Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

Design and Implementation of Direct Torque Control Drive of Three-Phase Induction Motor based on DSP

Submitted by

Tesfaye Meberate

A thesis submitted to Addis Ababa Institute of Technology, School of Graduate Studies, Addis Ababa University in partial fulfillment of the requirement for the Degree of Master of Science in Electrical and Computer Engineering (Control Engineering)

Advisor: Dr. Mengesha Mamo

June 2019

Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

Design and Implementation of Direct Torque Control Drive of Three-Phase Induction Motor based on DSP

Sumitted by
Tesfaye Meberate

APPROVED 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 title "DSP Implementation of Direct Torque Control Drive of Induction Motor using TMDSHVMTRPFCKIT'', is my original work, has not been presented for the fulfillment of a degree in this or any other University and all sources and materials used for the thesis is acknowledged.

Tesfaye Meberate Anteneh

Name Signature

Addis Ababa Institute of Technology

Addis Ababa,Ethiopia June 28,2019

Place Date

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

Dr. Mengesha Mamo

Advisor's Name Signature

Acknowledgment

I want to start expressing my thanks gratefully to Dr. Mengesha Mamo, Assoc. prof., School of Electrical and Computer Engineering because he gave me this thesis title and Continues to advise me up to last of my work.

Last, but not the least I wish to express my gratitude almighty of GOD; he gave me grace, abundantly bless and day-to-day strength from start to the finishing of this work. Without this effort would not have been successful.

Abstract

Direct torque control (DTC) technology is a superior, modern and a new type of AC machine drive control technology developed after vector control technology. In this Thesis, the Direct Torque Control (DTC) scheme for Induction Motor Drive using Space Vector Modulation (SVM) technique is studied. DTC provides excellent properties of regulation, even if without rotational speed feedback. The technology directly controls the instantaneous value of the electromagnetic torque and stator flux-linkage through the control of stator flux linkage and electromagnetic torque in the stationary coordinate system and has the advantages of rapid torque response, simple control structure, and easy digitalization. These quantities are estimated with only stator voltages, stator currents, and stator resistance.

The induction motor has been modeled and simulated in the stationary d-q reference frame and its transient and steady-state characteristics are drawn. The MATLAB Simulink consists of an induction motor mathematical model, a two-level hysteresis comparator for stator flux control, a three-level hysteresis comparator for torque control, a switching table for voltage vector selection, stator flux position identifier, a three-phase voltage source inverter (VSI), flux and torque estimators. The hardware circuit is composed of in addition to the MATLAB Simulink component it has three-phase Induction machine, TMDSHVMTRPFCKIT with other necessary equipment. The switching table is employed for selecting the optimum inverter output voltage vectors so as to attain a fast torque response, low inverter switching frequency, and low harmonic losses in the code. The simulation of DTC schemes (Conventional DTC) has been carried out using MATLAB/SIMULINK and the results are discussed. From estimated speed of simulation result it has good transient and steady state performance with overshoot 1.531% and with approximated steady state error 0.0002. High Voltage Motor Control and PFC Development kit (TMDSHVMTRPFCKIT) with TMS320F28035 Control Card is programmed using code composer studio for hardware implementation of the DTC scheme and the results are discussed. steady state estimated stator flux module has fluxuation from the reference with an error of 2.8% maximum ripple.

Key Words: Direct Torque Control, Code Composer Studio, Flux Hysteresis Controller, Torque Hysteresis Controller, Flux and Torque Estimator, Speed Estimator, Stationary Reference Frame, Voltage Source Inverter, Squirrel Cage Induction Motor.

Table of Content

List of Tables

List of Figures

List of Symbols and Abbreviation

ADC	Analog-Digital Converter
ASD	Adjustable Speed Drives
CCS	Code Composer Studio
DTC	Direct Torque Control
FOC	Field Oriented Control
IGBT	Insulated Gate Bipolar Transistor
IDE	Integrated Development Environment
IM	Induction Motor
IPM	inverter power module
JTAG	Joint Test Action Group
PI	Proportional and Integral
PWM	Pulse Width Modulation
PU	per unit
SVM	Space Vector Modulation
USB	Universal Serial Bus
VSI	Voltage Source Inverte
A_s	stator phase A axis
B_s	stator phase B axis
C_s	stator phase C axis
v_{A_s}	stator phase A axis volage
v_{B_s}	stator phase B axis voltage
v_{C_s}	stator phase C axis voltage
$i_{A_s}(t)$	stator phase As currents
$i_{B_s}(t)$	stator phase Bs currents
$i_{C_s}(t)$	stator phase Cs currents
S_{a1}, S_{a2}	switching states of phase A_s voltage
S_{b1}, S_{b2}	switching states of phase B_s voltage
S_{c1}, S_{c2}	switching states of phase C_s voltage
d_s	direct axis, the stationary reference frame of reference fixed to stator axis
q_s	quadrature axis in the stationary reference frame
d_r	direct axis rotor reference frame

de direct axis synchronous reference frame

qe quadrature synchronous reference frame

q_r quadrature axis rotor reference frame

$v^s_{d_s}$ voltage in the stationary reference frame in direct axis

$v^s_{q_s}$ voltage in the stationary reference frame in quadrature axis

i_s resultant stator current

v_s resultant stator voltage

i^s_{ds} stationary direct axis current

i^s_{qs} stationary quadrature axis current

i_{d_r} current in the rotor reference frame

ψ_s resultant stator flux linkage

ψ_{d_s} direct axis stationary flux linkage

ψ_{qs} quadrature axis stationary flux linkage

θ the angle between stator As axis and de axis

θsl the angle between dr axis and de axis

θs the angle between As axis and resultant current is

θr the angle between the As axis and dr axis

Lm mutual inductance

Lr rotor winding inductance

Ls stator winding inductance

x cross product

* multiplication

Superscript

s stator

r rotor

' referred

Subscript

s stationary reference frame

r rotor reference frame

CHAPTER 1

Introduction

1.1 General Background

In the past, DC motors were used extensively in areas where a variable-speed operation was required since their flux and torque could be controlled easily by the field and armature current [1]. In particular, the separately excited DC motor has been used mainly for applications where there was a requirement of fast response and four-quadrant operation with high performance near zero speed. However, DC motors have certain disadvantages, which are due to the existence of the commutator and the brushes. That is, they require periodic maintenance; they cannot be used in explosive or corrosive environments and they have limited commutator capability under high-speed, high-voltage operational conditions. These problems can be overcome by the application of alternating-current motors, which can have simple and rugged structure, high maintainability and economy; they are also robust and immune to heavy overloading. their small dimension compared with DC motors allows AC motors to be designed with substantially higher output ratings for low weight and low rotating mass.

Variable-speed AC drives have been used in the past to perform relatively undemanding roles in applications which preclude the use of DC motors, either because of the working environment or commutator limits [1]. Because of the high cost of efficient, fast switching frequency static inverters, the lower cost of AC motors has also been a decisive economic factor in multi-motor systems. However, as a result of the progress in the field of power electronics, the continuing trend is towards cheaper and more effective power converters, and single motor AC drives compete favorably on a purely economic basis with the DC drives.

Among the various AC drive systems, those which contain the squirrel cage induction motor have a particular cost advantage [1]. The cage motor is simple and rugged and is one of the cheapest machines available at all power ratings. Owing to their excellent control capabilities, variable speed drives incorporating AC motors and employing modern static converters and torque control can well compete with high-performance.

Vector control techniques incorporating fast microprocessors and Digital Signal Processors have made possible the application of induction-motor drives for high-performance applications. In the past, such control techniques would have not been possible because of the complex hardware and software required to solve the complex control problem. torque control

In AC machines is achieved by controlling the motor currents. In an AC machine, both the phase angle and the magnitude of the current has to be controlled, or in other words, the current vector has to be controlled [1]. From this concept, vector control terminology is discovered. in AC machines the field flux and the spatial angle of armature mmf require external control. In the absence of this control, the spatial angles between the various fields in AC machines vary with the load and yield unwanted oscillating dynamic response. With vector control of AC machines, the torque- and flux-producing current components are decoupled and the transient response characteristics are similar to those of a separately excited AC machine, and the system will adapt to any load disturbances and/or reference value variations as fast as a DC machine. So, Vector control terminology is one of the control techniques of the AC drive; was described previously. and Vector-controlled drives are one particular type of torque-controlled drive. The other type of high-performance torque-controlled drive is the so-called direct-torque-controlled drive.

It is expected that with the rapid developments in the field of microelectronics, torque control of various types of AC machines will become a commonly used technique when, even though high dynamic performance is not required, servo like high performance plays a secondary role to reliability and energy (efficiency) savings [1]. to reach the best efficiency of AC motor drive, many novel techniques of control has been developed in the last few years.

Following the early works of Blaschke and Hasse, and largely due to the pioneering work of Professor Leonhard, vector control of AC machines has become a powerful and frequently adopted technique worldwide [1]. In recent years, numerous important contributions have been made in this field by contributors from many countries. Many industrial companies have marketed various forms of an induction motor and synchronous-motor drives using vector control. At present direct torque-controlled drives are receiving great attention worldwide, although they were first introduced by German and Japanese researchers more than 30 years ago. Presently only one large manufacturer is marketing one form of direct torque-controlled induction motor drive.

1.2 Objective of The Thesis

The main objective is to improve the dynamic performance of Induction Motor Drive by implementing a Space Vector Modulation (SVPWM) based VSI fed induction motor drive.

The overall objectives are:

- To develop the equivalent d-q model of Squirrel cage Induction motor in the stationary reference frame for its control analysis and its closed loop operation.
- To design DTC drives with developed induction motor model in the MATLAB/Simulink.
- To develop the program code in Code Composer Studio (CCS) version 6 using C programming language for direct torque controlling of the motor.
- To Develop and Use the PI speed(sensorless) controller program for the purpose of our system to be a closed loop system.
- To develop the estimated speed of an induction motor from the measured and estimated stator variables of an induction motor.

1.3 Motivation

Motor control is my favorite technical topic in electrical drives and power electronics. One of the reasons I like it because in a lot of other Electrical Engineering topics there is a kind of finite limit in terms of what we do. Whereas, motor control has no limit. During motor control all engineering application come together with such as power electronics, analog-digital converter, digital-analog converter, software's, DSP, dynamics, mechanics and etc. for this reason motor control becomes a most challenging task, we must master it.

The electric drives used in industry are Adjustable Speed Drives and in most of these drives, Induction motors are applied. Induction motors are today the most widely used AC machines due to the advantageous mix of low cost, reliability, and performance. So effective control of IM parameters e.g. speed, torque, and flux are of utmost importance. From the investigation of the control method, it is known that torque control of IM can be achieved according to different techniques ranging from inexpensive Volts/Hz ratio strategy to sophisticated sensorless vector control scheme. But every method has its own disadvantages like losses, need of separate current control loop, coordinate transformation (thus increasing the complexity of the controller), torque and current ripple, etc. So, it is very much necessary to design a controller to obtain an ideal motor drive system which would have high efficiency, low torque ripple, and minimum current distortion.

1.4 Litrature Review

The literature on the control of induction motor drive is very much diversified over various controlling aspects. Because of the non-linearity properties, we can do many types of research;

even if by taking one parameter we can observe different properties along with this parameter. [2] gives an outline of the mathematical modeling of various machines including an induction machine. The various reference frames and important transformations required for transferring the quantities from one reference frame to another has been reviewed. [3] defines the steady state performance of induction motors and subsequently the d-q model in both synchronously rotating reference frame and the stationary reference frame. Then, state space analysis of flux linkages is derived mainly for MATLAB simulation.

Direct vector control and indirect vector control are the two different types of control techniques applied, in which the former determines the magnitude and the position of the rotor flux vector directly using sensors like the Hall Effect sensor, search coil which is a disadvantage. More or less Texas instrument motor control team done the implementation of direct filed oriented and indirect field oriented based on rotor flux either in stationary reference frame or rotor reference frame with speed sensor or speed estimation techniques in the TMDSHVMTRPFC KIT. But direct torque control is different from this drive techniques.

A comparative study on the two most popular control strategies for induction motor drives: Field-Oriented Control (FOC) and DTC have been presented. The comparison is based on various criteria including basic control characteristics, dynamic performance, parameter sensitivity, and implementation complexity [4]. Amongst all control methods, DTC scheme seems to be mainly remarkable because this technique is independent of machine rotor parameters and does not require any speed or position sensors.

1.5 Overview of The Thesis

This thesis studies the implementation of IM drive based on direct torque control system which consists of TMDSHVMTRPFCKIT, 3 Phase Induction motor and with TMS320F28035 control card chip as the core in the kit. Chapter 1**,** it deals with at the starting general introduction of a motor control system and the objective, motivation, a review, and an overview of the Thesis. Chapter 2 deals instantaneous space vector theory and mathematical modeling of an induction motor. Chapter 3 contains the theoretical foundations of direct torque control technology with the flow of DSP software. On this basis, the working principle and control strategy of the DTC system is studied in depth, and an implementation method is presented. Chapter 4 shows the simulation results of DTC drive obtained from MATLAB/SIMULINK platform and discussion of the waveforms obtained from real-time code composer studio graph. Chapter 5 gives a conclusion and directions for future work.

CHAPTER 2

Voltage Source Inverter Fed Induction Motor Modeling

2.1 Introduction

Induction machines are simple, rugged and are considered to be a workhorse of the industry. due to this induction motor control drive dominate the world market. Control of induction motor drive supplied by voltage source inverter instead of three phase source comes from main. This mechanism helps us to control both electromagnetic torque and stator flux-linkage easily and directly. The inverter gives us easily controlled generated PWM signals and the amplitude is controlled by DC link. From this, we construct a phase voltage. we also sense phase current from the lower side of the inverter instead of using other expensive current and voltage sensor.

2.2 Coordinate Transformation

we know that we are going to control a three-phase induction motor. But the control strategy is not based on three-phase quantities. It is based on the d-q model of the machine. Therefore, to understand the vector control principle/direct torque control principle, a good understanding of the d-q modal is mandatory. The machine model can be described by a set of differential equations with time-varying mutual inductance. three phase AC machine can be represented by an equivalent two-phase machine in which ds and qs correspond to the mutually perpendicular axes called direct axis and the quadrature axis of the stator respectively,dr and qr correspond to the mutually perpendicular direct and quadrature axes of the rotor respectively. In stationary reference frame, both ds and qs axes are fixed on the stator, whereas these are rotating at an angle with respect to the rotor in the rotating reference frame. The rotating reference frame may either be fixed on the rotor or it may be rotating at synchronous speed. In a synchronously rotating reference frame with sinusoidal supply, the machine variables appear as dc quantities in steady state condition [2]. If the variables (voltages, currents or flux linkages) associated with the stator winding are replaced by variables associated with fictitious winding rotating with the rotor at synchronous speed, all the time-varying inductances can be eliminated. similarly, time-varying inductances in the voltage equation of an induction machine can be eliminated by transforming the rotor variables to variables associated with the fictitious stationary winding.

2.2.1 Three Phase to Two Phase Transformation

The voltages V_{As},V_{Bs} and V_{Cs} are the voltages of As-Bs-Cs phases respectively. Now assuming that the ds-qs stationary axes are oriented at an angle as shown and the voltages along ds-qs axes to be v^s_{ds} and v^s_{qs} respectively, the three-phase voltages stationary can be transformed to stationary two-phase voltages according to the following equations [2]. It can be understood as transforming the three windings of the induction motor to two windings, as it is shown in figure 2.1 [1].

$$v_{A_s} = v^s_{ds}\cos\theta + v^s_{qs}\sin\theta \tag{2.1}$$

$$v_{B_s} = v^s_{ds}\cos(\theta - 120^0) + v^s_{qs}\sin(\theta - 120^0) \tag{2.2}$$

$$v_{C_s} = v^s_{ds}\cos(\theta + 120^0) + v^s_{qs}\sin(\theta + 120^0) \tag{2.3}$$

The phase voltages in matrix form can be written as:

$$\begin{bmatrix} v_{As} \\ v_{Bs} \\ v_{C_s} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 1 \\ \cos(\theta - 120^0) & \sin(\theta - 120^0) & 1 \\ \cos(\theta + 120^0) & \sin(\theta + 120^0) & 1 \end{bmatrix} \begin{bmatrix} v^s_{ds} \\ v^s_{qs} \\ 0 \end{bmatrix} \tag{2.4}$$

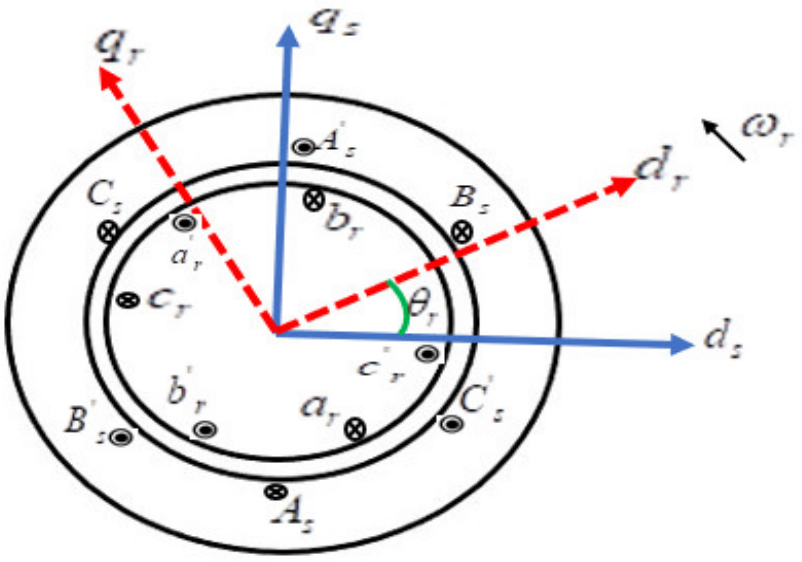


Figure 2.1: Cross section of symmetrical three-phase AC machine

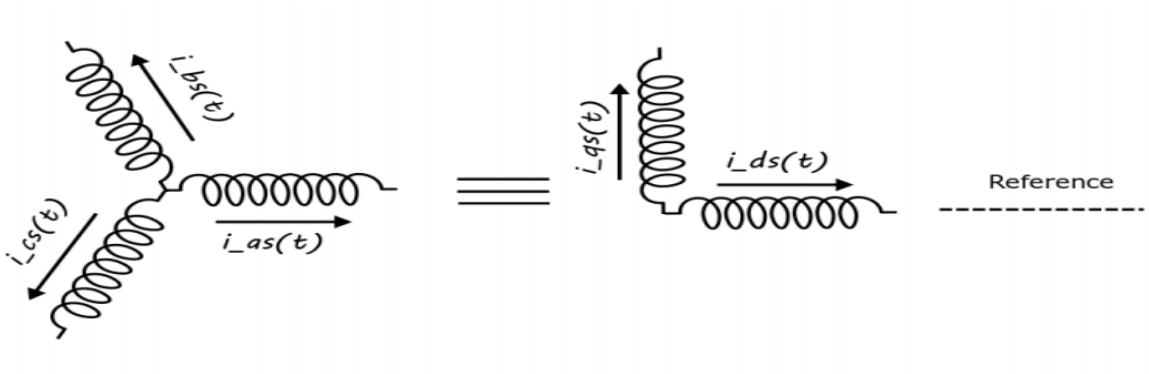


Figure 2.2: abc-to-dq transformation

By inverse transformation, v^s_{ds} and v^s_{qs}can be written in terms of three-phase voltages in matrix form as follows:

$$\begin{bmatrix} v^s_{ds} \\ v^s_{qs} \\ 0 \end{bmatrix} = \frac{2}{3}\begin{bmatrix} \cos\theta & \cos(\theta-120^\circ) & \cos(\theta+120^\circ) \\ \sin\theta & \sin(\theta-120^\circ) & \sin(\theta+120^\circ) \\ 0.5 & 0.5 & 0.5 \end{bmatrix}\begin{bmatrix} v_{A_s} \\ v_{B_s} \\ v_{C_s} \end{bmatrix} \tag{2.5}$$

To simplify the analysis of three-phase quantities from the above transformation we assume that we are in a stationary reference frame and ds the axis is aligned with one of the axes from the three-phase quantities in this case A_s. from this we have θ become zero degrees.

$$\left.\begin{aligned} v_{As} &= v^s_{ds} \\ v_{Bs} &= \frac{-1}{2}v^s_{ds} - \frac{\sqrt{3}}{2}v^s_{qs} \\ v_{cs} &= \frac{-1}{2}v^s_{ds} + \frac{\sqrt{3}}{2}v^s_{qs} \end{aligned}\right\} \tag{2.6}$$

Inversely,

$$\left.\begin{aligned} v^s_{ds} &= v_{As} \\ v^s_{qs} &= \frac{1}{\sqrt{3}}v_{Bs} - \frac{1}{\sqrt{3}}v_{Cs} \end{aligned}\right\} \tag{2.7}$$

2.2.2 Two-Phase Stationary to Two-Phase Synchronously Rotating Frame Transformation

The stationary ds-qs are transformed to synchronously rotating de-qe reference frame which is rotating at speed ω with respect to ds-qs axes with the help of the following figure below [5,1].

The transformation of rotating frame parameters to the stationary frame is according to the following relations:

$$\begin{bmatrix} v^{e}_{d_s} \\ v^{e}_{q_s} \end{bmatrix} = \begin{bmatrix} \cos(\omega t) & \sin(\omega t) \\ -\sin(\omega t) & \cos(\omega t) \end{bmatrix} \begin{bmatrix} v^{s}_{ds} \\ v^{s}_{qs} \end{bmatrix} \tag{2.8}$$

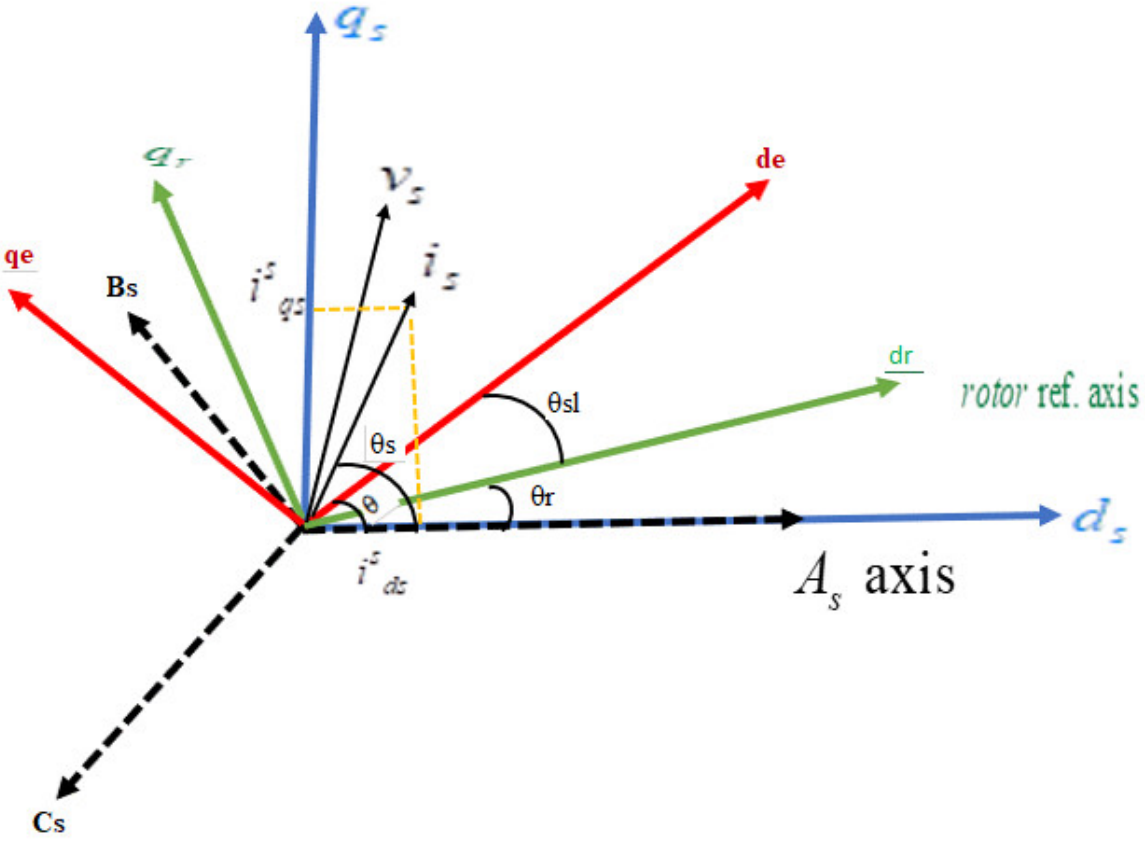


Figure 2.3: The relation of coordinate transformation

2.3 Dynamic Modeling of Induction Motor

Induction machine has nonlinear characteristics by nature. Form previous experience induction motor has complicated formulae to drive the expression of voltage, current, flux-linkage, and instantaneous electromagnetic torque and other expressions. However, by using fundamental physical laws and space vector theory we can easily to show that our expression is simplified and it is easy to in terms of controlling, understanding, designing, programming, and analysis. Due to this, the space phasor of various quantity will be discussed and with the help of this space phasor model of induction machine it gives detail explanation of fundamental principles involved in direct torque control machines [1].

2.3.1 Space Phasor Model of an Induction Machine

Space phasor representation of voltage, current, flux-linkage, etc. are introduced as follows. If the stator winding is supplied by a system of three-phase currents $i_{As}(t)$, $i_{Bs}(t)$ and $i_{Cs}(t)$ which can vary arbitrarily in time [1].

$$i_s(t) = i_{A_s}(t)\cos\theta + i_{B_s}(t)\cos(\theta - 120) + i_{C_s}(t)\cos(\theta - 240) \tag{2.9}$$

$$i_s(t) = \frac{2}{3}[i_{A_s}(t) + a i_{B_s}(t) + a^2 i_{C_s}(t)] \tag{2.10}$$

Based on the above equation,

$$i_s(t) = |i_s(t)| e^{j\theta_s} \tag{2.11}$$

Where $i_s(t)$is the complex space phasor of the three-phase stator currents in the complex plane in the stationary reference frame fixed to the stator. α= $e^{j\frac{2\pi}{3}}$ and $\alpha = e^{j\frac{4\pi}{3}}$. $|i_s(t)|$ is the magnitude of the stator current space phasor, θs is the angle of this current to the stator stationary reference frame fixed to the stator. The voltage and flux linkage also has the same types of expression.

It should be also possible to introduce space phasors utilizing two-axes theory; in ds (direct axis which is aligned with one of the stator phases As) and qs (quadrature axis). So, we can also represent voltage, current, and flux-linkage in two axes theory. thus, the stator-current space vector in the stationary reference frame fixed to the stator can be expressed in two-axis theory as follows [1]:

$$i_s = i_{ds} + j i_{qs} \tag{2.12}$$

These two-phase currents id_s and iq_s are related to the three-phase stator current as follows:

$$\left.\begin{aligned} i_{q_s} &= k\left[\frac{\sqrt{3}}{2}(i_{B_s} - i_{C_s})\right] \\ i_{d_s} &= k\left[(i_{A_s} - \frac{1}{2} i_{B_s} - \frac{1}{2} i_{C_s})\right] \end{aligned}\right\} \tag{2.13}$$

where k = $\frac{2}{3}$. similarly, to the definition of stator space phasor phase currents, it is possible to define the space phasor of the stator flux-linkage in three phases as well as in two-axis theory (two phases). The expression of the total instantaneous value of flux-linkage for the three-phase windings [6]:

$$\psi_s(t) = \frac{2}{3}[\psi_{A_s}(t) + a\psi_{B_s}(t) + a^2\psi_{C_s}(t)] \tag{2.14}$$

Stator flux-linkage in two-phase (direct and quadrature components of flux-linkage)

$$\psi_s = \psi_{ds} + j\psi_{qs} = |\psi_s| e^{i\rho_s} \tag{2.15}$$

ρs is the phase angle of total flux-linkage with respect to the real axis of the stationary reference frame. Stator voltage space phasor refer to stator stationary reference frame can be expressed similarly as above like current and flux-linkage.

$$v_s(t) = \frac{2}{3}[v_{A_s}(t) + av_{B_s}(t) + a^2 v_{C_s}(t)] = v_{d_s}(t) + jv_{q_s}(t) \tag{2.16}$$

The relationship between three-phase and two-phase voltages immediately follows from the equation 2.16.

$$v_{ds} = \frac{2}{3}(v_{As} - \frac{1}{2}v_{Bs} - \frac{1}{2}v_{Cs}) \tag{2.17}$$

$$v_{qs} = \frac{1}{\sqrt{3}}(v_{Bs} - v_{Cs}) \tag{2.18}$$

All the above analysis holds also for rotor components.

2.3.2 Induction Machine Equations

For simplicity we consider smooth air-gap, three-phase winding, sinusoidally distributed mmf in the winding, resistance, and reactance are taken to be constant, etc. the stator voltage equation of induction machine by using three-phase variables can be written as follows:

$$\left.\begin{aligned} v_{As}(t) &= R_s i_{As}(t) + \frac{d\Psi_{As}(t)}{dt} \\ v_{Bs}(t) &= R_s i_{Bs}(t) + \frac{d\Psi_{Bs}(t)}{dt} \\ v_{Cs}(t) &= R_s i_{Cs}(t) + \frac{d\Psi_{Cs}(t)}{dt} \end{aligned}\right\} \tag{2.19}$$

The general expression can be written as,

$$v_s(t) = R_s i_s(t) + \frac{d\Psi_s(t)}{dt} \tag{2.20}$$

The most two common reference frames used in the analysis of the induction machine are stationary reference frame and synchronously rotating reference frame. We discussed how the

transformation from stationary to synchronous and vice versa. The voltage equation in two-phase expressed as follows based on the figure 2.4 and figure 2.5 [7]. all the parameters are in the stator reference frame.

$$\left.\begin{aligned} v^{s}_{ds}(t) &= R_s i_{ds}(t) + \frac{d\Psi_{ds}(t)}{dt} \\ v^{s}_{qs}(t) &= R_s i_{qs}(t) + \frac{d\Psi_{qs}(t)}{dt} \end{aligned}\right\} \tag{2.21}$$

$$\left.\begin{aligned} v_{d_r} &= R_r i_{d_r}(t) + \frac{d\psi_{d_r}(t)}{dt} + \omega_r \psi_{q_r} = 0 \\ v_{q_r} &= R_r i_{q_r}(t) + \frac{d\psi_{q_r}(t)}{dt} - \omega_r \psi_{d_r} = 0 \end{aligned}\right\} \tag{2.22}$$

The flux linkages equations referred to the stationary reference frame [6],

$$\left.\begin{aligned} \psi_{ds} &= L_s i_{ds} + L_m i_{dr} \\ \psi_{qs} &= L_s i_{qs} + L_m i_{qr} \\ \psi_{dr} &= L_r i_{dr} + L_m i_{ds} \\ \psi_{qr} &= L_s i_{qr} + L_m i_{qs} \end{aligned}\right\} \tag{2.23}$$

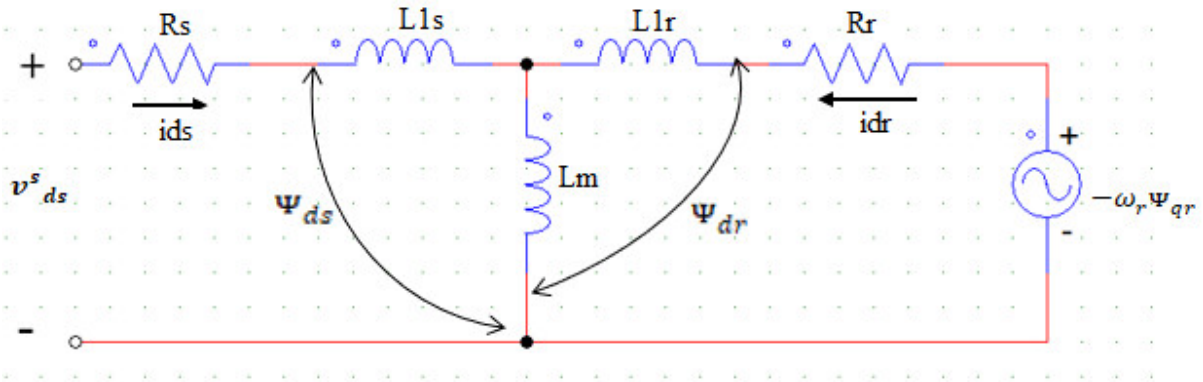


Figure 2.4: Equivalent circuit in direct axes referred to the stationary reference frame

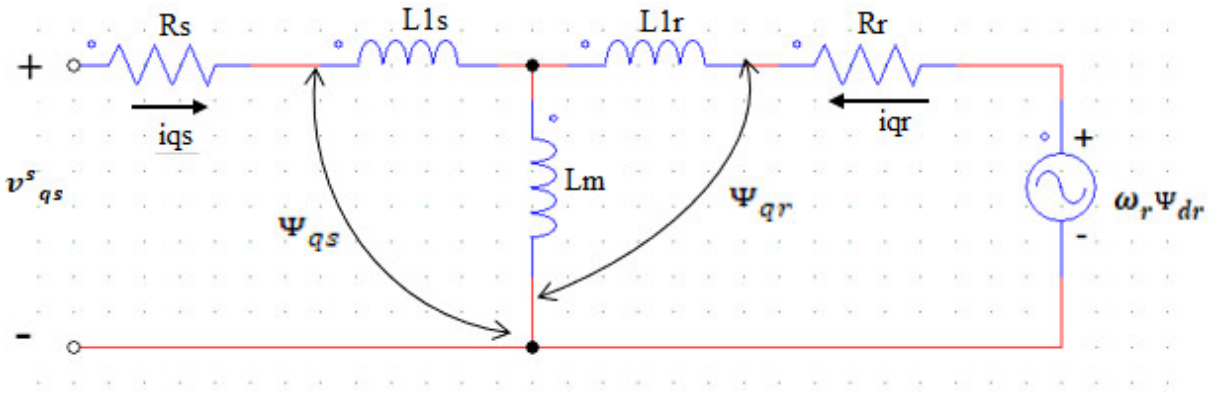


Figure 2.5: Equivalent circuit in quadrature axes referred to the stationary reference frame

Finally, the developed electromagnetic torque can be expressed in several ways. general torque expression of AC machine can be written as:

$$T_e = k(\psi_s \times i'_r) \tag{2.24}$$

Where k is constant, i'_r the rotor current referred to the stator at stationary reference frame and Ψ'_r is the resultant rotor flux refered to the stator. Recalling from the equation 2.23, we have stator and rotor flux-linkage space vector equation in the stationary reference frame fixed to the stator [6].

$$\left.\begin{aligned} \psi_s &= L_s i_s + L_m i'_r \\ \psi'_r &= L_r i'_r + L_m i_s \end{aligned}\right\} \tag{2.25}$$

The power conversion takes place in the induction machine can be written as follows [1]:

$$P_{elec} = P_{loss} + P_{mech} + P_{field} \tag{2.26}$$

p_{elec}input electrical energy.

p_{loss}stator and rotor loss of energy.

p_{mec}developed mechanical energy.

$p_{field}$the stored magnetic energy in the field

$$P_{mech} = P_{elec} - P_{loss} - P_{field} \tag{2.27}$$

Replace equation 2.27 In Machin parameter forms in the stationary reference frame fixed to the stator.

$$T_e \omega_r = \frac{3}{2} R_e\{[(v_s i_s^*) + (v'_r i'^*_r)] - [R_s i_s^2 + R_r i'^2_r] - [\frac{d\psi_s}{dt} i_s^* + \frac{d\psi_r}{dt} i'^*_r]\} \tag{2.28}$$

By considering some assumption in stationary reference frame we can put in most simplified from;

$$T_e\omega_r = -\frac{3}{2}\omega_r(\psi'_r \times i'_r) \tag{2.29}$$

$$T_e = -\frac{3}{2}(\frac{p}{2}\psi'_r \times i'_r) \tag{2.30}$$

Using this equation and above derived expression of $\Psi'_r$and i'_r, we get many expressions of torque in a different frame of reference either fixed to the stator or rotor using three phases quantitates or two-phase quantities. Here is the list of expressions.

$$T_e = \frac{3}{2}\frac{p}{2}(\psi_s \times i_s) \tag{2.31}$$

$$T_e = \frac{3}{2}\frac{p}{2}(\psi_{d_s} i_{q_s} - \psi_{q_s} i_{d_s}) \tag{2.32}$$

$$T_e = -\frac{3}{2}\frac{p}{2}(\frac{L_m}{L_s L_r - L_m^2}\psi_s \times \psi'_r) \tag{2.33}$$

$$T_e = -\frac{3}{2}\frac{p}{2}(\frac{L_m}{L_s L_r - L_m^2}\psi_s \psi'_r \sin(\theta_{sr})) \tag{2.34}$$

2.4 Voltage Source Inverter

According to the type of AC output waveform, the power converter topologies can be classified into voltage source inverters (VSIs) and current source inverters (CSIs). VSI is the most widely used semiconductor power converters in ASDs and many industrial applications [3].

A basic three phase, two levels, VSI is a six-step bridge inverter, consisting of minimum six power electronics switches (i.e. IGBTs, MOSFETs) and six feedback diodes connected antiparallel to switches. A step can be defined as the change in firing from one switch to the next switch in the proper sequence. For a six-step inverter, each step is of 60^0 intervals for one cycle of 360^0. The switches would be gated at regular intervals of 60^0 in the proper sequence to get a three-phase ac output voltage. Figure 2.6 shows the power circuit diagram of three phase VSI. The capacitor connected into the input terminals is to maintain the input dc voltage constant and this also suppresses the harmonics fed back to the dc source [8]. Three-phase loads are star connected.

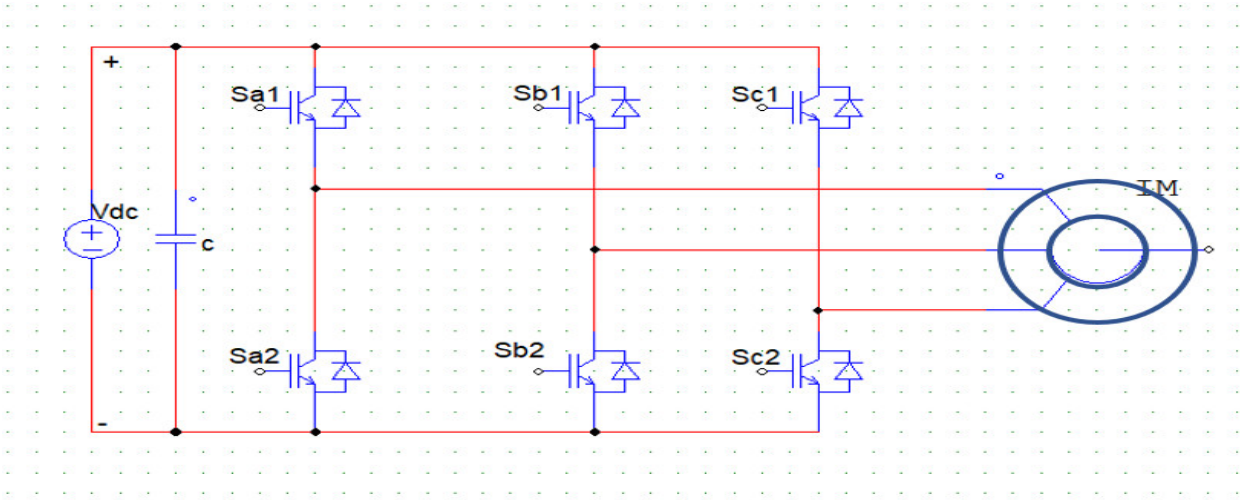


Figure 2.6: Three-phase voltage source inverter

The six switches are divided into two groups; upper three switches as a positive group (i.e. S_{a1}, S_{b1}, S_{c1}) and lower three as a negative group of switches (i.e. S_{a2}, S_{b2}, S_{c2}). The standard three-phase VSI topology has eight valid switching states which are given in Table 2.1. Of the eight valid switching states, two are zero voltage states (0 and 7 in Table 2.1) which produce zero ac line voltages and, in this case, the ac line currents freewheel through either the upper or lower components [8]. The remaining states (1 to 6 in Table 2.1) are active states which produce non-zero ac output voltages. From these six active states, we get the space vector phase voltage corresponding to the inverter in each switching states can be obtained will be discussed in the next chapter. The inverter moves from one state to another in order to generate a given voltage waveform. Thus, the resulting ac output line voltages consist of discrete values of voltages [8].

states	**S_{a1}**	**S_{b1}**	**S_{c1}**	**Voltage Vector**
0	0	0	0	v_0
1	1	0	0	v_1
2	1	1	0	v_2
3	0	1	0	v_3
4	0	1	1	v_4
5	0	0	1	v_5
6	1	0	1	v_6
7	1	1	1	v_7

Table 2.1: Switching states of a three-phase VSI

S_a has value either 0 (lower switch) or 1 (higher switch). For the 1 value S_a reparented by S_{a1} and for 0 value it is S_{a2}. The other switches S_b and S_c are following the same procedure.

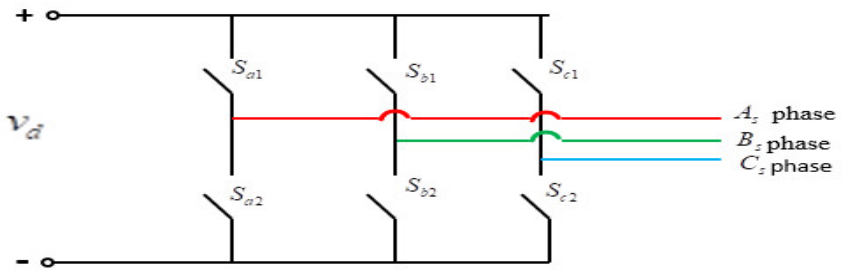


Figure 2.7: Ideal voltage source inverter

Generally, A voltage-type inverter (see Figure 2.7) consists of three groups and six switches Composition. There are eight possible switch combinations for the three groups of switches. It can be stated that when the " + " pole is connected, the switching state of the phase is " 1"; conversely, when the "-" pole is connected, the switching state is " 0 ". The six active voltage vectors can be expressed as follows:

$$v_s = v_d e^{j(k-1)\frac{\pi}{3}} \tag{2.35}$$

Where k=1,2,3,4,5,6. The above voltage can be in stationary reference fixed to the stator. this can be shown below in the figure 2.8. Each sector contains the voltage vector in the middle [1].

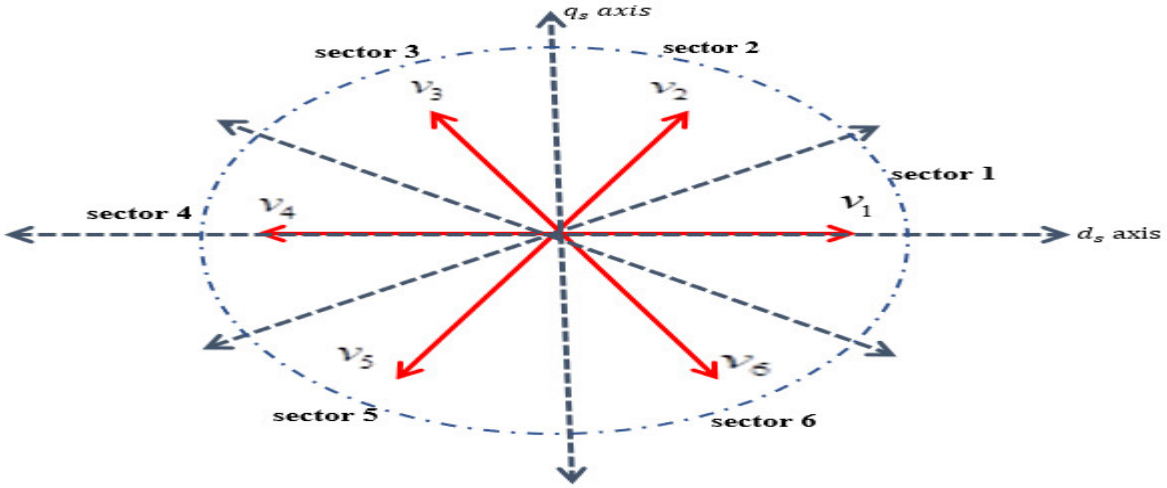


Figure 2.8: Space vectors of inverter output voltage and sectors

Sector 1	Sector 2	Sector 3	Sector 4	Sector 5	Sector 6
$-30^0<s(1)\leq 30^0$	$30^0<s(2)\leq 90^0$	$90^0<s(3)\leq 150^0$	$150^0<s(4)\leq 210^0$	$210^0<s(5)\leq 270^0$	$270^0<s(6)\leq 330^0$

Table 2.2: The position of flux linkage (angle)

CHAPTER 3

Direct Torque Control and DSP Software Development Strategy

3.1 Introduction

The control methods for induction motors can be divided into two parts: vector control and scalar control strategies. Many applications require that the speed of the motor be variable without changing the torque. The speed of induction motors can be changed in three different ways [9,10]:

- changing the number of pole pairs
- vary the magnitude of the supplied voltage or
- vary the frequency of the supplied voltage.

Changing the number of pole pairs causes huge curve shifting, and varying the magnitude of the supplied voltage gives a solution only in a small range. The most promising possibility to change the speed of the motor is varying of the frequency. However, this is not the best solution, because the impedances increase and the currents decrease with frequency. This offers a possibility of control through both the voltage and the frequency changing. This idea was the basis of a control method which is capable of controlling both speed and torque of induction motors, the so-called Volts/Hertz constant method [8]. The scalar control method is based on varying two parameters simultaneously. The speed can be varied by increasing or decreasing the supply frequency, but this results in a change of impedances. The change of impedance leads to an increase or decrease in current. If the current is small, the torque of the motor decreases. If the frequency decreases or the voltage increases, the coils can be burned or saturation can occur in the iron of coils. To avoid these problems, it is necessary to vary the frequency and the voltage at the same time. According to the equation of induced voltage, the V/Hz constant control gives constant flux in the stator

$$v_s = R_s i_s + E_{bemf} \tag{3.1}$$

$$E_{bemf} = -\frac{d\Psi}{dt} = kBf \tag{3.2}$$

B is flux density proportional to torque.

f is frequency, k is constant.

E_{bemf} back electromotive force.

At higher speed

$$v_s = kBf \tag{3.3}$$

Torque-speed control can be solved by the linear variation of the two parameters.

$$B \propto \frac{v_s}{f}$$

Scalar controlled drives give a somewhat inferior performance, but easy to implement. Their importance has been diminished recently because of the superior performance of vector-controlled drives which is demanded in many applications. Field-oriented control (FOC) is one of the vector control methods in which the stator currents of a three-phase AC electric motor are identified as two orthogonal components. One component defines the magnetic flux of the motor, the other the torque. FOC is classified as Direct Field Oriented Control (DFOC) and Indirect Field Oriented Control (IFOC). The FOC method has an attractive feature but it suffers from some drawbacks, such as the requirement of coordinate transformations, current controllers, sensitive to parameter variations, PWM modulators, switching frequency, rotor position measurement, and control tuning loops. The use of speed encoder is associated with some drawbacks, such as the requirement of shaft extension, reduction of mechanical robustness of the motor drive, reduces the drive reliability and not suitable for hostile environments, and also costlier. The speed control system of the drive calculates the corresponding current references from the flux and torque references. Proportional-integral (PI) controllers are used to keep the measured current components at their reference values.

The high-performance frequency controlled PWM inverter fed IM drive should be characterized by:

1. quick flux and torque response
2. available maximum output torque in the wide range of speed operation region
3. constant switching frequency
4. unipolar voltage PWM
5. low flux ripple and low torque ripple
6. robustness for parameter variation

These features depend on the applied control strategy. The main goal of the chosen a control method is to provide simplicity (simple algorithm, simple tuning, and operation with small controller dimension leads to the low price of the final product).

The key to high-performance AC motor drive is the instantaneous torque control. Direct torque control (DTC) is another torque control method following vector control. Vector control is to control the torque by controlling the decoupled current. DTC directly focuses on the control of the torque, and this “direct control” concept is not only used for torque control, but also for the flux chain control [11].

Compared with vector control, direct torque control has the following main features:

1. Analyze the mathematical model of the AC motor directly in the stator coordinate system. Based on this, control the motor's flux linkage and torque. It eliminates complicated transformations and calculations such as vector rotation transformations. Therefore, the signal processing required by it is particularly simple and the control signals used to make it possible for the observer to make a direct and explicit determination of the physical process of the AC motor.
2. The stator flux linkage is used for the magnetic field orientation. As long as the stator resistance is known, it can be observed. This greatly reduces the control performance in vector control technology and is susceptible to parameter changes.
3. The concept of space vector was used to analyze the mathematical model of the three-phase AC motor and control its physical quantity, making the problem particularly simple and clear.
4. The control structure is simple and easy to implement all-digital.

At present, DTC is divided into two different schemes in terms of magnetic flux trajectory: the hexagonal scheme and a circular scheme. Because the induction motor is powered by a three-phase symmetrical sine wave, the air gap magnetic potential of the motor is circular. At this time, the motor loss, torque ripple, and noise are the minima. Therefore, in the medium and low power applications, we are tending to use the circular flux trajectory scheme; The hexagonal scheme is only considered in some high-power areas (switching frequency and switching losses have large limits) not discuss in this Thesis.

3.2 Conventional DTC

The conventional DTC scheme is a closed loop control scheme, the important elements of the control structure being: the power supply circuit, a three-phase voltage source inverter, the induction motor, the speed controller to generate the torque command and the DTC controller. The DTC controller again consists of torque and flux estimation block, two hysteresis

controllers and sector selection block; the output of the DTC controller is the gating pulses for the inverter. A typical DTC system control principle is shown in Figure 3.1 [1,6,7,8].

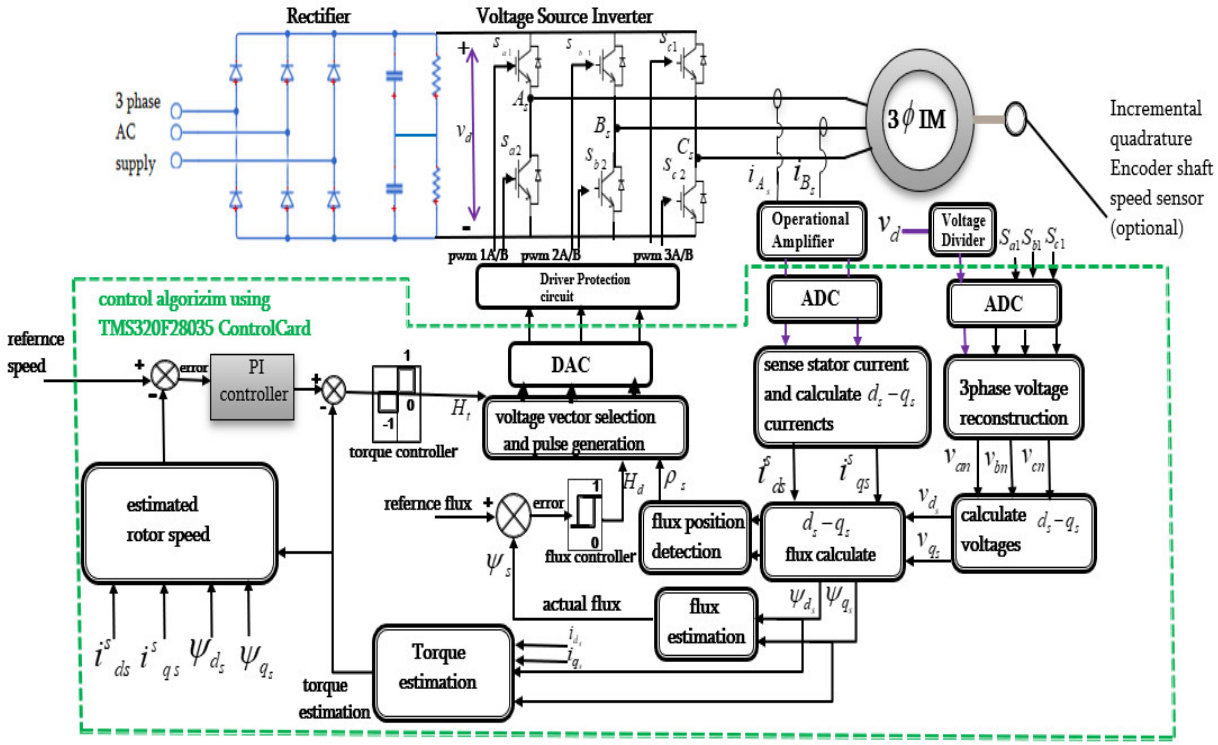


Figure 3.1: Block diagram of the implemented DTC scheme for IM drives

In addition to the above main feature, DTC scheme does not require coordinate transformation and as all the control procedures are carried out in the stationary frame of reference. So, this scheme does not suffer from parameter variations to the extent that other control techniques do. Also, there is no feedback current control loop due to which the control actions do not suffer from the delays inherent in the current controllers, no pulse width modulator, no PI controllers, and no rotor speed or position sensor. So, it is a sensorless control technique which operates the motor without requiring a shaft mounted mechanical sensor.

The flux loop controller has two levels of digital output and the torque control loop has three levels of digital output. The feedback flux and torque are calculated from the induction machine terminal voltages and currents. The three-phase terminals quantities are converted into two phases stationary ds-qs components, which are used for estimating motor torque and stator linked flux. Based on the resultant flux position and the errors in flux magnitude and in torque, a three-dimensional look-up table is referred to decide the inverter switching.

3.2.1 Analysis of The Working Principle of DTC System

As already mentioned in chapter 3 equation (3.24-3.34), from many other equations the produced electromagnetic torque in stationary reference frame referred to the stator described as follows:

$$T_e = \frac{3}{2}\frac{p}{2}(\psi_s \times i_s) \tag{3.4}$$

$$T_e = -\frac{3}{2}\frac{p}{2}\left(\frac{L_m}{L_s L_r - L_m^2}\psi_s \psi'_r \sin(\theta_{sr})\right) \tag{3.5}$$

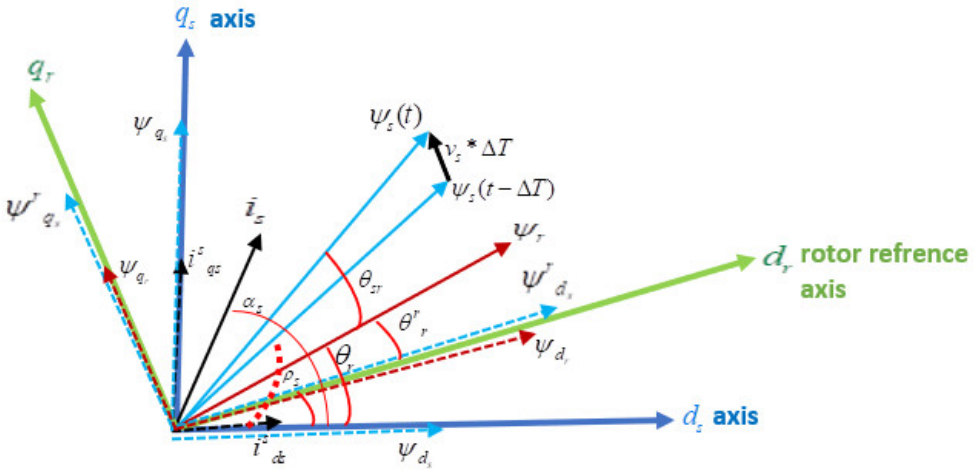


Figure 3.2: The relationship of stator and rotor flux linkage referred to a stationary frame

From the figure 3.2 [7], the angle between stator and rotor flux linkage in the stationary reference frame is $\theta_{sr} = \rho_s - \theta_r$(this angle is in torque equation 3.5). also, from the equation (3.5) the electromagnetic torque developed by the motor depends on the angle θ_{sr} . So, the torque can be controlled by changing the position of stator flux with respect rotor flux as from figure 3.2 and equation (3.5). this figure also used for the rotor speed estimation techniques. The details will discuss in the future.

3.2.2 Stator Flux Estimation

Rotor flux linkages Ψ_r cannot be changed quickly because of the large time constant Tr and also because of the presence of leakage reactance. Hence it can be considered as a constant, compared to stator flux linkages Ψ_s which change in a given sample period. Stator

flux linkages can be changed quickly by changing the applied stator voltage vector in the equation below.

$$\frac{d\psi_s(t)}{dt} = v_s(t) - R_s i_s(t) \tag{3.6}$$

To simplify the analysis and to see the core effect of stator flux linkage by stator voltage neglecting the stator resistance effect. Then we get the following expression:

$$\left.\begin{aligned} \frac{d\psi_s(t)}{dt} &\approx v_s(t) \\ \Delta\psi_s(t) &\approx v_s(t)\Delta t \end{aligned}\right\} \tag{3.7}$$

$\boldsymbol{\Delta\Psi_s(t)}$is the change of flux linkage below and above the reference flux linkage. From this equation the change of stator flux linkage space vector directly affected by the stator voltage space vector [1]. This also, shows that the direction of motion of the stator flux linkage is the same as the direction of motion of the voltage space vector at this time. If the voltage space vector is a vector, the stator flux is in a spatially stationary state. This voltage also can be expressed by two-phase axes in the stationary reference frame (ds-qs) and the suitable voltage obtained from the inverter.

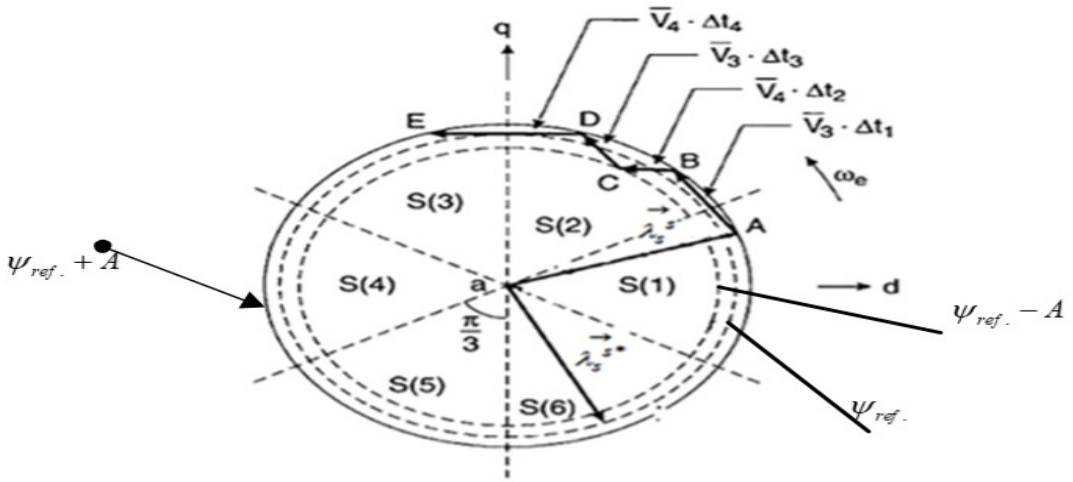


Figure 3.3: Circular trajectory of stator flux

Therefore, through proper selection of the working vectors, the magnetic flux can form a quasi-circular trajectory in space, thus achieving the purpose of maintaining the magnetic flux amplitude as a constant value. The stator ds-qs axes flux linkage is given by equation (2.21).

$$\psi_{d_s}(t) = \int [v_{d_s}(t) - R_s i_{d_s}(t)]dt \tag{3.8}$$

$$\psi_{q_s}(t) = \int [v_{q_s}(t) - R_s i_{q_s}(t)]dt \tag{3.9}$$

The resultant estimated and the position of stator flux linkage respectively given by

$$\psi_s(t) = \sqrt{(\psi_{d_s}(t))^2 + (\psi_{q_s}(t))^2} \tag{3.10}$$

$$\rho_s = a\tan 2\frac{\psi_{q_s}(t)}{\psi_{d_s}(t)} \tag{3.11}$$

3.2.3 Selection of Voltage Space Vector

We must select appropriate voltages from inverter switches in order to control induction motor properly. Torque can be controlled by controlling the angle θ_{sr}between stator flux linkages and rotor flux linkages. If an increase of the torque is required, then apply the voltage vectors which advance the flux linkage space vector in the direction of rotation and if a decrease is required, voltage vectors are applied which oppose the direction of the torque and if zero torque is required then zero switching vector is applied which minimizes the inverter switching.

By assuming the stator flux space vector lies in the kth sector(k=1,2,3,4,5,6), we have 60^0 separate sectors and 6 active voltage vectors, its magnitude can be increased by using voltage vectors v_k, v_{k+1}, v_{k-1},however it is decreased by $v_{k+2}, v_{k+3}, v_{k-2}$.this voltage also affects the torque [1].

3.2.4 Hysteresis Controller

In both torque and flux control, the control of both uses a simple based on a bang-bang controller. The inverter switch state is directly determined by the controller output; the digital values of flux and torque and space where the flux is located. The drive performance is influenced by the width of the hysteresis bands in terms of flux and torque ripples and switching frequency of power electronics devices. In each sampling time, the switching state of the inverter is updated.

Flux hysteresis controller

It is based on two level controllers; on-off controller. So that it has two digital outputs Hd (0, and 1). The flux error $\Delta\Psi_s$ is the input to the hysteresis controller; the output from the comparator, then we get the digital output based on the hysteresis controller. In this case, the flux deviates at some value from the fixed reference value of the flux we call this deviation $\Delta\Psi_s$. In specific flux position, we want to either increase the flux or decrease it. To increase the flux, we make it Hd=1, to decrease the flux make it Hd=0.

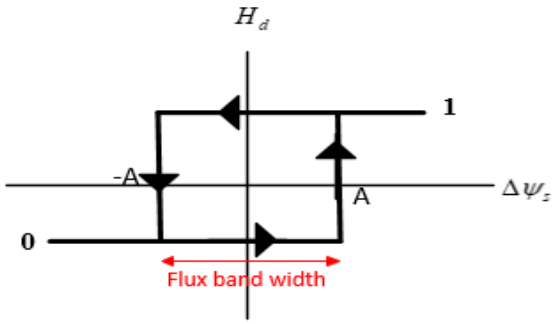


Figure 3.4: Characteristics of flux hysteresis controller

$\Delta\Psi_s$= flux reference – estimated flux=$\Psi_{ref} - \Psi_{estimated}$. Also, note that $\Psi_{estimated} = \Psi_s$. $\Delta\Psi_s$is the flux error shown in figure 3.4 [12]. we have two important points point A (the upper band) and point -A (lower band). The value of upper band point A is $\Psi_{ref} * 0.01$; for this Thesis flux hysteresis comparator design. This value is the flux bandwidth. By comparing this value with that of error flux ($\Delta\Psi_s$) then generate Hd (either 1 or 0) digital values.

Hd=1 for $\Psi_s \leq \Psi_{ref-A}$

Hd=0 for $\Psi_s \geq \Psi_{ref-A}$

When we generalize in pseudo algorism form as follow.

```
if (error flux (ΔΨs) >upper band limit(A))
{
Return 1;
}
else if (error flux <lower band limit)
{
```

```
Return 0;
}
else Return last out;
```

Torque hysteresis controller

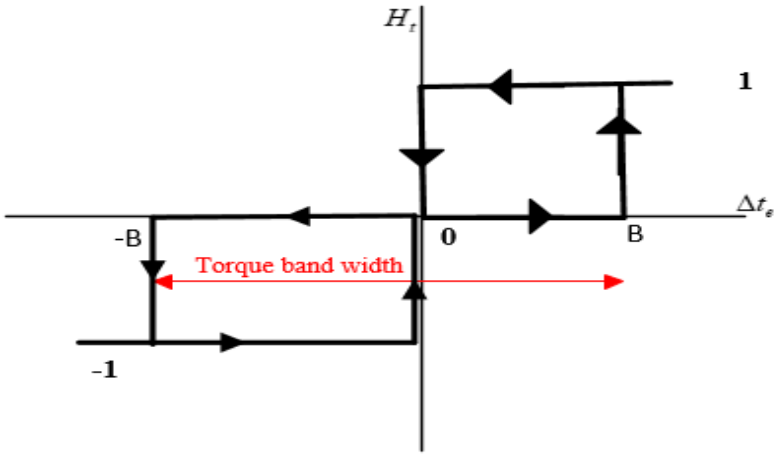


Figure 3.5: Characteristics of torque hysteresis controller

$\Delta t_e = T_{refernce} - T_{actual}$, T_{actual} is estimated torque based on stator flux and stator currents. $T_{refernce}$ is the output of PI controller and Δt_e is the torque error status as shown in figure 3.5 [12] . If we want to increase the torque Ht=1; torque decrease is required Ht=-1 and if no change in torque required Ht=0. So, the torque hysteresis controller has three digitals outputs. The torque hysteresis comparator has three digital output its implementation method is more complex than that of flux hysteresis implementation. The upper band of torque has value B equals to 0.01 for this Thesis design case. The pseudo code for torque hysteresis comparator is written as follows.

```
if (T_actual <T_refernce -B)
{
Return 1;
}
else if (T_actual =T_refernce)
{
Return 0;
}
else (T_actual >T_refernce+B) {Return -1;}
```

3.2.5 Formation of Switching Table

The sectors of the stator flux space vector are denoted from s(1) to s(6) . combining Hd, Ht and ρ_s we can select the optimum voltage vector and then fed into the invertor of high voltage motor control KIT; to realize the switching states of the inverter.

H_d	H_t	$s(1)$	$s(2)$	$s(3)$	$s(4)$	$s(5)$	$s(6)$
1	**1**	v_2	v_3	v_4	v_5	v_6	v_1
1	**0**	v_7	v_0	v_7	v_0	v_7	v_0
1	**-1**	v_6	v_1	v_2	v_3	v_4	v_5
0	**1**	v_3	v_4	v_5	v_6	v_1	v_2
0	**0**	v_0	v_7	v_0	v_7	v_0	v_7
0	**-1**	v_5	v_6	v_1	v_2	v_3	v_4

$v_1(100);\ v_2(110);\ v_3(010);\ v_4(011);\ v_5(001);\ v_6(101);\ v_7(111);\ v_0(000)$

Table 3.1: Switching vector selection table

3.3 DSP and Software Development Environment

A digital signal processor (DSP) chip is a high-speed microprocessor. Texas Instruments (TI) introduced the TMS320F28035 (C280) chip, which not only inherited the advantages of fast DSP chips and real-time control but also used F280 (C280) as the core chip because it is facing in the field of digital motor control.

In the currently available motor control systems, many technologies are incorporated. The processing power of DSP and the availability of built-in modules like ADC, PWM, QEP input, SPI, I2C, and high-speed UART make the TI's TMS320F28035 digital signal processor the best choice as the master processor. The DSP code is developed using C language in Code Composer Studio IDE. The constructed control system has many advantages such as simple structure, high reliability, etc. It is highly favored and will be widely used.

3.4 DSP Peripheral Interface

The DSP processor is interfaced with many peripherals either internally in the DSP or other external devices, circuits to assist the motor control operations, to perform the test procedure and to observe output results during experiments.

3.5 DSP Processor Circuits

3.5.1 TMS320F28035 Piccolo™ Microcontroller

The DSP processor selected is a 32 bit highly advanced microcontroller with internal clock speed up to 60MHz (16.6ns cycle time), 128KB flash, programmable CLA. The F2803x Piccolo™ family of microcontrollers provides the power of the C28x core and Control Law Accelerator (CLA) coupled with highly integrated control peripherals in low pin-count devices. It supports 16x16 or 32x32 Multiply-Accumulate (MAC) operations. It supports advanced peripheral features and can support up to 14 PWM outputs, 1 Quadrature Encoder Inputs, 2 SPI modules, one Inter-Integrated Circuit (I2C) bus, 16 channels of 12bit ADC and JTAG Boundary Scan Support for emulators. It is also supported by Code Composer Studio IDE. Code efficient software development in an assembly as well as in compiler is supported. It is having extensive power saving modes also.

3.5.2 ADC Interface

The ADC module consists of 16 ADC channels of 12-bit ADC with a built-in sample-and-hold (S/H) circuit. Analog input voltage range is 0.0 V to 3.0 V. The fast conversion rate up to 60MHz ADC clock is supported and it results 12-bit for high resolution. The various sensors connected to the machine are analog in nature and the values are to be fed after converting it to digital values. The ADC inputs of the DSP processor are used directly by configuring it for this purpose. The ADC block is used with a very precise 2.048 V reference voltage and it is generated on-board using different IC regulator. Signal conditioning is also provided at the ADC block internally to avoid the interferences. The digital values generated for different analog inputs are,

Digital value=0,when input is 0 volt

Digital value=4095, when input is 3 volt

Digital value=4095 $\frac{input\ analog\ value-ADCLO}{3}$, when input is between 0 volt and 3 volt and ADCLO pin of microcontroller.

To obtain the specified accuracy of the ADC, proper board layout is very critical. To achieve the best results the analog input pins should not run in close proximity to the digital signal paths. Filtering mechanisms are also used to minimize switching noise on the digital lines from getting coupled to the ADC inputs. Furthermore, proper isolation techniques must be used to isolate the ADC module power pins from the digital supply. The analog and digital sections

are provided with isolated ground points. The following program codes configure the ADC input of the tms320f28035 DSP.

```
Void ADCinitialization ()
{
//end AdcChannel Select
AdcRegs.ADCSOC0CTL.bit.CHSEL= ch_no;
AdcRegs.ADCSOC1CTL.bit.CHSEL= ch_no;
AdcRegs.ADCSOC2CTL.bit.CHSEL= ch_no;
AdcRegs.ADCSOC3CTL.bit.CHSEL= ch_no;
AdcRegs.ADCSOC4CTL.bit.CHSEL= ch_no;
AdcRegs.ADCSOC5CTL.bit.CHSEL= ch_no;
// set ADC sample window to the desired value
AdcRegs.ADCSOC0CTL.bit.ACQPS= ACQPS_Value;
AdcRegs.ADCSOC1CTL.bit.ACQPS= ACQPS_Value;
AdcRegs.ADCSOC2CTL.bit.ACQPS= ACQPS_Value;
AdcRegs.ADCSOC3CTL.bit.ACQPS= ACQPS_Value;
AdcRegs.ADCSOC4CTL.bit.ACQPS= ACQPS_Value;
// Setup each SOC's ADCINT trigger source //ADCINT2 starts SOC0-7
AdcRegs.ADCINTSOCSEL1.bit.SOC0= 2;
AdcRegs.ADCINTSOCSEL1.bit.SOC1= 2;
AdcRegs.ADCINTSOCSEL1.bit.SOC2= 2;
AdcRegs.ADCINTSOCSEL1.bit.SOC3= 2;
AdcRegs.ADCINTSOCSEL1.bit.SOC4= 2;
AdcRegs.ADCINTSOCSEL1.bit.SOC5= 2;
```

3.5.3 PWM Output

Each ePWM module of TMS320F28035 DSP processor operates based on dedicated 16-bit time base counter with period and frequency control. It is having programmable phase-control support for lag or lead operation relative to other ePWM modules. The phase control is hardware locked and phase relationship based on a cycle-by-cycle basis. Dead-band generation is with independent rising and falling edge delay control. All events can trigger both CPU interrupts and ADC start of conversion (SOC) to operate in real time and programmable event prescaling minimizes CPU overhead on interrupts.

The PWM outputs are used to generate the test signals for the drives. Pulses with various frequencies and duty cycles are generated and are used for the testing of the drives with a different motor application. Essentially, the frequency of the PWM signal is crucial and there are two factors affecting the attainable frequency of the PWM signal: frequency of the system clock and the prescalar values of the timer modules. In the PWM module configuration register, values can be set to control the phase shift, to insert a dead band and to activate the interrupt service routine to make some other modules operational. The codes for PWM generation in DSP are shown below.

```
void pwmIntialization(void)
{
// CONTROL OF 3-PHASE INVERTER -COMMONLY USED IN MOTOR CONTROL
EPwm1Regs.TBPRD = 3000;
EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB_DIV4; // Clock ratio to SYSCLKOUT
EPwm1Regs.TBCTL.bit.CLKDIV = TB_DIV4;
EPwm1Regs.TBPHS. half. TBPHS = 0; // Set Phase register to zero
// Symmetrical mode for Time Base counter
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN;
// PHASE REGISTER ENABLE PHSEN
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE;
// Time-base period register (TBPRD) has a shadow register
// Shadow register provides a temporary holding location for the active register
EPwm1Regs.TBCTL.bit.PRDLD = TB_SHADOW;
// Synchronization Output Select SYNCOSEL
// Time-base counter equal to zero (TB_CTR = 0x0000)
EPwm1Regs.TBCTL.bit.SYNCOSEL = TB_CTR_ZERO; // Sync down-stream module
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR=Zero
// force ePWMA output high (counter incrementing)
EPwm1Regs.AQCTLA.bit.CAU = AQ_SET;
EPwm1Regs.AQCTLA.bit.CAD = AQ_CLEAR;
// Dead-band generation with independent rising and falling edge delay control
EPwm1Regs.DBCTL.bit.OUT_MODE = DB_FULL_ENABLE; // enable Dead-band module
EPwm1Regs.DBCTL.bit.POLSEL = DB_ACTV_HIC; // Active Hi complementary
EPwm1Regs.DBFED = 120; // FED = 50 TBCLKs
EPwm1Regs.DBRED = 120; // RED = 50 TBCLK
```

In motor control we have voltage source inverter, the input to this inverter is the pulse width modulation. The frequency of the PWM is controlled by the TBPRD register in the microcontroller. In my design case the switching frequency of the TMDSHVMTRPFCKITIPM is limited to 10000hertz(10kHz). So according to this we can limit our frequency as follows.

$$TBPRD = \frac{T_{PWM}}{2(T_{SYSCLKOUT})(HSPCLKDIV)(CLKDIV)}$$

$$T_{PWM} = \frac{1}{10000} sec$$

Frequency of Sysclkout=60,000,000Hz(60MHz), based on this value, set appropriate HSPCLKDIV=0 and CLKDIV=0 value finally we get the TBPRD=3000 register values, to control the switching frequency.

The other main thing we must consider is the values of the rising edge and falling edge delay. This is used in dead band Most power electronic systems require pairs of PWM pulse series

to control two power switches in such a way, that if one switch is on (conducting) the upper leg, the other switch is off (open circuit) the lower leg of the inverter.

In my implementation case, the operation of inverter is generation complementary signals in the upper and lower side of the inverter. The complementary signal means if one is high the other must be low, otherwise, a fault exists. The upper and lower PWM signal must not be a switch at the same time. We need dead time; falling and rising edge dead time. the formula to calculate falling-edge-delay and rising-edge-delay are:

falling-edge-delay=(DBFED)(T_{TBCLK})

rising-edge-delay=(DBRED)(T_{TBCLK})

$$T_{TBCLK} = \frac{T_{SYSCLKOUT}}{(HSPCLKDIV)(CLKDIV)}$$

Based on the above formula the falling edge delay and rising edge delay limited to 2 microseconds.

3.5.4 JTAG Header

The XDS100v2 is the second generation of the XDS100 family of low-cost JTAG debug probes (emulators) for TI processors. Designed to deliver full-featured JTAG connectivity at a low cost, the XDS100 is the family of choice for entry-level debugging of TI microcontrollers, processors and wireless devices. The XDS100 family supports the traditional IEEE1149.1 and operates with interface levels of +1.8 V and 3.3 V. All XDS100 models support either USB2.0 Full Speed (11Mbps) or High Speed (480Mbps) connection to the host. The XDS100 family is fully compatible with TI's Code Composer Studio IDE. This combination gives a complete hardware development environment which includes an Integrated Debug Environment, Compiler, and full hardware debugging and Trace capability of TI microcontrollers and processors. The JTAG XDS100 connections are shown in the Figure 3.6 based on high voltage digital motor control kit schematic diagram.

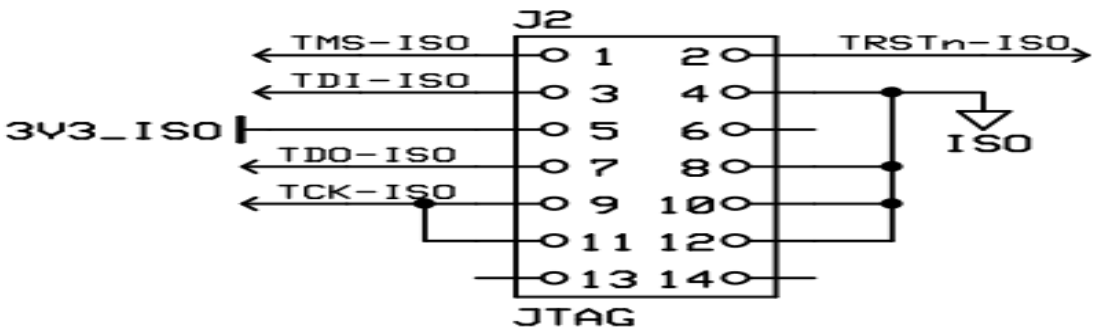


Figure 3.6: JTAG connector pins to DSP in the kit

3.5.5 Power Supply Unit

The various blocks and interface modules of DSP need power supplies of +14 Volt, 3.3 Volt, and 5 Volt DC volt. These voltages are derived onboard by using various voltage regulators.

3.6 Code Composer Studio™ – IDE

Code Composer Studio (CCS) is an integrated development environment (IDE) to develop applications for Texas Instruments (TI) embedded processors. I used Code Composer Studio (CCS) integrated development environment (IDE) tools to develop a program. Code Composer Studio™ IDE gives you the ability to edit, build, and debug highly optimized embedded applications.

Code Composer Studio is primarily used for embedded project design and low-level JTAG based debugging. However, the latest releases are based on unmodified versions of the Eclipse open source IDE, which can be easily extended to include support for OS level application debug (Linux, Android, and Windows Embedded) and open source compiler suites such as control SUIT. The figure shows a screenshot of CCS.

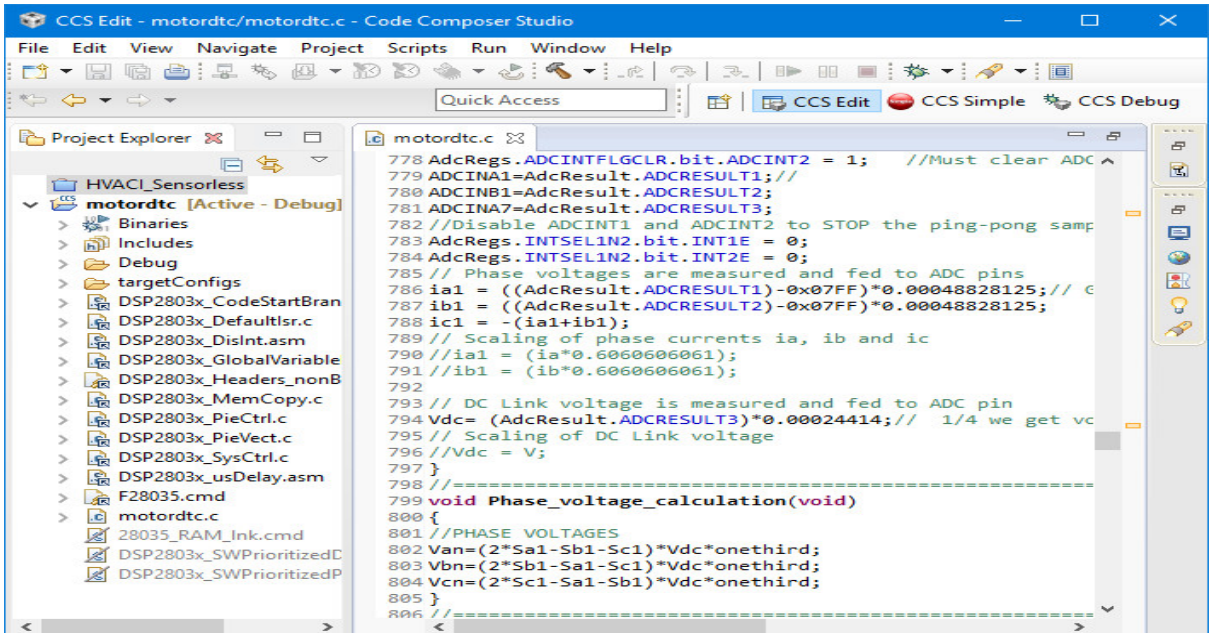


Figure 3.7: The screenshot of code composer IDE for DSP programming

3.7 High Voltage Motor Control and PFC Developer's Kit

From many other motor control tools, HVM is one of the technologies designed, developed by Texas instrument company. TMDSHVMTRPFCKIT is a DIMM100 control CARD based motherboard evaluation module. The High Voltage Motor Control and PFC Developer's Kit

provides a great reference platform to learn and experiment with digital control of high voltage motors with Texas Instruments' C2000™ 32-bit microcontroller family. It supports code composer studio IDE. This kit is a superb, all-around motor inverter design tool, showcasing control of the most common types of high voltage, three phase motors, including AC induction motor (ACI), brushless DC (BLDC), and permanent magnet synchronous motors (PMSM).

Digital motor control methods are developed and implemented, including trapezoidal, variable frequency (V/F), and field-oriented control (FOC) using sensored or sensorless techniques. But still, there is no Direct Torque Control demonstration or implementation using this development kit.

Figure 3.8 High voltage motor control and PFC developer's Kit (TMDSHVMTRPFCKIT)

CHAPTER 4

DTC Drive MATLAB Simulation Modeling and DSP Implementation Results

4.1 Introduction

MATLAB Simulation analysis of three phase squirrel cage induction motor based on direct torque control drive scheme is simulated using MATLAB 2016a platform. Using the set of equations provided in chapters 2 and 3, the model is implemented. The Simulink modeling contains modeled three-phase induction motor in a stationary reference frame, rotor speed estimator, stator flux linkage estimation block, stator flux hysteresis comparator, stator flux linkage position identifier block, estimated torque block, torque controller (hysteresis comparator) optimal switching table block and voltage source inverter. Various parameters are plotted within a given time interval in seconds.

4.2 Simulation Analysis

The simulation analysis is based on the normal operating rating of induction machine not in per unit system (PU), but TMDSHVMTRPFCKIT is work based on per unit during implementation. All the equation is divided by its corresponding base values based on table 4.2 during implementation. The following are the parameters of three-Phase SCIM used during the simulation and also these parameters are used in the implementation.

Parameters	Values
Rated power (P)	370watt
Stator resistance (R_s)	11.05ohm
Rotor resistance (R_r)	6.11ohm
Stator linkage inductance (L_s)	0.316423H
Rotor linkage inductance (L_r)	0.316423H
Mutual inductance (L_m)	0.293939H
Moment of inertia (J)	0.009kg.m^2
Number of poles (p)	4
Stator rated current (i_s)	1.1Amp
Rated frequency (f)	50Hz
Rated electrical(mechanical) rotor speed	2640(1320) rpm
Rated stator phase rms voltage	220volt

Table 4.1: System specification of squirrel cage induction motor

The 3-phase stator winding of the Squirrel-cage Induction Motor (SCIM) is connected as star or Y Connection without return of neutral path.

Base parameters	Values
Base phase voltage	220V
Base phase current	10A
Base flux	2.93939 volt-sec/rad
Base frequency	50Hz
Base torque	22.604485 N.M
Switching frequency of IPM	10kHz
Base Dc Bus Voltage	300volt

Table 4.2: Base values of IPM voltage source inverter and induction motor

4.2.1 Induction Motor Modeling for Simulation Analysis

Depend on the mathematical equation (3.19-3.34) we got the stationary reference frame dynamic modeling of the three-phase induction machine.

$$\left.\begin{aligned} v^{s}_{d_s} &= R_s i_{d_s}(t) + \frac{d\psi_{d_s}(t)}{dt} \\ v^{s}_{q_s} &= R_s i_{q_s}(t) + \frac{d\psi_{q_s}(t)}{dt} \end{aligned}\right\} \tag{4.1}$$

We have also another expression for the above stator voltage equation by substituting the flux linkage based on current model equations

$$\psi_{d_s} = L_s i_{d_s} + L_m i_{d_r} \tag{4.2}$$

$$\psi_{q_s} = L_s i_{q_s} + L_m i_{q_r} \tag{4.3}$$

$$\left.\begin{aligned} v^{s}_{d_s} &= R_s i_{d_s}(t) + \frac{d}{dt}(L_s i_{d_s} + L_m i_{d_r}) \\ v^{s}_{d_s} &= R_s i_{d_s}(t) + \frac{di_{d_s}(t)}{dt} L_s + L_m \frac{di_{d_r}(t)}{dt} \end{aligned}\right\} \tag{4.4}$$

$$\left.\begin{aligned} v^{s}_{q_s} &= R_s i_{q_s}(t) + \frac{d}{dt}(L_s i_{q_s} + L_m i_{q_r}) \\ v^{s}_{q_s} &= R_s i_{q_s}(t) + \frac{di_{q_s}(t)}{dt} L_s + L_m \frac{di_{q_r}(t)}{dt} \end{aligned}\right\} \tag{4.5}$$

The same procedure for the rotor voltage equation

$$\left.\begin{aligned} R_r i_{d_r}(t) + L_r \frac{di_{d_r}}{dt} + L_m \frac{i_{d_s}}{dt} + \omega_r (L_r i_{q_r} + L_m i_{q_s}) = 0 \\ R_r i_{q_r}(t) + L_r \frac{di_{q_r}}{dt} + L_m \frac{di_{q_s}}{dt} - \omega_r (L_r i_{d_r} + L_m i_{d_s}) = 0 \end{aligned}\right\} \quad (4.6)$$

The other equation is from the relation of speed and torque; i.e. mechanical equation.

$$J \frac{d\omega_m}{dt} = T_{estimated} - T_{load} \quad (4.7)$$

ω_r *is* rotor shaft electrical speed in rad per second. ω_m is the mechanical rotor speed in rad per second. substituting equation 4.2&4.3 into the estimated torque equation 2.32. Finally, we have the electrical speed equation as and assume load torque is zero:

$$\left.\begin{aligned} \omega_r &= [polepair(pp)]\omega_m \\ \frac{J}{pp}\frac{d\omega_r}{dt} &= \frac{3}{2} pp(i_{q_s}(L_s i_{d_s} + L_m i_{d_r}) - i_{d_s}(L_s i_{q_s} + L_m i_{q_r})) \end{aligned}\right\} \quad (4.8)$$

the above five differential equation(4.4,4.5,4.6,4.8) is used for the induction machine Simulink model to see the direct torque control drive performance.

4.2.2 Stator Flux Estimator

In reference to stator flux estimator from equation (3.21), we have the derivative term or integral term in our equation. In a digital implementation, we must change this equation from continues to discrete. In a continuous form of an equation, we know Laplace (s) term for derivative, $\frac{1}{s}$ for the integral term. In a digital implementation, we also define the following based on the types of approximation we used. In this case backward approximation. Replace $\frac{1}{s}$ term (the integral term) by: $\frac{1}{s} = \frac{TsZ}{Z-1}$ where s is continuous Laplace term and Z is discrete operator, T_s is the sampling time.

$$\left.\begin{aligned} v^s_{d_s} &= R_s i_{d_s}(t) + \frac{d\psi_{d_s}(t)}{dt} \\ v^s_{q_s} &= R_s i_{q_s}(t) + \frac{d\psi_{q_s}(t)}{dt} \end{aligned}\right\} \quad (4.9)$$

Based on equation 3.8 and 3.9, for both equations replace the integral term by $\frac{TsZ}{Z-1}$ as discussed in section 4.2.2. We have the following discrete form,

$$\left.\begin{aligned}\psi_{q_s} &= T_s(v^s{}_{q_s} - R_s i_{q_s}) + \psi_{q_{s-1}} \\ \psi_{d_s} &= T_s(v^s{}_{d_s} - R_s i_{d_s}) + \psi_{d_{s-1}}\end{aligned}\right\} \quad (4.10)$$

$\Psi_{q_{s-1}}$ is the previous value of the stator flux in the q axis, $\Psi_{d_{s-1}}$ the previous value of the stator flux in d axis. We use a lowpass filter with cutoff frequency $3\frac{rad}{sec}$ after sampling.

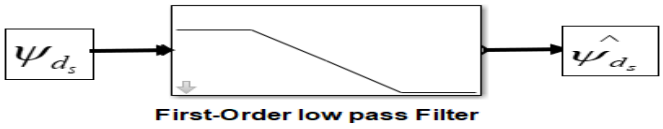


Figure 4.1: Low pass filter

The transfer function of the lowpass filter is $\frac{1}{1+s\tau_c}$, $\tau_c = \frac{1}{2\pi fc}$ is the time constant. by using some analysis as above, the backward approximation can relate the $\widehat{\Psi_{ds}}$ and Ψ_{ds}.

$\widehat{\Psi_{ds}} = \frac{1}{1+s\tau_c}\Psi_{ds}$, and for q axis flux $\widehat{\Psi_{qs}} = \frac{1}{1+s\tau_c}\Psi_{qs}$.

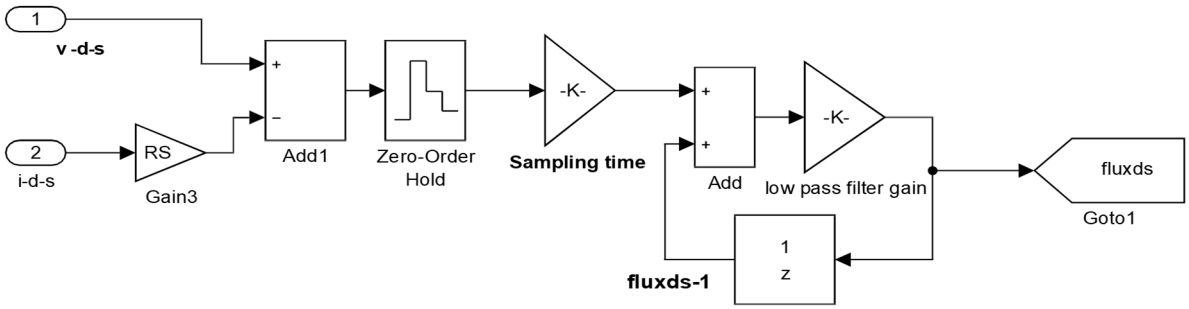


Figure 4.2: Digital simulink block diagram for stator d axis flux in a stationary frame

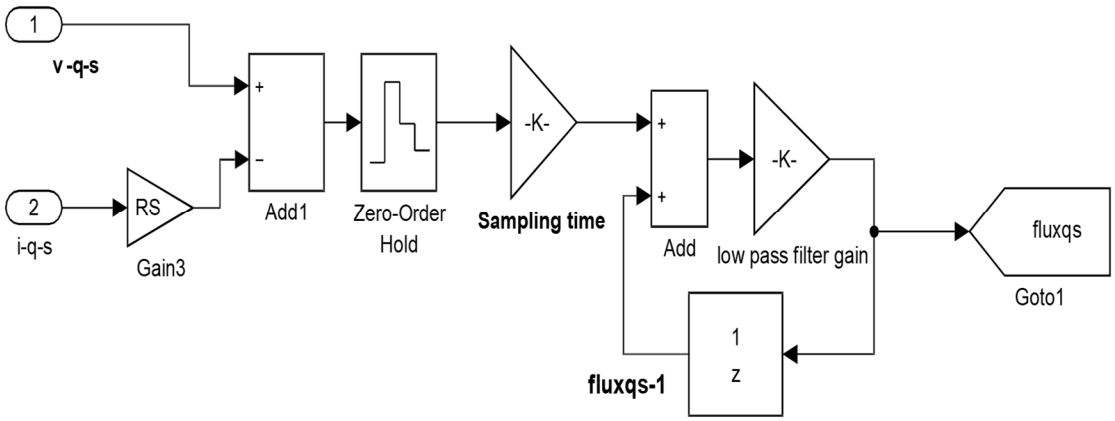


Figure 4.3: Digital simulink block diagram for stator q axis flux in the stationary frame

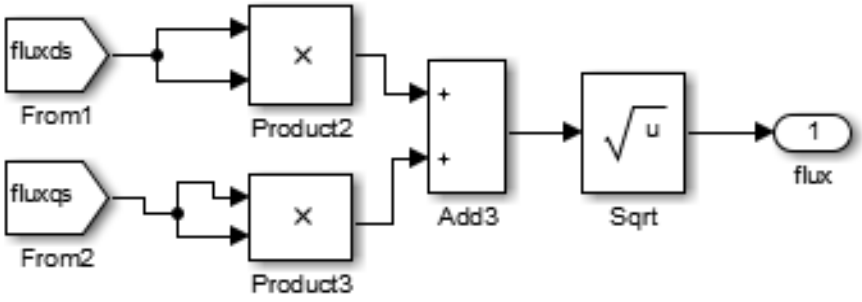


Figure 4.4: Stator flux calculation simulink block

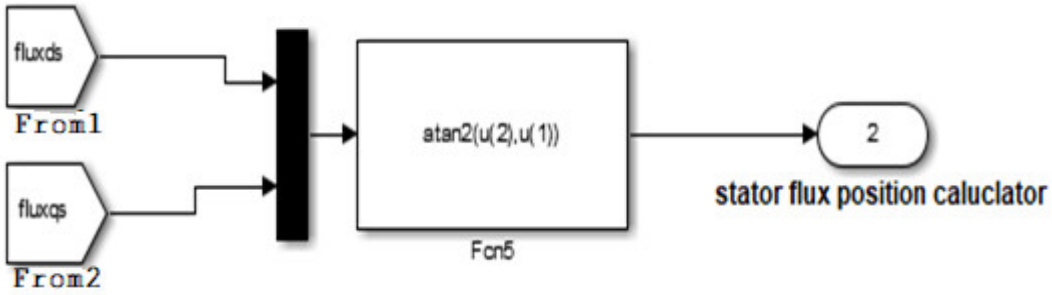


Figure 4.5: Flux position simulink block

4.2.3 Torque Estimator

Torque is estimated based on the derived equation 2.32. we derived many types of torque equation. But the selected equation is measured directly from the sensed parameters so it reduces the Burdon on the microcontroller.

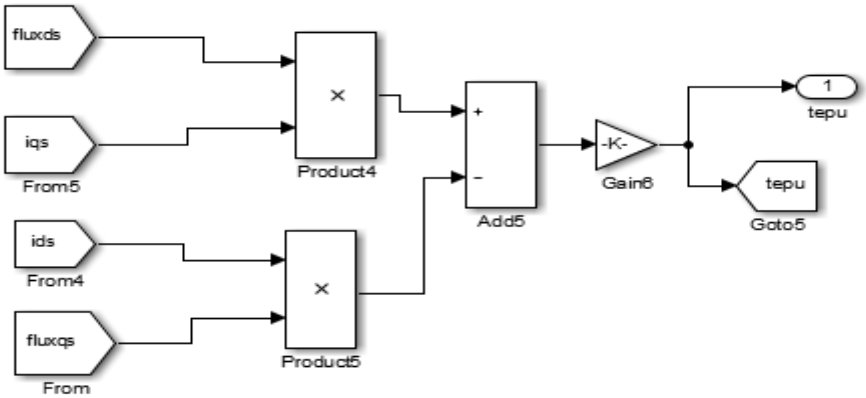


Figure 4.6: Torque estimator simulink block

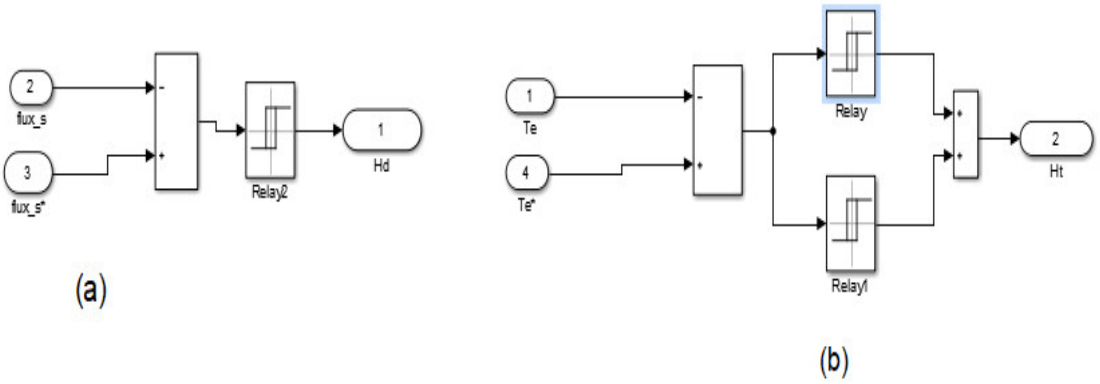


Figure 4.7: Hysteresis block a) Flux hysteresis controller b) Torque hysteresis controller

4.3 Speed Estimation

Rotor Speed sensing techniques are not used speed sensors such as quadrature incremental encoder instead of this it uses it's own estimated and measured electrical output parameters like voltage, current, stator flux linkage, rotor flux linkage, and developed torque. Speed estimation of an induction machine is more complex than other synchronous due to the slip speed. So, for asynchronous machines, we have many tricks for rotor speed estimation. from [11], various techniques are described which can be used in high-performance drives for the estimation of the synchronous speed, slip speed, rotor speed, rotor angle, and various machine flux linkages. Thus, the following techniques are given:

1. Open-loop estimators using monitored stator voltages and currents;
2. Estimators using the spatial-saturation stator phase third-harmonic voltage;
3. Estimators using saliency (geometrical, saturation) effects;
4. Model reference adaptive systems (MRAS);
5. Observers (Kalman, Luenberger);
6. Estimators using artificial intelligence (neural network, fuzzy-logic-based systems etc)

The first speed estimator techniques using monitored stator voltage and currents are used for simulation and implementation. The [12] main advantages of sensorless-controlled drives are the reduced hardware complexity, the lower cost, the reduced size of the drive machine, the elimination of the sensor cables, the better noise immunity, the increased reliability, and the lower maintenance requirements and lower robustness. Our analysis is in the stationary reference frame fixed to the stator; the differentiation of stator flux angle gives the synchronous speed. which is defined from equation 3.11.

$$\omega_{syn} = \frac{d\rho_s}{dt} = \frac{d}{dt}(a\tan 2\frac{\psi_{q_s}(t)}{\psi_{d_s}(t)}) \tag{4.11}$$

When we differentiate the above equation using the property of inverse tangent function differentiation.

$$\omega_{syn} = \frac{\psi_{d_s}(t)\frac{d}{dt}\psi_{q_s}(t) - \psi_{q_s}(t)\frac{d}{dt}\psi_{d_s}(t)}{\psi_s^{\,2}} \tag{4.12}$$

We can also express synchronous speed based on rotor flux linkages angle.

$$\omega_{syn} = \frac{d\rho_r}{dt} \tag{4.13}$$

$$\omega_{syn} = \frac{\psi_{d_r}(t)\frac{d}{dt}\psi_{q_r}(t) - \psi_{q_r}(t)\frac{d}{dt}\psi_{d_r}(t)}{\psi_r^{\,2}} \tag{4.14}$$

The d-q component of the rotor flux linkage is obtained based on the equation described in (2.23&2.25). both of the equation is not giving us the exact estimated rather it gives disturbed higher frequency noise signal due to modeling error. During digital implementation, this can be removed by first order a low pass filter. We use a low pass filter with cutoff frequency $5\frac{rad}{sec}$.

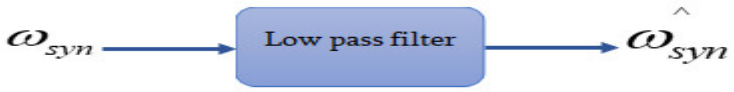


Fig 4.8: Low pass filter for synchronous speed estimation

$$\widehat{\omega_{syn}} = \omega_{syn}\frac{1}{1+s\tau_c} \tag{4.15}$$

$$\hat{\omega}_{syn}\tau_c s + \hat{\omega}_{syn} = \omega_{syn} \tag{4.16}$$

From equation 4.16 , using backward approximation techniques replace : $\frac{1}{s} = \frac{TsZ}{Z-1}$, we get;

$$\hat{\omega}_{syn} = \omega_{syn}\frac{T_s}{T_s+\tau_c} - \hat{\omega}_{syn-1}\frac{\tau_c}{T_s+\tau_c} \tag{4.17}$$

The main challenging task during asynchronous machine speed estimation is to obtain slip speed estimation. We have from many tricks of slip speed estimation [1].

$$\omega_{slip} = \frac{2R_r T_{estimated}}{polepair 3\psi_r^{\,2}} \tag{4.18}$$

This slip estimation techniques have simple mathematical analysis and can be easy for coding from other methods, but it needs resultant rotor flux linkage. So, this rotor flux linkage can be obtained from equation 2.25 described in chapter 2, i.e. the current model of flux linkage. Also, the main thing we are going to do in this slip estimation techniques, the synchronous speed estimation method also changed from the above. The slip speed is depending on rotor flux linkage speed [1].

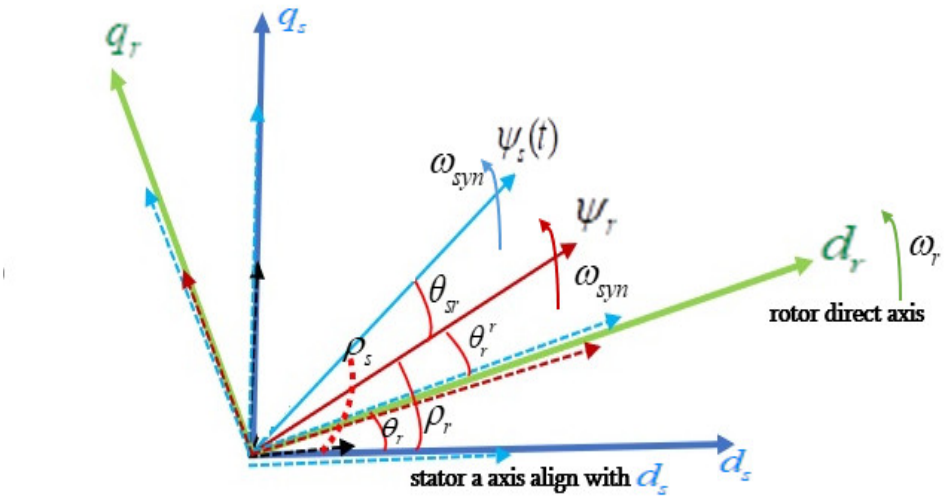


Figure 4.9: d-q stationary axis relationship with speed

Then the rotor speed is estimated as follows.

$$\omega_r = \omega_{syn} - \omega_{slip} \tag{4.19}$$

$$\omega_r = \frac{\psi_{d_r}(t)\frac{d}{dt}\psi_{q_r}(t) - \psi_{q_r}(t)\frac{d}{dt}\psi_{d_r}(t)}{\psi_r^{\,2}} - \frac{2R_r T_{estimated}}{pp 3\psi_r^{\,2}} \tag{4.20}$$

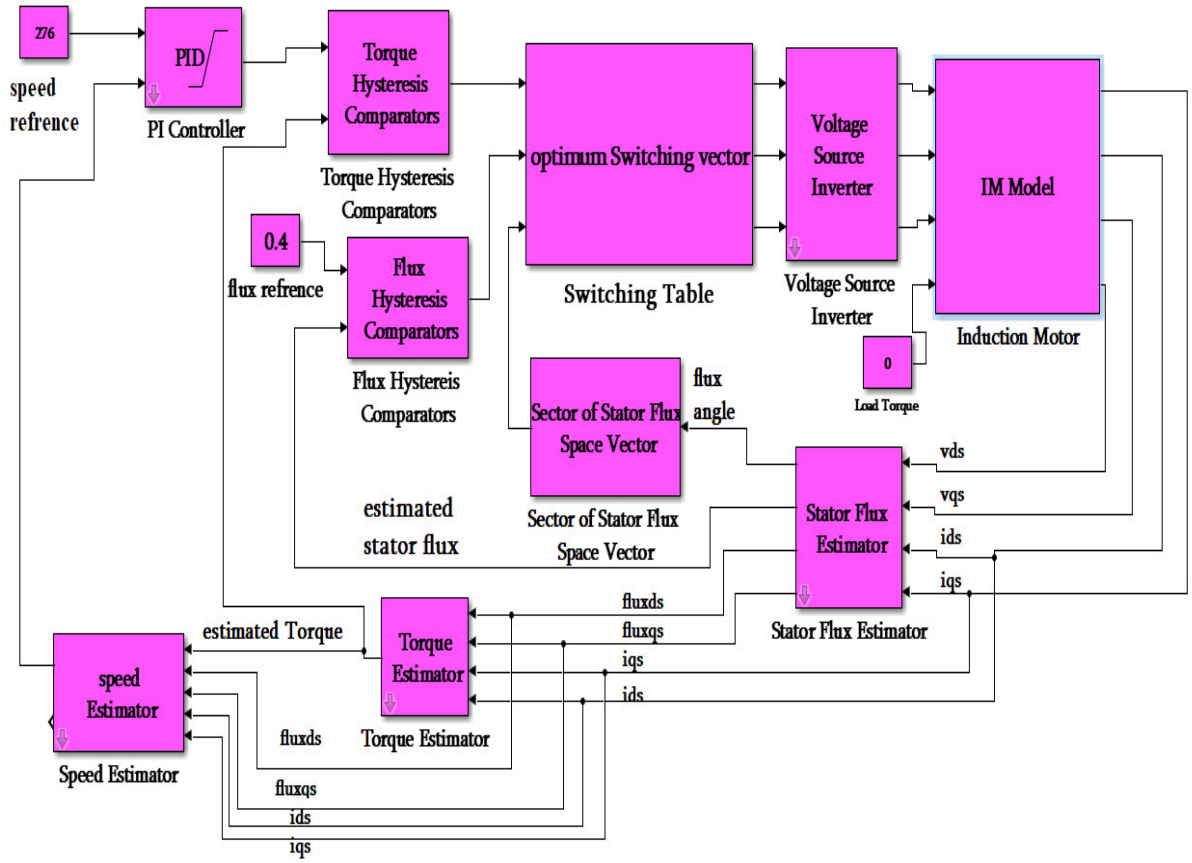


Figure 4.10: Overall MATLAB simulation system

4.4 Simulation Results

The simulation results are based on Overall MATLAB simulation system presented above figure. The details of each block will be presented in the appendix. also, some of the blocks are described in this chapter. The Simulink block generally has the direct torque control IM drive with a closed-loop speed control system using estimated speed as feedback comparing with the reference speed using a PI as a speed controller. The PI controller gains are obtained by trial and error, due to this, it is the most challenging task in this Thesis to get exact controller parameters. The MATLAB plots without load are shown below. The reference values for the different parameter are shown in below table.

Reference parameter	Values
Stator flux	0.4volt-sec/rad
Rotor Electrical Speed reference	276 rad/sec (maximum rotor speed)
K_P	50
K_I	0.03
VDC(dc bus voltage)	200V

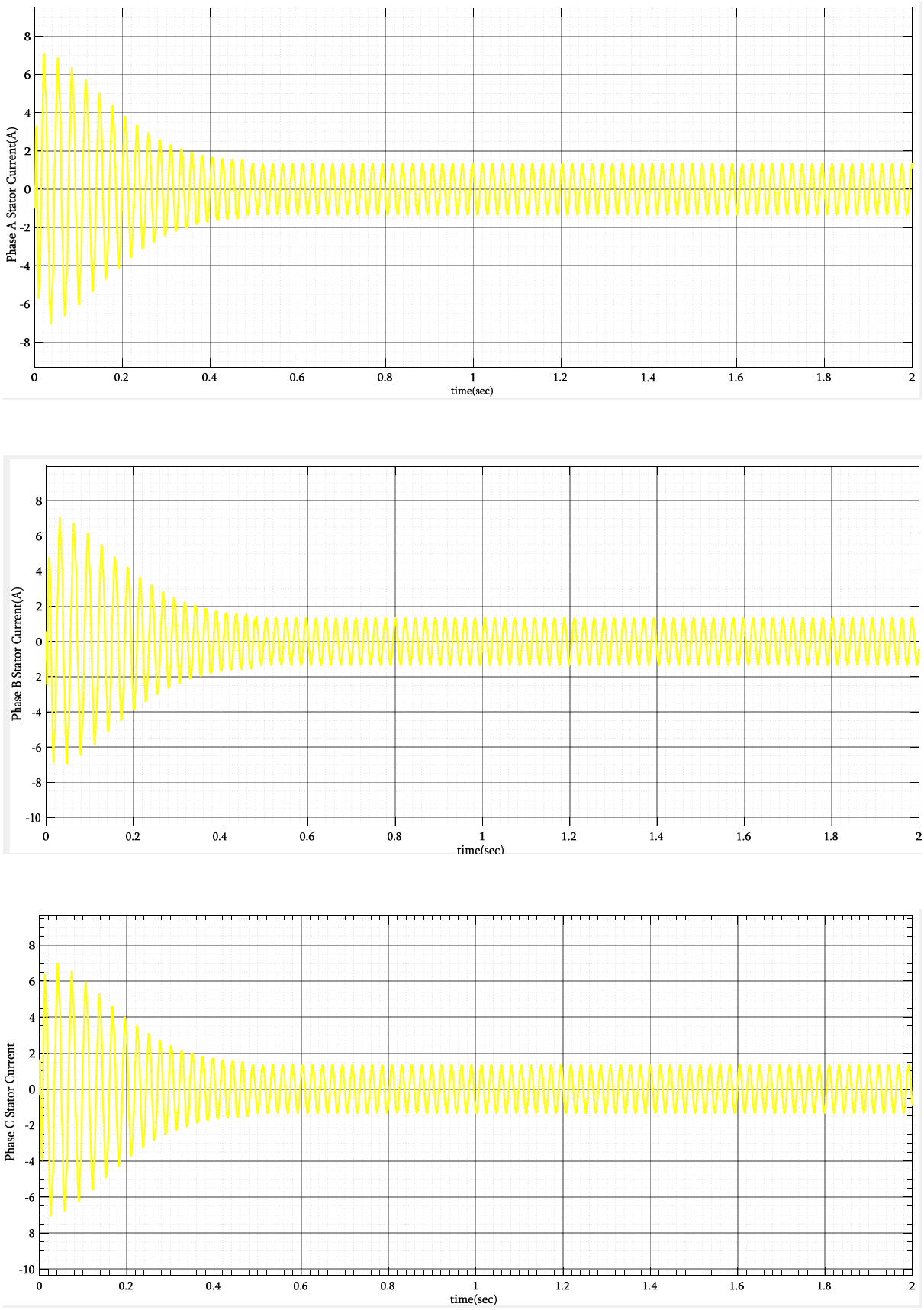


Figure 4.11: Stator three-phase currents MATLAB simulink results

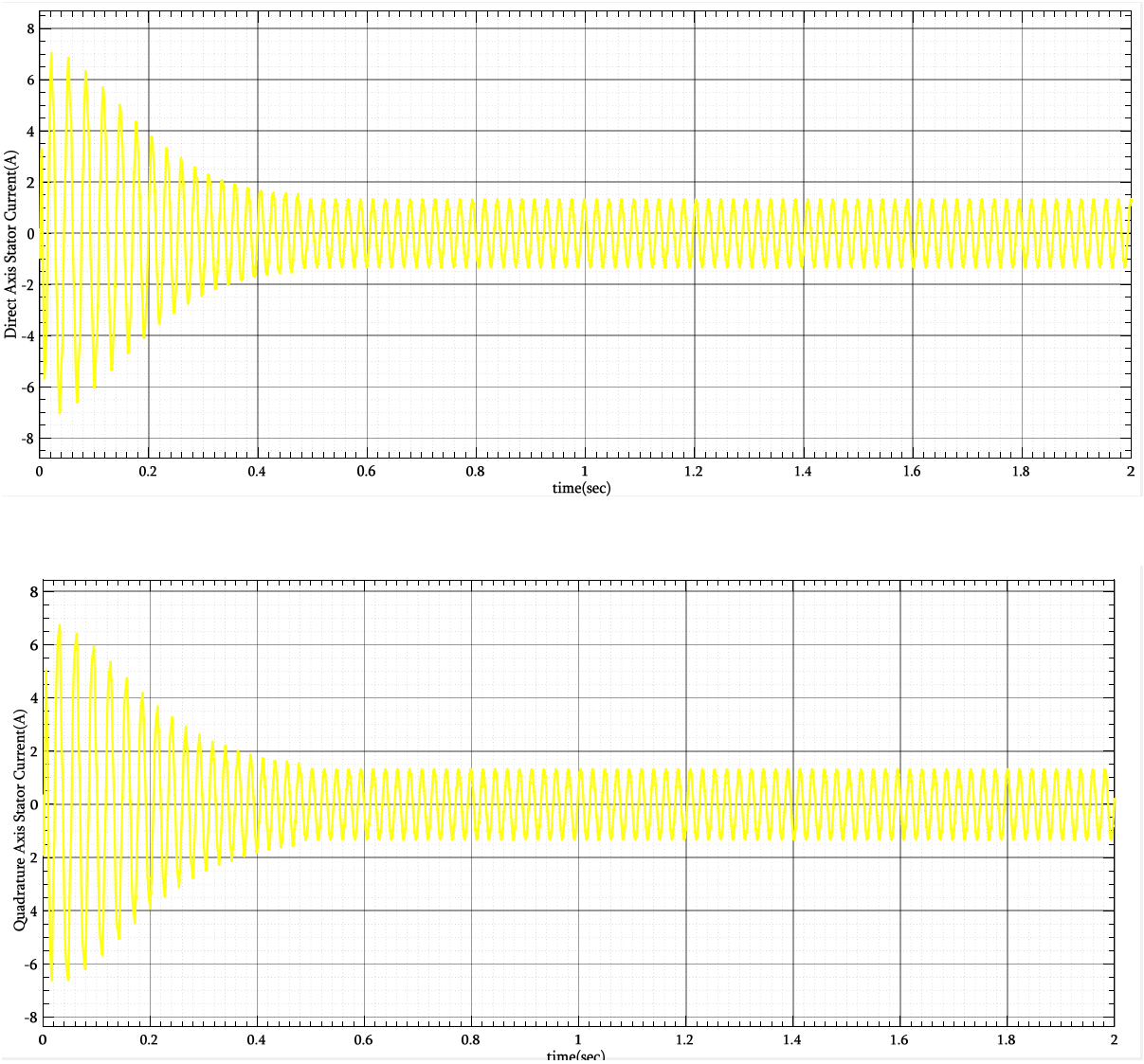


Figure 4.12: Stationary direct and quadrature axis current

The above plot figure 4.11 shows that initially the stator phase current has larger value which comes from speed PI controller gains but after some time when it reaches at steady it gets smaller values which almost gives the rating current of the motor. Also, we observe that the three phase stator winding current has approximately sinusoidal waveforms with constant frequency. We have also the direct and quadrature axis currents as shown in figure 4.12; the direct axis current almost the same as the stator phase A current and from the figure 4.12 the angle between the direct and quadrature stator current has 90^0 phase shift as expected. We observe this phase shift also in direct and quadrature axis stator flux below in figure 4.13.

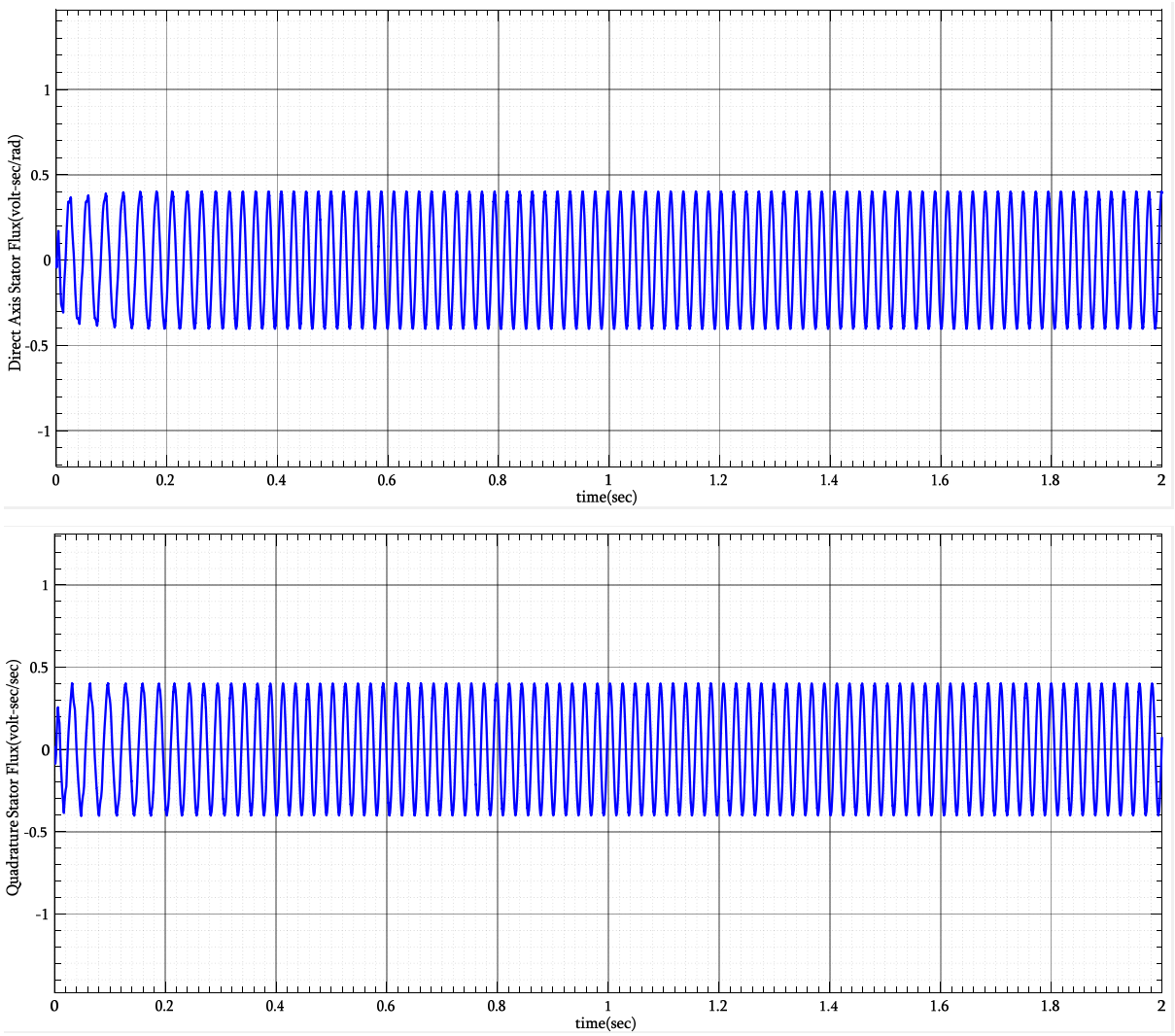
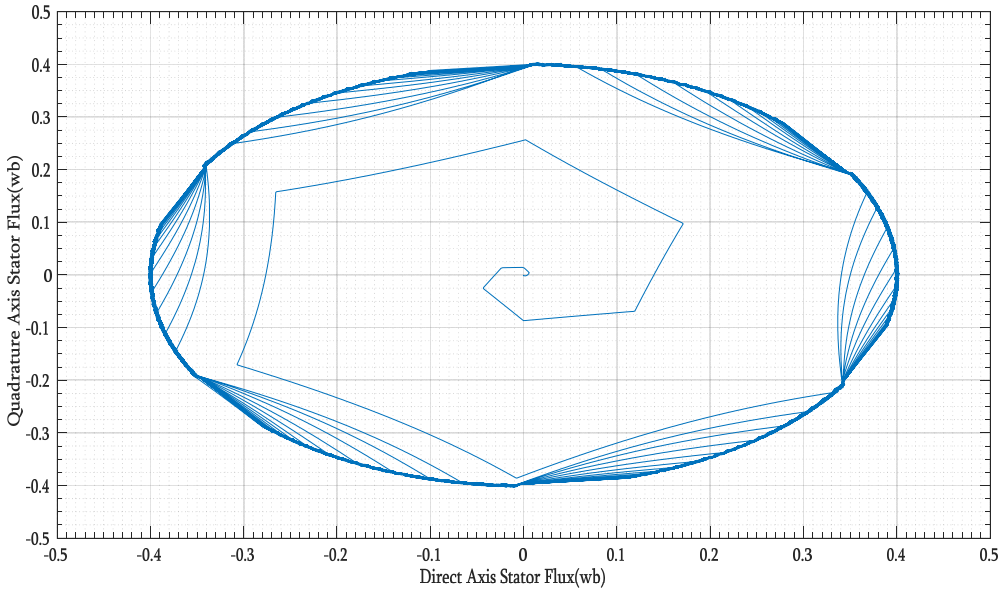
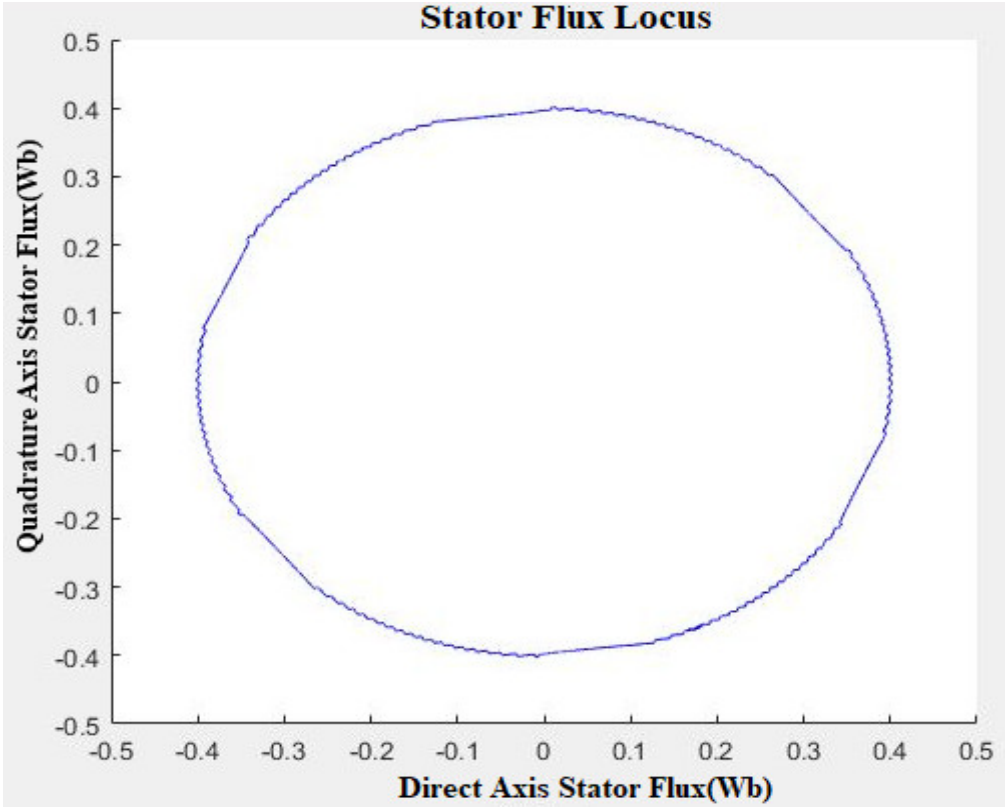


Figure 4.13: Stationary direct and quadrature axis flux



(a)



(b)

Figure 4.14: Stator flux locus(a)Transient flux locus b) Steady-state flux locus)

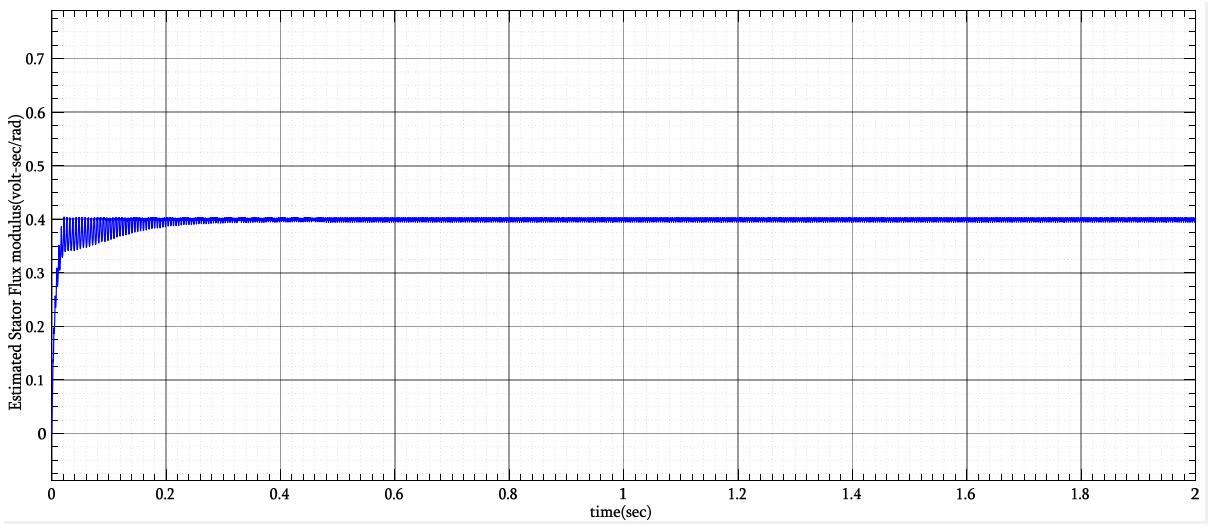


Figure 4.15: Estimated stator flux modulus

From figure 4.15, we can see the resultant stator flux value maintained constant throughout the running time. The flux locus in figure 4.14 indicates that it plots the flux in the direct axis (horizontal) verses in the quadrature axis(vertical). So, the locus must be maintained at the flux reference otherwise the flux estimation technique is not proper.

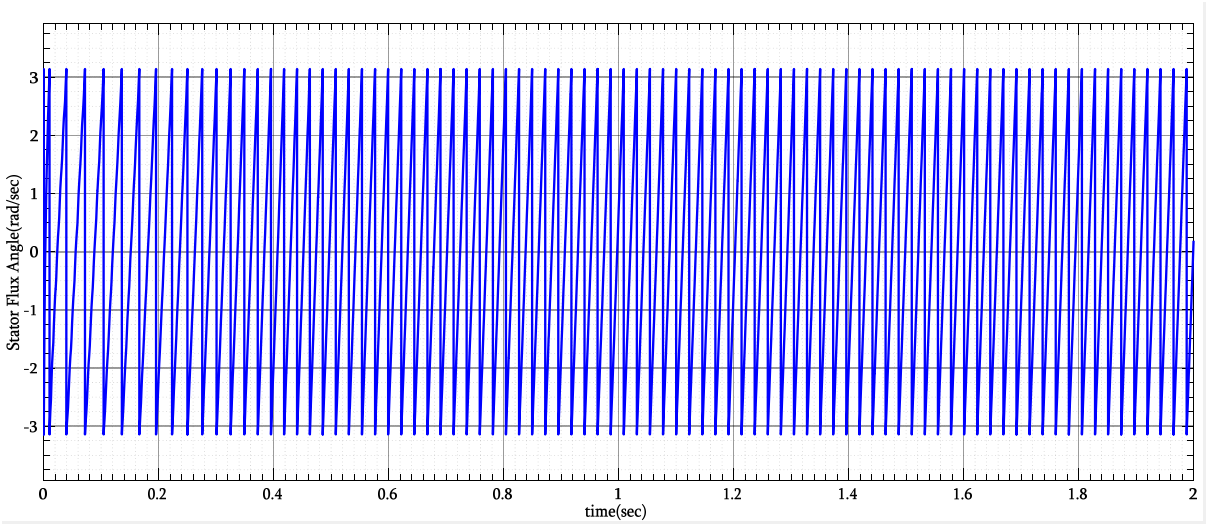


Figure 4.16: Estimated stator flux angle

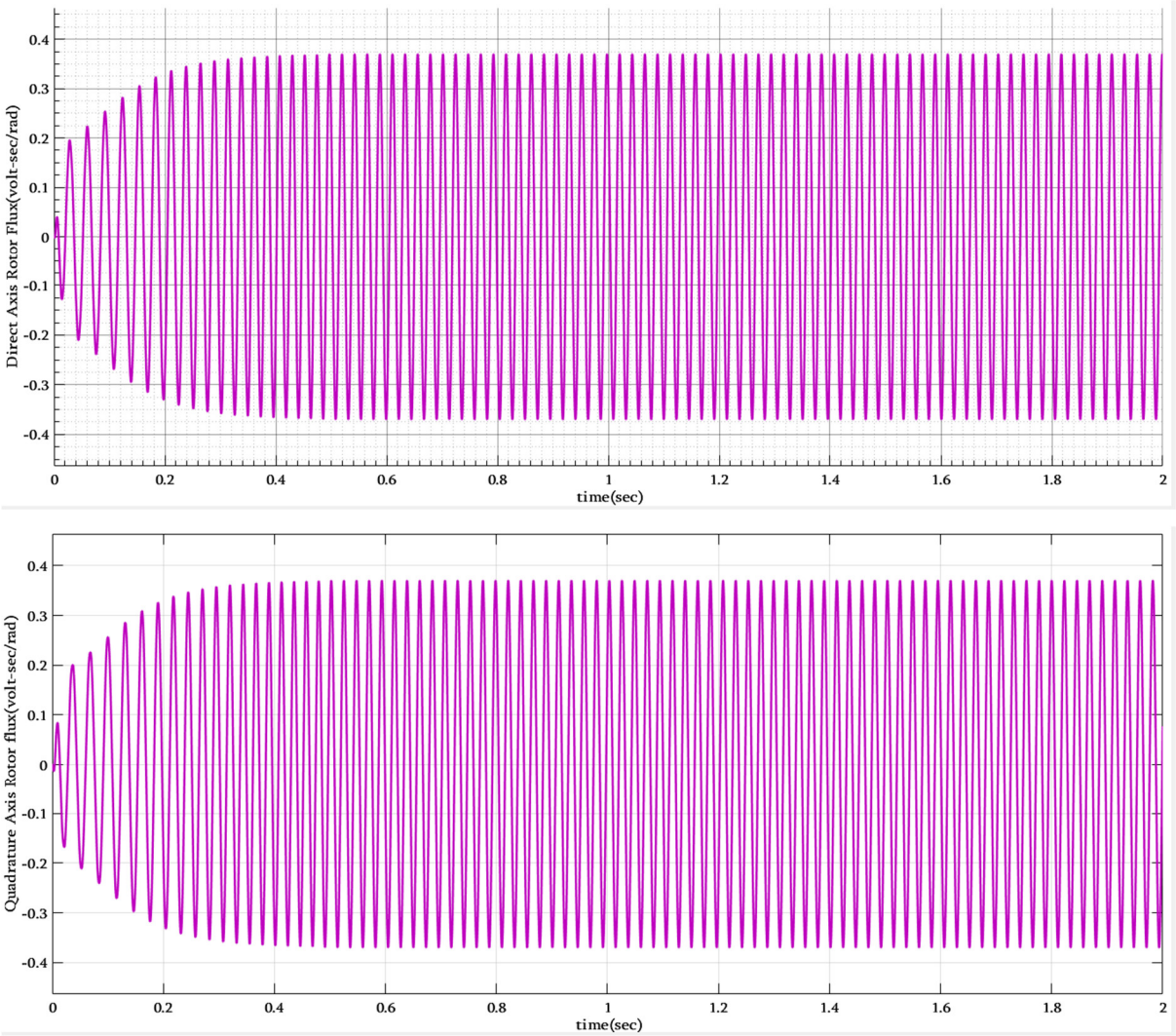
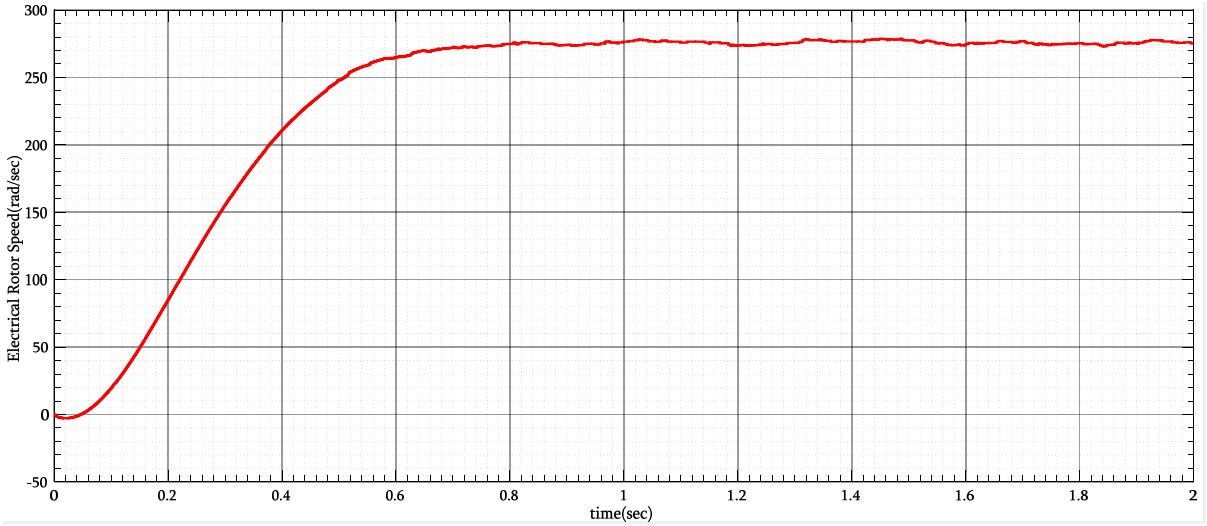
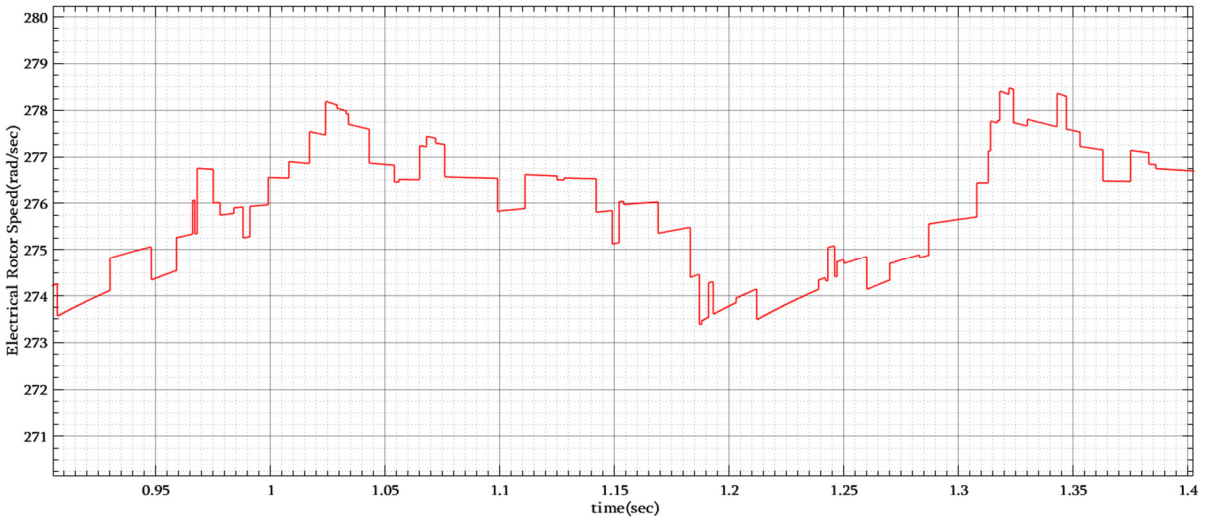


Figure 4.17: Estimated direct and quadrature rotor flux in the stationary reference frame



(a)



(b)

Figure 4.18: a) Estimated rotor speed without load torque b) rotor speed in zoom view

The reference speed is the step input with value 276 $\frac{rad}{sec}$, we got the estimated speed reaches its reference speed approximately at 0.5 second and it remains constant after this time. We also observed that after the speed reaches its reference speed the stator winding currents (approximately sinusoidal waveforms), the direct axis stator flux (has sinusoidal waveforms), quadrature axis stator flux (has sinusoidal waveforms), and the stator flux angle(position) has stable waveforms with a constant frequency.

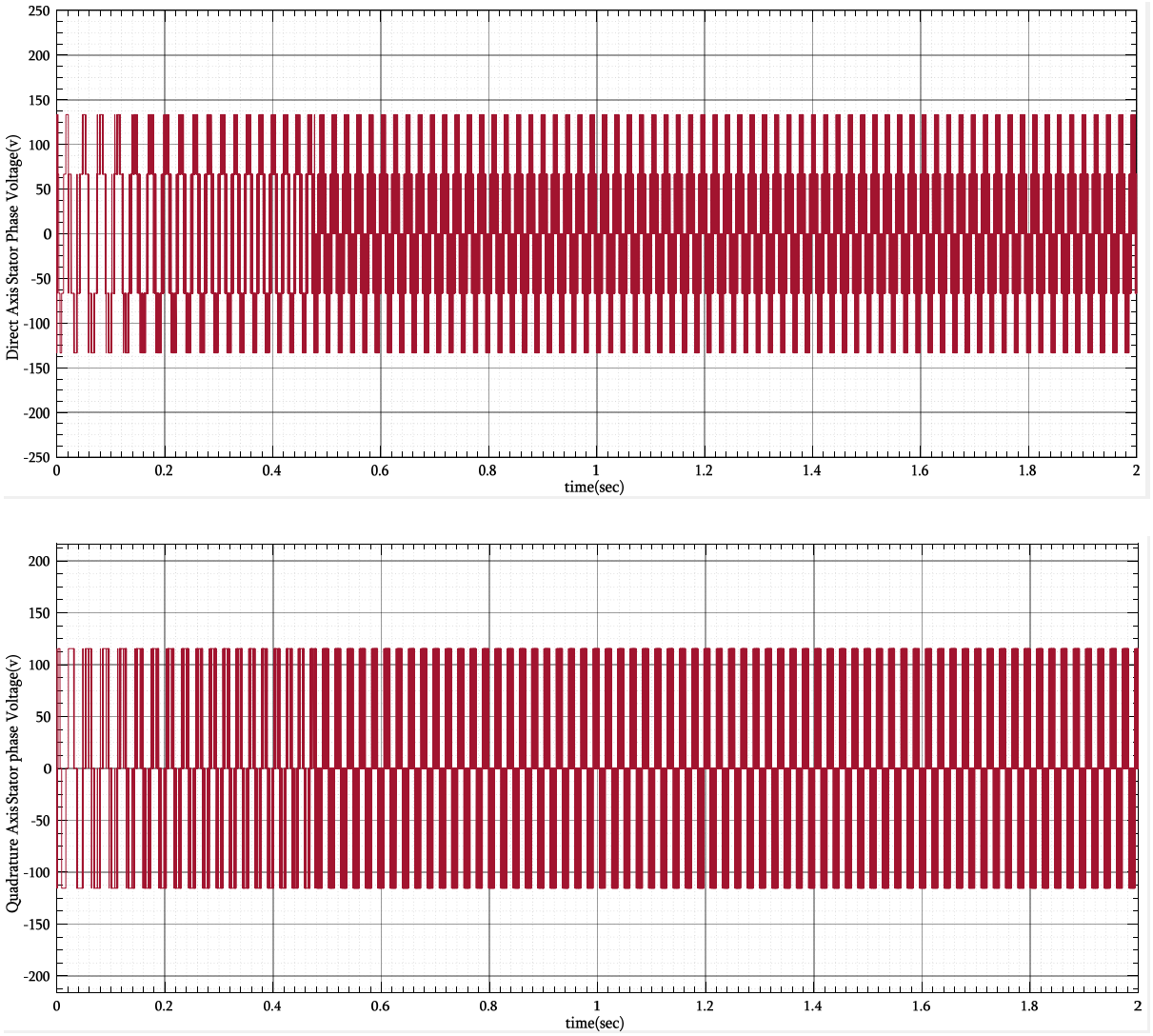


Figure 4.19: Phase voltage in the stationary d-q axis

during digital implementation the most important part is the estimation of stator flux, torque, and rotor speed based on the measured (sensed) currents i_{as} and i_{bs} converted to i_{ds} and i_{qs} using transformation and sensed dc bus voltage with reconstructed PWM signals (from this v_{ds} and v_{qs}), then the whole system is work based on i_{ds}, $i_{qs,}$ v_{ds}, $v_{qs.}$ The digital implementation always considered with sampling time $T_{s.}$

4.5 DSP Implementation of DTC Drive

The three-phase squirrel cage induction motor is driven by the conventional voltage-source inverter. The digital signal controller TMS320F28035 control card is being used to generate the six switching signals using DTC scheme and SVPWMM-DTC Scheme for the six-power switching devices. The two input phase currents of the motor (ia1 and ib1) are measured from the inverter using the current sensors and are fed to the pins A1 and B3 of ADC module

CONTROL CARD via OPA2350 operational amplifier IC circuitry. Similarly, the DC link voltage Vdc across the input capacitors is measured using a voltage divider and sent to ADC pin A7 of TMS320F28305 control card. This DC voltage is necessary to calculate the three phase voltages of the motor when the switching functions are known. Using the sensed phase currents and sensed dc voltage, the phase voltages in the stator referred to the stationary, rotor speed, electromagnetic torque, and the stator flux linkages are calculated and plotted. By tuning the PI controller properly, the required speed performance is achieved.

The following hardware components are needed for the experimental setup:

- 3 phase DeLorenzo squirrel cage induction motor
- TMS320F28035 control card and on-board JTAG emulation via USB
- TMDSHVMTRPFCKIT (high voltage motor control kit)
- PC with Code Composer Studio CCSv6 installed
- DC Variable Dual Power Supply 0-30V
- Digital multimeter, digital oscilloscope

4.6 TMS320F28035 Control Card Controller

TMS320F28035 is a digital signal controller based on High-Performance Static CMOS Technology. The multiple bus architecture, commonly termed Harvard Bus, enables the C28035 to fetch an instruction, read a data value and write a data value in a single cycle. The main features are 32-bit (IEEE 754) fixed-point unit (FPU), six-channel DMA controller, and enhanced control peripherals. It runs at a CPU frequency up to 60MHz. It is an ideal controller for Industrial AC inverter drive applications, solar Inverter applications, etc. the use of a fixed-point library always leads to extended execution time for each mathematical instruction, which involves floating-point data type variables. For a real-time control application, this extended calculation time is not very welcome. And, to make it worse, real-time applications are quite often very cost sensitive.

we are implemented Direct Torque Control drive; has a complex mathematical equation. Running tms320f28035 controller in Realtime application to get a better solution without using floating variables. The variables used in this implementation is called IQ (Integer - Quotient) variables. These variables are efficient in the motor control application. All motor control algorisms in this kit is works based on IQ math variables and in per unit system (PU). All the mathematical equation described in chapter 6 are converted in per unit system based on the kit

and the motor base values before the algorism is coded. The code loaded to Tms320f28035 controller is worked in per unit (PU) system. Per unit, the calculation is provided in the appendix.

Development Tools

The quickest way to begin working with a C28x device is to acquire high voltage motor control kit for initial development, which, in one package, includes:

• On-board JTAG emulation via USB or parallel port

• Appropriate emulation driver

• Code Composer Studio IDE for programming

Texas Instruments provides an integrated solution called Control SUITE^{TM} for industrial control purpose which consists of necessary motor control math header files, driver-specific header files, source files, library, linker command files for developing an executable program.

4.7 Experimental Setup

The complete block diagram of direct torque control induction motor drive is shown below.

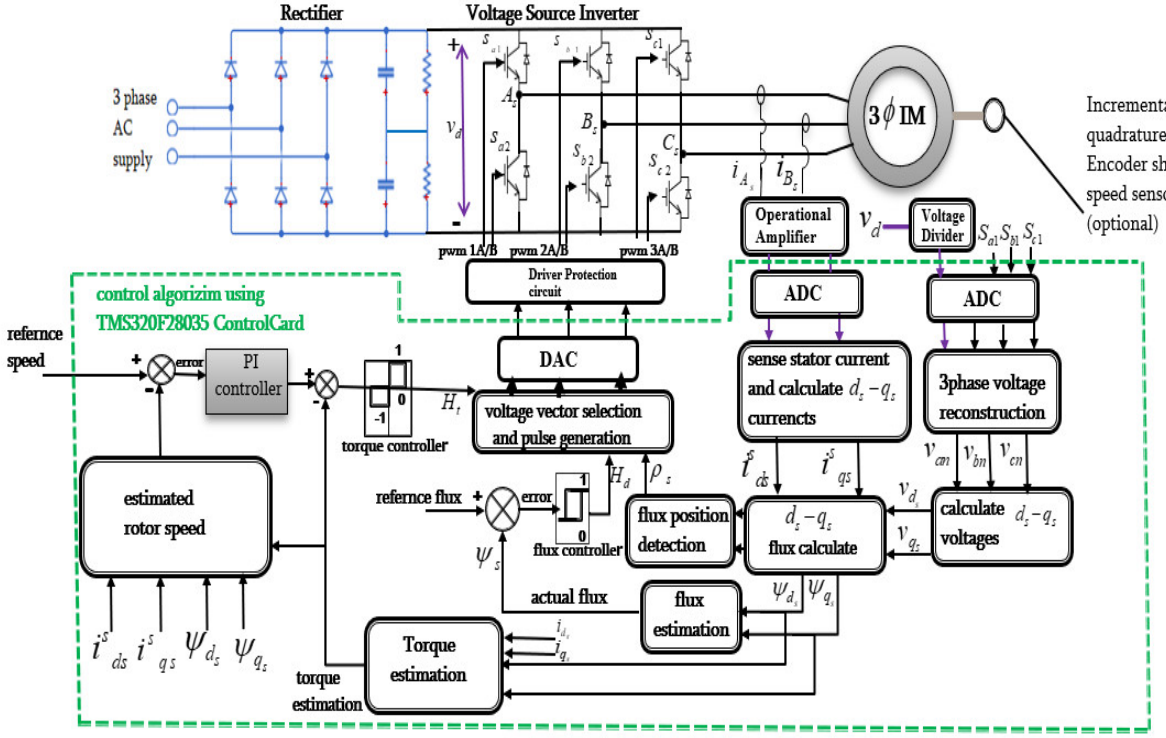


Figure 4.20: Block diagram of DTC drive algorism

The complete hardware setup for the project shown below in the figure below.

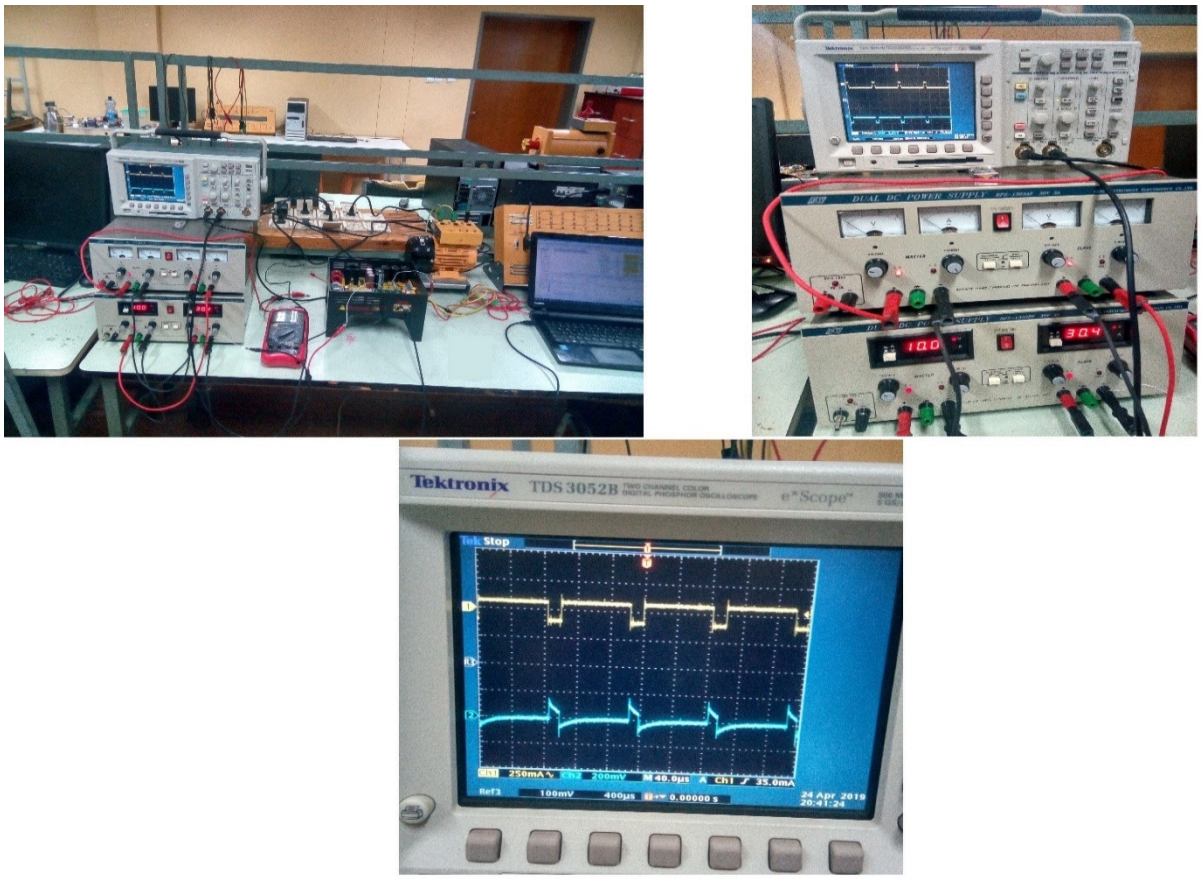


Figure 4.21: Hardware setup of DTC based induction motor drive

4.8 Experimental Results

These experimental results are obtained after the dc bus voltage set to 120volt with the reference speed of step input (0.3pu)1080rpm rotor speed. But this dc voltage is not the limited voltage,we can increace further upto 300volt. When we increase the dc bus voltage then we get 0.3pu estimated speed at 120dc bus voltage. All these real-time signals are captured from the code composer studio during the motor running at this operating point. The horizontal axis value represents sampling frequency and the vertical axis represent the variable value in per unit.

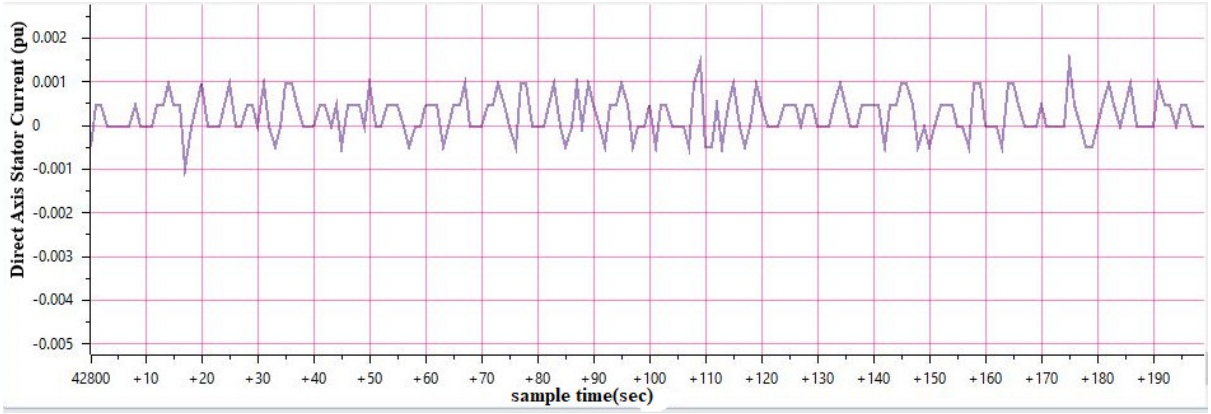


Figure 4.22: Direct axis stator current

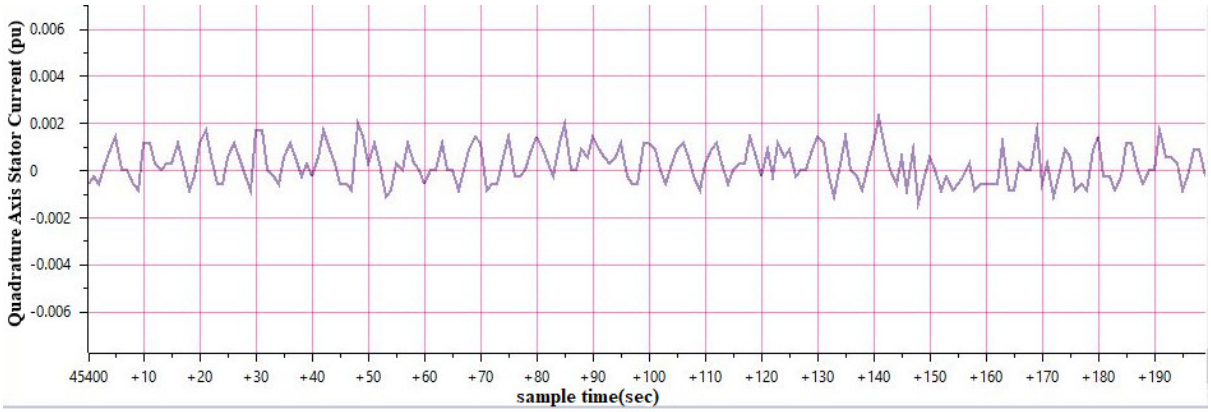


Figure 4.23 Quadrature axis stator current

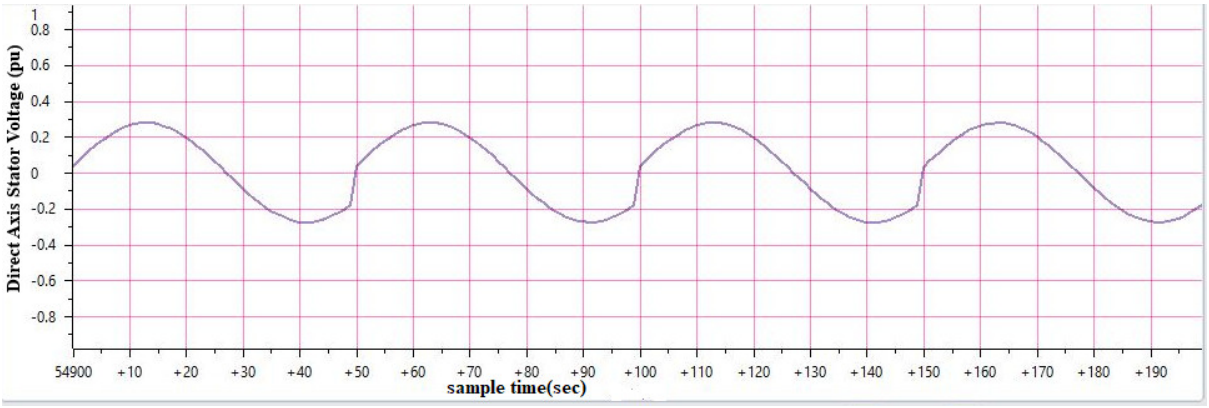


Figure 4.24: Direct axis stator phase voltage

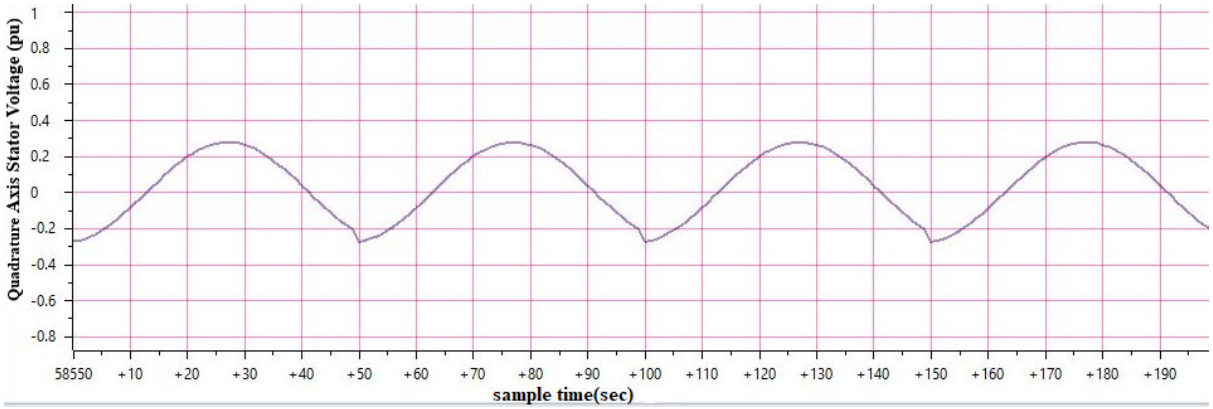


Figure 4.25: Quadrature axis stator phase voltage

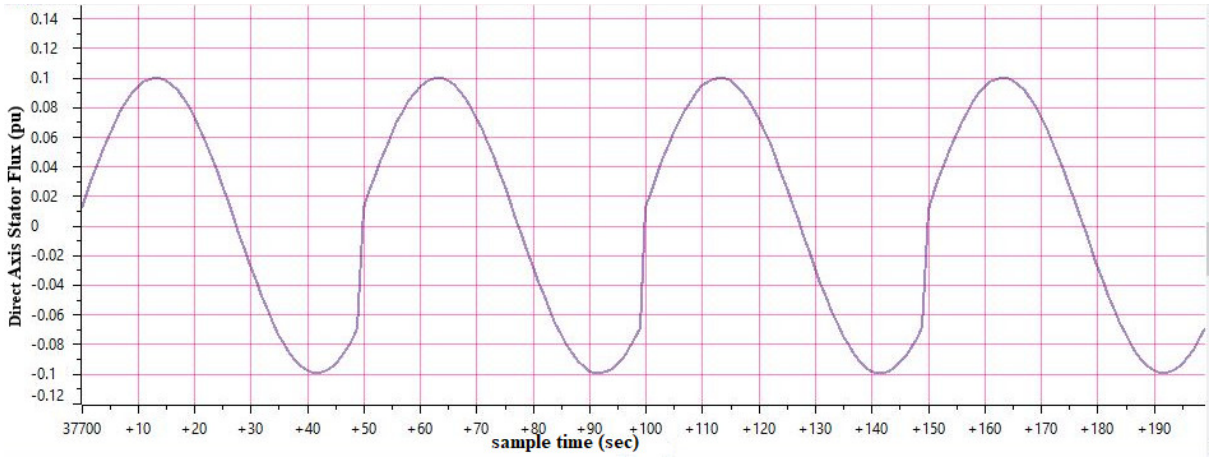


Figure 4.26: Direct axis stator flux

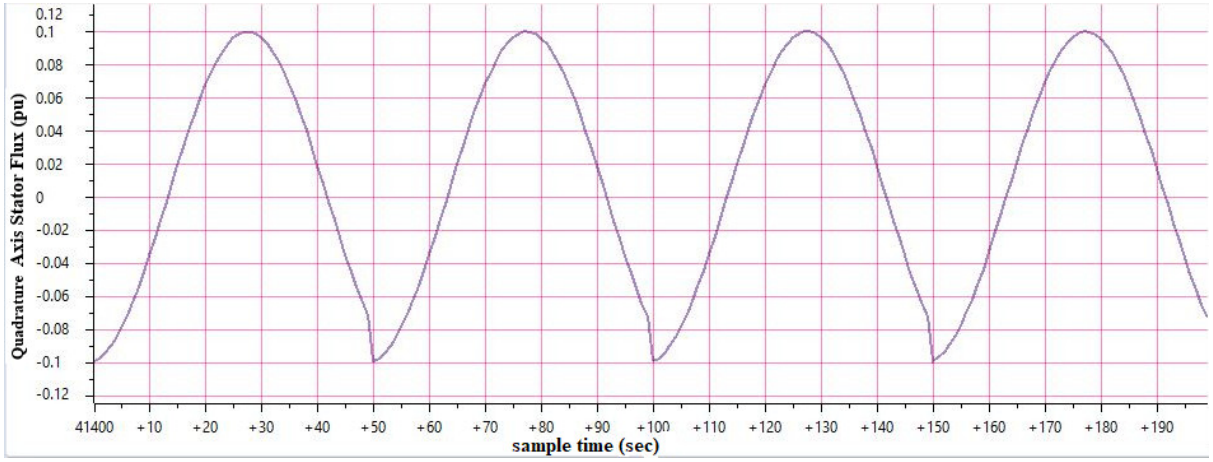


Figure 4.27: Quadrature axis stator flux

From the above figure (4.22-4.27) real-time graph of direct axis and quadrature axis variables (current, voltage, and flux) the waveforms are sinusoidal with amplitude varying. This proves the reference frame is a stationary reference frame. Also, we know the fact that the angle

between the direct axis and quadrature axis is 90^0 phase shifts. So, when we see the waveforms of direct axis stator flux and quadrature axis stator flux the angle between them has 90^0 shift, sinusoidal type waveform with an amplitude of the reference flux value (the same magnitude). from this, we have not a problem with flux estimation. further more, form the above real time ccs graph shown in figure (4.22-4.27) to get each variable values of the motor multiply the base values of each variables with correspoinding per unit values from real time graph as displayed in above figure (4.22-4.27).

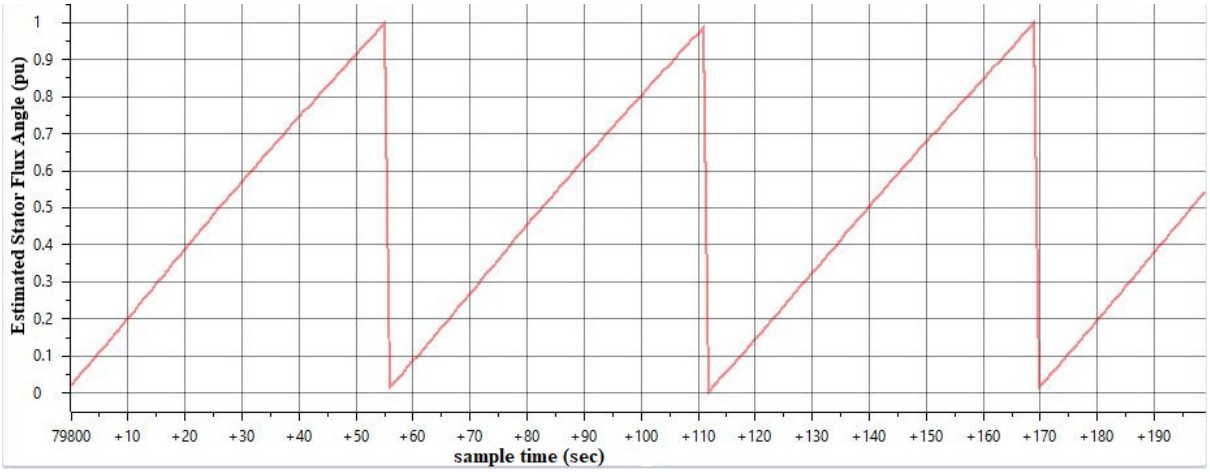


Figure 4.28: Stator flux angle

When we estimate the stator flux angle, it varies from 0 rad (0^0) to $2\pi rad(360^0)$. Since Our analysis is in per unit. atan2pu function returns the value 0 to 1 by considering 2 πrad as a base. But without considering per unit analysis this function returns the value from - π to π. This means it covers all four quadrants of the angle.

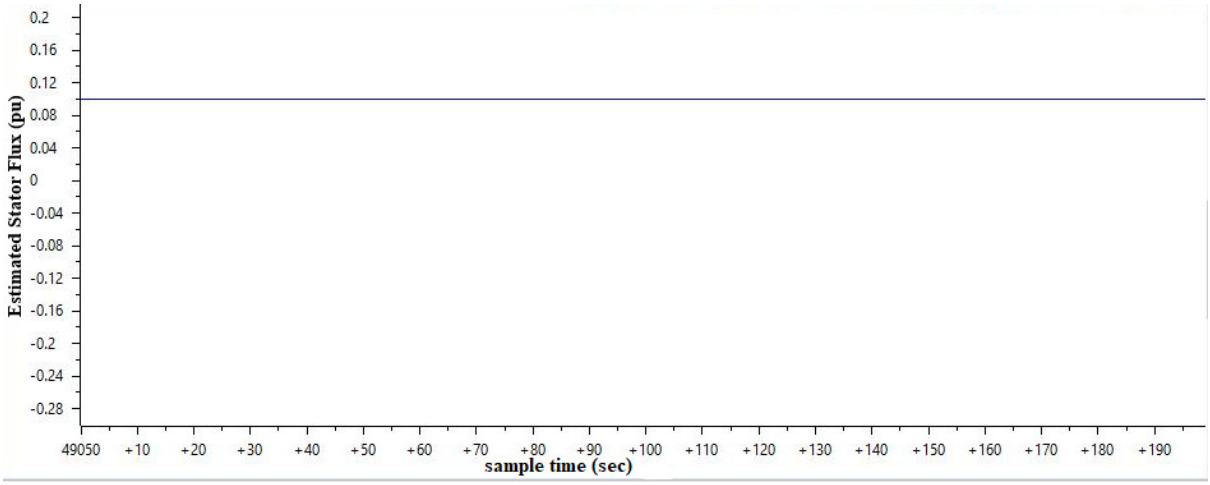


Figure 4.29: Estimated stator flux modulus

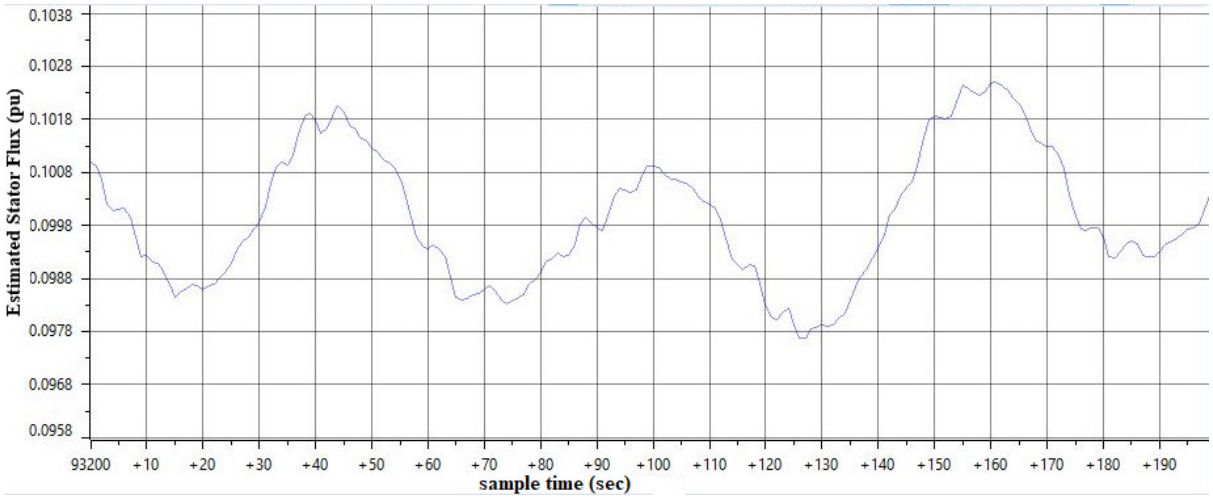


Figure 4.30: Estimated stator flux in zoom view

From figure 4.28, this real-time graph tells us the value of the estimated resultant stator flux (the square root of the estimated direct and quadrature axis stator flux). But this value is around the reference resultant stator flux value 0.1Wb. but not exactly 0.1Wb as we seen from figure 4.29, it has some fluctuation from the reference value with a maximum ripple 2.8% i.e maximum experimental result -reference value is divided by the reference value.

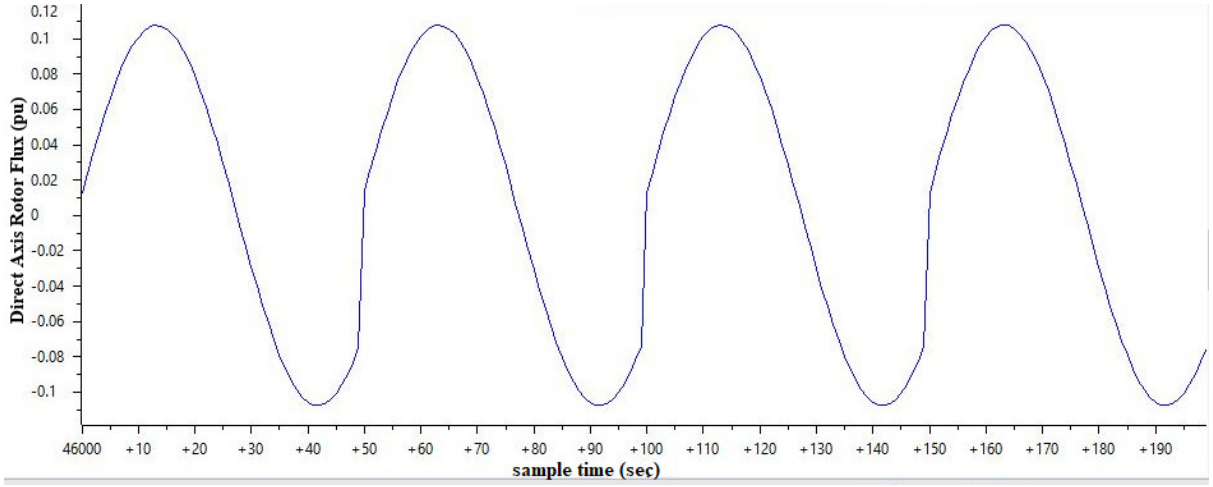


Figure 4.31: Direct axis rotor flux in the stationary reference frame

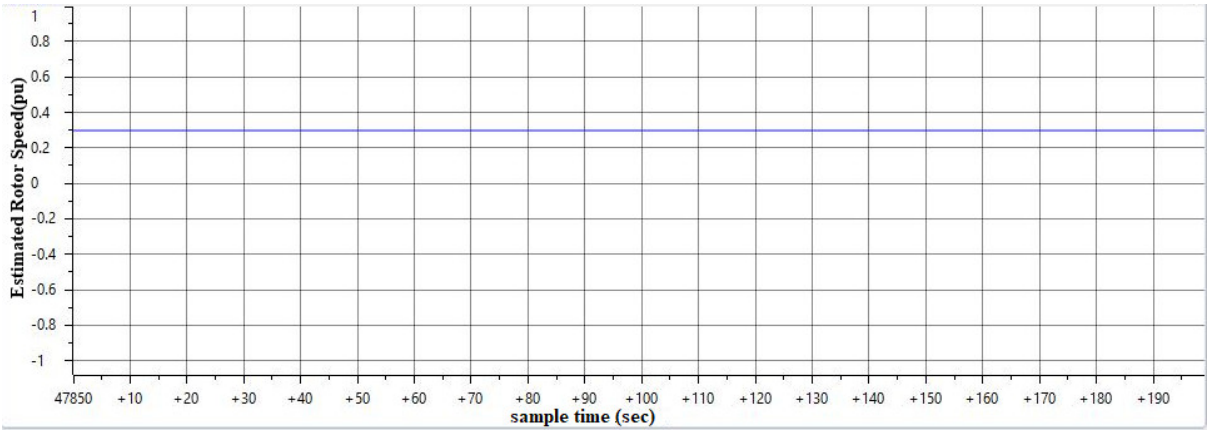


Figure 4.32: Estimated rotor speed

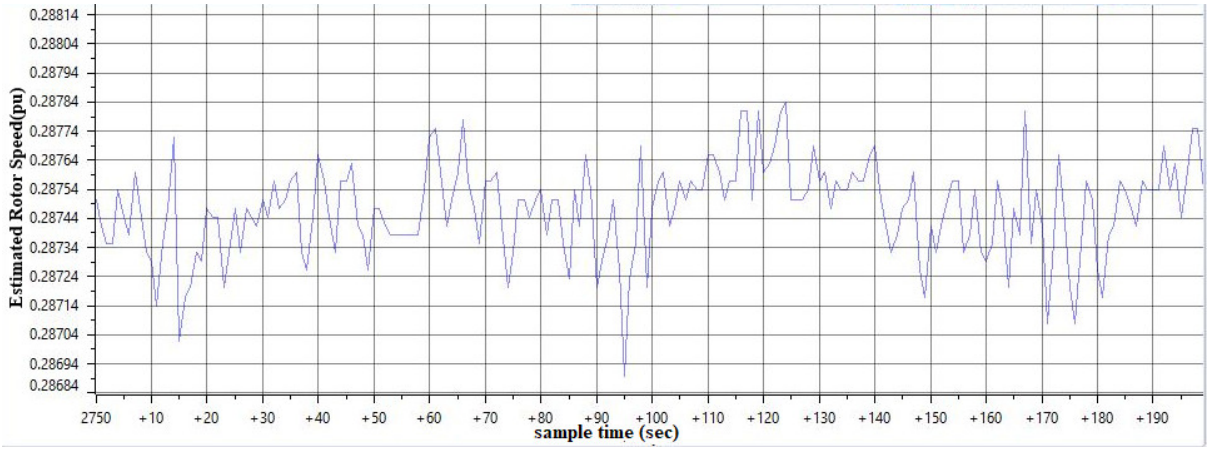


Figure 4.33: Estimated rotor speed in zoom view

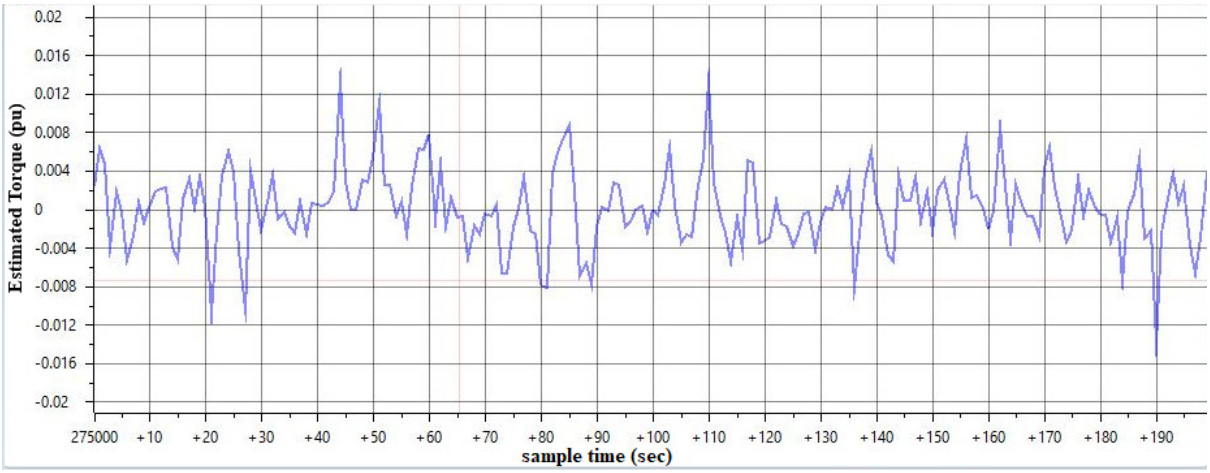


Figure 4.34: Developed electromagnetic torque

When we observe the performance of speed estimator at different values, we observe the following experimental results. The experimental results are captured from the real-time code composer graph, and we use TACHOMETER at the motor shaft to compare these results. The

dc bus voltage is set to 120volt. Starting from 0pu to approximated value 0.04pu, we do not observe the rotation of the shaft. This proves the drive sytem do not operate at low speed as we discussed.

Speed reference=0.001pu

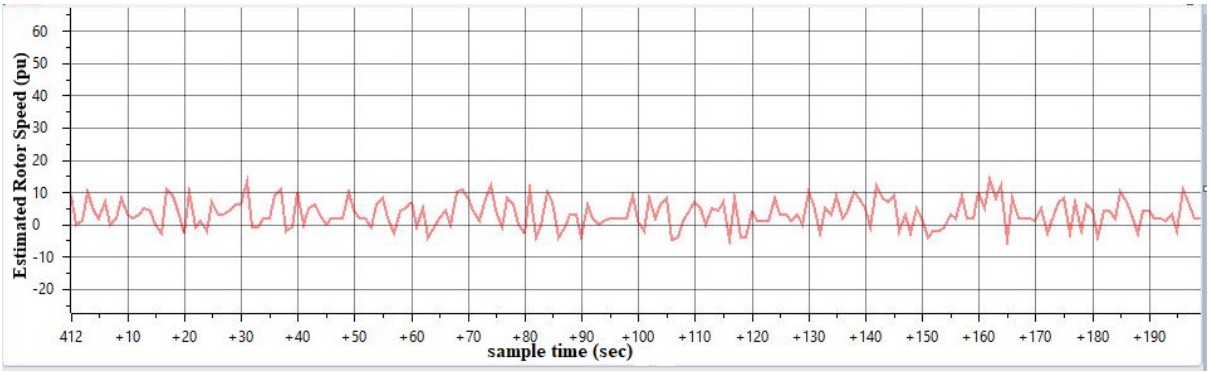
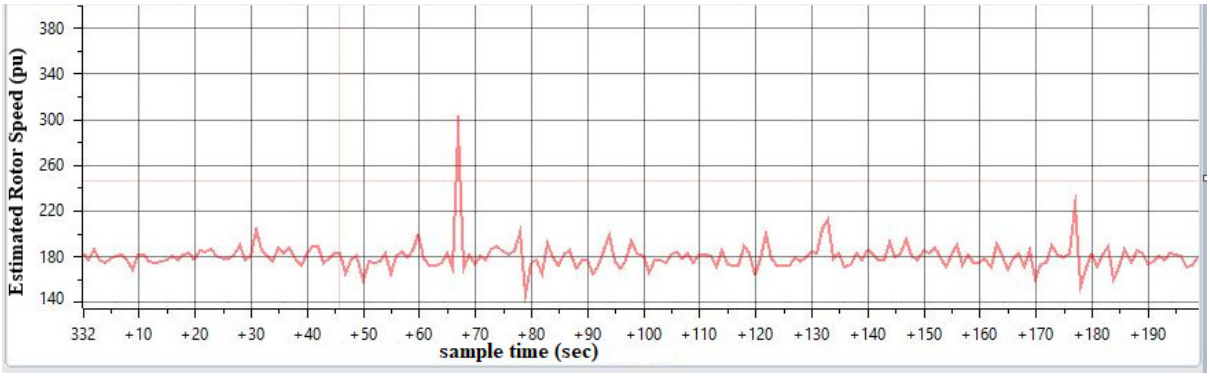
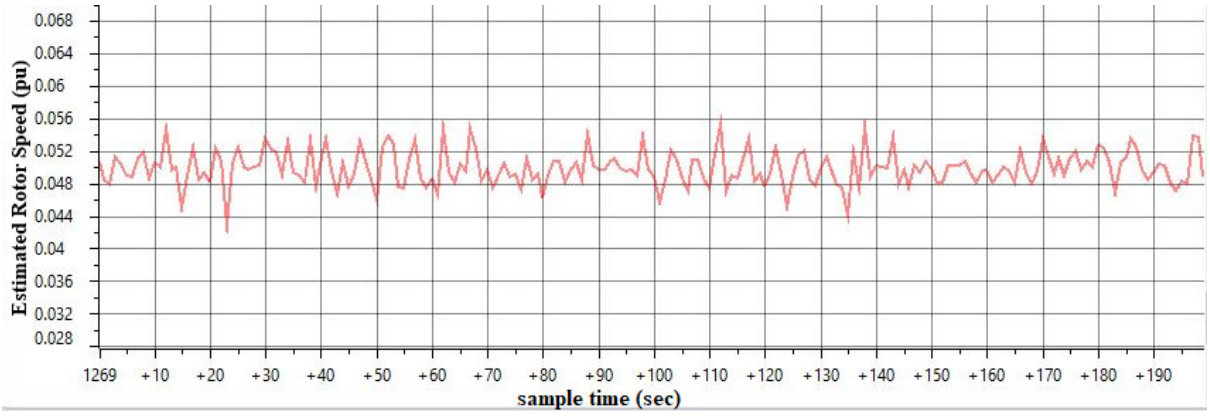


Figure 4.35: Estimated speed in rpm

Speed reference= 0.05pu



(a)

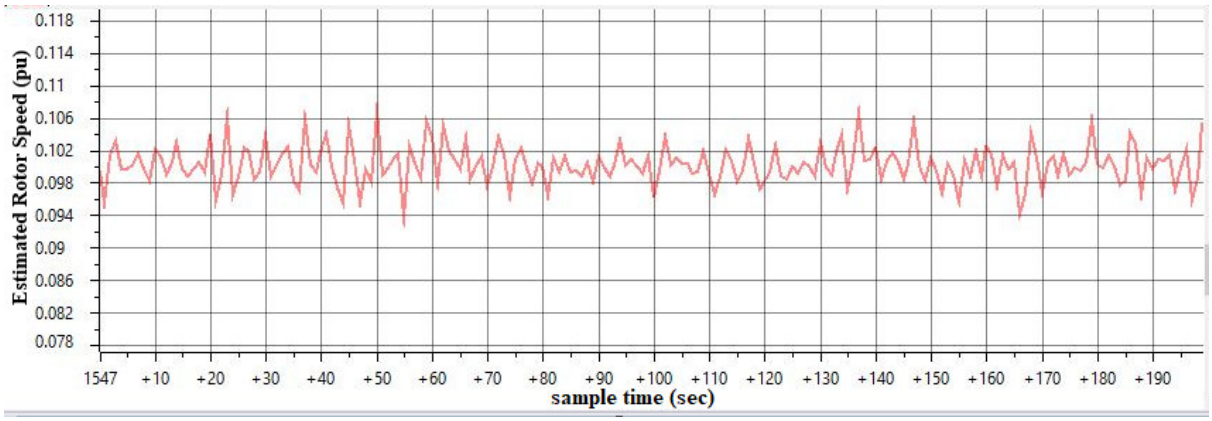


(b)

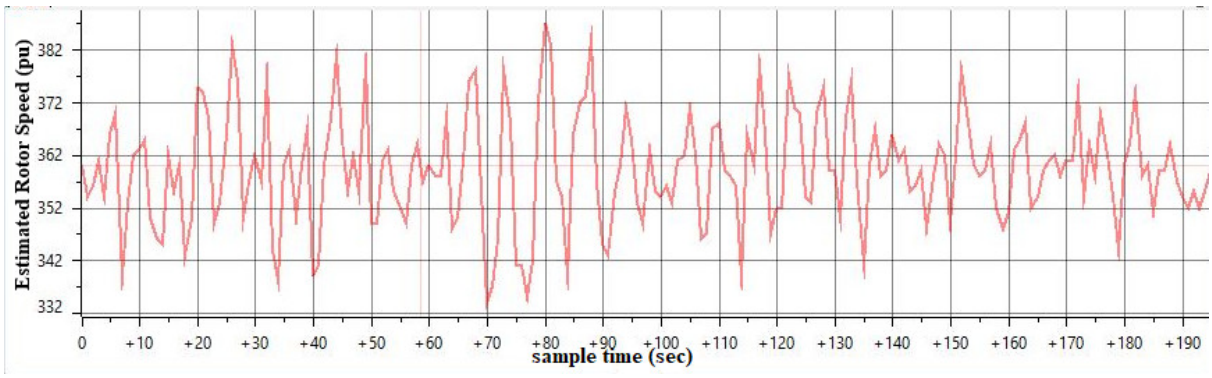
(c)

Figure 4.36: a) Per unit estimated speed b) Estimated speed in rpm c) Measured speed in rpm. The base speed is 3000rpm, 3000*0.05=150rpm. The tachometer reading is 77rpm, due to the problem of capturing.

Speed reference=0.1pu



(a)

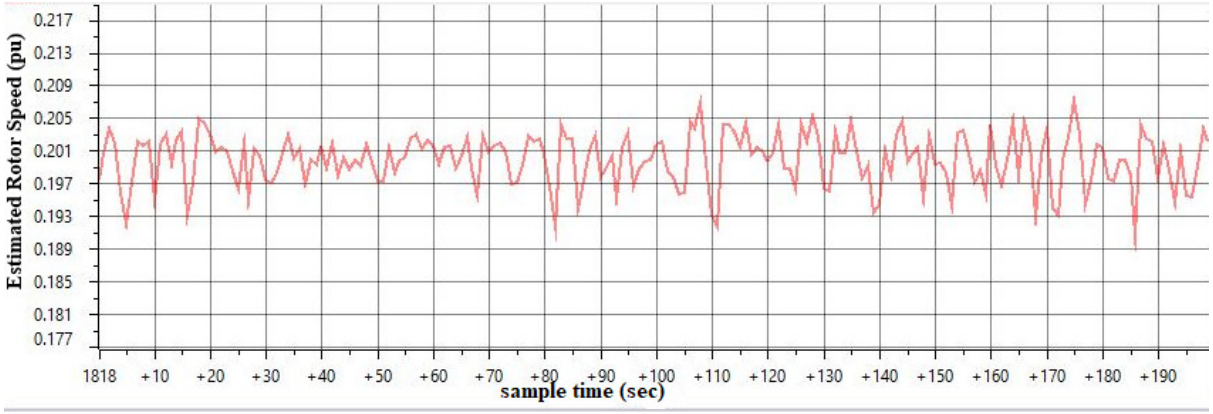


(b)

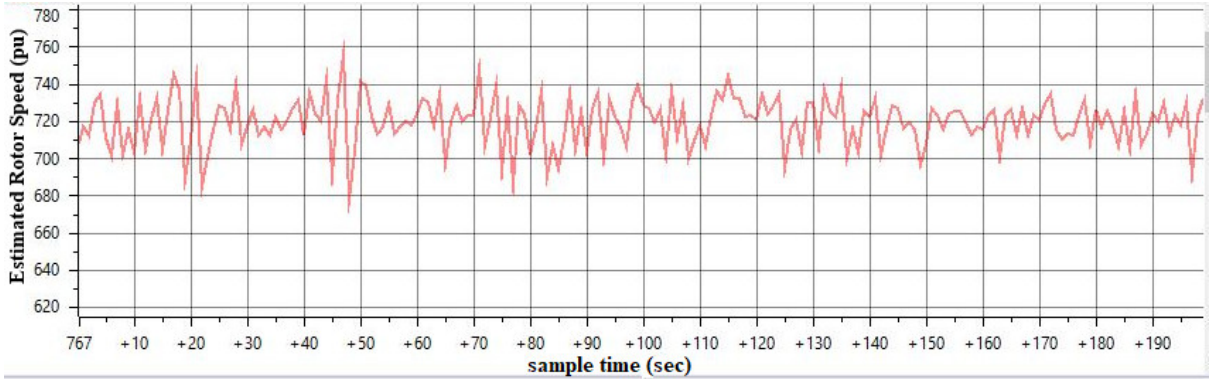
(c)

Figure 4.37: a) Per unit estimated speed b) Estimated speed in rpm c) Measured speed in rpm

Speed reference=0.2pu



(a)

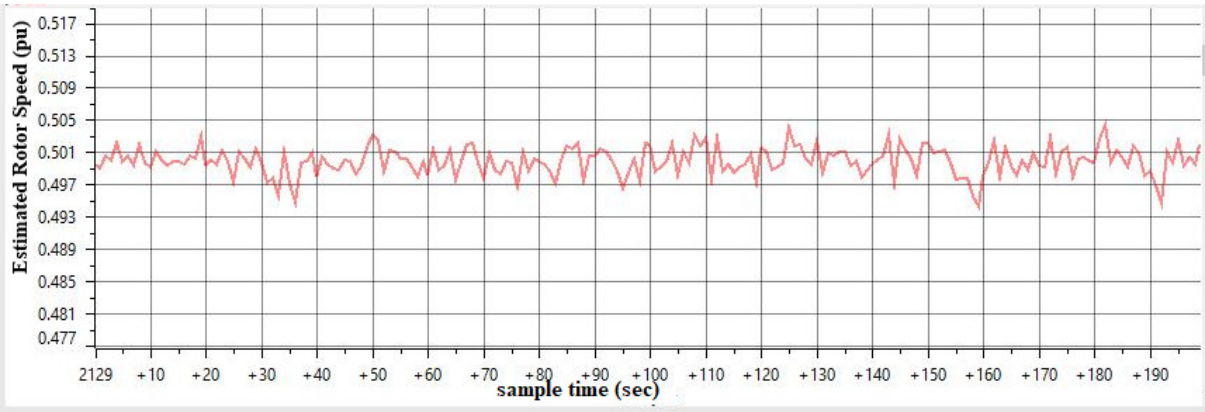


(b)

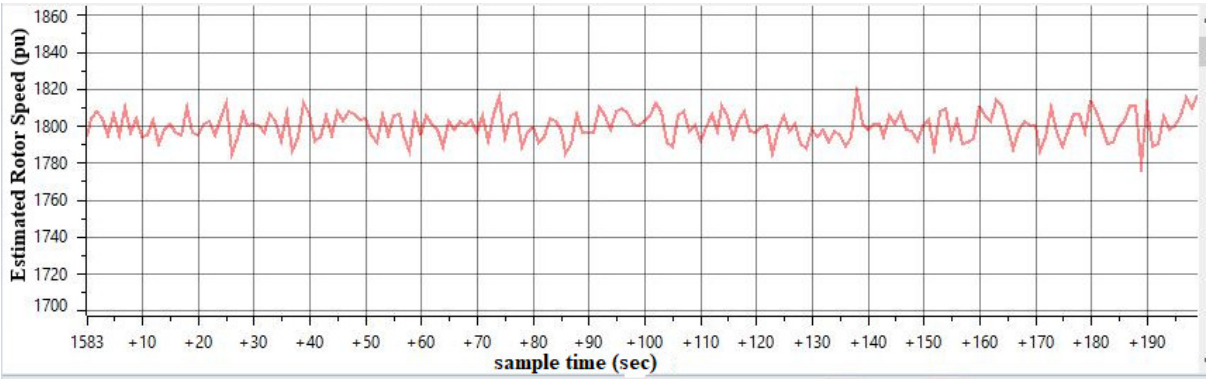
(c)

Figure 4.38: a) Per unit estimated speed b) Estimated speed in rpm c) Measured speed in rpm

Speed reference =0.5pu



(a)



(b)

(c)

Figure 4.39: a) Per unit estimated speed b) Estimated speed in rpm c) Measured speed in rpm

The speed of 0.5pu; 3000*0.5=1500rpm. Almost tachometer reading is the same as the calculated value. Generally, From the above result when we compare the code composer real-time graph with that of tachometer value, we see both values are approximately equal. There is fluctuation in the estimated speed, as we observe this in the real-time graph. Also, the tachometer reading does not give the constant speed values (it varies). This indicates the rotor shaft speed is fluctuating. The problems in reading the rotation of shaft speed are Capturing the tachometer reading values is hard.

Also, at lower speed range we observed more fluctuation of speed as we see from the real-time graph, due to the integration problem we do not get the proper value of estimated parameters (flux, torque), this degrades the performance of speed estimation. In addition, the PI controller gain is fixed.

CHAPTER 5

Conclusion and Future Work

5.1 Conclusion

the presented work deals with the description of the principle of direct torque-controlled induction motor drive without using a speed sensor (using speed estimation techniques). The stationary frame of reference is used for our analysis during design and modeling our system in MATLAB/Simulink toolbox and in code composer studio c programming direct torque-controlled drive algorism. The $d_s - q_s$ modeling procedure and the modeling equation in the direct torque-controlled induction motor drive with a detailed discussion of the result obtained from the simulation model are presented. Also discussed the detailed results obtained from the implementation aspect of the direct torque-controlled induction motor drive with speed estimation techniques.

The only parameter used during stator flux estimation is stator resistance and accuracy is affected at low stator voltage; the voltage drop on resistance is higher. This is the case at low speed because the dynamic operation is very poor at such condition. the accuracy of estimation of stator flux is very high; it uses direct incoming stator currents and voltages and this also does not have a complex mathematical equation.

We could also demonstrate that DTC allows decoupling control of stator flux linkage and electromagnetic torque. Stator flux linkages are proportional to the applied voltage. Also, when we set reference speed at different values the stator flux must not be changed with respect to the reference speed. This indicated decoupling control of flux and torque; controlling torque means indirectly controlling speed.

Simulation and experimental results show that the DTC system based on the voltage model of flux linkage can realize the rotation speed of the stator flux linkage by applying the voltage vector while maintaining the constant value of the stator flux linkage with different motor speed adjustment.

The tuning mechanism of the PI controller used in Simulink is by using trial and error. This makes worse the performance of the rotor speed. from MATLAB Simulink the performance of speed response will be further improved by using an appropriate tuning mechanism of PI controller gains.

The inadequacies of the experimental results are obvious. This also happened in the simulation results. There is a fluctuation of estimated variables such as stator flux, torque, and speed are observed in real time graph.

DTC allows us to good torque control in transient as well as in steady state operating condition. We can observe this effect from real-time ccs graph. All the estimation in DTC drive is based on the measured and sensed two stator currents and dc bus voltages, this makes even our system very simple. The phase voltages are reconstructed from the dc bus voltage and the generated PWM signal from microcontroller using code. This technique eliminates the phase voltage sensor. This generated PWM signal also uses the motor to spin in the desired and designed rotor speed.

When we see the MATLAB Simulink results and the real-time ccs graph, the performance of transient and steady-state response is good, so the direct torque control drive algorism can work safely without damaging our motor.

the flux and torque controller are based on hysteresis controller; the switching states of the inverter affected by this comparator. We may get unpredictable switching frequency in the PWM signal until our system gets a steady state.

In short, as a new type of modern AC motor control technology, DTC technology does have its own unique advantages, but there are still many problems to be solved in the future either in theory or practice.

5.2 Future Work

- By using the speed sensor (incremental encoder) the performance of speed will be compared.
- Replacing the PI controller by fuzzy or artificial neural network or combination of the two controllers the performance will be compared.
- By increasing the switching frequency of the inverter; increasing the sampling time; this makes the estimation accuracy.
- By using different types of stator resistance compensation techniques, increase the performance at low-speed operation.

Reference

[1] P. Vas, "Sensorless Vector and Direct Torque Control", Oxford University Press, 1998.

[2] Paul C. Krause, Oleg Wasynzcuk and Scott D. Sudhoff, “Analysis of Electric Machinery and Drive Systems”, 2nd ed., IEEE power Series, Inc., 2002.

[3] Bimal K. Bose, "Modern Power Electronics and Ac Drives", 1st ed., Bernard Goodwin, Ed. USA: Prentice Hall, Inc., 2002.

[4] Hoang L.H, “Comparison of field-oriented control and direct torque control for induction motor drives”, Thirty-Fourth Industrial Applications Society Annual Meeting/Industry Applications Conference, Vol.2, pp. 12451252, October 1999.

[5] Chee-Mun Ong, "Dynamic Simulation of Electric Machinery Using MATLAB Simulink”, Prentice Hall, Inc.,1998.

[6] M. Cirrincione, M. Pucci, and G. Vitale, "Power Converters and AC Electrical Drives with Linear Neural Networks", CRC Press,2012.

[7] Ned Mohan, "Advanced Electric Drives: Analysis, Control, and Modeling Using MATLAB/Simulink", 1st ed. John Wiley & Sons, Inc., 2014.

[8] Andrzej M. Trzynadlowski, "Control of Induction Motors", David J. Irwin, Ed. San Diego, USA: Academic press, 2001.

[9] A. E. Fitzgerald, Charles Kingsley, Jr., Stephen D. Umans, "Electric Machinery",6th ed., McGraw-Hill Companies, Inc., 2003.

[10] Stephen J. Chapman, " Electric Machinery Fundamentals",5th ed., McGraw-Hill Education, Inc.,2011

[11] Tze-Fun Chan and Keli Shi, "Applied Intelligent Control of Induction Motor Drives",1st ed., John Wiley & Sons (Asia) Pte Ltd, 2011.

[12] A K. Mandal, "Introduction to Control Engineering Modeling, Analysis and Design ", New Age International (P) Ltd.,2006.

Appendix

Appendix A: Per unit calculation used in algorism

From specification Maximum inverter current 10A. base current is 10A. induction motor phase current limited to 1Amp. when we sense the two-phase currents; limited to 0.1Amp. dc bus voltage is adjusting proportionally base on the voltage divider. So, we get sensed phase current and dc voltage in per unit values. Then we must be put our equation in per unit format as follows.

$$\psi_{base} = \mathrm{L_m(i_{base})} \quad \text{A.1}$$

$$T_{base} = \frac{3(v_{base})(i_{base})Pole}{4\pi(f_{base})} \quad \text{A.2}$$

$$\frac{d\psi_{d_s}(t)}{dt} = v^s{}_{d_s} - R_s i_{d_s}(t) \quad \text{A.3}$$

$$\psi_{base}\frac{d\psi_{d_s\,pu}(t)}{dt} = v_{base}(v^s{}_{d_s\,pu}) - i_{base}(R_s)(i_{d_s\,pu}(t)) \quad \text{A.4}$$

$$\frac{d\psi_{q_s}(t)}{dt} = v^s{}_{q_s} - R_s i_{q_s}(t) \quad \text{A.5}$$

$$\psi_{base}\frac{d\psi_{q_s\,pu}(t)}{dt} = v_{base}v^s{}_{q_s\,pu} - i_{base}R_s i_{q_s\,pu}(t) \quad \text{A.6}$$

$$\rho_s = a\tan 2PU\,\frac{\psi_{q_s\,pu}(t)}{\psi_{d_s\,pu}(t)} \quad \text{A.7}$$

$$T_e = \frac{3}{2}\frac{p}{2}(\psi_{d_s} i_{q_s} - \psi_{q_s} i_{d_s}) \quad \text{A.8}$$

$$T_{base}T_{e\,pu} = \frac{3}{2}\frac{p}{2}(\psi_{base}(\psi_{d_s\,pu})(i_{q_s\,pu})i_{base} - \psi_{base}(\psi_{q_s})(i_{d_s\,pu})i_{base}) \quad \text{A.9}$$

$$\omega_{base}(\omega_{slip\,pu}) = \frac{2R_r(T_{estimated\,pu})T_{base}}{polepair(3)(\psi_{r\,pu}\psi_{base})^2} \quad \text{A.10}$$

$$\omega_{base}(\omega_{syn\,pu}) = 2\pi\left(\frac{d\theta_{s\,pu}}{dt}\right) \quad \text{A.11}$$

$$\omega_{base}(\omega_{syn\,pu}) = \frac{(\psi_{base})^2(\psi_{d_r\,pu}(t)\frac{d}{dt}\psi_{q_r\,pu}(t) - \psi_{q_r\,pu}(t)\frac{d}{dt}\psi_{d_r\,pu}(t))}{(\psi_{base}\psi_{r\,pu})^2} \quad \text{A.12}$$

Appendix B: DTC programming code in code composer studio using c language

```
//===========================================================================
void init28035(void)
{
InitSysCtrl();
Gpio_select();
DINT;//put next to wdogdisable inside in sys control
InitPieCtrl();
IER = 0x0000;//put next to wdogdisable inside in sys control
IFR = 0x0000;//put next to wdogdisable inside in sys control
InitPieVectTable();
init_adc();
}
void adc_signals(void)
{
AdcRegs.ADCSOCFRC1.all = 0x00FF;  // Force Start SOC0-7 to begin ping-pong
sampling
while (AdcRegs.ADCINTFLG.bit.ADCINT2 == 0){}
AdcRegs.ADCINTFLGCLR.bit.ADCINT2 = 1; //Must clear ADCINT1 flag since INT1CONT = 0
ADCINA1=AdcResult.ADCRESULT1;//
ADCINB1=AdcResult.ADCRESULT2;
ADCINA7=AdcResult.ADCRESULT3;
//Disable ADCINT1 and ADCINT2 to STOP the ping-pong sampling
AdcRegs.INTSEL1N2.bit.INT1E = 0;
AdcRegs.INTSEL1N2.bit.INT2E = 0;
// Phase voltages are measured and fed to ADC pins
ia1 = ((AdcResult.ADCRESULT1)-0x07FF)*0.00048828125;// GIVES THE VALUE B/N
0.0495&-0.0495 and then 0.03&-0.03
ib1 = ((AdcResult.ADCRESULT2)-0x07FF)*0.00048828125;
ic1 = -(ia1+ib1);
// Scaling of phase currents ia, ib and ic
//ia1 = (ia*0.6060606061);
//ib1 = (ib*0.6060606061);
// DC Link voltage is measured and fed to ADC pin
Vdc= (AdcResult.ADCRESULT3)*0.00024414;//  1/4 we get voltage b/n 0.5
// Scaling of DC Link voltage
//Vdc = V;
}
//===========================================================================
```

```
void Phase_voltage_calculation(void)
{
//PHASE VOLTAGES
Van=(2*Sa1-Sb1-Sc1)*Vdc*onethird;
Vbn=(2*Sb1-Sa1-Sc1)*Vdc*onethird;
Vcn=(2*Sc1-Sa1-Sb1)*Vdc*onethird;
}
//=============================================================================
void stationary_ref_frame_transformation(void)
{
// Three phase to two-phase transformation
Vds= twothrd*(Van-(0.5*Vbn)-(0.5*Vcn));
Vqs=0.57735*(Vbn-Vcn);
// Three phase to two-phase transformation
ids=twothrd*(ia1-(0.5*ib1)-(0.5*ic1));
iqs=0.57735*(ib1-ic1);
}

void flux_torque_calculation(void)
{
// On integrating d-axis and q-axis components of stator voltages
d_flux=((((Vds*80.5316749)-(ids*11.05*3.4021))*0.0001)+d_flux)*0.9995;
q_flux=((((Vqs*80.5316749)-(iqs*11.05*3.4021))*0.0001)+q_flux)*0.9995;
square=(d_flux*d_flux)+(q_flux*q_flux);
// Actual stator flux
d_act=(sqrt(square));

/ Actual torque T= (3/2)(P/2) (iqs*d_flux - ids*q_flux)
T_act = 3.90107096*((iqs*d_flux)-(ids*q_flux));

void sector_calculation(void)
{
theta = atan2(q_flux,d_flux);
//theta=((angle*0.318309886)*180);
if ((theta >= -PI/6) && (theta < PI/6))
{
sector = 1;
}
else if ((theta >=PI/6) && (theta <PI/2))
{
sector = 2;
```

```
}
else if ((theta >=PI/2) && (theta < (5*PI)/6))
{
sector = 3;
}
else if ((theta >=(5*PI)/6) && (theta <PI))
{
sector = 4;
}
else if ((theta > -PI) && (theta <(5*PI)/6))
{
sector = 4;
}
else if ((theta >= (-5*PI)/6) && (theta <-PI/2))
{
sector = 5;
}
else if ((theta >= -PI/2) && (theta < -PI/6))
{
sector = 6;
}
}

//=============================================================================
void flux_hysteresis(void)
{
err_flux= fluxref - d_act;
// Two level hysteresis
if(err_flux > (fluxref*0.01))
Hd=1;
else if(err_flux < -(fluxref*0.01))
Hd=0;
}

//=============================================================================
void Torque_hysteresis(void)
{
err_t= PI_out - T_act;
// Three level hysteresis
if(err_t > 0.01)
Ht=1;
```

```
else if(err_t < -0.01)
Ht=-1;
else if((Ht==1) && (0 < err_t < 0.01))
Ht=1;
else if((Ht==-1) && (0 < err_t < 0.01))
Ht=0;
else if((Ht==-1) && (-0.01 < err_t < 0))
Ht=-1;
else if((Ht==1) &&(-0.01 < err_t < 0))
Ht=0;
}

//=============================================================================
void speed_calculation(void)
{
if (speed_ref_f < speed_ref)
{
speed_ref_f = speed_ref_f + 0.01;
}
else if (speed_ref_f > speed_ref)
{
speed_ref_f = speed_ref_f - 0.01;
}
else speed_ref_f=speed_ref;
//estimation of speed calculation
phidr=T6*d_flux+T5*ids;
phiqr=T6*q_flux+T5*iqs;
phirsqr=(phidr*phidr)+(phiqr*phiqr);
wsyn=((phiqr*phidr)-(phidr*phiqr))/(Ts*(phirsqr+T4));
wsyne=T1*wsyn+T2*wsyne;
wslip=(T_act*T3)/(phirsqr+T4);
wrotor=wsyne-wslip;
if(wrotor>=1) wrotor=1;
else if(wrotor<=-1) wrotor=-1;
else wrotor=wrotor;
}

//=============================================================================

void pi_controller(void)
{
```

```
err_speed=speed_ref_f-wrotor;
prop=Kp*err_speed;
ui=ui+Ki*prop+kc*saterr;
outpresat=prop+ui;
if(outpresat>=i_max)
PI_out=i_max;
else if(outpresat<=i_min)
PI_out=i_min;
else
PI_out=outpresat;
saterr=PI_out-outpresat;
if(PI_out>=pi_max)
{
PI_out = pi_max;
}
else if(PI_out <=pi_min)
{
PI_out = pi_min;
}
else PI_out=PI_out;
}
//=============================================================================
void switching_table(void)
{
out=0;
// Switching of 8 voltage vectors V0,V1,V2,V3,V4,V5,V6,V7
// active voltage vectors = {V2,V3,V4,V5,V6,V1};
if ((Hd==1) && (Ht==1))
{
temp[0]=2;
temp[1]=3;
temp[2]=4;
temp[3]=5;
temp[4]=6;
temp[5]=1;
}
// zero voltage vectors = {V7,V0,V7,V0,V7,V0};
else if ((Hd==1) && (Ht==0))
{
temp[0]=7;
```

```
temp[1]=0;
temp[2]=7;
temp[3]=0;
temp[4]=7;
temp[5]=0;
}
// active voltage vectors = {V6,V1,V2,V3,V4,V5};
else if ((Hd==1) && (Ht==-1))
{
temp[0]=6;
temp[1]=1;
temp[2]=2;
temp[3]=3;
temp[4]=4;
temp[5]=5;
}
// active voltage vectors = {V3,V4,V5,V6,V1,V2};
else if ((Hd==0) && (Ht==1))
{
temp[0]=3;
temp[1]=4;
temp[2]=5;
temp[3]=6;
temp[4]=1;
temp[5]=2;
}
// zero voltage vectors = {V0,V7,V0,V7,V0,V7};
else if ((Hd==0) && (Ht==0))
{
temp[0]=0;
temp[1]=7;
temp[2]=0;
temp[3]=7;
temp[4]=0;
temp[5]=7;
}
// active voltage vectors = {V5,V6,V1,V2,V3,V4};
else if ((Hd==0) && (Ht==-1))
{
temp[0]=5;
```

```
temp[1]=6;
temp[2]=1;
temp[3]=2;
temp[4]=3;
temp[5]=4;
}
// zero voltage vectors = {V7,V0,V7,V0,V7,V0};
else
{
temp[0]=7;
temp[1]=0;
temp[2]=7;
temp[3]=0;
temp[4]=7;
temp[5]=0;
}
// According to the sector, the voltage vector is selected using look-up table
// The selected voltage vector is given to variable 'out'
switch (sector)
{
case 1:
{
out=temp[0];
break;
}
case 2:
{
out=temp[1];
break;
}
case 3:
{
out=temp[2];
break;
}
case 4:
{
out=temp[3];
break;
}
```

```
case 5:
{
out=temp[4];
break;
}
default:
out=temp[5];
}
// 'out' is a variable storing each element of array 'temp'.
switch(out)
{
case 1:
{  // pulses for phases A,B,C = {1,0,0};
pulse[0]=1;
pulse[1]=0;
pulse[2]=0;
break;
}
case 2:
{  // pulses for phases A,B,C = {1,1,0};
pulse[0]=1;
pulse[1]=1;
pulse[2]=0;
break;
}
case 3:
{  // pulses for phases A,B,C = {0,1,0};
pulse[0]=0;
pulse[1]=1;
pulse[2]=0;
break;
}
case 4:
{  // pulses for phases A,B,C = {0,1,1};
pulse[0]=0;
pulse[1]=1;
pulse[2]=1;
break;
}
case 5:
```

```
{  // pulses for phases A,B,C = {0,0,1};
pulse[0]=0;
pulse[1]=0;
pulse[2]=1;
break;
}
case 6:
{  // pulses for phases A,B,C = {1,0,1};
pulse[0]=1;
pulse[1]=0;
pulse[2]=1;
break;
}
case 7:
{  // pulses for all the phases = {1,1,1};
pulse[0]=1;
pulse[1]=1;
pulse[2]=1;
break;
}
default:
{  // no pulse given
pulse[0]=0;
pulse[1]=0;
pulse[2]=0;
// the signal below sa1 sb1 and sc1 determines the switching states of the
inverter
// Switching pulses for phase-A
Sa1=pulse[0];
// Switching pulses for phase-B
Sb1=pulse[1];
// Switching pulses for phase-C
Sc1=pulse[2];
```

Appendix C: Block diagrams in MATLAB simulink system

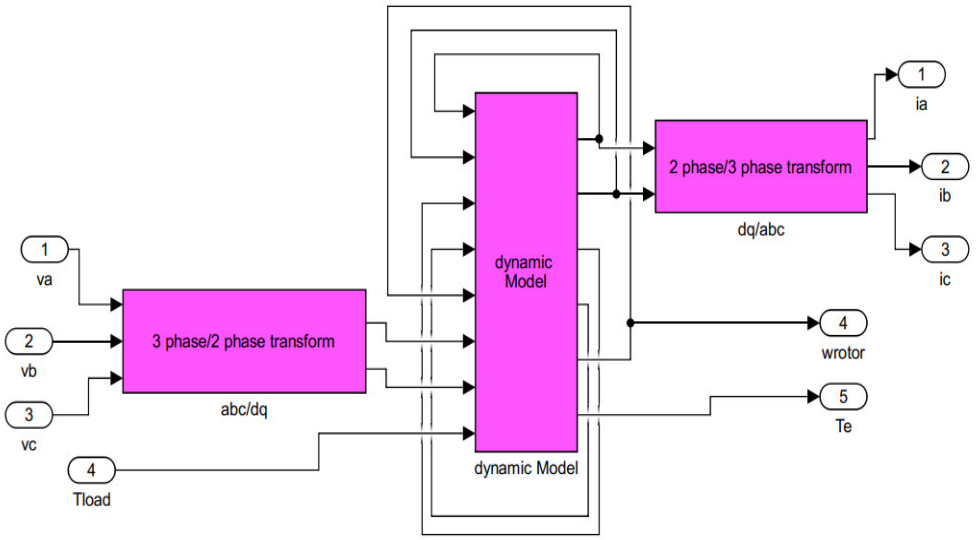


Figure C.1: Induction motor modeling in the stationary reference frame

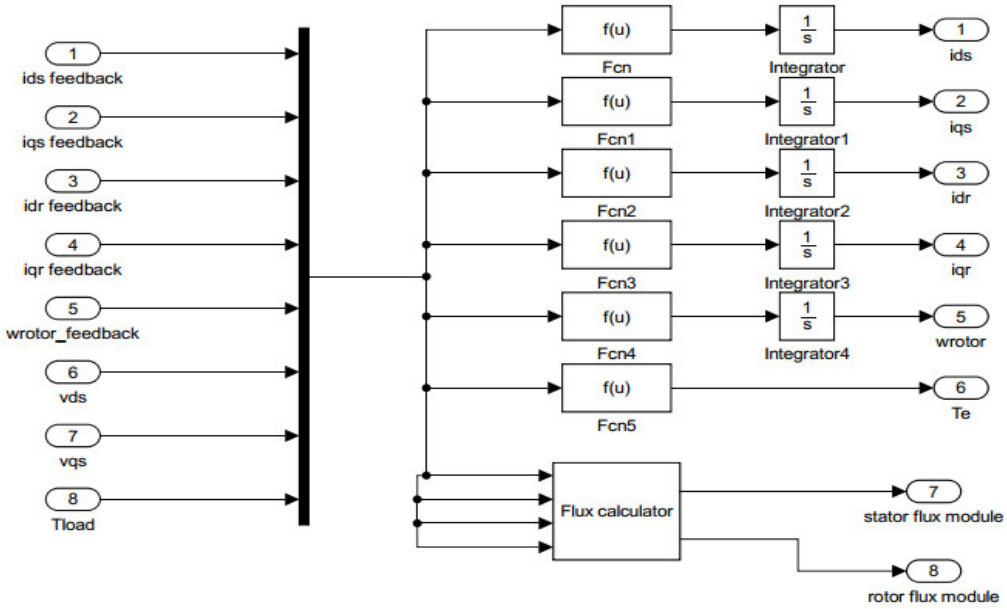


Figure C.2: Dynamic model of induction motor

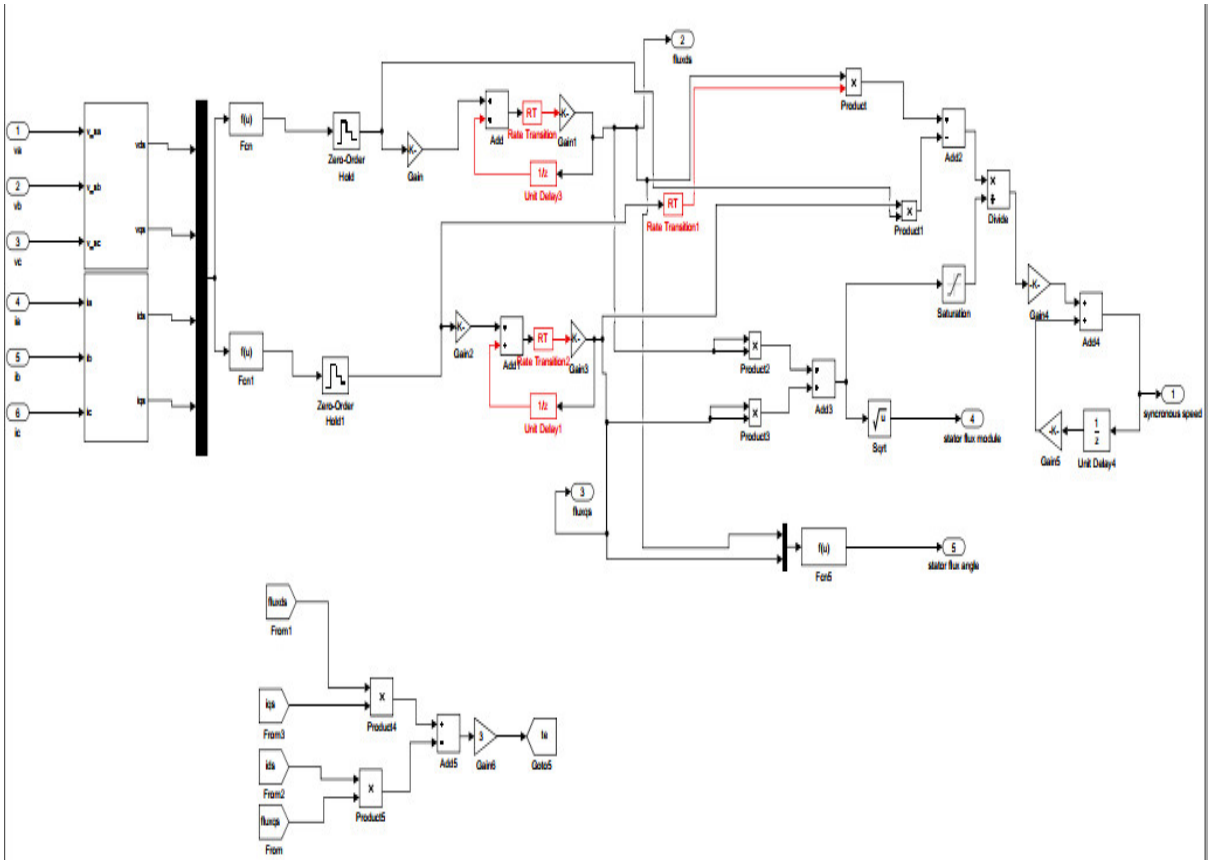


Figure C.3: Stator flux and torque estimation

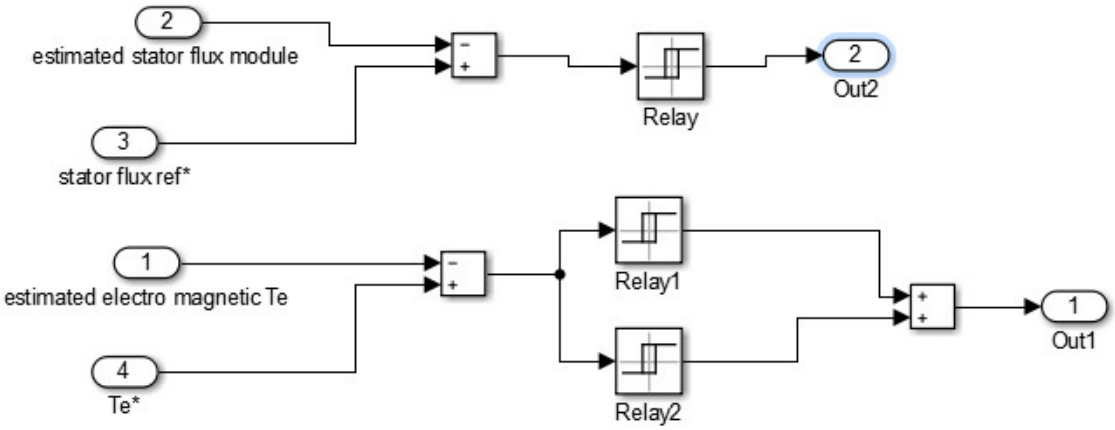


Figure C.4: Flux and torque hysteresis comparator

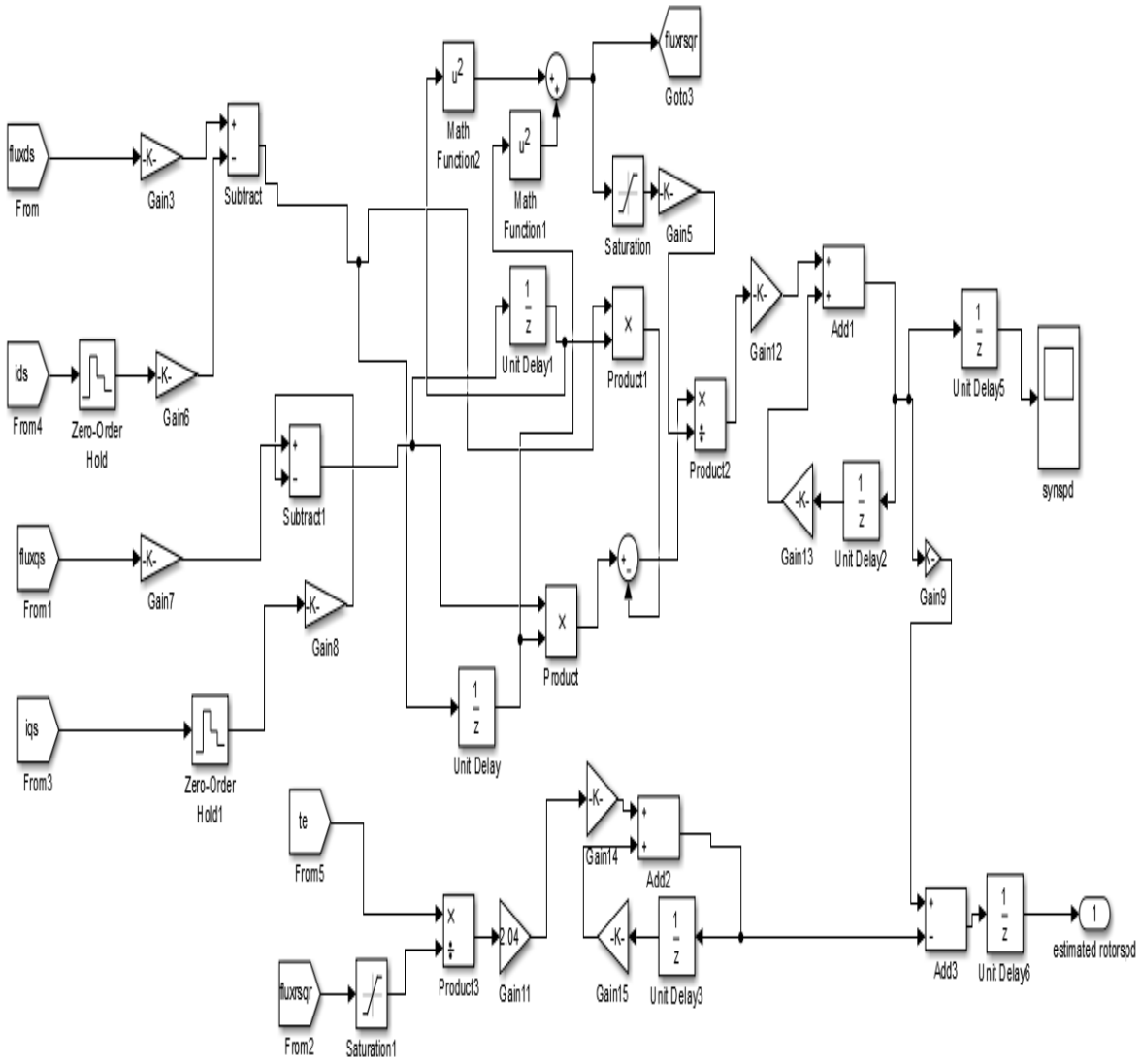


Figure C.5: Rotor speed estimation

Appendix D: High voltage Motor Control KIT

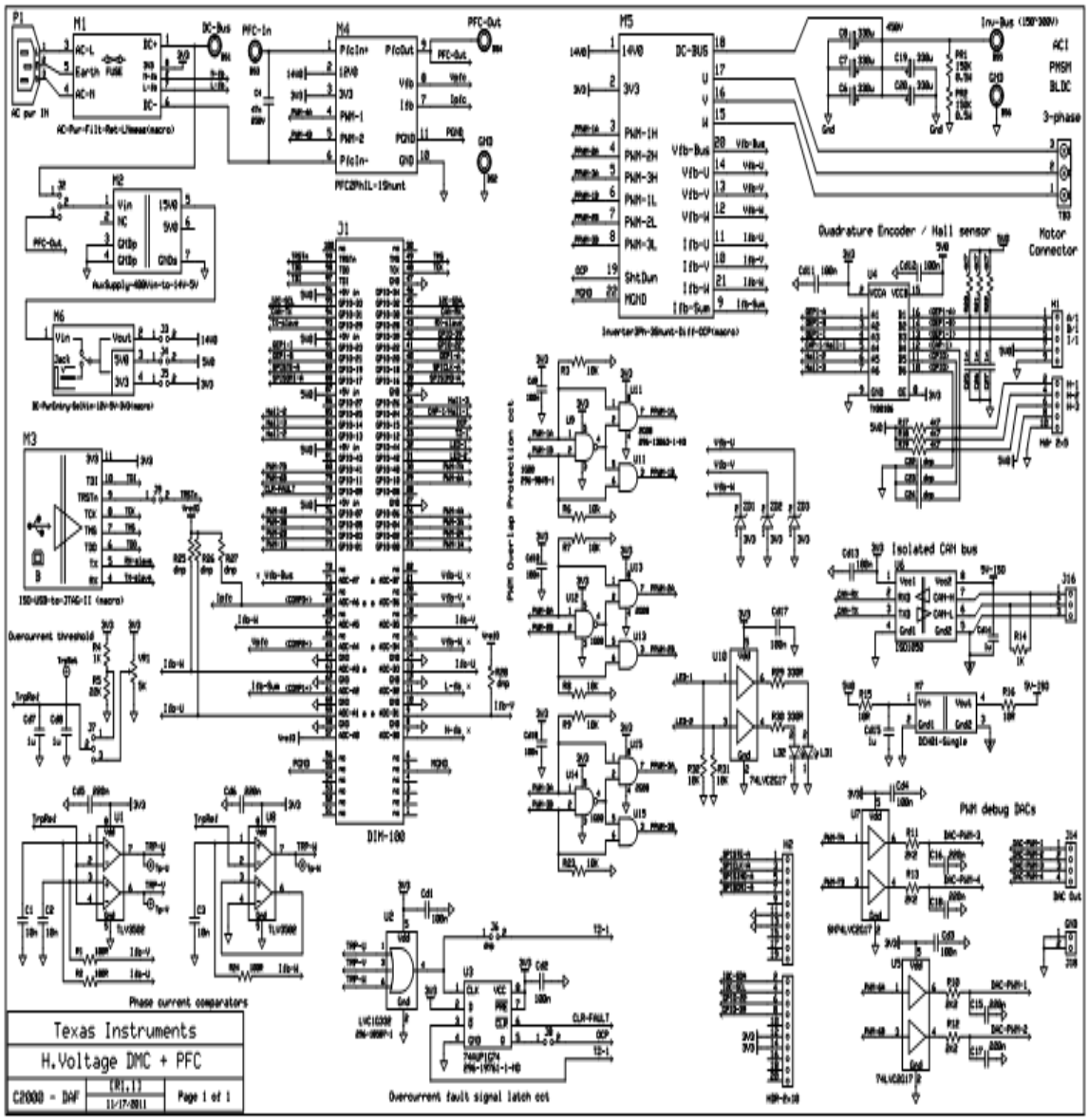


Figure D.1: Circuit schematic of TMDSHVMTRPFCKIT