper windings are not considered because the motor is designed to operate in a drive system with field-oriented control.

In 1997 Jang-Mok, K. and Seung-Ki, S. [4] proposed a novel flux-weakening scheme for an Interior Permanent Magnet Synchronous Motor (IPMSM). It was implemented based on the output of the synchronous PI current regulator reference voltage to PWM inverter. The on-set of flux weakening and the level of the flux were adjusted inherently by the outer voltage regulation loop to prevent the saturation of the current regulator. Attractive features of this flux weakening scheme included no dependency on the machine parameters, the guarantee of current regulation at any operating condition, and smooth and fast transition into and out of the flux weakening mode. Experimental results at various operating conditions including the case of detuned parameters were presented to verify the feasibility of the proposed control scheme.

The general objective of this thesis is:

- To implement field-weakening that can be applied to PMSM.

And **the specific objectives are:**

- To study and develop DSP controller for vector control of PMSM.
- To implement field weakening controller using speed and position sensor.

The thesis is organized into 6 chapters including this introduction. Chapter 2 presents a theoretical review of permanent magnet motors drives which includes permanent magnet materials, classification of the permanent magnet motors, mathematical modeling of PMSM etc..., Chapter 3 deals with the modeling of PMSM, closed loop control techniques used for PM motor drives, and principle of Field Weakening Operation. In the fourth chapter the DSP controller used in this thesis and the software organization is discussed. The fifth chapter is dedicated to the simulation and the DSP processor results. Finally, Chapter 6 presents general conclusions and recommendations for future work.

Chapter 2

General characteristics of PM Synchronous Motor

Synchronous motors are those which rotate at the speed of the stator revolving field, which is called the synchronous speed. The synchronous speed, ω_s is determined by the frequency of the stator supply, f_s and the number of stator pole pairs, p. Unlike the induction motor, the rotor also has p-pole pairs, excited by a separate DC or permanent magnet (PM) source. The stator of a three-phase synchronous motor normally has a sine distributed three-phase winding. When excited with a three-phase balanced supply, a rotating magnetic field develops.

2.1 Overview of PMSM

A permanent magnet synchronous motor (PMSM) is a motor that uses permanent magnets to produce the air gap magnetic field rather than using electromagnets. These motors have significant advantages, attracting the interest of researchers and industry for use in many applications.

2.1.1 Permanent Magnet Materials

The properties of the permanent magnet material will affect directly the performance of the motor and proper knowledge is required for the selection of the materials and for understanding PM motors. The earliest manufactured magnet materials were hardened steel. Magnets made from steel were easily magnetized. However, they could hold very low energy and it was easy to demagnetize. In recent years other magnet materials such as Aluminum Nickel and Cobalt alloys (ALNICO), Strontium Ferrite or Barium Ferrite (Ferrite), Samarium Cobalt (First generation rare earth magnet) (SmCo) and Neodymium Iron-Boron (Second generation rare earth magnet) (NdFeB) have been developed and used for making permanent magnets. The rare earth magnets are categorized into two classes: Samarium Cobalt (SmCo) magnets and Neodymium Iron Boride (NdFeB) magnets. SmCo magnets have higher flux density levels but they are very expensive. NdFeB magnets are the most common rare earth magnets used in motors these days.

2.1.2 Classification of Permanent Magnet Motors

PM motors are classified on the basis of the flux density distribution and the shape of Current excitation. They are PMSM and PM brushless motors (BLDC). The PMSM has a Sinusoidal-shaped back EMF and is designed to develop sinusoidal back EMF waveforms.

They have the following:

1. Sinusoidal distribution of magnet flux in the air gap
2. Sinusoidal current waveforms
3. Sinusoidal distribution of stator conductors.

BLDC has a trapezoidal-shaped back EMF and is designed to develop trapezoidal back EMF waveforms. They have the following:

1. Rectangular distribution of magnet flux in the air gap
2. Rectangular current waveform
3. Concentrated stator windings.

2.2 Modeling of Permanent Magnet Synchronous Motor

Detailed modeling of PM motor drive system is required for proper simulation of the system. The d-q model has been developed on rotor reference frame as shown in figure 2.1. At any time t, the rotating rotor d-axis makes and angle θ with the fixed stator phase axis.

Fig.2.1 illustrates a conceptual cross-sectional view of a 3-phase, 2-pole PM synchronous motor along with two reference frames. The stator reference axis for the a-phase is chosen to the direction of maximum mmf when a positive a-phase current is supplied at its maximum level. Reference axis for b- and c- stator frame are chosen 120° and 240° (electrical angle) ahead of the a-axis, respectively. Following the convention of choosing the rotor reference frame, the direction of permanent magnet flux is chosen as the d-axis, while the q-axis is 90 degrees ahead of the d-axis. The angle of the rotor d-axis with respect to the stator a-axis is θ.As the machine turns, the d-q reference frame is rotating at a speed of $\omega = d\theta/dt$, while the stator a-b-c- axes are fixed in space. We will find out later that the choice of this rotating frame greatly simplifies the dynamic equations of the model.

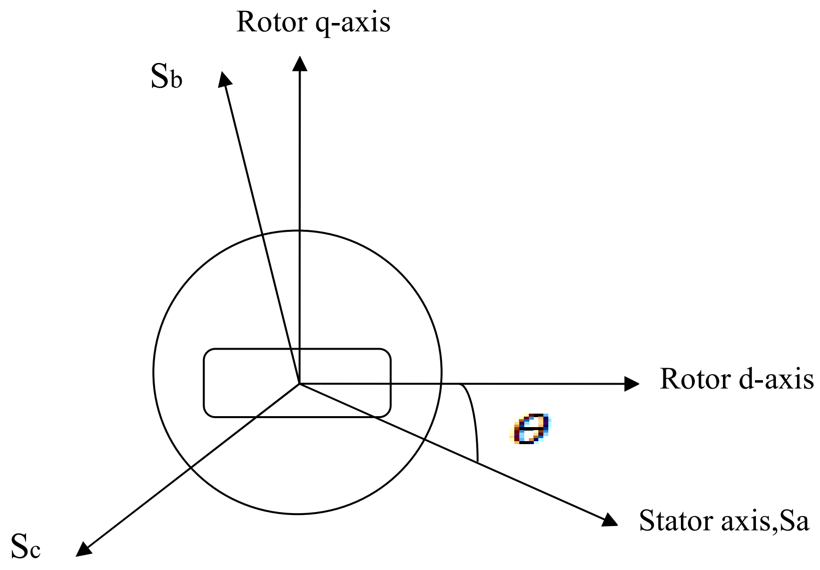


Figure 2.1 Three - Phase PMSM motor with one pole pair permanent magnet

2.2.1 Mathematical Description of PMSM

The electrical dynamic equation in terms of phase variables can be written as follows in the stator phase quantities [5]

$$v_a = R_s i_a + \frac{d\psi_a}{dt} \tag{2.2}$$

$$v_b = R_s i_b + \frac{d\psi_b}{dt} \tag{2.3}$$

$$v_c = R_s i_c + \frac{d\psi_c}{dt} \tag{2.4}$$

For this model, input power P_i can be represented as

$$P_i = v_a i_a + v_b i_b + v_c i_c \qquad (2.5)$$

The mathematical models in phase quantities given above are not usually convenient for control. It is usually convenient to transform the phase quantities to rotating reference frame quantities which is discussed in the next section.

2.2.2 Rotating transformation

In this section, the formal mathematical transformations from three phase stator quantities to their direct axis and quadrature axis components are presented .These transformations are then used to express the governing equations for a synchronous machine in terms of the so called direct-axis and quadrature-axis (dq0) quantities. The physical consideration used in the transformation is the equivalence of mmf and/or the power invariance. A transformation for transforming the quantities of a fixed frame to rotating frame is given hereunder.

Now, let S represent any of the variables (current, voltage, and flux linkage) to be transformed from the phase quantities (a-b-c) frame to dq0 frame. The transformation in matrix form is given by [5]

$$\begin{bmatrix} S_d \\ S_q \\ S_0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos\theta & \cos(\theta - 120^0) & \cos(\theta + 120^0) \\ \sin\theta & \sin(\theta - 120^0) & \sin(\theta + 120^0) \\ 0.5 & 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} S_a \\ S_b \\ S_c \end{bmatrix} \qquad (2.6)$$

Here S_0 component is called the zero sequence component, and under balanced 3-phase system this component is always zero.

Equation (2.6) is a linear transformation and hence its inverse transformation exists and is given by [5]

$$\begin{bmatrix} S_a \\ S_b \\ S_c \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 1 \\ \cos(\theta - 120^0) & \sin(\theta - 120^0) & 1 \\ \cos(\theta + 120^0) & \sin(\theta + 120^0) & 1 \end{bmatrix} \begin{bmatrix} S_d \\ S_q \\ S_0 \end{bmatrix} \qquad (2.7)$$

Now, by applying the transform of Eqn. 2.6 to voltages, flux-linkages and currents of Eqs.2.2-2.4, we get a set of simple equations as [5]:

$$v_d = R_s i_d + \frac{d}{dt}\psi_d - \omega\psi_q \tag{2.8}$$

$$v_q = R_s i_q + \frac{d}{dt}\psi_q + \omega\psi_d \tag{2.9}$$

Where

$$\psi_q = L_q i_q \tag{2.10}$$

$$\psi_d = L_d i_d + \psi_m \tag{2.11}$$

Each synchronous inductance is made up of self inductance (which includes leakage inductance) and contributions from other two phase currents. Now, a more convenient equation may result by inserting the flux linkages terms from Eqs.2.10-2.11 into Eqs.2.8-2.9 as

$$v_d = (R_s + L_d \frac{d}{dt})i_d - \omega L_q i_q \tag{2.12}$$

$$v_q = (R_s + L_q \frac{d}{dt})i_q + \omega L_d i_d + \omega\psi_m \tag{2.13}$$

Or in the matrix notation

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = \begin{bmatrix} R_s & \frac{d}{dt}L_d \\ R_s & \frac{d}{dt}L_q \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} -\omega L_q i_q \\ \omega L_d i_d + \omega\psi_m \end{bmatrix} \tag{2.14}$$

Fig. 2.2 shows a dynamic equivalent circuit of a PM synchronous machine based on Eqs.2.12-2.13. Note that in practice, magnetic circuits are subject to saturation as current increases. Especially, when I_q is increased, the value of L_q is decreased and ψ_m and L_d is subject to armature reaction. Since I_d is maintained to zero or negative value (demagnetizing) in most operating conditions, saturation of L_d rarely occurs.

For this model, instantaneous power can be derived from Eq. 2.5 via transformation as [5]

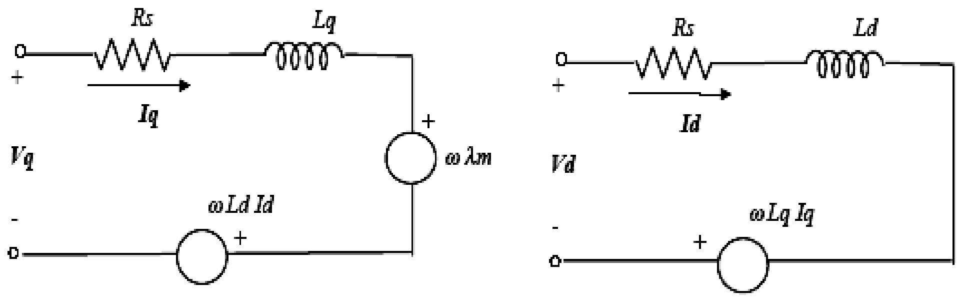
$$P_i = (3/2)[v_q i_q + v_d i_d] \quad , \tag{2.15}$$

The output power can be obtained by replacing v_d and v_q by the associated speed voltages as

$$P_o = (3/2)[-\omega\psi_q i_d + \omega\psi_d i_q] \tag{2.16}$$

2.2.3 Equivalent Circuit of Permanent Magnet Synchronous Motor

Equivalent circuits of the motors are used for study and simulation of motors. From the d-q modeling of the motor using the stator voltage equations the equivalent circuit of the motor can be derived. Assuming rotor d axis flux from the permanent magnets is represented by a constant current source, figure 2.2 is obtained.



(a) q-axis circuit (b) d-axis circuit

Figure 2.2 Equivalent Circuit of a PM Synchronous Motor

The produced torque T, which is power (eqn. (2.16)) divided by mechanical speed can be represented as

$$T_m = (3/2)\, p[\psi_m i_q + (L_d - L_q) i_q i_d] \ , \tag{2.17}$$

$$T_m = (3/2)\, p(\psi_m i_q) \ , \quad \text{for surface mounted} \tag{2.18}$$

$$= T_L + J\frac{d\omega}{dt} + B\omega$$

Chapter 3

Field weakening control of PMSM

Control of PM motors is performed using field oriented control for the operation of synchronous motor as a dc motor. The stator windings of the motor are fed by an inverter that generates a variable frequency variable voltage. Instead of controlling the inverter frequency independently, the frequency and phase of the output wave are controlled using a position sensor as shown in figure 3.1.

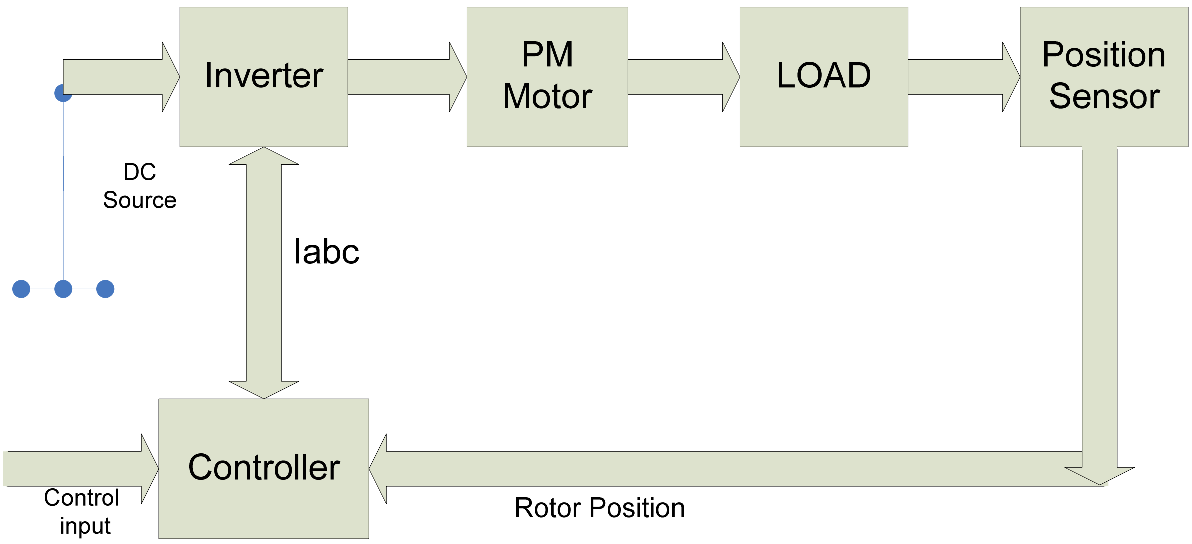


Figure 3.1 Control of Synchronous Motor

This describes the most efficient form of vector control scheme: the Field Orientated Control. It is based on three major points: the machine current and voltage space vectors, the transformation of a three phase speed and time dependent system into a two co-ordinate time invariant system. The control of AC machine in this manner acquires every advantage of DC machine control and frees itself from the mechanical commutation drawbacks. Furthermore, this control structure, by achieving a very accurate steady state and transient control, leads to high dynamic performance in terms of response times and power conversion.

Usually, high-performance motor drives require fast and accurate response, quick recovery from any disturbances, and insensitivity to parameter variations. The dynamic behavior of an ac motor can be significantly improved using vector control theory where motor variables are transformed into an orthogonal set of d-q axes such that speed and torque can be controlled separately. This gives the PMSM machine the highly desirable dynamic performance capability of the separately excited dc machine, while retaining the general advantages of the ac over dc motors.

3.1 Field Oriented Control (FOC)

Field oriented control was invented in the beginning of 1970s and it demonstrates that an induction motor or synchronous motor could be controlled like a separately excited dc motor by the orientation of the stator mmf or current vector in relation to the rotor flux to achieve a desired objective [8]. In order for the motor to behave like DC motor, the control needs knowledge of the position of the instantaneous rotor flux or rotor position of permanent magnet motor. This needs a resolver or an absolute optical encoder. Knowing the position, the three phase currents can be calculated. Its calculation using the current matrix depends on the control desired. Some control options are constant torque and flux weakening. These options are based in the physical limitation of the motor and the inverter. The limit is established by the rated speed of the motor, at which speed the constant torque operation finishes and the flux weakening starts.

The field orientated control is an efficient method to control a synchronous motor in adjustable speed drive applications with quickly changing load in a wide range of speeds including high speeds where field weakening is required.

The Field Orientated Control (FOC) consists of controlling the stator currents represented by a vector. This control is based on projections which transform a three phase time and speed dependent system into a two co-ordinate (d and q co-ordinates) time invariant system. These projections lead to a structure similar to that of a DC machine control. Field orientated controlled machines need two constants as input references: the torque component (aligned with the q co-ordinate) and the flux component (aligned with d co-ordinate). As Field Orientated Control is simply based on projections the control structure handles instantaneous electrical quantities. This makes the control accurate in every working operation (steady state and transient) and independent of the limited bandwidth mathematical model.

In the field weakening control, which is the subject of this thesis, the field component stator current (the direct-axis current) is controlled to weaken the air-gap field, simulating the field weakening by reducing field current in conventional synchronous motor.

3.2 Review of Field/Flux weakening (FW) operation

The principle of field weakening operation is exemplified by the separately excited DC commutator motor drive, which shows an ideal field-weakening characteristic. Thereafter the main types of brushless synchronous AC motors are introduced and their coherence to the separately excited DC commutator motor drive is outlined.

3.2.1 Principal of field-weakening operation

The separately excited DC commutator motor drive shows an ideal field-weakening characteristic. Therefore it is appropriate to consider the principal of field-weakening operation using this familiar motor characteristic. This is shown schematically in Figure3.2. There are two circuits, called the field circuit and armature circuit. Let $i_f(t)$ and $i_a(t)$ be the field current and armature current, respectively. Then the torque generated by the motor is given by: [5]

$$T_m(t) = K_m i_f(t) i_a(t) = J\frac{d\omega}{dt} + T_d + B\omega \qquad (3.1)$$

Where K_m is motor constant. The generated torque T_m is used to drive a load through the shaft. Assume that the total equivalent moment inertia and the friction coefficient of the load, the shaft, the gears, the rotor of the motor, etc, are J and B, respectively. Let ω be the angular speed of the motor and T_d (t) the load (or disturbance) torque. Frequently the control of motor voltage is combined with field-current control in order to achieve the widest possible speed range. With such dual control, base speed can be defined as the normal-armature-voltage, full-field speed of the motor. Speeds above base speed are obtained by reducing the field current; speeds below base speed are obtained by armature –voltage control.

Now we shall consider two different cases.

Field-Controlled DC Motors

For a field-controlled DC motor, we assume $i_a(t)$ = constant. In this case, $V_f(t)$ becomes the control signal. Let

$$K_f = K_m i_a \tag{3.2}$$

then

$$T_m(t) = K_f i_f(t) \tag{3.3}$$

The field current satisfies the following equation

$$V_f = R_f i_f + L_f \frac{di_f}{dt} \tag{3.4}$$

$$T_m(t) = K_m i_f(t) i_a(t) \tag{3.5}$$

$$V_b = K_b \Phi_f \omega \tag{3.6}$$

$$\phi_f = f(i_f) \tag{3.7}$$

where the voltage V_b is the voltage generated in the armature coil because of the motion of the coil in the motor's magnetic field and is usually called the back emf (electromotive force) voltage. V_b is proportional to the angular speed of the motor where K_b is a constant and ϕ_f is the flux in the air gap which is related to the field current by $\phi_f = f(i_f)$.

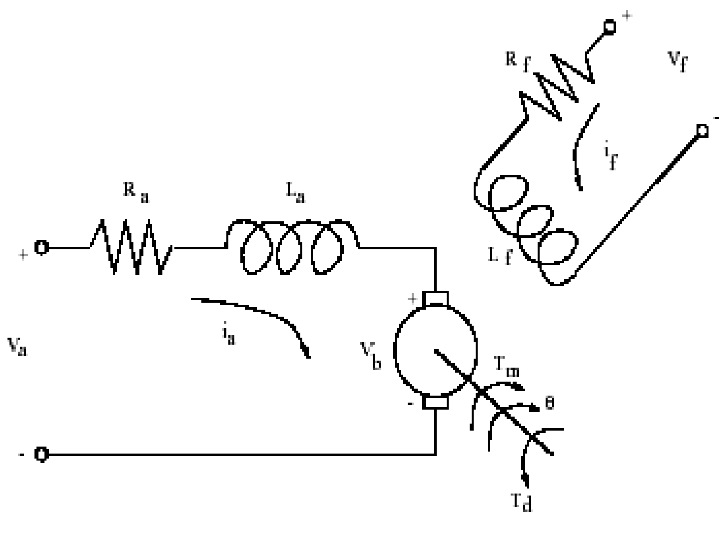


Figure 3.2 separately excited DC commutator motor drive

For increasing speeds above rated value, armature voltage may be held constant at rated level and field current may be reduced (weakened) so as to regulate armature current to rated value. Hence in this region, torque will be directly proportional to field current and speed will be inversely proportional to field current if there is no saturation. The output power will hence be constant. The torque is inversely proportional to the speed increase. As the power is constant beyond the rated speed ($P_{out} = P_b$), this is called ***constant-power region or field weakening region***. These terms are normally used interchangeably in the literature though one should be careful with the latter as it is usually possible to keep increasing the speed at a reduced power. Figure 3.3 shows the ideal field-weakening characteristics for a drive with a limited inverter volt-ampere rating capability.

Armature-Controlled DC Motors

In this case, we assume $i_f(t)$ = constant and $V_a(t)$ becomes the control signal.

Let

$$K_a := K_m i_f \qquad (3.8)$$

Then

$$T_m(t) = K_a i_a(t) \qquad (3.9)$$

The armature current satisfies the following equation

$$V_a = R_a i_a + L_a \frac{di_a}{dt} + V_b \qquad (3.10)$$

Hence for increasing speed of DC motor, up to rated level, field current may be held constant and applied voltage increased from zero to rated value so as to hold armature current at rated value. Thus at low speed, the rated armature current i_a and the rated excitation-flux linkage ψ_b are used to obtain the rated torque T_b. The voltage V_a and output power P_{out} both rise linearly with speed. Hence in this region, torque will be constant and output power will be proportional to speed. This operating range is referred to as ***constant-torque or constant-flux region.***

Figure 3.3 shows a plot of maximum power and maximum torque versus speed for a synchronous motor under variable-frequency operation.

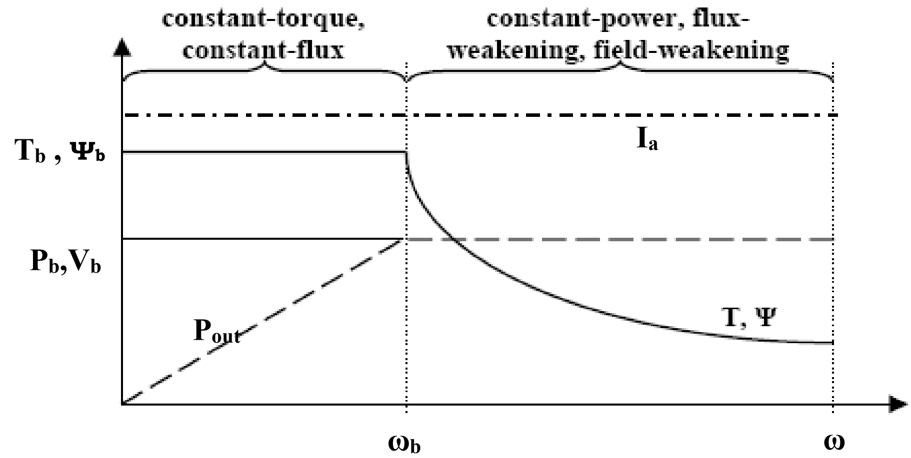


Figure 3.3 Ideal field-weakening drive characteristics [28]

Real motors do not have flat output power against speed characteristics above rated speed ωb. In Figure 3.4 the dashed line is the ideal field-weakening characteristic and the solid line is the actual characteristic. Rated power is the output power at rated speed ωb with rated torque Tb. The inverter utilization is the ratio of rated power to the ideal output power. This is less than unity as the motor does not have unity power factor and 100% efficiency under rated operating conditions. The constant-power speed range (CPSR) is the speed range over which rated power can be maintained.

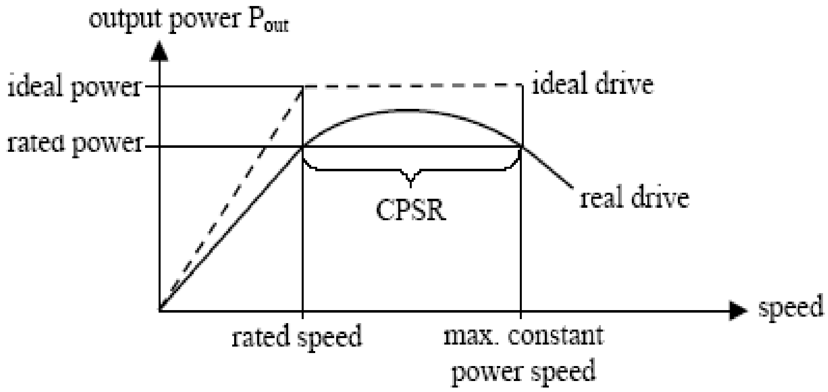


Figure 3.4 Characteristics of field-weakening parameters

3.2.2 Field-weakening operation of Permanent Magnet Motors

The separately excited DC machine has separate windings for the excitation and torque producing currents. Permanent magnet synchronous motors have a single stator winding which generates a current phasor *I*. This current phasor can be Split into the two components in d- and q-axis, I_d and I_q.

$$I^2 = I_d^2 + I_q^2 \tag{3.11}$$

In permanent magnet machines the flux is produced by magnets. Hence the magnetic (excitation) field or flux can not be controlled by varying the field current. The permanent magnets can be pictured as "fixed excitation flux" sources Ψ_m. However, flux control (or field-weakening) is achieved by introducing an opposing field Ψ_F against the fixed excitation from the magnets. It is achieved by injecting a negative d-current I_d (or field current I_F), as shown in Figure 3.5.

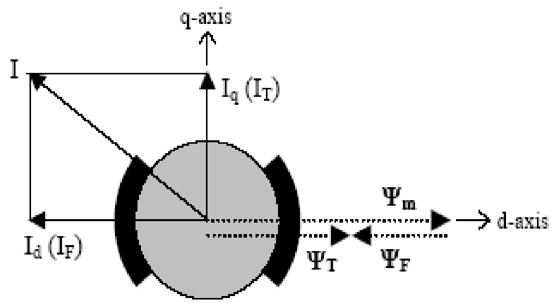
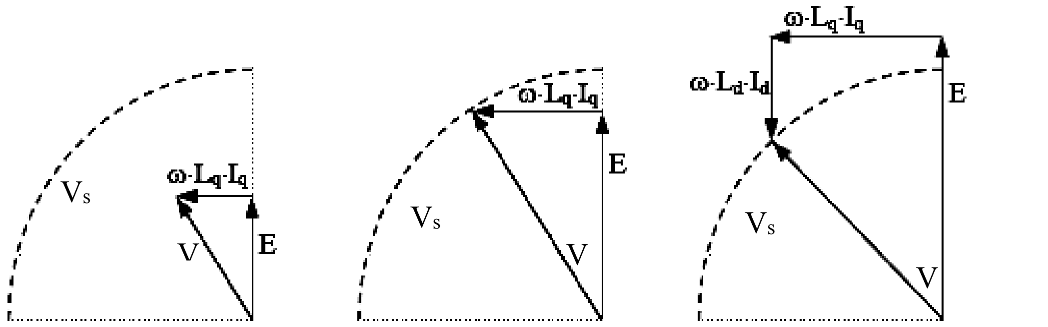


Figure 3.5 Flux-weakening of permanent magnet motor

The concept of using an imposing field can be further explained with a simple vector diagram. Figure 3.6a shows the voltage phasor diagram when the motor is running at a low speed well below the rated speed. When the motor is operated at rated conditions, as shown in Figure 3.6b , it can be noted that the voltage vector is on the voltage limit contour (maximum possible voltage Vs). It is virtually impossible to increase the speed with keeping a current I in the q-axis once the induced voltage E equals the rated voltage. In order to increase the speed beyond this limit, the current phasor can be rotated towards the negative d-axis (introduction of a negative d-axis current I_d). Figure 3.6c shows that the voltage vector V is kept within the voltage limit.



a) at low speed **b) at rated speed** **c) beyond rate speed**

Figure 3.6 Voltage phasor diagram of the PM motor at ideal conditions

The voltage limit V_s of the PM motor can be expressed as

$$V_s^2 \geq \omega^2[(\psi^2 + L_d I_d)^2 + (L_q.I_q)^2 \qquad (3.12)$$

Where ω is the electrical operating speed, Ψ_m is the magnet flux; L_d and L_q are the d-axis and q-axis synchronous inductances. The torque equation of a PM motor can generally be expressed as

$$T = (3/2).p.\psi_m I_q \qquad (3.13)$$

One can notice that the generated torque comprises two parts, the magnet torque and the reluctance torque. The total torque varies according to machine parameters as the saliency ratio ξ or the magnet thickness (defining the magnet flux Ψ_m).

The output power is from the shaft torque T and the electrical speed

$$P_{out} = T.\omega \qquad (3.14)$$

.

3.3.3 Field Weakening control realization

Two schemes are possible to implement a field weakening operation. The simplest is the standard open loop control for the d-axis current reference. The steps are shown below:

1) Measuring rotor position and speed ω from a sensor which is set in motor rotation axis.
2) The motor at the flux weakening region with a speed loop, Iq* is obtained from the PI controller.
3) Calculate Id* using equation (3.12) :

$$V_s^2 = \omega^2[(\psi^2 + L_d I_d)^2 + (L_q.I_q)^2$$

By taking the voltage phasor and current phasor the constant maximum value, where,

$$v_S = \sqrt{v_d^2 + v_q^2} \qquad (3.15)$$

$$i_S = \sqrt{i_d^2 + i_q^2} \qquad (3.16)$$

And by taking $$i_q = \sqrt{i_s^2 - i_d^2} \qquad (3.17)$$

The maximum voltage phasor is given by:

$$Vs = 2/3\ Vdc \qquad (3.18)$$

We can calculate I_q from equation (3.17) after we find I_d. i.e.

$$i_q = \sqrt{i_s^2 - i_d^2} \qquad (3.19)$$

If the Iq request from PI controller is more than the Iq calculated by the flux-weakening module, then the final I_{qref} is made equal to the calculated one. But if the Iq request from PI controller is less than the Iq calculated by the flux-weakening module, then the final Iqref is made equal to the Iq from the PI controller.

Despite the relative simplicity of realization, it has the following drawbacks:

- The reference current equation must be set in the worst condition of operation; it corresponds to the lower line voltage. It gives a low utilization of the inverter with higher voltages.
- High-speed reliability: to guarantee the correct operation of the control at high speeds it would have been necessary to reduce further the voltage capability of the inverter.
- The reference current equation depends on the motor electrical characteristics, and it's also necessary to consider the characteristics dispersion in its determination.

A closed loop control avoids these negative effects. It consists in feeding back a proportional integral (PI) regulator with the motor d and q axis voltages applied to the motor and calculate a new reference for the magnetizing current.

The realization process of equivalent flux-weakening control is as follows:

1) V_{dr} *and* V_{qr} Are used to determine the neutral-phase voltage.

$$v_r = \sqrt{v_{dr}^2 + v_{qr}^2} \qquad (3.20)$$

2) Vr and the reference value base voltage are controlled with a PI regulator to generate the output I_{dr}. In the field weakening region, $|I_{dr}|$ increases. In order not to go over the nominal stator current i_{smax}, i_{qmax} is decreased with the relation:

$$i_{q\max} = \sqrt{i_{s\max}^2 - i_{dr}^2} \qquad (3.21)$$

3) Two motor phase currents are measured. These measurements feed the Clarke transformation module. The outputs of this projection are designated i_a and i_b. These two components of the current are the inputs of the Park transformation that gives the current in the d,q rotating reference frame. The i_{Sd} and i_{Sq} components are compared to the references i_{Sdref} and i_{Sqref} which are the output of the flux weakening module. The outputs of the current regulators are v_{Sdref} and v_{Sqref}; they are applied to the inverse Park transformation. The outputs of this projection are v_{Saref} and v_{Sbref} which are the components of the stator vector voltage in the a,b stationary orthogonal reference frame. These are the inputs of the Space Vector PWM. The outputs of this block are the signals that drive the inverter. Note that both Park and inverse Park transformations need the rotor flux position. The following figure summarizes the basic scheme of field weakening control with FOC.

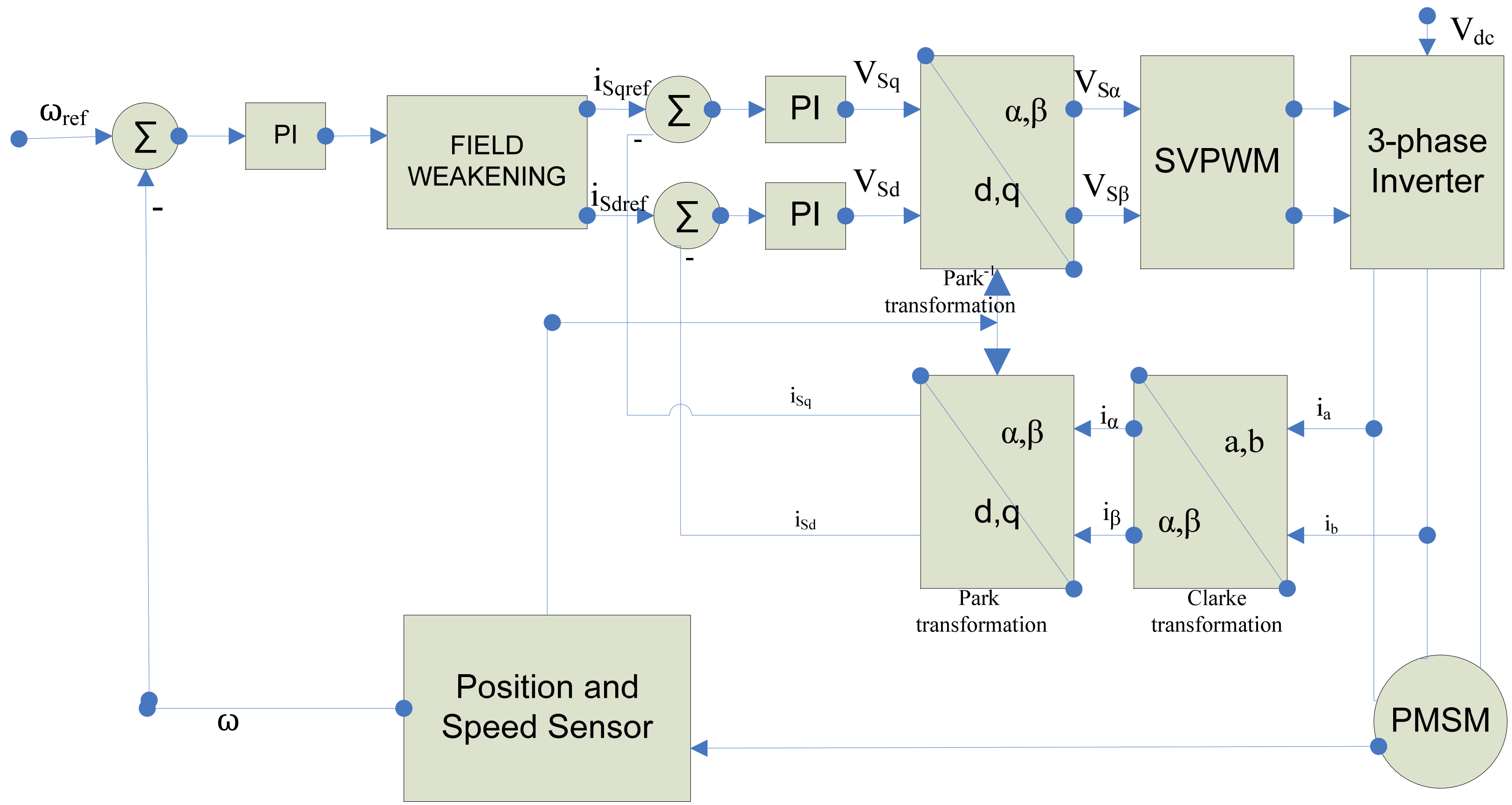


Figure 3.7 Basic scheme of FOC for PMSM-motor

Chapter 4

DSP in motor control and Software Organization

4.1 Motor control trend

Traditionally motor control was designed with analog components as they are easy to design and can be implemented with relatively inexpensive components. However, there are several drawbacks with analog systems. Aging and temperature can bring about component variation causing the system to need regular adjustment, as the part count increases, the reliability of the system decreases. Analog components raise tolerance issues and upgrades are difficult, as the design is hardwired.

Digital systems offer improvements over analog designs [10]. Drift is eliminated since most functions are performed digitally, upgrades can easily be made in software and part count is also reduced since digital systems can handle several functions on chip.

Digital Signal Processors go on further to provide high-speed, high resolution and sensor less algorithms in order to reduce system costs. Providing a more precise control to achieve better consumption or radiation performances often means performing more calculations. The use of some 1-cycle multiplication & addition instructions included in a DSP speeds-up calculations.

Generally fixed point DSPs are preferred for motor control for two reasons. Firstly, fixed point DSPs cost much less than the floating point DSPs. Secondly, for most applications a dynamic range of 16 bits is enough. If and when needed, the dynamic range can be increased in a fixed point processor by doing floating-point calculations in software.

4.2 Benefits of the DSP controllers

The performances of an AC synchronous motor are strongly dependent on its control. DSP controllers enable enhanced real time algorithms as well as sensor less control. The combination of both allows a reduction in the number of components and optimizes the design of silicon to achieve a system cost reduction [8] [12].

A powerful processor such as a DSP controller does the following: [20]

- favors system cost reduction by an efficient control in all speed ranges implying right dimensioning of power device circuits
- performs high-level algorithms due to reduced torque ripple, resulting in lower vibration and longer life time
- enables a reduction of harmonics using enhanced algorithms, to meet easier requirements and to reduce filters cost
- removes speed or position sensors by the implementation of sensorless algorithms
- decreases the number of look-up tables which reduces the amount of memory required
- real-time generation of smooth near-optimal reference profiles and move trajectories, resulting in better-performing
- controls power switching inverters and generates high-resolution PWM outputs
- provides single chip control system

4.3 DSP Development Board

The control hardware **eZdspTM LF2401** introduced by Texas Instruments has been used in this Thesis.

4.4 Features of the eZdspTM LF2401

The eZdspTM LF2401 has the following features: [21]

- TMS320LF2401A Digital Signal Processor
- Zero Insertion Force (ZIF) Socket for Processor

- 40 MIPS operating speed
- 1K words on-chip RAM
- 8K words on-chip Flash memory
- Onboard 10 MHz. CPU clock from gate array
- 2 Expansion Connectors (analog, I/O)
- Onboard IEEE 1149.1 JTAG Controller
- 5-volt only operation with supplied AC adapter
- TI Code Composer tools driver
- On board IEEE 1149.1 JTAG emulation connector
- Voltage control for parallel port JTAG FLASH programming

The eZdspTM LF2401 consists of five major blocks of logic:

- Analog Interface Connector
- I/O Interface Connector
- JTAG Interface
- Parallel Port JTAG Controller Interface
- Voltage Control

Figure 4-1 shows the layout of the LF2401 eZdsp.

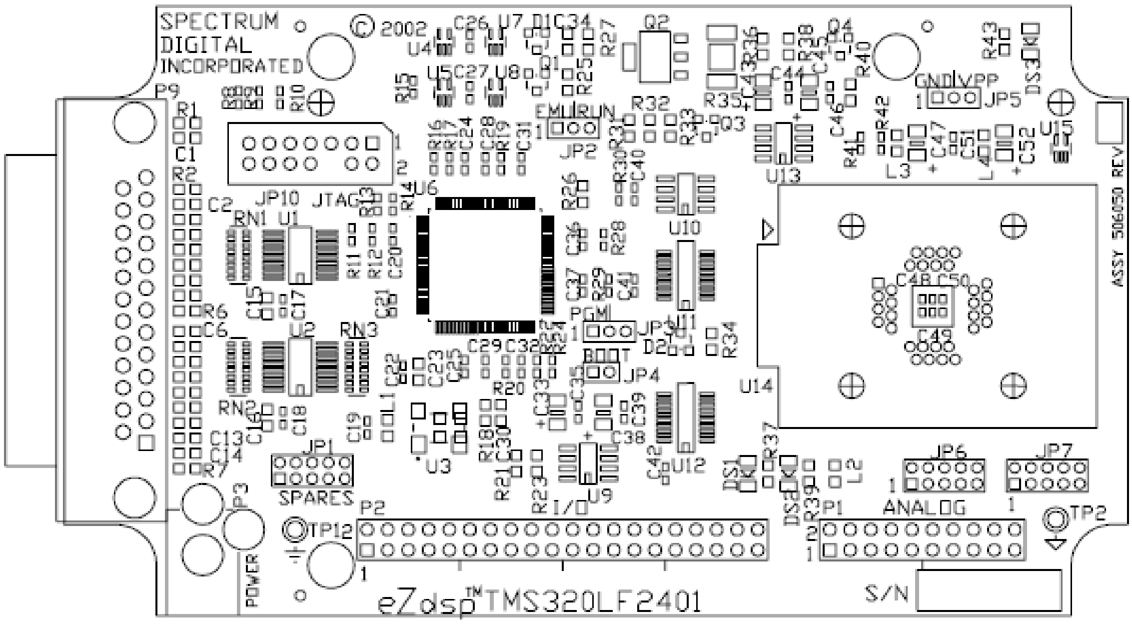


Figure 4-1 The layout of the LF2401 eZdsp.

Figure 4-2 TMS320LF2401A eZdsp Starter Kit

4.5 TMS320LF2401A Digital Signal Processor

The **TMS320LF2401A** has the following features: [18]

- High-Performance Static CMOS Technology
  − 25-ns Instruction Cycle Time (40 MHz)
  − 40-MIPS Performance
  − Low-Power 3.3-V Design
- Based on TMS320C2xx DSP CPU Core
  − Code-Compatible with 240x and F243/F241/C242
- − Instruction Set Compatible With F240
- On-Chip Memory
  − Up to 8K Words x 16 Bits of Flash EEPROM (2 Sectors) (LF2401A)
  − 8K Words x 16 Bits of ROM (LC2401A)
  − Programmable "Code-Security" Feature for the On-Chip Flash/ROM
- − Up to 1K Words x 16 Bits of Data/Program RAM

− 544 Words of Dual-Access RAM

− Up to 512 Words of Single-Access RAM

- Boot ROM

− SCI Bootloader

- Event-Manager (EV) Module (EVA), Which Includes:

− Two 16-Bit General-Purpose Timers

− Seven 16-Bit Pulse-Width Modulation (PWM) Channels Which Enable:

    − Three-Phase Inverter Control

    − Center- or Edge-Alignment of PWM Channels

    − Emergency PWM Channel Shutdown With External PDPINTA Pin

− Programmable Deadband (Deadtime) Prevents Shoot-Through Faults

− One Capture Unit for Time-Stamping of External Events

− Input Qualifier for Select Pins

− Synchronized A-to-D Conversion

− Designed for AC Induction, BLDC, Switched Reluctance, and Stepper Motor Control

- Small Foot-Print (7 mm × 7 mm) Ideally Suited for Space-Constrained Applications
- Watchdog (WD) Timer Module
- 10-Bit Analog-to-Digital Converter (ADC)

− 5 Multiplexed Input Channels

− 500 ns Minimum Conversion Time

− Selectable Twin 8-State Sequencers Triggered by Event Manager

- Serial Communications Interface (SCI)
- Phase-Locked-Loop (PLL)-Based Clock Generation
- Up to 13 Individually Programmable, Multiplexed General-Purpose Input/output (GPIO) Pins
- User-Selectable Dual External Interrupts (XINT1 and XINT2)
- Power Management:

− Three Power-Down Modes

− Ability to Power Down Each Peripheral Independently

- Real-Time JTAG-Compliant Scan-Based Emulation, IEEE Standard 1149.1† (JTAG)
- Development Tools Include:

− Texas Instruments (TI) ANSI C Compiler, Assembler/Linker, and Code Composer Studio Debugger

− Evaluation Modules

− Scan-Based Self-Emulation (XDS510)

− Broad Third-Party Digital Motor Control Support

- 32-Pin VF Low-Profile Quad Flat pack (LQFP)
- Extended Temperature Options (A and S)

− A: − 40°C to 85°C

− S: − 40°C to 125°C

The 32-pin diagram is shown below

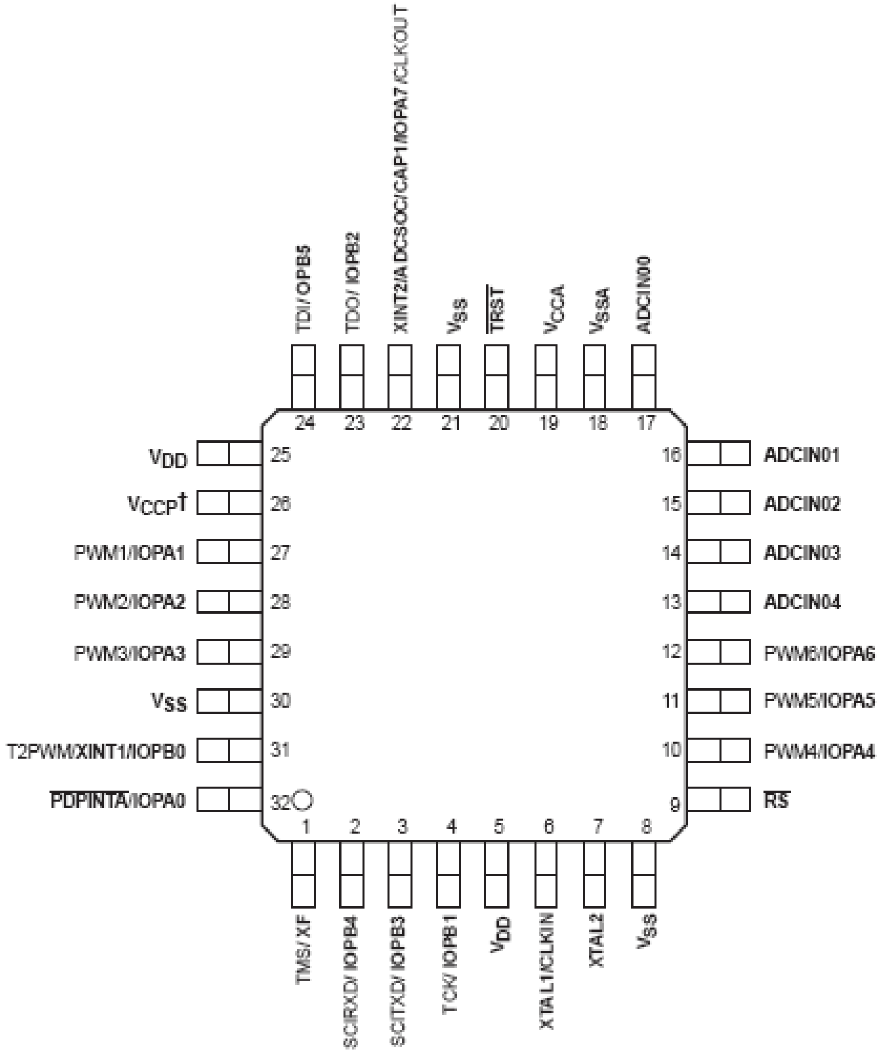


Figure 4-3 TMS320LF2401A 32-pin out diagram

4.6 FOC Software Organization

The software for the implementation of FOC of PMSM is based on two modules:

- The initialization module and
- Interrupt (Control) module

Initialization module Description

The initialization module is performed only once at the beginning. After a processor reset, the initialization module performs the following tasks: [13] [15] [17]

- DSP setup: ADC, GPIO, watchdog,Deadband, event manager, etc
- Variables initializations
- Interrupt enable
- Waiting loop

Interrupt module Description

The interrupt module is based on a waiting loop interrupted by the PWM underflow event. When this interrupt flag is set the corresponding Interrupt Service Routine (ISR) is acknowledge and serviced [13] [15] [16] [17]. This module handles the whole FOC algorithm.

The complete FOC algorithm is computed within the PWM ISR, which periodically computed according to a fixed PWM period value. In this thesis, a PWM frequency of 10 KHz has been chosen.

The sampling period (T) of 100µs (10 KHz) can be achieved by setting the timer period T1PER to 2000 (PWMPRD=07d0h).

The goal of the interrupt module is to update the stator voltage reference and to ensure the regulation of stator currents and rotor mechanical speed.

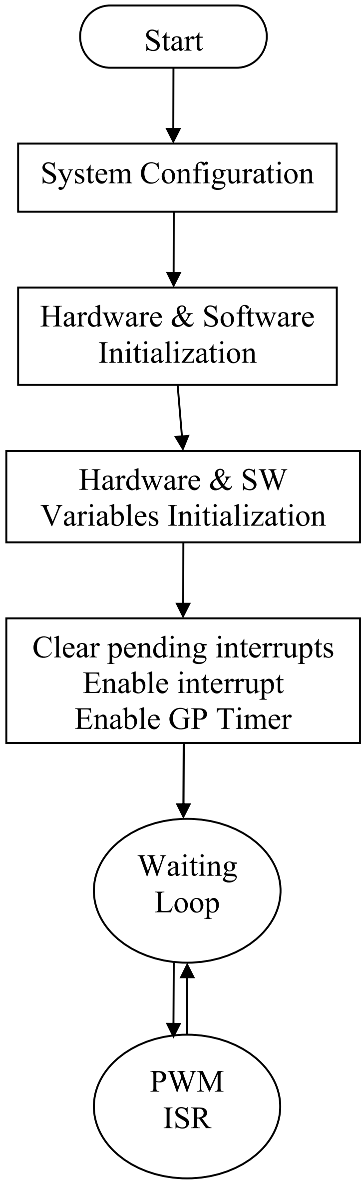


Figure 4-4 Initialization Module Flowchart

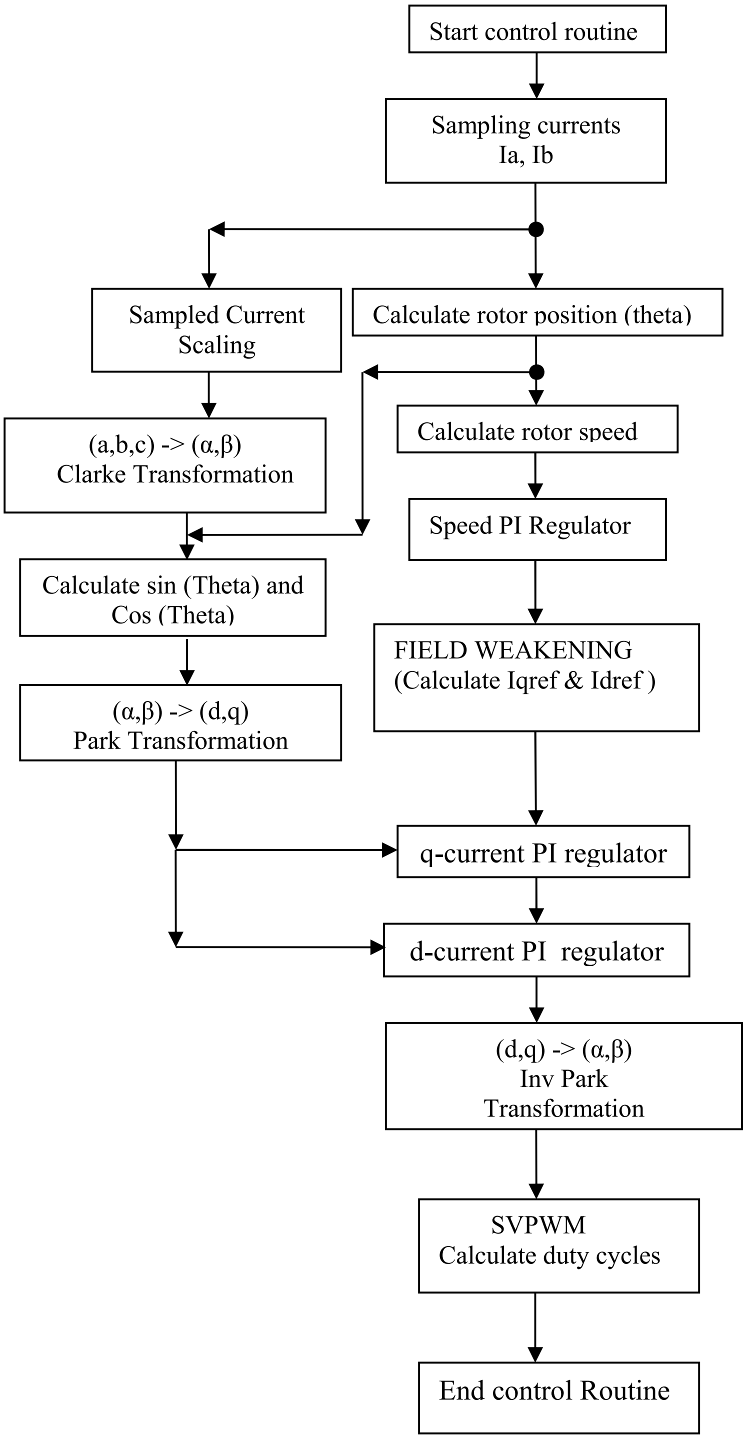


Figure 4-5 Interrupt Module Flowchart

4.6.1 PU Model and Base Values

A **per-unit system** is the expression of system quantities as fractions of a defined base unit quantity. Calculations are simplified because quantities expressed as per-unit are the same regardless of the voltage level. [7]

The Per Unit model (PU) is associated with reduced value notion. Using a fixed-point DSP, it is necessary to reduce the amplitude of the variables in order to get a fractional part with a maximum precision. The notion of Per Unit model is introduced to use this fixed-point feature.

Since the TMS320F240 is a fixed-point DSP, a per unit (p.u.) model is used for the motor variable representations. In this model all quantities are referred to base values. The advantage of this method is that it can be used for any motor (different parameters, power, user requirements, etc.) by simply changing the base values without changing any part of the software.

The base values are determined from the nominal values with the following equations:

$$I_{base} = \sqrt{2} * I_{nominal} \tag{4.1}$$

$$V_{base} = \sqrt{2} * V_{nominal} \tag{4.2}$$

$$\omega_{base} = 2 * \Pi * f_n \tag{4.3}$$

The quantities in p.u. are defined by:

$$i = \frac{I}{I_{base}} \tag{4.4}$$

$$v = \frac{V}{V_{base}} \tag{4.5}$$

$$speed = \frac{synchronous\_speed(\omega)}{\omega_{base}} \tag{4.6}$$

4.6.2 Numerical Consideration

The TMS320F240x is part of the 16bit fixed-point DSP family. The native length of a word is 16bit on this family. To represent real numbers on this fixed-point architecture, a Q12 format has chosen. I.e. 4 bits are dedicated to the integer part and 12 bits are dedicated to the fractional part.

This format has been chosen because the drive control quantities are (most of the time) not greater than four times their nominal values (in other words not greater than four when the per unit model is considered). Otherwise, a different format will be chosen. So using a representation range of [-8;8] ensures that the software values can handle each drive control quantity not only during steady state operation but during transient operation as well. In this format if the p.u. variable is 1, the correspondent word is 01000h (2^{12}=4096). This representation allows for both of the above mentioned variables to be eight times bigger than the base quantities.

4.6.3 Interrupt Modules

The interrupt modules to execute the different tasks of the FOC are described under. The modules described are:

- The input for the FOC
  - Current Sensing & Scaling Module
  - Position & Speed Sensing and Scaling Module
- Generation of Sine Θ, Cosine Θ
- Co-ordinate transformations : Clarke, Park, $Park^{-1}$
- Space Vector Modulation
- Speed regulation, current regulation
- **Field Weakening**

4.6.3.1 Current Sensing & Scaling Module

This module handles the conversion of the 3 stator phase currents into their basic binary representation.

There is a method of measuring the phase currents using a simple, cheap resistor. The method is to measure only the DC line current and then recombine the 3 phase currents using the inverter switching states. As the inverter's switching state is controlled by the Digital Signal Processor, it is possible to know the exact electrical route taken by the input current through the inverter. We can thus directly relate the phase currents to the line current. [23]

The stator currents can be expressed as follows depending on the switching states:

idc = ia when (Sa, Sb, Sc) = (1,0,0)
Idc = -ia when (Sa, Sb, Sc) = (0,1,1)
idc = ib when (Sa, Sb, Sc) = (0,1,0)
idc = -ib when (Sa, Sb, Sc) = (1,0,1)
idc = ic when (Sa, Sb, Sc) = (0,0,1)
idc = -ic when (Sa, Sb, Sc) = (1,1,0)
idc = 0 when (Sa, Sb, Sc) = (1,1,1)
idc = 0 when (Sa, Sb, Sc) = (0,0,0)

However, under certain conditions the periods are very short. In this case, due to the transistor commutation times, dead-bands and response delays of the processing electronics, the actual phase current is invisible on the line current. As a result it is not possible to estimate the phase currents from the line current under these circumstances.

In this thesis the method of obtaining the phase currents is to measure them directly. This method consists of using two shunts to measure the currents flowing in each of the inverter legs. The shunts are placed in the circuit so that the current going into and out of the inverter flow across them.

The current output needs to be rearranged and scaled so that they can be used by the control software as 4.12 format values. The complete structure of the module is described under.

4.6.3.1.1 Current Sensing

Two shunt resistors sense the phase currents. The shunt terminal voltages are sampled and converted by the TMS320F2401 Analog to Digital Converter and stored in the variable ia and ib. In this application the shunt output signal can be either positive or negative. This signal thus needs to be translated in the range of (0; 3.3V) by the analog interface to allow the single voltage ADC module to read positive and negative values. To do this a voltage shift of 1.65V is applied to meet the [0, 3.3V] input range of the ADC. This 1.65V analog offset is digitally subtracted from the conversion result, giving thus a signed integer value of the sensed current.

The TMS320LF2401A has 10-Bit Analog-to-Digital Converter (ADC). So the output value of the ADC is (0, 1023), which is (-Imax, Imax) <=> (0, 3.3V).

The selected ADC inputs pins are ADCIN00 for phase a, and ADCIN01 for phase b. The voltages Vadcin0 and Vadcin8 are sample and converted by the dual 10 bit ADC. The result of the conversion is stored in binary format in the variables ia and ib.

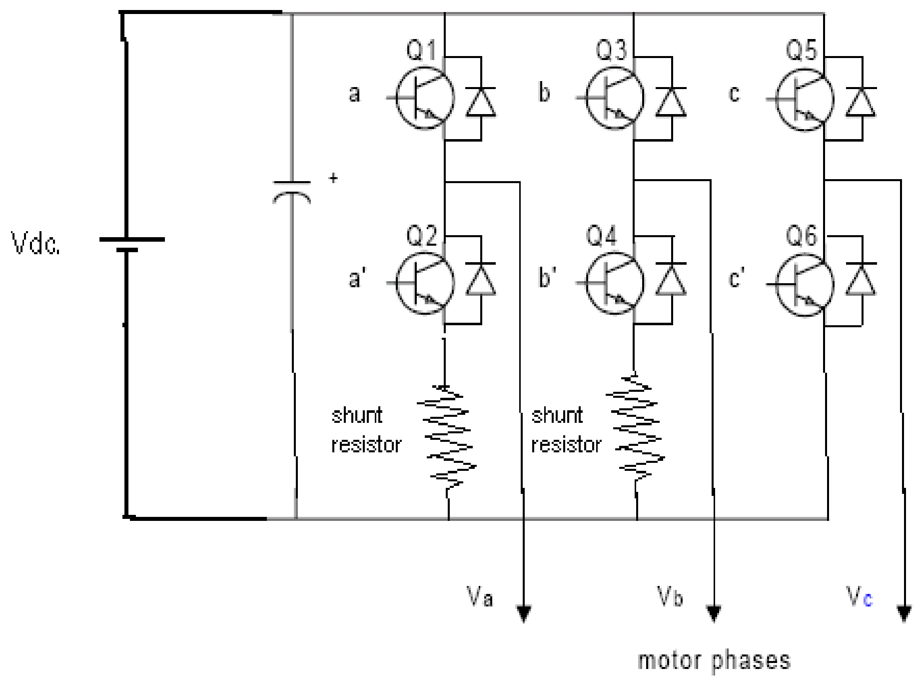


Figure 4-6 Inverter with two shunts for current measurement

Interrupt Module

Once that the AD Converter is correctly set, it will be used during every PWM interrupt to sample and convert the stator phase currents. A conversion is done as follows.

4.6.3.1.2 Current scaling

The binary representation of the sensed phase currents must now be expressed with the p.u. model and then be converted into the 4.12 format. To scale the current, we need to find a scaling factor K that makes the correspondence between the binary representation of the currents and their Q12 representation associated to the PU model of the currents. In other words, the aim of the translation is to find a factor K such as:

$$I_{PUQ12} = I_{binary} * K \qquad (4.7)$$

For I_{binary}= 512, that is I_{max}:

$$I_{PUQ12} = \frac{I_{\max}}{I_{base}} * 2^{12} \qquad (4.8)$$

In this thesis, I_{max}=1.3A & I_{base}=0.765A. So that,

$$I_{PUQ12} = 1.7 * 4096 = 6963$$

This implies that, $K = \frac{I_{PUQ12}}{I_{binary}} = 13.6$

The DSP is not able to perform directly the multiplication of a binary by a real number as it is a fixed-point device. The real operation performed is to multiply by the Q12 fixed-point representation of Kcurrent. Finally the result of the multiplication is divided by 2^{12} to contradict the Q12 multiplication.

In the application software, the constant K is called Kcurrent and its representation in Q12 is given by: Kcurrent = 13.6 $*2^{12}$ = 55,706 <=> D99Ah (Q12).

The following procedure has been followed for current scaling:

First the upper bits of the accumulator are masked (the binary representation of the sensed current which is 10 bit). Since the ADC result has only 10 bit, the masking helps to keep the binary representation of the variables.

As discussed in current sensing module, we now subtract 512 from the sensed current to contradict the analog offset of 1.65V that was previously introduced. +1.3A is now represented in binary by +512 and –1.3A by -512.

4.6.3.2 Position & Speed Sensing and Scaling Module

The position/speed sensor in this thesis is prepared locally for the thesis by integrating laser sensor (Transmitter and Receiver) and 128 slots encoder. The receiver sensor is activated by the light of an internal LED. When the light is hidden, the sensor sends a logical “0”. When the light passes through one of the 128 slots of the encoder, a logical “1” is sent.

Figure 4-7 128 slot position encoder

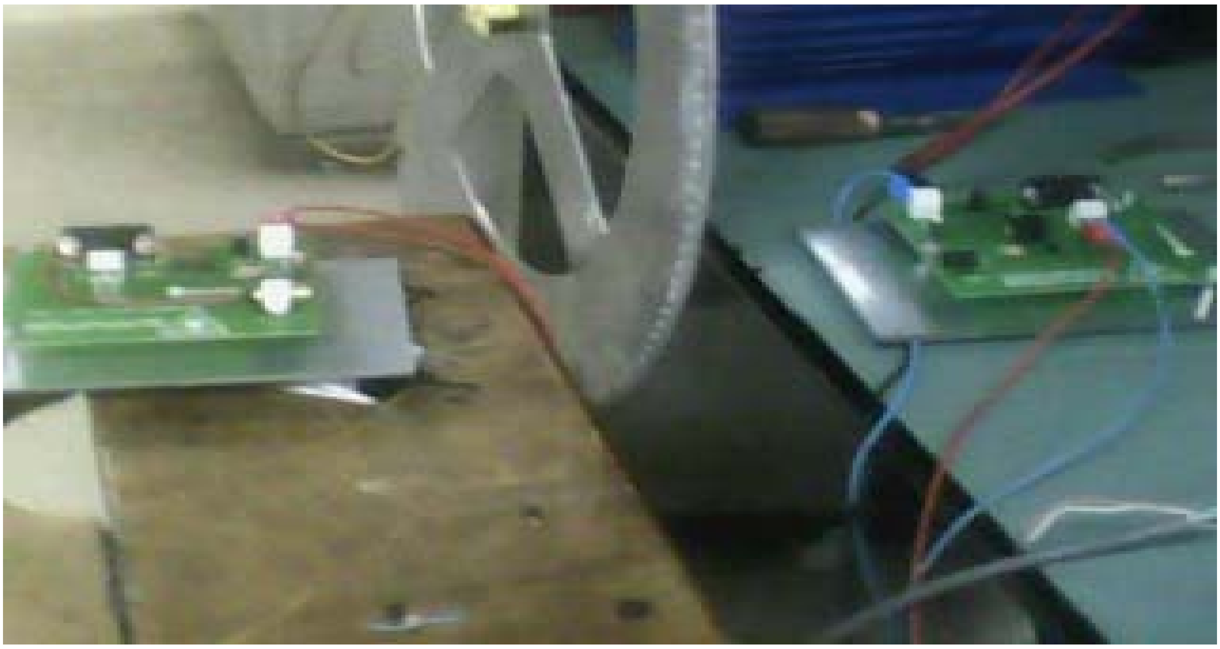


Figure 4-8 laser sensors with position encoder

The DSP Capture unit detects the rising and falling edges of the sensor. The count of the edges detected by the Capture unit is stored in the register CAPCOUNT which I use it as a counter. And the software converts the number of pulses sent by the encoder into position and speed of the rotor shaft.

The position encoder in this thesis generates 128 pulses per mechanical revolution. Every slot generates 2 edges: 1 rising and 1 falling edge. These edges are detected by the capture unit, meaning that 256 edges are detected per mechanical revolution.

4.6.3.2.1 Position Sensing

The relative mechanical angular displacement calculated between two sampling period is equal to:

$$\Delta\theta = \frac{CAPCOUNT(t) - CAPCOUNT(t - \Delta t)}{encpulses} * 360^\circ \qquad (4.9)$$

Where encpulses is equal to 256.

The absolute mechanical position is computed every sampling period as follows:

$$\theta(t) = \theta(t - \Delta t) + \Delta\theta \qquad (4.10)$$

A software rollover is used when the calculated angle exceeds 360°.

4.6.3.2.2 Speed Sensing & Scaling

The mechanical speed is computed periodically to provide a feedback to the PI speed regulator. The update of the speed information is not as critical as the update of the currents. The reason is that the mechanical response time constant of the system is very slow compared to the electrical one. Therefore, the mechanical speed is updated on a lower time base than the electrical quantities. The speed regulation loop frequency is realized in this thesis using a software counter. This counter increment on PWM interrupts. Its period is the software variable called SPEEDSTEP. The counter variable is named speedstep. When speedstep is equal to SPEEDSTEP, the number of counted pulses is stored in another variable called speedtmp and multiplied by Kspeed to get the motor speed.

In this thesis the value of SPEEDSTEP is 20, i.e. the mechanical speed is updated every 20 PWM interrupts.

Kspeed is the constant that multiplies the encoder increment to calculate the real speed. To calculate Kspeed I use the following method:
First I calculate how much time will be elapsed when the counter reaches SPEEDSTEP, i.e.
SPEEDSTEP*PWMPRD=20*100us = 2msecond.
As the base speed is 3000RPM, it means that the motor revolves 50 revolution per second, this implies that 50*256 pulses= 12,800 pulses per second. So, speedtmp=12,800pulses/second*2msecond=26pulses.

Finally, Kspeed*speedtmp= 4096 (which is base speed value)

Kspeed= 157.5 which is 09D8h in 12.4 format

4.6.3.3 Generation of Sine and Cosine values

The Coordinate Transformations (Park, Park-1) use the value of the rotor electrical position in order to handle the stator current vector projection in a rotating frame. The electrical position is not directly used in this transforms but the sine and cosine values of this electrical position.

To obtain both sine and cosine from the electrical angle, a sine look-up table and indirect addressing mode by auxiliary register have been implemented. The table contains 256 words to represent sine values of electrical angles in the range [0; 360°] (the table is shown in the appendix).

θ varies from 0 to 4095 (which is after sensing and scaling the position sensor). As only 256 words are available to represent this range, Θ is stored into the variable **index** that will be used to address the lookup table.

The content of the table raw pointed by **index + Sine table address** is fetched in indirect addressing mode via auxiliary register. This content is stored in the variable sin Θ that will be used in the Park transforms.

To have the cosine value of the electrical angle, 90° are added to θ, this operation is corresponds to add 64 (256/4) = 40h to the value of index. The result is stored in the variable cosθ.

4.6.3.4 Co-ordinate transformations: Clarke, Park, Park^{-1}

The FOC consists of controlling the components of the motor stator currents, represented by a vector, in a rotating reference frame d,q aligned with the rotor flux. The vector control system requires the dynamic model equations of the PMSM motor and returns the instantaneous currents and voltages in order to calculate and control the variables. [11]

The Clarke transform uses three-phase currents ia, ib and ic to calculate currents in the two-phase orthogonal stator axis: i_α and i_β. These two currents in the fixed coordinate stator phase are transformed to the isd and isq currents components in the d,q frame with the Park transform.

4.6.3.4.1 Clarke transformation

The stator phase current vector is projected from a 3-phase (a,b,c) system in a (α,β) non-rotating frame by the Clarke transformation. The mathematical equations are:

$$i_\alpha = i_a \tag{4.11}$$

$$i_\beta = \frac{1}{\sqrt{3}} i_a + \frac{2}{\sqrt{3}} i_b \tag{4.12}$$

$$i_a + i_b + i_c = 0 \tag{4.13}$$

4.6.3.4.2 Park transformation

The two phases α, β frame representation calculated with the Clarke transform is then fed to a vector rotation block where it is rotated over an angle q to follow the frame d,q attached to the rotor flux.

The rotation over an angle θ is done according to the formulas:

$$i_d = i_\alpha * \cos\theta + i_\beta * \sin\theta \quad (4.14)$$

$$i_q = -i_\alpha * \sin\theta + i_\beta * \cos\theta \quad (4.15)$$

4.6.3.4.3 Park^{-1} transformation

The vector in the d, q frame is transformed from d, q frame to the two phases α, β frame representation calculated with a rotation over an angle θ according to the formulas:

(4.16)

$$v_\beta = v_d * \sin\theta + v_q * \cos\theta \quad (4.17)$$

4.6.3.5 Space Vector Modulation

The Space Vector Modulation is used to generate the voltages applied to the stator phases. It uses a special scheme to switch the power transistors to generate pseudo sinusoidal currents in the stator phases. This switching scheme comes from the translation of the (α, β) voltage reference vector into an amount of time of commutation (on/off) for each power transistors. [22]

There are eight possible combinations of on and off patterns for the three upper power transistors that feed the three phase power inverter. Notice that the on and off states of the lower power transistors are opposite to the upper ones and so are completely determined once the states of the upper power transistors are known. It has been shown to generate less harmonic distortion in the output voltages and or currents applied to the phases of an AC motor and provide more efficient use of supply voltage in comparison with direct sinusoidal modulation technique. The eight voltage vectors defined by the combination of the switches are represented in the following Figure.

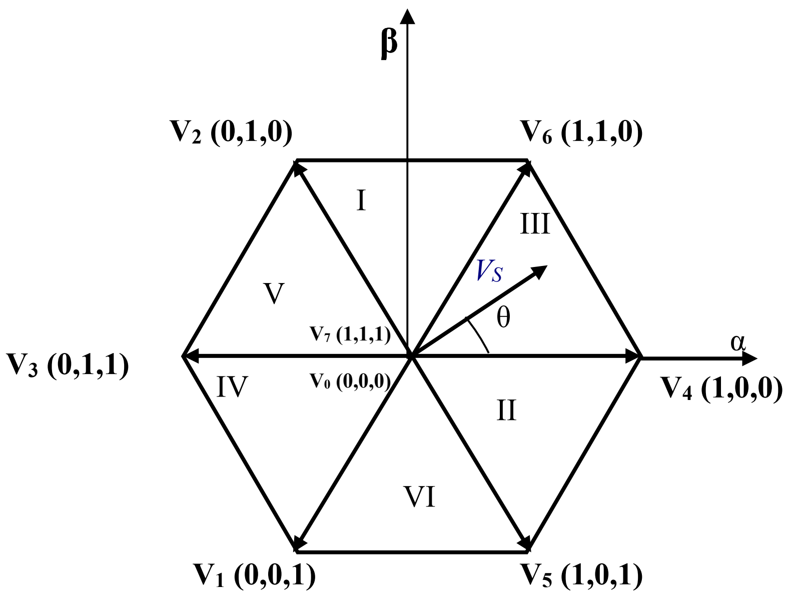


Figure 4.9 Inverter voltage space vector [9]

Due to the fact that all three Line-to-Neutral voltages must sum to zero, a space vector VS is easily calculated by the formula

$$V_a = V\cos(\omega t) \qquad (4.18)$$

$$V_b = V\cos(\omega t + \frac{2\pi}{3}) \quad (4.19)$$

$$V_C = V\cos(\omega t + \frac{4\pi}{3}) \quad (4.20)$$

$$V_S = V_{an}.e^{j0} + V_{bn}.e^{2\pi/3} + V_{cn}.e^{-j2\pi/3} \quad (4.21)$$

Where, V_{an}, V_{bn} and V_{cn} are the phase voltages of the inverter output.

The phase to neutral voltage of the inverter output for the three phases is given by

$$V_{an} = (1/3)(2*V_{ao} - V_{bo} - V_{co})$$

$$V_{bn} = (1/3)(2*V_{bo} - V_{ao} - V_{co}) \quad (4.22)$$

$$V_{cn} = (1/3)(2*V_{co} - V_{ao} - V_{bo})$$

The expression of the 3 phase voltages in the (α, β) frame are given by the general Clarke transform equation:

$$\begin{bmatrix} V_{S\alpha} \\ V_{S\beta} \end{bmatrix} = \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}\begin{bmatrix} V_{an} \\ V_{bn} \\ V_{cn} \end{bmatrix} \quad (4.23)$$

The eight combinations and the derived output phase voltages and the 3 phase voltage in terms of DC supply voltage Vdc in the (α, β) frame, according to equations (4.21) and (4.22), are shown in Table 4.1.

a	**b**	**c**	V_{AO}	V_{BO}	V_{CO}	V_{AN}	V_{BN}	V_{CN}	$V_{S\alpha}$	$V_{S\beta}$	
0	0	0	-1/2	-1/2	-1/2	0	0	0	0	0	V_0
0	0	1	-1/2	-1/2	1/2	-1/3	-1/3	2/3	-1/3	$-1/\sqrt{3}$	V_1
0	1	0	-1/2	1/2	-1/2	-1/3	2/3	-1/3	-1/3	$1/\sqrt{3}$	V_2
0	1	1	-1/2	1/2	1/2	-2/3	1/3	1/3	-2/3	0	V_3
1	0	0	1/2	-1/2	-1/2	2/3	-1/3	-1/3	2/3	0	V_4
1	0	1	1/2	-1/2	1/2	1/3	-2/3	1/3	1/3	$-1/\sqrt{3}$	V_5
1	1	0	1/2	1/2	-1/2	1/3	1/3	-2/3	1/3	$1/\sqrt{3}$	V_6
1	1	1	1/2	1/2	1/2	0	0	0	0	0	V_7

Table 4.1 Power bridges output voltages and stator voltages

The nonzero vectors form the axes of a hexagon as shown in Figure4-9. The angle between any adjacent two non-zero vectors is 60 degrees. The zero vectors are at the origin and apply zero voltage to a motor. The eight vectors are called the basic space vectors and are denoted by V1 , V2 , V3 ,V4,V5 ,V6 ,V0 and V7.The same transformation can be applied to the desired output voltage to get the desired reference voltage vector VS in the d-q plane. The objective of space vector PWM technique is to approximate the reference voltage vector VS by a combination of the eight switching patterns. One simple means of approximation is to require the average output of the inverter (in a small period, T) to be the same as the average of VS in the same period.

4.6.3.5.1 Projection of the Stator Reference Voltage Vs

The method used to approximate the desired stator reference voltage with only eight possible states of switches is to combine adjacent vectors of the reference voltage and to modulate the time of application of each adjacent vector.

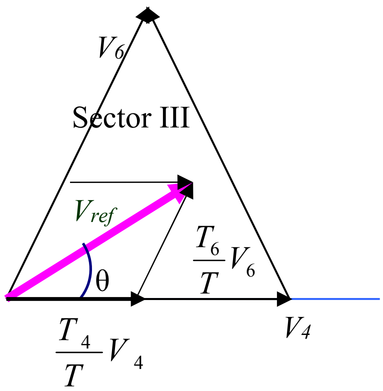


Figure 4.10 Projection of the Reference Voltage Vector

In the above Figure, the reference voltage Vref is in the third sector and the application time of Each adjacent vector is given by:

$$T = T_4 + T_6 + T_0 \qquad (4.24)$$

$$V_{ref} = \frac{T_4}{T}V_4 + \frac{T_6}{T}V_6 \qquad (4.25)$$

For every sector, commutation duration is calculated. The amount of times of vector application can all be related to the following variables:

$$X = \sqrt{3}V_{DCinvT}V_{\beta ref} \qquad (4.26)$$

$$Y = \frac{\sqrt{3}}{2}V_{DCinvT}V_{\beta ref} + \frac{3}{2}V_{DCinvT}V_{\alpha ref} \qquad (4.27)$$

$$Z = \frac{\sqrt{3}}{2}V_{DCinvT}V_{\beta ref} - \frac{3}{2}V_{DCinvT}V_{\alpha ref} \qquad (4.28)$$

In order to know which of the above variable apply, the knowledge of the sector in which the reference voltage vector is, needed. To determine this sector, a simple approach is to calculate the projections Va, Vb and Vc of the reference voltage vector in the (a,b,c) plane. These projections are then compared to 0.

4.6.3.5.2 Space Vector Algorithm

The first step is to determine in which sector the voltage vector defined by $V_{\alpha\ ref}$, $V_{\beta\ ref}$ is found. The following few code lines give the sector as output [9] [30]:

Sector determination

If $V_a > 0$ *Then* $A=1$ *else* $A=0$
If $V_b > 0$ *Then* $B=1$ *else* $B=0$
If $V_c > 0$ *Then* $C=1$ *else* $C=0$
Sector $= A+2B+4C$

Depending on the sector, two adjacent vectors are chosen. The binary representations of two adjacent basic vectors are different by only one bit, so that only one of the upper transistors switches when the switching pattern switches from one vector to the adjacent one. The two vectors are time weighted by [t1, t2] in a sample period T to produce the desired output voltage.
The second step is to calculate and saturate the duration of the two sector boundary vectors application as shown below:

Case sector of

1 $t_1 = Z$, $t_2 = Y$
2 $t_1 = Y$, $t_2 = -X$
3 $t_1 = -Z$, $t_2 = X$
4 $t_1 = -X$, $t_2 = Z$
5 $t_1 = X$, $t_2 = -Y$
6 $t_1 = -Y$, $t_2 = -Z$

The next step is to perform saturation control of t1 and t2 as shown below:

$$IF\ (t_1 + t_2) > PWMPRD$$

THEN

$$t_{1\,sat} = t_1 \frac{PWMPRD}{t_1 + t_2}$$

$$t_{2\,sat} = t_2 \frac{PWMPRD}{t_1 + t_2}$$

The third step is to compute the three necessary duty cycles. This is shown below:

$$t_{aon} = \frac{PWMPRD - t_1 - t_2}{2}$$

$$t_{bon} = t_{aon} + t_1$$

$$t_{con} = t_{bon} + t_2$$

The last step is to give the right duty cycle (txon) to the right motor phase (in other words, to the right CMPRx) according to the sector. The table below depicts how this is determined.

Sector	**1**	**2**	**3**	**4**	**5**	**6**
CMPR1	t_{bon}	t_{aon}	t_{aon}	t_{con}	t_{con}	t_{bon}
CMPR2	t_{aon}	t_{con}	t_{bon}	t_{bon}	t_{aon}	t_{con}
CMPR3	t_{con}	t_{bon}	t_{con}	t_{aon}	t_{bon}	t_{aon}

Table 4.2 Assigning the Right Duty Cycle to the Right Motor Phase

4.6.3.6 Current and speed regulation

The PI controller is an effective means of regulating torque and voltage magnitudes to the desired values. It also improves the steady state error. This is achieved by providing a gain for the error term with an integral component correction.

Kp is the proportional gain and Ki is the integral gain of the feedback loop.

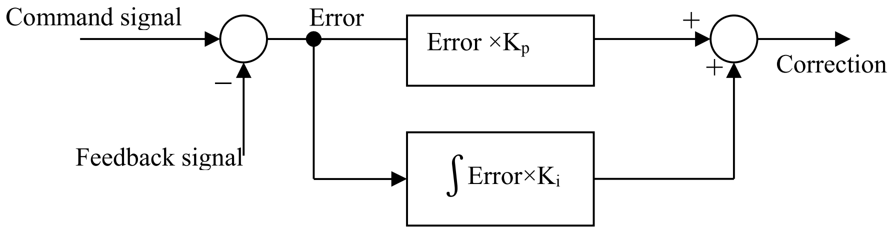


Figure 4.11 PI Controller

The Field Oriented Control of a PMSM needs three PI Controller. One for each current in rotating frame (I_d and I_q) and one for the angular speed. (See Figure 3.7)

The structure of a PI speed controller is shown in Figure 4-12.

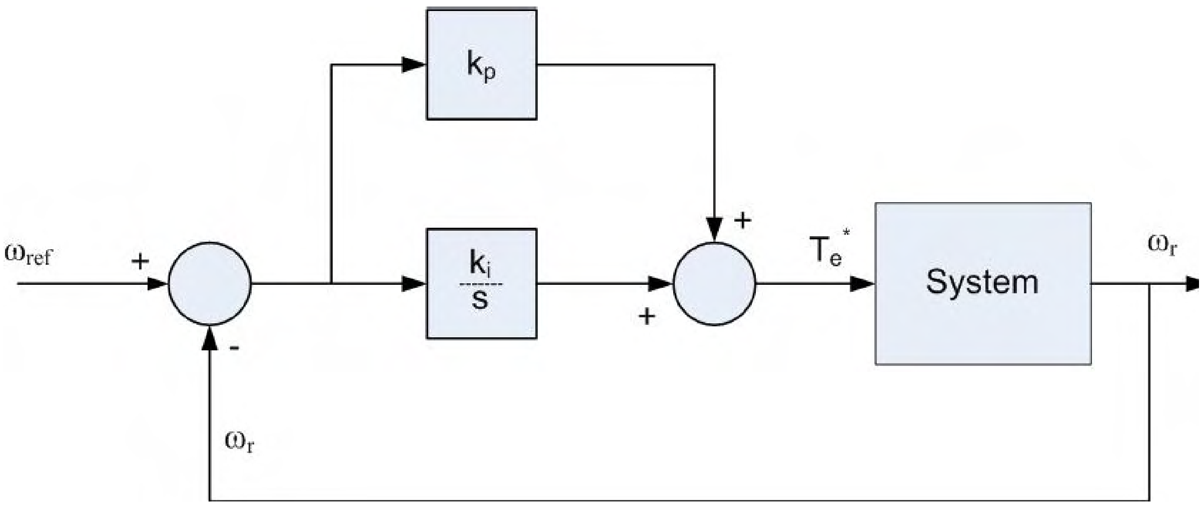


Figure 4.12 Structure of PI regulator

4.6.3.7 Field Weakening

The procedure for implementing field weakening was described in section 3.3.3. The square root of (Vdr2 +Vqr2) is calculated in real time using the Newton-Raphson method with the recursive equation: [30]

$$X(n) = 0.5 * (X(n-1) + \frac{N}{X(n-1)} \qquad (4.29)$$

4.6.4 Gate Driver

There are many situations where signals and data need to be transferred from one subsystem to another within a piece of electronics equipment, or from one piece of equipment to another, without making a direct ohmic electrical connection. The DSP PWM (pulse width modulation) outputs are isolated from the power board by Optocouplers.

Optocoupler is a combination of a light source and a photosensitive detector. In the optocoupler, or photon coupled pair, the coupling is achieved by light being generated on one side of a transparent insulating gap and being detected on the other side of the gap without an electrical connection between the two sides.

The DSP PWM (pulse width modulation) outputs are isolated from the IGBT inverter (Type SKM 75GB128D) by Gate Driver which uses Optocouplers.

Chapter 5

Simulation Results & DSP processor Outputs

5.1 Drive System Simulations in Simulink

5.1.1 Simulation Tools

In order to validate the mathematical analysis and, hence, to establish the effectiveness of the proposed field weakening control scheme, the performance of PMSM drive based on the proposed control scheme is investigated by simulation. Therefore this section of the thesis is concerned with formulating a simulation using SIMULINK that will be used to develop the closed loop control of the system under field weakening.

SIMULINK is a toolbox extension of the MATLAB program. It is a program for simulating dynamic systems. Simulink has the advantages of being capable of complex dynamic system simulations, graphical environment with visual real time programming and broad selection of tool boxes. The simulation environment of Simulink has a high flexibility and expandability which allows the possibility of development of a set of functions for a detailed analysis of the electrical drive. Its graphical interface allows selection of functional blocks, their placement on a worksheet, selection of their functional parameters interactively, and description of signal flow by connecting their data lines using a mouse device. System blocks are constructed of lower level blocks grouped into a single maskable block. Simulink simulates analogue systems and discrete digital systems.

5.1.2 Simulation of PMSM Drive

The PM motor drive simulation was built in several steps like the Co-ordinate transformation, and control circuit. The coordinate transformation variables are built using Clarke, Park and Park^{-1} transformation.

The inverter is implemented in Simulink. The inverter consists of the "universal bridge" block from the power systems tool box with the parameter of the IGBT. The voltages and currents in the motor and in all the devices of the inverter can be obtained. For proper control of the inverter using the reference currents, SVPWM is implemented to generate the gate pulses for the IGBT's.

The motor parameters used for simulation are given in table 7.1. These parameters were taken from MATLAB SIMULINK models

Table 5.1 Permanent Magnet Motor Parameters (Motor specification)

dc link voltage	*300V*
Inertia	*0.001469Kg.m^2*
magnetic flux linkage	*0.171Wb*
pole pairs	*4*
Rated Power	*10KW*
Mechanical speed(rated)	*2300RPM*
stator resistance	*0.4578ohm*
q-axis inductance	*0.00334H*
d-axis inductance	*0.00334H*
Ismax	*30A*
friction coefficient	*0.0003035Nms*

Using all the drive system blocks the complete system block has been developed as shown in figure 5.1.

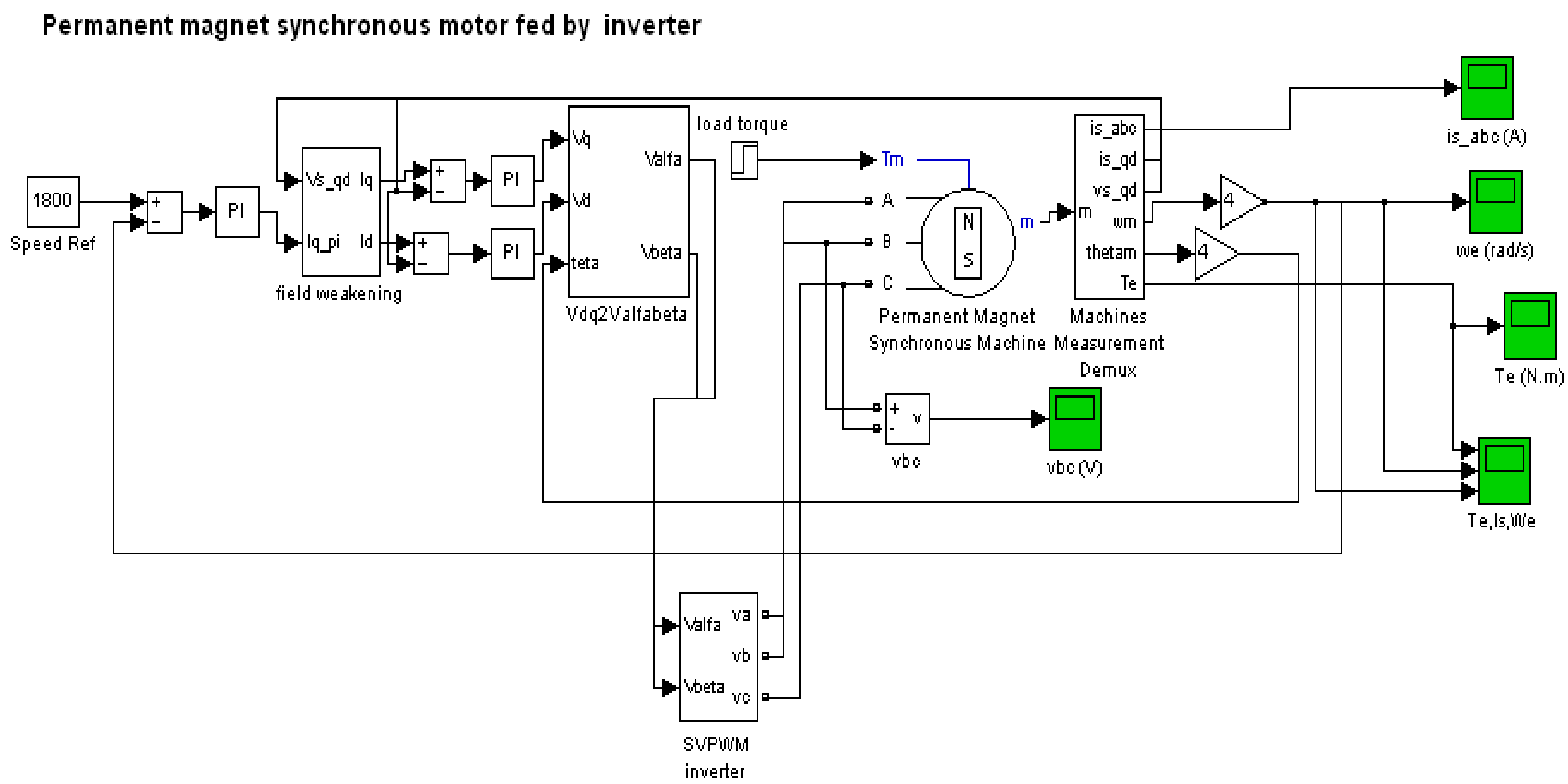


Figure 5.1 PM Motor Drive System for flux weakening operation in Simulink

5.1.3 Simulation Results

The system built in Simulink for a PMSM drive system has been tested with SVPWM control methods at flux-weakening regions of operation. The motor is operated with constant torque up to its rated speed and beyond that rated speed flux-weakening mode is adopted. Simulation results are given at electrical speed of 1800 radians per second (The rated electrical speed is 1000 radians per second). The above speeds represent above rated speed of the motor.

The simulation was carried out using current control techniques to study the performance of the motor drive. The technique is SVPWM. The plots of current, torque and speed are given under. The torque Vs speed graph is also presented to compare the characteristic of the PMSM drive system with and without field-weakening application.

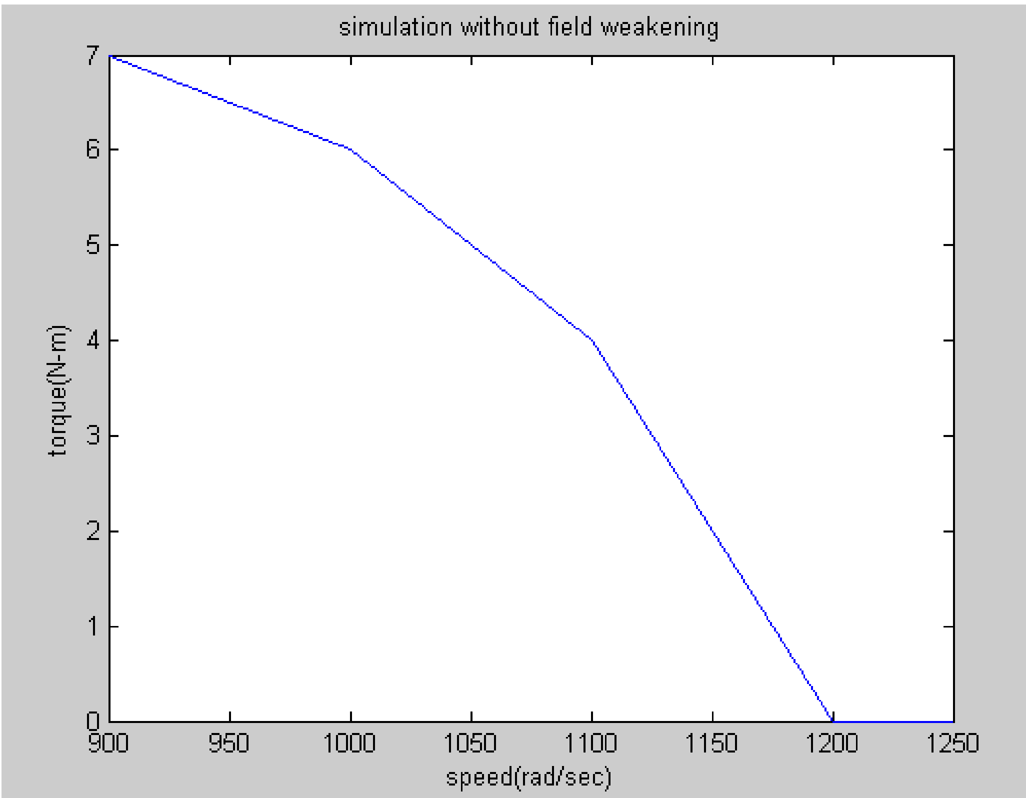


Figure 5. 2 torque versus speed characteristics without field weakening

Figure 5.2 demonstrates the synchronous motor control without the application of field weakening control. The base speed is about 1000 rad/sec. The motor can generate torque only up to around 1200 rad/sec at which the torque dies to zero. But, when we apply field weakening control, the motor can generate torque in wider speed range, up to around 2600 rad/sec, which is more than two times the rated speed as it shows in figure 5.3.

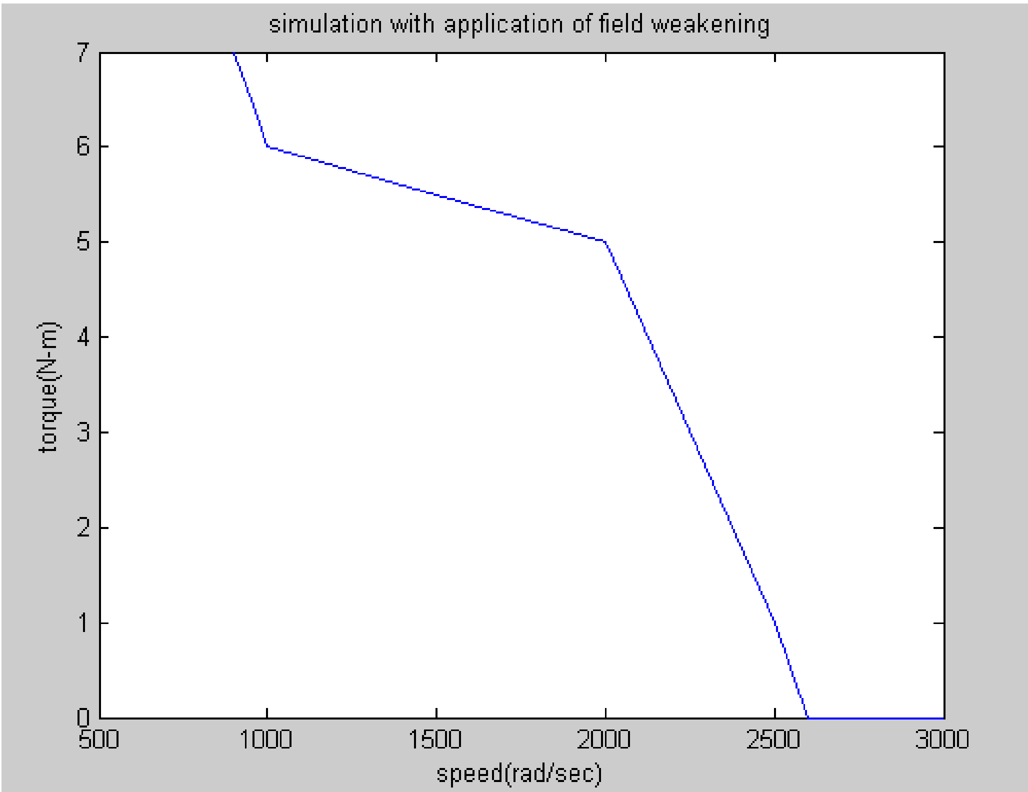


Figure 5. 3 Torque versus speed characteristics with the application of field weakening

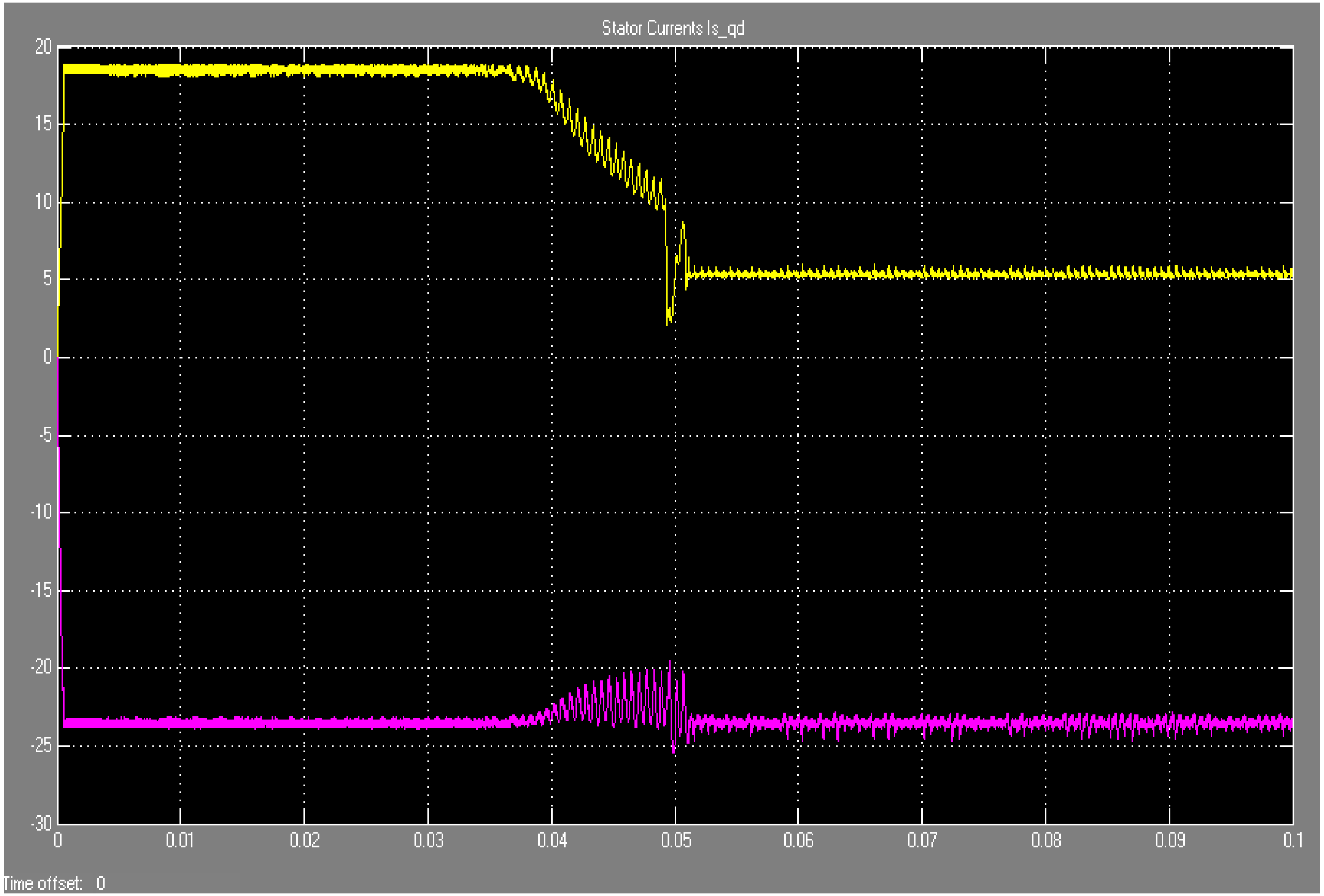


Figure 5.4A Is_qd Currents (in ampere) versus Time (in second) at 1800 rad/s (with application of field weakening)

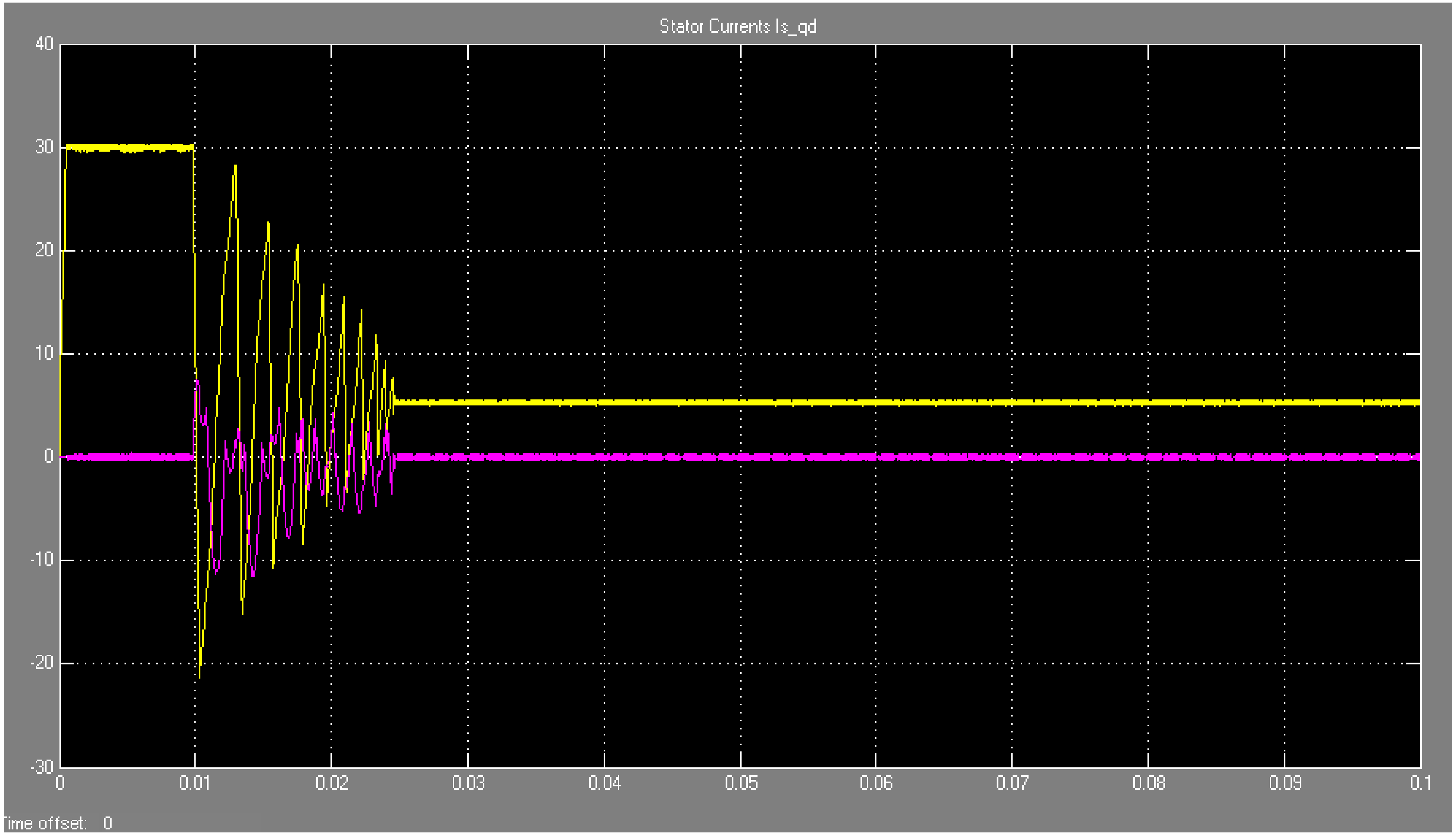


Figure 5. 4B Is_qd Currents (in ampere) versus Time (in second) at 1000 rad/s (without application of field weakening)

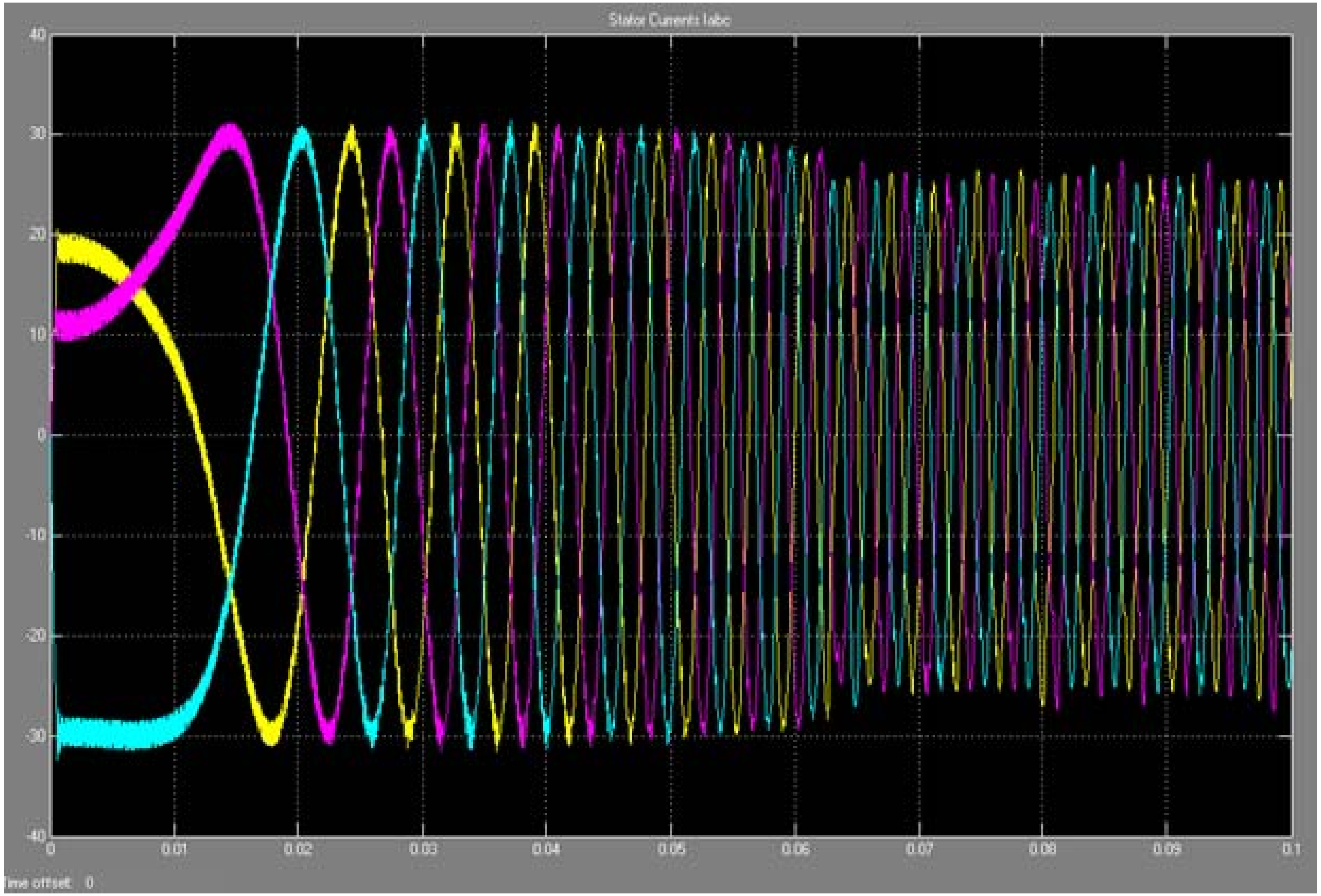


Figure 5.5 Iabc Currents versus Time (in second) at 1800 rad/s

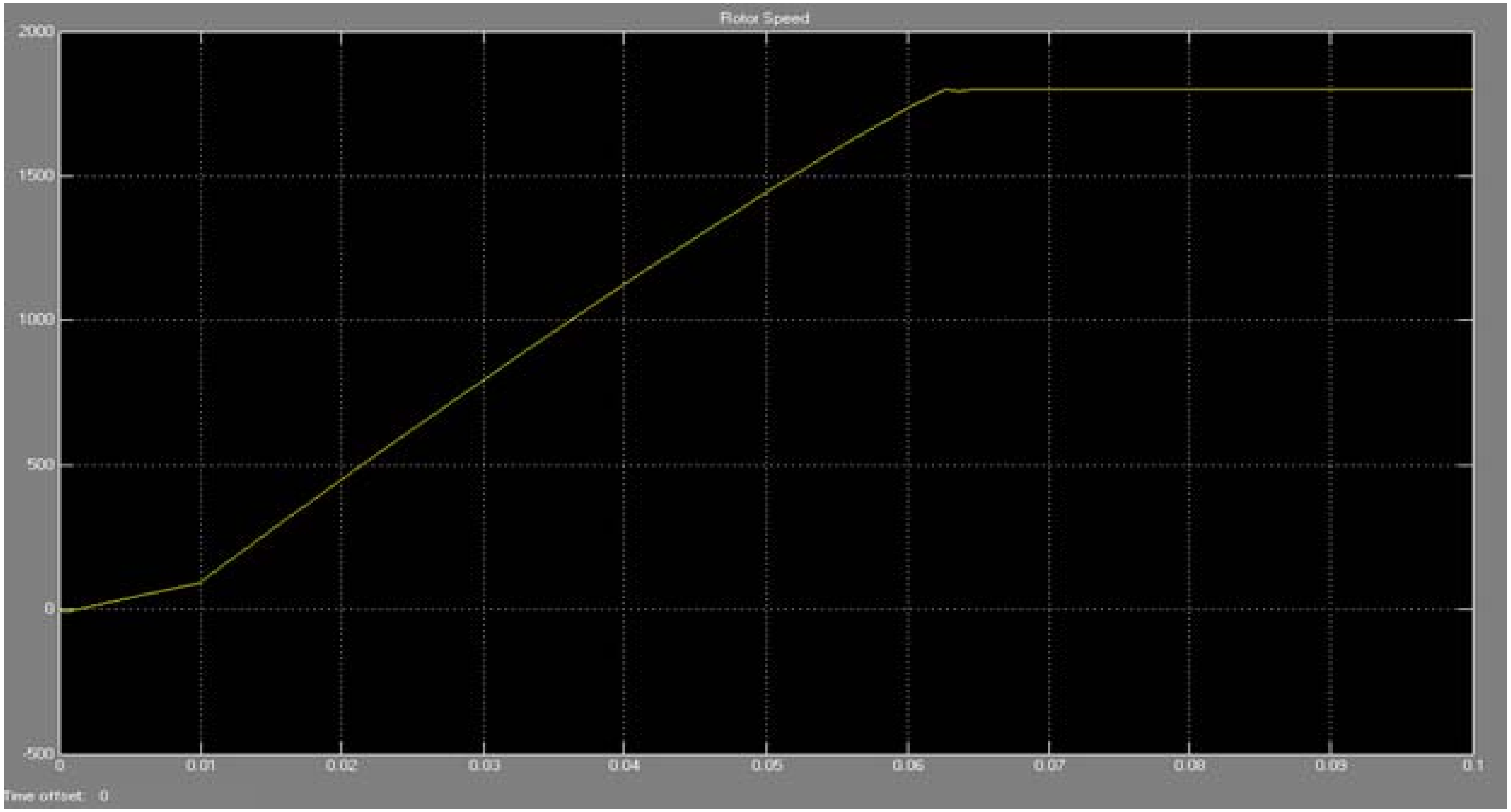


Figure 5.6 Motor Electrical Speed versus Time (in second) at 1800 rad/s

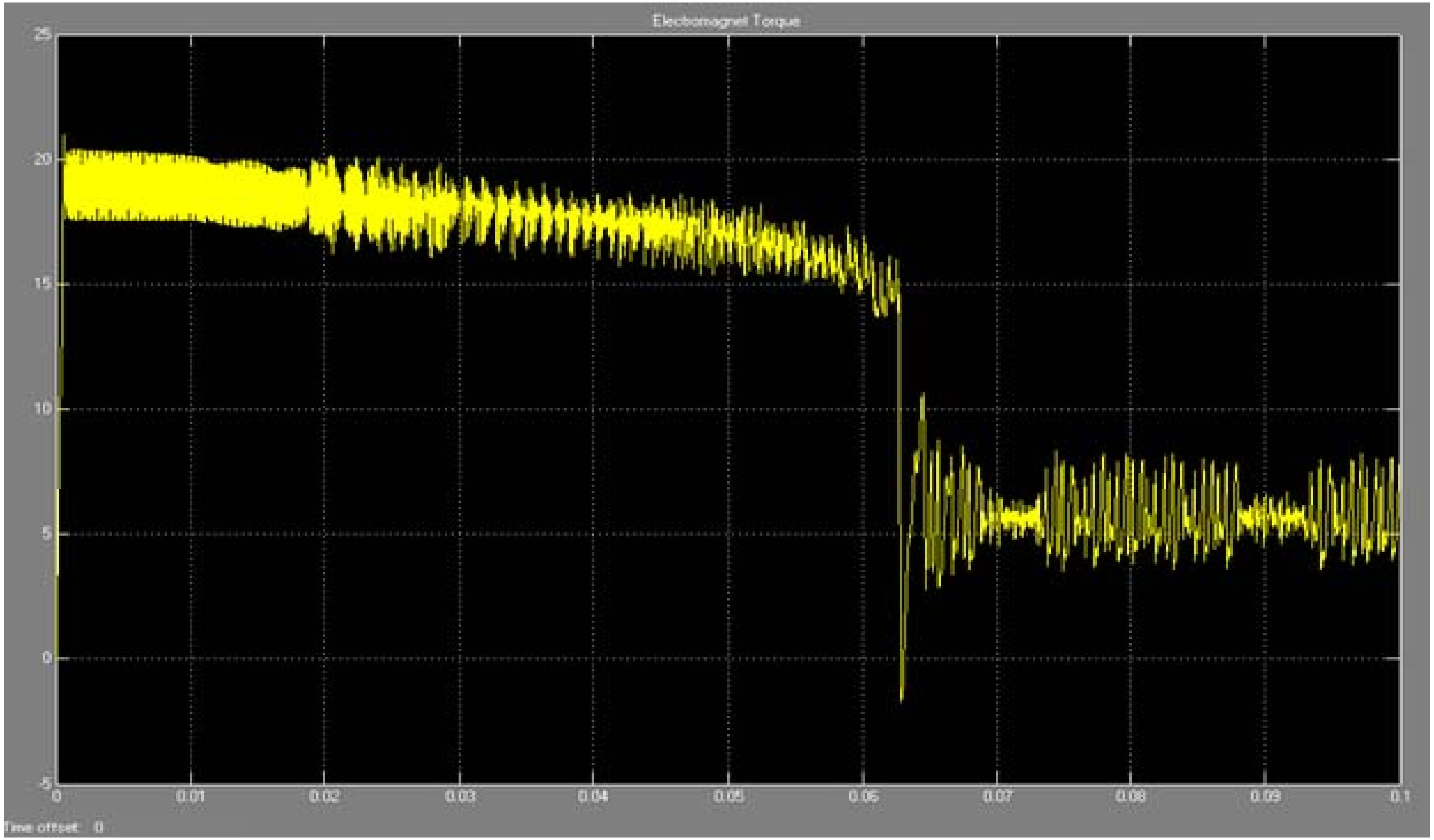


Figure 5.7 Developed Torque (in Nm) versus Time (in second) at 1800 rad/s

In Figure 5.4 Both d and q axis current are present. It is clearly shown that the d-axis current in the constant torque region is almost equal to zero (Figure 5.4B) and as speed increases into field weakening the d-axis current increases (Figure 5.4A).

Figures 5.5, 5.6 and 5.7 demonstrate the dynamic performance of the motor as it accelerates from zero speed to 1800 rad/sec.

Fig. 5.5 shows the phase currents which are relatively high during acceleration to generate the accelerating torque and reduce to the steady state value at constant speed. It is clear that the current is non sinusoidal at the starting and becomes sinusoidal when the motor reaches the controller command speed at steady state.

Figure 5.6 shows a variation of the speed with time. The steady state speed is the same as that of the commanded reference speed, which is 1800 rad/sec. Figure 5.7 shows the developed torque of the motor. The starting torque is quiet high and the steady state value of torque is reduced to 6 N-m at reference speed.

5.2 DSP processor Outputs

All the modules, which include the Synchronous Motor, are fully assembled and tested in the Lab. Because of lack of Permanent Magnet Synchronous Motor, Synchronous motor is used for this thesis by making the field constant.

(a)

(b)

Figure 5.8 Fully assembled system in Lab

Before the system is fully assembled, most of the modules were tested individually.

5.2.1 Gate Driver Circuit

In this thesis, PC123 Optocouplers are used. One leg of the circuit is tested with protous professional and with DSP pulse width modulation output. The circuit and the result of the output with DSP PWM input are shown in the following figure.

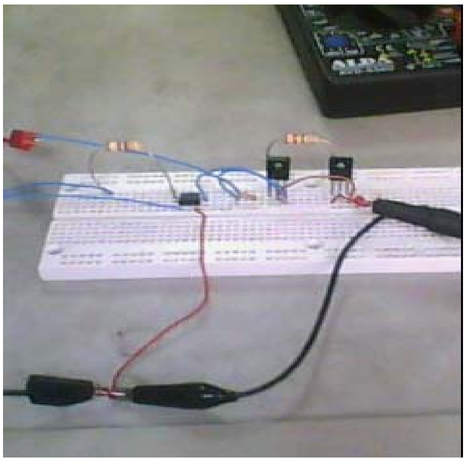


Figure5.9A One leg of the Gate Driver circuit

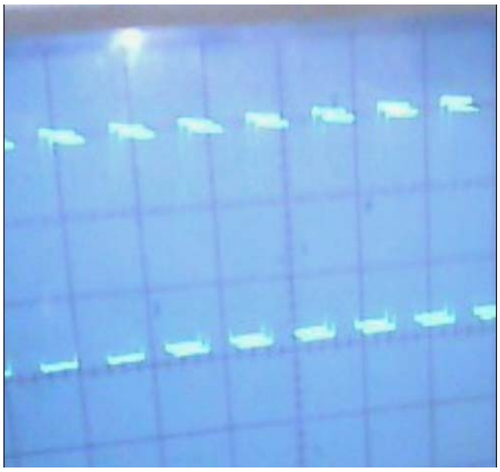


Figure5.9B 15V output of gate driver circuit with 3.3 DSP PWM input

The above figures shows that the DSP PWM output, which is 3.3 V is injected as an input for the Gate Driver circuit; and the gate driver amplifies the input to 15V which is the input for the inverter leg.

5.2.2 Analogue to Digital Converter

The shunt terminal voltages are sampled and converted by the TMS320F2401 Analog to Digital Converter to sense the current. The 10-Bit Analog-to-Digital Converter (ADC) of the TMS320LF2401A is tested by giving the result of the ADC from result register (RESULT0) to the compare register as a PWM duty cycle.

And PWM pattern is found as shown in figure 5.10. This result proof that:

1. The ADC of TMS320LF2401A works
2. The assembly code of ADC in current sensing module is applicable.

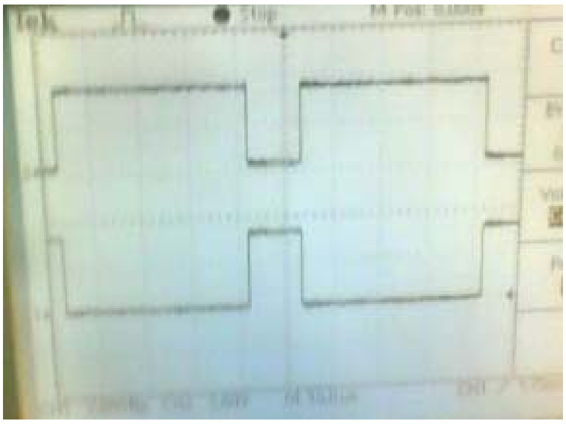


Figure5.10 PWM pattern from TMS320LF2401A

The duty cycle is 100µs.

5.2.3 Position Encoder with Laser Sensor

The sensor is tested by rotating the motor with the encoder manually. The sensor gives square wave signal that is “0” and 1”. The laser sensor output is 15V, so voltage divider is used to reduce the voltage to 3V as the DSP maximum input voltage is 3.3 Volt.

5.2.4 Overall System

Since the encoder does not give absolute information on the rotor position, only a relative position can be computed from a known position. Because of this, initial phase is specified to place the rotor in a known position at start. The system is tested by giving different electrical position at start.

When different initial phase is specified, it is found different computational time which results in different three phases from the inverter. These results are what expected from the DSP program.

In the next figure the different pattern read from the DSP is shown. As the oscilloscope available in the laboratory has only two channels, only two phases are shown.

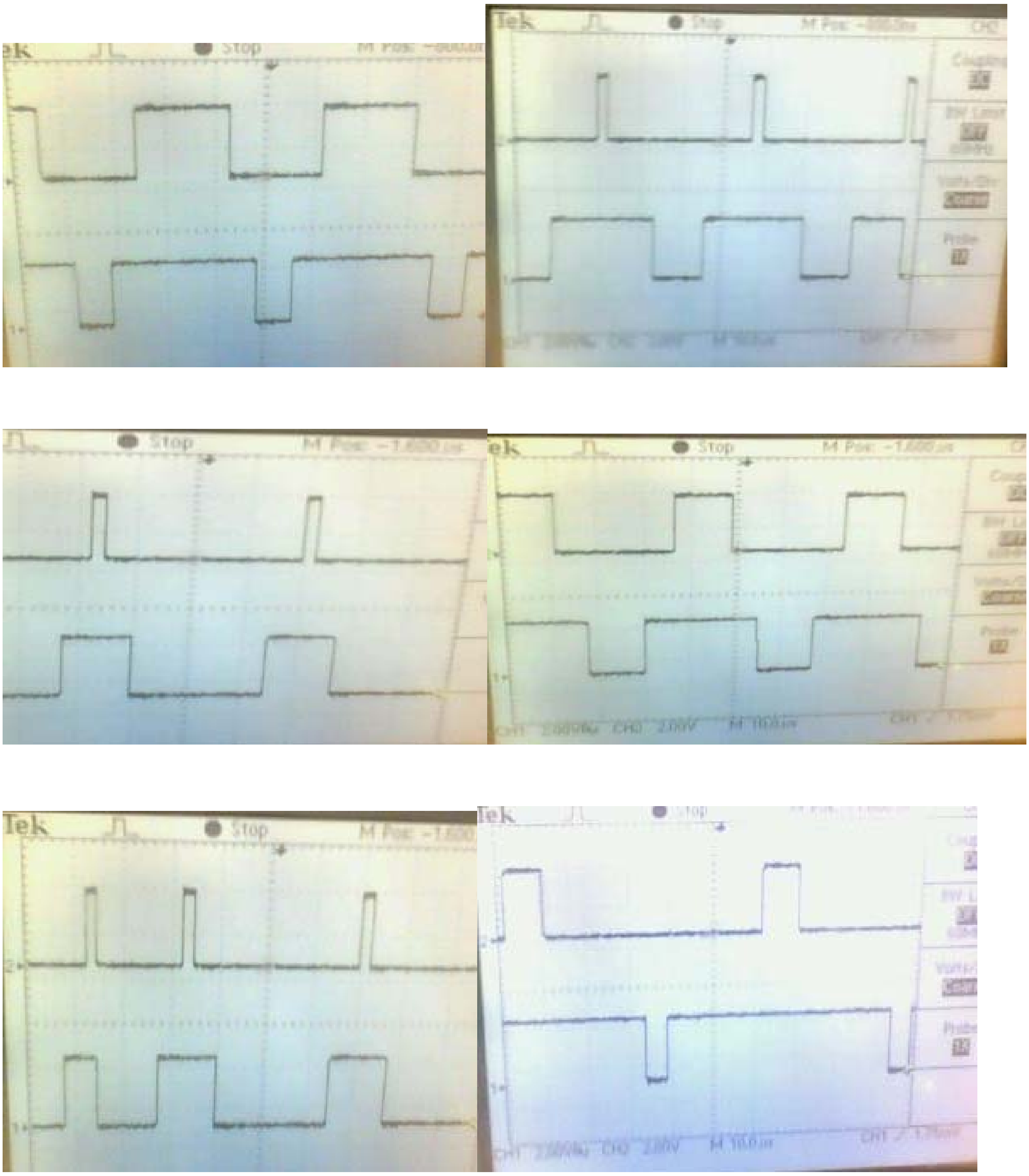


Figure 5.11 Different pattern of PWM from SVPWM (which shows different sectors)

Using a three-leg inverter, eight transistor configurations are possible. These configurations generate eight vectors two of which are 'zero' vectors. The remaining vectors divide the plane into six sectors. The Space Vector Modulation uses a special scheme to switch the power transistors to generate the eight vectors, which are the translation of the (α, β) voltage reference vector into an amount of time of commutation (on/off) for each power transistors. The above figure (figure 5.11) shows these different PWM patterns, which are in different sectors where the reference voltage located. And these different patterns appear by specifying different initial phase which results in different reference voltage.

Chapter 6

Conclusion and Recommendation

6.1 Conclusion

In this thesis, a detailed modeling, analysis and simulation of flux-weakening control methods of permanent magnet motor drives is presented. Simulation using MATLAB and Some of the modules of field oriented control for a three-phase permanent magnet drive based on the Texas Instruments TMS320LF2401A DSP Controller is performed. And results are given.

The work shows that the highly efficient permanent magnet synchronous motors can be controlled using field oriented control and the air-gap flux can be weakened using the direct-axis component current for wide speed range.

This work can be used as a reference for further work and the experimental systems developed can be used for further investigation of the system.

6.2 Recommendation

In this thesis, the experimental system development was a challenge due to the unavailability of components locally. However, most of the components (position sensors; Gate Drivers and Inverters) have been developed during this work. Therefore, we recommend the continuation of this high-tech and important research.

References

Periodicals:

[1] T. Sebastian, G. Slemon, and M. Rahman, "Modelling of permanent magnet synchronous motors," Magnetics, IEEE Transactions on, vol. 22, pp. 1069-1071, 1986.

[2] P. Pillay and R. Krishnan, "Modeling of permanent magnet motor drives," Industrial Electronics, IEEE Transactions on, vol. 35, pp. 537-541, 1988.

[3] Arash Hassanpour Isfahani, and Siavash Sadeghi, “Design of a Permanent Magnet Synchronous Machine for the Hybrid Electric Vehicle”,1996

[4] K. Jang-Mok and S. Seung-Ki, "Speed control of interior permanent magnet synchronous motor drive for the flux weakening operation," Industry Applications, IEEE Transactions on, vol. 33, pp. 43-48, 1997.

Books:

[5] R. Krishnan, “ELECTRIC MOTOR DRIVES: Modeling, Analysis and Control” Virginia Tech,Blacksburg,VA,2001

[6] Malcolm Barnes, “Practical Variable Speed Drives and Power Electronics” MSEE, Automated Control Systems, Australia,1998

[7] Paul C.Krause, Oleg wasynczuk and Scott D. sudhoff, “Analysis of Electric Machinery and Drive Systems” IEEE series on power engineering, Second Edition2002

Technical and application Reports:

[8] Texas Instruments, “Digital Signal Processing Solution for Permanent Magnet Synchronous Motor”, Application Note Literature Number: BPRA044, 1997

[9] Texas Instruments Inc., “Implementation of a Speed Field Oriented Control of 3-phase PMSM Motor using TMS320F240” , www.ti.com, Texas Instruments, Literature number SPRA588, Europe, 1999

[10] Texas Instruments Inc., “Field Orientated Control of 3-Phase AC-Motors”, www.ti.com, Texas Instruments Literature number BPRA073, Europe, 1998

[11] Texas Instruments Inc., “Clarke and Park Transforms on the TMS320C2xx”, www.ti.com, Texas Instruments Literature number BPRA048, Europe, 1998

[12] Texas Instruments Inc., “Implementation of Vector Control for PMSM Using the TMS320F240 DSP” Michel Platnic,1998

[13] TMS320F/C24x DSP Controllers Reference Guide, CPU and Instruction Set, Literature Number: SPRU160C, June 1999

[14] David M. Alter, Getting Started in C and Assembly Code With the TMS320LF240x DSP, Application Report, SPRA755A - July 2002

[15] TMS320C1x/C2x/C2xx/C5x Assembly Language Tools User’s Guide, SPRU018D, 1995

[16] TMS320F/C240 DSP Controllers Reference Guide, Peripheral Library and Specific Devices, Literature Number: SPRU161D, November 2002

[17] TMS320LF/LC240xA DSP Controllers, Reference Guide, System and Peripherals Literature Number: SPRU357C, Revised May 2006

[18] TMS320LF2401A, TMS320LC2401A DSP Controllers, SPRS161K − MARCH 2001 − REVISED JULY 2007

[19] David M. Alter, Getting Started in C and Assembly Code with the TMS320LF240x DSP, Application Report, SPRA755A - July 2002

[20] Implementation of a Speed Controlled Brushless DC Drive Using TMS320F240 Literature Number: BPRA064, Texas Instruments Europe, July 1997

[21] eZdspTM LF2401 Technical Reference 506055-0001 Rev. A April 2002 SPECTRUM DIGITAL, INC.,2002

[22] Space-Vector PWM with TMS320C24x/F24x Using Hardware and Software Determined Switching Patterns, Application Report SPRA524, March 1999

[23] Three phase current measurements using a single line resistor on the TMS320F240 Literature Number: BPRA077 Texas Instruments Europe, May 1998

[24] Milan Brejl & Michal Princ “Permanent Magnet Synchronous Motor Vector Control, Driven by eTPU on MPC5500” System Application Engineers Roznov Czech System Center ,1997

[25] S. Van Haute, G. Terörde, K. Hameyer, R. Belmans, “MODELLING AND IMPLEMENTATION OF A PERMANENT MAGNET SYNCHRONOUS MOTOR DRIVE USING A DSP DEVELOPMENT ENVIRONMENT” Katholieke Universiteit Leuven, Belgium,1998

[26] Zhang Qianfan, Song Liwei, Zheng Ping, Cheng Shukang, "Permanent Magnet Synchronous Motor Drive System Applying on Hybrid Electric Car",1994

[27] Shigeo Morimoto, Member, IEEE, Yoji Takeda, Member, IEEE, Keita Hatanaka, Yi Tong, And Takao Hirasa, Member, IEEE "Design and Control System of Inverter-Driven Permanent Magnet Synchronous Motors for High Torque Operation", March 1998

[28] John S. Hsu, "A Machine Approach for Field Weakening of Permanent-Magnet Motors" 'Oak Ridge National Laboratory,1994

[29] Y.K. Chin, J. Soulard, "A Permanent Magnet Synchronous Motor for Traction Applications of Electric Vehicles" Royal Institute of Technology (KTH) Department of Electrical Engineering Permanent Magnet Drives (PMD) Research Group Stockholm, Sweden,2002

[30] Texas Instruments Inc., "Implementation of Vector Control for PMSM Using the TMS320F240 DSP" www.ti.com, Texas Instruments, Literature number SPRA494, 1998

Appendix TMS320LF2401A FOC Software

```
;________________________________________________________________________
; Addis Ababa University, ECE

; Implementation of a SVPWM control of 3 phase PMSM with Field Weakening

; File Name: SVPWMF.asm
;
; Author:   Henok Berehanu
;________________________________________________________________________
;
;            Global symbol declaration
;-----------------------------------------------------------------
           .def     _c_int0, T1PR_ISR,PHANTOM,GISR4   ;entry points
           .include "240xA.h"                  ;header file
;________________________________________________________________________
;
;   ** Uninitialized Variables in B1 DARAM  **
;________________________________________________________________________

Kcurrent .word 0D9Ah; 8.8 format

SQRT3inv .word 093dh ;1/SQRT(3) 4.12 format

SQRT32   .word 0ddbh ;SQRT(3)/2 4.12 format

;*** PWM modulation constants

PWMPRD .set 07d0h ; PWM Period=2*2000 -> Tc=2*2000*25ns=100us (25ns; resolution)

Tonmax .set 0 ; minimum PWM duty cycle

MAXDUTY .set PWMPRD-2*Tonmax ; maximum utilization of the inverter

;*** PI current regulators parameters

Ki .word 07Ah ;4.12 format = 0.03
```

```
Kpi .word 999h ;4.12 format = 0.60 (include period)

Kcor .word 0cch ;4.12 format = 0.05
;Kcor = Ki/Kpi

;*** PI speed regulators parameters

Kispeed .word 7ah ;4.12 format = 0.03

Kpispeed .word 06800h ;4.12 format = 6.5

Kcorspeed .word 12h ;4.12 format = 0.0046


;*** Vqr and Vdr limitations

Vbase .set 01000h ;BEMF at base speed

Vmin .set 0ec00h ;4.12 format = -1.25 pu

Vmax .set 01400h ;4.12 format = 1.25 pu

;*** Is and isdref limitations

;*** Is and isdref limitations

ismax .set 01300h ;4.12 format

isdrefmin .set 0ed00h ;4.12 format

isdrefmax .set 00000h ;4.12 format = 0A (1000h = Ibase)

zero .word 0h

;*** Encoder variables and constants


;revolution

;*** Speed and estimated speed calculation constants

Nbase .set 1000h ;Base speed
n_ref .set 0400h
Kspeed   .set  097ch;8.8 format

SPEEDSTEP   .set 20
```

```
;*** Initialization phase Iqr

iSqrefinit .set 01000h ;4.12 format

;*** Encoder variables and constants

Encpulses .set 256 ;number of encoder pulses per mechanical
                   ;revolution


   .bss CAPCOUNT,1
   .bss TEMP,1
   .bss tmp,1
   .bss one,1
   .bss ia,1 ;phase current ia
   .bss ib,1 ;phase current ib
   .bss ic,1 ;phase current ic
  .bss Ua,1 ; (not used)
   .bss Ub,1 ; (not used)
   .bss Uc,1 ; (not used)
   .bss sin,1 ;generated sine wave value
   .bss t1,1 ;SVPWM T1 (see SV PWM references for details)
   .bss t2,1 ;SVPWM T2 (see SV PWM references for details)
  .bss cos,1 ;generated cosine wave value
   .bss Va,1 ;Phase 1 voltage for sector calculation
   .bss Vb,1 ;Phase 2 voltage for sector calculation
   .bss Vc,1 ;Phase 3 voltage for sector calculation
   .bss VDC,1 ;DC Bus Voltage
   .bss taon,1 ;PWM commutation instant phase 1
   .bss tbon,1 ;PWM commutation instant phase 2
   .bss tcon,1 ;PWM commutation instant phase 3
    .bss teta,1 ;rotor electrical position in the range [0;1000h]
              ;4.12 format = [0;360] degrees
     .bss iSalfa,1 ;alfa-axis current
    .bss iSbeta,1 ;beta-axis current
    .bss vSal_ref,1 ;alfa-axis reference voltage
    .bss vSbe_ref,1 ;beta-axis reference voltage
    .bss iSdref,1 ;d-axis reference current
   .bss iSqref,1 ;q-axis reference current
   .bss iSd,1 ;d-axis current
   .bss iSq,1 ;q-axis current
   .bss vSdref,1 ;d-axis reference voltage
   .bss vSqref,1 ;q-axis reference voltage
  .bss epiq,1 ;q-axis current regulator error
   .bss epid,1 ;d-axis current regulator error
```

```
.bss xiq,1 ;q-axis current regulator integral component
.bss xid,1 ;d-axis current regulator integral component
.bss n,1 ;speed
;.bss n_ref,1 ;speed reference
 .bss epispeed,1 ;speed error (used in speed regulator)
.bss xispeed,1 ;speed regulator integral component
.bss X,1 ;SVPWM variable
.bss Y,1 ;SVPWM variable
.bss Z,1 ;SVPWM variable
.bss sectordisp,1 ;SVPWM sector for display
.bss initphase,1 ;flag for initialization phase
.bss Vr,1 ;(not used)
.bss iSqrefmin,1 ;iSq min limitation
.bss iSqrefmax,1 ;iSq max limitation


.bss sector,1 ;SVPWM sector
.bss VDCinvT,1 ;used in SVPWM
.bss index,1 ;pointer used to access sine look-up table
.bss upi,1 ;PI regulators (current and speed) output
.bss elpi,1 ;PI regulators (current and speed) limitation error
.bss tmp1,1 ;tmp word
.bss accb,2 ;2 words buffer
.bss acc_tmp,2 ;2 words to allow swapping of ACC
.bss encoderold,1 ;encoder pulses value stored in the previous
            ;sampling period
.bss encincr,1 ;encoder pulses increment between two
            ;consecutive sampling periods
.bss speedtmp,1 ;used to accumulate encoder pulses increments
            ;(to calculate the speed each speed sampling period)
.bss speedstep,1 ;sampling periods down counter used to
            ;define speed sampling period
.bss epvr,1 ;PI error for field weakening
.bss xvr,1 ;PI integral term for field weakening
```

```
;________________________________________________________________
;
;          ** Start of main body of code **
;________________________________________________________________
       .text       ;
     _c_int0:


     lar ar5,#60h


**************************************************
* Macro
* Notes: this function uses the Newton-Raphson method:
* X(n) = 0.5 * ( X(n-1) + N/X(n-1) )
**************************************************
isqrt .macro
 clrc sxm
 lacc *+ ;lacc Vr, X(0)
 sfr ;Initial approximate root = N/2
 sacl *+ ;
 splk #10,* ;
isqrt?: ;X(n) = 0.5 * ( X(n-1) + N/X(n-1) ):
 sbrk #2
 lacc *+ ;
 rpt #15
 subc *
 and #0FFFFh
 add *
 sfr
 sacl *+
 lacc * ;Repeat until iterations completed:
 sub #1
 sacl *
 bcnd isqrt?,NEQ
 mar *- ;
 lacc *
 mar *-
 setc sxm
 .endm
```

```
;-----------------------------------------------------------------
;         ** System configuration **
;-----------------------------------------------------------------
; Configure system registers
;
     LDP    #0E0h
     SPLK   #0000001000000101b, SCSR1  ;system control & status Register1


     SPLK   #0000000000001011b, SCSR2  ;system control & status Register2

                ;point at I/O register page

    SPLK   #01h,  SCSR4

     SPLK   #0E8h, WDCR

     SPLK   #00h,  SCSR4


;----------------------------------------------------------------------
;    No wait states for prog. & data spaces
;---------------------------------------------------------------------
     LDP  #TEMP
     SPLK #0000000001000000b, TEMP
     OUT  TEMP, WSGR
;***************************************************************
;* Variables initialization
;****************************************************************
 ldp #ia
 lacc ismax
 sacl iSqrefmax
 neg
 sacl iSqrefmin
 zac
 sacl iSqref
 sacl iSdref
 sacl n_ref
 sacl iSdref
 sacl index
 sacl xid
 sacl xiq
 sacl xispeed
 sacl upi
```

```
 sacl elpi
 sacl Va
 sacl Vb
 sacl Vc
 sacl initphase
 sacl CAPCOUNT
  splk #0535h,VDCinvT
;----------------------------------------------------------------
              ;ADC INITIALIZATION
;--------------------------------------------------------------
      LDP #0E1h
      SPLK #0100000000000000b, ADCTRL1; Reset ADC module
      NOP
      SPLK #0011000000000000b, ADCTRL1; setup and enable ADC module

   SPLK #1, MAXCONV              ; Maximum of 2 conversions
   SPLK #02121h, CHSELSEQ1
   ;SPLK # 0010000000000000h,ADCTRL2 ;start the conversion

;-----------------------------------------------------------------------
     ;Setup the software stack
;-------------------------------------------------------------------------

stk_len .set 20                ;stack length
stk     .usect "stack",stk_len   ;reserve space for stack
        LAR AR1, #stk          ;AR1 is the stack pointer
;-----------------------------------------------------------------------------------------
      LDP  #0E1h


     SPLK #0000000001111110b,MCRA


;=====================================================================
;        Configure GP Timer 1
;-------------------------------------------------------------------------
     LDP #0E8h

 SPLK #0000h, T1CON

 SPLK #0000000000000000b, GPTCONA

 SPLK #0000h, T1CNT

  SPLK #PWMPRD, T1PR
```

```
 SPLK #0000010111101100b, DBTCONA

 SPLK #0666h, ACTRA

 splk #500,CMPR1 ;no current sent to the motor

 splk #700,CMPR2

 splk #1000,CMPR3
;==========================================================================
=======
;--------------------------------------------------------------------------
    SPLK #0200h, COMCONA

    SPLK #8200h, COMCONA
    SPLK #8800h, T1CON
    lacc T1CON

     or #40h

     sacl T1CON
     SPLK #20b0h,CAPCONA
     SPLK #0100h,CAP1FIFO
;==========================================================================
          ;Setup the core interrupts
;--------------------------------------------------------------------------
     LDP    #0h              ;set data page
     SPLK   #0h,  IMR        ;clear the IMR register
     SPLK   #111111b,IFR      ;clear any pending core interrupts
     SPLK   #001010b,IMR      ;enable core interrupt,INT2 & INT4

     LDP    #0E8h            ;set data page on EVA
     SPLK   #0FFFFh, EVAIFRA  ;clear all EVA group A interrupts
     SPLK   #0FFFFh, EVAIFRB  ;clear all EVA group B interrupts
     SPLK   #0FFFFh, EVAIFRC  ;clear all EVA group C interrupts
     SPLK   #00200h, EVAIMRA  ;Mask all but GPT1 UF int. (INT2)
     SPLK   #00h,    EVAIMRB  ;Mask all int.
     SPLK   #0001h, EVAIMRC  ;Mask all exept capture ints

;__________________________________________________________________________
;
;          ** Initialize variables **
;__________________________________________________________________________


     CLRC    INTM                    ; Enable global interrupts
```

```
;_____________________________________________________________________
;
;              ** START OF MAIN LOOP **
;_____________________________________________________________________
loop:
      NOP
        B loop; branch to loop
;_______________________________________________________________
;
;          ** Handle interrupts **
;_______________________________________________________________
T1PR_ISR:
;Context save to the software stack
     MAR  *,AR1         ;AR1=stack pointer
     MAR  *+            ;skip one stack location (required)
     SST  #1, *+        ;save ST1
     SST  #0, *+        ;save ST0
     SACH *+            ;save ACCH
     SACL *+            ;save ACCL
     LDP  #0E8h         ;point to EVA reg page
     SPLK #078fh,EVAIFRA   ;Clear group B flags

     LDP        #0E1h    ; Point at ADC reg page

      SPLK #1,  MAXCONV              ; Maximum of 2 conversions
   SPLK #1010h, CHSELSEQ1

   SPLK #0010000000000000b,ADCTRL2 ; Start the conversions

     ; Start the conversions
             NOP                      ; Wait for Busy bit to set.
             NOP
             NOP
             NOP

CHK_EOS1:

             BIT           ADCTRL2, 3              ; Wait for SEQ1 Busy bit to clear (bit 12)
            BCND       CHK_EOS1, TC             ; If TC=1, keep looping.
            RPT #8
           NOP                                  ; the conversion results are now available in the
                                                       ; RESULTSn regs

             LACC     RESULT0, 10               ; The result of this conversion lays in the 10 upper
                                              ; bits of the ADC FIFOs. Therefore a left shift of
            LDP        #ia                      ; 10 bit is performed to obtain the result of the
```

```
 SACH      ia                          ; conversion in the upper word of the accumulator.
LACC     RESULT1, 10
LDP      #ib
SACH      ib


;*** Initialization phase

lacl initphase ;are we in initialization phase ?
bcnd Run,NEQ
lacc #0fc00h ;if yes, set teta = 0fc00h 4.12 format = -90
          ; degrees

sacl teta ;
lacc #iSqrefinit ;q-axis reference current = initialization
              ;q-axis reference current
sacl iSqref ;
lacc #0 ;zero some variables and flags
 sacl iSdref ;
 sacl teta ;
sacl encoderold ;
 sacl n ;
sacl speedtmp ;
 lacc #SPEEDSTEP ;restore speedstep to the value SPEEDSTEP
              ; for next speed
              ;control loop
 sacl speedstep ;
 ldp #CAPCOUNT
  splk #0,CAPCOUNT ;zero Incremental Encoder value if
              ;initialization step
  ldp #initphase
  splk #1h, initphase
  b Init ;there is no need to do position and
        ;calculation
         ;in initialization phase (the rotor is locked)

Run

;*** Encoder pulses reading
  LDP #CAPCOUNT

  lacc CAPCOUNT ;we read the encoder pulses and ...
  ;neg ;encoder plug in the opposite direction ?
   ldp #ia
  sacl tmp
  sub encoderold ;subtract the previous sampling period value
```

```
                ;to have the increment that we'll
                ;accumulate in encoder
  sacl encincr ;
  add teta ;
  bcnd encmagzero,GT,EQ;here we start to normalize teta
                    ; value to the range [0;Encpulses-1]
  add #Encpulses ;the value of teta could be negative
                 ;it depends on the rotating direction
                  ;(depends on motor windings
                   ;to PWM Channels connections)
encmagzero
  sacl teta ;now teta value is positive but could be
            ;greater than Encpulses-1
  sub #Encpulses ;we subtract Encpulses and we check whether
            ;the difference is negative. If it is we
                ;already have the right value in teta
  bcnd encminmax,LT ;
 sacl teta ;otherwise the value of teta_m is greater
           ;than Encpulses and so we have to store the
encminmax ;right value ok, now teta contains the
          ; right value in the range
   lacc tmp ;[0,Encpulses-1]
                ;the actual value will be the old one during
          ; the next sampling period
     sacl encoderold


;*******************
;* Teta calculation
;*******************
       lacc teta
        and #0fffh
         sacl teta
;*** END theta calculation

;*****************************************************
;* Calculate speed and update reference speed variables
;*****************************************************
  lacc speedstep ;are we in speed control loop ? (SPEEDSTEP
               ;times current control loop)
   sub #1 ;
    sacl speedstep ;
    bcnd nocalc,GT ;if we aren't, skip speed calculation
;*** Speed calculation from encoder pulses

   lt speedtmp ;multiply encoder pulses by Kspeed (8.8
```

```
                        ; format constant)
                        ;to have the value of speed
    mpy #Kspeed ;
    pac ;
    rpt #11 ;
    sfr  ;
     sacl n
     lacc #0 ;zero speedtmp for next calculation
     sacl speedtmp ;
     lacc #SPEEDSTEP ;restore speedstep to the value SPEEDSTEP
      sacl  speedstep ;for next speed control loop


;***************************************************
;* Speed regulator with integral component correction
;***************************************************
    lacc n_ref
    sub n
    sacl epispeed
    lacc xispeed,12
     lt epispeed
    mpy Kpispeed
    apac
    sach upi,4
                ;here start to saturate
     bit upi,0
    bcnd upimagzeros,NTC ;If value +ve branch
    lacc iSqrefmin
    sub upi
    bcnd neg_sat,GT ;if upi<iSqrefmin then branch to saturate
    lacc upi ;value of upi is valid
     b limiters
neg_sat
     lacc iSqrefmin ;set acc to -ve saturated value
      b limiters
upimagzeros ;Value is positive
     lacc iSqrefmax
     sub upi ;
     bcnd pos_sat,LT ;if upi>iSqrefmax then branch to saturate
     lacc upi ;value of upi valid
      b limiters
pos_sat
      lacc iSqrefmax ;set acc to +ve saturated value

limiters
     sacl iSqref ;Store the acc as reference value
```

```
    sub upi
    sacl elpi
    lt elpi
    mpy Kcorspeed
    pac
    lt epispeed
     mpy Kispeed

    apac
  add xispeed,12
   sach xispeed,4

;***end speed regulation


;**************************************************
;* Field-weakening algorithm with PI regulator
;* Calculation of sqrt(Vdr^2 + Vqr^2)
;* only if n>Nbase/2
;**************************************************
  lacc #Nbase
   sfr
  sub n
  bcnd nocalc,GEQ ;calculate field-weakening if n>Nbase/2
  lar ar5,#60h
 zac
 mpy #0
 mar *,ar5
 spm 2 ;4.12 multiplication format
 sqra vSdref
 sqra vSqref
 apac
 sach *
 isqrt ;calculate the square root
 spm 0
 sacl Vr,6
;***********************************************************
;* Voltage regulator with integral component correction
;* (Vbase,Vr)->(isdref)
;***********************************************************
 lacc #Vbase
 sub Vr
 sacl epvr
 lacc xvr,12
 lt epvr
 mpy Kpi
```

```
 apac
 sach upi,4
 bit upi,0
 bcnd upimagzerov,NTC
 lacc #isdrefmin
 sub upi ;
 bcnd neg_satv,GT ;if upi<Vmin branch to saturate
 lacc upi ;value of upi is valid
 b limiterv
neg_satv
 lacc #isdrefmin ;set ACC to neg saturation
 b limiterv
upimagzerov ;Value was positive
 lacc #isdrefmax
 sub upi ;
 bcnd pos_satv,LT ;if upi>Vmax branch to saturate
 lacc upi ;value of upi is valid
 b limiterv
pos_satv
 lacc #isdrefmax ;set ACC to pos saturation
limiterv
 sacl iSdref ;Always negative
 sub upi
 sacl elpi
 lt elpi
 mpy Kcor
 pac

 lt epvr
 mpy Ki
 apac
 add xvr,12
 sach xvr,4
;*** END voltage regulator with integral component correction

;**************************************************
;* Field-weakening algorithm iqr limitation
;* for PI regulator
;* Calculation of sqrt(ismax^2 - isdref^2)
;* Output iSqrefmax
;**************************************************
 lar ar5,#60h
 zac
 mpy #0
 mar *,ar5
 spm 2 ;4.12 multiplication format
```

```
 sqra iSdref
 sqrs ismax ;substract
 apac
 sach *
 isqrt ;calculate the square root
 spm 0
 sacl iSqrefmax,6
 neg
 sacl iSqrefmin,6
;*** END field weakening routines
****************************************************
* Encoder update
****************************************************
nocalc ;branch here if we don't have to calculate the speed
 lacc speedtmp ;use the actual encoder increment to update the
              ;increments accumulator used to calculate the speed
    add encincr ;
   sacl speedtmp ;
;*** END Measured speed and reference speed variables updating

Init
;*********************************************
;* Sampled current scaling
;* to nominal current 1000h <-> I_nominal
;*********************************************
 ldp #ia
 lacc ia
 and #3ffh
 sub #440 ;then we have to subtract the offset (2.5V) to
         ; have positive and negative values of the
          ; sampled current
 sacl tmp
 spm 3
 lt tmp
 mpy Kcurrent
 pac
 sfr
 sfr
 sacl ia ;sampled current ia, f 4.12
 lacc ib
 and #3ffh
 sub #440
 sacl tmp
 lt tmp
 mpy Kcurrent
 pac
```

```
sfr
sfr
sacl ib
add ia
neg
sacl ic ;ic = -(ib+ia)
spm 0

;*******************************************
;* (a,b,c) -> (alfa,beta) axis transformation
;* iSalfa = ia
;* iSbeta = (2 * ib + ia) / sqrt(3)
;*******************************************
lacc ia
sacl iSalfa
lacc ib,1 ;iSbeta = (2 * ib + ia) / sqrt(3)
add ia
sacl tmp
lt tmp
mpy SQRT3inv ;SQRT3inv = (1 / sqrt(3)) = 093dh
         ;4.12 format = 0.577350269
pac
sach iSbeta,4

;*******************************************
;* Sine and cosine wave calculation from
;* teta values using sine look-up table
;*******************************************
lacc teta ;teta range is [0;1000h] 4.12 format = [0;360]
        ;so we have a pointer (in the range [0;0ffh])
        ;to the sine look-up table in the second and
         ;third nibble
rpt #3
sfr
and #0ffh ;now ACC contains the pointer to access the table
sacl index
add #sintab
sacl tmp
lar ar5,tmp
nop ; prevent pipeline conflict
nop
mar *,ar5
lacl *
nop
sacl sin ;now we have sine value
lacl index ;the same thing for cosine ... cos(teta)
```

```
          ;sin(teta+90°)
 add #040h ;90 degrees = 40h elements of the table
 and #0ffh
 sacl index ;we use the same pointer (we don't care)
 add #sintab
 sacl tmp
 lar ar5,tmp
 lacc *
 sacl cos ;now we have cosine value

;*******************************************
;* d-axis and q-axis current calculation
;* (alfa,beta) -> (d,q) axis transformation
;* iSd = iSalfa * cos(teta) + iSbeta * sin(teta)
;* iSq =-iSalfa * sin(teta) + iSbeta * cos(teta)
;*******************************************
 lacc #0
 lt iSbeta ;TREG0=iSbeta
 mpy sin ;PREG=iSbeta*sin(teta)
 lta iSalfa ;ACC+=PREG ; TREG0=iSalfa
 mpy cos ;PREG=iSalfa*cos(teta)
 mpya sin ;ACC+=PREG ; PREG=iSalfa*sin(teta)
 sach iSd,4
 lacc #0 ;ACC=0
 lt iSbeta ;TREG0=ibeta
 mpys cos ;ACC-=(PREG=iSalfa*sin(teta))

 apac ;ACC+=PREG
 sach iSq,4

;**********************************************************
;* q-axis current regulator with integral component correction
;* (iSq,iSqref)->(vSqref)
;**********************************************************
iq_reg:
 lacc iSqref
 sub iSq
 sacl epiq
 lacc xiq,12
 lt epiq
 mpy Kpi
 apac
 sach upi,4
 bit upi,0
 bcnd upimagzeroq,NTC
 lacc #Vmin
```

```
 sub upi
 bcnd neg_satq,GT ;if upi<Vmin branch to saturate
 lacc upi ;value of upi is valid
 b limiterq
neg_satq
 lacc #Vmin ;set ACC to neg saturation
 b limiterq
upimagzeroq ;Value was positive
 lacc #Vmax
 sub upi ;
 bcnd pos_satq,LT ;if upi>Vmax branch to saturate
 lacc upi ;value of upi is valid
 b limiterq
pos_satq
 lacc #Vmax ;set ACC to pos saturation
limiterq
 sacl vSqref ;Save ACC as reference value
 sub upi
 sacl elpi
 lt elpi
 mpy Kcor
 pac
 lt epiq
 mpy Ki
 apac
 add xiq,12
 sach xiq,4

;************************************************************
;* d-axis current regulator with integral component correction
;* (iSd,iSdref)->(vSdref)
;************************************************************
 lacc iSdref
 sub iSd
 sacl epid
 lacc xid,12
 lt epid
 mpy Kpi
 apac
 sach upi,4
 bit upi,0
 bcnd upimagzerod,NTC
 lacc #Vmin
 sub upi
 bcnd neg_satd,GT ;if upi<Vmin branch to saturate
 lacc upi ;value of upi is valid
```

```
 b limiterd
neg_satd
 lacc #Vmin ;set ACC to neg saturation
 b limiterd
upimagzerod ;Value was positive
 lacc #Vmax
 sub upi ;
 bcnd pos_satd,LT ;if upi>Vmax branch to saturate
 lacc upi ;value of upi is valid
 b limiterd
pos_satd
 lacc #Vmax ;set ACC to pos saturation
limiterd
 sacl vSdref ;Save ACC as reference value
 sub upi
 sacl elpi
 lt elpi
 mpy Kcor
 pac
 lt epid
 mpy Ki
 apac
 add xid,12
 sach xid,4

;***********************************************
;* alfa-axis and beta-axis voltages calculation
;* (d,q) -> (alfa,beta) axis transformation
;* vSbe_ref = vSqref * cos(teta) + vSdref * sin(teta)
;* vSal_ref =-vSqref * sin(teta) + vSdref * cos(teta)
;***********************************************
 lacc #0
 lt vSdref ;TREG0=vSdref
 mpy sin ;PREG=vSdref*sin(teta)
 lta vSqref ;ACC+=PREG ; TREG0=vSqref
 mpy cos ;PREG=vSqref*cos(teta)
 mpya sin ;ACC+=PREG ; PREG=vSqref*sin(teta)
 sach vSbe_ref,4
 lacc #0 ;ACC=0
 lt vSdref ;TREG0=vSdref
 mpys cos ;ACC-=(PREG=vSqref*sin(teta))
 apac ;ACC+=PREG
 sach vSal_ref,4
```

```
;**********************************************
;* Phase 1(=a) 2(=b) 3(=c) Voltage calculation
;* (alfa,beta) -> (a,b,c) axis transformation
;* modified exchanging alfa axis with beta axis
;* for a correct sector calculation in SVPWM
;* Va = vSbe_ref
;* Vb = (-vSbe_ref + sqrt(3) * vSal_ref) / 2
;* Vc = (-vSbe_ref - sqrt(3) * vSal_ref) / 2
;**********************************************
 lt vSal_ref ;TREG0=vSal_ref
 mpy SQRT32 ;PREG=vSal_ref*(SQRT(3)/2)
 pac ;ACC=PREG
 sub vSbe_ref,11 ;ACC-=vSbe_ref*2^11
 sach Vb,4
 pac ;ACC=PREG
 neg ;ACC=-ACC
 sub vSbe_ref,11 ;ACC-=vSbe_ref*2^11
 sach Vc,4
 lacl vSbe_ref ;ACC=vSbe_ref
 sacl Va ;Va=ACCL

;************************************
;* SPACE VECTOR Pulse Width Modulation
;* (see SVPWM references)
;************************************
 lt VDCinvT
 mpy SQRT32
 pac
 sach tmp,4
 lt tmp
 mpy vSbe_ref
 pac
 sach X,4
 lacc X ;ACC = vSbe_ref*K1
 sach accb
 sacl accb+1 ;ACCB = vSbe_ref*K1
 sacl X,1 ;X=2*vSbe_ref*K1


 lt VDCinvT
 splk #1800h,tmp
 mpy tmp ;implement mpy #01800h
 pac
 sach tmp,4
```

```
 lt tmp
 mpy vSal_ref
 pac
 sach tmp,4
 lacc tmp ;reload ACC with vSal_ref*K2
 add accb+1
 add accb,16
 sacl Y ;Y = K1 * vSbe_ref + K2 * vSal_ref
 sub tmp,1
 sacl Z ;Z = K1 * vSbe_ref - K2 * vSal_ref

;*** 60 degrees sector determination
 lacl #0
 sacl sector
 lacc Va
 bcnd Va_neg,LEQ ;If Va<0 do not set bit 1 of sector
 lacc sector
 or #1
 sacl sector ;implement opl #1,sector
Va_neg lacc Vb
 bcnd Vb_neg,LEQ ;If Vb<0 do not set bit 2 of sector
 lacc sector
 or #2
 sacl sector ;implement opl #2,sector
Vb_neg lacc Vc
 bcnd Vc_neg,LEQ ;If Vc<0 do not set bit 3 of sector
 lacc sector
 or #4
 sacl sector ;implement opl #4,sector
Vc_neg

;*** END 60 degrees sector determination
;*** T1 and T2 (= t1 and t2) calculation depending on the sector number

 lacl sector ;(see SPACE VECTOR Modulation references for;details)
 sub #1
 bcnd no1,NEQ
 lacc Z
 sacl t1
 lacc Y
 sacl t2
 b t1t2out
no1
 lacl sector
 sub #2
 bcnd no2,NEQ
```

```
 lacc Y
 sacl t1
 lacc X
 neg
 sacl t2
 b t1t2out
no2
 lacl sector
 sub #3
 bcnd no3,NEQ
 lacc Z
 neg
 sacl t1
 lacc X
 sacl t2
 b t1t2out
no3
 lacl sector
 sub #4
 bcnd no4,NEQ
 lacc X
 neg
 sacl t1
 lacc Z
 sacl t2
 b t1t2out
no4
 lacl sector
 sub #5
 bcnd no5,NEQ
 lacc X
 sacl t1
 lacc Y
 neg
 sacl t2
 b t1t2out
no5
 lacc Y
 neg
 sacl t1
 lacc Z
 neg
 sacl t2
t1t2out
 lacc t1 ;t1 and t2 minumum values must be Tonmax
 sub #Tonmax
```

```
 bcnd t1_ok,GEQ ;if t1>Tonmax then t1_ok
 lacl #Tonmax
 sacl t1

t1_ok
 lacc t2
 sub #Tonmax
 bcnd t2_ok,GEQ ;if t2>Tonmax then t2_ok
 lacl #Tonmax
 sacl t2
t2_ok
;*** END t1 and t2 calculation
 lacc t1 ;if t1+t2>2*Tonmax we have to saturate t1 and t2
 add t2 ;
 sacl tmp ;
 sub #MAXDUTY ;
 bcnd nosaturation,LT,EQ
;*** t1 and t2 saturation
 lacc #MAXDUTY,15 ;divide MAXDUTY by (t1+t2)
 rpt #15 ;
 subc tmp ;
 sacl tmp ;
 lt tmp ;calculate saturate values of t1 and t2
 mpy t1 ;t1 (saturated)=t1*(MAXDUTY/(t1+t2))
 pac ;
 sach t1,1 ;
 mpy t2 ;t2 (saturated)=t2*(MAXDUTY/(t1+t2))
 pac ;
 sach t2,1 ;
;*** END t1 and t2 saturation
nosaturation
;*** taon,tbon and tcon calculation
 lacc #PWMPRD ;calculate the commutation instants taon,
;tbon and tcon
 sub t1 ;of the 3 PWM channels
 sub t2 ;taon=(PWMPRD-t1-t2)/2
 sfr ;
 sacl taon ;
 add t1 ;tbon=taon+t1
 sacl tbon ;
 add t2 ;tcon=tbon+t2
 sacl tcon ;
;*** END taon,tbon and tcon calculation
;*** sector switching
 lacl sector ;depending on the sector number we have
 sub #1 ;to switch the calculated taon, tbon and tcon
```

```
 bcnd nosect1,NEQ ;to the correct PWM channel
;(see SPACE VECTOR Modulation references for details)
 bldd tbon,#CMPR1 ;sector 1
 bldd taon,#CMPR2
 bldd tcon,#CMPR3
 b RESTORE
nosect1
 lacl sector
 sub #2
 bcnd nosect2,NEQ
 bldd taon,#CMPR1 ;sector 2
 bldd tcon,#CMPR2 ;
 bldd tbon,#CMPR3 ;
 b RESTORE
nosect2
 lacl sector
 sub #3
 bcnd nosect3,NEQ
 bldd taon,#CMPR1 ;sector 3
 bldd tbon,#CMPR2 ;
 bldd tcon,#CMPR3 ;
 b RESTORE
nosect3
 lacl sector
 sub #4
 bcnd nosect4,NEQ
 bldd tcon,#CMPR1 ;sector 4
 bldd tbon,#CMPR2 ;
 bldd taon,#CMPR3 ;
 b RESTORE
nosect4
 lacl sector
 sub #5
 bcnd nosect5,NEQ
 bldd tcon,#CMPR1 ;sector 5
 bldd taon,#CMPR2 ;
 bldd tbon,#CMPR3 ;
 b RESTORE
nosect5
 bldd tbon,#CMPR1 ;sector 6
 bldd tcon,#CMPR2 ;
 bldd taon,#CMPR3 ;
 b RESTORE
```

```
;______________________________________________________________________________
;
; Theta and sine lookup table for obtaining both sine and cosine values from the electrical angle
;______________________________________________________________________________

sintab:

 .word 0
.word 101
.word 201
.word 301
.word 401
.word 501
.word 601
.word 700
.word 799
.word 897
.word 995
.word 1092
.word 1189
.word 1285
.word 1380
.word 1474
.word 1567
.word 1660
.word 1751
.word 1842
.word 1931
.word 2019
.word 2106
.word 2191
.word 2276
.word 2359
.word 2440
.word 2520
.word 2598
.word 2675
.word 2751
.word 2824
.word 2896
.word 2967
.word 3035
```

```
.word 3102
.word 3166
.word 3229
.word 3290
.word 3349
.word 3461
.word 3513
.word 3564
.word 3612
.word 3659
.word 3703
.word 3745
.word 3784
.word 3822
.word 3857
.word 3889
.word 3920
.word 3948
.word 3973
.word 3996
.word 4017
.word 4036
.word 4052
.word 4065
 .word 4076
.word 4085
.word 4091
.word 4095
.word 4096
.word 4095
.word 4091
.word 4085
.word 4076
.word 4065
.word 4052
.word 4036
.word 4017
.word 3996
.word 3973
.word 3948
.word 3920
.word 3889
.word 3857
.word 3822
.word 3784
.word 3745
```

```
.word 3703
.word 3659
.word 3612
.word 3564
.word 3513
.word 3461
.word 3406
.word 3349
.word 3290
.word 3229
.word 3166
.word 3102
.word 3035
.word 2967
.word 2896
.word 2824
.word 2751
.word 2675
.word 2598
.word 2520
.word 2440
.word 2359
.word 2276
.word 2191
.word 2106
.word 2019
.word 1931
.word 1842
.word 1751
.word 1660
.word 1567
.word 1474
.word 1380
.word 1285
.word 1189
.word 1092
.word 995
.word 897
.word 799
.word 700
.word 601
.word 501
.word 401
.word 301
.word 201
.word 101
```

```
.word 0
 .word 65435
 .word 65335
 .word 65235
 .word 65135
.word 65035
 .word 64935
 .word 64836
 .word 64737
 .word 64639
 .word 64541
.word 64444
 .word 64347
 .word 64251
 .word 64156
.word 64062
 .word 63969
.word 63876
.word 63785
.word 63694
.word 63605
.word 63517
.word 63430
.word 63345
.word 63260
.word 63177
.word 63096
.word 63016
.word 62938
.word 62861
.word 62785
.word 62712
.word 62640
.word 62569
.word 62501
.word 62434
.word 62370
.word 62307
.word 62246
.word 62187
.word 62130
.word 62075
.word 62023
.word 61972
.word 61924
.word 61877
```

```
.word 61833
.word 61791
.word 61752
.word 61714
.word 61679
.word 61647
.word 61616
.word 61588
.word 61563
.word 61540
.word 61519
.word 61500
.word 61484
.word 61471
.word 61460
.word 61451
.word 61445
.word 61441
.word 61440
.word 61441
.word 61445
.word 61451
.word 61460
.word 61471
.word 61484
.word 61500
.word 61519
.word 61540
.word 61563
.word 61588
.word 61616
.word 61647
.word 61679
.word 61714
.word 61752
.word 61791
.word 61833
.word 61877
.word 61924
.word 61972
.word 62023
.word 62075
.word 62130
.word 62187
.word 62246
.word 62307
```

```
.word 62370
.word 62434
.word 62501
.word 62569
.word 62640
.word 62712
.word 62785
.word 62861
.word 62938
.word 63016
.word 63096
.word 63177
.word 63260
.word 63345
.word 63430
.word 63517
.word 63605
.word 63694
.word 63785
.word 63876
.word 63969
.word 64062
.word 64156
.word 64251
.word 64347
.word 64444
.word 64541
.word 64639
.word 64737
.word 64836
.word 64935
.word 65035
.word 65135
.word 65235
.word 65335
.word 65435
```

```
 LDP  #0            ;point to memory page 0
;context restore from the software stack
     MAR  *, AR1         ;ARP = AR1
     MAR  *-            ;SP points to last entry
     LACL *-            ;restore ACCL
     ADD  *-,16          ;restore ACCH
     LST  #0, *-        ;restore ST0
     LST  #1, *-        ;restore ST1, unskip one stack location
     CLRC INTM           ;re-enable interrupts
     RET               ;return from the interrupt

GISR4:
    MAR  *,AR1          ;AR1=stack pointer
     MAR  *+            ;skip one stack location (required)
     SST  #1, *+        ;save ST1
     SST  #0, *+        ;save ST0
     SACH *+            ;save ACCH
     SACL *+
    ldp #CAPCOUNT
    LACC CAPCOUNT
           ADD #1h
           SACL CAPCOUNT
    LDP  #0            ;point to memory page 0
;context restore from the software stack
RESTORE
     MAR  *, AR1         ;ARP = AR1
     MAR  *-            ;SP points to last entry
     LACL *-            ;restore ACCL
     ADD  *-,16          ;restore ACCH
     LST  #0, *-        ;restore ST0
     LST  #1, *-        ;restore ST1, unskip one stack location
     CLRC INTM
     RET

PHANTOM:
     CLRC INTM
     RET
```

DECLARATION

I, the undersigned, hereby declare that this thesis is my original work, performed under the supervision of Dr. Mengesha Mamo , and has not been presented as a thesis for a degree program in this or any other university, and all sources of materials used for the thesis are duly acknowledged.

Name: Henok Berehanu

Signature: ___________

Place: Addis Ababa

Date of submission: __________

This thesis has been submitted for examination with my approval as a university advisor.

Dr. Mengesha Mamo

Signature_____________

December 2010

Addis Ababa, Ethiopia