Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

Analysis and DSP Implementation of Sensor-less Direct FOC of Three-Phase Induction Motor Using Open-Loop Speed Estimator

By

TESHOME HAMBISSA

Thesis Submitted To Addis Ababa Institute of Technology in Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical and Computer Engineering (Control Engineering)

Advisor:

Dr. MENGESHA MAMO

May 17, 2017

Addis Ababa, Ethiopia

Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

Analysis and DSP Implementation of Sensor-less Direct FOC of Three-Phase Induction Motor Using Open-Loop Speed Estimator

By:

Teshome Hambissa

Approval by Board of Examiners

_____	_____	_____
Chairman School of Graduate Committee	Signature	Date
Dr.Mengesha Mamo	_____	_____
Advisor	Signature	Date
_____	_____	_____
Internal Examiner	Signature	Date
_____	_____	_____
External Examiner	Signature	Date

Declaration

I, the undersigned, declared that this MSc thesis is my original work, has not been presented for fulfillment of a degree in this or any other University and all sources and materials used for the thesis is acknowledged.

Teshome Hambissa _____

Name Signature

Addis Ababa _____

Place Date of Submission

This thesis work has been submitted for examination with my approval as a University Advisor.

Dr. Mengesha Mamo _____

Advisor's Name Signature

Acknowledgments

I want to start expressing a sincere acknowledgement to my advisor, Dr. MENGESHA MAMO for giving me the opportunity to research under his guidance and supervision. I received motivation, comments, encouragement and continuous guidance from him during my graduate studies.

And to the Father Almighty, may you continue to give me strength and vision that I may follow your path…..to eternal salvation.

Last but not least, I would like to thank my wife for her great encouragement from day to day during my study.

List of Symbols

$\theta_{\varphi r}$	Rotor flux angle
φ_{ds}^{s}	d axis stator flux in stationary frame
φ_{qs}^{s}	q axis stator flux in stationary frame
φ_{ds}^{e}	d axis stator flux in synchronous frame
φ_{qs}^{e}	q axis stator flux in synchronous frame
w_e	Angular synchronous speed
w_r	Angular rotor speed
w_{sl}	Angular slip speed
θ_e	Angle between the synchronous frame and the stationary frame
i_{ds}^{e}	d axis stator current in synchronous frame
i_{qs}^{e}	q axis stator current in synchronous frame
i_{ds}^{s}	d axis stator current in stationary frame
i_{qs}^{s}	q axis stator current in stationary frame
i_{as}	Phase a-stator current
i_{bs}	Phase b-stator current
i_{as}	Phase c- stator current
L_m	Magnetizing inductance
L_{ls}	Stator leakage inductance
L_{lr}	Rotor leakage inductance
L_s	Stator self-inductance
L_r	Rotor self-inductance
X_m	Stator magnetizing reactance
X_{ls}	Stator leakage reactance
X_{lr}	Rotor leakage reactance
X_s	Stator self-reactance
X_r	Rotor self-reactance
R_s	Stator resistance
R_r	Referred rotor resistance
T_{em}	Electromagnetic torque

T_L	Load Torque
T_r	Rotor time-constant
V_{as}	Phase-a stator voltage
V_{bs}	Phase-b stator voltage
V_{cs}	Phase-c stator voltage
V_{qs}^{s}	d axis stator voltage in stationary frame
V_{ds}^{s}	q axis stator voltage in stationary frame
V_{ds}^{e}	d axis stator voltage in synchronous frame
V_{qs}^{e}	q axis stator voltage in synchronous frame

List of Abbreviation

V_dc	Dc-link voltage
HVACI	High Voltage AC induction
ISR	Interrupt Service routine
ADC	Analog to Digital Converter
DFOC	Direct Field Oriented Control
S/W	Software
SOC	System on Chip
SVGEN	Space Vector PWM with Quadrature Control
ACI_FE	AC Induction Flux Estimator
ACI_SE	AC Induction Speed Estimator
VMFE	Voltage Model Flux Estimator
CMFE	Current Model Flux Estimator

Abstract

This thesis work deals with sensor-less speed control of induction motor in its entire speed range. Flux linkage and rotor speed estimators are designed, simulated and implemented on 180W induction motor using Texas Instrument DSP board TMDSHVMTRPFCKIT. Estimated flux-linkage has been used as reference for direct field oriented control of the machine currents and voltages. MATLAB software is used for design simulation of the estimators while Texas Instruments TMS320F2803x, power module with space vector modulation, inverter and code composer studio for software development, is used for practical experimentation. PI controller was used for both speed and current controller. The simulation and experimental result demonstrated good dynamic and steady state performance 3.38% overshoot and 31.4 rad/sec bandwidth have been achieved with 0.00375 steady state error.

Keywords: sensor-less , Texas Instrument , direct field oriented control

Contents

List of Tables

List of Figures

Chapter One

Introduction

1.1. Background Information

Ac drive based on full digit control has reached the status of a matured technology. Ongoing research has concentrated on the elimination of the speed sensor at the machine shaft without deteriorating the dynamic performance of the drive control system [1]. Speed estimation is an issue of particular interest with induction motor drives where the mechanical speed of the rotor is generally different from the speed of the revolving magnetic field.

The advantage of speed control of sensor-less induction motors are reduced hardware complexity and lower cost, reduced the size of drive machine, elimination of the sensor cable, better noise immunity, increased reliability and less periodic maintenance requirement. The operation in hostile environment mostly requires a motor without speed sensor. Besides these it is cheaper than the other types of electrical motors. However, a nonlinear model of induction motor (IM) along with inaccessibility rotor currents make the control of IM a challenging task. Despite the challenges several schemes have been used to control the induction motor [2].These schemes can be classified into two main category, Scalar control and Vector control.

Scalar control: one of the first ways of controlling induction motor was the Volts/hertz speed control in which the motor was excited with constant voltage to frequency ratio in order to maintain a constant air gap flux and hence provide maximum torque sensitivity. This method is relatively simple but does not yield satisfactory results for high performance application. This is due to the fact that in the scalar control, an inherent coupling exists between the torque and air gap flux, which leads to sluggish response of the induction motor.

Field oriented control (Vector control),to overcome the limitation of the scalar control method, field oriented method was developed. The basic principle of vector control is to split the components of stator current responsible for production of flux, and torque. And to accomplish that, the state variables used in dynamic model of IM are transformed into a reference frame in which independent control of the machine flux and torque achieved. During the last few decades the field of controlled electrical drives has undergone rapid expansion due mainly to the benefits of microcontrollers. These technological improvements have enabled the development of very effective AC drive control with lower power dissipation hardware and more accurate control structures. The electrical drive

controls become more accurate in the sense that not only are the DC quantities controlled but also the three phase ac currents and voltages are managed by so-called vector controls. This thesis briefly describes the implementation of the most efficient form of a vector control scheme: the Field Orientated Control method. 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 and effective Space Vector Pulse Width Modulation pattern generation.

1.1. Statement of the problem

In recent years, a large number of speed sensor-less vector control systems for induction motor (IM) have been proposed. Speed information is generally provided by a speed transducer on the motor shaft. Recently, low cost and high performance digital signal processors (DSP) become available allowing obtaining speed by means of digital estimators integrated with motor control. This solution represents an advantage in terms of costs, simplicity and mechanical reliability of the drive.

The scalar V/f method is able to provide speed variation but does not handle transient condition control and is valid only during steady state. This method is most suitable for applications without position control requirements or the need for high accuracy of speed control and leads to over-currents and overheating, which necessitate a drive which is then oversized and no longer cost effective and does not increase dynamic performance. The common approach to increase dynamic performance and stability of speed sensor-less field oriented control systems is Vector control. The main contribution of this paper is a novel speed sensor-less vector control based on field oriented control (DFOC).

1.2. Motivation

Digital signal processing technology is enabling cost effective and energy efficient control system design. In addition to the advantage of digital control system, running IM without position mechanical sensor has its own advantage over IM with position sensor:

- It minimizes the drive cost.
- It decreases mechanical size of the machine
- It increases the reliability of the system.
- It improves the noise immunity efficiency of the system.
- It avoids the complex process of embedding the position sensor into the stator.

1.3. Objective

The above facts invite working in this area. Hence, this thesis has a general objective of developing an embedded controller for a sensorless speed control of induction motor using a Texas Instrument DSP of TMS320F2803x with specific objectives of:

- To design a flux estimator for position and speed of the motor.
- To design PI speed and current controller of the motor.
- To implement on Texas Instruments TMS320F2803x and analyze the performance of the estimator.
- To simulating the speed control of the induction motor using vector control realized by space vector which has been done on MATLAB/SIMULINK

1.4. Thesis methodology

For the accomplishment of open loop speed estimator controlling of IM using DSP, the following methodologies have been followed:

- Gathering & survey related works from various literature and publication.
- Study and design the appropriate modelling of the system.
- Design an appropriate estimator & Controller for the system.
- Analyze the design with Matlab/Simulink simulation.
- Build a rotor flux estimator and controller algorithm to program the system into DSP.
- Code the algorithm with C programming language in to the DSP using code composer studio.

Rotor Flux Position: Knowledge of the rotor flux position is the core of the FOC. In fact if there is an error in this variable the rotor flux will not be aligned with the d-axis and i_{sd} and i_{sq}, the flux and torque components of the stator current, will be incorrect. In this implementing system, the direct (rotor) flux oriented control system with flux and open-loop speed estimators is described. The overall block diagram of this thesis work can be depicted in Figure. 1.1. Shown below.

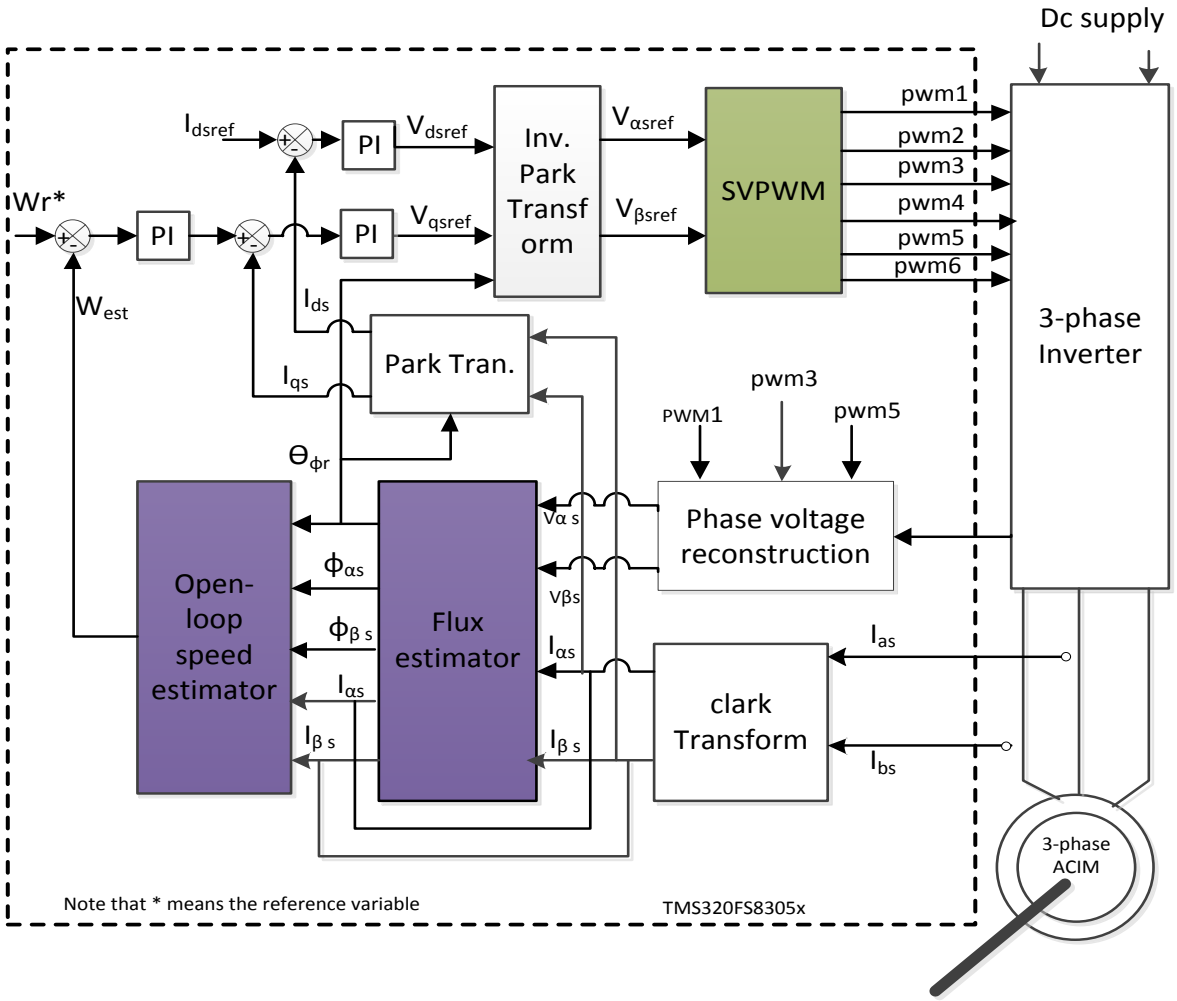


Figure 1.1: Overall block diagram of direct rotor flux oriented control

1.5. Literature review

As stated above, induction motor drive system is becoming more and more a competitive system in many high performance motion drive application. There are three main major components in an induction motor drive system: an induction motor, a power electronic device and a controller. The field orientation control (FOC), integrating modern control theory, power electronic and DSP/micro-processor technology, has made possible the development of high performance induction motor drive systems. The state of art of speed sensor-less control and motor loss minimization is reviewed. The existing problems in the implementation of induction motor drive system are outlined.

AC drives based on full digital control have reached the status of a mature technology [10], [20]. Ongoing research has concentrated on the elimination of the speed sensor at the machine shaft without deteriorating the dynamic performance of the drive control system. Problems associated with sensorless control systems have mainly included parameter sensitivity, integrator drift, and problems at low frequencies. Some have tried to solve these problems by redesigning the induction machine (Jansen et al, 1994a).

Investigation of ways to estimate the flux and speed of the induction machine has been extensively studied in the past two decades. Classically, the rotor flux was measured by using a special sensing element, such as Hall-effect sensors placed in the air-gap. An advantage of this method is that additional required parameters, rotor leakage inductance, Magnetizing inductance, and rotor self-inductance are not significantly affected by changes in temperature and flux level. However, the disadvantage of this method is that a flux sensor is expensive and needs special installation and maintenance.

Another flux and speed estimation technique is saliency based with fundamental or high frequency signal injection, P.L.Jansen and R.D.Lorenz(IEEE Tran. IA, vol. 31, no.2, pp.240-247,Mar/April, 1995). One advantage of saliency technique is that the saliency is not sensitive to actual motor parameters, but this method fails at low and zero speed level. When applied with high frequency signal injection, the method may cause torque ripples, vibration and audible noise [9].

Gabriel (IEEE Tran. IA, vol.16, no.2, pp.186.192, 1980) avoided the special flux sensors and coils by estimating the rotor flux from the terminal quantities (stator voltages and currents) [10]. This technique requires the knowledge of the stator resistance along with the stator-leakage, and rotor leakage inductances and the magnetizing inductance. This method is not working at low speed. By introducing of current model

and working with voltage model in open-loop speed control, Gabriel's approach is modified and the system's simulation as well as experimental setup is successful.

It is clear that the control systems with mechanical speed sensor have a higher cost, reduce the reliability and require constant maintenance. Induction motor control systems without speed sensor eliminate above disadvantages. This is the reason that nowadays the sensorlees speed control of induction motors industry is in focus of many researchers (Kubota, 2002), (Kwon, 2004, overshoot 0.87 %, settling time 0.11 s and steady state error 0.67%), (Jemli, 1998), (Salem, 2005). Nowadays field oriented control is the most common way for sensorlees speed control of induction motors. Independence torque and flux control is the main advantage of this method (Quang, 2008).

In order to implement the sensorlees speed control, the dynamic model of induction motor must be able to allow independent torque and flux control (Trzynadlowski, 2000). Independent torque and flux control of induction motor provides high accuracy and fast dynamic speed response even in transient regimes. Rotor flux oriented control of three phase induction motors is one of the most effective method for sensorlees speed control between the various methods of field oriented control, because it is simple to implement and very robust method.

1.6. Thesis organization

The thesis is organized into six chapters including this introduction. **In chapter 2**, brief over view about principle operation of induction motor, concept of vector control and scheme of field orientation are present. Furthermore, the dynamic model of IM along with space vector pulse width modulation (SVPWM) and PI speed and current controller are presented and are discussed in this chapter. **Chapter 3** deals with the design of senseless open loop speed estimation by using direct field oriented control (DFOC) along with rotor flux estimation. A simulation result for the speed tracking performance for IM drive under rotor flux estimation is discussed rigorously in **chapter 4**.The developed software by code composer studio for the proposed IM drive scheme and the open loop practical experimentation of speed control is are present and discussed in **chapter 5**.and finally, concluding remarks and future works are presented in **chapter 6**.

Chapter Two

Induction Machine Modeling

2.1. Introduction

An induction motor like any other rotating machine consists of a stator (the fixed part) and a rotor (the moving part) separated by air gap. The stator contains electrical windings housed in axial slots. The induction machine used in the industries are mainly three phase, except for small power were single phase machines are common. The two names for the same type of motor, Induction motor and Asynchronous motor, describe the two characteristics in which this type of motor differs from DC motors and synchronous motors. Induction refers to the fact that the field in the rotor is induced by the stator currents, and asynchronous refers to the fact that the rotor speed is not equal to the stator frequency. No sliding contacts and permanent magnets are needed to make an induction motor work, which makes it very simple and cheap to manufacture. As motors, they are rugged and require very little maintenance. However, their speeds are not as easily controlled as with DC motors. They draw large starting currents, and operate with a poor lagging factor when lightly loaded [3,15].

2.2. Principle of operation of induction machine

Induction motors have two main components, the stator and the rotor. The stator carries a three-phase winding that receives power from the supply. The rotor carries a winding that is in the form of a set of single-bar conductors placed in slots just below the surface of the rotor. The slots have a narrow opening at the surface of the rotor, which serves to lock the conductor bars in position. Each end of each bar conductor is connected to a short-circuiting ring, one at each end of the rotor. The stator winding is a conventional type as found in three-phase generators and synchronous motors.The three-phase stator winding produces a rotating field of constant magnitude, which rotates at the speed corresponding to the frequency of the supply and the number of poles in the motor and which related by equation (2.1)[25].

$$N_s = \frac{120}{P} f_s \tag{2.1}$$

Where Ns is synchronous speed in rpm, f_s is the frequency of the applied excitation in Hz, and P is the number of poles. The higher the number of poles the lower the speed of the rotation. The rotor speed cannot reach the same speed as that of the stator field, otherwise there would be no induced emfs and currents in the rotor, and no torque would be developed. Consequently when the rotor speed is near to the synchronous speed

the torque begins to decrease rapidly until it matches that of the load and rotational friction and windage losses. When this balance is achieved the speed will remain constant and with rotor frequency reduced with slip and related as in equation (2.2) [14, 25].

$$N_r = \frac{N_s - N_r}{N_s} \; f_s \tag{2.2}$$

Where s is the slip.

2.3. Induction machine dynamic model

The control of an induction motor can be made similar to that of a DC machine with vector control technique, where it is possible to have independent control of flux and torque. In order to achieve this, the mathematical model of the motor in a rotating reference frame has to be synchronized either to the stator, air gap or rotor flux vector. For this one should know the angle of the stator, air gap or rotor flux vector along with their magnitude.

2.4. General equations of AC Motors

AC induction motor is represented by a stator and a concentric rotor with three-phase windings and a narrow air gap between the smooth surfaces of the rotor and the stator. Only the stator windings are fed by a voltage or current source inverter. The connection between the stator and the rotor is made by the flux, which means there is no mechanical connection between them apart from the bearings. For deducing the general equations of the AC induction motor (with "squirrel cage" rotor), let as make the following assumptions:

- neglect the cooper losses and the slots in the machine;
- spatial distribution of fluxes and amper turns wave are considered sinusoidal
- all the losses due to wiring, saturation, and slot effects can be neglected
- Stator and rotor permeability are assumed to be infinite.

Model of an induction machine in the synchronous frame [15, 16, 25]:

Stator dq0 voltage equations:

$$v_{ds}^{e} = R_s i_{ds}^{e} - \omega_e \varphi_{ds}^{e} + \frac{d\varphi^{e}{}_{ds}}{dt} \tag{2.3}$$

$$v_{qs}^{e} = R_s i_{qs}^{e} + \omega_e \varphi_{qs}^{e} + \frac{d\varphi^{e}{}_{qs}}{dt} \tag{2.4}$$

Where v_{ds}^{e}, v_{qs}^{e} *is stator voltages along* $d, q -$ *axis respectively*

Rotor dq0 voltage equations:

$$v_{dr}^{e} = R_r i_{dr}^{e} - (\omega_e - \omega_r)\varphi_{dr}^{e} + \frac{d\varphi^{e}_{dr}}{dt} \quad (2.5)$$

$$v_{qr}^{e} = R_r i_{qr}^{e} + (\omega_e - \omega_r)\varphi_{qr}^{e} + \frac{d\varphi^{e}_{qr}}{dt} \quad (2.6)$$

Where v_{dr}^{e}, v_{qr}^{e} *is rotor voltages along* $d, q - axis$ *respectively*

Stator and rotor linkage flux in synchronous reference frame in Metrics form is given as follow.

$$\begin{bmatrix} \varphi_{ds}^{e} \\ \varphi_{qs}^{e} \\ \varphi_{dr}^{e} \\ \varphi_{qr}^{e} \end{bmatrix} = \begin{bmatrix} L_s & 0 & L_m & 0 \\ 0 & L_s & 0 & L_m \\ L_m & 0 & L_r & 0 \\ 0 & L_m & 0 & L_r \end{bmatrix} \begin{bmatrix} i_{ds}^{e} \\ i_{qs}^{e} \\ i_{dr}^{e} \\ i_{qr}^{e} \end{bmatrix} \quad (2.7)$$

And the torque developed by the motor in terms of q-axis current and flux linkage, using 'rms' convention [27], is

$$T_e = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_r}\left(\varphi_{rd} i_{sq} - \varphi_{rq} i_{sd}\right) \quad (2.8)$$

The electromagnetic torque of the mechanical system is given by;

$$T_e = J\frac{dw_r}{dt} + f w_r + T_L \quad (2.9)$$

Where w_r *is rotor speed in RPM* and J-is moment of inertia and f-friction coefficient.

Model of an induction machine in the Stationary reference frame [15, 16, 25]:

Let us choose first a standstill co-ordinate system $\omega_e = 0$(where ω_e is synchronous speed), which means that the common co-ordinate system is a stationary one. Replacing the value $\omega_e = 0$ in the general equations of the motor given by (2.3), (2.4), (2.5) and (2.6), the following vectorial voltage equations conclude:

Stator voltage equations in stationary reference frame:

$$v_{ds}^{s} = R_s i_{ds}^{s} + \frac{d\varphi_{ds}^{s}}{dt} \tag{2.10}$$

$$v_{qs}^{s} = R_s i_{qs}^{s} + \frac{d\varphi_{qs}^{s}}{dt} \tag{2.11}$$

Rotor voltage equations in stationary reference frame*:*

$$v_{dr}^{s} = R_s i_{dr}^{s} + \frac{d\varphi_{dr}^{s}}{dt} + w_r\, \varphi_{qr}^{s} = 0 \tag{2.12}$$

$$v_{qr}^{s} = R_s i_{qr}^{s} + \frac{d\varphi_{qr}^{s}}{dt} - w_r\, \varphi_{dr}^{s} = 0 \tag{2.13}$$

The stator and rotor voltage equations and the electromagnetic torque equations can be represented in matrix form as given below separately;

$$\begin{bmatrix}\varphi_{ds}\\ \varphi_{qs}\end{bmatrix} = \begin{bmatrix}L_s & 0\\ 0 & L_s\end{bmatrix}\begin{bmatrix}i_{ds}\\ i_{qs}\end{bmatrix} + \begin{bmatrix}L_m & 0\\ 0 & L_m\end{bmatrix}\begin{bmatrix}i_{dr}\\ i_{qr}\end{bmatrix} \tag{2.14}$$

$$\begin{bmatrix}\varphi_{dr}\\ \varphi_{qr}\end{bmatrix} = \begin{bmatrix}L_r & 0\\ 0 & L_r\end{bmatrix}\begin{bmatrix}i_{dr}\\ i_{qr}\end{bmatrix} + \begin{bmatrix}L_m & 0\\ 0 & L_m\end{bmatrix}\begin{bmatrix}i_{ds}\\ i_{qs}\end{bmatrix} \tag{2.15}$$

So, neglecting the magnetic saturation of the circuit and iron losses, the flux linkage equations (2.3 and 2.4) of the circuit can be represented in matrix form as given below separately:-

$$\begin{bmatrix}v_{ds}\\ v_{qs}\end{bmatrix} = \begin{bmatrix}R_s & 0\\ 0 & R_s\end{bmatrix}\begin{bmatrix}i_{ds}\\ i_{qs}\end{bmatrix} + \frac{d}{dt}\begin{bmatrix}\varphi_{ds}\\ \varphi_{qs}\end{bmatrix} + \begin{bmatrix}0 & -w_e\\ w_e & 0\end{bmatrix}\begin{bmatrix}\varphi_{ds}\\ \varphi_{qs}\end{bmatrix} \tag{2.16}$$

$$\begin{bmatrix}v_{dr}\\ v_{qr}\end{bmatrix} = \begin{bmatrix}R_r & 0\\ 0 & R_r\end{bmatrix}\begin{bmatrix}i_{dr}\\ i_{qr}\end{bmatrix} + \frac{d}{dt}\begin{bmatrix}\varphi_{dr}\\ \varphi_{qr}\end{bmatrix} + \begin{bmatrix}0 & -(w_e - w_r)\\ (w_e - w_r) & 0\end{bmatrix}\begin{bmatrix}\varphi_{dr}\\ \varphi_{qr}\end{bmatrix} \tag{2.17}$$

$$T_e = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_r}\begin{bmatrix}\varphi_{dr} & \varphi_{qr}\end{bmatrix}\begin{bmatrix}i_{ds}\\ -i_{qs}\end{bmatrix} \tag{2.18}$$

Solving equation (2.15) for the rotor currents i_{dr}^{s} and i_{qr}^{s} , and putting these values in equation (2.14), we get:

$$\begin{bmatrix}\varphi_{ds}\\ \varphi_{qs}\end{bmatrix} = \begin{bmatrix}\sigma L_s & 0\\ 0 & \sigma L_s\end{bmatrix}\begin{bmatrix}i_{ds}\\ i_{qs}\end{bmatrix} + \begin{bmatrix}\frac{L_m}{L_r} & 0\\ 0 & \frac{L_m}{L_r}\end{bmatrix}\begin{bmatrix}\varphi_{dr}\\ \varphi_{qr}\end{bmatrix} \tag{2.19}$$

Where the total leakage factor σ is defined as $\sigma = 1 - \frac{L_m^2}{L_s.L_r}$

Putting the values of rotor d- and q-axis voltage as zero and the values of rotor d- and q-axis currents from equation (2.15) in (2.16), we get;

$$\frac{d}{dt}\begin{bmatrix}\varphi_{dr}\\ \varphi_{qr}\end{bmatrix} = \frac{L_m}{T_r}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}i_{ds}\\ i_{qs}\end{bmatrix} - \frac{1}{T_r}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\varphi_{dr}\\ \varphi_{qr}\end{bmatrix} + (w_e - w_r)\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}\begin{bmatrix}\varphi_{dr}\\ \varphi_{qr}\end{bmatrix} \qquad (2.20)$$

Again, substituting, the values of stator fluxes from equation (2.19) in (2.16), we get,

$$\frac{d}{dt}\begin{bmatrix}i_{ds}\\ i_{qs}\end{bmatrix} = \begin{bmatrix}-R_1 & w_e\\ w_e & -R_1\end{bmatrix}\begin{bmatrix}i_{ds}\\ i_{qs}\end{bmatrix} + \begin{bmatrix}R_2 & R_3 w_r\\ -R_3 w_r & R_2\end{bmatrix}\begin{bmatrix}\varphi_{dr}\\ \varphi_{qr}\end{bmatrix} + \begin{bmatrix}\frac{1}{\sigma L_s} & 0\\ 0 & \frac{1}{\sigma L_s}\end{bmatrix}\begin{bmatrix}v_{ds}\\ v_{qs}\end{bmatrix} \qquad (2.21)$$

So, the mathematical model or the state space model of the induction motor in terms of stator currents and rotor flux linkages as the state variables are obtained by combining equation (2.20) and (2.21), as follows.

$$\frac{d}{dt}\begin{bmatrix}i_{ds}\\ i_{qs}\\ \varphi_{dr}\\ \varphi_{qr}\end{bmatrix} = \begin{bmatrix}-R_1 & w_e & R_2 & R_3 w_r\\ -w_e & -R_1 & R_3 w_r & R_2\\ R_5 & 0 & -R_4 & w_{sl}\\ 0 & R_5 & -w_{sl} & -R_4\end{bmatrix}\begin{bmatrix}i_{ds}\\ i_{qs}\\ \varphi_{dr}\\ \varphi_{qr}\end{bmatrix}\begin{bmatrix}\frac{1}{\sigma L_s} & 0\\ 0 & \frac{1}{\sigma L_s}\\ 0 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}v_{ds}\\ v_{qs}\end{bmatrix} \qquad (2.22)$$

Where $R_1 = \frac{1}{\sigma L_s}\left(R_s + R_r\frac{L_m^2}{L_r^2}\right)$, $R_2 = \frac{1}{\sigma L_s}\frac{R_r L_m}{L_r^2}$, $R_3 = \frac{1}{\sigma L_s}\frac{L_m}{L_r}$, $R_4 = \frac{1}{T_r}$ *and* $R_5 = \frac{R_r L_m}{L_r}$

$$w_{sl} = w_e - w_r = slip\ speed\ ,$$

2.5. Field Orientated Control (FOC)

Field-oriented control (also referred to as vector control) is often used in high performance drive applications. A simple control such as the V/Hz strategy has limitations on the performance. Such as, dynamic performance was unsatisfactory because of saturation effect and the electrical parameter variation with temperature. This results in excessive current and over-heating, which necessitates the drive to be oversized. This over-design no longer makes the motor cost effective due to high cost of the drive circuitry [1].

To achieve better dynamic performance, a more complex control scheme needs to be applied, to control the induction motor. With the mathematical processing power offered by the microcontrollers, we can implement advanced control strategies, which use mathematical transformations in order to decouple the torque generation and the magnetization functions in an AC induction motor. Such de-coupled torque and magnetization control is commonly called rotor flux oriented control, or simply Field Oriented Control (FOC) [15],[17-19].

2.5.1. Basic Principles of Field-Oriented Control

In order to understand the spirit of the Field Oriented Control technique, let us start with an overview of the separately excited direct current (DC) Motor. In this type of motor, the excitation for the stator and rotor is independently controlled. An electrical study of the DC motor shows that the produced torque and the flux can be independently tuned. The strength of the field excitation (i.e. the magnitude of the field excitation current) sets the value of the flux. The current through the rotor windings determines how much torque is produced. The commutator on the rotor plays an interesting part in the torque production. The commutator is in contact with the brushes, and the mechanical construction is designed to switch into the circuit the windings that are mechanically aligned to produce the maximum torque. This arrangement then means that the torque production of the machine is fairly near optimal all the time. The key point here is that the windings are managed to keep the flux produced by the rotor windings orthogonal to the stator field.

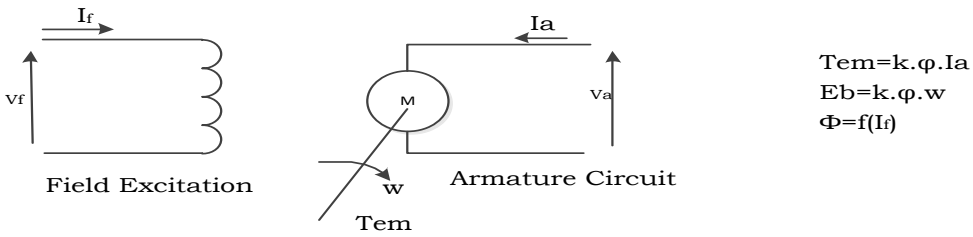


Figure 0-1.1 Separated excitation DC motor model, flux and torque are independently controlled.

Induction machines do not have the same key features as the DC motor. However, in both cases we have only one source that can be controlled which is the stator currents. On the synchronous machine, the rotor excitation is given by the permanent magnets mounted onto the shaft. On the synchronous motor, the only source of power and magnetic field is the stator phase voltage. Obviously, as opposed to the DC motor, flux and torque depend on each other. The goal of the FOC (also called vector control) on synchronous and asynchronous machine is to be able to separately control the torque producing and magnetizing flux components. The control technique goal is to (in a sense) imitate the DC motor's operation [24].

2.5.2. The Basic Scheme for the FOC

The following diagram summarizes the basic scheme of torque control with FOC:

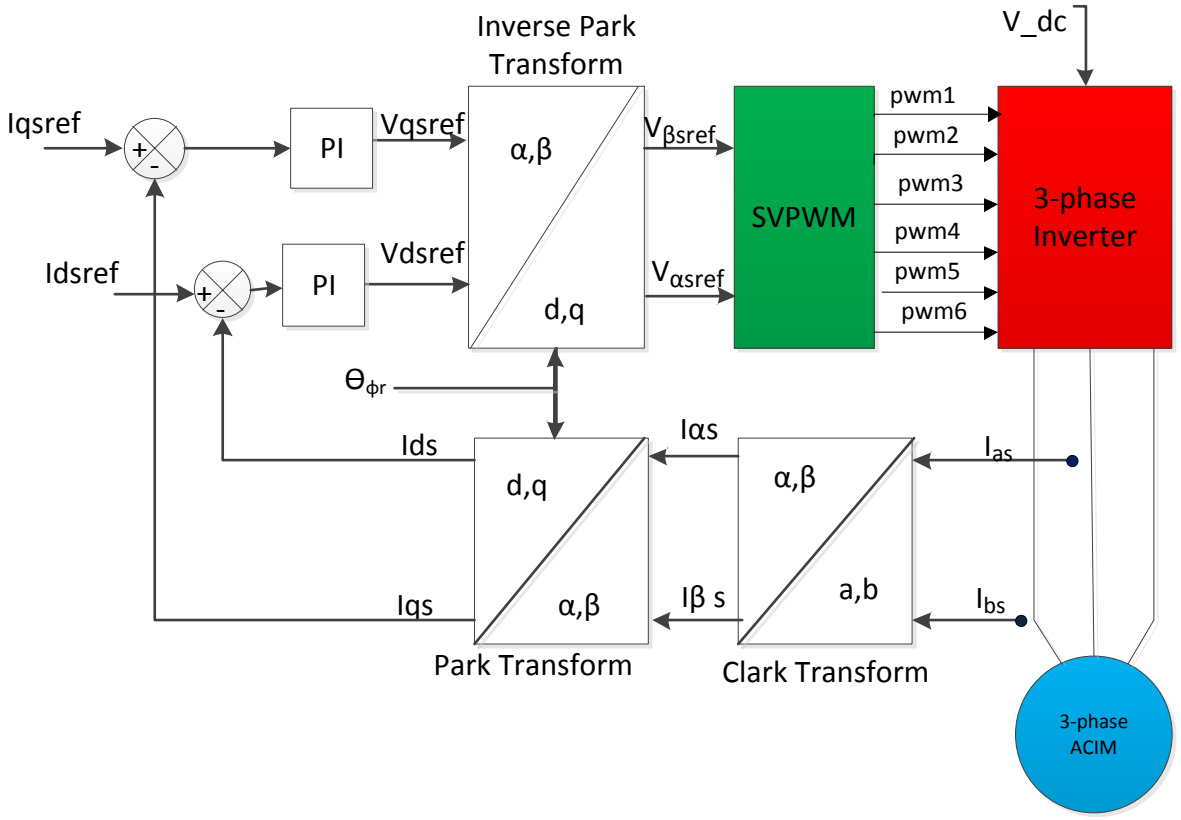


Figure 2.2: Basic scheme of FOC for ACI motor

Two motor phase currents are measured. These measurements feed the Clarke transformation module. The outputs of this projection are designated $i_{s\alpha}$ and $i_{s\beta}$. 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 is_{dref} (the flux reference) and is_{qref} (the torque reference). At this point, this control structure shows an interesting advantage: it can be used to control induction machines by simply changing the flux reference and obtaining rotor flux position. Since induction motors need a rotor flux creation in order to operate, the flux reference must not be zero. This conveniently solves one of the major drawbacks of the "classic" control structures: "the portability from asynchronous to synchronous drives". The torque command isqref could be the output of the speed controller when we use a speed FOC. The outputs of the current controllers are Vsdref and Vsqref; they are applied to the inverse Park transformation. The outputs of this projection are Vsαref and Vsβref which are the components of the stator vector voltage in the 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. Obtaining this rotor flux position depends on the AC machine type (synchronous or asynchronous machine).

The Field Orientated Control 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.

The concept of field orientation control is used to accomplish a decoupled control of flux and torque. This concept is copied from dc machine direct torque control that has three requirements [4]:

- an independently controlled armature current to overcome the effects of armature winding resistance, leakage inductance and induced voltage
- an independently controlled constant value of flux
- an independently controlled orthogonal spatial angle between the flux axis and magneto motive force (MMF) axis to avoid interaction of MMF and flux.

If all of these three requirements are met at every instant of time, the torque will follow the current, allowing an immediate torque control and decoupled flux and torque control. Next a two phase d-q model of an induction machine rotating at the synchronous speed is introduced which will help to carry out this decoupled control concept to the induction machine. This model can be summarized by the following equations [5] (see chapter 3 for detail):

$$v_{ds}^{e} = \frac{d\varphi_{ds}^{e}}{dt} - w_e \varphi_{qs}^{e} + R_s i_{ds}^{e} \tag{2.23}$$

$$v_{qs}^{e} = \frac{d\varphi_{qs}^{e}}{dt} + w_e \varphi_{ds}^{e} + R_s i_{qs}^{e} \tag{2.24}$$

$$0 = \frac{d\varphi_{dr}^{e}}{dt} - (w_e - w_r)\varphi_{qr}^{e} + R_r i_{dr}^{e} \tag{2.25}$$

$$0 = \frac{d\varphi_{qr}^{e}}{dt} + (w_e - w_r)\varphi_{dr}^{e} + R_r i_{qr}^{e} \tag{2.26}$$

$$\varphi_{ds}^{e} = L_s i_{ds}^{e} + L_m i_{dr}^{e} \quad (2.27)$$

$$\varphi_{qs}^{e} = L_s i_{qs}^{e} + L_m i_{qr}^{e} \quad (2.28)$$

$$\varphi_{dr}^{e} = L_r i_{dr}^{e} + L_m i_{ds}^{e} \quad (2.29)$$

$$\varphi_{qr}^{e} = L_r i_{qr}^{e} + L_m i_{qs}^{e} \quad (2.30)$$

This model is quite significant to synthesize the concept of field-oriented control. In this model it can be seen from the torque expression (2.8) that if the rotor flux along the q-axis is zero, then all the flux is aligned along the d-axis and therefore the torque can be instantaneously controlled by controlling the current along q-axis. Then the question will be how it can be guaranteed that all the flux is aligned along the d-axis of the machine. When a three-phase voltage is applied to the machine, it produces a three-phase flux both in the stator and rotor. The three-phase fluxes can be converted into equivalents developed in two-phase stationary $(d^s - q^s)$ frame. If this two phase fluxes along $(d^s - q^s)$ axes are converted into an equivalent single vector then all the machine flux will be considered as aligned along that vector. This vector commonly specifies us $d^e -$ axis which makes an angle θ_e with the stationary frame $d^s -$ axis. The $q^e -$ axis is set perpendicular to the d^e-axis. The flux along the $q^e -$ axis in that case will obviously be zero. The phasor diagram Figure 2.3 shows these axes. The angle θ_e keeps changing as the machine input currents change. Thus the problem is to know the angle θ_e accurately, so that the d-axis of the $(d^e - q^e)$ frame is locked with the flux vector.

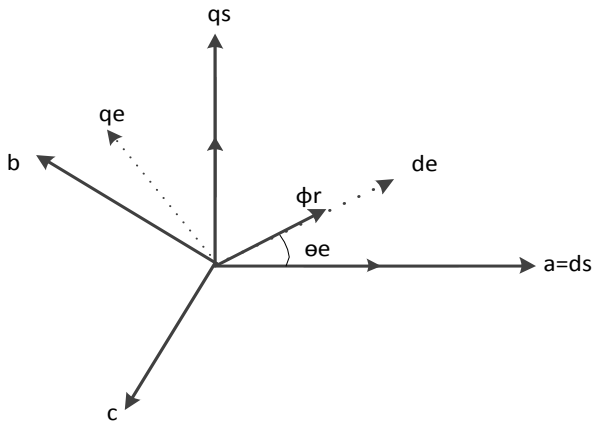


Figure 2.3: Phasor diagram of the field oriented drive system

The control input can be specified in terms of two phase synchronous frame i_{ds}^{e} and i_{qs}^{e}. i_{ds}^{e} is aligned along the $d^e -$ axis i.e. the flux vector, so does i_{qs}^{e} with the $q^e -$ axis. These two-phase synchronous control inputs

are converted into two-phase stationary and then to three- phase stationary control inputs. To accomplish this, the flux angle θ_e must be known precisely. The angle θ_e can be found either by Indirect Field Oriented Control (IFOC) or by Direct Field Oriented Control (DFOC).This thesis is intended to apply DFOC as it is proposed. The controller implemented in this fashion that can achieve a decoupled control of the flux and the torque is known as field oriented controller. The block diagram is shown in Figure 2.2. In the field oriented control the flux can be regulated in the stator, air-gap or in rotor flux orientation [4]-[7].

2.6. Background on Direct Field Oriented (DFO) Control

DFO control is a feedback method consisting of obtaining the principal command vector control parameters i_{ds}^{e*} and i_{qs}^{e*} , that is, the stator current torque and flux components referred to a rotating reference frame, from estimates of the rotor flux. This rotating reference frame moves synchronously with the rotor, allowing the manipulation of the flux and torque separately and in a decoupled manner. To do so, two reference frame transformations are needed; The first one to transform the three-phase reference frame variables (a-b-c) into a two-phase stationary reference frame variables (d^s-q^s) And the second one to transform these to the synchronously rotating reference frame(d^e-q^e) as explained below.

2.6.1. Clarke transformation

The Clarke transformation converts a three-phase signals such as currents, voltage, and flux from three-phase coordinate system (a, b, c) into a two-phase coordinate orthogonal system (α, β). The DFOC and sensorless control relies heavily on accurate flux estimation. DFOC is most often used for sensorless control, because the flux estimator used to estimate the synchronous speed or angle can also be used to estimate the machine speed. Investigation of ways to estimate the flux and speed of the induction machine has been extensively studied in the past two decades. Classically, the rotor flux was measured by using a special sensing element, such as Hall-effect sensors placed in the air-gap. An advantage of this method is that additional required parameters, L_{1r}, L_m, and L_r are not significantly affected by changes in temperature and flux level. However, the disadvantage of this method is that a flux sensor is expensive and needs special installation and maintenance. Another flux and speed estimation technique is saliency based with fundamental or high frequency signal injection. One advantage of saliency technique is that the saliency is not sensitive to actual motor parameters, but this method fails at low and zero speed level. When applied with high frequency signal injection, the method may cause torque ripples, vibration and audible noise [9].

Gabriel avoided the special flux sensors and coils by estimating the rotor flux from the terminal quantities (stator voltages and currents) [28]. This technique requires the knowledge of the stator resistance along with the stator-leakage, and rotor leakage inductances and the magnetizing inductance. This method is commonly known as the Voltage Model Flux Estimator (VMFE). In general, the word observer refers to estimators that employ integration process models, and implies essentially a real-time simulation of the physical process at hand, i.e., the induction machine in this work. The stator flux in the stationary reference frame can be estimated by the equations:

$$\frac{d\varphi_{ds}^{s}}{dt} = v_{ds}^{s} - i_{ds}^{s}R_s \tag{2.31}$$

$$\frac{d\varphi_{qs}^{s}}{dt} = v_{qs}^{s} - i_{qs}^{s}R_s \tag{2.32}$$

Then the rotor flux can be expressed as:

$$\varphi_{dr}^{s} = \frac{L_r}{L_m}(\varphi_{ds}^{s} - L_\sigma i_{ds}^{s}) \tag{2.33}$$

$$\varphi_{qr}^{s} = \frac{L_r}{L_m}\left(\varphi_{qs}^{s} - L_\sigma i_{qs}^{s}\right) \tag{2.34}$$

Where, $L_\sigma = L_s - \frac{L_m^2}{L_r}$ *is the transient leakage inductance.*

In this model, integration of the low frequency signals, dominance of stator IR drop at low speed and leakage inductance variation result in a less precise flux estimation. Integration at low frequency has been studied and there are three different alternatives [11]. Estimation of rotor flux from the terminal quantities depends on parameters such as stator resistance and leakage inductance.

The Current Model Flux Estimator (CMFE) is an alternative approach to overcome the problems caused by the changes in leakage inductance and stator resistance at low speed. The current model flux observer is considered to outperform the voltage model flux observer at low speeds. Also, its accuracy is relatively unaffected by the leakage inductance for any operating condition [13]. In this model flux can be estimated as:

$$\frac{d\varphi_{dr}^{s}}{dt} = -\frac{1}{T_r}\varphi_{dr}^{s} - w_r\varphi_{qr}^{s} + \frac{L_m}{T_r}i_{ds}^{s} \tag{2.35}$$

$$\frac{d\varphi_{qr}^{s}}{dt} = -\frac{1}{T_r}\varphi_{qr}^{s} - w_r\varphi_{dr}^{s} + \frac{L_m}{T_r}i_{qs}^{s} \tag{2.36}$$

However, CMFE does not work well at high speeds due to its sensitivity against the changes in the rotor resistance. Jansen did an extensive study on both VMFE and CMFE, based on direct field oriented control, discussed the design and accuracy assessment of various flux estimators, and compared and analyzed the alternative flux estimators [12]. To further improve the observer performance, closed loop rotor flux estimator are proposed which use the estimated stator current error or the estimated stator voltage error to estimate the rotor flux [12]-[13].

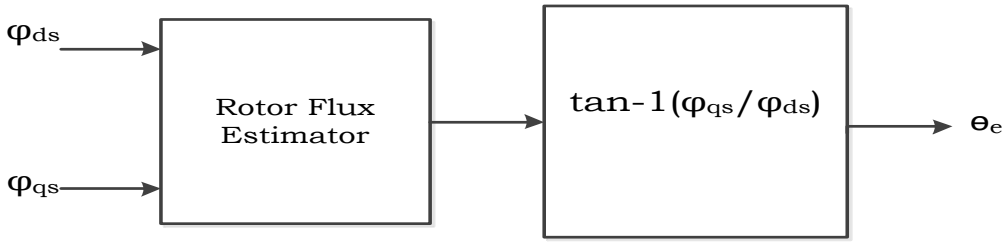


Figure 2.4: Direct field oriented drive system

2.7. Controller Design

2.7.1. PI Controller Background

The conventional proportional-integral controller remains the most popular design approach used in industrial applications due to its simplicity and reliability for the control of first and second order plants, and even high order plants with well-defined conditions. A well- tuned PI controller is capable in achieving an excellent performance [5-6]. However, it suffers a crucial disadvantage of getting a poor performance whenever the plant is subjected to some kind of disturbance or, the plant has high order nonlinear structure. Figure 2.5 shows the Simplified block diagram of the speed control of induction motor using a PI controller [21].

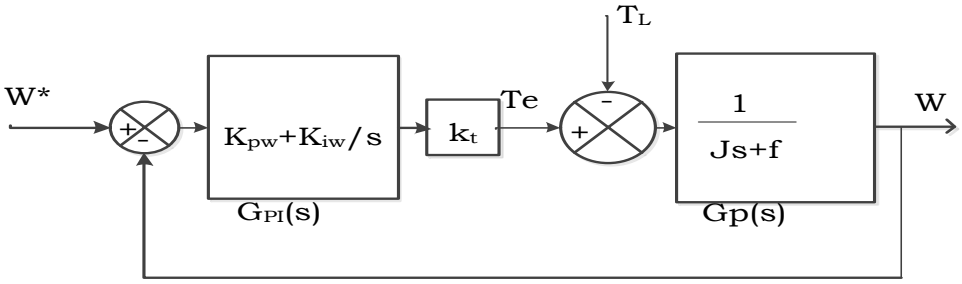


Figure 2.5: model of Speed PI Controller

Where w^* is the reference rotor angular speed, w is the rotor angular speed, $e = w^* - w$ is the tracking speed error, K_{pw} is the proportional gain of speed, K_{iw} is the integral gain, f is the total damping coefficient, T_L is

load torque,kt denotes torque constant, Te denotes the electromagnetic torque. The time rate of change of rotor mechanical speed in s-domain from the torque balance equation of (2.9) is:

$$\frac{dw}{dt}+\frac{fw}{J}=\frac{T_e-T_L}{J} \tag{2.37}$$

For perfect decoupling control, the total flux needs to be aligned with the d-axis so that the q-axis rotor flux is set to be zero and the d-axis rotor flux remains constant, then

$$\varphi_{qr}=0 \qquad \& \qquad \frac{d\varphi_{qr}}{dt}=0 \tag{2.38}$$

Substituting equation (2.38) into equation (2.8), the electromagnetic torque becomes,

$$T_e=\frac{3}{2}\frac{P}{2}\frac{L_m}{L_r}\,\varphi_{rd}i_{sq}=k_t i_{sq} \tag{2.39}$$

Where, $k_t=\frac{3}{2}\frac{P}{2}\frac{L_m}{L_r}\,\varphi_{rd}$

Now substituting equation (2.39) into (2.37) and making load torque zero, equation (2.37) becomes

$$J\frac{dw}{dt}+fw=k_t i_{sq} \tag{2.40}$$

From equation (2.40), the transfer function of the speed controller is given by

$$G_P(s)=\frac{w(s)}{i_{sq}(s)}=\frac{k_t}{Js+f} \tag{2.41}$$

It can be concluded from equation (2.41) that the controller transfer functions are all first order system and The parameters K_{pw} and K_{iw} of the continuous controller are obtained by the following steps.

Step 1: First calculate the open loop transfer function of the plant.

Step 2: Second derive the loop gain of the control system using pole-zero cancellation method or any convenient approach and

step 3**:** lastly obtain the controller parameters from the closed loop transfer function.

1. Open loop transfer function of the plant as can be seen in fig 5 for zero load torque is given by

$$G_0(s)=G_{PI}(s)\times G_P(s)=k_t\left(\frac{sK_{pw}+K_{iw}}{Js^s+sf}\right) \tag{2.42}$$

2. Now, from the closed loop transfer function of the overall system, we get,

$$G_c(s)=\frac{G_o(s)}{1+G_o(s)}=\frac{\frac{k_t}{J}\left(\frac{sK_{pw}+K_{iw}}{Js^2+sf}\right)}{s^2+\left(\frac{f+k_tK_{pw}}{J}\right)s+\frac{k_tK_{iw}}{J}} \tag{2.43}$$

Comparing the second order characteristic equation of the closed loop transfer function for speed with the standard second order characteristic equation $s^2 + 2\xi w_n s + w_n^2$, where w_n is undamped natural frequency and ξ is damping ratio , we can obtain the P-I speed controller gains such as:[12]

$$s^2 + \left(\frac{f + k_t K_{pw}}{J}\right)s + \frac{k_t K_{iw}}{J} = s^2 + 2\xi s + w_n^2$$

$$So, \quad \frac{k_t K_{iw}}{J} = w_n^2 \quad or \quad K_{iw} = \frac{J w_n^2}{k_t}$$

$$\text{Also,} \quad \frac{f + k_t K_{pw}}{J} = 2\xi w_n \quad or \quad K_{pw} = \frac{2J\xi w_{n-f}}{k_t}$$

The ζ value is set to 0.707 in this thesis. The value 0.707 generally results in a step response with fast settling time and reasonable overshoot. Increasing the value within the range from 0.707 and 1 will reduce the overshoot of the step response, but will increase the settling time.

The rotor speed, and current controller are each controlled by a separate PI module. $K_{pw}\left(1 + \frac{1}{T_i S}\right)$ is the PI controller; where Kpw-proportional gain, $\frac{K_{pw}}{T_i}$ is integral gain, Ti- integral action time. The discrete form of the PI controller becomes by backward difference formula:

$C(Z) = K_p + K_i x\left(\frac{T_s Z}{Z-1}\right)$ $\quad ; K_p = K_{pw}$, $K_{iw} = \frac{K_{pw}}{T_i} x Sampling\ time$ where $T_s - sampling\ time$

Current controller gain

As can be given on equation (2.21), the d-axis and q-axis current becomes:

$$\frac{di_{ds}}{dt} = -R_1 i_{ds} + w_e i_{qs} + R_2 \varphi_{dr} + \frac{v_{ds}}{\sigma L_s} \tag{2.44}$$

$$\frac{di_{qs}}{dt} = -R_1 i_{qs} - w_e i_{ds} + R_3 w_r \varphi_{dr} + \frac{v_{qs}}{\sigma L_s} \tag{2.45}$$

Where $w_e i_{qs} + R_2 \varphi_{dr}$ and $-w_e i_{ds} + R_3 w_r \varphi_{dr}$ are the d and q-axis decoupling components. For linearity these expressions are need to be eliminated. For perfect decoupling, φ_{qr}=0.

Also, the presence of the coupled terms makes the system nonlinear. So, to make the system linear we need to replace these nonlinear coupled terms by $v_{d_{decopule}}$ and $v_{q_{decopule}}$. So, the state-space equations of the induction motor drive for the electrical subsystems, from equation (2.44) and (2.45) can be represented as:

$$\frac{di_{ds}}{dt} = -R_1 i_{ds} + \frac{v_{ds}}{\sigma L_s} + v_{d_{decopule}}$$

$$\frac{di_{qs}}{dt} = -R_1 i_{qs} + \frac{v_{qs}}{\sigma L_s} + v_{d_{decopule}}$$

To achieve linear control of stator voltage it is necessary to remove the decoupling terms. These terms can be considered as disturbance and are cancelled by using a decoupled method that utilizes nonlinear feedback of the coupling voltage. The transfer function for the d-q current controllers of the vector controlled induction motor drives after removing the decouple term are given as follows.

$$G(s)_{d\_PI} = \frac{i_{ds}(s)}{v_{ds}(s)} = \frac{\frac{1}{\sigma L_s}}{s + R_1}$$

$$G(s)_{q\_PI} = \frac{i_{qs}(s)}{v_{qs}(s)} = \frac{\frac{1}{\sigma L_s}}{s + R_1}$$

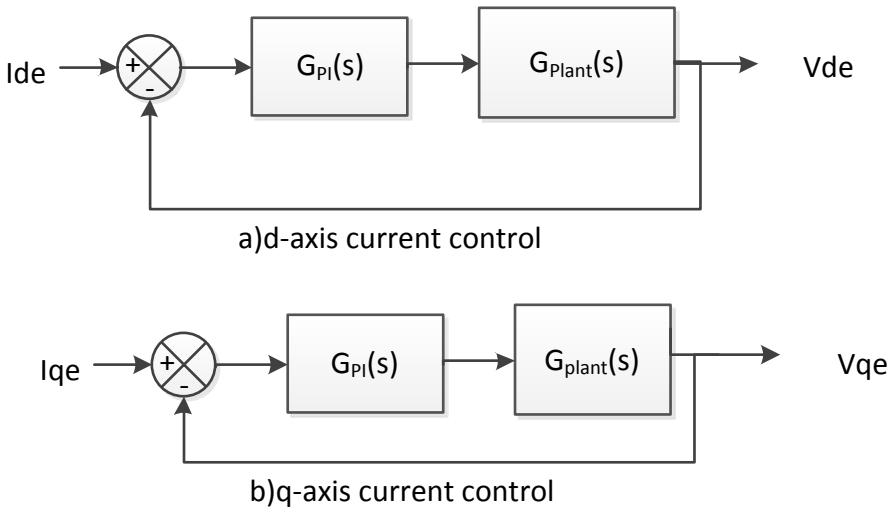


Figure 2.6: closed loop diagram of dq-axis current control

Where, $G_{PI}(s) = K_{pd} + \frac{K_{id}}{s}$ and $G_{Plant}(s) = \frac{\frac{1}{\sigma L_s}}{s + R_1}$

By using SISOTools methods of automatic tuning system, proportional and integral constants of speed and current controllers are determined and given in table 2.1

The MATLAB code need to generate current gains are written as shown below

```
%.... speed controller  by sisotool methods just to determine Kp and Ki...%
    num=[1/(a*Ls)];              % num=numerator
    den=[1 R1];                   % den=denominator
    current=tf(num,den)
    sisotool(current)
```

The root locus plot and parameter determination of the controller gain are shown below.

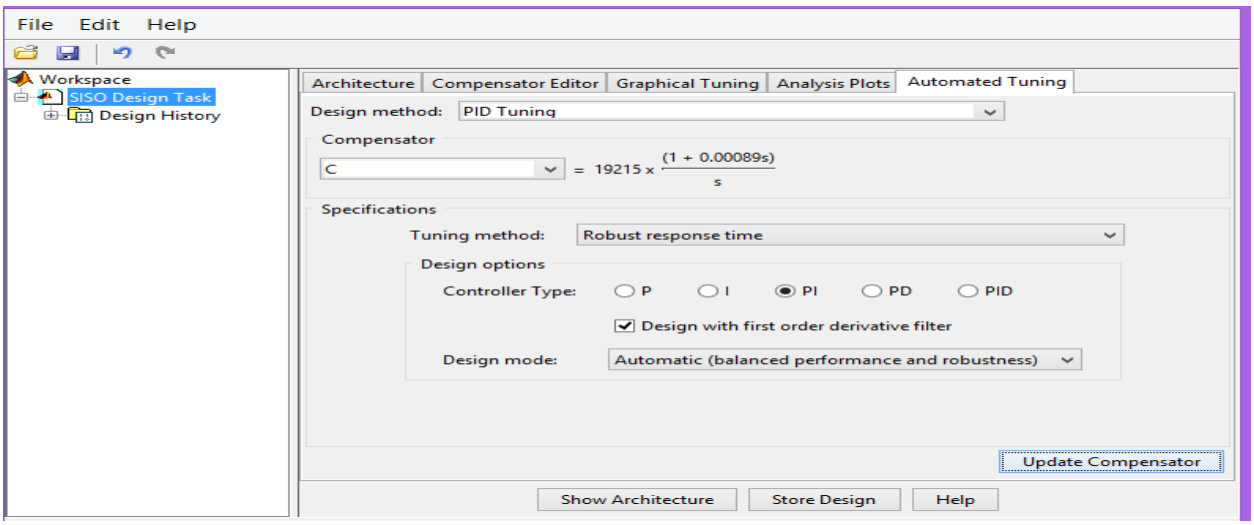
From the control and estimation parameter manager shown below

$$K_i = 19215 \qquad \text{and}$$

$$\frac{K_p}{K_i} = 0.00089 \rightarrow K_p = K_i * 0.00089$$

There fore $K_p = 19215 * 0.00089 = 17.10$

The speed Pi gains are designed with the same fashion as current gains.



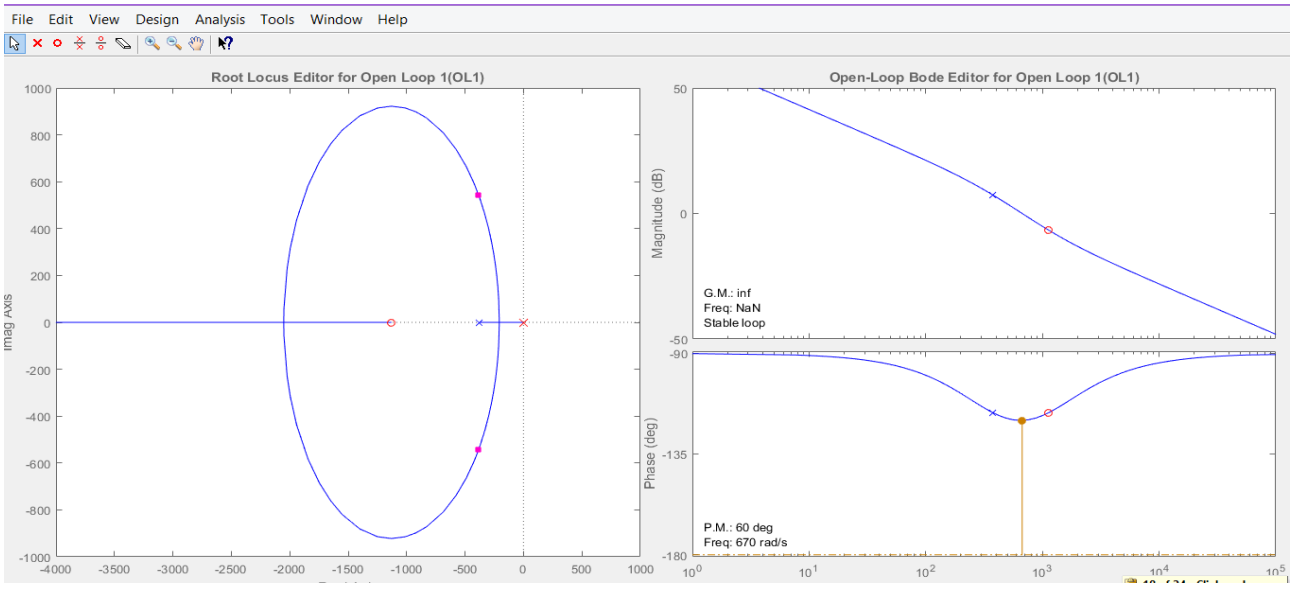


Table 2.1: Proportional (Kp) and Integral (Ki) Gains of PI Controller

PI controller	Kp	Ki
Speed control loop	0.04409216	0.009756
Inner d_e -q_e current loops	17.10	19215

2.8. Pulse Width Modulation with Space Vector Theory

Space Vector PWM: The space vector PWM (SVM) method is an advanced, computation-intensive PWM method and is possibly the best method among the all PWM techniques for variable-frequency drive application. Because of its superior performance characteristics, it has been finding wide spread application in recent years.

Space vector modulation (SVM) is an algorithm for the control of pulse width modulation (PWM)[1].It is used for the creation of alternating current (AC) waveforms; most commonly to drive 3 phase AC powered motors at varying speeds from DC using multiple class-D amplifiers. There are various variations of SVM that result in different quality and computational requirements. One active area of development is in the reduction of total harmonic distortion (THD) created by the rapid switching inherent to these algorithms.

To implement space vector modulation a reference signal V_{ref} is sampled with a frequency f_s ($T_s = 1/f_s$). The reference signal may be generated from three separate phase references using the transform. The reference vector is then synthesized using a combination of the two adjacent active switching vectors and one or both of the zero vectors. Various strategies of selecting the order of the vectors and which zero vector(s) to use exist. Strategy selection will affect the harmonic content and the switching losses.

2.8.1. Principle of Space Vector PWM

The circuit model of a typical three-phase voltage source PWM inverter is shown in Figure.2.7. S_1 to S_6 are the six power switches that shape the output, which are controlled by the switching variables $a, a^{'} b, b^{'}, c and c^{'}$. When an upper transistor is switched on, i.e., when a, b or c is 1, the corresponding lower transistor is switched off, i.e., the corresponding $a^{'}, b^{'}, or c^{'}$ is 0. Therefore, the on and off states of the upper transistors S_1, S_3 and S_5 can be used to determine the output voltage.

The relationship between the switching variable vector $[a \quad b \quad c]^t$ and the line-to-line voltage vector $[V_{ab} \quad V_{bc} \quad V_{ca}]^t$ is given by:

$$\begin{bmatrix} V_{ab} \\ V_{bc} \\ V_{ca} \end{bmatrix} = V_{dc} \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad (2.46)$$

Also, the relationship between the switching variable vector [a, b, c]t and the phase voltage vector $[V_{ab} \quad V_{bc} \quad V_{ca}]^t$ can be expressed below.

$$\begin{bmatrix} V_{an} \\ V_{bn} \\ V_{cn} \end{bmatrix} = \frac{V_{dc}}{3} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad (2.47)$$

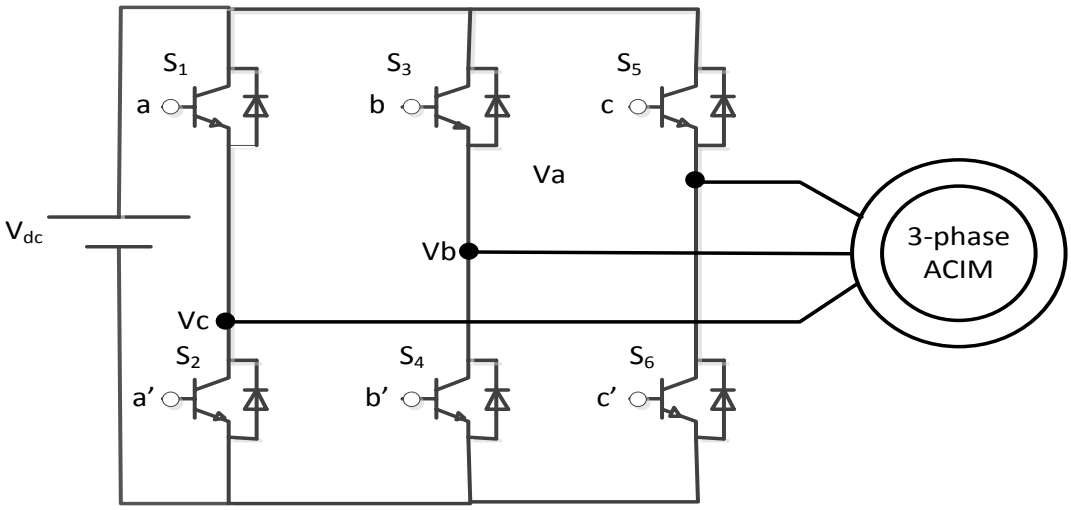


Figure 2.7: Three-phase voltage source PWM Inverter

As illustrated in Fig. 2.8, there are eight possible combinations of on and off patterns for the three upper power switches. The on and off states of the lower power devices are opposite to the upper one and so are easily determined once the states of the upper power transistors are determined. According to equations (2.45) and (2.46), the eight switching vectors, output line to neutral voltage (phase voltage), and output line-to-line voltages in terms of DC link Vdc, are given in Table1 and Figure. 2.8 shows the eight inverter voltage vectors (V_0 to V_7)

Table 2.2: Switching vectors, phase voltages and output line to line voltages

Voltage Vector	Switching Vectors			Line to neutral voltage			Line to line voltage		
	S_1	S_3	S_5	V_{an}	V_{bn}	V_{cn}	V_{ab}	V_{bc}	V_{ca}
V_0	0	0	0	0	0	0	0	0	0
V_1	1	0	0	2/3	-1/3	-1/3	1	0	-1
V_2	1	1	0	1/3	1/3	-2/3	0	1	-1
V_3	0	1	0	-1/3	2/3	-1/3	-1	1	0
V_4	0	1	1	-2/3	1/3	1/3	-1	0	1
V_5	0	0	1	-1/3	-1/3	2/3	0	-1	1
V_6	1	0	1	1/3	-2/3	1/3	1	-1	0
V_7	1	1	1	0	0	0	0	0	0

(Note that the respective voltage should be multiplied by V_{dc})

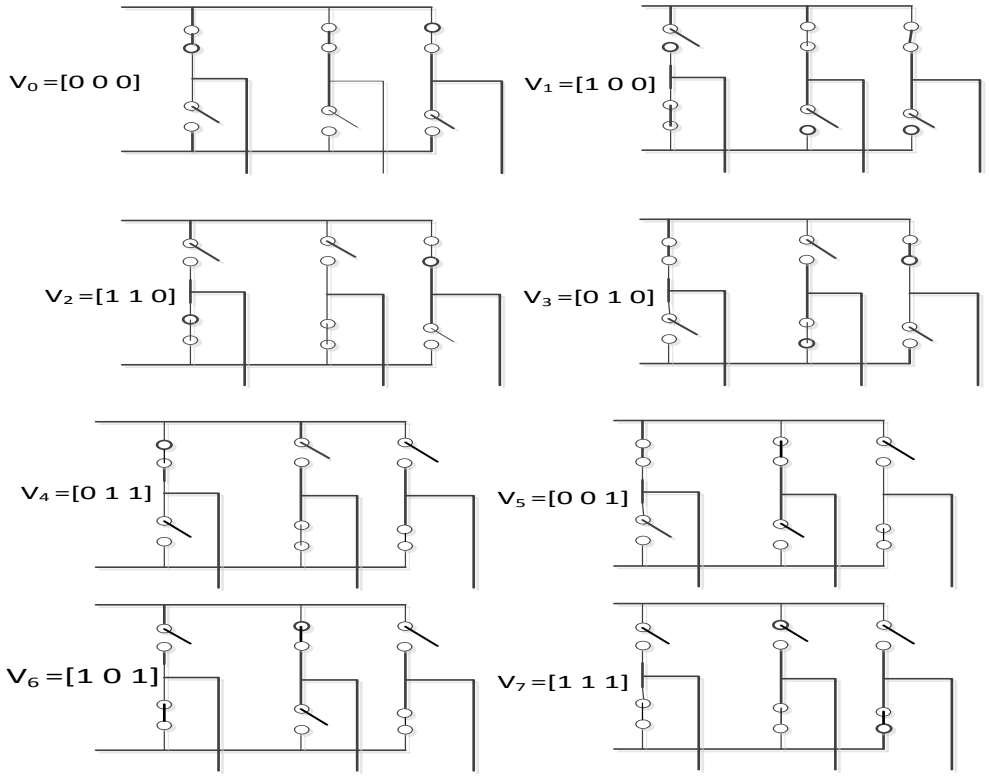


Figure 2.8: The eight inverter voltage vectors (V0 to V7)

Space Vector PWM (SVPWM) refers to a special switching sequence of the upper three power transistors of a three-phase power inverter. It has generated less harmonic distortion in the output voltages and or currents applied to the phases of an AC motor and provides more efficient use of supply voltage compared with sinusoidal modulation technique as shown in Figure 2.9

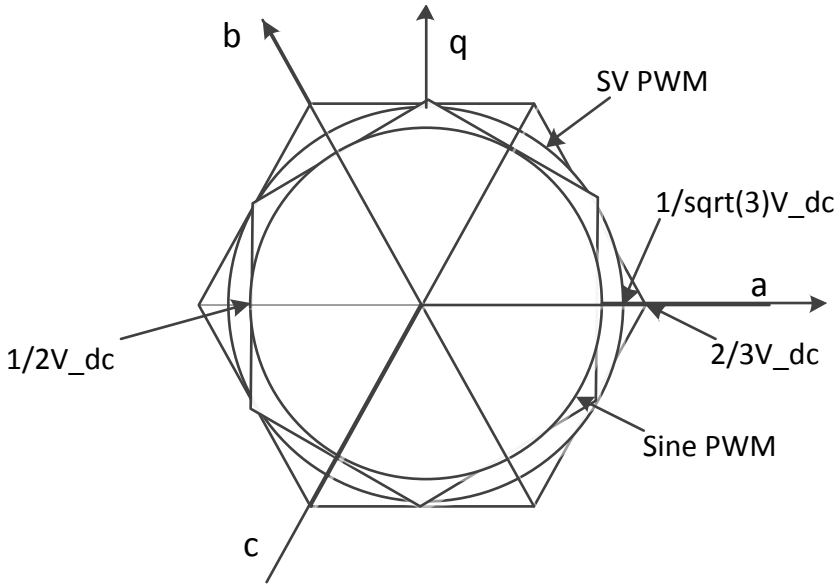


Figure 2.9: Locus comparison of maximum linear control voltage in Sine PWM and SVPWM [29]

To implement the space vector PWM, the voltage equations in the *abc* reference frame can be transformed into the stationary *dq* reference frame that consists of the horizontal (d) and vertical (q) axes as depicted in Figure 2.10

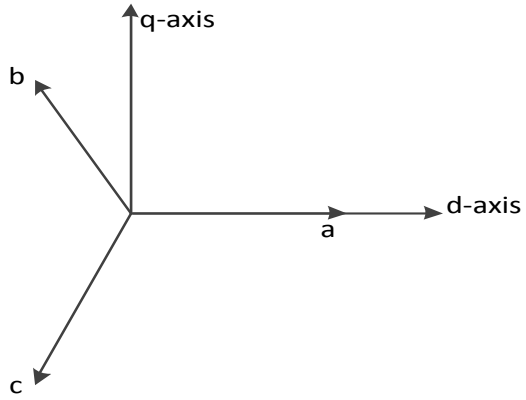


Figure 2.10: The relationship of abc reference frame and stationary dq reference frame.

From this figure, the relation between these two reference frames is shown below:

$$f_{dqo} = K f_{abc}$$

Where $K=\frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$ $f_{dqo} = [f_d \quad f_q \quad f_0]^t$ and $f_{abc} = [f_a \quad f_b \quad f_c]^t$

and f denotes either a voltage or a current variable.

As described in Figure. 2.10, this transformation is equivalent to an orthogonal projection of $[a \quad b \quad c]^t$ onto the two-dimensional perpendicular to the vector $[1 \quad 1 \quad 1]^t$ (the equivalent d-q plane) in a three-dimensional coordinate system. As a result, six non-zero vectors and two zero vectors are possible. Six nonzero vectors (V_1 - V_6) shape the axes of a hexagonal as depicted in Figure 2.11 and feed electric power to the load. The angle between any adjacent two nonzero vectors is 60 degrees. Meanwhile, two zero vectors (V_0 and V_7) are at the origin and apply zero voltage to the load. The eight vectors are called the basic space vectors and are denoted by V_0, V_1, V_2, V_3, V_4, V_5, V_6, and V_7. The same transformation can be applied to the desired output voltage to get the desired reference voltage vector V_{ref} in the d-q plane. The objective of space vector PWM technique is to approximate the reference voltage vector V_{ref} using the eight switching patterns. One simple method of approximation is to generate the average output of the inverter in a small period, T to be the same as that of V_{ref} in the same period.

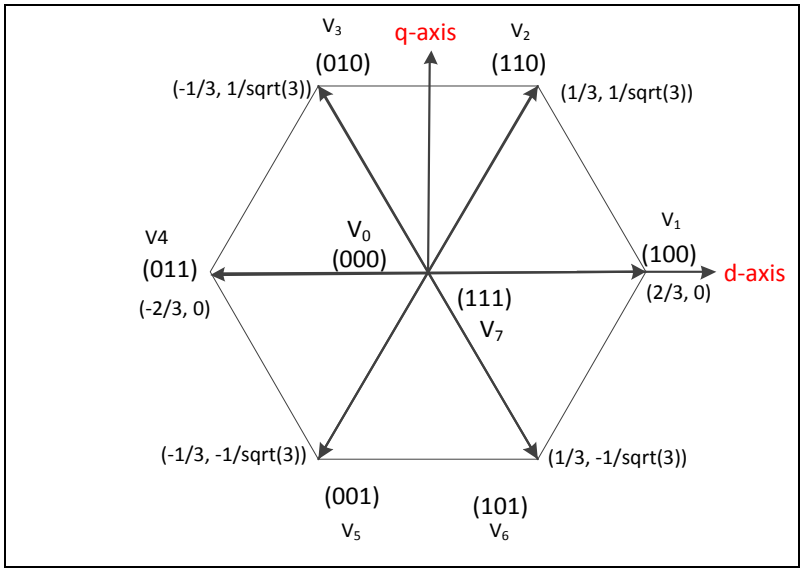


Figure 2.11: Basic switching vectors and sectors.

Therefore, space vector PWM can be implemented by the following steps:

- Step 1. Determine V_d, V_q, V_{ref}, and angle (α)
- Step 2. Determine time duration T_1, T_2, T_0
- Step 3. Sector Determination
- Step 4. Determine the switching time of each transistor (S_1 to S_6)

Step 1: Determine V_d, V_q, V_{ref}, and angle (α)

From Fig 12. Vd, Vq, Vref, and angle (α) can be determined as follows:

$$V_d = V_{an} - V_{bn}\cos(60) - V_{cn}\cos(30) = V_{an} - \frac{V_{bn}}{2} - \frac{V_{cn}}{2}$$

$$V_q = 0 + V_{bn}\cos(30) - V_{cn}\cos(30) = \frac{\sqrt{3}}{2}V_{bn} - \frac{\sqrt{3}}{2}V_{cn}$$

$$V_{ref} = \sqrt{{V_d}^2 + {V_q}^2}$$

$$\alpha = arctan\left(\frac{V_q}{V_d}\right) = \omega t = 2\pi f t$$

Where f=fundamental frequency

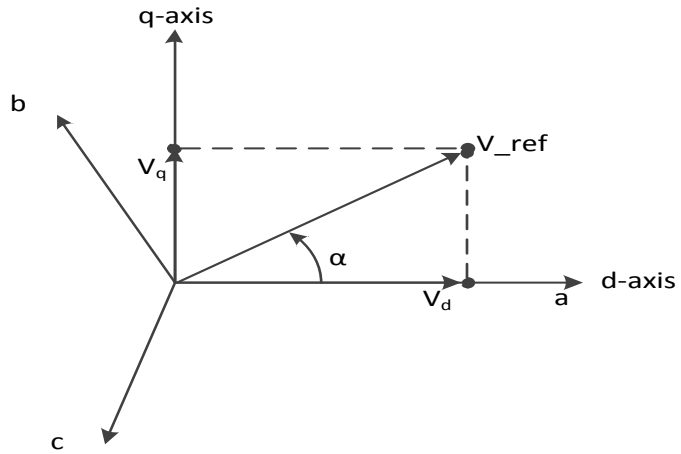


Figure 2.12: Voltage Space Vector and its components in (D, Q).

Step 2: Determine time duration T_1, T_2, T_0

From Fig 12. The switching time duration can be calculated as follows:

- Switching time duration at Sector 1

$$\int_0^{T_z} \overline{V}_{ref}\,dt = \int_0^{T_1} \overline{V}_1\,dt + \int_{T_1}^{T_1+T_2} \overline{V}_2\,dt + \int_{T_1+T_2}^{T_z} \overline{V}_0\,dt$$

$$\therefore\ T_z.\overline{V}_{ref} = \left(T_1.\overline{V}_1 + T_2.\overline{V}_2.\right)$$

$$\Rightarrow T_z \cdot \left|\overline{V}_{ref}\right| \cdot \begin{bmatrix}\cos(\alpha)\\ \sin(\alpha)\end{bmatrix} = T_1.\frac{2}{3}.V_{dc}.\begin{bmatrix}1\\0\end{bmatrix} + T_2.\frac{2}{3}.V_{dc}.\begin{bmatrix}\cos(60)\\ \sin(60)\end{bmatrix}$$

(Where $0 \le \alpha \le 60^0$)

$$\therefore\ T_1 = T_z.a.\frac{\sin(\frac{\pi}{3}-\alpha)}{\sin(\frac{\pi}{3})} \qquad \textbf{and} \qquad T_0 = T_z - (T_1 + T_2), \quad \left\{ where \quad T_z = \frac{1}{f_z}\ and\ a = \frac{\left|\overline{V}_{ref}\right|}{\frac{2}{3}V_{dc}} \right\}$$

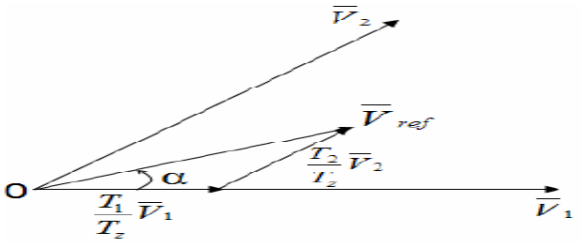


Figure 2.13: Reference vector as a combination of adjacent vectors at sector 1.

- Switching time duration at any Sector:

$$T_1 = \frac{\sqrt{3}.T_z.\left|\overline{V}_{ref}\right|}{V_{dc}}\left(\sin\left(\frac{\pi}{3}-\alpha+\frac{n-1}{3}\pi\right)\right)$$

$$= \frac{\sqrt{3}.T_z.\left|\overline{V}_{ref}\right|}{V_{dc}}\left(\sin\left(n\frac{\pi}{3}-\alpha\right)\right)$$

$$= \frac{\sqrt{3}.T_z.\left|\overline{V}_{ref}\right|}{V_{dc}}\left(\sin(\frac{n}{3}\pi)\cos(\alpha)-\cos(\frac{n}{3}\pi)\sin(\alpha)\right)$$

$$\therefore T_2 = \frac{\sqrt{3}.T_z.\left|\overline{V}_{ref}\right|}{V_{dc}}\left(\sin\left(\alpha-\frac{n-1}{3}\pi\right)\right)$$

$$= \frac{\sqrt{3}.T_z.\left|\overline{V}_{ref}\right|}{V_{dc}}\left(\sin(\alpha)\cos(\frac{n-1}{3}\pi)-\cos(\alpha)\sin(\frac{n-1}{3}\pi)\right)$$

$$T_0 = T_z - (T_1 + T_2),$$

Step 3: Sector Determination $\{ wheren = 1 through\ 6\ (that is, sector 1 to 6) \quad 0 \le \alpha \le 60\}$

It is necessary to know in which sector the reference output lies in order to determine the switching time and sequence. The phase voltages correspond to eight switching states, six non-zero vectors and two zero vectors at the origin. Depending on the reference voltage $\vec{v}_d$, and $\vec{v}_q$, the angle of the reference vector can be used to determine the sector as shown in Table 1.3

Table 1.3: Sector identification

Sector	Degrees
1	$0 < \theta \le 60°$
2	$60° < \theta \le 120°$
3	$120° < \theta \le 180°$
4	$180° < \theta \le 240°$
5	$240° < \theta \le 300°$
6	$300° < \theta \le 360$

The duty cycle computation is done for each triangular sector formed by two state vectors. The magnitude of each switching state vector is $2v_{\_dc}/\sqrt{3}$ and the magnitude of a vector to the midpoint of the hexagon line from one vertex to another is $v_{dc}/\sqrt{3}$. The reference space vector rotates and moves through different sectors of the complex plane as time increases. In each PWM cycle, the reference vector $\vec{v}_{ref}$ is sampled at a fixed input sampling frequency(f_s). During this time, the sector is determined and the modulation vector $\vec{v}_{ref}$ is mapped onto two adjacent vectors. The non-zero vectors are written as:

$$\vec{v}_m = \frac{2}{3} v_{dc} e^{j(m-1)\frac{\pi}{3}}$$

Where m=1,2,3,4,5,6

Therefore the non-zero vector for $\vec{v}_m$ and $\vec{v}_{m+1}$ become

$$\vec{v}_m = \frac{2}{3} v_{dc} \left[\cos(m-1)\frac{\pi}{3} + j \sin(m-1)\frac{\pi}{3}\right]$$

$$\vec{v}_{m+1} = \frac{2}{3} v_{dc} \left[\cos\left(m\frac{\pi}{3}\right) + j \sin\left(m\frac{\pi}{3}\right)\right]$$

Due to symmetry in the patterns in the six sectors, the integration can be carried out only on half pulse width modulation $T_S/2$. Zero voltages are applied during null state times.

Thus the product of the reference voltage vector $\vec{v}_{ref}$ and $T_S/2$ are the sum of the voltage multiplied by the time interval of the chosen space vector. The reference voltage vector $\vec{v}_{ref}$ can be represented as function of $\vec{v}_m$and $\vec{v}_{m+1}$ as follows.

$$\vec{v}_{ref}\ \frac{T_s}{2} = \vec{v}_m T_a + \vec{v}_{m+1}\, T_b \qquad (2.45)$$

$$\vec{v}_{ref} = \vec{v}_d + \mathrm{j}\vec{v}_q$$

Where T_a and T_b denote the required on time of the active state vectors $\vec{v}_m$ and $\vec{v}_{m+1}$ during each sample period, and k is the sector number denoting the reference location. The calculated times T_a and T_b are applied to the switches to produce space vector PWM switching patterns based on each sector. The switching time is arranged according to the first half of the switching period while the half is a reflection forming asymmetrical pattern. If $\vec{v}_{ref}$ lies exactly in the middle between two vectors.

Assuming that the reference voltage and the voltage vector $\vec{v}_m$ and $\vec{v}_{m+1}$ are constant during each pulse width modulation period T_s and splitting the reference voltage $\vec{v}_{ref}$ into its real and imaginary components gives the following result.

Table 4 shows that the pulse patterns generated by space vector PWM in the given sector.

Table 2.4: Duty cycle calculation

Sector	θ	T_a	T_b
1	$0 < \theta \leq 60^0$	$\frac{3v_\alpha}{4} - \sqrt{3}\frac{v_\beta}{4}$	$\sqrt{3}\frac{v_\beta}{2}$
2	$60 < \theta \leq 120^0$	$\frac{3v_\alpha}{4} + \sqrt{3}\frac{v_\beta}{4}$	$\frac{-3v_\alpha}{4} + \sqrt{3}\frac{v_\beta}{4}$
3	$120 < \theta \leq 180^0$	$\sqrt{3}\frac{v_\beta}{2}$	$\frac{-3v_\alpha}{4} - \sqrt{3}\frac{v_\beta}{4}$
4	$180 < \theta \leq 240^0$	$\frac{-3v_\alpha}{4} + \sqrt{3}\frac{v_\beta}{4}$	$-\sqrt{3}\frac{v_\beta}{2}$
5	$240 < \theta \leq 300^0$	$\frac{-3v_\alpha}{4} - \sqrt{3}\frac{v_\beta}{4}$	$\frac{3v_\alpha}{4} - \sqrt{3}\frac{v_\beta}{4}$
6	$300 < \theta \leq 360^0$	$-\sqrt{3}\frac{v_\beta}{2}$	$\frac{3v_\alpha}{4} + \sqrt{3}\frac{v_\beta}{4}$

Step 4. Determine the switching time of each transistor (S_1 to S_6)

It is necessary to arrange the switching sequence so that the switching frequency of each inverter leg is minimized. There are many switching patterns that can be used to implement SVPWM. To minimize the switching losses, only two adjacent active vectors and two zero vectors are used in a sector [8] [13]. To meet this optimal condition, each switching period starts with one zero vectors and end with another zero vector

during the sampling time. This rule applies normally to three phase inverters as a switching sequence. Therefore, the switching cycle of the output voltage is double the sampling time and the two output voltage waveforms become symmetrical. Table 2.3 shows a symmetric switching sequence. Referring to this table, the binary representations of two adjacent basic vectors differ in only one bit, so that only one of the upper transistors switches is closed when the switching pattern moves from one vector to an adjacent one.

The two vectors are time weighted in a sample period T_s to produce the desired output voltage.

Table 2.5: Seven Segment Switching Sequence.

Space vector	Switching state	Off state switch	On state switch	Vector definition
$\vec{v}_0$	000	S_1, S_3, S_5	S_4, S_6, S_2	$\vec{v}_0 = 0$
$\vec{v}_1$	100	S_4, S_3, S_5	S_1, S_6, S_2	$\vec{v}_1 = \frac{2}{3}V_{-dc}$
$\vec{v}_2$	110	S_4, S_6, S_5	S_1, S_3, S_2	$\vec{v}_2 = \frac{2}{3}V_{\_dc}e^{j\frac{\pi}{3}}$
$\vec{v}_3$	010	S_1, S_6, S_5	S_4, S_3, S_2	$\vec{v}_3 = \frac{2}{3}V_{-dc}e^{j2\frac{\pi}{3}}$
$\vec{v}_4$	011	S_1, S_6, S_2	S_4, S_3, S_5	$\vec{v}_4 = \frac{2}{3}V_{\_dc}e^{j\frac{3\pi}{3}}$
$\vec{v}_5$	001	S_1, S_3, S_2	S_4, S_6, S_5	$\vec{v}_5 = \frac{2}{3}V_{\_dc}e^{j\frac{4\pi}{3}}$
$\vec{v}_6$	101	S_4, S_3, S_2	S_1, S_6, S_5	$\vec{v}_6 = \frac{2}{3}V_{\_dc}e^{j\frac{5\pi}{3}}$
$\vec{v}_7$	111	S_4, S_6, S_2	S_1, S_3, S_5	$\vec{v}_7 = \frac{2}{3}V_{\_dc}$

Chapter three

Flux and Speed Estimation for Sensor less DFOC of IM

3.1. Introduction

In this chapter, Estimators configured for direct field-orientation (DFO) are investigated. The field orientation is catagorized in two ways as Direct Field Orientation and Indirect Field Orientation. The basic difference of these methods underlies in the manner of detecting the synchronous speed. In IFO, the slip angle is computed and added to the rotor speed to find the synchronous speed. One must calculate, therefore, the slip-angle and estimate the rotor angle. On the other hand, in DFO, the synchronous speed is computed from the ratio of flux of dq-axes.

3.2. Flux estimator of the three-phase induction motor

3.2.1. Estimation of the Flux Linkage Vector

Most of the sensor less control schemes rely directly or indirectly on the estimation of the stator flux linkage vector, ys being defined as the time integral of the induced voltage,

$$\frac{d\varphi^{s,v}}{dt} = u_s - R_s i_s + u_{off}, \quad \varphi_s(0) = \varphi_{so} \tag{3.1}$$

Where, u_{off} represents all disturbances such as offsets, unbalances and other errors present in the estimated induced emf. A major source of error in the emf is due to the changes in the model parameter R_s. The estimation of the flux vectors requires the integration of (3.1) in real-time. The integrator, however, will have an infinite gain at zero frequency, and the unavoidable offsets contained in the integrator input then make its output gradually drift away beyond limits.

3.2.2. Flux Estimation in Continuous Time

Flux estimator used in this chapter can compute both the synchronous speed and the rotor speed. The logic underlying this flux estimator is basically an advanced voltage model approach in which integration of the back-emf is calculated and compensated for the errors associated with pure integrator and stator resistance R_s measurement at low speeds. At high speeds, the voltage model provides an accurate stator flux estimate because the machine back emf dominates the measured terminal voltage. However, at low speeds, the stator IR drop becomes significant, causing the accuracy of the flux estimate to be sensitive to the estimated stator resistance. Due to this effect, at low excitation frequencies flux estimation based upon voltage model are generally not capable of achieving high dynamic performance at low speeds [12]. Consequences of these problems are compensated with the addition of a closed-loop in the flux observer. Basically, the fluxes obtained by current

model are compared with those obtained by the voltage model with reference to the current model, or the current model with reference to the voltage model according to the range in which one of these models is superior to other [23]. In this flux observer the voltage model is corrected by the current model through a basic PI block. In the end, the stator fluxes are used to obtain rotor fluxes and rotor flux angle.

The rotor flux linkage dynamics in synchronously rotating reference frame $(W = W_e = W_{\varphi r})$ being as [23];

$$\frac{d\varphi_{dr}^{e,i}}{dt} = \frac{L_m}{T_r} i_{ds}^{e} - \frac{1}{T_r}\varphi_{dr}^{e,i} + (w_e - w_r)\varphi_{qr}^{e,i} \tag{3.2}$$

$$\frac{d\varphi_{qr}^{e,i}}{dt} = \frac{L_m}{T_r} i_{qs}^{e} - \frac{1}{T_r}\varphi_{qr}^{e,i} + (w_e - w_r)\varphi_{dr}^{e,i} \tag{3.3}$$

Where L_m is the magnetizing inductance (H), $T_r = \frac{L_r}{R_r}$ is the rotor time-constant (sec), and w_r is the electrical angular velocity of the rotor (rad/sec). In the current model, the total rotor flux-linkage is aligned with the d-axis component, and hence;

$$\varphi_r^{e,i} = \varphi_{dr}^{e,i} \qquad and \qquad \varphi_{qr}^{e,i} = 0$$

Substitution of $\varphi_{qr}^{e,i} = 0$ into (3.2) and (3.3) yields the oriented rotor flux dynamics as;

$$\frac{d\varphi_{dr}^{e,i}}{dt} = \frac{L_m}{T_r} i_{ds}^{e} - \frac{1}{T_r}\varphi_{dr}^{e,i} \tag{3.4}$$

$$\varphi_{qr}^{e,i} = 0 \tag{3.5}$$

Note that (3.4) and (3.5) are the commonly recognized forms of the rotor flux vector equations. When, the rotor flux linkages in (3.4) and (3.5) undergoes the inverse park transformation in the stationary reference frame the result becomes[23].

$$\varphi_{dr}^{s,i} = \varphi_{dr}^{e,i}\, cos(\theta_{\varphi r}) - \varphi_{qr}^{e,i}\, sin(\theta_{\varphi r}) = \varphi_{dr}^{e,i}\, cos(\theta_{\varphi r}) \tag{3.6}$$

$$\varphi_{qr}^{s,i} = \varphi_{qr}^{e,i}\, cos(\theta_{\varphi r}) + \varphi_{dr}^{e,i}\, sin(\theta_{\varphi r}) = \varphi_{dr}^{e,i}\, sin(\theta_{\varphi r}) \tag{3.7}$$

Where $\theta_{\varphi r}$ is the rotor flux angle (rad). The stator flux linkages in stationary reference frame are then computed using (3.8) and (3.9) as;

$$\varphi_{ds}^{s,i} = L_s i_{ds}^{s} + L_m i_{dr}^{s} \tag{3.8}$$

$$\varphi_{qs}^{s,i} = L_s i_{qs}^{s} + L_m i_{qr}^{s} \tag{3.9}$$

$$\varphi_{dr}^{s,i} = L_r i_{dr}^{s} + L_m i_{ds}^{s} \tag{3.8a}$$

$$\varphi_{qr}^{s,i} = L_r i_{qr}^{s} + L_m i_{qs}^{s} \tag{3.9a}$$

And from equation (2.14)

$$i_{dr}^{s} = \frac{\varphi_{dr}^{s,i} - L_m i_{ds}^{s}}{L_r} \qquad \& \qquad i_{qr}^{s} = \frac{\varphi_{qr}^{s,i} - L_m i_{qs}^{s}}{L_r} \tag{3.10}$$

By substituting equation (3.10) into (3.8 & 3.9), the stator flux linkage in stationary reference frame becomes;

$$\varphi_{ds}^{s,i} = L_s i_{ds}^{s} + L_m i_{dr}^{s} = \left(\frac{L_s L_r - L_m^2}{L_r}\right) i_{ds}^{s} + \frac{L_m}{L_r}\varphi_{dr}^{s,i} \tag{3.11}$$

$$\varphi_{qs}^{s,i} = L_s i_{qs}^{s} + L_m i_{qr}^{s} = \left(\frac{L_s L_r - L_m^2}{L_r}\right) i_{qs}^{s} + \frac{L_m}{L_r}\varphi_{qr}^{s,i} \tag{3.12}$$

Where L_s and L_r are the stator and rotor self-inductance (H), respectively. The stator flux linkages in the voltage model, however, are computed by integrating the back emf's and compensated voltages taken into account.

$$\varphi_{ds}^{s,v} = \int \left(u_{ds-}^{s} i_{ds}^{s} R_s - u_{comp,ds}\right) dt \tag{3.13}$$

$$\varphi_{qs}^{s,v} = \int \left(u_{qs-}^{s} i_{qs}^{s} R_s - u_{comp,qs}\right) dt \tag{3.14}$$

The compensated voltages, on the other hand, are computed by the PI control law as follows [23]:

$$u_{comp,ds} = K_p\left(\varphi_{ds}^{s,v} - \varphi_{ds}^{s,i}\right) + \frac{K_p}{T_i}\int \left(\varphi_{ds}^{s,v} - \varphi_{ds}^{s,i}\right) dt \tag{3.15}$$

$$u_{comp,qs} = K_p\left(\varphi_{qs}^{s,v} - \varphi_{qs}^{s,i}\right) + \frac{K_p}{T_i}\int \left(\varphi_{qs}^{s,v} - \varphi_{qs}^{s,i}\right) dt \tag{3.16}$$

The proportional gain K_p and the reset time T_i are chosen such that the flux linkages computed by the current model becomes dominant at low speed. The reason for that is the back emfs computed by the voltage model result to be extremely low at this speed range (even zero for back emfs at zero speed). While the motor is running at high speed range, the flux linkages computed by voltage model becomes dominant over the flux linkage components computed through the current model. Once the stator flux linkages in (3.13) and (3.14) are calculated, the rotor flux linkages based on the voltage model are computed once more through (3.17) and (3.18) which are only rearranged forms of (3.11) and (3.12), as In current model, rotor flux linkage from equation (3.11) and (3.12) rearranged as:

$$\varphi_{dr}^{s,i} = -\left(\frac{L_s L_r - L_m^2}{L_m}\right) i_{ds}^{s} + \frac{L_r}{L_m}\varphi_{ds}^{s,i} \qquad and$$

$$\varphi_{qr}^{s,i} = -\left(\frac{L_s L_r - L_m^2}{L_m}\right) i_{qs}^{s} + \frac{L_r}{L_m}\varphi_{qs}^{s,i}$$

Therefore, the rotor flux linkage in voltage model is written in similar way;

$$\varphi_{dr}^{s,v} = -\left(\frac{L_s L_r - L_m^2}{L_m}\right) i_{ds}^{s} + \frac{L_r}{L_m}\varphi_{ds}^{s,v} \tag{3.17}$$

$$\varphi_{qr}^{s,v} = -\left(\frac{L_s L_r - L_m^2}{L_m}\right) i_{qs}^{s} + \frac{L_r}{L_m}\varphi_{qs}^{s,v} \tag{3.18}$$

It is then a straight process to compute the rotor flux angle based on the voltage model as;

$$\theta_{\varphi r} = tan^{-1}\left(\frac{\varphi_{dr}^{s,v}}{\varphi_{qr}^{s,v}}\right) \tag{3.19}$$

3.2.3. Flux Estimation in Discrete Time

It is clear that the DSP from its name is in discretized form, so that each continuous form of flux estimator as well as speed estimator should be discretized.

The oriented rotor flux dynamics in (3.4) is discretized by using backward approximation as follows [23]:

$$\frac{\varphi_{dr}^{e,i}(k) - \varphi_{dr}^{e,i}(k-1)}{T} = \frac{L_m}{T_r} i_{ds}^{e}(k) - \frac{1}{T_r}\varphi_{dr}^{e,i}(k) \tag{3.20}$$

Where T being the sampling period (sec). When rearranged (3.20) gives

$$\varphi_{dr}^{e,i}(k) = \frac{T_r}{T + T_r}\varphi_{dr}^{e,i}(k-1) - \frac{L_m x T}{T + T_r} i_{ds}^{e}(k) \tag{3.21}$$

The stator flux linkages in (3.13) and (3.14) are discretized by using trapezoidal approximation as;

$$\varphi_{ds}^{s,v}(k) = \varphi_{ds}^{s,v}(k-1) + \frac{T}{2}\left(e_{ds}^{s}(k) + e_{ds}^{s}(k-1)\right) \tag{3.22}$$

$$\varphi_{qs}^{s,v}(k) = \varphi_{qs}^{s,v}(k-1) + \frac{T}{2}\left(e_{qs}^{s}(k) + e_{qs}^{s}(k-1)\right) \tag{3.23}$$

Where the back emf's are computed as;

$$e_{ds}^{s}(k) = u_{ds}^{s}(k) - i_{ds}^{e}(k)R_s - u_{comp,ds}^{s}(k) \tag{3.24}$$

$$e_{qs}^{s}(k) = u_{qs}^{s}(k) - i_{qs}^{e}(k)R_s - u_{comp,qs}^{s}(k) \tag{3.25}$$

Similarly, the PI control laws in (3.15) and (3.16) are also discretized by using trapezoidal approximation as;

$$u_{comp,ds}^{s}(k) = K_p\left(\varphi_{ds}^{s,v}(k) - \varphi_{ds}^{s,i}(k)\right) + u_{comp,ds,i}^{s}(k-1) \tag{3.26}$$

$$u_{comp,qs}^{s}(k) = K_p\left(\varphi_{qs}^{s,v}(k) - \varphi_{qs}^{s,i}(k)\right) + u_{comp,qs,i}^{s}(k-1) \tag{3.27}$$

Where the accumulating integral terms are;

$$u^s_{comp,ds,i}(k) = u^s_{comp,ds,i}(k-1) + \frac{K_p T}{T_i}\left(\varphi^{s,v}_{ds}(k) - \varphi^{s,i}_{ds}(k)\right)$$

$$= u^s_{comp,ds,i}(k-1) + K_p K_I\left(\varphi^{s,v}_{ds}(k) - \varphi^{s,i}_{ds}(k)\right)$$

$$u^s_{comp,qs,i}(k) = u^s_{comp,qs,i}(k-1) + \frac{K_p T}{T_i}\left(\varphi^{s,v}_{qs}(k) - \varphi^{s,i}_{qs}(k)\right)$$

$$= u^s_{comp,qs,i}(k-1) + K_p K_I\left(\varphi^{s,v}_{qs}(k) - \varphi^{s,i}_{qs}(k)\right)$$

Where $K_I = \frac{T}{T_I}$

3.2.4. Flux Estimation in Discrete Time and Per-Unit

All equations are needed to be normalized into per-unit by the specified base quantities. Firstly, the rotor flux linkage in current model (3.21) is normalized by dividing the base flux linkage (φ_{base}) as follow [23];

$$\varphi^{e,i}_{dr,pu}(k) = \frac{T_r}{T+T_r}\varphi^{e,i}_{dr,pu}(k-1) - \frac{T}{T+T_r} i^e_{ds},pu\ (k) \qquad (3.28)$$

$$\left\{Where\ ;\ \frac{\varphi^{e,i}_{ds}}{\varphi_{base}} = \varphi^{e,i}_{ds,pu}\ ,\qquad \frac{i^{e,i}_{ds}}{i_{base}} = i^{e,i}_{ds,pu}\ \ \&\ \ i_{base} = \frac{\varphi_{base}}{L_m}\right\}$$

Where $\varphi_{base} = L_m I_{base}$ is the base flux linkage (volt.sec) and I_{base} is the base current (amp).Next, the stator flux linkages in the current model (3.11) and (3.12) are similarly normalized by dividing the base flux linkage as

$$\varphi^{s,i}_{ds,pu}(k) = \left(\frac{L_s L_r - L_m^2}{L_r L_s}\right) i^s_{ds,pu}(k) + \frac{L_m}{L_r}\varphi^{s,i}_{dr,pu}\ (k) \qquad (3.29)$$

$$\varphi^{s,i}_{qs,pu}(k) = \left(\frac{L_s L_r - L_m^2}{L_r L_s}\right) i^s_{qs,pu}(k) + \frac{L_m}{L_r}\varphi^{s,i}_{qr,pu}\ (k) \qquad (3.30)$$

$$\left\{Where\ ;\ \frac{\varphi^{s,i}_{ds}}{\varphi_{base}} = \varphi^{s,i}_{ds,pu}\ ,\qquad \frac{i^{s,i}_{ds}}{i_{base}} = i^{s,i}_{ds,pu}\ \ \&\ \ i_{base} = \frac{\varphi_{base}}{L_s}\right\}$$

Then, the back emf's in (3.24) and (3.25) are normalized by dividing the base phase voltage V_{base}.

$$e^s_{ds,pu}(k) = u^s_{ds,pu}(k) - \frac{I_{base} R_s}{V_{base}} i^e_{ds,pu}\ (k) - u^s_{comp,ds,pu}(k) \qquad (3.31)$$

$$e^s_{qs,pu}(k) = u^s_{qs,pu}(k) - \frac{I_{base} R_s}{V_{base}} i^e_{qs,pu}\ (k) - u^s_{comp,qs,pu}(k) \qquad (3.32)$$

$$\left\{Where\ ;\ \frac{e_{ds}^{s}}{v_{base}} = e_{ds,pu}^{s}\ ,\qquad i_{ds} = i_{ds,pu}^{e}\ i_{base}\ \&\frac{i_{ds}R_s}{V_{base}} = \frac{i_{ds,pu}^{e}i_{base}R_s}{V_{base}}\right\}$$

Next, the stator flux linkages in the voltage model (3.22) and (3.23) are divided by the base flux linkage.

$$\varphi_{ds,pu}^{s,v}(k) = \varphi_{ds,pu}^{s,v}(k-1) + \frac{TV_{base}}{I_{base}L_m}\left(\frac{e_{ds}^{s}(k) + e_{ds}^{s}(k-1)}{2}\right) \tag{3.33}$$

$$\varphi_{qs,pu}^{s,v}(k) = \varphi_{qs,pu}^{s,v}(k-1) + \frac{TV_{base}}{I_{base}L_m}\left(\frac{e_{qs}^{s}(k) + e_{qs}^{s}(k-1)}{2}\right) \tag{3.34}$$

$$\left\{Where\ ;\ \frac{e_{ds}^{s}}{\varphi_{base}} = \frac{V_{base}}{I_{base}L_m}\right\}$$

Similar to (3.17) and (3.18) the normalized rotor flux linkages in the voltage model are:

$$\varphi_{dr,pu}^{s,v}(k) = -\left(\frac{L_sL_r - L_m^2}{L_m^2}\right)i_{ds,pu}^{s}(k) + \frac{L_r}{L_m}\varphi_{ds}^{s,v}(k) \tag{3.35}$$

$$\varphi_{qr,pu}^{s,v}(k) = -\left(\frac{L_sL_r - L_m^2}{L_m^2}\right)i_{qs,pu}^{s}(k) + \frac{L_r}{L_m}\varphi_{qs}^{s,v}(k) \tag{3.36}$$

Since the flux estimator of Induction motor module requires eight constants (K1,…, K8) to be input basing on the machine parameters, base quantities, mechanical parameters, and sampling period, the discrete-time, per-unit equations are rewritten in terms of these constants. The rotor flux linkages developed by the current model in synchronously rotating reference frame ($w_e = w_{\varphi r}$) are:

$$\varphi_{dr,pu}^{e,i}(k) = K_1\varphi_{dr,pu}^{e,i}(k-1) - K_2 i_{ds,pu}^{e}(k) \tag{3.37}$$

Where,

$$K_1 = \frac{T_r}{T + T_r} \qquad and \qquad K_2 = \frac{L_m xT}{T + T_r}$$

The rotor flux linkages developed by the current model in the stationary reference frame (w_e= 0) are:

$$\varphi_{ds,pu}^{s,i}(k) = K_4 i_{ds,pu}^{s}(k) + K_3\varphi_{dr,pu}^{s,i}(k) \tag{3.38}$$

$$\varphi_{qs,pu}^{s,i}(k) = K_4 i_{qs,pu}^{s}(k) + K_3\varphi_{qr,pu}^{s,i}(k) \tag{3.39}$$

Where,

$$K_3 = \frac{L_m}{L_r} \qquad and \qquad K_4 = \frac{L_sL_r - L_m^2}{L_rL_s}$$

The back emf's developed by the voltage model in the stationary reference frame ($w_e = 0$) is

$$e^s_{ds,pu}(k) = u^s_{ds,pu}(k) - K_5 i^e_{ds,pu}(k) - u^s_{comp,ds,pu}(k) \quad (3.40)$$

$$e^s_{qs,pu}(k) = u^s_{qs,pu}(k) - K_5 i^e_{qs,pu}(k) - u^s_{comp,qs,pu}(k) \quad (3.41)$$

Where,

$$K_5 = \frac{I_{base} R_s}{V_{base}}$$

The stator flux linkages developed by the voltage model in the stationary reference frame ($w_e = 0$) are:

$$\varphi^{s,v}_{ds,pu}(k) = \varphi^{s,v}_{ds,pu}(k-1) + K_6\left(\frac{e^s_{ds}(k) + e^s_{ds}(k-1)}{2}\right) \quad (3.42)$$

$$\varphi^{s,v}_{qs,pu}(k) = \varphi^{s,v}_{qs,pu}(k-1) + K_6\left(\frac{e^s_{qs}(k) + e^s_{qs}(k-1)}{2}\right) \quad (3.43)$$

Where,

$$K_6 = \frac{TV_{base}}{I_{base} L_m}$$

The rotor flux linkages developed by the voltage model in the stationary reference frame ($w_e = 0$) are:

$$\varphi^{s,v}_{dr,pu}(k) = -\ K_8 i^s_{ds,pu}(k) + \ K_7 \varphi^{s,v}_{ds}(k) \quad (3.44)$$

$$\varphi^{s,v}_{qr,pu}(k) = -\ K_8 i^s_{qs,pu}(k) + \ K_7 \varphi^{s,v}_{qs}(k) \quad (3.45)$$

Where,

$$K_7 = \frac{L_r}{L_m} \ \ \& \ \ K_8 = \frac{L_s L_r - L_m^2}{L_m^2}$$

The rotor flux angle developed by the voltage model and is computed by referring to a look-up table of 0°-45° entries.

$$\theta_{\varphi r}(\mathrm{k}) = \tan^{-1}\left(\frac{\varphi^{s,v}_{dr}(k)}{\varphi^{s,v}_{qr}(k)}\right) \quad (3.46)$$

$$\theta_{\varphi r,pu}(\mathrm{k}) = \frac{1}{2\pi}\tan^{-1}\left(\frac{\varphi^{s,v}_{dr,pu}(k)}{\varphi^{s,v}_{qr,pu}(k)}\right) \quad (3.47)$$

In fact, equations (3.37)-(3.47) are mainly employed to compute the estimated flux linkages in per-unit. The required parameters for this module are summarized as follows:

The machine parameters:

- stator resistance (R_s)
- rotor resistance (R_r)

- stator leakage inductance (L_{sl})
- rotor leakage inductance (L_{rl})
- magnetizing inductance (L_m)

The based quantities:

- base current (I_{base})
- base phase voltage (V_{base})

The sampling period:

- sampling period (T)

Notice that the stator self-inductance is $L_s = L_{s1} + L_m$(H) and the rotor self-inductance is $L_r = L_{r1} + L_m$(H).

3.3. Speed estimator of the 3-ph induction motor

In this section a speed estimator for the 3-phase induction motor based upon its mathematical model is investigated. The estimator's accuracy relies heavily on knowledge of critical motor parameters. The open loop speed estimator employed in this DFOC structure is a well-known method based on stationary reference frame. The structure of this algorithm is quite easy when compared to the advanced estimation techniques. The synchronous speed,w_e can be easily calculated from the derivative of the rotor flux angle in (3.19).

$$w_e = \frac{d\theta_{\varphi r}}{dt} = \frac{1}{(\varphi_r^s)^2}\left(\varphi_{dr}^s \frac{d\varphi_{qr}^s}{dt} - \varphi_{qr}^s \frac{d\varphi_{dr}^s}{dt}\right) \tag{3.48}$$

Substituting (3.2) and (3.3) into (3.48), and a suitable rearrangement of them gives the synchronous speed, w_e as;

$$w_e = \frac{d\theta_{\varphi r}}{dt} = w_r + \frac{L_m}{T_r}\left(\frac{\varphi_{dr}^s i_{qs}^s - \varphi_{qr}^s i_{ds}^s}{(\varphi_r^s)^2}\right) \tag{3.49}$$

$$Where \ \frac{L_m}{T_r}\left(\frac{\varphi_{dr}^s i_{qs}^s - \varphi_{qr}^s i_{ds}^s}{(\varphi_r^s)^2}\right)$$

The second term of the right hand in (3.49) is known as slip that is proportional to the electromagnetic torque as seen in equation (2.8) when the rotor flux magnitude is maintaining constant. Thus, the rotor speed can be found as

$$\therefore \quad w_r = w_e - \frac{L_m}{T_r}\left(\frac{\varphi_{dr}^s i_{qs}^s - \varphi_{qr}^s i_{ds}^s}{(\varphi_r^s)^2}\right) \quad w_s \, , is \ slip \ speed \tag{3.50}$$

3.3.1. Speed estimation in discrete Time and Per-Unit

If the per-unit concept set forth here is applied to (3.50) and dividing everything by base speed w_{base} and multiplying the slip speed by the square of base flux will give us

$$w_{r,pu} = w_{e,pu} - \frac{1}{w_{base}T_r}\left(\frac{\frac{\varphi^s_{dr}}{\varphi_{base}}\left(\frac{i^s_{qs}L_m}{\varphi_{base}}\right) - \frac{\varphi^s_{qr}}{\varphi_{base}}\left(\frac{i^s_{ds}L_m}{\varphi_{base}}\right)}{\left(\frac{\varphi^s_r}{\varphi_{base}}\right)^2}\right)$$

$$w_{r,pu} = w_{e,pu} - \frac{1}{w_{base}T_r}\left(\frac{\varphi^s_{dr,pu}\, i^s_{qs,pu} - \varphi^s_{qr,pu}\, i^s_{ds,pu}}{\left(\varphi^s_{r,pu}\right)^2}\right) \quad (3.51)$$

Where,

$$\varphi^s_{dr,pu} = \frac{\varphi^s_{dr}}{\varphi_{base}} \quad , \quad i^s_{qs,pu} = \frac{i^s_{qs}L_m}{\varphi_{base}} \quad \& \quad \frac{L_m}{\varphi_{base}} = \frac{1}{I_{base}}$$

Note that $w_{base} = 2\pi f_{base}$ is the base electrical angular velocity (rad/sec), $\varphi_{base} = L_m I_{base}$ is the base flux linkage (volt.sec), and I_{base} is the base current (amp). Equivalently, another form for (3.51) is

$$w_{r,pu} = w_{e,pu} - K_1\left(\frac{\varphi^s_{dr,pu}\, i^s_{qs,pu} - \varphi^s_{qr,pu}\, i^s_{ds,pu}}{\left(\varphi^s_{r,pu}\right)^2}\right) \quad (3.52)$$

Where,

$$K_1 = \frac{1}{w_{base}T_r}$$

The per-unit synchronous speed can be calculated as

$$w_{e,pu} = \frac{1}{2\pi f_{base}}\frac{d\theta_{\varphi r}}{dt} = \frac{1}{f_{base}}\frac{d\theta_{\varphi r,pu}}{dt} \quad (3.53)$$

$$Where\ \theta_{\varphi r,pu} = \frac{\theta_{\varphi r}}{2\pi}$$

Where f_{base} the base is electrical (supplied) frequency (Hz) and 2π is the base angle (rad).

Discretizing equation (3.53) by using the backward approximation, yields

$$w_{e,pu}(k) = \frac{1}{f_{base}}\left(\frac{d\theta_{\varphi r,pu}(k) - d\theta_{\varphi r,pu}(k-1)}{T}\right) \quad (3.54)$$

Where T being the sampling period (sec). Equivalently, another form for (3.54) is

$$w_{e,pu}(k) = K_2\left(d\theta_{\varphi r,pu}(k) - d\theta_{\varphi r,pu}(k-1)\right) \quad (3.55)$$

Where $K_2 = \frac{1}{f_{base}T}$ which is usually a large number. In practice, the typical waveforms of the rotor flux angle, $\theta_{\varphi r,pu}$, in both directions can be seen in Figure 3.1. In order to take the discontinuity of angle from 360° to 0° (CCW) or from 0° to 360° (CW) into account, the differentiator is simply operated only within the differentiable range as seen in this figure. This differentiable range does not cause significant loss of information to compute the estimated speed.

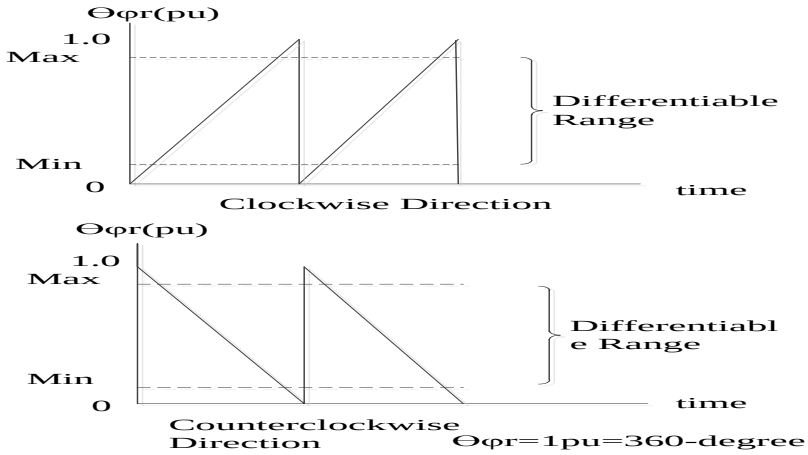


Figure3.1: The waveforms of rotor flux angle in both directions

In addition, the synchronous speed in (3.55) is necessary to be filtered out by the low-pass filter in order to reduce the amplifying noise generated by the pure differentiator in (3.55). The simple 1st –order low-pass filter is used, then the actual synchronous speed to be used is the output of the low-pass filter, $w_{e,pu}{}^*$, seen in (3.50). The continuous-time equation of 1st-order low-pass filter is as

$$\frac{dw_{e,pu}{}^*}{dt} = \frac{1}{T_c}\left(w_{e,pu} - w_{e,pu}{}^*\right) \tag{3.56}$$

Where $T_c = \frac{1}{2\pi f_c}$ is the low-pass filter time-constant (sec), and fc is the cut-off frequency (Hz). Using backward approximation, then (3.506) becomes

$$w_{e,pu}{}^*(k) = K_3 w_{e,pu}{}^*(k-1) + K_4 w_{e,pu}(k) \tag{3.57}$$

$$where \quad K_3 = \frac{T_c}{T_c + T} \quad and \quad K_4 = \frac{T}{T_c + T}$$

As a result, only three equations (3.52), (3.55), and (3.57) are mainly employed to compute the estimated speed in per-unit.

Chapter four

Simulation Results and Discussion

4.1. Introduction

In this thesis, the simulation of the model and controller was performed using MATLAB/Simulink software packages. Five different simulations were carried out to observe the effectiveness of the derived algorithms for the flux and speed estimation with open loop speed estimator scheme. In these simulations the PI regulators are tuned as to give a satisfactory speed response when parameter errors are absent.

The induction motor used in the design was already available in the standard Simulink library. The blocks that had to be constructed were the Rotor flux based speed estimator and direct vector control strategy is employed. The rotor flux based open-loop speed estimator estimates the rotor flux angle which is used for calculate synchronous speed. This synchronous speed is needed for rotor flux speed estimation. The simulation parameters used in the simulations are given in Table 4-1.

Table 4.1: Three phase squirrel cage 180W, 1320 rpm, 380 V, and 50Hz IM specifications:

No	Parameter	Symbol	Value
1	PWM inverter frequency	Freq_Switching	10kHz
2	Stator Resistance	R_s	11.05Ω
3	Rotor Resistance	R_r	6.11Ω
4	Stator Inductance	L_s	0.316426H
5	Rotor Inductance	L_r	0.316426H
6	Magnetizing Inductance	L_m	0.2939H
7	Moment of inertia	J	0.009kg.m2
8	Friction factor	f	0.00061N.m.s
9	No of pole pair	p	2
10	Initial condition	In_c	0

The simulation block diagram for the open loop speed estimator by Direct (rotor) flux observer can be seen in Figure 4.1. It comprises a number of components and subsystem; the desired flux angle, estimated flux, SVPWM, speed and current controller, and the estimated speed. Outputs of the system which have been used

to evaluate the performance of the speed estimators are the estimated flux angle, the estimated speed, estimated rotor flux, quadrature axis current that generate desired torque and stator currents.

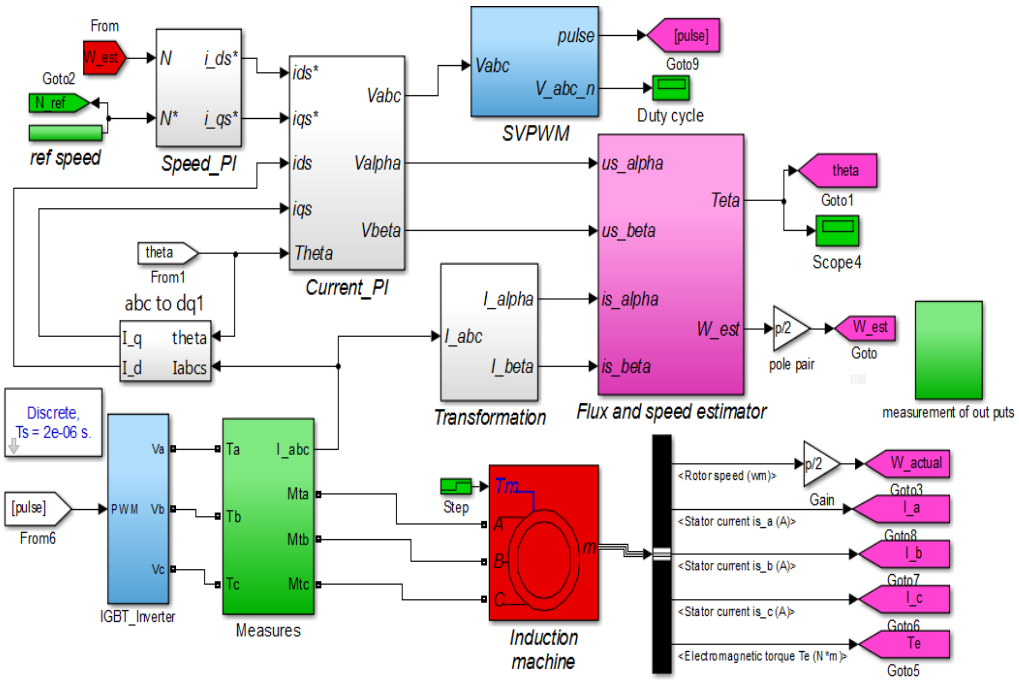


Figure 4.1: Rotor flux based speed Estimator of three phase-IM

The error of the system which is yielded by the sum of desired and estimated speed constitutes the input of the speed controller (PI). Figure 4.2 describes the block diagram of PI speed control which is perfectly used to give the desired q-axis current for production of electromagnetic torque. The input of this block is rotor estimated speed that the value depends on the desired estimated rotor flux angle (eq. 3.19) while the output of this block become the input of current controller (PI).

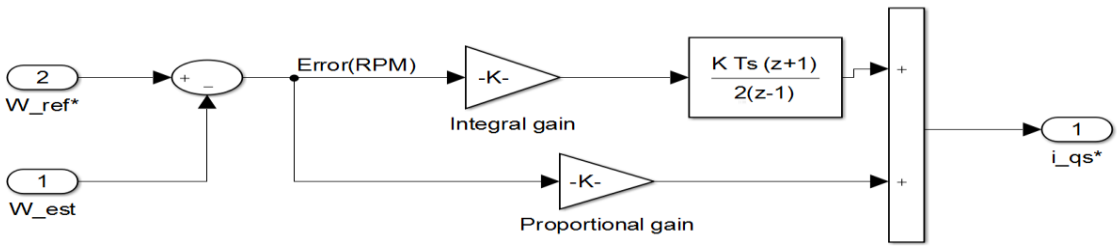


Figure 4.2: PI speed controller block

Figure 4.3.. Shows the block diagram of the current controller for both d-and q-axis current which has an important task in controlling the desired dq-axis voltage. By converting those voltages into abc transformation, we can easily develop VSPWM with the help of V_dc for triggering the gates of an inverter.

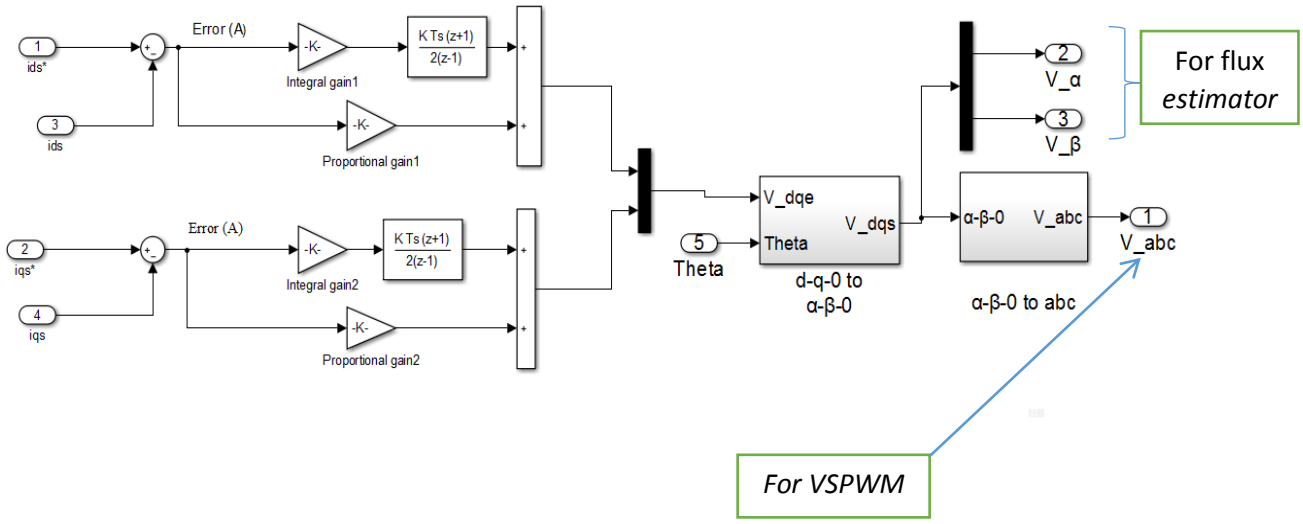


Figure 4.3: PI current controller block

Figure 4.4 illustrates a general block diagram of the estimated speed, rotor flux position and estimated rotor flux system dynamics. Where the speed and rotor flux position are estimated from estimated rotor flux. The outputs of the rotor flux estimator are the desired rotor flux position and estimated speed. As can be seen from figure 4.4, there is no feedback speed to the estimator. That is why it is open-loop estimator.

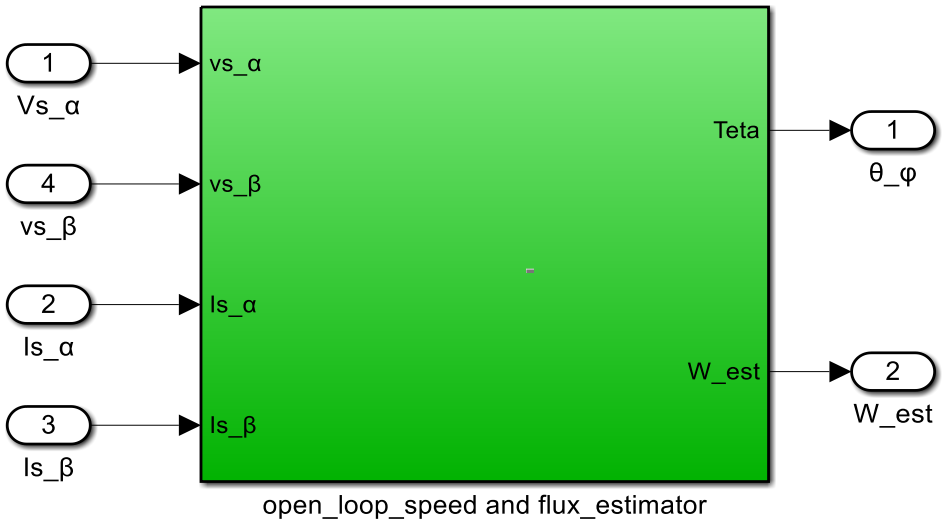


Figure 4.4: rotor flux, rotor flux angle and speed estimator block

Figure 4.5 illustrates a block diagram of the open loop speed estimator system dynamics where the synchronous speed is obtained from the derivation of rotor flux angle. The rotor estimated speed is estimated based on synchronous speed and slip speed which directly proportional to electromagnetic torque.

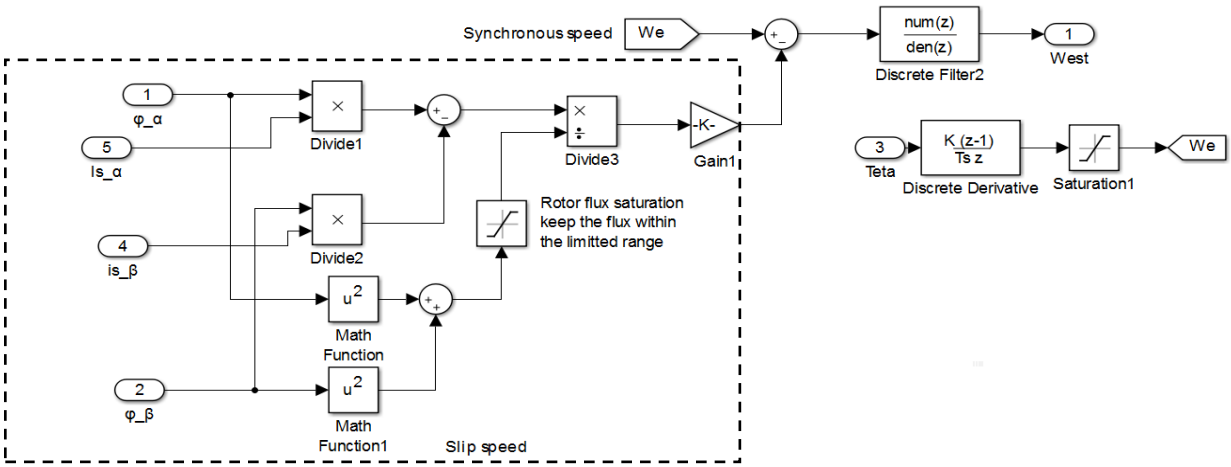


Figure 4.5: open-loop speed estimator

4.2. Simulation Results

The SVPWM algorithm implemented in this thesis by DSP is simulated before proceeding through the experimental works to verify its results. Figures 4.6 to 4-15 show the simulation results of rotor flux based speed estimator by DFOC that was developed using MATLAB/Simulink with the zero initial condition. The aim of this simulation is to understand the speed estimation accuracy when the load torque is zero. The simulation result of rotor flux based sensorless speed control of induction motor drive was carried out to evaluate its tracking performance.

The first simulation is the angle of rotor flux with respect to the stationary frame ds-axis is estimated with the proposed flux linkages in rotating reference frame are found then the rotor flux angle estimation follows the calculation method given in section 3.3.1. Figure 4.6 shows the rotor flux angle in radian. Note on the graph that the time elapsed from -3 radian angle to +3 rad angle is 0.05 sec. This valuable information gives us the synchronous speed. Because it says that rotor flux rotates at this speed with respect to the stationary ds-axis. By this knowledge the rotating reference frame current components (for example i_{de} and i_{qe}) can easily be transformed to stationary reference frame components (for example i_{ds} and i_{qs}) or vice versa. The estimated

rotor flux which is used to generated rotor flux angle with reference speed of 100 rad/sec is shown as in figure 4.7. Both fluxes are 90-degree out of phase.

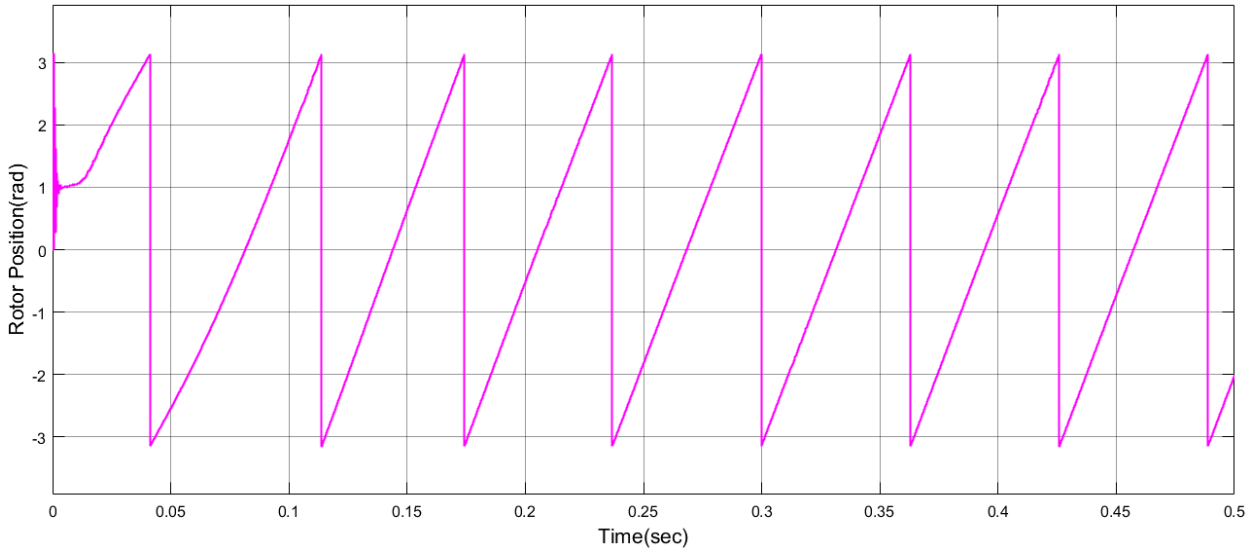


Figure 4.6: rotor flux position vs time

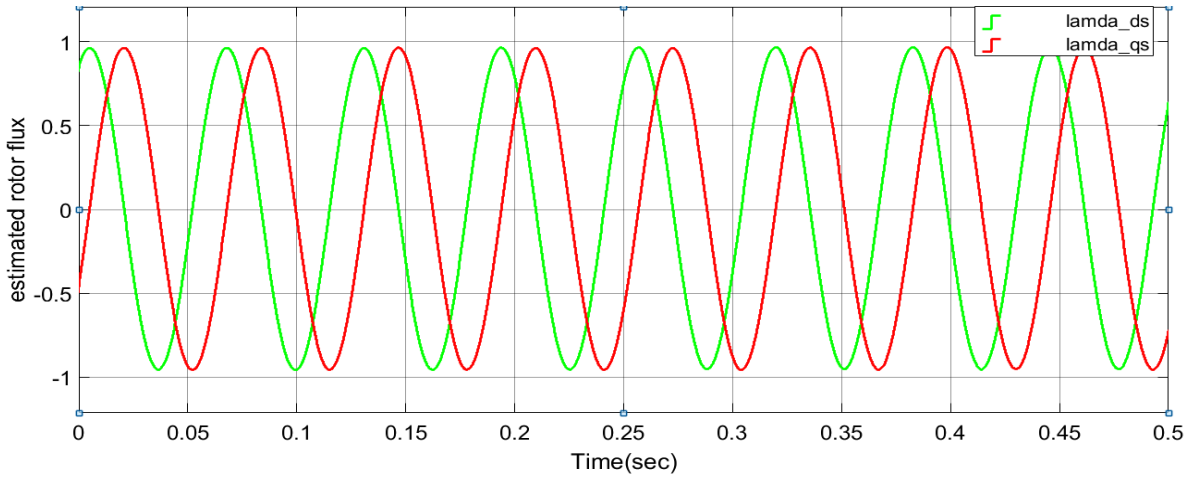


Figure 4.7: rotor flux in alpha-beta reference frame vs time

The second simulation result for the rotor flux estimated based speed control of 3-phase IM is the three phase stator currents which are generated by the three phase voltage source inverter. This three phase inverter is controlled by SVPWM block for appropriate stator current generation. These three phase currents should be equal magnitude and 120° phase shift with each other for appropriate rotating flux generation as shown in Figure 4.8.

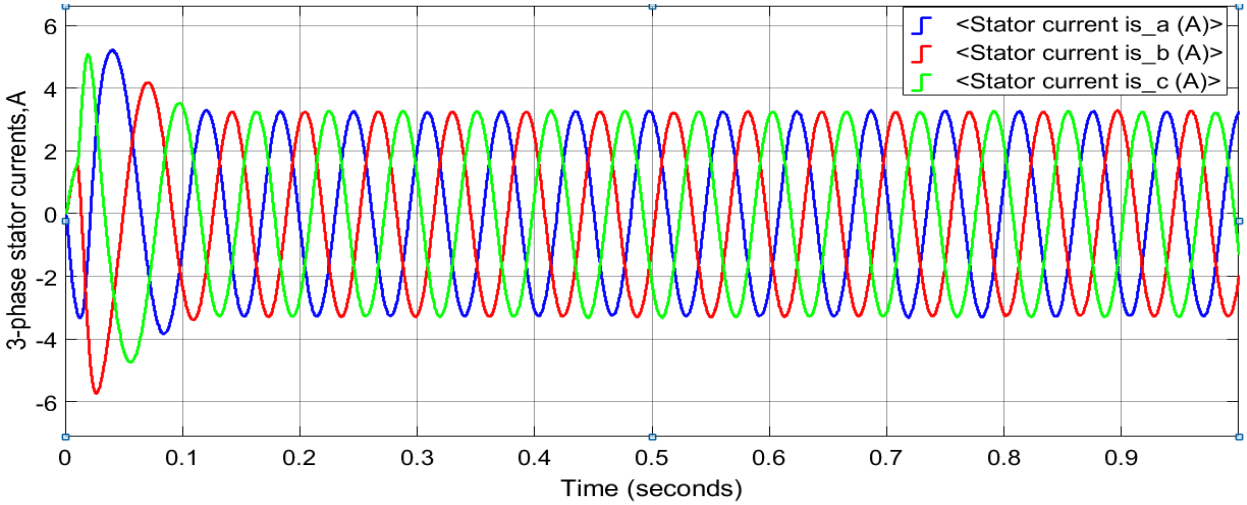


Figure 4.8: Three phase stator current in A vs. time

As we see from in the above Figure, the appropriate stator phase current value is generated with good accuracy. Hence the system can feed the appropriate stator voltage to the Induction Motor. If the voltage applied to the motor is applied with appropriate magnitude and frequency, the speed of the motor is respected as set to the reference value.

Figure 4.9 shows simulation result for an actual and estimated speed with reference speed of 100rad/sec. The system can also follow the reference signal with a rise time of less than 0.15 second and settling time of 0.23second. The system also gives an overshoot value of 0.038.

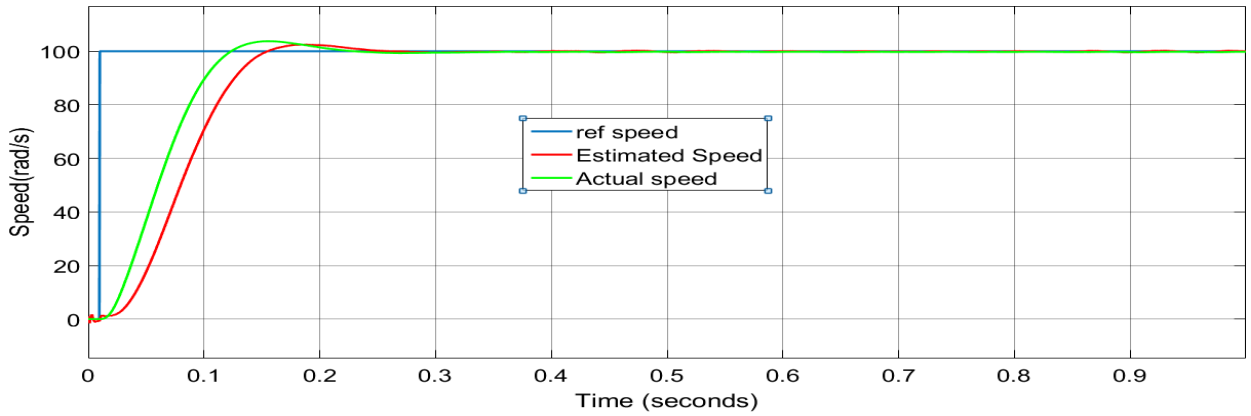


Figure 4.9: Actual and Estimated speed vs. time

As can be seen from diagram shown below, the error signal between actual and estimated speed is ±0.3rad/sec which is tolerable. So the actual speed tracks the reference speed with minimum and acceptable error even the estimator is open loop.

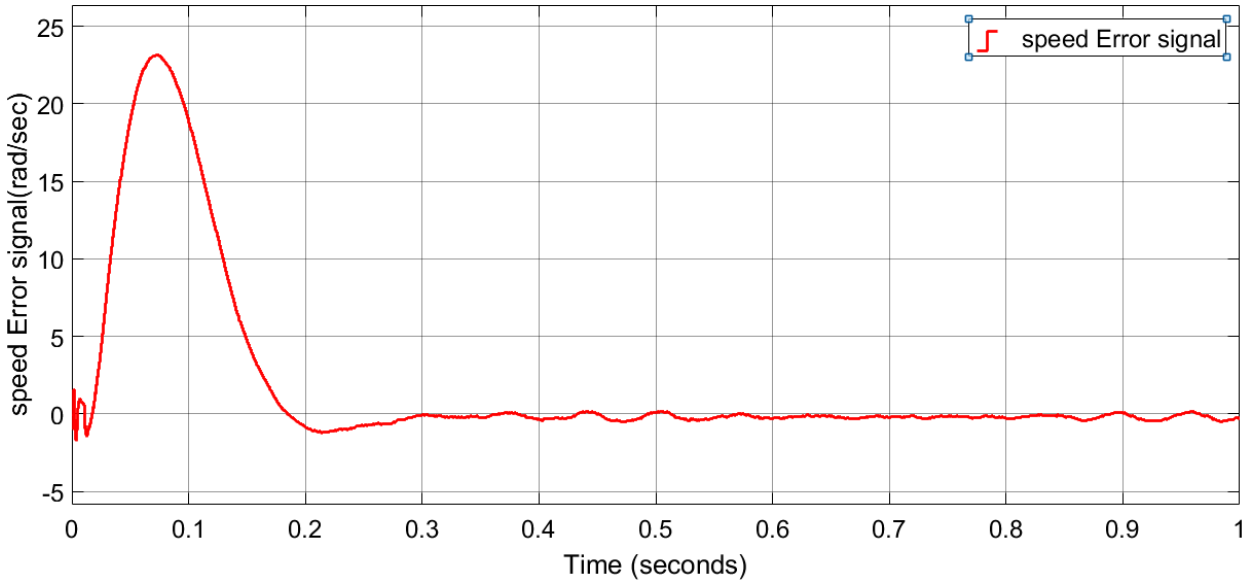


Figure 4.10: Error signal between actual and estimated speed in rad/sec vs. time

Using the same parameters in the induction motor and rotor flux based open loop speed estimator, the tracking performance of the estimator can be examined by changing the speed reference of the system.

As shown in Figure 4.11 the proposed estimator tracks the pulse with different reference speed input (stair case input). This shows the tracking performance of the estimator and actual speed to the reference speed can be examined by changing the reference of the system.

A reference speed of 20rad/s was initially applied at t = 0.5 sec, increased to 60rad/s ,100rad/s and return back to zero speed at t = 1 sec, t=1.5 sec and at t=1second respectively. As shown in Figure 4.11 the estimated and the actual speed follow the reference speed with good accuracy and with some tolerable overshoot and it takes 0.15sec to track the reference speed at different level of speed including low speed region.

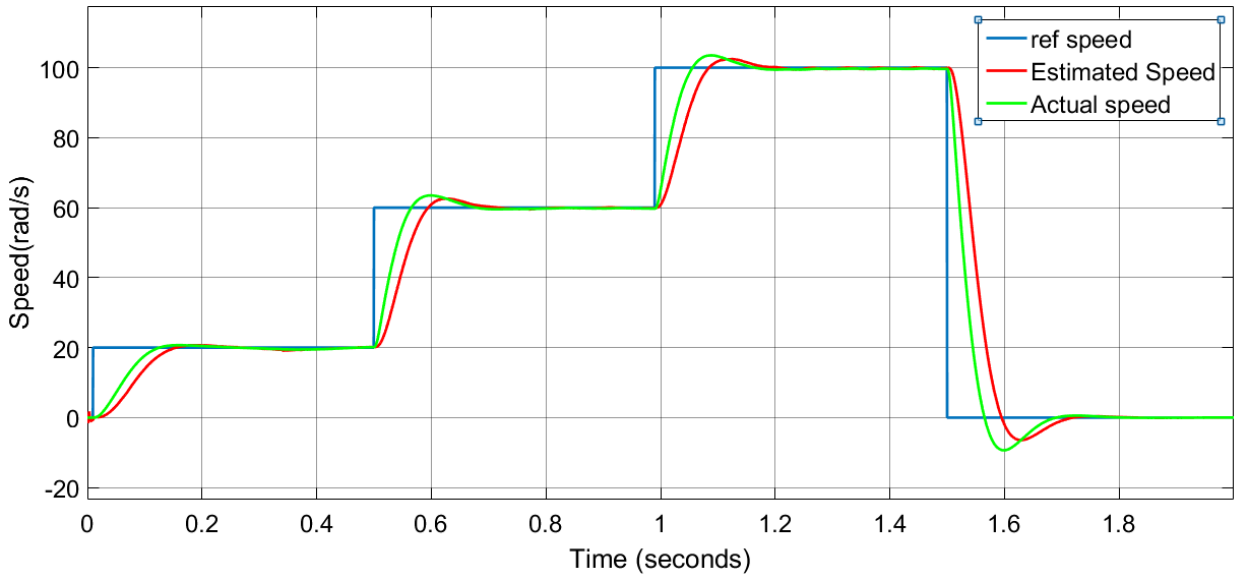
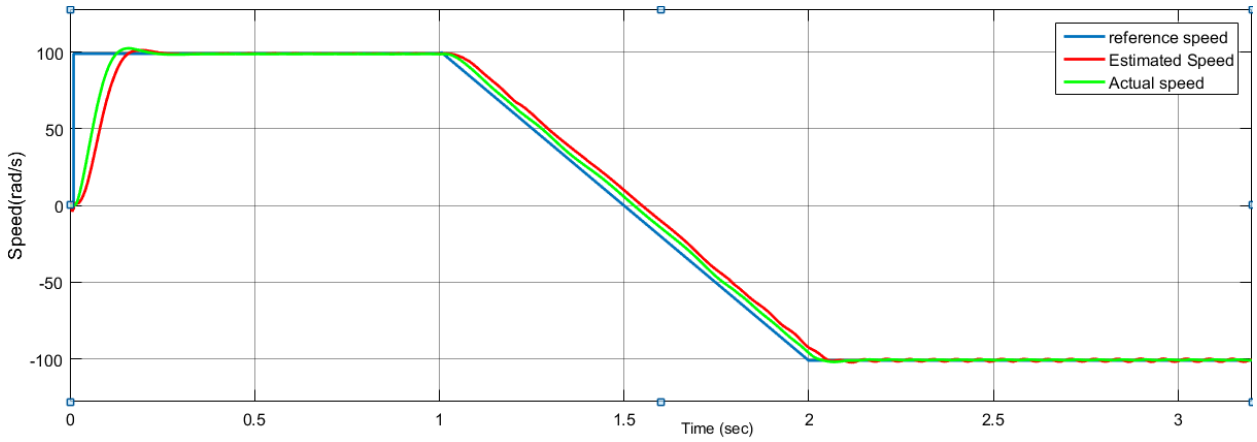
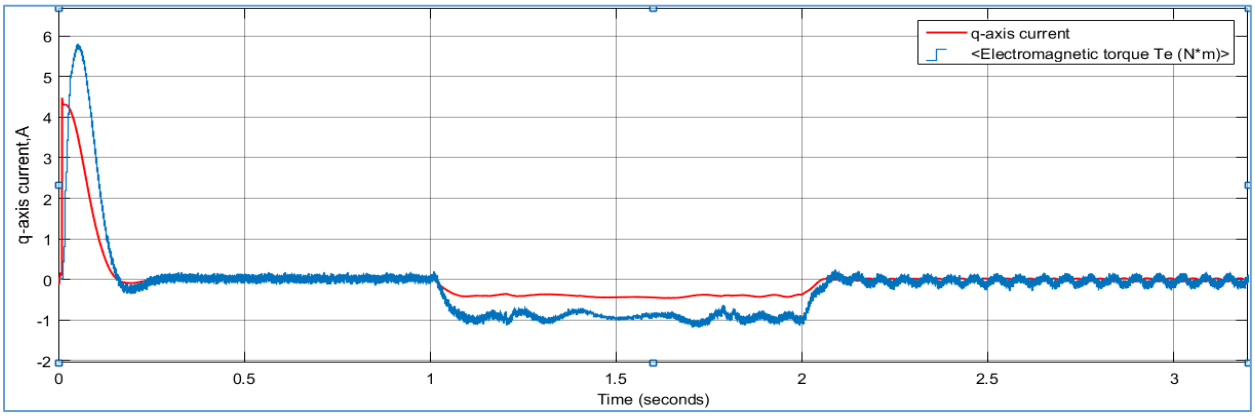


Figure 4.11: Reference speed and speed response at four ramped step speed levels.

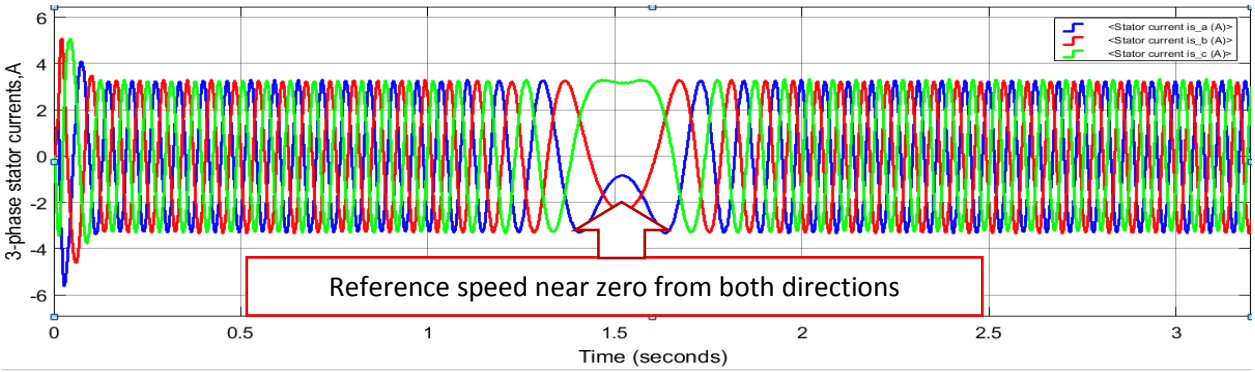
Another reference input signal is ramp with decelerating torque for 1 to 2 sec and constant between 0-0.02sec and 2-3.5sec. It is clear that, the torque is negative between 1sec to 2sec as the reference speed is decelerating for this time interval and is more negative than the q-axis current as it is expected in figure (4.12,b). And also the three stator currents unable to make a phase shift of 120-degree phase shift with each other when the reference speed near to zero from both direction as clearly indicated in figure (4.12,c).



a) actual and estimated speed vs time with accelerating and decelerating speed reference



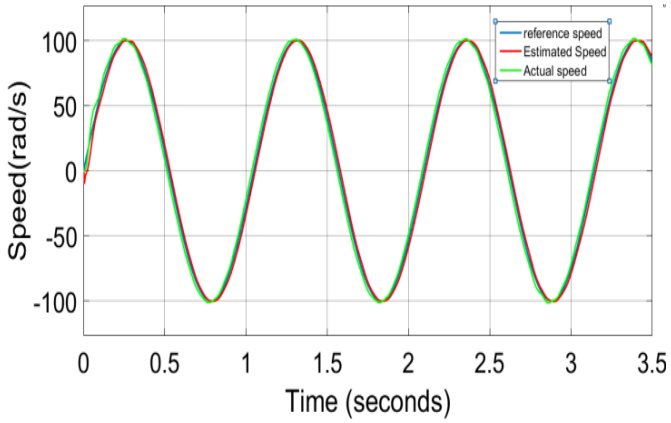
b) q-axis current vs time and Electromagnetic torque vs time



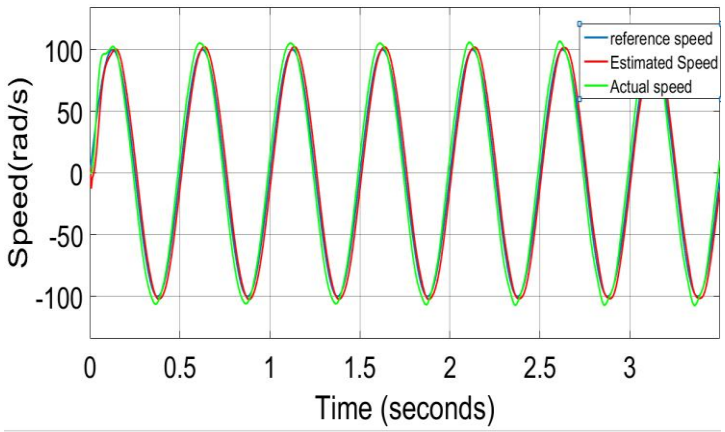
c) The three phase stator currents for this accelerating and decelerating reference speed

Figure 4.12: Actual and Estimated speed with ramp input

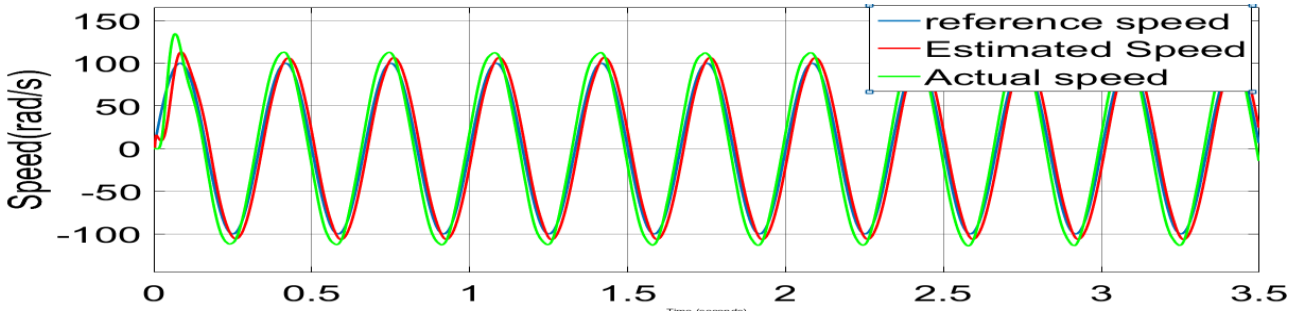
The other most important reference speed is the case of sinusoidal input with suitable bandwidth. The speed tracks its reference value within acceptable bandwidth. We will look at for five different frequencies to select at which bandwidth; the actual speed tracks the reference speed. Figure 4.13 (a) is the sinusoidal input of bandwidth 6.28 rad/sec. It is clear that the speed tracks the reference sinusoidal input with minimum error of about 0.296 but increase as the band width becomes 31.4 rad/sec. For bandwidth above 31.4 rad/sec ,the actual speed cannot track the given reference speed.



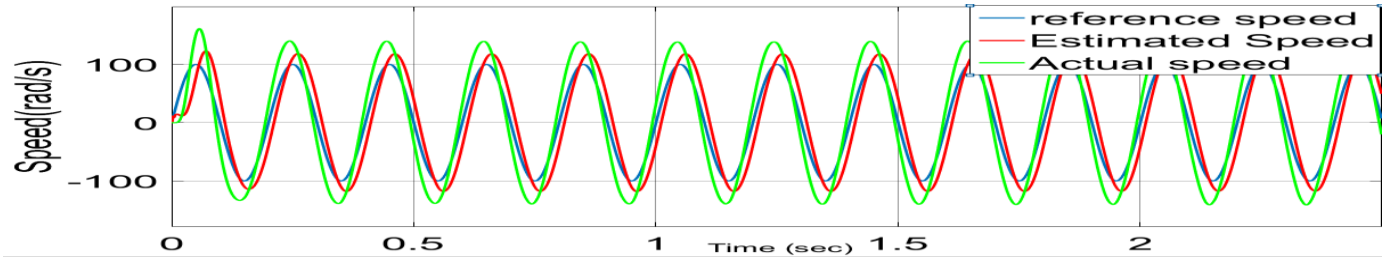
a) Frequency of 6.28rad/sec



b) frequency of 12.56 rad/sec



c) frequency of 18.84 rad/sc



d) frequency of 31.4 rad/sec

Figure 4.13: actual and estimated speed vs time with sinusoidal reference input

The case of low speed or zero reference speed is shown in figure 4.14. As can be shown, it is not easy to control speed of induction motor by open loop and rotor flux estimator at low speed even at zero speed.it seems like oscillating.

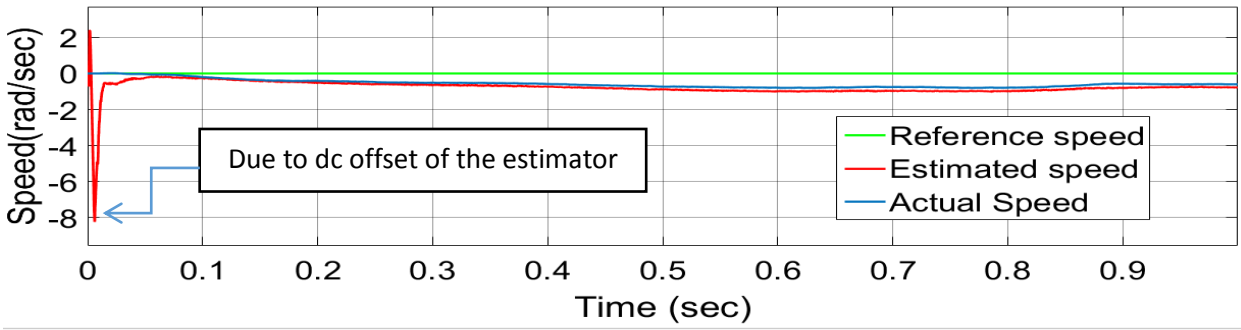
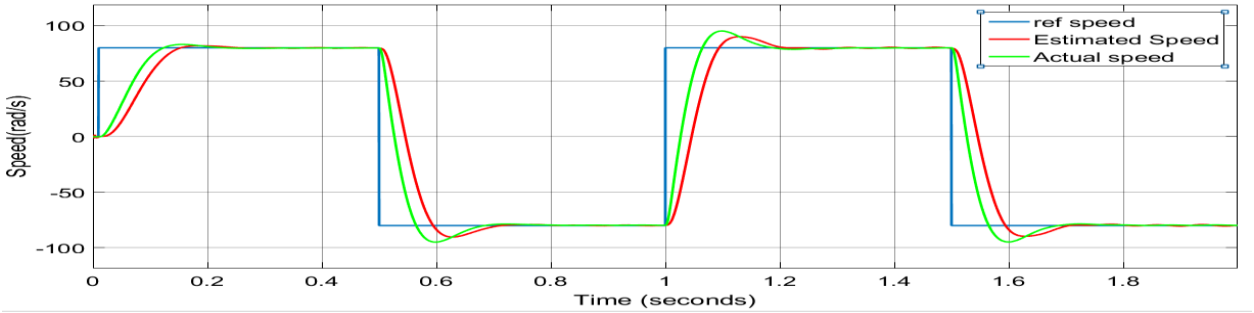
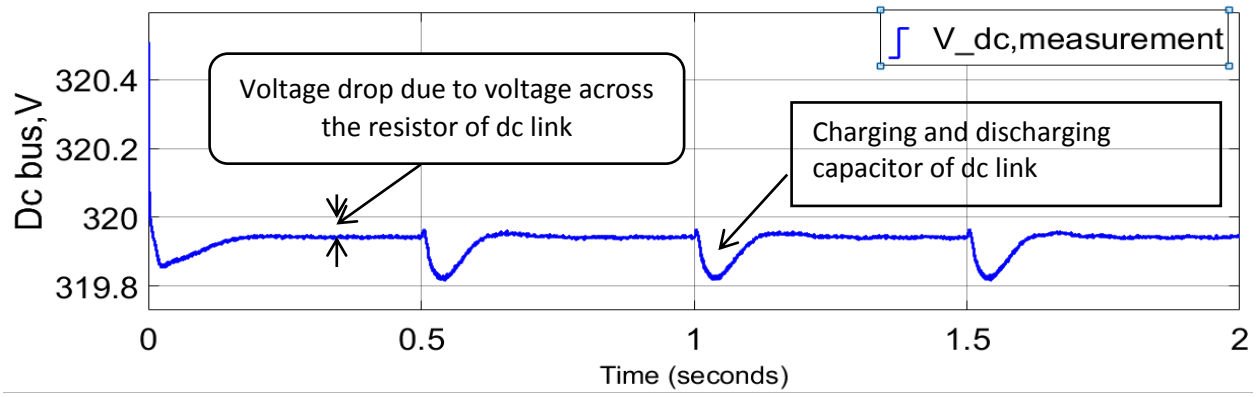


Figure 4.14: actual and estimated speed vs time with zero reference input

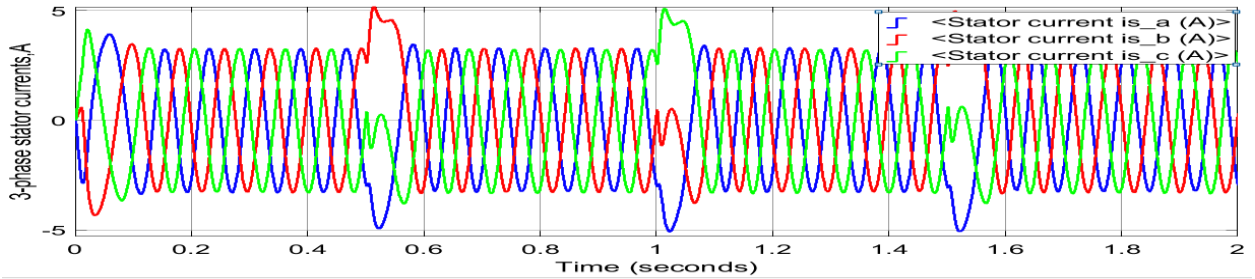
Finally the estimator is working for regenerative mode of operation. Motor speed is changed from +80 rad/s to -80rad/s . The drive is operated in motoring mode up to 0.5second. Thereafter, it enters in the regenerating mode of operation for 0.5 second (which is the total time of 1 sec) and come back to motoring region after 0.5 second. As shown in Figure 4.15 the estimate and the actual speed follow the reference speed successfully with certain transient response in the regenerative mode. When the motor enter the regenerative mode, the overshot is increased and becomes 0.398.When the motor enter into the regenerative mode, the direction of motor is reversed as it is expected in experimental part. Correspondingly, the dc bus voltage is charging for 0.015 sec when the drive is operating in motoring mode and discharging for the same in the regenerating mode. It is also clear that, the starting of the three phase stator current increasing when the drive is enter from one region to the other operating region.



a)actual, estimated and reference speed of variable speed drive



b)Dc bus voltage

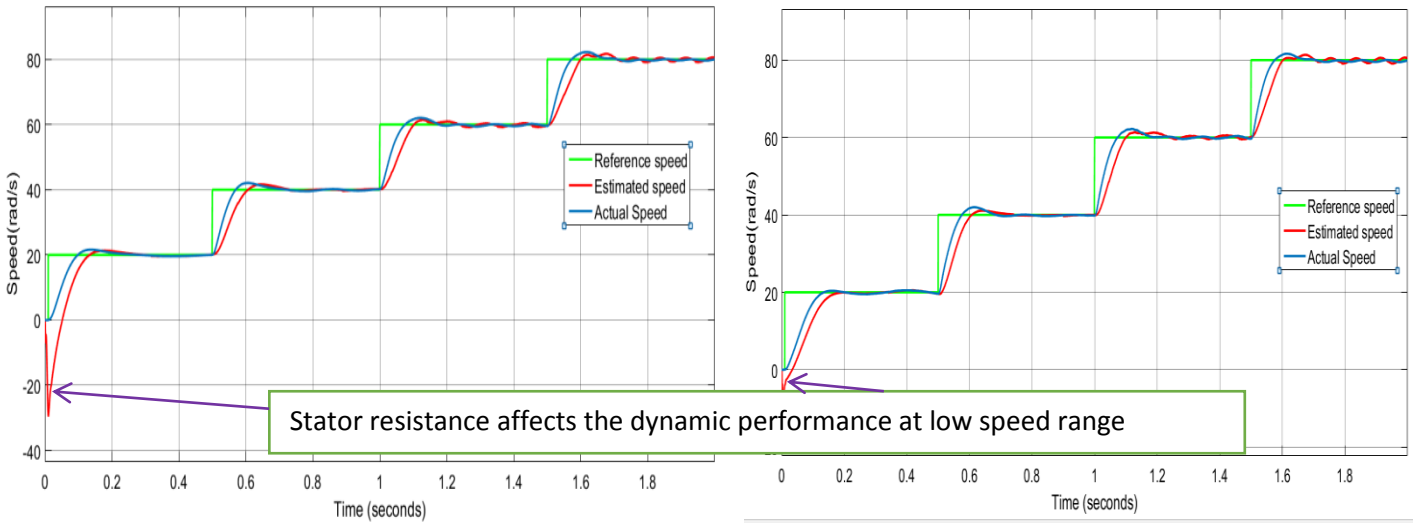


c) 3-phase Stator current

Figure 4.15: The actual and estimated speed performance in regenerative mode and corresponding dc bus & stator currents at motoring and generating mode of operation.

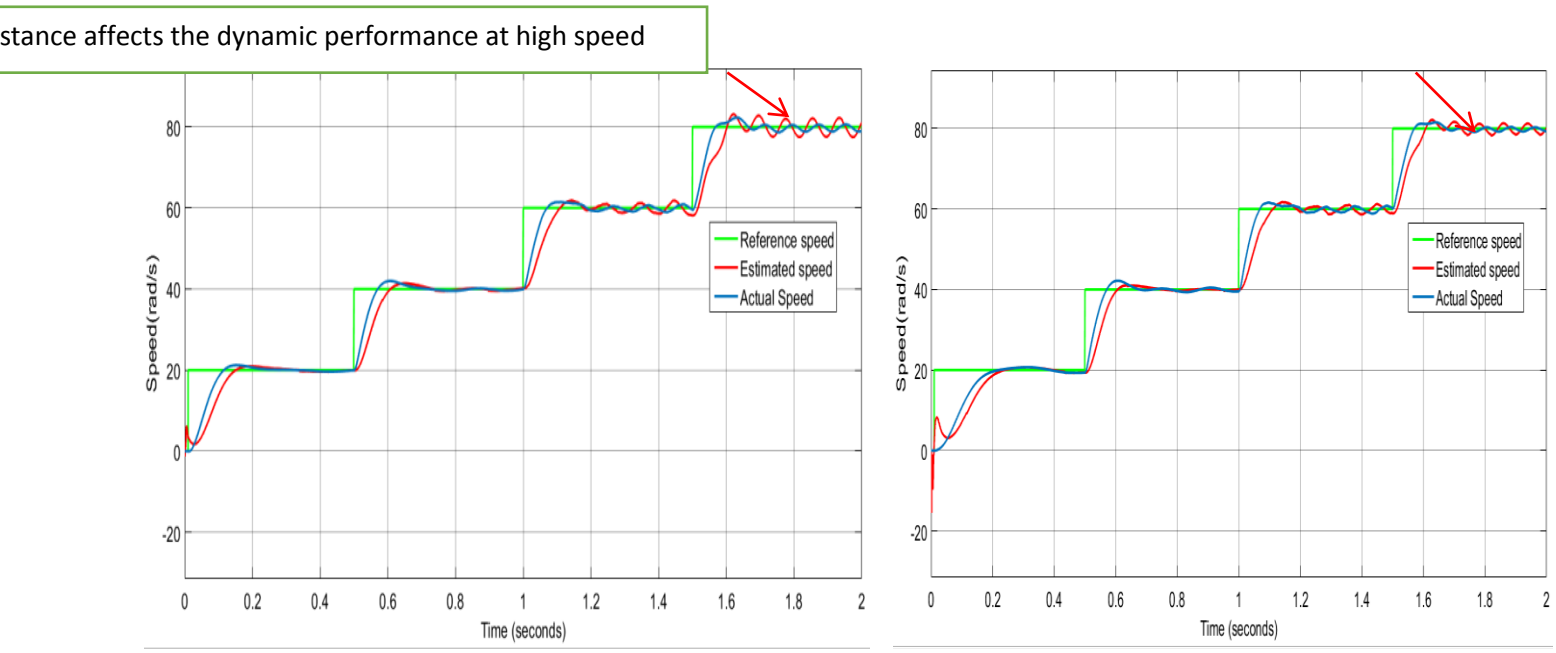
The effect of stator and rotor resistance on rotor flux speed estimator

Assume the rotor and stator resistance are vary increasing 50% and decreasing 50% with their nominal value and the other parameters are constant; with the speed 58%,43%, 29% and 14% of the rated value. As shown in Figure 4.16(a and b) when the stator resistance varies by ±50% of its nominal value the estimated speed exactly follow the actual speed for high reference speed and it takes a long rise time to track for low speed region. There is a transient problem at low speed range. As shown in Figure 4.17(a and b), when rotor resistance changes ±50%,there is a problem at high speed and it takes a long rise time to track the reference speed.



a) 50% increasing of Rs b) 50% decreasing Rs

Figure 4.16: Variation of Rs from its nominal value for different rotor speed



50% increasing of Rr b) 50% decreasing of Rr

Figure 4.17: Variation of Rr from its nominal value for different rotor speed

The effect of external load on the torque response

To find the torque response quickness the motor is stared with no-load for 0.75 sec. This value is increased to 2Nm (33.33% of rated torque) after 0.75 second which result in a drop of motor speed. This happens because of the mismatch in the torques, i.e. the developed torque is less than the load torque. To compensate for this mismatch, the controller increases q-axis current which generate the developed torque and then the motor speed increases and comes back to the set point as shown in Figure 4.18 after 1.4 sec.

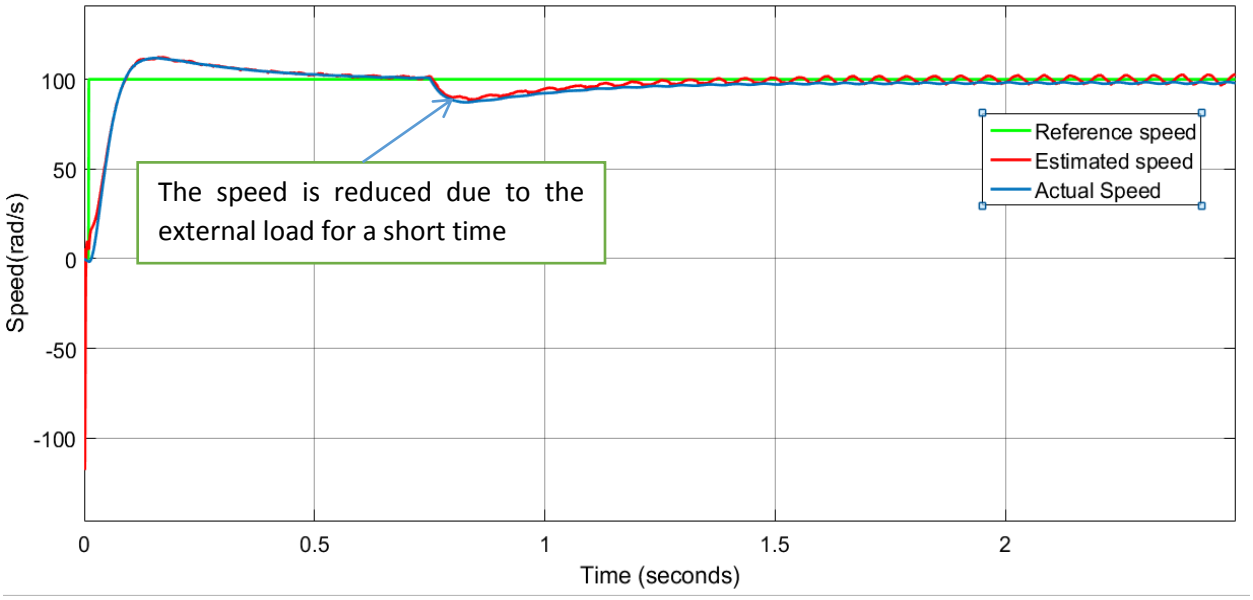


Figure 4.18: Load torque effect on rotor flux based open loop speed estimator

Chapter Five

Implementation of Flux and Speed Estimation for Sensor-less DFOC

5.1. Introduction

To test the performance of the proposed scheme, experimental work is carried out on the induction motor with the same parameters given in Table 4.1. Experimental setup of the induction motor control developed system is shown in Figure 5.1, which is based on a high voltage motor control developer's kit produced by Texas instruments Company. The PWM frequency for the experiment is 60 kHz and the ISR frequency 10 kHz is selected for best operation of the motor. Experimental data are captured for display and analysis via a graph tool in the code composer studio (CCS).

The overall experimental system contains both hardware and software. The hardware includes: High voltage motor control kit with TMS320F28035 DSP control card, three phase induction motor, PC with Code Composer Studio (CCSv6.0) installed, digital oscilloscope for capturing PWM outputs, digital multi-meter, JTAG probe, and variable DC power supply. The software includes different coordinate transformation algorithm, controller's algorithm, rotor flux based open loop speed estimator algorithm, space vector pulse width modulation algorithm and the kit interrupt software algorithm. The overall experimental block diagram contains mainly three modules: piccolo TMS320F28035 control card, power electronics module and induction motor as shown in Figure 5.1.

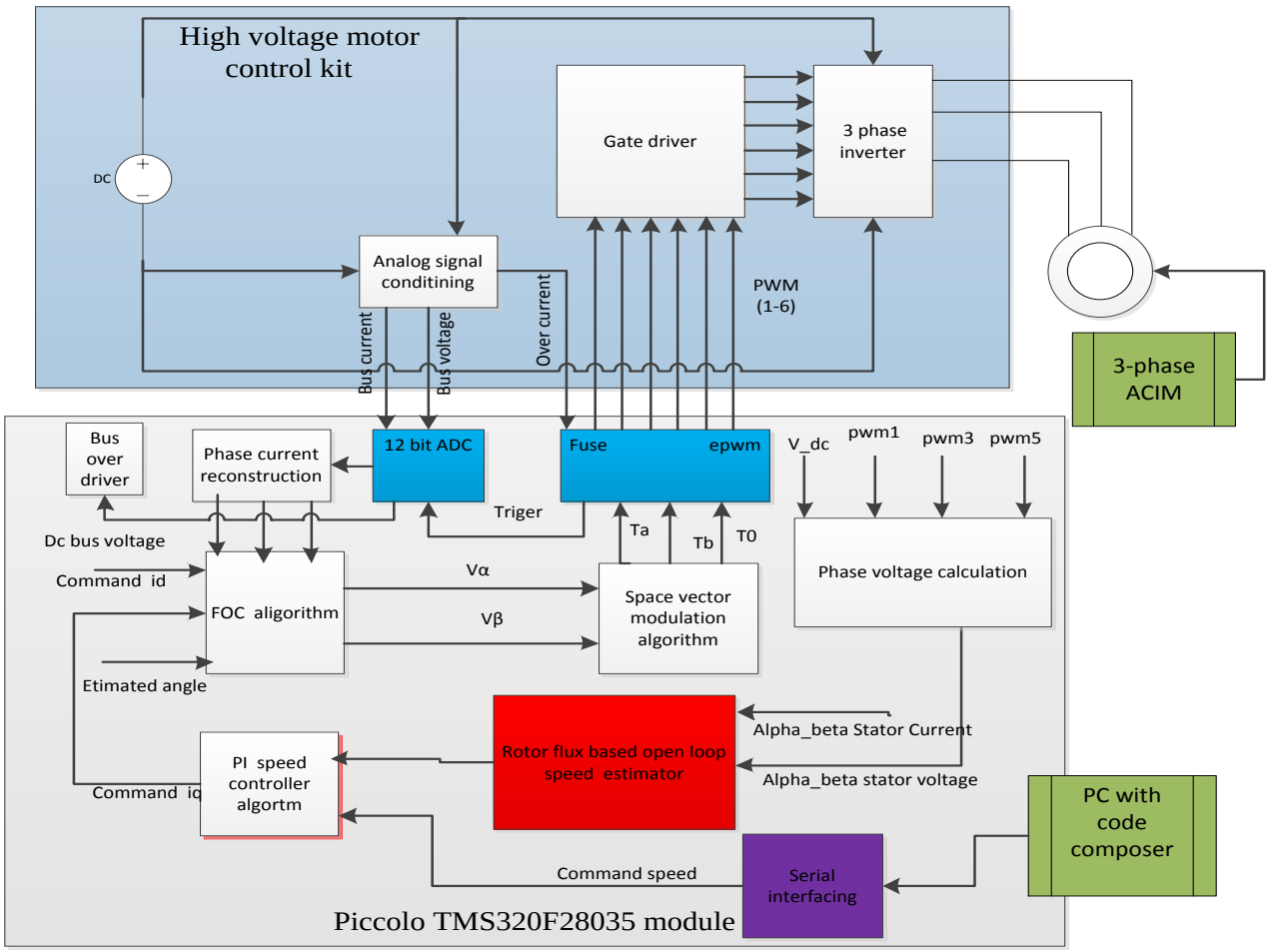


Figure 5.1: Experimental block diagram for Rotor flux based senserless speed control of IM

The digital motor control software library is the collection of digital motor control software modules. These modules allow users to quickly build or customize their own systems. The library supports the AC induction motor and comprises both peripheral dependent software drives and TMS320F28035 control card dependent modules.

As shown in Figure 5.1 the power electronics module include: gate driver which control three phase voltage source inverter, analog signal conditioning circuit used to sense stator current by opto-coupler mechanism and DC bus voltage, and three phase invertor to drive the motor. The second module is piccolo TMS320F28035 control card consist: control algorithms and hardware module. Control algorithms include like FOC, PI speed controller, phase current and voltage reconstruction, rotor flux based open loop speed estimator and SVPWM algorithm.

5.2. Characteristics of HVMTRPFCKIT

The High Voltage Digital Motor Control (DMC) and Power Factor Correction (PFC) kit (HVMTRPFCKIT), provides a great way to learn and experiment with digital control of high voltage motors. Furthermore, the module is an excellent platform to develop and run software from the TMS320F28035 processor. To simplify code development and shorten debugging time, a C2000 Tools Code Composer drive is provided. In addition, an on board JTAG connector provides interface to emulators, operating with other debuggers to provide C high level language debug.

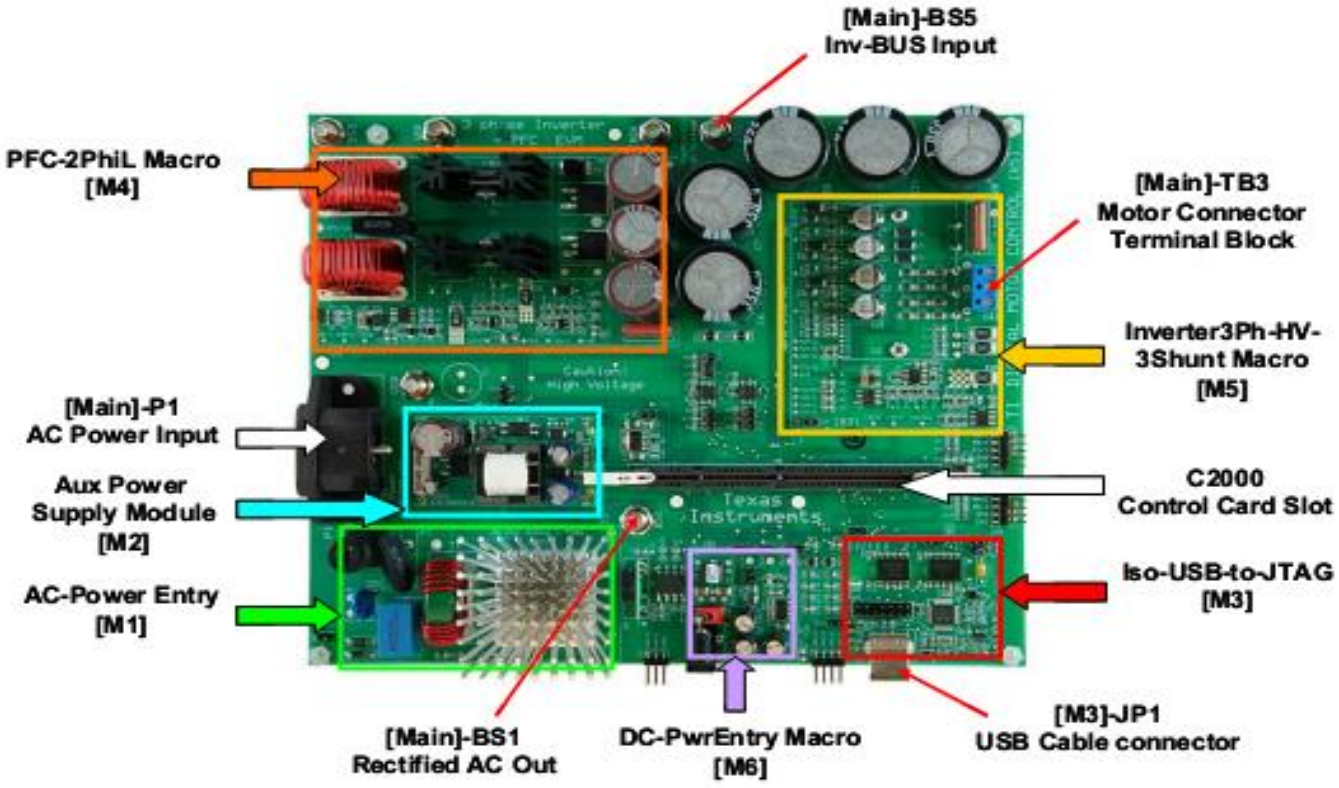


Figure5.2: HVDMCMTRPFC kit board macros [29]

Key Features of the TMS320F28035 Piccolo™ Microcontrollers [29]

The TMS320F28035 Piccolo family of microcontrollers provides quick evaluation boards for the TMS320F28035, C2000 controller as well as a noise resistant plug-in card with isolate JTAG emulation build in to enable quick debug and bring up of boards requiring isolated emulator.

Features

- High-Efficiency 32-Bit CPU (TMS320C28x)
  - Fast Interrupt Response and Processing
  - Unified Memory Programming Model
  - Code-Efficient (in C/C++ and Assembly)
- Clamping diode protection at ADC input pins
- Anti-aliasing filter (noise filter) at ADC input pins
- Isolated JTAG Emulator though the USB connector
- High-speed performance,
- Cost effectiveness.
- Isolated Serial connection using USB Serial of the FTDI chip, though the USB connector
- The emulator drives its power from the USB connector and connection to the controller is isolated using digital isolators.

Hardware resource mapping [30]

The overall system implementing a 3-ph induction motor control is depicted in Fig.5.3. The induction motor is driven by the conventional voltage-source inverter. The TMS320F2803x is being used to generate the six pulse width modulation (PWM) signals using a space vector PWM technique, for six power switching devices in the inverter. Two input currents of the induction motor (ia and ib) are measured from the inverter and they are sent to the TMS320F2803x via two analog-to-digital converters (ADCs). In addition, the DC-bus voltage in the inverter is measured and sent to the TMS320F2803x via an ADC as well. This DC-bus voltage is necessary in order to calculate three phase voltages of induction motor when the switching functions are known.

HVACI Sensor-less project has the following properties:

Development and Emulation	Code Composer Studio V6.0 with real-time debugging
Target Controller	TMS320F2803x or TMS320F28035x
PWM Frequency	10 kHz PWM (Default), 60 kHz PWMDAC
PWM Mode	Symmetrical with a programmable dead band
Interrupts	ADC, end of conversion – Implements 10 kHz ISR execution rate
Peripherals	Used PWM 1 / 2 / 3 for motor control
	PWM 6A, 6B, 7A and 7B for DAC outputs (7A and 7B only 2803x)

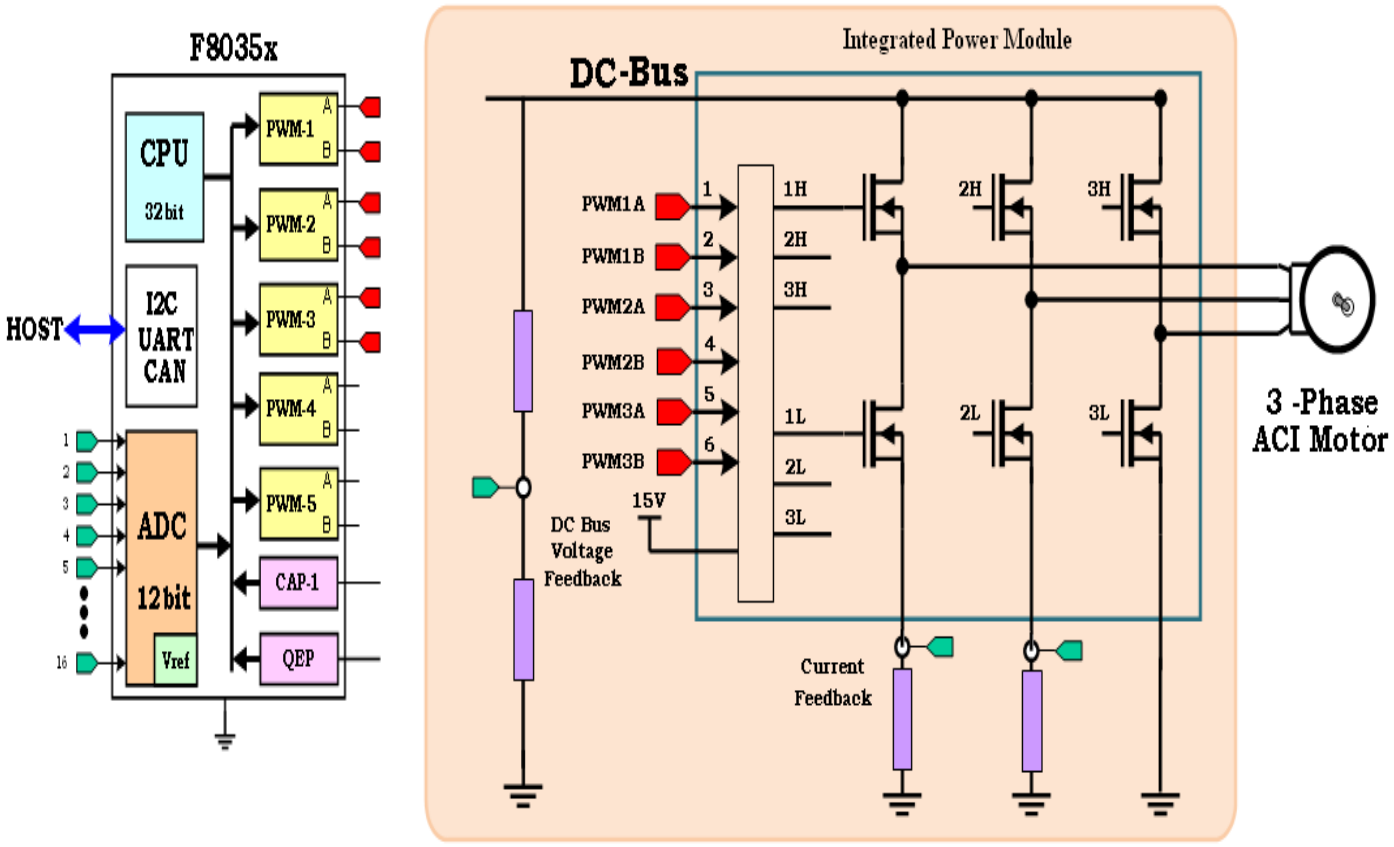


Figure 5.3: A 3-ph induction motor drive implementation with C2000 MCU[29]

Table 5.1: PWM and ADC resource allocation

Macro name	Signal name	PWM /ADC channel no mapping	Function
3-phase inverter	PWM -1H	PWM -1A	Inverter driver PWM
	PWM -1L	PWM -1B	Inverter driver PWM
	PWM -2H	PWM -2A	Inverter driver PWM
	PWM -2L	PWM -2B	Inverter driver PWM
	PWM -3H	PWM -3A	Inverter driver PWM
	PWM -3L	PWM -3B	Inverter driver PWM
	Ifb-U	ADC-B3, A1	Low side U-phase current sense
	Ifb-V	ADC-B5, B1	Low side V-phase current sense
	Ifb-W	ADC-A3, A5	Low side W-phase current sense
	Ifb-Sum	ADC-A2	DC Bus return current sense
	Vfb-Sum	ADC-A7	DC Bus Return voltage sense
	Vfb-U	ADC-B7	U- phase voltage sense
	Vfb-V	ADC-B6	V- phase voltage sense
	Vfb-W	ADC-B4	W- phase voltage sense
2 phase PFC	PWM -1	PWM -4A	PFC phase 1 drive PWM
	PWM -2	PWM -4B	PFC phase 2 drive PWM
	Ipfc	ADC-A6	Return current sense
	Vpfc	ADC-A4	PFC Output voltage sense
	L-fb	ADC-B2	Line voltage sense
	N-fb	ADC-B0	Neutral voltage sense
Main board	DAC -1	PWM -6A	Driving DAC signal
	DAC -2	PWM -6B	Driving DAC signal
	DAC-3	PWM -7A	Driving DAC signal
	DAC -4	PWM -7B	Driving DAC signal

System Algorithm for Rotor flux based sensor-less DFOC speed control of IM

The overall system algorithm is based on the initialization module and interrupt subroutine module.

initialization module

- DSP setup: core, clocks, ADC, GPIO, Event manager, etc.
- Variable initialization.
- Describes enabling the DSP modules to accept interrupt and from where the interrupt is originated.
- Waiting loop: It is a loop which waits for indefinite time length until the interrupt sub module sends the interrupt starting signal.

Interrupt subroutine module description:

- Reading ADC output current value and scaling up it: The stator current sensor output is connected to the ADC input of the digital signal processor. The DSP processor reads this current value from the ADC result register.
- Different coordinate transformations for field oriented control.
- Rotor speed estimation algorithm: It is rotor flux based speed estimator algorithm.
- Speed and current control algorithm: It is Proportional Integral (PI) controller for controlling the speed and the current of the motor.
- SVPWM algorithm: it is an algorithm used to generate a nearly sinusoidal stator current by the inverter using q-axis and d-axis stator voltages.

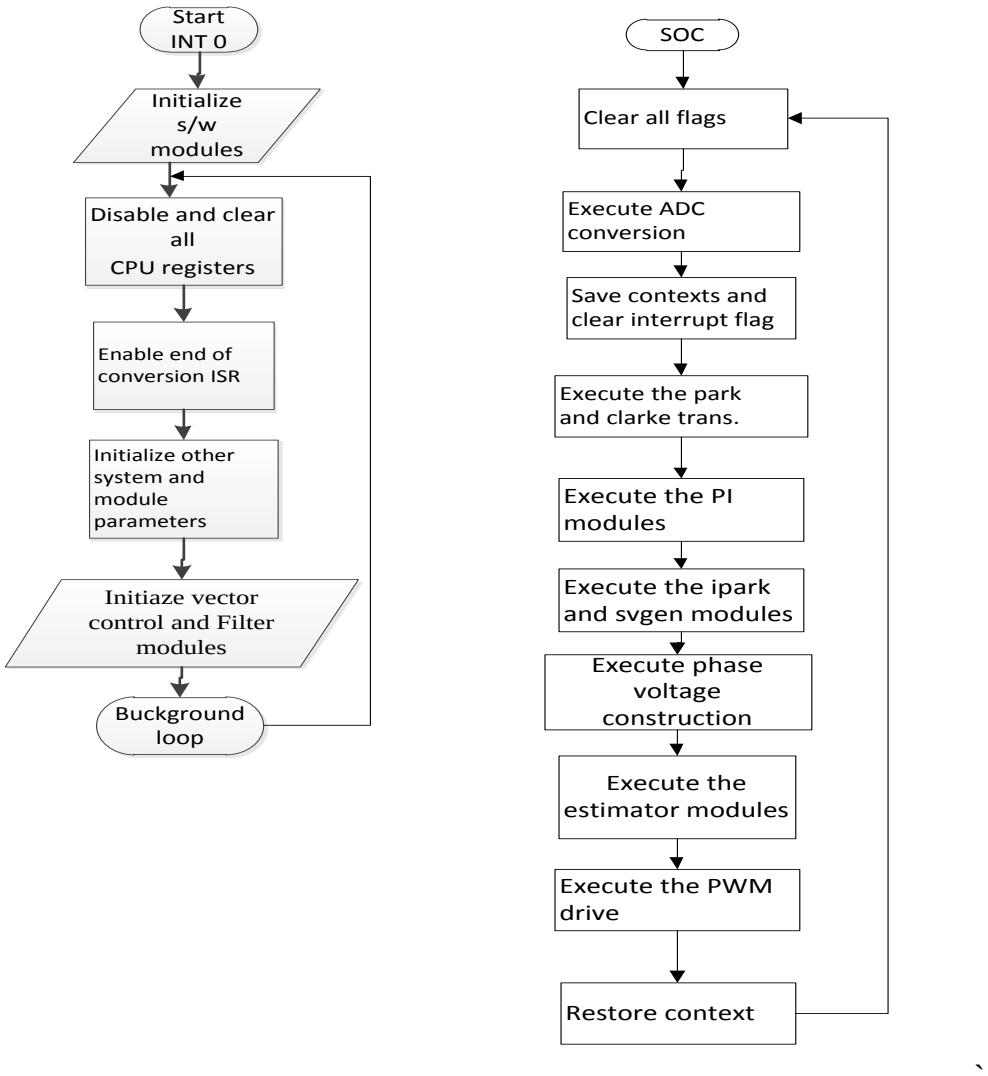


Figure 5.4: Flowchart of the proposed system

Code Composer Studio (CCS)

Code Composer is an integrated development environment (IDE) to develop applications for Texas Instruments (TI) embedded processors. The code development is carried out on an application suite called code composer studio (CCS v6.0) and made by Texas instruments. It supports all their families and variations of DSP. The key feature of the CCS is to serve the whole development chain and supports C and Assembly programming language. In Figure 5.4 the outline of the programming interface is stated. CCS consists of four main parts: project window, program window, watch window, and output window. When choosing the compile command, potential errors will show up in the Output window. The Watch window enables

tracking of variable values while running the program. The CCS communicates with DSP through the computer standard parallel port.

5.3. The DSP

In order to run the real-time control algorithm and create PWM signals, Texas Instruments' (TI) TMS320 processor is used in this work. Texas Instruments' TMS320 family consists of fixed-point, floating-point, multiprocessor digital signal processors (DSPs). TMS320 DSPs have an architecture designed specifically for real-time signal processing. The following characteristics make the TMS320 family a suitable choice for a wide range of processing applications:

- Flexible instruction set,
- Inherent operational flexibility,
- High-speed performance,
- Innovative parallel architecture,
- Cost effectiveness.

5.4. Interface Card

In order to convey information back and forth between the power stage and DSP an interface card has been designed. Moreover, suitable signal amplification, signal filtering and hardware protection properties are added to this interface card. The dc-link voltage is sensed with a voltage sensor (LV25_P) on the interface card.

5.5. Experimental setup

The experimental setup includes the host computer which is used to program the system algorithm through code composer studio v6.0.USB cable is used to connect HVDMCMTRPFC kit with host computer. The source code on the host computer is debugged and loaded to the memory of the piccolo TMS320F28035 control card through this cable. Controller Power comprises of the 15V, 5Vand 3.3V that the board uses to power the control card and the logic and sensing circuit present on the board. This power can be sourced from Auxiliary Power supply module [M2]. Auxiliary Power supply module [M2] can generate 15V and 5V DC from rectified AC.

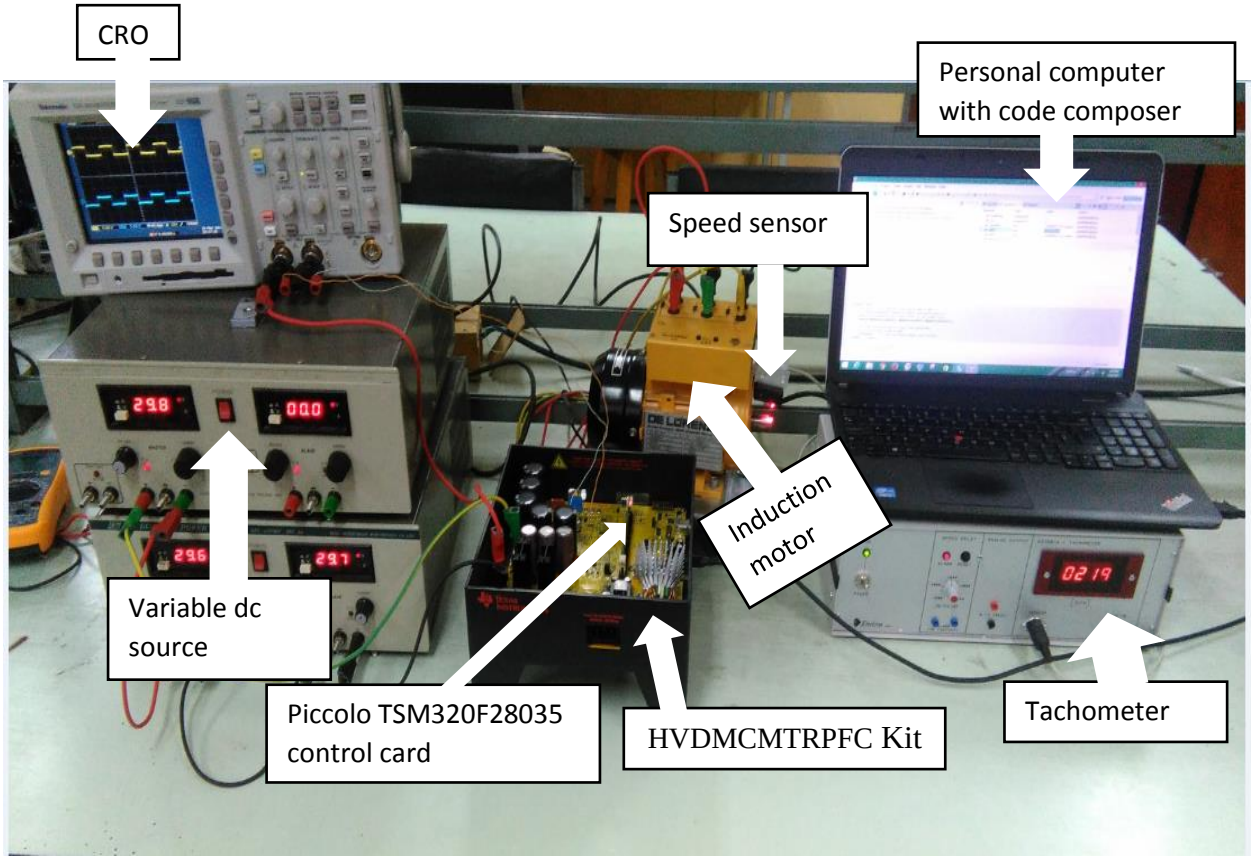
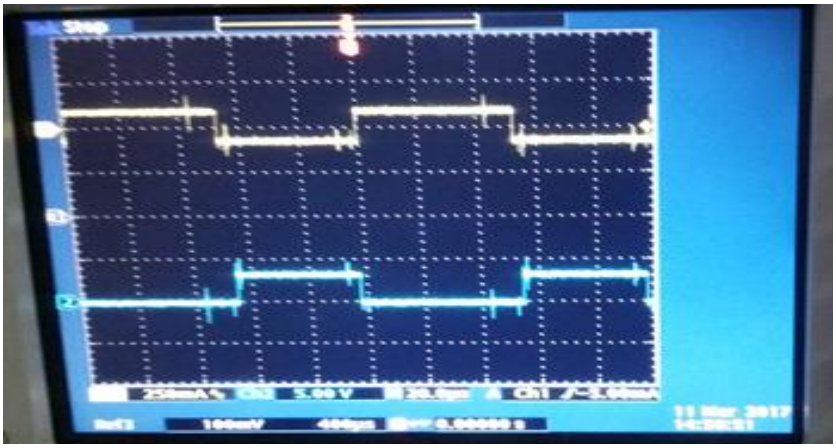


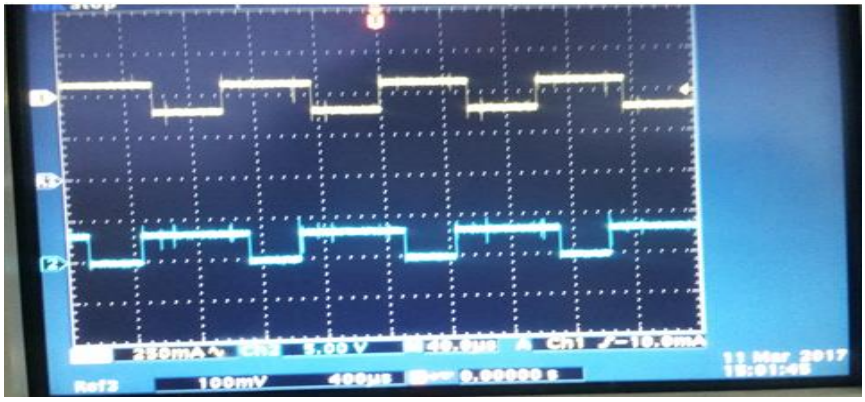
Figure 5.5: Experimental setup of the proposed system

5.6. Experimental Result

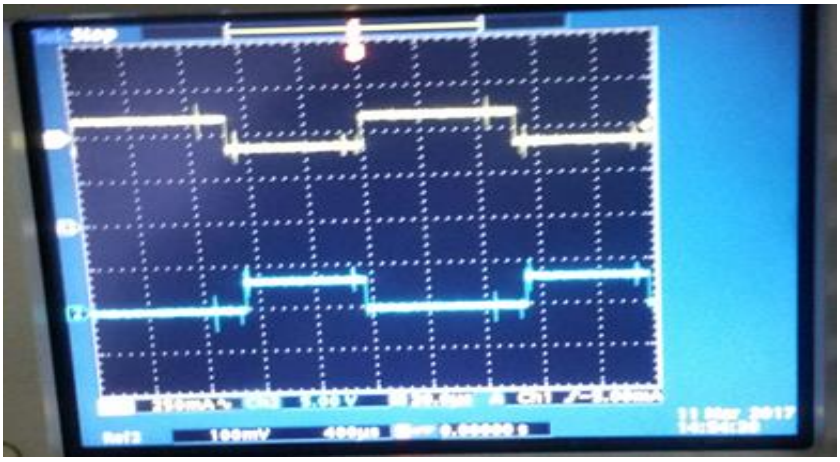
The main target in this experimental investigation is to control the motor at variable speed by generating the PWM signal which generates the appropriate sinusoidal current on the stator of the motor through DSP. This makes the motor to rotate at the required speed and control the motor at different speed including zero speed. The six PWM output signals are shown in Figure 5.6 while the motor is running 0.3pu.



a) PWM_1H_1L



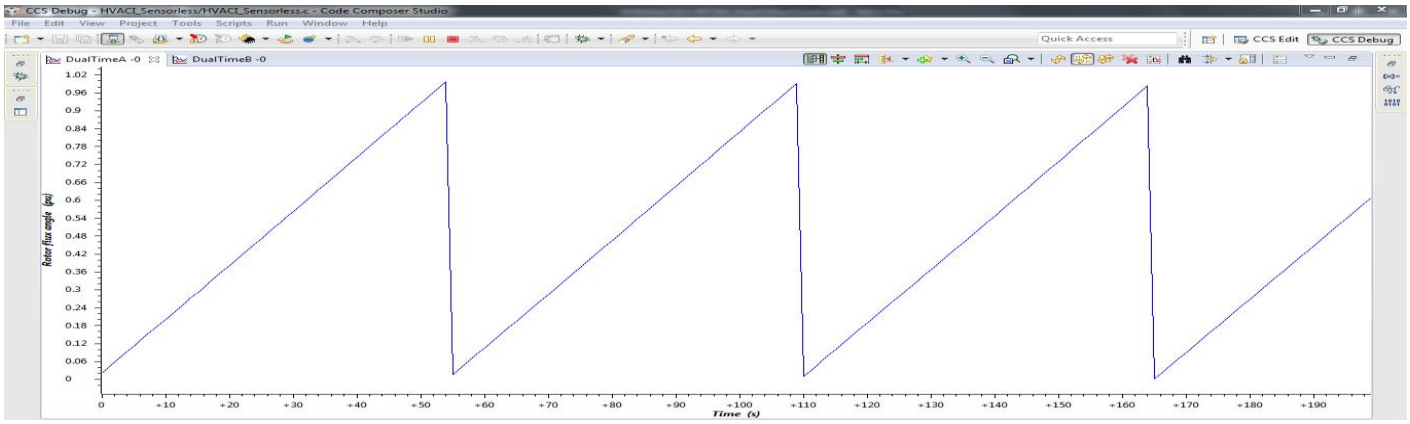
b) PWM_2H_2L



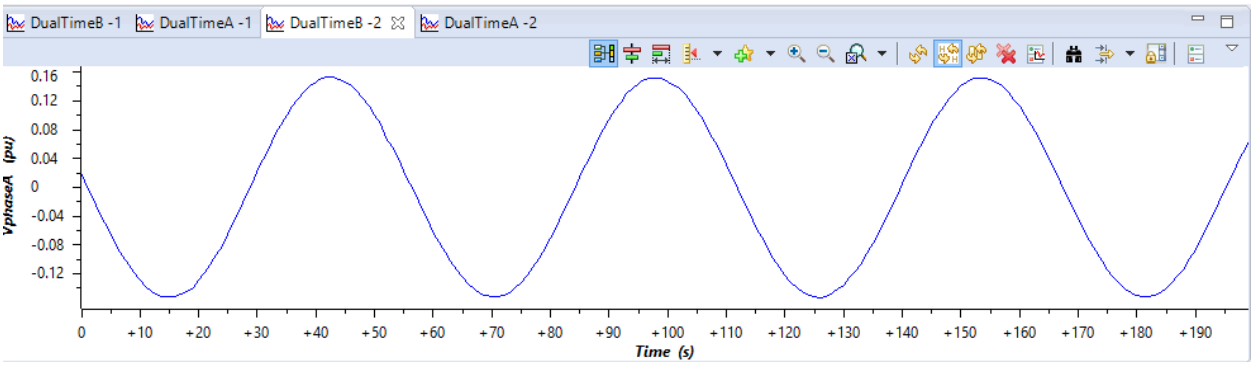
c) PWM_3H_3L

Figure 5.6: the six PWM output signal from the DSP while the motor is running 0.3pu

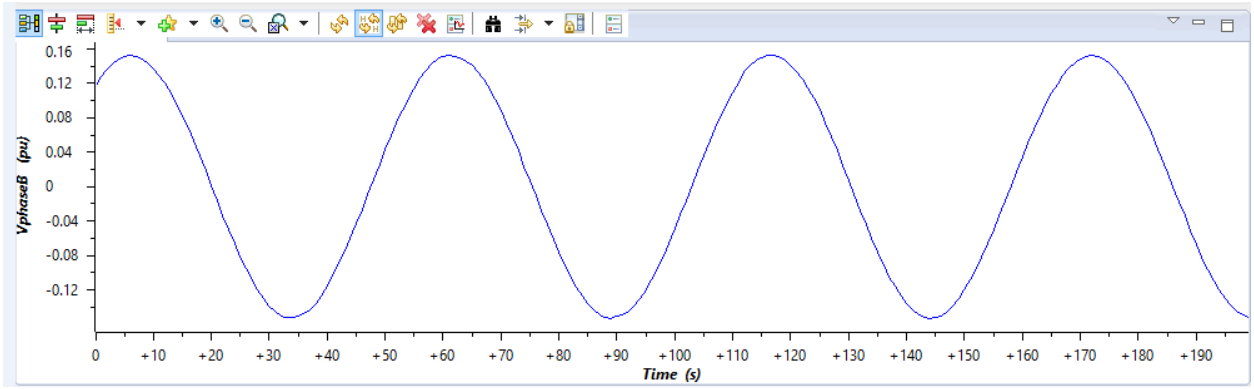
As be seen in Figure 5.7 the estimated angle, stator current and stator voltages have been seen while the motor is running at 0.3pu. From this the estimated angle and the stator current has been good accuracy and almost similar with the simulation result shown in Figure 4.6 of the actual angle and Figure 4.8 of the stator current.



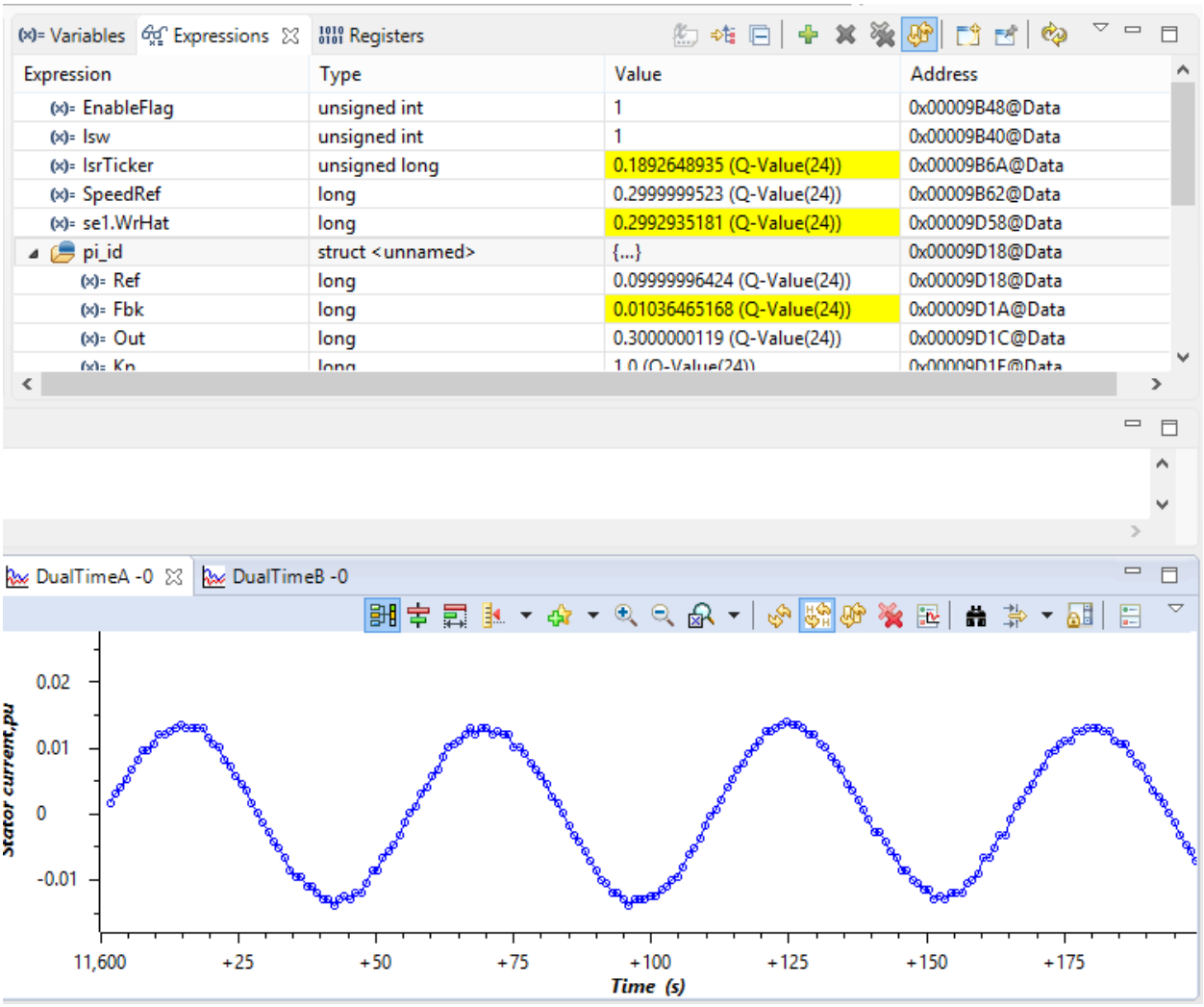
a) rotor position



b) V_alpha



c)V_beta



d) one of the stator current

Figure 5.7: (a)Estimated rotor angle, (b & c) stator voltage in stationary reference frame and (d) stator current while the motor is running at 0.3pu.

It can be seen in Figure 5.8 and Figure 5.9; that the induction motor tracks the desired speeds well in wide ranges, which demonstrates the effectiveness of the proposed scheme. From these the rotor speed has been good accuracy and almost similar with that of the simulation result. The implementation result shows that rotor flux based open loop speed estimator can estimate the rotor speed with 0.000134409 pu of steady state error and good transient performance has been achieved.

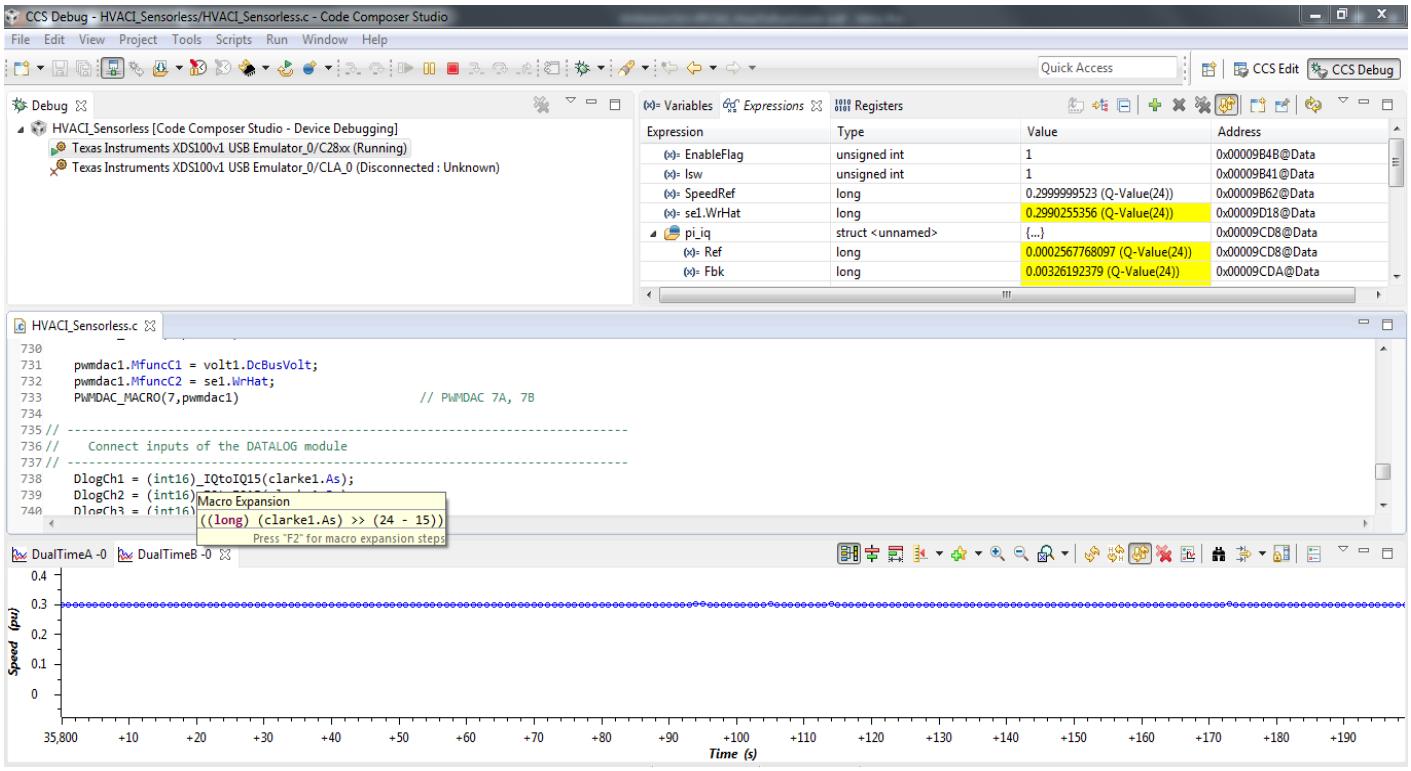


Figure 5.8: rotor speed while the motor is running at 0.3 pu

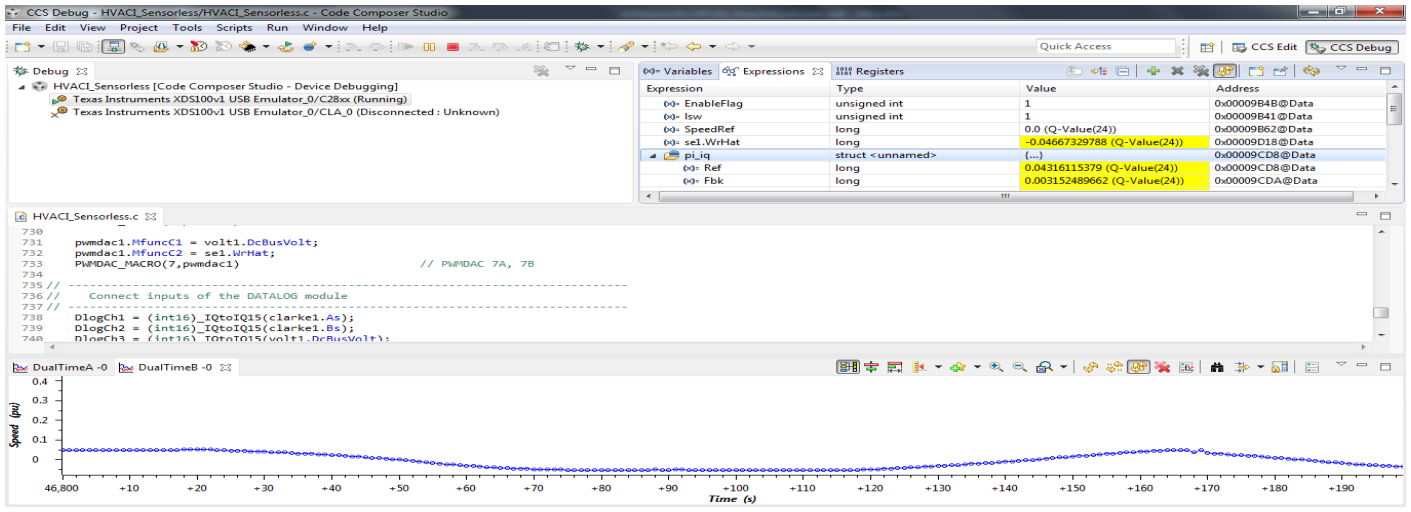
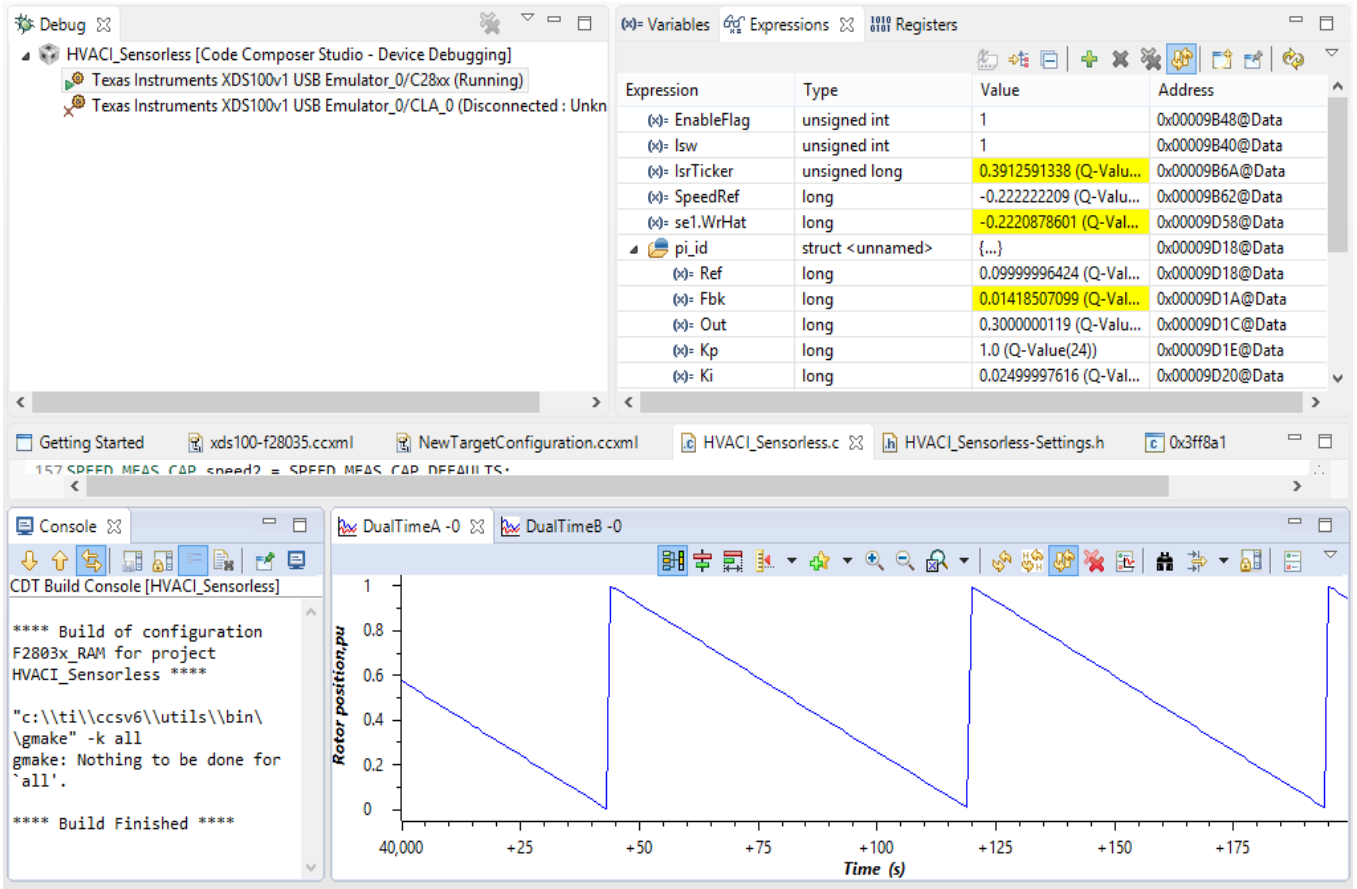


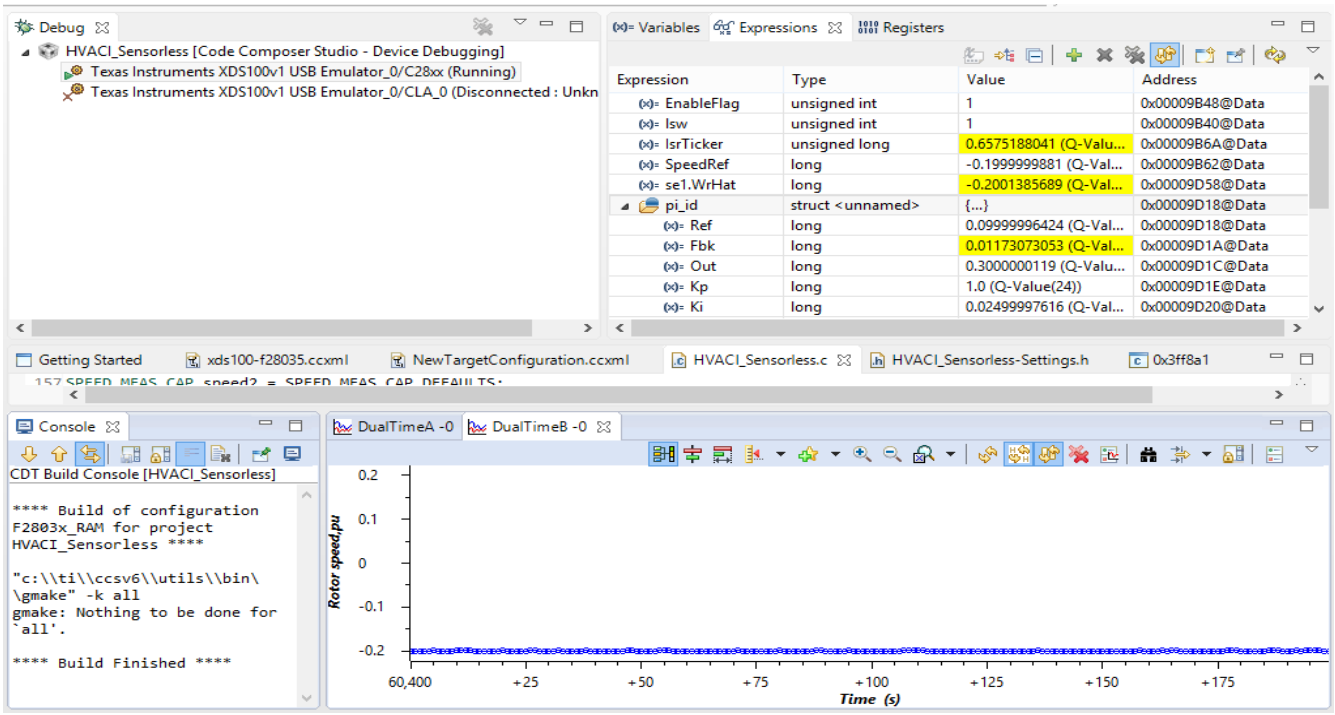
Figure 5.9: Snapshot of CCS programming interface while the motor is running at 0 pu

As can be seen from figure 5.9 above, the snapshot of CCS programming interface with zero pu motor speed is slightly oscillates as expected at low speed. It is not an easy task to control the induction motor at low speed by rotor flux estimation technique.

Finally the estimator is working for regenerative mode of operation. As can be seen in figure 5.8, the drive is working in the regenerative mode of operation with reference speed of -0.2 pu. In this case the motor reverse its direction and the rotor position becomes in the counter clock wise as it is indicated in figure (5.8,a).This is similar with the situation in figure 3.1.The actual speed tracks the reference and becomes -0.2 pu as indicated in figure (5.8 ,b)



a)



b)

Figure 5.10: Snapshot of CCS programming interface while the drive is operating in the regenerative mode and at the speed of -0.2 pu

Chapter Six

Conclusions and Recommendations

6.1. Conclusions

In this thesis, design and experimental implementation of rotor flux based speed estimator alleviate the problem due to mechanical position sensor for speed control of induction motor has been investigated. The computational techniques used to simplify the rotor flux open loop speed estimator design and its implementation on Texas instrument TMS320F2803 control card is presented. Rotor flux based Open loop speed estimator has been found to be well suited to the speed estimation of an induction motor. The simulation results show open loop speed estimator can estimate the speed with good performance for speed control of induction motor with error of about $\pm\ 0.3 rad/sec$.

Proposed estimator estimates the actual speed of the motor with a steady state error of 0.00375 with good transient response with rise time of less than 0.1 second and settling time of 0.15second has been achieved.

The experimental investigation is implemented using Texas Instrument, Piccolo TMS320F28035 control card. The experimental results also demonstrates smooth speed control and maximum steady state error of 0.000134409 pu and good transient response.

6.2. Recommendations

Future Research Suggestions is in the following direction.

1. This simulations and experimental works show the benefits of open loop speed estimator models using rotor flux estimation techniques. Due to the variation of rotor parameters it is not easy to control the speed at very low speed range. But, it is possible to make this analysis simple by using closed loop rotor flux observers that use the estimated stator current error or the estimated stator voltage error to estimate the rotor flux instead.
2. The machine parameters and the rotor time-constant can be estimated and used in the flux observer module and the open-loop control can be improved.

References

[1] Joachim Holtz, "Sensorless Control of Induction Motor Drives, " *proceedings of the IEEE*, vol. 90, no. 8, august 2002.

[2] Ch Chengaiah and Silva Prasad, "Performance of Induction Motor Drive by Indirect vector controlled method using PI and Fuzzy Controllers," *International Journal of Science, Environment*, vol. 2, no. 3, pp. 475-469, 2013.

[3] Masiala, Vafakhah, Knight, and Salmon (2007), "Performances of PI and Fuzzy Logic Speed Control of Field Oriented Induction Machine Drives", *IEEE Transaction.*

[4] D.W.Novotny and T.A.Lipo, "Vector Control and Dynamics of AC Drives", *Oxford University Press Inc., Oxford, New York, 1997.*

[5] Madhavi, and Muley (2013), "Speed Control of Induction Motor using PI and PID Controller", *IOSR Journal of Engineering (IOSRJEN)*, Vol. 3, Issue 5, pp. 25-30.

[6] Benudhar, Mohanty, and Swagat (2010), "A Comparative Study on Fuzzy and PI Speed Controllers for Field-Oriented Induction Motor Drive", *Modern Electric Power Systems, Wroclaw, Poland MEPS'*10.

[7] Ashok Kusagur, S. F. Kodad, and B. V. Sankarram, "Modelling of Induction Motor & Control of Speed Using Hybrid Controller Technology," *Journal of Theoretical and Applied Information Technology*, 2009.

[8] Dheeraj, and Meghna (2013), "Comparison of Vector Control Techniques for Induction Motor Drives", *Indian Journal of Electrical Biomedical Engineer*, Vol. 1.

[9] P.L.Jansen and R.D.Lorenz "Transducer less position and velocity estimation in induction an salient ac machines", *IEEE Tran. IA*, vol. 31, no.2, pp.240-247, Mar/April, 1995 [10] S. Senthilkumar and S. Vijayan, "Simulation of High Performance PID Controller 56 for Induction Motor Speed Control with Mathematical Modeling".

[10] R.Gabriel, W.Leonhard and C.Nordby " Field oriented control of standard AC motor using microprocessor", *IEEE Tran. IA*, vol.16, no.2, pp.186.192, 1980

[11] H.Jun and W.Bin "New integration Algorithms for Estimating Motor Flux over a Wide Speed Range" *IEEE Tran. IE*, vol. 13, no.5, pp.969-977, 1998

[12] P.L.Jansen and R.D.Lorenz "Observer based direct field orientation: Analysis and comparison of alternative methods", *IEEE Tran. IA*, vol. 30, no.4, pp.945-953, Jul/Aug, 1994

[13] G.C.Vershese and S.R.Sanders "Observers for flux estimation in the induction machine" *IEEE Tran. IE*, vol. 35, no.1, pp.85-94, 1998

[14] Shelby Mathew and Bobin K. Mathew, "Direct Torque Control of Induction Motor Using Fuzzy

Logic Controller," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 2, no. 1, pp. 386-394, December 2013.

[15] F. Blaschke, "The Principle of Field Orientation as Applied to the New Trans- vector Closed Loop Control Systems for Rotating Machines," *Siemens Review*, vol. 39, no. 5, pp. 217-220, 1972.

[16] Ashish Chourasia, Vishal Srivastava, Abhishek Choudhary, and Sakshi Praliya, "Comparison study of Vector Control of Induction Motor Using Rotor Flux Estimation by Two Different Methods," *International Journal of Electronic and Electrical Engineering*, vol. 7, no. 3, pp. 201-206, 2014. 23

[17] Andrzej M. Trzynadlowski, "The Field Orientation Principle in Control of Induction Motors", *1st ed. Thomas A. Lipo, Ed. Massachusetts, USA: Kluwer Academic*, 1994.

[18] Rakesh Singh Lodhi and Payal Thakur, "Performance & Comparison Analysis of Indirect Vector Control of Three Phase Induction Motor," *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. 10, pp. 716-724, Octorber 2013.

[19] Andrzej M. Trzynadlowski, "Contorl of Induction Motors", *David J. Irwin, Ed. San Diego, USA: A Harcourt SCience and Technology Company*, 2001.

[20] Bemal K. Bose, "Modren Power Electronics and Ac Drives", *1st ed., Bernard Goodwin, Ed. USA: Prentice Hall, Inc*., 2002.

[21] F.Z.Peng, T.Fukao "Robust Speed Identification for Speed Sensorless Vector Control of Induction Motors" *IEEE Tran. IA* vol. 30, no. 5, pp.1234-1239, Oct.1994

[22] L.Zhen and L. Xu "Sensorless Field Orientation Control of Induction Machines Based on Mutual MRAS Scheme" *IEEE Tran. IE* vol. 45 no. 5, pp. 824-831, October 1998

[23] C.Lascu, I.Boldea, F.Blaabjerg "A Modified Direct Torque Control for Induction Motor Sensorless Drive" *IEEE Tran. IA* vol. 36, no. 1, pp. 122-130, Jan 2000

[24] Hamid Khan, "Field Oriented Control," *Renesas*, 2008.

[25] Ashutosh Mishra and Prashant Choudhary, "Speed Control of An Induction Motor By Using Indirect Vector Control Method," *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 12, pp. 144-150, December 2012.

[26] Utkin, V. I.. "Variable Structure Control Optimization". 1992. Springer-Verlag.

[27] S. Campbell, H. A. Toliyat "DSP-based electromechanical motion control", CRC Press, 2004

[28] "Modern Power Electronics and AC Drives" - Bimal K. Bose.

[29] "TMS320C28x DSP. CPU and Instruction Set Reference Guide, Texas Instruments", *Inc. Dallas*, TX, [Online]. Available: http://www.ti.com.

[30] Bilal Akin, Manish Bhardwaj , "Sensorless Field Oriented Control of 3-Phase Induction Motors" *Texas Instruments, Inc*

Appendix

Appendix A

```
// C-Code for rotor flux based open loop speed estimator

//Description:      Primary system file for Real Implementation of rotor flux based Sensorless
                    Field Orientation Control for Induction Motors
=============================================================================*/
                                  by Teshome Hambissa
=============================================================================*/
#ifndef __ACI_OPEN_LOOP_CONST_H__
#define __ACI_ OPEN_LOOP _CONST_H__

typedef struct      { float32  Rs;            // Input: Stator resistance (ohm)
                float32  Rr;                  // Input: Rotor resistance (ohm)
                float32  Ls;                  // Input: Stator inductance (H)
                float32  Lr;                  // Input: Rotor inductance (H)
                float32  Lm;                  // Input: Magnetizing inductance (H)
                float32  Ib;                  // Input: Base phase current (amp)
         float32  Vb;              // Input: Base phase voltage (volt)
         float32  Ts;              // Input: Sampling period in sec
         float32  Tr;              // Parameter: Rotor time constant
         float32  K1;              // Output: constant using in rotor flux calculation
        float32  K2;               // Output: constant using in rotor flux calculation
                        float32  K3;  // Output: constant using in rotor flux calculation
                        float32  K4;  // Output: constant using in stator current calculation
                        float32  K5;  // Output: constant using in stator current calculation
                        float32  K6;  // Output: constant using in stator current calculation
                        float32  K7;  // Output: constant using in stator current calculation
                        float32  K8;  // Output: constant using in torque calculation
       } ACIOPEN_LOOP_CONST;

/*-----------------------------------------------------------------------------
      Default initalizer for the ACI OPEN_LOOP _CONST object.
-----------------------------------------------------------------------------*/
#define ACIOPEN_LOOP _CONST_DEFAULTS {0,0,0,0, 0,0,0,0, 0,0,0,0,0,0,0,0,0,
}
/*-----------------------------------------------------------------------------
      ACI OPEN_LOOP _CONST MACRO Definition
-----------------------------------------------------------------------------*/
#define ACI OPEN_LOOP _CONST_MACRO(v)                       \
                                                            \
/* Rotor time constant (sec)*/                              \
   v.Tr = v.Lr/v.Rr;                                        \
   v.K1 = v.Tr/(v.Tr+v.Ts);                                 \
   v.K2 = v.Ts/(v.Tr+v.Ts);                                 \
   v.K3 = v.Lm/v.Lr;                                        \
   v.K4 = (v.Ls*v.Lr-v.Lm*v.Lm)/(v.Lr*v.Lm);                \
   v.K5 = v.Ib*v.Rs/v.Vb;                                   \
   v.K6 = v.Vb*v.Ts/(v.Lm*v.Ib);                            \
   v.K7 = v.Lr/v.Lm;                                        \
   v.K8 = (v.Ls*v.Lr-v.Lm*v.Lm)/(v.Lm*v.Lm);

#endif
```

```
/* ==========================================================================
File name:       ACI_OPEN_LOOP.H   (IQ version)
============================================================================*/

#ifndef __ACI_ OPEN_LOOP _H__
#define __ACI_ OPEN_LOOP H__
typedef struct {  _iq  ThetaFlux;      // Output: Rotor flux angle
                  _iq  IQsS;          // Input: Stationary q-axis stator current
                  _iq  IDsS;          // Input: Stationary d-axis stator current
                  _iq  IDsE;          // Variable: Measured current in sync. reference frame
                  _iq  K1;            // Parameter: Constant using in current model
                  _iq  FluxDrE;       // Variable: Rotating d-axis rotor flux (current model)
                  _iq  K2;            // Parameter: Constant using in current model
                  _iq  FluxQrS;       // Variable: Stationary q-axis rotor flux (current model)
                  _iq  FluxDrS;       // Variable: Stationary d-axis rotor flux (current model)
                  _iq  K3;            // Parameter: Constant using in stator flux computation
                  _iq  K4;            // Parameter: Constant using in stator flux computation
                  _iq  FluxDsS;      // Variable: Stationary d-axis stator flux (current model)
                  _iq  FluxQsS;      // Variable: Stationary q-axis stator flux (current model)

         _iq  PsiDsS;      // Variable: Stationary d-axis stator flux (voltage model)
         _iq  Kp;          // Parameter: PI proportionnal gain
         _iq  Error;       // Parameter: Error term
         _iq  UiDsS;       // Variable: Stationary d-axis integral term
         _iq  UCompDsS;     // Variable: Stationary d-axis compensated voltage
         _iq  Ki;           // Parameter: PI integral gain
         _iq  PsiQsS;       // Variable: Stationary q-axis stator flux (voltage model)
         _iq  UiQsS;        // Variable: Stationary q-axis integral term
         _iq  UCompQsS;     // Variable: Stationary q-axis compensated voltage
         _iq  EmfDsS;       // Variable: Stationary d-axis back emf
         _iq  UDsS;         // Input: Stationary d-axis stator voltage
         _iq  K5;           // Parameter: Constant using in back emf computation
         _iq  K6;           // Parameter: Constant using in back emf computation
         _iq  EmfQsS;       // Variable: Stationary q-axis back emf
         _iq  UQsS;         // Input: Stationary q-axis stator voltage
         _iq  K8;           // Parameter: Constant using in rotor flux computation
         _iq  K7;           // Parameter: Constant using in rotor flux computation
         _iq  PsiDrS;       // Output: Stationary d-axis estimated rotor flux
         _iq  PsiQrS;      // Output: Stationary q-axis estimated rotor flux
         _iq  OldEmf;      // Variable: Old back-emf term
         _iq  Sine;        // Variable: Sine term
         _iq  Cosine;      // Variable: Cosine term
                           } ACIFE;

/*-----------------------------------------------------------------------------
      Default initalizer for the ACIFE object.
-----------------------------------------------------------------------------*/
#define ACIOPEN_LOOP_DEFAULTS {  0,0,……..    /*  ThetaFlux  */    \

/*-----------------------------------------------------------------------------
      ACI Flux Estimator MACRO Definition
-----------------------------------------------------------------------------*/
#define ACIOPEN_LOOP_MACRO(v)
/* Calculate Sine and Cosine terms for Park/IPark transformations */     \
```

```
      v.Sine        = _IQsinPU(v.ThetaFlux);
                                                                    \
      v.Cosine = _IQcosPU(v.ThetaFlux);
                                                                    \
/* Park transformation on the measured stator current*/
                                                                    \
      v.IDsE = _IQmpy(v.IQsS,v.Sine);
                                                                    \
      v.IDsE += _IQmpy(v.IDsS,v.Cosine);
                                                                    \
      /* The current model section (Classical Rotor Flux Vector Control Equation)*/
      v.FluxDrE = _IQmpy(v.K1,v.FluxDrE) + _IQmpy(v.K2,v.IDsE);                    \

/* Inverse park transformation on the rotor flux from the current model*/        \
      v.FluxDrS = _IQmpy(v.FluxDrE,v.Cosine);
                                                                    \
      v.FluxQrS = _IQmpy(v.FluxDrE,v.Sine);

/* Compute the stator flux based on the rotor flux from current model*/          \
      v.FluxDsS = _IQmpy(v.K3,v.FluxDrS) + _IQmpy(v.K4,v.IDsS);                    \
      v.FluxQsS = _IQmpy(v.K3,v.FluxQrS) + _IQmpy(v.K4,v.IQsS);                    \

/* Conventional PI controller section */
                                                \
      v.Error =  v.PsiDsS - v.FluxDsS;
                                                \
      v.UCompDsS = _IQmpy(v.Kp,v.Error) + v.UiDsS;
                                                \
      v.UiDsS = _IQmpy(v.Kp,_IQmpy(v.Ki,v.Error)) + v.UiDsS;
                                                \
      v.Error =  v.PsiQsS - v.FluxQsS;
                                                \
      v.UCompQsS = _IQmpy(v.Kp,v.Error) + v.UiQsS;
                                                \
      v.UiQsS = _IQmpy(v.Kp,_IQmpy(v.Ki,v.Error)) + v.UiQsS;                       \

/* Compute the estimated stator flux based on the integral of back emf*/         \
      v.OldEmf = v.EmfDsS;
                                                                    \
      v.EmfDsS = v.UDsS - v.UCompDsS - _IQmpy(v.K5,v.IDsS);
                                                                    \
      v.PsiDsS = v.PsiDsS + _IQdiv2(_IQmpy(v.K6,(v.EmfDsS + v.OldEmf)));           \

      v.OldEmf = v.EmfQsS;
                                                                    \
      v.EmfQsS = v.UQsS - v.UCompQsS - _IQmpy(v.K5,v.IQsS);
                                                                    \
      v.PsiQsS = v.PsiQsS + _IQdiv2(_IQmpy(v.K6,(v.EmfQsS + v.OldEmf)));           \

/* Estimate the rotor flux based on stator flux from the integral of back emf*/  \

      v.PsiDrS = _IQmpy(v.K7,v.PsiDsS) - _IQmpy(v.K8,v.IDsS);
                                                                    \
      v.PsiQrS = _IQmpy(v.K7,v.PsiQsS) - _IQmpy(v.K8,v.IQsS);
```

```
/* Compute the rotor flux angle*/
                    \
      v.ThetaFlux = _IQatan2PU(v.PsiQrS,v.PsiDrS);

#endif // __ACI_OPEN_LOOP_H__
/* ==========================================================================
File name:       ACI_ OPEN_LOOP_SPEED _CONST.H
============================================================================*/
#ifndef __ACI_ OPEN_LOOP_SPEED _CONST_H__
#define __ACI_ OPEN_LOOP_SPEED SE_CONST_H__

typedef struct     { float32  Rr;                  // Input: Rotor resistance (ohm)
                           float32  Lr;            // Input: Rotor inductance (H)
                           float32  Tr;            // Variable: Rotor time constant
                           float32  fb;              // Input: Base electrical frequency (Hz)
                           float32  Wb;              // Variable: Base angular speed (rad/s)
                  float32  fc;              // Input: Cut-off frequency of lowpass filter (Hz)
                  float32  Tc;              // Variable: Time constant (sec)
             float32  Ts;                 // Input: Sampling period in sec
             float32  K1;          // Output: constant using in rotor flux calculation
                        float32  K2;  // Output: constant using in rotor flux calculation
                        float32  K3; // Output: constant using in rotor flux calculation
                        float32  K4;  // Output: constant using in stator current calculation
                         } ACIOPEN_LOOP_SPEED _CONST;

/*-----------------------------------------------------------------------------
      Default initalizer for the ACI OPEN_LOOP_SPEED _CONST object.
-----------------------------------------------------------------------------*/
#define ACIOPEN_LOOP_SPEED _CONST_DEFAULTS {0,0,0,0,0,0,0,0, 0,0,0,0,    \
                          }
/*-----------------------------------------------------------------------------
      ACI_SE_CONST macro definition
-----------------------------------------------------------------------------*/
#define PI 3.14159265358979
#define ACIOPEN_LOOP_SPEED _CONST_MACRO(v)
            /* Rotor time constant (sec) */                                    \
      v.Tr = v.Lr/v.Rr;                                                  \
                                                                         \
/* Lowpass filter time constant (sec) */                                 \
      v.Tc = 1/(2*PI*v.fc);                                              \
                                                                         \
      v.Wb = 2*PI*v.fb;                                                  \
      v.K1 = 1/(v.Wb*v.Tr);                                              \
      v.K2 = 1/(v.fb*v.Ts);                                              \
      v.K3 = v.Tc/(v.Tc+v.Ts);                                           \
      v.K4 = v.Ts/(v.Tc+v.Ts);
#endif

/* ==========================================================================
File name:       ACI_SE.H  (SPEED ESTIMATOR)
============================================================================*/
#ifndef __ACI_OPEN_LOOP_SPEED_H__
#define __ACI_OPEN_LOOP_SPEED _H__

typedef struct {  _iq     IQsS;          // Input: Stationary q-axis stator current
```

```
                    _iq   PsiDrS;        // Input: Stationary d-axis rotor flux
                    _iq   IDsS;          // Input: Stationary d-axis stator current
                    _iq   PsiQrS;       // Input: Stationary q-axis rotor flux
                    _iq   K1;           // Parameter: Constant using in speed computation
                    _iq   SquaredPsi;   // Variable: Squared rotor flux
                    _iq  ThetaFlux;     // Input: Rotor flux angle
                    _iq21 K2;           // Parameter: Constant using in differentiator (Q21) -
                                           independently with global Q
                     _iq  OldThetaFlux;      // Variable: Previous rotor flux angle

                     _iq  K3;           // Parameter: Constant using in low-pass filter
                     _iq21 WPsi;        // Variable: Synchronous rotor flux speed (Q21) -
                                        independently with global Q
                     _iq  K4;           // Parameter: Constant using in low-pass filter
                     _iq        WrHat;              // Output: Estimated speed in per unit
                    Uint32 BaseRpm;   // Parameter: Base rpm speed (Q0) - independently with
                                           global Q
                    int32 WrHatRpm;    // Output: Estimated speed in rpm (Q0) -   independently
                                        with global Q
                     _iq   WSlip;                   // Variable: Slip
                    _iq   WSyn;                 // Variable: Synchronous speed
                          } ACIOPEN_LOOP_SPEED;

/*-----------------------------------------------------------------------------
Default initalizer for the ACIOPEN_LOOP_SPEED object.
-----------------------------------------------------------------------------*/
#define ACIOPEN_LOOP_SPEED _DEFAULTS {  0, 0, 0, 0,_IQ(0.1),0,0, _IQ21(0.1), 0, _IQ(0.1),0,
_IQ(0.1), 0, 3600,0,0,0
             }
#define DIFF_MAX_LIMIT    _IQ(0.80)
#define DIFF_MIN_LIMIT    _IQ(0.20)
/*-----------------------------------------------------------------------------
      ACI Speed Estimator MACRO Definition
-----------------------------------------------------------------------------*/
#define ACIOPEN_LOOP_SPEED _MACRO(v)
                                \
                  /*  Slip computation */
                                        \
      v.SquaredPsi = _IQmpy(v.PsiDrS,v.PsiDrS)+_IQmpy(v.PsiQrS,v.PsiQrS);                \
                                                                                         \
      v.WSlip= _IQmpy(v.K1,(_IQmpy(v.PsiDrS,v.IQsS) - _IQmpy(v.PsiQrS,v.IDsS)));         \
      v.WSlip= _IQdiv(v.WSlip,v.SquaredPsi);
            \
/*    Synchronous speed computation    */
                                                                                         \
      if ((v.ThetaFlux < DIFF_MAX_LIMIT)&(v.ThetaFlux > DIFF_MIN_LIMIT))                 \
/*    Q21 = Q21*(GLOBAL_Q-GLOBAL_Q)*/
                                                                                         \
              v.WSyn = _IQmpy(v.K2,(v.ThetaFlux - v.OldThetaFlux));
                                                                                         \
      else  v.WSyn = v.WPsi;

/* low-pass filter, Q21 = GLOBAL_Q*Q21 + GLOBAL_Q*Q21      */
                                                        \
      v.WPsi = _IQmpy(v.K3,v.WPsi) + _IQmpy(v.K4,v.WSyn);
```

```
/* Q21 = Q21 - GLOBAL_Q */
      v.OldThetaFlux = v.ThetaFlux;
      v.WrHat = v.WPsi - _IQtoIQ21(v.WSlip);
/* Limit the estimated speed between -1 and 1 per-unit */        \
      v.WrHat=_IQsat(v.WrHat,_IQ21(1),_IQ21(-1));
      v.WrHat = _IQ21toIQ(v.WrHat);
/* Q0 = Q0*GLOBAL_Q => _IQXmpy(), X = GLOBAL_Q */
      v.WrHatRpm = _IQmpy(v.BaseRpm,v.WrHat);

#endif // __ACI_SE_H__

/* ==============================================================================
System Name:        HVACI_Sensorless
File Name:          HVACI_Sensorless.C
Description: Primary system file for Real Implementation of rotor flux based Sensorless
                    Field Orientation Control for Induction Motors
============================================================================== */

// Include header files used in the main function

#include "PeripheralHeaderIncludes.h"
#define   MATH_TYPE      IQ_MATH
#include "IQmathLib.h"
#include "HVACI_Sensorless.h"
#include "HVACI_Sensorless-Settings.h"
#include <math.h>
#ifdef FLASH
#pragma CODE_SECTION(MainISR,"ramfuncs");
#pragma CODE_SECTION(OffsetISR,"ramfuncs");
#endif
void main(void)
{
      DeviceInit();// Device Life support & GPIO
// Only used if running from FLASH
// Note that the variable FLASH is defined by the compiler
#ifdef FLASH
// Copy time critical code and Flash setup code to RAM
// The  RamfuncsLoadStart, RamfuncsLoadEnd, and RamfuncsRunStart
// symbols are created by the linker. Refer to the linker files.
      MemCopy(&RamfuncsLoadStart, &RamfuncsLoadEnd, &RamfuncsRunStart);
// Call Flash Initialization to setup flash waitstates
// This function must reside in RAM
      InitFlash(); // Call the flash wrapper init function
#endif //(FLASH)
   // Waiting for enable flag set
   while (EnableFlag==FALSE)
    {
      BackTicker++;
    }
```