

**STATE ESTIMATION TECHNIQUES FOR SPEED SENSORLESS FIELD  
ORIENTED CONTROL OF INDUCTION MOTORS**

**A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
OF  
THE MIDDLE EAST TECHNICAL UNIVERSITY**

**BY**

**BİLAL AKIN**

**IN PARTIAL FULLFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF  
MASTER OF SCIENCE  
IN  
THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING**

**AUGUST-2003**

Approval of the Graduate School of Natural and Applied Sciences.

---

Prof. Dr. Canan ÖZGEN

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

---

Prof. Dr. Mübeccel DEMIREKLER

Chairman of the Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

---

Prof. Dr. Aydın Ersak

Supervisor

Examining Committee Members

Prof. Dr. Yıldırım ÜÇTUĞ ( Chairman )

Prof. Dr. Aydın ERSAK

Prof. Dr. Bahri ERCAN

Assoc. Prof. Dr. İsmet ERKMEN

Asst. Prof. Dr. Ahmet M. HAVA

## **ABSTRACT**

# **STATE ESTIMATION TECHNIQUES FOR SPEED SENSORLESS FIELD ORIENTED CONTROL OF INDUCTION MOTORS**

Akın, Bilal

M.Sc. Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. Aydın Ersak

August, 2003

This thesis presents different state estimation techniques for speed sensorless field oriented control of induction motors. The theoretical basis of each algorithm is explained in detail and its performance is tested with simulations and experiments individually.

First, a stochastic nonlinear state estimator, Extended Kalman Filter (EKF) is presented. The motor model designed for EKF application involves rotor speed, dq-axis rotor fluxes and dq-axis stator currents. Thus, using this observer the rotor speed and rotor fluxes are estimated simultaneously. Different from the widely accepted use of EKF, in which it is optimized for either steady-state or transient operations, here using adjustable noise level process algorithm the optimization of EKF has been done for both states; the steady-state and the transient-state of operations. Additionally, the measurement noise immunity of EKF is also investigated.

Second, Unscented Kalman Filter (UKF), which is an updated version of EKF, is proposed as a state estimator for speed sensorless field oriented control of induction motors. UKF state update computations, different from EKF, are derivative free and they do not involve costly calculation of Jacobian matrices. Moreover, variance of each state is not assumed Gaussian, therefore a more realistic approach is provided by UKF. In this work, the superiority of UKF is shown in the state estimation of induction motor.

Third, Model Reference Adaptive System is studied as a state estimator. Two different methods, back emf scheme and reactive power scheme, are applied to MRAS algorithm to estimate rotor speed.

Finally, a flux estimator and an open-loop speed estimator combination is employed to observe stator-rotor fluxes, rotor-flux angle and rotor speed. In flux estimator, voltage model is assisted by current model via a closed-loop to compensate voltage model's disadvantages.

Keywords: Induction motor drive, sensorless field-oriented control, state estimation, EKF, UKF, MRAS

## ÖZ

# HIZ DUYAÇSIZ ALAN YÖNLENDİRMELİ ENDÜKSİYON MOTOR DENETİMİNDE DURUM TAHMİN TEKNİKLERİ

Akın, Bilal

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Danışmanı : Prof. Dr. Aydın Ersak

Ağustos,2003

Bu çalışmada hız duyaçsız alan yönlendirmeli endüksiyon motor denetiminde uygulamaya yönelik durum tahmin yöntemleri geliştirilmiştir. Sunulan tüm yöntemlerin kuramsal içeriği ayrıntılı olarak araştırılmış ve bu yöntemlerin başarımları benzetim yoluyla ve deneysel olarak test edilmiştir.

İlk olarak, doğrusal olmayan sistemlerde durum tahmini için geliştirilmiş olan EKF yöntemi ele alınmıştır. Bu yöntemde uyarlanan motor modeli, rotor hızı, rotor akıları ve rotor akımları aynı anda birlikte tahmin edilmeye yönelik olarak tasarlanmıştır. Genellikle EKF başarımı ya kararlı-durum ya da geçici-durum için ayrı ayrı olarak en iyilendirilmeye çalışılır. Burada kullanılan ANLP yöntemiyle desteklenerek EKF nin başarımı hem kararlı-durumda hem geçici-durumda birlikte en iyileştirilmiştir. Ek olarak EKF yönteminin ölçüm hatalarına olan duyarlılığında test edilmiştir.

EKF' ye ek olarak EKF'nin geliştirilmiş bir versiyonu olan UKF yöntemi, endüksiyon motorlarında bir durum tahmin tekniği olarak sunulmuştur. UKF

yönteminde sistemi doğrusal yapmak için uygulanan türev alma yöntemleri ve bu yöntemler için gerekli olan ve hesaplamaları zorlaştıran bazı basamaklar kullanılmamıştır. Ayrıca UKF modeli belirsizlikleri gerçeğe daha yakın bir tarzda hesaplar. UKF nin bu üstün özelliklerinin motor durum tahminine nasıl olumlu yansıdığı gösterilmiştir.

Bunlara ek olarak MRAS yöntemi de rotor hız tahmini için endüksiyon makinesi modeline uyarlanmıştır. Bunun için MRAS modeli geri besleme ve reaktif güç yöntemleri şeklinde isimlendirilen iki farklı algoritma ile denenmiş ve hız tahmini bu algoritmalarla yapılmıştır.

Son olarakta, geliştirilmiş bir akı tahmin yöntemi ve bir açık döngülü hız tahmin yöntemi durum tahmini için uygulanmıştır. Bu yöntemlerle, stator-rotor akılarını, rotor hızını ve rotor açısını hesaplamak mümkündür. Burada kullanılan akı tahmin yönteminde gerilim yöntemi olarak bilinen akı tahmin yöntemi akım modeli ile kapalı bir döngü sayesinde desteklenmiştir.

Anahtar Kelimeler : Endüksiyon motor sürücüsü, sensörsüz alan yönlendirmeli kontrol, durum tahmini, EKF, UKF, MRAS

## **ACKNOWLEDGMENTS**

I would like to express my sincere gratitude to my supervisor Prof. Dr. Aydin Ersak for his encouragement and guidance throughout the study. I also thank him not only for his technical assists but for his friendship in due course of development of the thesis.

Also, I thank Mr. Umut Orguner and Dr. Ahmet Hava for their technical advice and continuous support during my studies.

Finally, my special thanks go to Mr. Günay Şimsek, Mr. Eray Özçelik, Mr. Ertan Murat and Tübitak Bilten PEG Group for their help during the experimental stage of this work.

## TABLE OF CONTENTS

ABSTRACT .....	iii
ÖZ .....	v
ACKNOWLEDGEMENTS .....	vii
TABLE OF CONTENTS .....	viii
LIST OF TABLES .....	xii
LIST OF FIGURES .....	xiii
LIST OF SYMBOLS .....	xii
CHAPTER	
1.INTRODUCTION .....	1
1.1 OVERVIEW of THE CHAPTERS .....	2
2.LITERARURE REVIEW.....	4
2.1 INDUCTION MACHINE CONTROL.....	4
2.2 FOC OF INDUCTION MACHINE .....	4
2.2.1 IFOC.....	8
2.2.2 DFO .....	10
2.3 VARIABLE SPEED CONTROL USING ADVANCED CONTROL ALGORITHMS .....	11
2.4 CONCLUSIONS .....	17
3. INDUCTION MACHINE MODELING AND FOC SIMULATION .....	19
3.1 THE INDUCTION MOTOR .....	19
3.1.1 PHYSICAL LAYOUT.....	19
3.2. MATHEMATICAL MODEL OF INDUCTION MOTOR .....	20
3.2.1 THREE-PHASE TRANSFORMATIONS .....	20
3.2.2 CLARK TRANSFORMATION .....	21

3.3 CIRCUIT MODEL OF A THREE-PHASE INDUCTION MOTOR	21
3.4 MACHINE MODEL IN ARBITRARY dq REFERENCE FRAME	25
3.4.1 dq0 VOLTAGE EQUATIONS	26
3.4.2 dq0 FLUX LINKAGE RELATIONS	27
3.4.3 dq0 TORQUE EQUATIONS	28
3.5 dq0 STATIONARY and SYNCHRONOUS REFERENCE FRAMES	30
3.6 SIMULATION OF IND. MOTOR IN STATIONARY FRAME	34
3.7. SIMULATION OF FOC DEVELOPED IN STATIONARY REFERENCE FRAME	38
4. PULSEWIDTH MODULATION with SPACE VECTOR THEORY	
4.1 INVERTERS	45
4.1.2 VOLTAGE SOURCE INVERTER	46
4.2 VOLTAGE SPACE VECTORS	47
4.3 SPACE VECTOR MODULATIONS	51
4.4 SVPWM APPLICATION TO THE STATIC POWER BRIDGE and IMPLEMENTATION USING DSP PLATFORM	53
4.5 EVENT MANAGER CONFIGURATION OF DSP FOR SVPWM	58
4.6 SIMULATION and EXPERIMENTAL RESULTS of SVPWM	60
5. KALMAN FILTER	68
5.1 SENSORLESS CONTROL	68
5.2 OBSERVERS	69
5.2.1 GENERAL THEORY ON OBSERVERS	70
5.3 KALMAN FILTER	71
5.4 EXTENDED KALMAN FILTER	75
5.4.1 APPLICATION OF THE EXTENDED KALMAN FILTER	75
5.4.2 MOTOR MODEL FOR EKF	76
5.4.3 DISCRETIZED AUGMENTED MACH. MODEL	78
5.4.4 IMPLEMENTATION OF THE DISCRETIZED EKF ALGORITHM	80
5.5 STATE ESTIMATION SIMULATION with EKF	85

5.6 UNSCENTED KALMAN FILTER .....	94
5.6.1 SIMULATION RESULTS .....	97
5.6.2 EXPERIMENTAL RESULTS .....	103
6. MODEL REFERENCE ADAPTIVE SYSTEMS .....	107
6.1 ADAPTIVE CONTROL.....	107
6.2 MODEL REFERENCE ADAPTIVE SYSTEMS.....	107
6.3 INTRODUCTION TO MRAS PRACTICE in MOTOR CONTROL APPLICATIONS .....	109
6.4 APPLICATION of POPOV's HYPERSTABILITY THEOREM and INTEGRAL INEQUALITY.....	111
6.5 BACK EMF MRAS SCHEME .....	113
6.5.1 ADAPTATION MECHANISMS and STABILITY of MRAS .....	115
6.6 REACTIVE POWER MRAS SCHEME .....	118
6.6.1 REFERENCE MODEL CONTINUOUS TIME REPRESENTATION.....	120
6.6.2 ADAPTIVE MODEL CONTINUOUS TIME REPRESENTATION.....	120
6.6.3 DICRETE TIME REPRESENTATION for MICROCONTROLLER IMPLEMENTATION.....	122
6.6.3.1 REFERENCE MODEL.....	122
6.6.3.2 ADAPTIVE MODEL .....	123
6.6.4 PU DICRETE TIME REPRESENTATION.....	124
6.6.4.1 REFERENCE MODEL.....	124
6.6.4.2 ADAPTIVE MODEL .....	124
6.7 SIMULATION OF THE MRAS SCHEME .....	125
6.8 EXPERIMENTAL RESULTS .....	134
7. FLUX and SPEED OBSERVERS FOR SENSORLESS DFO .....	135
7.1 FLUX OBSERVER .....	136
7.2 OPEN LOOP SPEED OBSERVER .....	138
7.3 EXPERIMENTAL RESULTS .....	140
8. THE HARDWARE AND SOFTWARE .....	147
8.1 HARDWARE OVERVIEW .....	147

8.1.1 THE MOTOR .....	147
8.1.2 THE MOTOR DRIVE.....	150
8.1.3 THE DSP .....	152
8.1.4 THE INTERFACE CARD .....	153
8.2 SOFTWARE OVERVIEW .....	156
8.2.1 SOFTWARE ORGANIZATION.....	156
8.2.2 BASE VALUES and PU MODEL.....	159
8.2.3 FIXED POINT ARITHMETIC .....	160
8.2.4 FOC SOFTWARE MODULES .....	162
9. CONCLUSION and FUTURE WORK .....	168
9.2 FUTURE WORK .....	169
REFERENCES .....	170
APPENDICES	
A S FUNCTION M FILE .....	175
B HYPERSTABILITY THEORY.....	178
C INTERFACE CARD SCHEMATICS .....	180
D OBSERVER PERFORMANCE TEST CODE .....	184

## LIST OF TABLES

### TABLE

3.1 Induction Machine Equations in Arbitrary Reference Frame.....	29
3.2- Induction Machine Equations in Stationary Reference Frame.....	31
3.3- Induction Machine Equations in Synchronously Rotating Reference Frame .....	33
4.1. Power Bridge Output Voltages .....	54
4.2. Stator Voltages in ( $\alpha$ - $\beta$ ) frame and related Voltage Vector.....	55
4.3- Assigned duty cycles to the PWM outputs.....	57
5.1-Discrete Kalman Filter .....	72
8.1- Motor Parameters.....	150

## LIST OF FIGURES

FIGURE	
2.1 Phasor Diagram of the Field Oriented Drive System.....	6
2.2 Field Oriented induction Motor Drive System .....	7
2.3 Indirect Field Oriented Drive System .....	8
2.4 Direct Field Oriented Drive System.....	8
3.1 Relationship between the $\alpha\beta$ abc quantities.....	21
3.2 Relationship between the dq and the abc quantities .....	23
3.3 Relationship between the qd and the abc quantities .....	23
3.4 Relationship between abc and arbitrary dq0.....	25
3.5 Equivalent circuit representation of an induction machine in the arbitrary reference frame .....	27
3.6 Equivalent circuit of an induction machine in the stationary frame .....	31
3.7 Equivalent cct. of an induction mach. in the synchronously rotating frame .....	32
3.8 A simulation of 3-phase AC quantities converted to both stationary frame ( $i_{qs}, i_{ds}$ ) and synchronously rotating frame( $i_{qe}, i_{de}$ ).....	34
3.9 No-Load Response of Stationary Frame Induction Motor Model.....	35
3.10 Open-loop torque-speed curve of the induction motor model at no-load .....	36
3.11 Pole paths of ( $\Delta w_r / \Delta T_{mech}$ ) from no-load to twice the rated torque .....	37
3.12 Root locus of ( $\Delta w_r / \Delta v_{qse}$ ) for varying gains .....	37
3.13 Root locus of ( $\Delta T_{em} / \Delta v_{qse}$ ) for varying gains .....	38
3.14 Step response of $w_r$ (pu) to one volt change in $v_{qse}$ .....	38
3.15 Step response of $w_r$ (pu) to unit change in $T_{load}$ .....	39
3.16 Applied mechanical torque, rotor speed and produced electromagnetic torque.....	40
3.17 Synchronous frame dq axis currents .....	40

3.18 Phase-A stator voltage and current .....	41
3.19 dq axis rotor fluxes and rotor flux .....	42
3.20 Referred rotor currents .....	42
3.21 Four quadrant speed reversal and phase voltage.....	43
3.22 Four quadrant speed reversal and phase voltage .....	43
3.23 Four-quadrant speed reversal and produced torque due to inertia.....	44
3.24 Four-quadrant speed reversal and rotor flux wave-form .....	44
4.1 Circuit diagram of three phase VSI.....	46
4.2 Three phase inverter with switching states .....	47
4.3 Eight switching state topologies of a voltage source inverter .....	48
4.4 First switching state $-V1$ (pnn).....	49
4.5 Representation of topology 1 in $(\alpha-\beta)$ plane.....	49
4.6 Non-zero voltage vectors in $(\alpha-\beta)$ plane.....	50
4.7 Representation of the zero voltage vectors in $(\alpha-\beta)$ plane.....	50
4.8 Output voltage vector (V) in $(\alpha-\beta)$ plane.....	51
4.9 Output line voltages in the time domain .....	52
4.10 Synthesis o the required output voltage vector in sector 1 .....	52
4.11 Phase gating signals in Symmetrical Sequence SVM.....	53
4.12 Power Bridge .....	54
4.13 Voltage Vectors.....	55
4.14 Projection of the reference Voltage Vector .....	56
4.15 Sector 3 PWM Patterns and Duty Cycles.....	58
4.16 Dead time band .....	60
4.17 SVPWM Algorithm Simulation.....	60
4.18 Simulated waveforms of duty cycles, ( $t_{aon}, t_{bon}, t_{con}$ ).....	61
4.19 Sector numbers of voltage vector.....	61
4.20 Duration of two sector boundary vectors ( $t1, t2$ ).....	62
4.21 The projections of the $V_a, V_b$ and $V_c$ of the reference voltage vector in the (a b c) plane -(X, Y, Z) .....	62
4.22 A typical line to line voltage output of SVPWM.....	63
4.23 SVPWM output with the signal sampled ( $m=0.4$ ).....	63
4.24 SVPWM output with the signal sampled ( $m=0.6$ )-Zoomed.....	64
4.25 Duty cycle of PWM1.....	64

4.26 Low-pass filtered form of PWM1 pulses .....	65
4.27 Sector number of the reference voltage.....	65
4.28 Duration of two boundary vectors ( $t_1, t_2$ ) .....	66
4.29 Projection vectors in abc plain (X,Y in time domain).....	66
4.30 Typical phase current of an induction motor driven by SVPWM under heavy load conditions .....	67
5.1 Block diagram of an observer.....	71
5.2 Block Diagram of Kalman Filter .....	74
5.3 High Speed, No-Load, Four Quadrant Speed Estimation with EKF .....	86
5.4 High Speed, No-Load, Four Quadrant Speed Estimation with EKF at Steady State.....	87
5.5 High Speed, No-Load, Speed Estimation with EKF – Steady State Performance Optimized.....	87
5.6 Low Speed, No-Load, Four Quadrant Speed Estimation with EKF.....	88
5.7 Low Speed, No-Load, Speed Estimation with EKF at Steady State to Transient State.....	88
5.8 High Speed, Full-Load, Speed Estimation with EKF .....	89
5.9 High Speed, No-Load, Speed Estimation using EKF with Adjustable Noise Level .....	89
5.10 Estimated States in (5.28) Respectively at No Load.....	90
5.11 State I and II (dq-axis Stator Currents) .....	91
5.12 State III and IV (dq-axis Rotor Fluxes with their magnitude).....	92
5.13 Injected noise to the stator currents in pu .....	93
5.14 Estimated rotor speed with measured noisy current .....	93
5.15 Induction motor actual states at no load four quadrant high speed reversal.....	98
5.16 Induction motor estimated states with UKF at no load four quadrant high speed reversal .....	98
5.17 Induction motor estimated speed at no-load four quadrant high speed reversal	99
5.18 Induction motor estimated speed at no-load four quadrant low speed reversal..	99
5.19 Induction motor estimated states at %100 rated torque and speed .....	100
5.20 Induction motor estimated speed at %100 rated torque and speed .....	101
5.21 Induction motor estimated speed at %10 rated torque and speed .....	101
5.22 Induction motor estimated speed optimized for steady state performance at	

%100 rated torque and speed using EKF and UKF .....	102
5.23 Induction motor estimated speed optimized for transient performance at %100 rated torque and speed using EKF and UKF .....	102
5.24 The estimated states I and II by EKF and the measured states I and II .....	103
5.25 The estimated states II and III by EKF and the magnitude of the rotor flux...	103
5.26 Rotor speed tracking performance of EKF obtained experimentally.....	104
5.27 The estimated states I and II by EKF and the measured states I and II .....	105
5.28 The estimated states II and III by UKF and the magnitude of the rotor flux ..	105
5.29 Rotor speed waveforms obtained experimentally by UKF and EKF under the same experimental conditions.....	106
6.1 General parallel MRAS scheme.....	108
6.2 Generalized Model Reference Adaptive System .....	109
6.3 MRAS based speed estimator scheme using space vector .....	109
6.4 Equivalent non-linear feedback system.....	111
6.5 Coordinates in stationary reference frame.....	113
6.6 Structure of the MRAS system for speed estimation.....	115
6.7 Equivalent nonlinear feedback system of MRAS.....	117
6.8 System structure of rotor speed observer using the tuning signal $I_m$ .....	119
6.9 The Simulink model of back emf MRAS scheme .....	126
6.10 The Simulink model of reactive power MRAS scheme.....	126
6.11 Four-quadrant speed reversal of 5 hp induction motor using reactive power MRAS scheme at no_load up to rated speed.....	127
6.12 Four-quadrant speed reversal of 20 hp induction motor using reactive power MRAS scheme at no_load up to rated speed.....	128
6.13 5hp induction motor speed estimation.....	129
6.14 5hp induction motor estimated speed speed using reactive power MRAS scheme, applied torque and stator q-axis current.....	130
6.15 20 hp induction motor estimated speed using reactive power MRAS scheme, applied torque and stator q-axis current .....	131
6.16 A typical estimated speed using back emf MRAS scheme .....	132
6.17 Rotor speed estimated by MRAS experimentally at no-load by back emf scheme .....	134
6.18 Speed tracking of the back emf MRAS scheme. ....	134

7.1 (a) stator phase currents under heavy load conditions (b) stator phase currents under no-load condition (c) stator phase voltages .....	141
7.2 dq-axis rotor fluxes in stationary frame obtained from current model .....	142
7.3 dq-axis stator fluxes in stationary frame obtained from the current model.....	142
7.4 The dq-axis stator fluxes in stationary frame obtained from the voltage model.....	143
7.5 Back emfs with added compensating voltages .....	143
7.6 dq-axis stationary frame rotor fluxes reconstructed by voltage model.....	144
7.7 q-axis stationary frame rotor flux reconstructed by voltage model with rotor flux magnitude .....	144
7.8 Rotor flux angle based on voltage model.....	145
7.9 Reference speed and estimated speed .....	145
7.10 Reference speed and estimated speed .....	146
8.1 Overall hardware configuration of the thesis.....	148
8.2 Approximate per Phase Equivalent Circuit for an Induction Machine .....	148
8.3 Diagram of dc measurement.....	149
8.4 Dc_link Circuit.....	150
8.5 Overall Diagram of the Inverter.....	152
8.6 Experimental setup (Interface card, DSP and inverter).....	155
8.7 Interface card .....	156
8.8 Software Flowchart .....	157
8.9 Overall FOC Algorithm Timing .....	159
8.10 Speed Sensorless FOC of Induction Motor – System Block Diagram Showing Software Modules .....	163

## LIST OF SYMBOLS

### SYMBOL

$e_{md}$	back emf d axis component
$e_{mq}$	back emf q axis component
$i_{ds}^e$	d axis stator current in synchronous frame
$i_{qs}^e$	q axis stator current in synchronous frame
$i_{ds}^s$	d axis stator current in stationary frame
$i_{qs}^s$	q axis stator current in stationary frame
$i_{ar}$	Phase-a rotor current
$i_{br}$	Phase-b rotor current
$i_{cr}$	Phase-c rotor current
$i_{as}$	Phase-a stator current
$i_{bs}$	Phase-b stator current
$i_{cs}$	Phase-c stator current
$L_m$	Magnetizing inductance
$L_{ls}$	Stator leakage inductance
$L_{lr}$	Rotor leakage inductance
$L_s$	Stator self inductance
$L_r$	Rotor self inductance
$R_s$	Stator resistance
$R_r$	Referred rotor resistance
$q_{md}$	reactive power d axis component
$q_{mq}$	reactive power q axis component
$T_{em}$	Electromechanical torque
$T_r$	Rotor time constant
$V_{as}$	Phase-a stator voltage
$V_{bs}$	Phase-b stator voltage
$V_{cs}$	Phase-c stator voltage

$V_{ar}$	Phase-a rotor voltage
$V_{br}$	Phase-b rotor voltage
$V_{cr}$	Phase-c rotor voltage
$V_{ds}^s$	d axis stator voltage in stationary frame
$V_{qs}^s$	q axis stator voltage in stationary frame
$V_{ds}^e$	d axis stator voltage in synchronous frame
$V_{qs}^e$	q axis stator voltage in synchronous frame
$V_{dc}$	Dc-link voltage
$X_m$	Stator magnetizing reactance
$X_{ls}$	Stator leakage reactance
$X_{lr}$	Rotor leakage reactance
$X_s$	Stator self reactance
$X_r$	Rotor self reactance
$\omega_e$	Angular synchronous speed
$\omega_r$	Angular rotor speed
$\omega_{sl}$	Angular slip speed
$\theta_e$	Angle between the synchronous frame and the stationary frame
$\theta_d$	Angle between the synchronous frame and the stationary frame when d axis is leading
$\theta_q$	Angle between the synchronous frame and the stationary frame when q axis is leading
$\theta_{\psi r}$	Rotor flux angle
$\psi_{ds}^s$	d axis stator flux in stationary frame
$\psi_{qs}^s$	q axis stator flux in stationary frame
$\psi_{ds}^e$	d axis stator flux in synchronous frame
$\psi_{qs}^e$	q axis stator flux in synchronous frame
$\psi_{as}$	Phase-a stator flux
$\psi_{bs}$	Phase-b stator flux
$\psi_{cs}$	Phase-c stator flux
$\psi_{ar}$	Phase-a rotor flux
$\psi_{br}$	Phase-b rotor flux
$\psi_{cr}$	Phase-c rotor flux

# **CHAPTER 1**

## **INTRODUCTION**

Induction motors are relatively rugged and inexpensive machines. Therefore much attention is given to their control for various applications with different control requirements. An induction machine, especially squirrel cage induction machine, has many advantages when compared with DC machine. First of all, it is very cheap. Next, it has very compact structure and insensitive to environment. Furthermore, it does not require periodic maintenance like DC motors. However, because of its highly non-linear and coupled dynamic structure, an induction machine requires more complex control schemes than DC motors. Traditional open-loop control of the induction machine with variable frequency may provide a satisfactory solution under limited conditions. However, when high performance dynamic operation is required, these methods are unsatisfactory. Therefore, more sophisticated control methods are needed to make the performance of the induction motor comparable with DC motors. Recent developments in the area of drive control techniques, fast semiconductor power switches, powerful and cheap microcontrollers made induction motors alternatives of DC motors in industry.

The most popular induction motor drive control method has been the field oriented control (FOC) in the past two decades. Furthermore, the recent trend in FOC is towards the use of sensorless techniques that avoid the use of speed sensor and flux sensor. The sensors in the hardware of the drive are replaced with state observers to minimize the cost and increase the reliability.

This work is mainly focused on the state observers to estimate the states that are used in the FOC algorithms frequently. For this purpose, two different Kalman Filtering techniques, EKF and UKF, are employed to estimate rotor speed and dq-axis rotor fluxes. Using these techniques, one can obtain very precise flux and speed information as shown in the simulations and experimental results. Furthermore, model reference adaptive system (MRAS) is used to estimate the rotor speed. The back emf and the reactive power schemes are applied to MRAS which are superior to the previous MRAS schemes proposed in the literature. In this work, it is also shown that the rotor speed estimation performance of these schemes is quite satisfactory in the simulations and experimental results. Finally, a combination of well-known open loop observers, voltage model and current model, is used to estimate the rotor flux and rotor flux angle which are employed in direct field orientation.

## **1.1 Overview of the Chapters**

This thesis is organized as follows:

The literature review is given in Chapter 2. The review mainly focused on field oriented control, sensorless control and state observers such as EKF, UKF and MRAS. The previous works are discussed briefly and compared with each other.

Chapter 3 presents a generalized dynamic mathematical model of the induction motor which can be used to construct various equivalent circuit models in different reference frames. A torque-speed control of induction machine with indirect field orientation is simulated to be familiar with the FOC.

Chapter 4 presents the theoretical background of space vector pulse width modulation (SVPWM) in detail. DSP implementation of SVPWM is also given in this part. Moreover, the simulation and the experimental results of SVPWM are illustrated.

Chapter 5 is devoted to Kalman filtering techniques. First the theoretical base of EKF is given in detail. The discretized model of the motor, which is applied to EKF, is studied for microcontroller implementation. Afterwards, derivative free, non-linear state estimator technique, UKF, is presented and compared with EKF. The performance of each technique is confirmed by simulations and experimental results.

In Chapter 6, MRAS method is introduced to estimate the rotor speed. Two different schemes are applied to MRAS for this task. The stability analysis and discretized forms of both schemes are given for microcontroller implementation. The performance of these schemes is examined under varying conditions in simulations. The simulations are supported with the experimental results.

Chapter 7 summarizes the development of a flux estimator with a well known speed estimator. The originality of the flux estimator is emphasized and experimental results are added for both estimators.

Chapter 8 introduces the experimental setup and the software of this thesis briefly.

Chapter 9 summarizes the thesis and concludes with the contributions associated with the observation techniques employed in FOC.

## **CHAPTER 2**

### **LITERATURE REVIEW**

An induction machine, a power converter and a controller are the three major components of an induction motor drive system. Some of the disciplines related to these components are electric machine design, electric machine modeling, sensing and measurement techniques, signal processing, power electronic design and electric machine control. It is beyond the scope of this research to address all of these areas: it will primarily focus on the issue related to the induction machine control. A conventional low cost volts per hertz or a high performance field oriented controller can be used to control the machine. This chapter reviews the principles of the field orientation control of the induction machines and outline major problems in its design and implementation.

#### **2.1 Induction Machine Control**

The controllers required for induction motor drives can be divided into two major types: a conventional low cost volts per hertz v/f controller and torque controller [1]-[4]. In v/f control, the magnitudes of the voltage and frequency are kept in proportion. The performance of the v/f control is not satisfactory, because the rate of change of voltage and frequency has to be low. A sudden acceleration or deceleration of the voltage and frequency can cause a transient change in the current, which can result in drastic problems. Some efforts were made to improve v/f control performance, but none of these improvements could yield a v/f torque controlled drive systems and this made DC motors a prominent choice for variable speed applications. This began to change when the theory of field orientation was introduced by Hasse

and Blaschke. Field orientation control is considerably more complicated than DC motor control. The most popular class of the successful controllers uses the vector control technique because it controls both the amplitude and phase of AC excitation. This technique results in an orthogonal spatial orientation of the electromagnetic field and torque, commonly known as Field Oriented Control (FOC).

## 2.2 Field Orientation Control of Induction Machine

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

- An independent control of armature current to overcome the effects of armature winding resistance, leakage inductance and induced voltage;
- An independent control of constant value of flux;

If all of these three requirements are met at every instant of time, the torque will follow the current, allowing an immediate torque control and decoupled flux and torque regulation.

Next, a two phase d-q model of an induction machine rotating at the synchronous speed is introduced which will help to carry out this decoupled control concept to the induction machine. This model can be summarized by the following equations (see chapter 3 for detail):

$$v_{qs}^e = p\psi_{qs}^e + w_e \psi_{ds}^e + r_s i_{qs}^e \quad (2.1)$$

$$v_{ds}^e = p\psi_{ds}^e - w_e \psi_{qs}^e + r_s i_{ds}^e \quad (2.2)$$

$$0 = p\psi_{qr}^e + (w_e - w_r)\psi_{dr}^e + r_r i_{qr}^e \quad (2.3)$$

$$0 = p\psi_{dr}^e - (w_e - w_r)\psi_{qr}^e + r_r i_{dr}^e \quad (2.4)$$

$$\psi_{qs}^e = L_s i_{qs}^e + L_m i_{qr}^e \quad (2.5)$$

$$\psi_{ds}^e = L_s i_{ds}^e + L_m i_{dr}^e \quad (2.6)$$

$$\psi_{qr}^e = L_m i_{qs}^e + L_r i_{qr}^e \quad (2.7)$$

$$\psi_{dr}^e = L_m i_{ds}^e + L_r i_{dr}^e \quad (2.8)$$

$$T_e = \frac{3P}{2} \frac{L_m}{L_r} (\psi_{dr}^e i_{qs}^e - \psi_{qr}^e i_{ds}^e) \quad (2.9)$$

$$T_e = Jp w_r + B w_r + T_L \quad (2.10)$$

and it is quite significant to synthesize the concept of field-oriented control. In this model it can be seen from the torque expression (2.9) that, if the flux along the q-axis can be made zero then all the flux is aligned along the d-axis and, therefore, the

torque can be instantaneously controlled by controlling the current along q-axis. Then the question will be how it can be guaranteed that all the flux is aligned along the d-axis of the machine. When three-phase voltages are applied to the machine, they produce three-phase fluxes both in the stator and the rotor. The three-phase fluxes can be represented in a two-phase stationary ( $\alpha$ - $\beta$ ) frame. If these two phase fluxes along ( $\alpha$ - $\beta$ ) axes are represented by a single-vector then all the machine flux will be aligned along that vector. This vector is commonly specified as d-axis which makes an angle  $\theta_e$  with the stationary frame  $\alpha$ -axis, as shown in Fig.2.1. The q-axis is set perpendicular to the d-axis. The flux along the q-axis in this case will be obviously zero. The phasor diagram Fig.2.1 presents these axes. When the machine input currents change sinusoidally in time, the angle  $\theta_e$  keeps changing. Thus the problem is to know the angle  $\theta_e$  accurately, so that the d-axis of the d-q frame is locked with the flux vector.

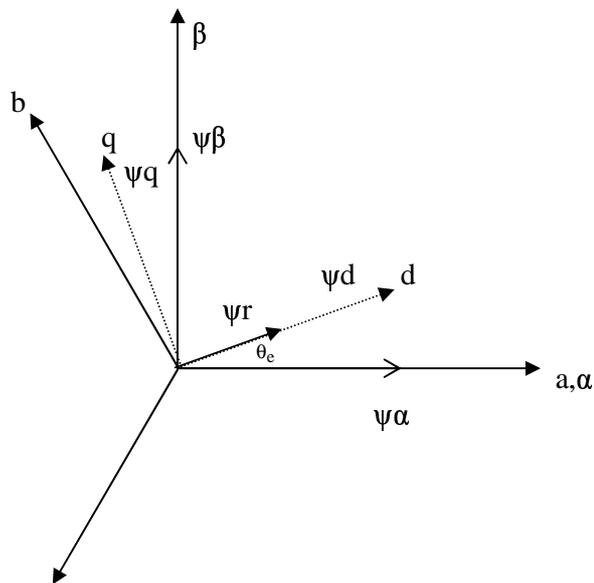


Fig.2.1- Phasor Diagram of the Field Oriented Drive System

The control inputs can be specified in two phase synchronously rotating d-q frame as  $i_{ds}^e$  and  $i_{qs}^e$  such that  $i_{ds}^e$  being aligned with the d-axis or the flux vector. These two-phase synchronous control inputs are converted into two-phase stationary quantities and then to three-phase stationary control inputs. To accomplish this the flux angle  $\theta_e$

must be known precisely. The angle  $\theta_e$  can be found either by Indirect Field Orientation control (IFO) or by Direct Field Orientation control (DFO). The controller implemented in this fashion that can achieve a decoupled control of the flux and the torque is known as field oriented controller. The block diagram is shown in the Fig.2.2 In the field-oriented controller the flux can be regulated in the stator, air-gap or rotor flux orientation [1]-[4].

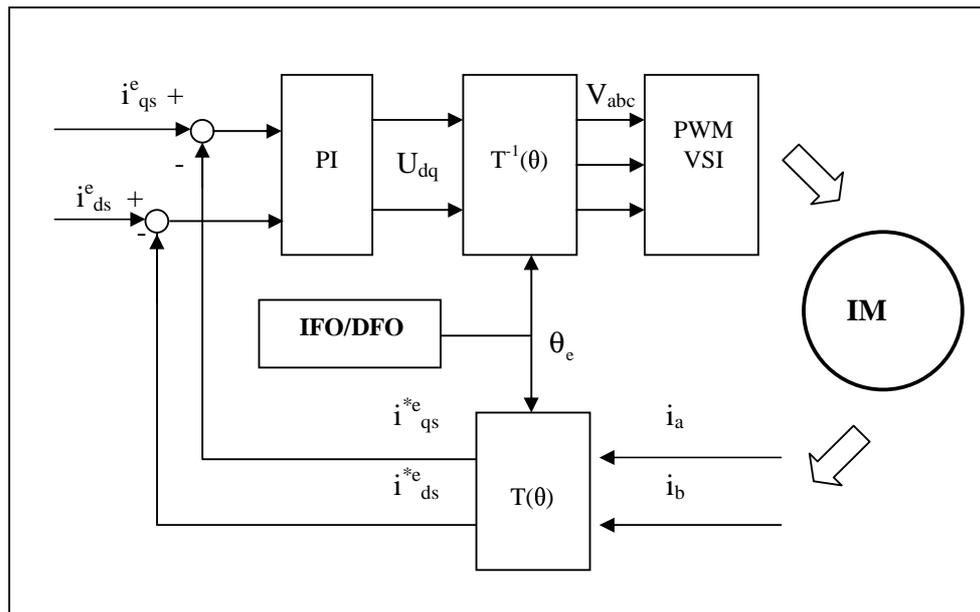


Fig.2.2- Field Oriented induction Motor Drive System

The control algorithm for calculation of the rotor flux angle  $\theta_e$  using IFO control is shown in the Fig 2.3. This algorithm is based on the assumption that, the flux along the q-axis is zero, which forces the command slip velocity to be  $w_{sl} = i_{qs}^e / (T_r \cdot i_{ds}^e)$  as a necessary and sufficient condition to guarantee that all the flux is aligned with d-axis and the flux along q-axis is zero. The angle  $\theta_e$  can then be determined as the sum of the slip and the rotor angles after integrating the respective velocities. This slip angle includes the necessary and sufficient condition for decoupled control of flux and torque. The rotor speed can be measured directly by using an encoder or can be estimated. In case the rotor speed is estimated, the control technique is known as sensorless control. This concept will be studied in detail in the following chapters. Fig 2.4. shows the control algorithm block diagram for DFO control. In this technique the flux angle  $\theta_e$  is classically calculated by sensing the air-gap flux through the use of

flux sensing coils, or can be calculated by estimating the flux along the  $\alpha$ - $\beta$  axes using the voltage and current signals.

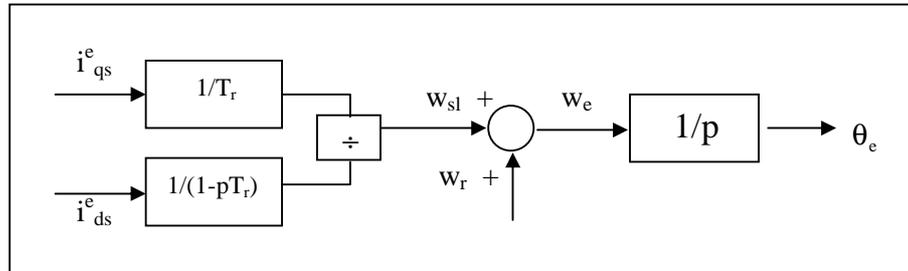


Fig.2.3- Indirect Field Oriented Drive System

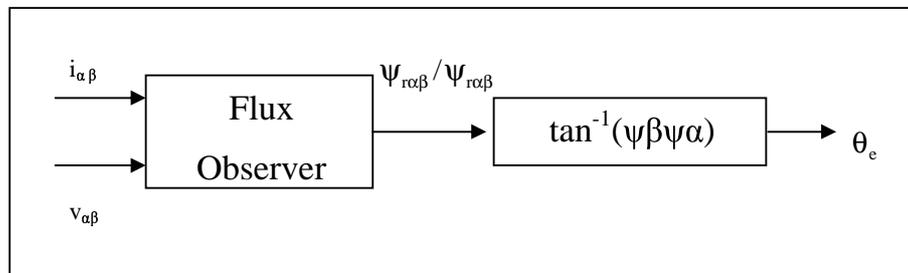


Fig.2.4- Direct Field Oriented Drive System

### 2.2.1 Indirect Field Orientation Control

In indirect field orientation, the synchronous speed  $w_e$  is the same as the instantaneous speed of the rotor flux vector  $\psi_{dr}^e$  and the d-axis of the d-q coordinate system is exactly locked on the rotor flux vector (rotor flux vector orientation). This facilitates the flux control through the magnetizing current  $i_{ds}^e$  by aligning all the flux with the d-axis while aligning the torque-producing component of the current with the q-axis. After decoupling the rotor flux and torque-producing component of the current components, the torque can be instantaneously controlled by controlling the current  $i_{qs}^e$ . The requirement to align the rotor flux with the d-axis of the d-q coordinate system means that the flux along the q-axis must be zero. This means that (2.7) becomes  $i_{qs}^e = -(L_r i_{qr}^e) / L_m$  and the current through the q-axis of the mutual inductance is zero.

Based on this restriction  $w_{sl}$  is :

$$w_{sl} = \frac{\frac{1}{T_r} i_{qs}^e}{\frac{1}{1-pT_r} i_{ds}^e} \quad (2.11)$$

These relations suggest that flux and torque can be controlled independently by specifying d-q axis currents provided the slip frequency is satisfied (2.11) at all instants.

The concept of indirect field oriented control developed in the past has been widely studied by researchers during the last two decades. The rotor flux orientation is both the original and usual choice for the indirect orientation control. Also the IFO control can be implemented in the stator and air-gap flux orientation as well. De Doncker [5] introduced this concept in his universal field oriented controller. In the air-gap flux the slip and flux relations are coupled equations and the d-axis current does not independently control the flux as it does in the rotor flux orientation. For the constant air-gap flux orientation, the maximum of the produced torque is %20 less than that of the other two methods [3]. In the stator flux orientation, the transient reactance is a coupling factor and it varies with the operating conditions of the machine. In addition, Nasar [3] shows that among these methods, rotor flux oriented control has linear torque curve. Therefore, the most commonly used choice for IFO is the rotor flux orientation.

The IFOC is an open loop, feed-forward control in which the slip frequency is fed-forward guaranteeing the field orientation. This feed-forward control is very sensitive to the rotor open circuit time constant,  $\tau_r$ . Therefore,  $\tau_r$  must be known in order to achieve a decoupled control of torque and flux components by controlling  $i_{qs}^e$  and  $i_{ds}^e$ , respectively. When  $\tau_r$  is not set correctly, the machine is said to be detuned and the performance will become sluggish due to loss of decoupled control of torque and flux. The measurement of the rotor time constant, its effects on the system performance and its adaptive tuning to the variations resulting during the operation of the machine have been studied extensively in the literature [6-8]. Lorenz, Krishnan and Novotny [6-8] studied the effect of temperature and saturation level on the rotor time constant and concluded that it can reduce the torque capability of the machine and torque/amperes of the machine. The detuning effect becomes more severe in the field-weakening region. Also, it results in a steady-state error and, transient

oscillations in the rotor flux and torque. Some of the advanced control techniques such as estimation theory tools and adaptive control tools are also studied to estimate rotor time constant and other motor parameters [25, 26, 29, 30, 31, 50, 61-63]

### 2.2.2 Direct Field Orientation

The DFO control and sensorless control rely heavily on accurate flux estimation. DFOC is most often used for sensorless control, because the flux observer used to estimate the synchronous speed or angle can also be used to estimate the machine speed. Investigation of ways to estimate the flux and speed of the induction machine has also been extensively studied in the past two decades. Classically, the rotor flux was measured by using a special sensing element, such as Hall effect sensors placed in the air-gap. An advantage of this method is that additional required parameters,  $L_{lr}$ ,  $L_m$ , and  $L_r$  are not significantly affected by changes in temperature and flux level. However, the disadvantage of this method is that a flux sensor is expensive and needs special installation and maintenance. Another flux and speed estimation technique is saliency based with fundamental or high frequency signal injection. One advantage of saliency technique is that the saliency is not sensitive to actual motor parameters, but this method fails at low and zero speed level. When applied with high frequency signal injection [9], the method may cause torque ripples, and mechanical problems.

Gabriel [10] avoided the special flux sensors and coils by estimating the rotor flux from the terminal quantities (stator voltages and currents). This technique requires the knowledge of the stator resistance along with the stator, rotor leakage inductances and magnetizing inductance. This method is commonly known as the *Voltage Model Flux Observer* (VMFO). The stator flux in the stationary frame can be estimated by the equations:

$$\dot{\Psi}_{ds}^s = v_{ds}^s - r_s i_{ds}^s \quad (2.13)$$

$$\dot{\Psi}_{qs}^s = v_{qs}^s - r_s i_{qs}^s \quad (2.14)$$

Then the rotor flux can be expressed as:

$$\Psi_{dr}^s = \frac{L_r}{L_m} (\Psi_{ds}^s - L_{\sigma} i_{ds}^s) \quad (2.15)$$

$$\Psi_{qr}^s = \frac{L_r}{L_m} (\Psi_{qs}^s - L_{\sigma} i_{qs}^s) \quad (2.16)$$

where  $L_\sigma = (L_s - L_m^2 / L_r)$  is the transient leakage inductance.

In this model, integration of the low frequency signals, dominance of stator resistance voltage drop at low speed and leakage inductance variation result in a less precise flux estimation. Integration at low frequency is studied by [11] and three different alternatives are given. Estimation of rotor flux from the terminal quantities depends on parameters such as stator resistance and leakage inductance. The study of parameter sensitivity shows that the leakage inductance can significantly affect the system performance such as stability, dynamic response, and utilizations of the machine and the inverter.

The *Current Model Flux Observer* (CMFO) is an alternative approach to overcome the problems of leakage inductance and stator resistance at low speed. In this model flux can be estimated as:

$$\dot{\Psi}_{dr}^s = -\frac{1}{T_r} \Psi_{dr}^s - w_r \Psi_{qr}^s + \frac{L_m}{T_r} i_{ds}^s \quad (2.17)$$

$$\dot{\Psi}_{qr}^s = -\frac{1}{T_r} \Psi_{qr}^s - w_r \Psi_{dr}^s + \frac{L_m}{T_r} i_{qs}^s \quad (2.18)$$

However, it does not work well at high speed due to its sensitivity to the rotor resistance. Jansen [12] did an extensive study on VMFO and CMFO based direct field orientation control, discussed the design and accuracy assessment of various flux observers, compared them, and analyzed the alternative flux observers. To further improve the observer performance, closed-loop rotor flux observers are proposed which use the estimated stator current error [12-13] or the estimated stator voltage error [13] to estimate the rotor flux. Furthermore, Lennart [14] proposed reduced order observers for this task.

### 2.3 Variable Speed Control Using Advanced Control Algorithms

There are two issues in motion control using field-oriented controlled (FOC) induction machine drives. One is to make the resulting drive system and the controller robust against parameter deviations and disturbances. The other is to make the system intelligent e.g. to adjust the control system itself to environment changes and task requirements. If the speed regulation loop fails to produce the command current

correctly, than the desired torque response will not be produced by the induction machine. In addition, such a failure may cause the degradation of slip command as well. As a result, a satisfactory speed regulation is extremely important not only to produce desired torque performance from the induction machine but also to guarantee the decoupling between control of torque and flux.

Conventionally, a PI controller has been used for the speed regulation to generate a command current for last two decades, and accepted by industry because of its simplicity. Even though, a well-tuned PI controller performs satisfactorily for a field-oriented induction machine during steady state. The speed response of the machine at transient, especially for the variable speed tracking, may sometimes be problematic. In last two decades, alternative control algorithms for the speed regulation were investigated. Among these, fuzzy logic, sliding mode, and adaptive nonlinear control algorithms gained much attention, however these controllers are not in the scope of this thesis.

A traditional rotor flux-oriented induction machine drive offers a better control performance but it often requires additional sensors on the machine. This adds to the cost and complexity of the drive system. To avoid these sensors on the machine, many different algorithms are proposed for the last three decades to estimate the rotor flux vector and or/ rotor shaft speed. The recent trend in field-oriented control is to use such algorithms based on the terminal quantities of the machine for the estimation of the fluxes and speed. They can easily be applied to any induction machine. Therefore, our focus in this study is also on these algorithms.

Before looking into individual approaches, the common problems of the speed and flux estimation are discussed briefly for general field-orientation and state estimation algorithms.

- *Parameter sensitivity:* One of the important problems of the sensorless control algorithms for the field-oriented induction machine drives is the insufficient information about the machine parameters which yield the estimation of some machine parameters along with the sensorless structure. Among these parameters stator resistance, rotor resistance and rotor time-constant play more important role than the other parameters since these values are more sensitive to temperature changes. The knowledge of the correct stator resistance  $r_s$ , is important to widen the operation region toward the lower speed range. Since at low speeds the

induced voltage is low and stator resistance voltage drop becomes dominant, a mismatching stator resistance induces instability in the system. On the other hand, errors made in determining the actual value of the rotor resistance  $r_r$ , may cause both instability of the system and speed estimation error proportional to  $r_r$  [15]. Also, correct  $\tau_r$  value is vital decoupling factor in IFOC.

- *Pure Integration:* The other important issue regarding many of the topologies is the integration process inherited from the induction machine dynamics where an integration process is needed to calculate the state variables of the system. However, it is difficult both to decide on the initial value, and prevent the drift of the output of a pure integrator. Usually, to overcome this problem a low-pass filter replaces the integrator.
- *Overlapping-loop Problems:* In a sensorless control system, the control loop and the speed estimation loop may overlap and these loops influence each other. As a result, outputs of both of these loops may not be designed independently; in some bad cases this dependency may influence the stability or performance of the overall system.

The algorithms, where terminal quantities of the machine are used to estimate the fluxes and speed of the machine, are categorized in two basic groups. First one is "the open-loop observers," in a sense that the on-line model of the machine does not use the feedback correction. Second one is "the closed-loop observers" where the feedback correction is used along with the machine model itself to improve the estimation accuracy. These two basic groups can also be divided further into subgroups based on the control method used. These can be summarized as:

Open-loop observers based on;

- Current model,
- Voltage model,
- Full-order observer,

Closed loop observers based on;

- Model Reference Adaptive Systems (MRAS),
- Kalman filter techniques,
- Adaptive observers based on both voltage and current model,
- Neural network flux and speed estimators,
- Sliding mode flux and speed estimators.

Open-loop observers, in general, use different forms of the induction machine differential equations. Current model based open-loop observers [12]-[14] use the measured stator currents and rotor velocity. The velocity dependency of the current model is very important since this means that although using the estimated flux eliminates the flux sensor, the position sensor is still required. On the other hand, voltage model based open-loop observers [12]-[14] use the measured stator voltage and current as inputs. These types of estimators require a pure integration that is difficult to implement for low excitation frequencies due to the offset and initial condition problems. Cancellation method open-loop observers can be formed by using measured stator voltage, stator current and rotor velocity as inputs, and use the differentiation to cancel the effect of the integration. However, it suffers from two main drawbacks. One is the need for the derivation which makes the method more susceptible to noise than the other methods. The other drawback is the need for the rotor velocity similar to current model. A full-order open-loop observer, on the other hand, can be formed using only the measured stator voltage and rotor velocity as inputs where the stator current appears as an estimated quantity. Because of its dependency on the stator current estimation, the full order observer will not exhibit better performance than the current model. Furthermore, parameter sensitivity and observer gain are the problems to be tuned in a full order observer design [16]. These open-loop observer structures are all based on the induction machine model, and they do not employ any feedback. Therefore, they are quite sensitive to parameter variations, which yield the estimation of some machine parameters along with the sensorless structure.

On the other hand, some kind of feedback may be helpful to produce more robust structures to parameter variations. For this purpose many closed-loop topologies are proposed using different induction machine models and control methods. Among these MRAS attracts attention and several different algorithms are

produced. In MRAS, in general, a comparison is made between the outputs of two estimators. The estimator which does not contain the quantity to be estimated can be considered as a reference model of the induction machine. The other one which contains the estimated quantity, is considered as an adjustable model. The error between these two estimators is used as an input to an adaptation mechanism. For sensorless control algorithms most of the times the quantity which differs the reference model from the adjustable model is the rotor speed. The estimated rotor speed in the adjustable model is changed in such a way that the difference between two estimators converges to zero asymptotically, and the estimated rotor speed will be equal to actual rotor speed. The basics of the analysis and design of MRAS are discussed in [2, 17]. In [15, 18, 19] voltage model is assumed as reference model, current model is assumed as the adjustable model and estimated rotor flux is assumed as the reference parameter to be compared. In [20] similar speed estimators are proposed based on the MRAS, and a secondary variable is introduced as the reference quantity by letting the rotor flux through a first-order delay instead of a pure integration to nullify the offset. However, their algorithms produce inaccurate estimated speed if the excitation frequency goes below certain level. In addition these algorithms suffer from the machine parameter uncertainties since the parameter variation in the reference model cannot be corrected. [19, 21] suggest an alternative MRAS based on the electromotive force rather than the rotor-flux as reference quantity for speed estimation where the integration problem has been overcome. Further in [21], another new auxiliary variable is introduced which represents the instantaneous reactive power for maintaining the magnetizing current. In this MRAS algorithm stator resistance disappear from the equations making the algorithm robust to that parameter. Zhen [22] proposed an interesting MRAS structure that is built with two mutual MRAS schemes. In this structure, the reference model and the adjustable models are interchangeable. For rotor speed estimation, one model is used as reference model and other model is used as adjustable model. The pure integration is removed from reference model. For stator resistance estimation the models switch their roles. [23-24] supported the MRAS scheme with ANN using its training and modeling of non-linear systems. MRAS scheme is also used for the on-line adaptation of the motor parameters in field oriented control techniques [25-26].

Kalman filter (KF) is another method employed to identify the speed and rotor-flux of an induction machine based on the measured quantities such as stator

current and voltage [27,28]. Kalman filter approach is based on the system model and a mathematical model describing the induction motor dynamics for the use of Kalman filter application. Parameter deviations and measurement disturbance are taken into consideration in KF. For this purpose covariance matrices of the KF must be properly initialized. KF itself works for linear systems, so for non-linear induction motor model extended Kalman filter (EKF) is used. However, KF approach is computationally intensive and depends on the accuracy of the model of the motor. In the EKF model proposed by [28], one can estimate rotor fluxes and rotor speed which makes the field orientation. EKF is also used for online parameter estimation of induction motor [29-31]. Reduced order models are also proposed to shorten and speed up the complex EKF algorithm [32]. A new KF technique for non-linear systems, Unscented Kalman Filter (UKF), is applied to induction machine state estimation in this thesis [33]. UKF is a derivative free KF technique which avoids costly calculation of Jacobian matrix, linearization and biasedness of the estimates [34-36].

Another method used for the sensorless control of induction motor is the neural network technique, which is based on a learning process. It has the advantage of tolerating machine parameter uncertainties. For speed estimation, a two-layered neural network, based on back propagation technique, is used and the neural network outputs are compared with the actual measurement values and error then back-propagated to adjust the weights such that the estimated speed converges to actual one. The neural network based sensorless control algorithms have the advantages of fault-tolerant characteristics. However, because of the neural network learning process these algorithms may suffer from the computational intensity.

Another approach is sliding mode control for FOC of induction machine. In the sliding mode technique, the control action is very strong and being switched into either “on” or “off” at high frequency. The command signals control directly the power devices. This type of control is also favorable because “on-off” is the only admissible mode of operation for the power converters. Therefore, it seems more natural to employ the algorithm towards discontinuous control.

In addition to the algorithms mentioned above, some of the proposed work is hard to classify because of their combined structure. For instance, [37-38] propose open-loop observer structures based on voltage model of the induction machine and attempt to avoid integration problem by using different low-pass filter structures. On

the other hand, some works use both voltage and current models of the induction machine to construct an open-loop observer structure and claims that rotor-flux estimation is insensitive to rotor time-constant variations. In [39], a nonlinear high-gain observer structure is proposed, and it is claimed that with the exact knowledge of stator resistance, flux and speed estimation convergence is guaranteed.

## 2.4 CONCLUSIONS

The literature review of DFOC, IFOC, flux, position and velocity estimation and speed control can be summarized as:

- The DFOC and IFOC are the methods for instantaneous torque and speed control of an induction motor drive system. These methods can be implemented with or without a speed sensor. An IFOC is synthesized by properly controlled slip-frequency which is necessary for the field-orientation.
- The main problem of an IFO drive system is the rotor time-constant deviation. The drive system torque control performance decreases if the rotor time-constant is not set precisely. Therefore, on-line estimation is necessary and is one of the main challenges for better performance of an IFOC. Most of the techniques proposed so far either need some special hardware or are very complex with respect to the software and require intensive calculations which put extra burden on the processor.
- The main problem in DFO control is precise rotor flux or position observation. This observation from terminal quantities is more desirable than the one including additional hardware.
- Voltage model and current model flux observers are the two most common ways to estimate the flux using the terminal quantities. The voltage model flux observer is dominated by stator IR drop at low speed, whereas the current model flux observer has problems of rotor time constant variations. Also the current model flux observer requires the rotor speed. Therefore, if the flux observer is being used for the sensorless control, an error in the estimated speed will be fed back in to the system. Thus will affect the observer accuracy.
- The proposed open-loop observers can be simple in the structure but they are susceptible to variety of errors that become specially detrimental at low stator

frequencies, including measurement, noise digital approximation errors, parameter detuning and DC offset in measurements, which ultimately may drive the observer instability.

- For the time-varying system model problems, closed-loop observers are proposed here feedback correction is used along with the machine model itself to improve the estimation accuracy. The algorithmic complexity and calculation intensity looks higher when compared with former solutions but the recent processors are fast enough to solve these algorithms in real-time applications. They also require a strong mathematical background to deal with. Their state estimation performance is studied in many applications and they are proved to be good alternatives for high performance ac drive area.

## CHAPTER 3

### INDUCTION MACHINE MODELING AND FOC SIMULATION

#### 3.1 The Induction Motor

The two names for the same type of motor, *Induction motor* and *Asynchronous motor*, describe the two characteristics in which this type of motor differs from DC motors and synchronous motors. Induction refers to the fact that the field in the rotor is induced by the stator currents, and asynchronous refers to the fact that the rotor speed is not equal to the stator frequency. No sliding contacts and permanent magnets are needed to make an induction motor work, which makes it very simple and cheap to manufacture. As motors, they rugged and require very little maintenance. However, their speeds are not as easily controlled as with DC motors. They draw large starting currents, and operate with a poor lagging factor when lightly loaded.

##### 3.1.1 Construction of the Three Phase Induction Motors (Physical Layout)

Most induction motors are of the rotary type with basically a stationary stator and a rotating rotor. The stator has a cylindrical magnetic core that is housed inside a metal frame. The stator magnetic core is formed by stacking thin electrical steel laminations with uniformly spaced slots stamped in the inner circumference to accommodate the three distributed stator windings. The stator windings are formed by connecting coils of copper or aluminum conductors that are insulated from the slot walls.

The rotor consists of a cylindrical laminated iron core with uniformly spaced peripheral slots to accommodate the rotor windings. In this thesis a squirrel cage rotor

induction motor is used. It has uniformly spaced axial bars that are soldered onto end rings at both ends. After the rotor core laminations are stacked in a mold, the mold is filled with molten aluminum. There is no insulation between the bars and alls of the rotor slots.

### 3.2 Mathematical Model of Induction Motor

During the entire report, a complex vector notation and some reference frame conversions are used. Since this is quite essential to the understanding of the rest of the theory, it will shortly be described in the next subsection.

#### 3.2.1 Three-Phase Transformations

In the study of generalized machine theory, mathematical transformations are often used to decouple variables, to facilitate the solutions of difficult equations with time varying coefficients, or to refer all variables to a common reference frame [39].

The most commonly used transformation is the polyphase to orthogonal two-phase (or two-axis) transformation. For the n-phase to two-phase case, it can be expressed in the form:

$$[f_{xy}] = [T(\theta)][f_{1,2,\dots,n}] \quad (3.1)$$

where

$$[T(\theta)] = \sqrt{\frac{2}{n}} \begin{bmatrix} \cos \frac{p}{2} \theta & \cos \left( \frac{p}{2} \theta - \alpha \right) & \dots & \cos \left( \frac{p}{n} \theta - (n-1)\alpha \right) \\ \sin \frac{p}{2} \theta & \sin \left( \frac{p}{2} \theta - \alpha \right) & \dots & \sin \left( \frac{p}{n} \theta - (n-1)\alpha \right) \end{bmatrix} \quad (3.2)$$

and  $\alpha$  is the electrical angle between the two adjacent magnetic axes of a uniformly distributed n-phase windings. The coefficient  $\sqrt{2/n}$ , is introduced to make the transformation power invariant.

Important subsets of the general n-phase to two-phase transformation, though not necessarily power invariant, are briefly discussed in the following part.

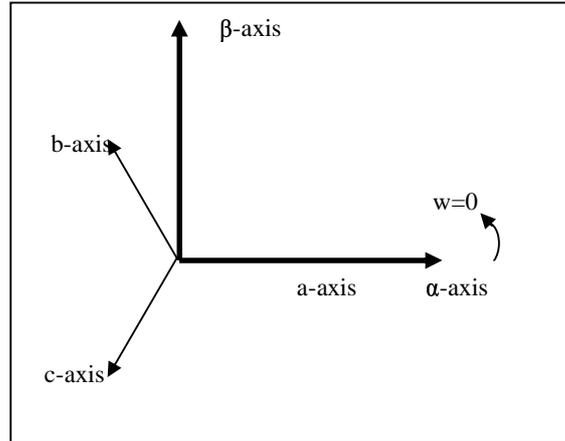


Fig.3.1- Relationship between the  $\alpha\beta$  abc quantities

### 3.2.2 Clark Transformation

The Clark transformation is basically employed to transform three-phase to two-phase quantities. The two-phase variables in stationary reference frame are sometimes denoted as  $\alpha$  and  $\beta$ . As shown in Fig.3.1 the  $\alpha$ -axis coincides with the phase-a axis and the  $\beta$ -axis lags the  $\alpha$ -axis by  $90^\circ$ .

$$[f_{\alpha\beta 0}] = [T_{\alpha\beta 0}][f_{abc}] \quad (3.3)$$

where the transformation matrix,  $[T_{\alpha\beta 0}]$ , is given by:

$$[T_{\alpha\beta 0}] = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (3.4)$$

The inverse transformation is:

$$[T_{\alpha\beta 0}]^{-1} = \begin{bmatrix} 1 & 0 & 1 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 1 \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 1 \end{bmatrix} \quad (3.5)$$

### 3.2.3 Park Transformation

The Park's transformation is a well-known transformation that converts the quantities to to-phase synchronously rotating frame. The transformation is in the form of:

$$[f_{dq0}] = [T_{dq0}(\theta_d)][f_{abc}] \quad (3.6)$$

where the dq0 transformation matrix is defined as :

$$[T_{dq0}(\theta_d)] = \frac{2}{3} \begin{bmatrix} \cos \theta_d & \cos\left(\theta_d - \frac{2\pi}{3}\right) & \cos\left(\theta_d + \frac{2\pi}{3}\right) \\ -\sin \theta_d & -\sin\left(\theta_d - \frac{2\pi}{3}\right) & -\sin\left(\theta_d + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (3.7)$$

and the inverse is given by:

$$[T_{dq0}(\theta_d)]^{-1} = \begin{bmatrix} \cos \theta_d & -\sin \theta_d & 1 \\ \cos\left(\theta_d - \frac{2\pi}{3}\right) & -\sin\left(\theta_d - \frac{2\pi}{3}\right) & 1 \\ \cos\left(\theta_d + \frac{2\pi}{3}\right) & -\sin\left(\theta_d + \frac{2\pi}{3}\right) & 1 \end{bmatrix} \quad (3.8)$$

where the  $\theta_d$  is the transformation angle.

The positive q-axis is defined as leading the positive d-axis by  $90^0$  in the original Park's transformation. Some authors define the q-axis as lagging the d-axis by  $90^0$ . The transformation with q-axis lagging d-axis is given by:

$$[T_{qd0}(\theta_d)] = \frac{2}{3} \begin{bmatrix} \cos \theta_q & \cos\left(\theta_q - \frac{2\pi}{3}\right) & \cos\left(\theta_q + \frac{2\pi}{3}\right) \\ \sin \theta_q & \sin\left(\theta_q - \frac{2\pi}{3}\right) & \sin\left(\theta_q + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (3.9)$$

with an inverse given by:

$$[T_{qd0}(\theta_q)]^{-1} = \begin{bmatrix} \cos \theta_q & \sin \theta_q & 1 \\ \cos\left(\theta_q - \frac{2\pi}{3}\right) & \sin\left(\theta_q - \frac{2\pi}{3}\right) & 1 \\ \cos\left(\theta_q + \frac{2\pi}{3}\right) & \sin\left(\theta_q + \frac{2\pi}{3}\right) & 1 \end{bmatrix} \quad (3.10)$$

where  $\theta_q$  is the transformation angle when q-axis is leading.

The relationship between the  $\theta_d$  and  $\theta_q$  is:

$$\theta_q = \theta_d + \frac{\pi}{2} \quad (3.11)$$

One can show that  $[T_{dq0}]$  and  $[T_{qd0}]$ , are basically the same, except for the ordering of the d and q variables. Both of the alternatives are shown in Fig.3.2 and Fig.3.3.

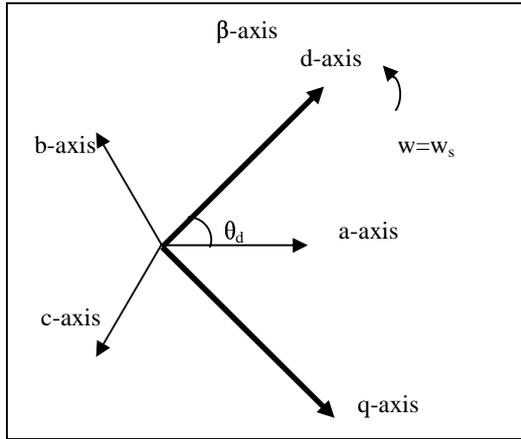


Fig.3.2- Relationship between the dq and the abc quantities

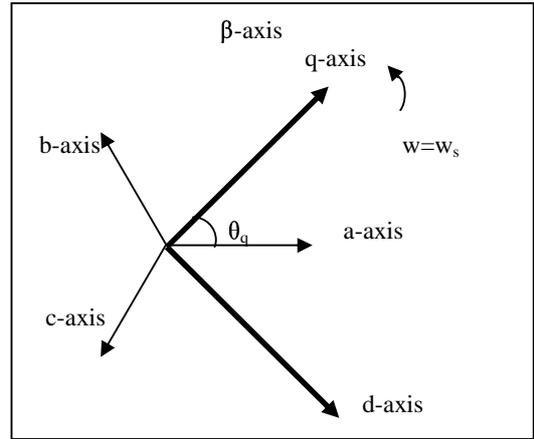


Fig.3.3 - Relationship between the qd and the abc quantities

### 3.3 Circuit Model of a Three Phase Induction Motor

Using the coupled circuit approach and motor notation, the voltage equations of the magnetically coupled stator a rotor circuit can be written as follows:

Stator Voltage Equations:

$$\begin{aligned} V_{as} &= i_{as} r_s + \frac{d\psi_{as}}{dt} \quad V \\ V_{bs} &= i_{bs} r_s + \frac{d\psi_{bs}}{dt} \quad V \\ V_{cs} &= i_{cs} r_s + \frac{d\psi_{cs}}{dt} \quad V \end{aligned} \quad (3.12)$$

Rotor Voltage Equations:

$$\begin{aligned} V_{ar} &= i_{ar} r_r + \frac{d\psi_{ar}}{dt} \quad V \\ V_{br} &= i_{br} r_r + \frac{d\psi_{br}}{dt} \quad V \\ V_{cr} &= i_{cr} r_r + \frac{d\psi_{cr}}{dt} \quad V \end{aligned} \quad (3.13)$$

In matrix notation, the flux linkages of the stator and rotor windings, in terms of the winding inductances and currents, may be written compactly as

$$\begin{bmatrix} \Psi_s^{abc} \\ \Psi_r^{abc} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{ss}^{abc} & \mathbf{L}_{sr}^{abc} \\ \mathbf{L}_{rs}^{abc} & \mathbf{L}_{rr}^{abc} \end{bmatrix} \begin{bmatrix} \mathbf{i}_s^{abc} \\ \mathbf{i}_r^{abc} \end{bmatrix} \text{ Wb.turns} \quad (3.14)$$

where

$$\begin{aligned} \Psi_s^{abc} &= (\Psi_{as}, \Psi_{bs}, \Psi_{cs})^t \\ \Psi_r^{abc} &= (\Psi_{ar}, \Psi_{br}, \Psi_{cr})^t \\ \mathbf{i}_s^{abc} &= (i_{as}, i_{bs}, i_{cs})^t \\ \mathbf{i}_r^{abc} &= (i_{ar}, i_{br}, i_{cr})^t \end{aligned} \quad (3.15)$$

and the superscript T denotes the transpose of the array.

The sub-matrices of the stator-to-rotor and rotor-to-rotor winding inductances are of the form:

$$\mathbf{L}_{ss}^{abc} = \begin{bmatrix} \mathbf{L}_{ls} + \mathbf{L}_{ss} & \mathbf{L}_{sm} & \mathbf{L}_{sm} \\ \mathbf{L}_{sm} & \mathbf{L}_{ls} + \mathbf{L}_{ss} & \mathbf{L}_{sm} \\ \mathbf{L}_{sm} & \mathbf{L}_{sm} & \mathbf{L}_{ls} + \mathbf{L}_{ss} \end{bmatrix} \text{ H} \quad (3.16)$$

$$\mathbf{L}_{rr}^{abc} = \begin{bmatrix} \mathbf{L}_r + \mathbf{L}_{rr} & \mathbf{L}_{rm} & \mathbf{L}_{rm} \\ \mathbf{L}_{rm} & \mathbf{L}_r + \mathbf{L}_{rr} & \mathbf{L}_{rm} \\ \mathbf{L}_{rm} & \mathbf{L}_{rm} & \mathbf{L}_r + \mathbf{L}_{rr} \end{bmatrix} \text{ H}$$

Those of the stator-to-rotor mutual inductances are dependent on the rotor angle, that is:

$$\mathbf{L}_{sr}^{abc} = [\mathbf{L}_{rs}^{abc}]^t = \mathbf{L}_{sr} \begin{bmatrix} \cos \theta_r & \cos\left(\theta_r + \frac{2\pi}{3}\right) & \cos\left(\theta_r - \frac{2\pi}{3}\right) \\ \cos\left(\theta_r - \frac{2\pi}{3}\right) & \cos \theta_r & \cos\left(\theta_r + \frac{2\pi}{3}\right) \\ \cos\left(\theta_r + \frac{2\pi}{3}\right) & \cos\left(\theta_r - \frac{2\pi}{3}\right) & \cos \theta_r \end{bmatrix} \text{ H} \quad (3.17)$$

where  $L_{ls}$  is the per phase stator winding leakage inductance,  $L_{lr}$  is the per phase rotor winding leakage inductance,  $L_{ss}$  is the self inductance of the stator winding,  $L_{rr}$  is the self inductance of the rotor winding,  $L_{sm}$  is the mutual inductance between stator windings,  $L_{rm}$  is the mutual inductance between rotor windings, and  $L_{sr}$  is the peak value of the stator to rotor mutual inductance.

Note that the idealized machine is described by six first-order differential equations, one for each winding. These differential equations are coupled to one

another through the mutual inductance between the windings. In particular, the stator-to-rotor coupling terms vary with time. Transformations like the dq or  $\alpha\beta$  can facilitate the computation of the transient solution of the above induction motor model by transforming the differential equations with time-varying inductances to differential equations with constant inductances.

### 3.4 Machine Model in Arbitrary dq0 Reference Frame

The idealized three-phase induction machine is assumed to have symmetrical airgap. The dq0 reference frames are usually selected on the basis of conveniences or computational reduction. The two commonly used reference frames in the analysis of induction machine are the stationary and synchronously rotating frames. Each has an advantage for some purpose. In the stationary rotating reference, the dq variables of the machine are in the same frame as those normally used for the supply network. In the synchronously rotating frame, the dq variables are steady in steady-state. Here, firstly the equations of the induction machine in an arbitrary reference frame which is rotating at a speed of ( $w$ ) in the direction of the rotor rotation will be derived. Those if the induction machine in the stationary frame can then be obtained by setting  $w=0$ , and those for the synchronously rotating frame are obtained by setting  $w = w_e$ . The relationship between the abc quantities and dq0 quantities of a reference frame rotating at an angular speed,  $w$ , is shown in Fig.3.4.

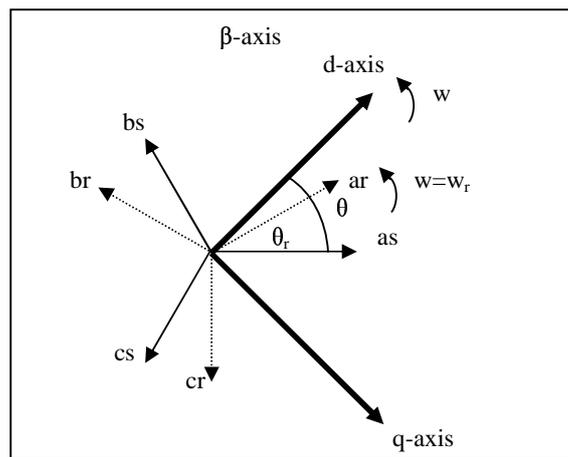


Fig.3.4- Relationship between abc and arbitrary dq0

The transformation equation from abc to this dq0 reference frame is given by:

$$\begin{bmatrix} f_q \\ f_d \\ f_0 \end{bmatrix} = [T_{qd0}(\theta)] \begin{bmatrix} f_a \\ f_b \\ f_c \end{bmatrix} \quad (3.18)$$

where the variable f can be the phase voltages, current, or flux linkages of the machine. The transformation angle,  $\theta(t)$ , between the q-axis of the reference frame rotating at a speed of  $w$  and the a-axis of the stationary stator winding may be expressed as :

$$\theta(t) = \int_0^t w(t)dt + \theta(0) \quad \text{elec. rad.} \quad (3.19)$$

Likewise, the rotor angle,  $\theta_r(t)$ , between the axes of the stator and rotor a-phases for a rotor rotating with speed  $w_r(t)$  may be expressed as:

$$\theta_r(t) = \int_0^t w_r(t)dt + \theta_r(0) \quad \text{elec. rad.} \quad (3.20)$$

### 3.4.1 dq0 Voltage Equations

In matrix notation, the stator winding abc voltage equations can be expressed as:

$$v_s^{abc} = p\psi_s^{abc} + r_s^{abc}i_s^{abc} \quad (3.21)$$

Applying the transformations given in (3.7) and (3.8), to the voltage, current and flux linkages eqn. (3.21) becomes

$$v_s^{qd0} = [T_{qd0}(\theta)]p[T_{qd0}(\theta)]^{-1}[\psi_s^{qd0}] + [T_{qd0}(\theta)]r_s[T_{qd0}(\theta)]^{-1}[i_s^{qd0}] \quad (3.22)$$

solving the equation above it becomes:

$$v_s^{qd0} = w \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \psi_s^{qd0} + p\psi_s^{qd0} + r_s^{qd0}i_s^{qd0} \quad (3.23)$$

where

$$w = \frac{d\theta}{dt} \quad \text{and} \quad r_s^{qd0} = r_s \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.24)$$

Likewise, the rotor voltage equation becomes:

$$v_r^{qd0} = (\omega - \omega_r) \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \psi_r^{qds} + p\psi_r^{qd0} + r_r^{qd0} i_r^{qd0} \quad (3.25)$$

### 3.4.2 qd0 Flux Linkage Relation

The stator qd0 flux linkages are obtained by applying  $T_{qd0}(\theta)$  to the stator abc flux linkages in (3.14).

$$\psi_s^{qd0} = [T_{qd0}(\theta)] (L_{ss}^{abc} i_{ss}^{abc} + L_{sr}^{abc} i_{sr}^{abc}) \quad (3.26)$$

skipping the transformation steps the stator and the rotor flux linkage relationships can be expressed compactly:

$$\begin{bmatrix} \psi_{qs} \\ \psi_{ds} \\ \psi_{0s} \\ \psi_{qr} \\ \psi_{dr} \\ \psi_{0r} \end{bmatrix} = \begin{bmatrix} L_{sl} + L_m & 0 & 0 & L_m & 0 & 0 \\ 0 & L_{sl} + L_m & 0 & 0 & L_m & 0 \\ 0 & 0 & L_{ls} & 0 & 0 & 0 \\ L_m & 0 & 0 & L'_{lr} + L_m & 0 & 0 \\ 0 & L_m & 0 & 0 & L'_{lr} + L_m & 0 \\ 0 & 0 & 0 & 0 & 0 & L_m \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{0s} \\ i_{qr} \\ i_{dr} \\ i_{0r} \end{bmatrix}$$

(3.27)

Substituting the (3.27) into voltage equations and then grouping q, d, 0, and  $\theta$  terms in the resulting voltage equations, we obtain the voltage equations that suggest the equivalent circuit shown in Fig.3.5.

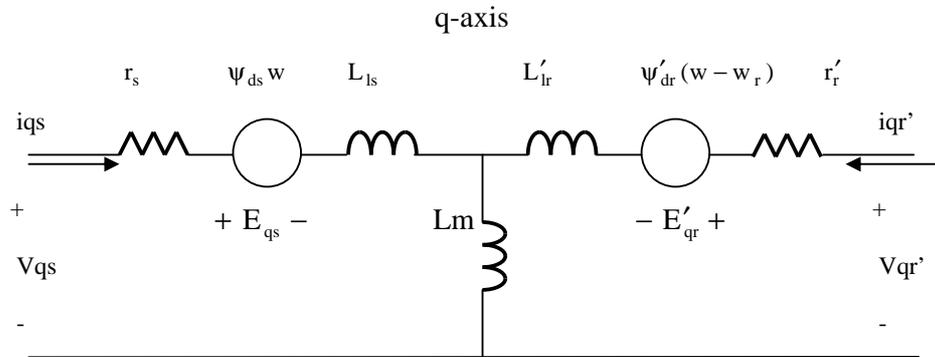


Fig.3.5- Equivalent circuit representation of an induction machine in the arbitrary reference frame

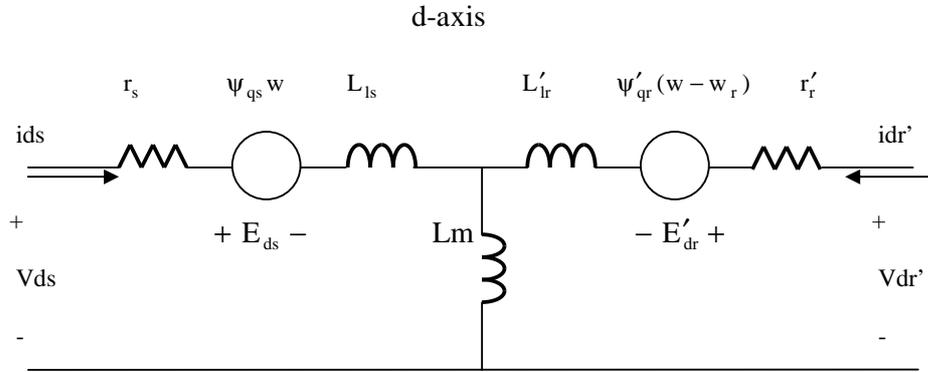


Fig.3.5- (continued) Equivalent circuit representation of an induction machine in the arbitrary reference frame

### 3.4.3 qd0 Torque Equations

The sum of the instantaneous input power to all six windings of the stator and rotor is given by :

$$p_{in} = v_{as}i_{as} + v_{bs}i_{bs} + v_{cs}i_{cs} + v'_{ar}i'_{ar} + v'_{br}i'_{br} + v'_{cr}i'_{cr} \quad W \quad (3.28)$$

in terms of dq quantities

$$p_{in} = \frac{3}{2}(v_{qs}i_{qs} + v_{ds}i_{ds} + 2v_{0s}i_{0s} + v'_{qr}i'_{qr} + v'_{dr}i'_{dr} + 2v'_{0r}i'_{0r}) \quad W \quad (3.29)$$

Using stator and rotor voltages to substitute for the voltages on the right hand side of (3.29) , we obtain three kinds of terms:  $i^2r$ ,  $i p \psi$ , and  $w \psi i$ . ( $i^2r$ ) terms are the copper losses. The ( $i.p\psi$ ) terms represent the rate of exchange of magnetic field energy between windings. The electromechanical torque developed by the machine is given by the sum of the ( $w.\psi i$ ) terms divided by mechanical speed, that is:

$$T_{em} = \frac{3}{2} \frac{p}{2w_r} [w(\psi_{ds}i_{qs} - \psi_{qs}i_{ds}) + (w - w_r)(\psi'_{dr}i'_{qr} - \psi'_{qr}i'_{dr})] \quad Nm \quad (3.30)$$

using the flux linkage relationships,  $T_{em}$  can also be expressed as follows:

$$T_{em} = \frac{3}{2} \frac{p}{2w_r} [w(\psi_{ds}i_{qs} - \psi_{qs}i_{ds}) + (w - w_r)(\psi'_{dr}i'_{qr} - \psi'_{qr}i'_{dr})] \quad Nm \quad (3.31)$$

Using the flux linkage relationships, one can show that

$$\begin{aligned}
 T_{em} &= \frac{3}{2} \frac{P}{2} (\Psi'_{qr} i'_{dr} - \Psi'_{dr} i'_{qr}) \quad \text{Nm} \\
 &= \frac{3}{2} \frac{P}{2} (\Psi_{ds} i_{qs} - \Psi_{qs} i_{ds}) \quad \text{Nm} \\
 &= \frac{3}{2} \frac{P}{2} L_m (i'_{dr} i_{qs} - i'_{qr} i_{ds}) \quad \text{Nm}
 \end{aligned} \tag{3.32}$$

One can rearrange the torque equations by inserting the speed voltage terms given below:

$$\begin{aligned}
 E_{qs} &= w \Psi_{ds} & E_{ds} &= -w \Psi_{qs} \\
 E'_{qr} &= (w - w_r) \Psi'_{dr} & E'_{dr} &= -(w - w_r) \Psi'_{qr}
 \end{aligned} \tag{3.33}$$

Table 3.1- Induction Machine Equations in Arbitrary Reference Frame

*Stator qd0 voltage equations:*

$$\begin{aligned}
 v_{qs} &= p \Psi_{qs} + w \Psi_{ds} + r_s i_{qs} \\
 v_{ds} &= p \Psi_{ds} - w \Psi_{qs} + r_s i_{ds} \\
 v_{0s} &= p \Psi_{0s} + r_s i_{0s}
 \end{aligned} \tag{3.34}$$

*Rotor qd0 voltage equations:*

$$\begin{aligned}
 v'_{qr} &= p \Psi'_{qr} + (w - w_r) \Psi'_{dr} + r'_r i'_{qr} \\
 v'_{dr} &= p \Psi'_{dr} - (w - w_r) \Psi'_{qr} + r'_r i'_{dr} \\
 v'_{0r} &= p \Psi'_{0r} + r'_r i'_{0r}
 \end{aligned} \tag{3.35}$$

where

$$\begin{bmatrix} \Psi_{qs} \\ \Psi_{ds} \\ \Psi_{0s} \\ \Psi_{qr} \\ \Psi_{dr} \\ \Psi_{0r} \end{bmatrix} = \begin{bmatrix} L_{sl} + L_m & 0 & 0 & L_m & 0 & 0 \\ 0 & L_{sl} + L_m & 0 & 0 & L_m & 0 \\ 0 & 0 & L_{ls} & 0 & 0 & 0 \\ L_m & 0 & 0 & L'_{lr} + L_m & 0 & 0 \\ 0 & L_m & 0 & 0 & L'_{lr} + L_m & 0 \\ 0 & 0 & 0 & 0 & 0 & L_m \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{0s} \\ i_{qr} \\ i_{dr} \\ i_{0r} \end{bmatrix} \tag{3.36}$$

Table 3.1- (continued) Ind. Machine Equations in Arbitrary Reference Frame

*Torque Equations:*

$$T_{em} = \frac{3}{2} \frac{P}{2w_r} \left[ w(\Psi_{ds}i_{qs} - \Psi_{qs}i_{ds}) + (w - w_r)(\Psi'_{dr}i'_{qr} - \Psi'_{qr}i'_{dr}) \right] \quad \text{Nm} \quad (3.37)$$

$$\begin{aligned} T_{em} &= \frac{3}{2} \frac{P}{2} (\Psi'_{qr}i'_{dr} - \Psi'_{dr}i'_{qr}) \quad \text{Nm} \\ &= \frac{3}{2} \frac{P}{2} (\Psi_{ds}i_{qs} - \Psi_{qs}i_{ds}) \quad \text{Nm} \\ &= \frac{3}{2} \frac{P}{2} L_m (i'_{dr}i_{qs} - i'_{qr}i_{ds}) \quad \text{Nm} \end{aligned} \quad (3.38)$$

### 3.5 qd0 Stationary and Synchronous Reference Frames

There is seldom a need to simulate an induction machine in the arbitrary rotating reference frame. But it is useful to convert a unified model to other frames. The most commonly used ones are, two marginal cases of the arbitrary rotating frame, stationary reference frame and synchronously rotating frame. For transient studies of adjustable speed drives, it is usually more convenient to simulate an induction machine and its converter on a stationary reference frame. Moreover, calculations with stationary reference frame is less complex due to zero frame speed (some terms cancelled). For small signal stability analysis about some operating condition, a synchronously rotating frame which yields steady values of steady-state voltages and currents under balanced conditions is used.

Since we have derived the equations of the induction machine for the general case, that is in the arbitrary rotating reference frame, the equations of the machine in the stationary and synchronously rotating reference frame,  $w$  to zero and  $w_e$ , respectively. To distinguish these two frames from each other, an additional superscript will be used,  $s$  for stationary frame variables and  $e$  for synchronously rotating frame variables.

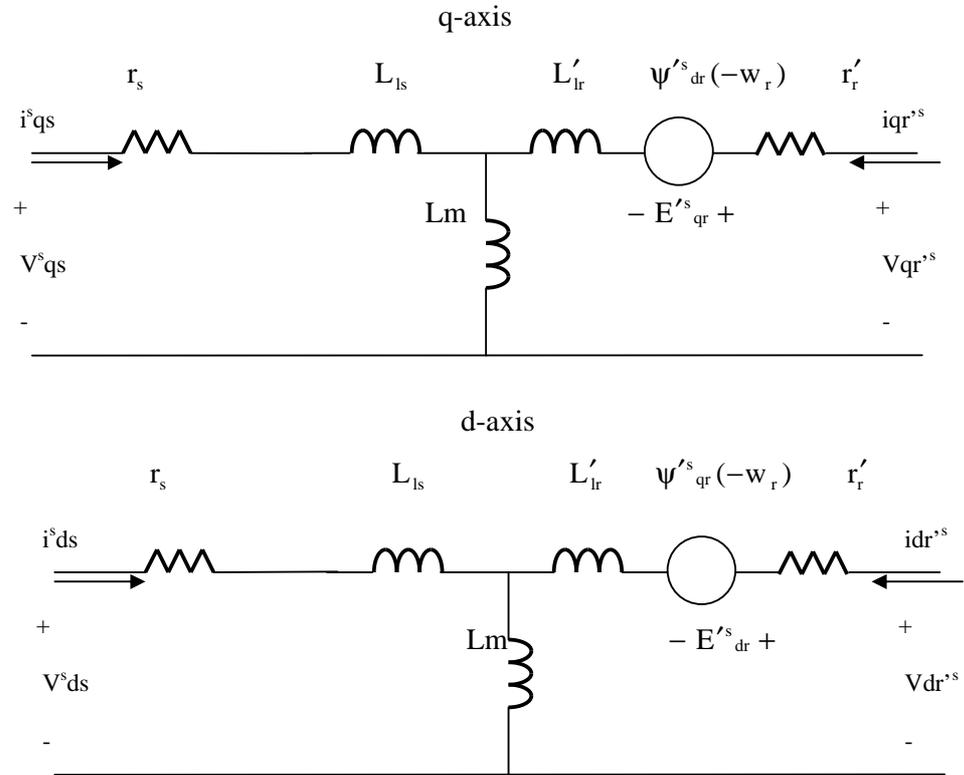


Fig3.6- Equivalent circuit of an induction machine in the stationary frame

Table 3.2-Induction Machine Equations in Stationary Reference Frame

<p><i>Stator qd0 voltage equations:</i></p> $  \begin{aligned}  v^s_{qs} &= p\psi^s_{qs} + r_s i^s_{qs} \\  v^s_{ds} &= p\psi^s_{ds} + r_s i^s_{ds} \\  v_{0s} &= p\psi^s_{0s} + r_s i^s_{0s}  \end{aligned}  $ <p><i>Rotor qd0 voltage equations:</i></p> $  \begin{aligned}  v'^s_{qr} &= p\psi'^s_{qr} + (-w_r)\psi'^s_{dr} + r'_r i'^s_{qr} \\  v'^s_{dr} &= p\psi'^s_{dr} + (w_r)\psi'^s_{qr} + r'_r i'^s_{dr} \\  v'_{0r} &= p\psi'_{0r} + r'_r i'_{0r}  \end{aligned}  $	<p>(3.39)</p> <p>(3.40)</p>
---	-----------------------------

Table 3.2- (continued) Induction Machine Equations in Stationary Reference Frame

where

$$\begin{bmatrix} \Psi_{qs}^s \\ \Psi_{ds}^s \\ \Psi_{0s}^s \\ \Psi_{qr}^s \\ \Psi_{dr}^s \\ \Psi_{0r}^s \end{bmatrix} = \begin{bmatrix} L_{sl} + L_m & 0 & 0 & L_m & 0 & 0 \\ 0 & L_{sl} + L_m & 0 & 0 & L_m & 0 \\ 0 & 0 & L_{ls} & 0 & 0 & 0 \\ L_m & 0 & 0 & L'_{lr} + L_m & 0 & 0 \\ 0 & L_m & 0 & 0 & L'_{lr} + L_m & 0 \\ 0 & 0 & 0 & 0 & 0 & L_m \end{bmatrix} \begin{bmatrix} i_{qs}^s \\ i_{ds}^s \\ i_{0s}^s \\ i_{qr}^s \\ i_{dr}^s \\ i_{0r}^s \end{bmatrix} \quad (3.41)$$

*Torque Equations:*

$$\begin{aligned} T_{em} &= \frac{3P}{2} (\Psi_{qr}^s i'_{dr} - \Psi_{dr}^s i'_{qr}) \quad \text{Nm} \\ &= \frac{3P}{2} (\Psi_{ds}^s i_{qs}^s - \Psi_{qs}^s i_{ds}^s) \quad \text{Nm} \\ &= \frac{3P}{2} L_m (i'_{dr} i_{qs}^s - i'_{qr} i_{ds}^s) \quad \text{Nm} \end{aligned} \quad (3.42)$$

The equivalent induction machine circuit and induction machine equations in the stationary reference frame are given above in Table 3.2 and Fig.3.6. In Fig3.7, 3-phase AC quantities are simulated in both stationary frame and synchronously rotating frame.

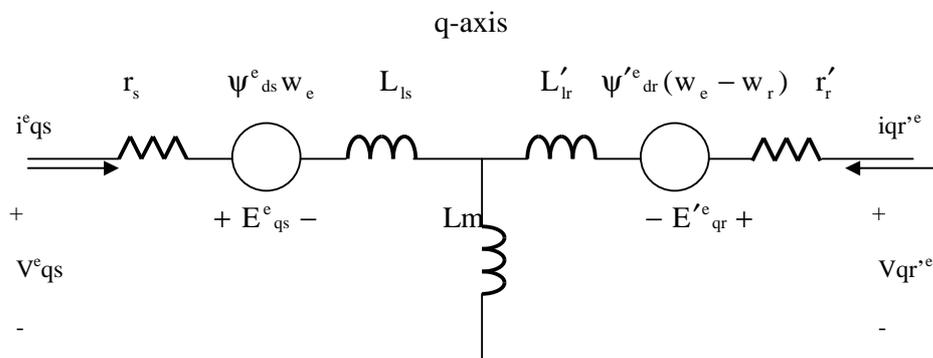


Fig.3.7- Equivalent circuit of an induction machine in the synchronously rotating frame

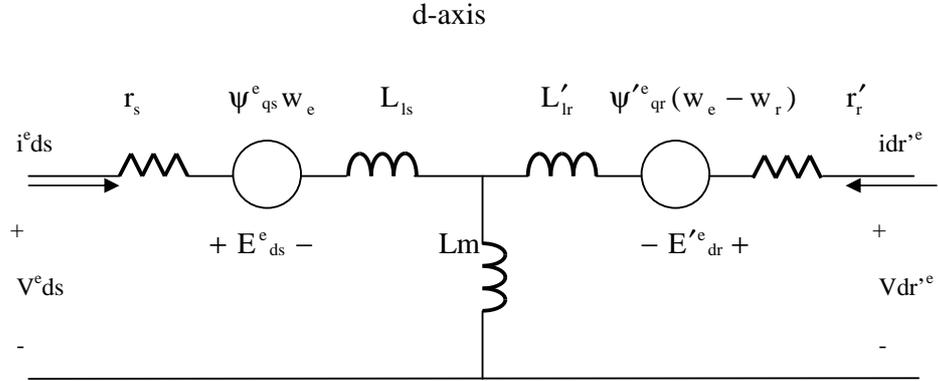


Fig.3.7- (continued) Equivalent cct of an induction machine in the synchronously rotating frame

Table3.3- Induction Machine Equations in Synchronously Rotating Reference Frame

*Stator qd0 voltage equations:*

$$\begin{aligned}
 v_{qs}^e &= p\psi_{qs}^e + \omega_e \psi_{ds}^e + r_s i_{qs}^e \\
 v_{ds}^e &= p\psi_{ds}^e - \omega_e \psi_{qs}^e + r_s i_{ds}^e \\
 v_{0s} &= p\psi_{0s} + r_s i_{0s}
 \end{aligned} \tag{4.43}$$

*Rotor qd0 voltage equations:*

$$\begin{aligned}
 v_{qr}^e &= p\psi_{qr}^e + (\omega_e - \omega_r) \psi_{dr}^e + r'_r i_{qr}^e \\
 v_{dr}^e &= p\psi_{dr}^e - (\omega_e - \omega_r) \psi_{qr}^e + r'_r i_{dr}^e \\
 v'_{0r} &= p\psi'_{0r} + r'_r i'_{0r}
 \end{aligned} \tag{4.44}$$

where

$$\begin{bmatrix} \psi_{qs}^e \\ \psi_{ds}^e \\ \psi_{0s} \\ \psi_{qr}^e \\ \psi_{dr}^e \\ \psi_{0r} \end{bmatrix} = \begin{bmatrix} L_{sl} + L_m & 0 & 0 & L_m & 0 & 0 \\ 0 & L_{sl} + L_m & 0 & 0 & L_m & 0 \\ 0 & 0 & L_{ls} & 0 & 0 & 0 \\ L_m & 0 & 0 & L'_{lr} + L_m & 0 & 0 \\ 0 & L_m & 0 & 0 & L'_{lr} + L_m & 0 \\ 0 & 0 & 0 & 0 & 0 & L_m \end{bmatrix} \begin{bmatrix} i_{qs}^e \\ i_{ds}^e \\ i_{0s} \\ i_{qr}^e \\ i_{dr}^e \\ i_{0r} \end{bmatrix} \tag{3.45}$$

Table3.3-(continued) Induction Machine Equations in Synchronously Rotating Reference Frame

*Torque Equations:*

$$\begin{aligned}
 T_{em} &= \frac{3}{2} \frac{P}{2} (\psi'_{qr} i'_{dr} - \psi'_{dr} i'_{qr}) \quad \text{Nm} \\
 &= \frac{3}{2} \frac{P}{2} (\psi^e_{ds} i^e_{qs} - \psi^e_{qs} i^e_{ds}) \quad \text{Nm} \\
 &= \frac{3}{2} \frac{P}{2} L_m (i'_{dr} i^e_{qs} - i'_{qr} i^e_{ds}) \quad \text{Nm}
 \end{aligned}
 \tag{3.46}$$

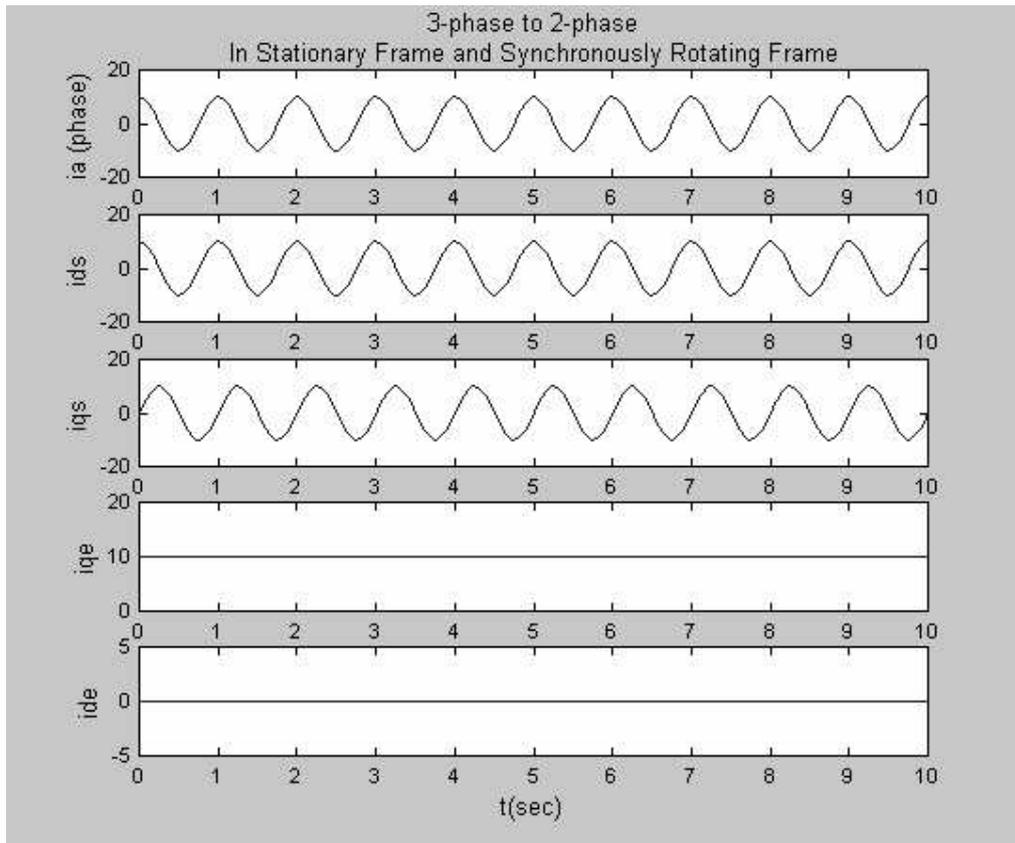


Fig.3.8- A simulation of 3-phase AC quantities converted to both stationary frame ( $i_{qs}, i_{ds}$ ) and synchronously rotating frame ( $i_{qe}, i_{de}$ )

### 3.6 Simulation of the Induction Motor in Stationary Frame

Using the stationary frame equations (3.39)-(3.42) induction motor is simulated in a stationary reference frame and used in the development of field orientation control techniques and state estimation techniques. Applying the appropriate voltages to the motor model either those obtained by using feedback information or direct open-loop voltage, one can observe the torque-speed responses and current-flux waveforms. Using this information, different alternative control techniques may be tested and developed.

In this model, three-phase voltages applied to the input are converted into two-phase stationary reference frame voltages. Once d-q phase voltages obtained, using the equations in Table 3.2 associated flux and current are calculated and then applied to electromechanical and mechanical torque equations to obtain torque-speed responses. Based on the stationary reference frame model Fig.3.9 shows the stator voltage, the stator current, the torque and the speed waveforms at no-load for a 1-hp motor. Torque vs speed curve obtained from the same model is shown in Fig.3.10 for no-load condition.

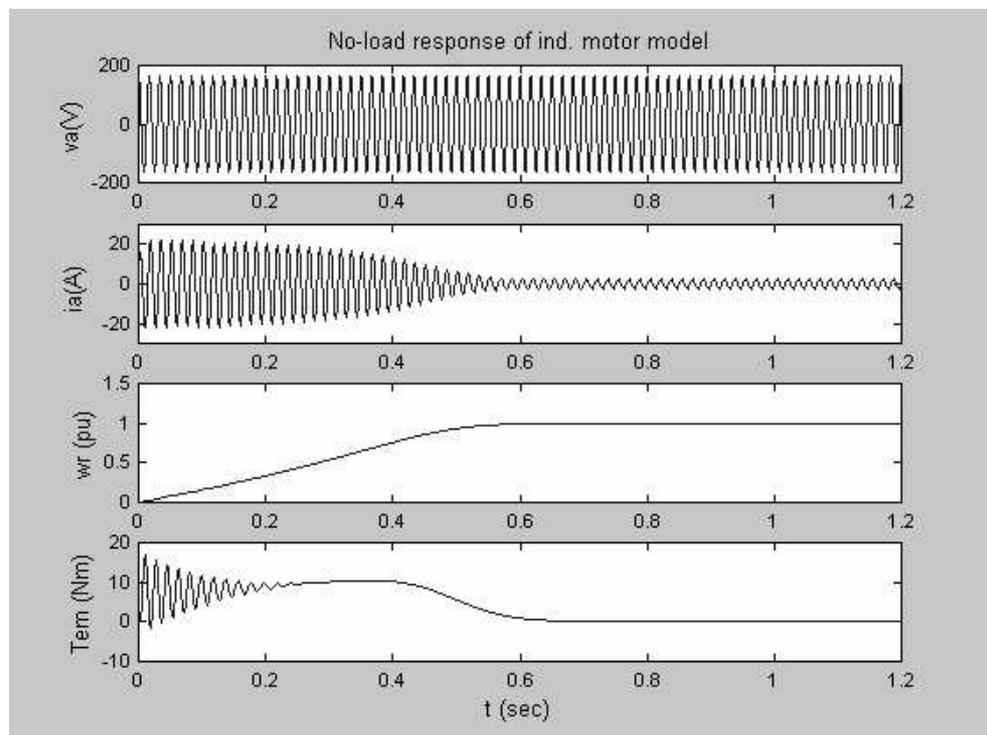


Fig.3.9- No-Load Response of Stationary Frame Induction Motor Model

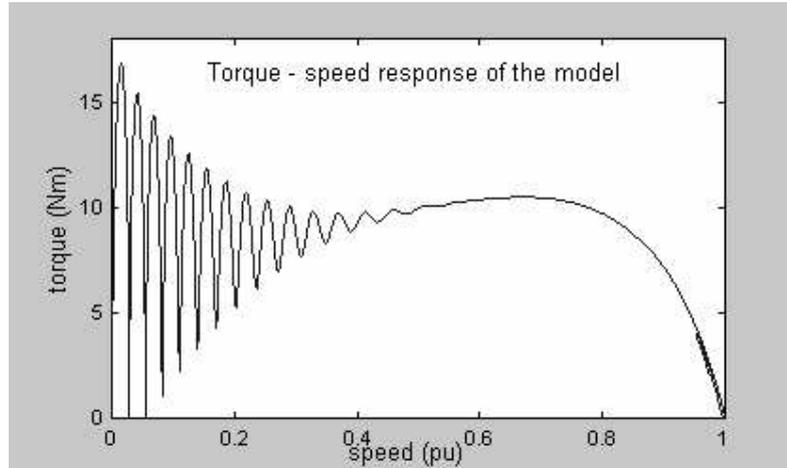


Fig.3.10- Open-loop torque-speed curve of the induction motor model at no-load

Induction machine model being non-linear it is needed for some cases to be linearized at many different operating points to make use of linear control techniques. Especially, the linear computation techniques based on the state-space model needs the use of linearized model of the induction machine at instantaneous operating points to define A,B,C,D matrices. Desired operating points may be found by using *trim* function in MATLAB. After that, *linmod* function is used to determine the A,B,C,D matrices of the small-signal model of the non-linear system about the chosen steady-state operating point. Furthermore, *ss2tf* command is used to determine the transfer function of the system at the chosen operating point whose instantaneous state-space matrices are calculated [39]. After calculating these steps one may conduct study on the stability analysis of the model.

For the induction machine model stability analysis, two-phase stationary frame voltages and applied mechanical load are considered as inputs. Two-phase stationary frame currents, electromechanical torque and rotor speed are considered as outputs of the system in state space representation of the model. In Fig.3.11 changing input3 (applied load) from zero to twice the rated torque, shift of the poles is observed. It is confirmed that poles of the transfer function of  $(\Delta w_r/\Delta T_{mech})$  are all on the left hand side of the real-axis. For a detailed stability analysis other transfer functions of the different output-input combinations may be investigated in the operational range. In addition to pole path, the stability analysis may be enriched using *rlocus* command to arrange the gains of the system; those do not exceed the stability limits using real machine parameters. Fig.3.12 and Fig.3.13 are the root locus examples of two different state transfer functions of the motor.

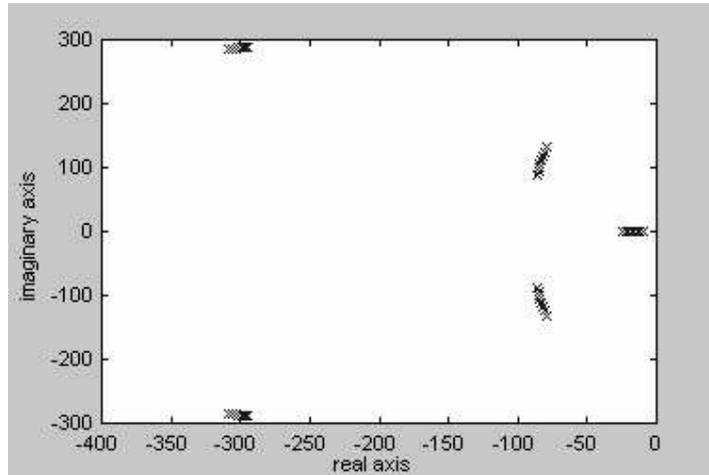


Fig.3.11-Pole paths of  $(\Delta w_r/\Delta T_{\text{mech}})$  from no-load to twice the rated torque

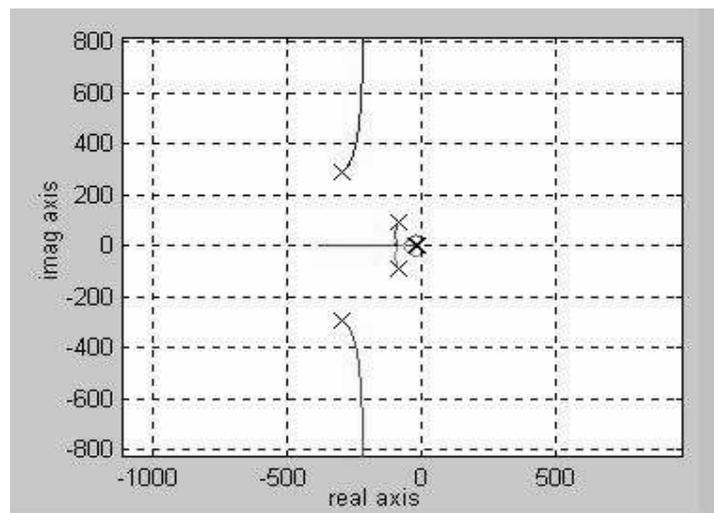


Fig.3.12 Root locus of  $(\Delta w_r/\Delta v_{qse})$  for varying gains

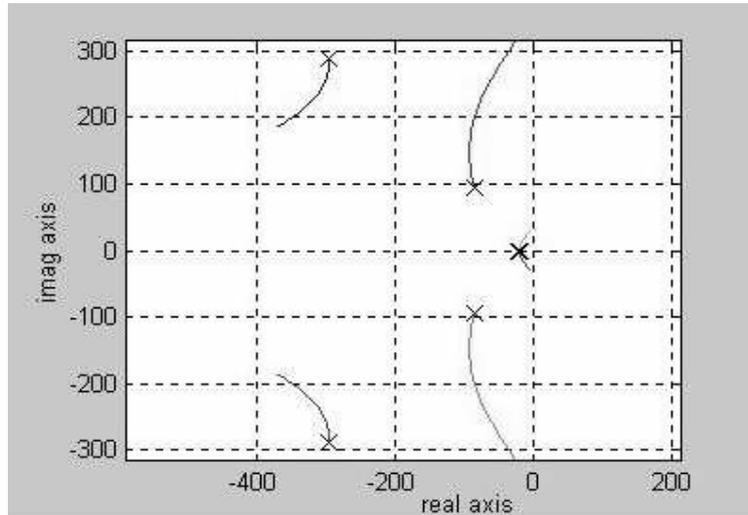


Fig.3.13-Root locus of  $(\Delta T_{em}/\Delta v_{qse})$  for varying gains

In addition to such a stability analysis, one may also investigate the step responses of the desired (output-input) transfer functions (see Fig.3.14-3.15).

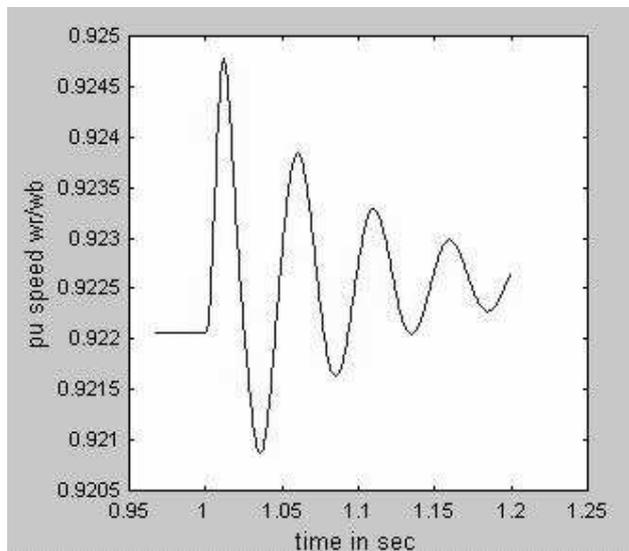


Fig.3.14-Step response of  $w_r$  (pu) to one volt change in  $v_{qse}$

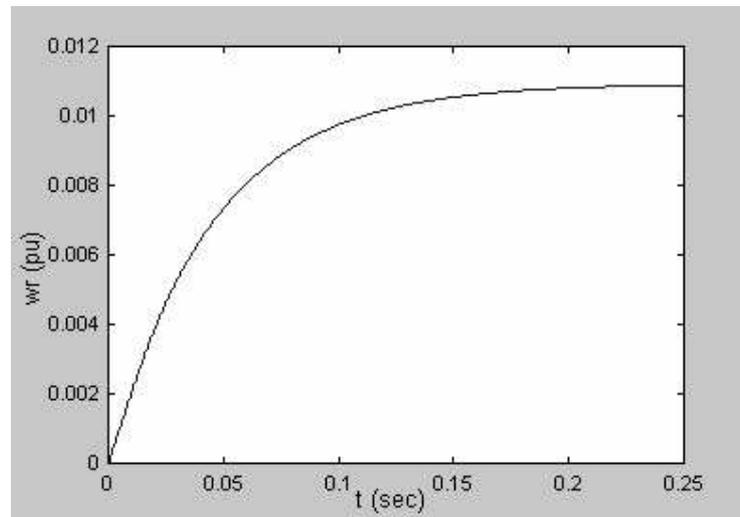


Fig.3.15-Step response of  $w_r$  (pu) to unit change in  $T_{load}$

Using the three different speed frames (arbitrary, stationary, synchronous) discussed in sections above oriented equivalent circuit models for air-gap flux, stator flux and rotor flux may also be derived. These models, however, are not considered here but left for future works.

### 3.7 Simulations of FOC developed in Stationary Reference Frame

State observers used in this thesis use stationary frame models for the sake of simplicity of the overall algorithm. Also in the field-oriented control simulations stationary axis dq model of a 20-hp induction motor is used. The simulation is implemented using MATLAB/Simulink. This simulation is implemented to be familiar with indirect field-oriented control and observe the variables at every stage of the control. Also one can observe how well the flux amplitude remains constant when the motor is loaded and the electromechanical torque is smooth. Related dq currents in the simulation may give significant clues about the field oriented control principle to a beginner.

In this simulation, reference dq currents are obtained according to the reference load torque and speed waveform. They are compared with the actual motor currents and the errors are input to PI controllers to obtain reference voltages. After this point simulations of Space Vector PWM and FOC are implemented separately due to very long simulation time of PWM part. Instead, the motor is fed with the first

harmonic of the PWM voltage to save time and simplify the simulation. The simulation results are given below:

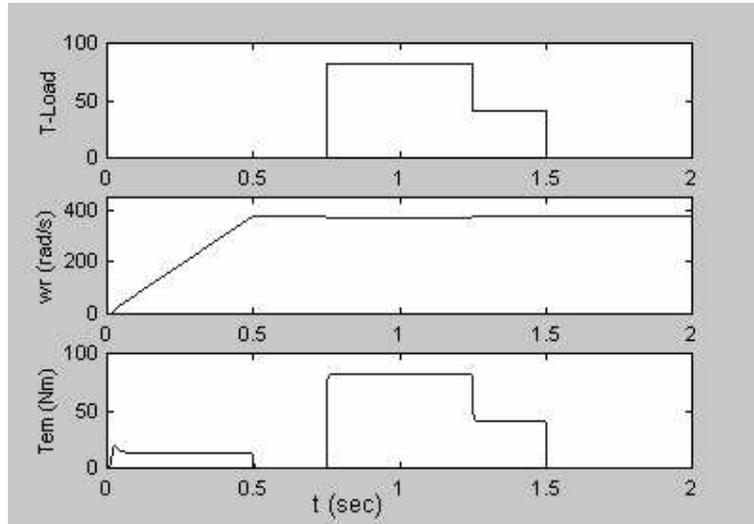


Fig 3.16- Applied mechanical torque, rotor speed and produced electromagnetic torque

Fig. 3.16 shows the load torque, the rotor speed and the produced electromechanical torque. In the first 0.5 sec., the motor produces electromechanical torque to overcome the effect of the inertia. In the no-load time interval,  $T_{em}$  is close to zero. As can be seen from F.g.3.16 very smooth torque is obtained with field-oriented control.

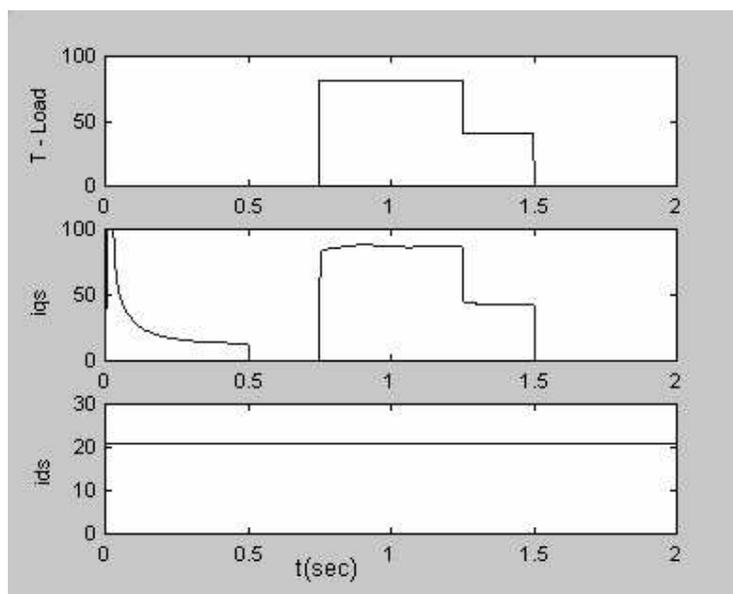


Fig.3.17- Synchronous frame dq axis currents

In Fig.3.17 synchronous frame currents; torque producing current component,  $i_{qs}$  and constant flux producing currents are shown. Note from Fig.3.17 that  $i_{qs}$  is proportional to the torque produced by the machine both during acceleration (see Fig 3.16) and at loaded regions. For constant flux operation, d-axis current,  $i_{ds}$  remains constant yielding a smooth flux in order to prevent torque oscillations. Fig.3.18 shows the stator phase voltage and the current. Note that the phase current increases proportionally to the load requirement. In Fig.3.19 dq stationary axis fluxes are shown. The magnitude of each flux component remains the same after the transient state. The rotor-flux is obtained as the square root of the sum of the squares of dq axis fluxes. Constant rotor-flux is vital for field-oriented control in controlling the torque perfectly. As in the case of DC motors, once constant flux is obtained, one can control the produced torque easily by controlling the torque producing current component,  $i_{qs}$  which is independent of the flux.

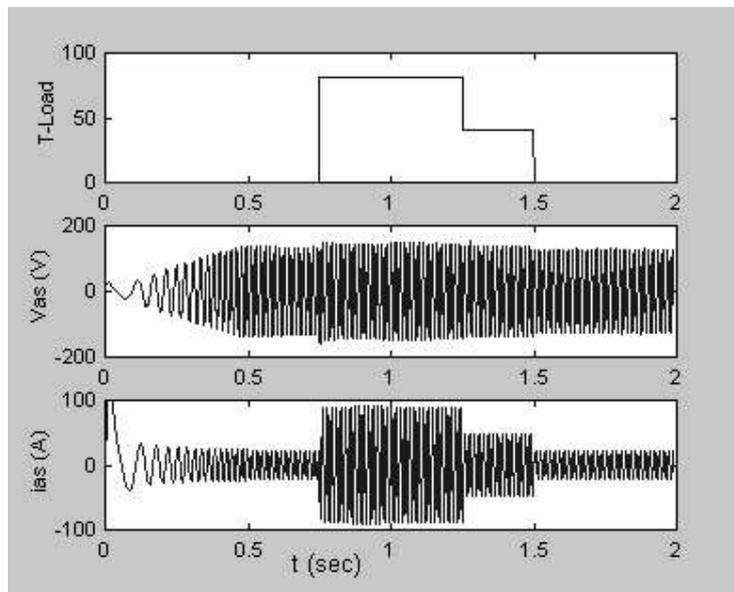


Fig. 3.18- Phase-A stator voltage and current

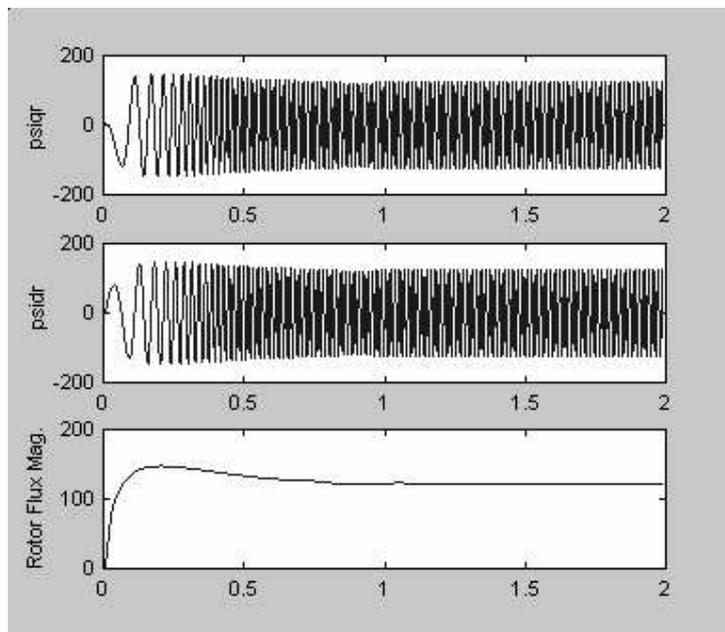


Fig.3.19 - dq axis rotor fluxes and rotor flux ( $T_{load}$  applied at 0.75 sec)

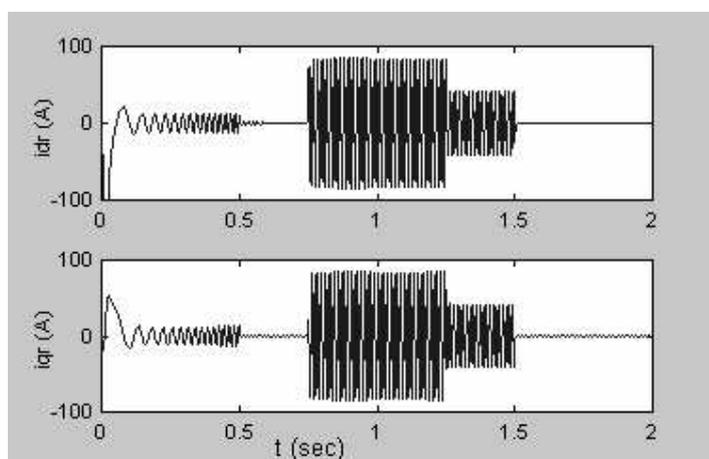


Fig.3.20- Referred rotor currents (dq-axis)

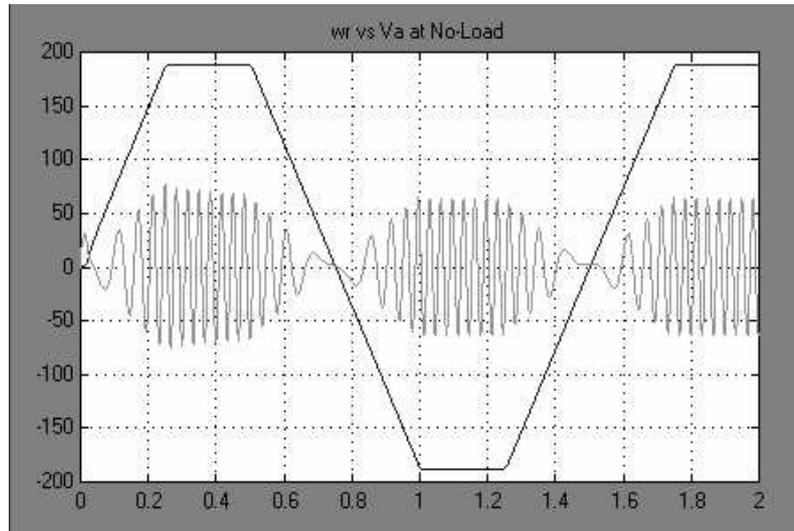


Fig.3.21- Four quadrant speed reversal and phase voltage

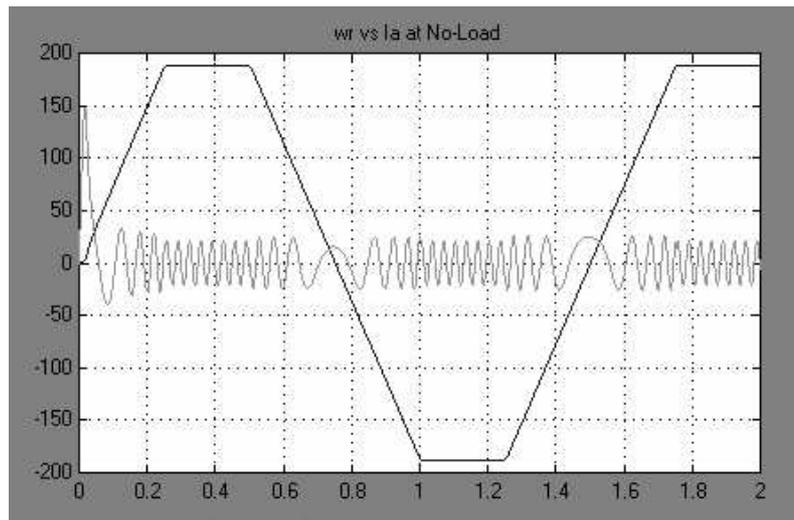


Fig.3.22- Four quadrant speed reversal and phase current

Fig.3.20 shows referred rotor currents. At no load case the rotor currents converge to zero due to unity slip. In Fig.3.21 and 3.22 four-quadrant speed reversal is given with phase voltage and current variations. The frequency and the magnitude of both the stator voltage and the stator current are controlled by FOC during the speed reversal operation.

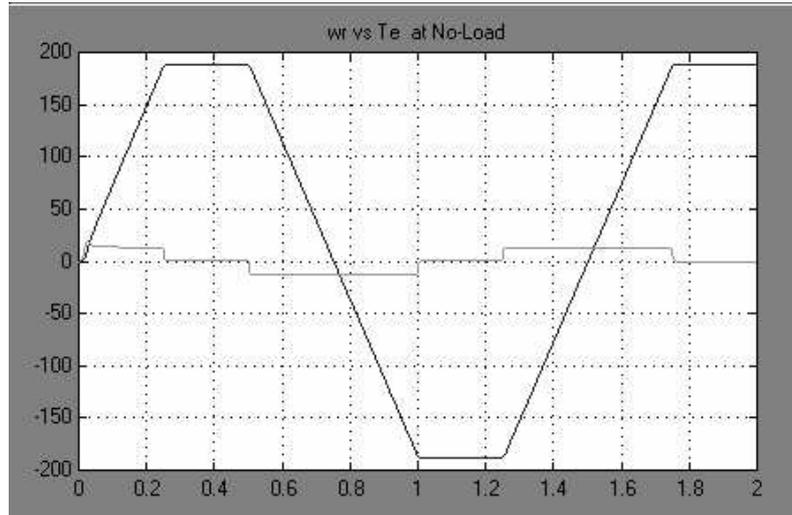


Fig.3.23- Four-quadrant speed reversal and produced torque due to inertia

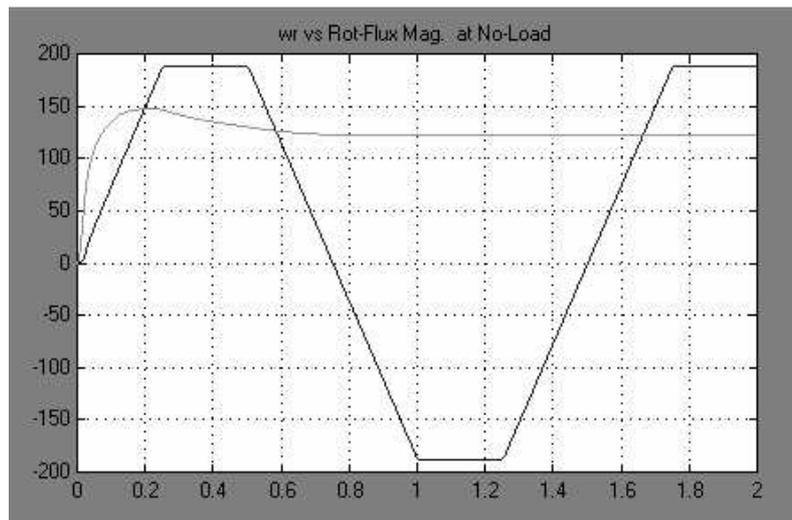


Fig.3.24- Four-quadrant speed reversal and rotor flux wave-form

In Fig.3.23 four-quadrant speed reversal waveform is given with produced torque. Since the operation is simulated at no-load, the torque produced due to the demand by the motor inertia is quite smooth because of the constant flux shown in Fig3.24. In Fig.3.24 we observe that the speed change does not affect the constant flux condition and this illustrates the satisfactory result of rotor field orientation.

## **CHAPTER 4**

### **PULSEWIDTH MODULATION with SPACE VECTOR THEORY**

#### **4.1 Inverters**

Three phase inverters, supplying voltages and currents of adjustable frequency and magnitude to the stator, are an important element of adjustable speed drive systems employing induction motors. Inverters with semiconductor power switches are d.c. to a.c. static power converters. Depending on the type of d.c. source supplying the inverter, they can be classified as voltage source inverters (VSI) or current source inverters (CSI). In practice, the d.c. source is usually a rectifier, typically of the three phase bridge configuration, with d.c. link connected between the rectifier and the inverter. The d.c. link is a simple inductive, capacitive, or inductive-capacitive low-pass filter. Since neither the voltage across a capacitor nor the current through an inductor can change instantaneously. A capacitive-output d.c. link is used for a VSI and an inductive-output link is employed in CSI.

VSI's can be either voltage or current controlled. In a voltage-controlled inverter, it is the frequency and magnitude of the fundamental of the output voltage that is adjusted. Feed-forward voltage control is employed, since the inverter voltage is dependent only on the supply voltage and the states of the inverter switches, and , therefore, accurately predictable. Current controlled VSI's require sensors of the output currents which provide the necessary control feedback.

The type of semiconductor power switch used in an inverter depends on the volt-ampere rating of the inverter, as well as on other operating and economic considerations, such as switching frequency or cost of the system. Taking into account the transient- and steady-state requirements, we have used 1200V, 40A IGBT switches. With appropriate heat sink, we can rise to 20 KHz, however at 10

KHz, switching losses and conduction losses become equal [40], moreover, complex mathematical algorithms require much time. Thus 10 KHz is selected as the switching frequency in our algorithms.

#### 4.1.2 Voltage Source Inverter (VSI)

A diagram of the power circuit of a three phase VSI is shown in the Fig.4.1. The circuit has bridge topology with three branches (phases), each consisting of two power switches and two freewheeling diodes. In the case illustrated and implemented in this thesis, the inverter is supplied from an uncontrolled, diode-based rectifier, via d.c. link which contains an LC filter in the inverted configuration. While this circuit represents a standard arrangement, it allows only positive power flow from the supply system to the load via typically three-phase power line. Negative power flow, which occurs when the load feeds the recovered power back to the supply, is not possible since the resulting negative d.c. component of the current in the d.c. link can not pass through the rectifier diodes. Therefore, in drive systems where the VSI-fed motor may not operate as a generator, more complex supply system must be used. These involve either a braking resistance connected across the d.c. link or replacement of the uncontrolled rectifier by a dual converter. As a future work, the inverter may be supported with braking resistance connected across the d.c. link via free wheeling diode and a transistor. When the power is returned by the motor, it is dissipated in the braking resistor which is called dynamic braking. The circuit diagram of three-phase VSI used in this project is shown in Fig.4.1.

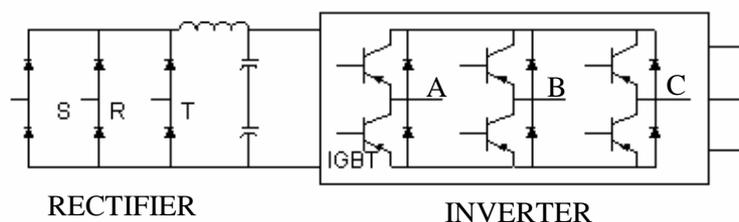


Fig.4.1- Circuit diagram of three phase VSI

Because of the constraints that the input lines must never be shorted and the output current must always be continuous, a voltage source inverter can assume only

eight distinct operational topologies. They are shown in Fig.4.2 and Fig.4.3. Six out of these eight topologies produce a non-zero output voltage and are known as non-zero switching states and the remaining two topologies produce zero output and are known as zero switching state.

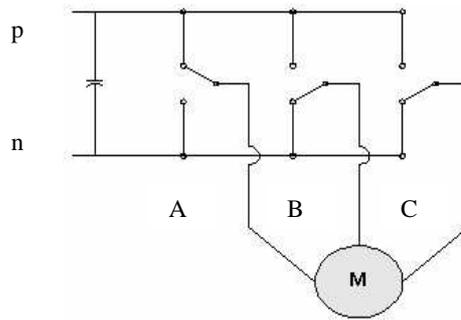


Fig.4.2- Three phase inverter with switching states

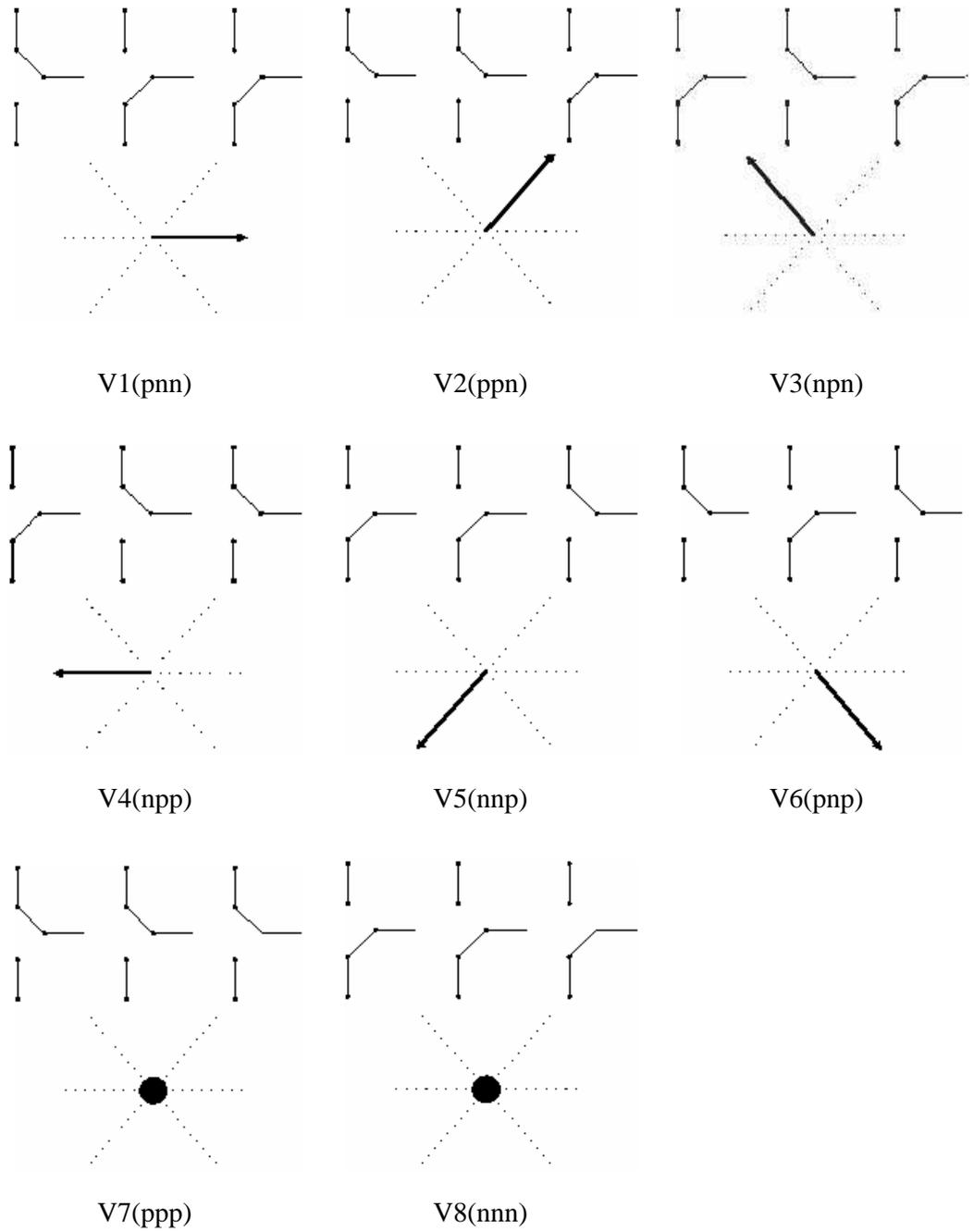


Fig.4.3- Eight switching state topologies of a voltage source inverter

## 4.2 Voltage Space Vectors

Space vector modulation for three leg VSI is based on the representation of the three phase quantities as vectors in two-dimensional ( $\alpha$ - $\beta$ ) plane. Considering the first switching state in Fig.4.4, line-to-line voltages are given by

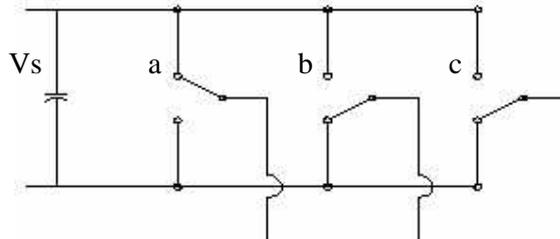


Fig.4.4- First switching state  $-V_1$  (pnn)

$$V_{ab} = V_s$$

$$V_{bc} = 0$$

$$V_{ca} = -V_s$$

This can be represented in ( $\alpha$ - $\beta$ ) plane as shown in Fig.4.5 where  $V_{ab}$ ,  $V_{bc}$  and  $V_{ca}$  are the three line voltage vectors displaced by  $120^\circ$  in space. The effective voltage vector generated by this topology is represented as  $V_1$  (pnn) in Fig.4.5. Here (pnn) refers to the three leg /phases a,b,c being either connected to the positive dc rail (p) or to the negative dc rail (n). For the first switching state  $V_1$ , phase a is connected to positive dc rail and phases b and c are connected to negative dc rail.

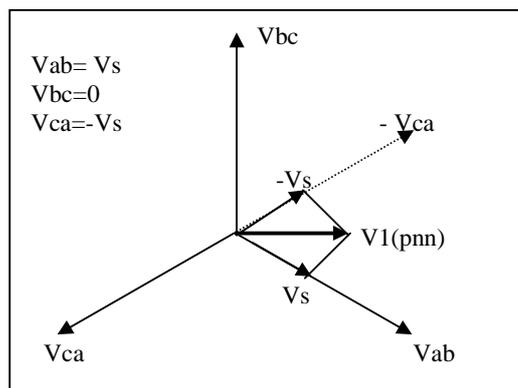


Fig.4.5- Representation of topology 1 in ( $\alpha$ - $\beta$ ) plane

Similar to the  $V_1$ , six non-zero voltage vectors can be shown as in Fig.4.6. The tips of these vectors form a regular hexagon. We define the area enclosed by two adjacent vectors, within the hexagon, as a sector.

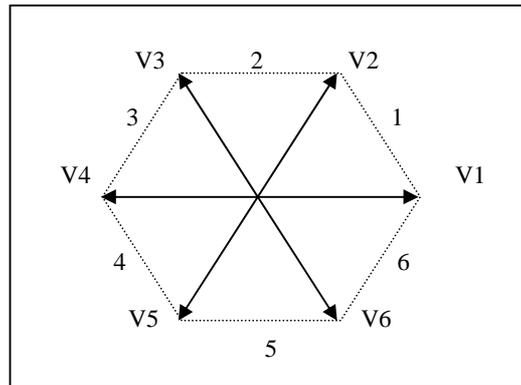


Fig.4.6- Non-zero voltage vectors in  $(\alpha-\beta)$  plane

The last two topologies of Fig.4.3 are zero state vectors. The output line voltages in these topologies are zero.

$$V_{ab}=0$$

$$V_{bc}=0$$

$$V_{ca}=0$$

These are represented as vectors which have zero magnitude and hence are referred as zero switching state vectors. They are represented with dot at the origin instead of vectors as shown in Fig.4.7.

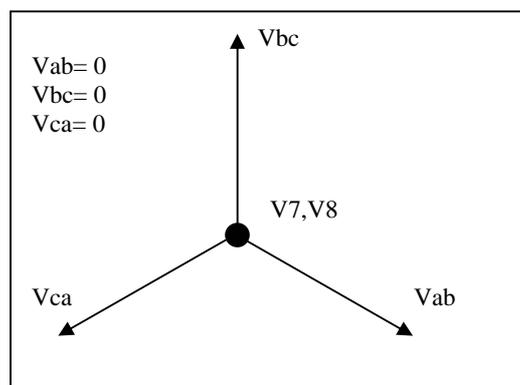


Fig.4.7- Representation of the zero voltage vectors in  $(\alpha-\beta)$  plane

### 4.3 Space Vector Modulation

In the literature there exist a number of PWM algorithms [41, 42]. The performance criteria of these algorithms are basically:

- 1-Current harmonics
- 2-Harmonic spectrum
- 3-Torque harmonics
- 4-Switching frequency
- 5-Dynamic performance
- 6-Polarity consistency rule

The well-known feed-forward PWM schemes are:

- a-Carrier based PWM
- b-Carrierless PWM
- c-Over-modulation
- d-Optimized feedforward PWM

In this thesis, we have implemented one of the well-known carrier based PWM technique, SVM and proved its high performance with respect to other techniques (e.g. Sinusoidal Modulation) [43]. Now let us look at the basics of SVM.

The desired three phase voltages at the output of the inverter could be represented by an equivalent vector  $\mathbf{V}$  rotating in the counter clockwise direction as shown in Fig.4.8. The magnitude of this vector is related to the instantaneous magnitude of the output voltage (see Fig.4.9.) and the period this vector takes to complete one revolution is the same as the fundamental time period of the output voltage.

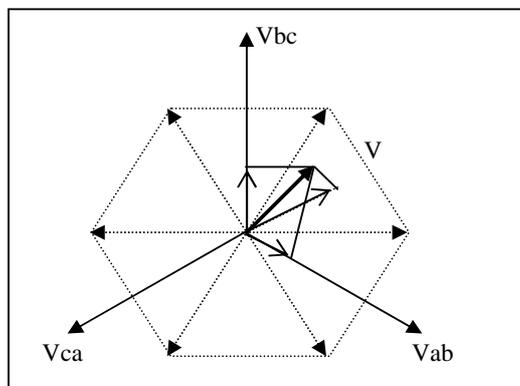


Fig.4.8- Output voltage vector ( $\mathbf{V}$ ) in ( $\alpha$ - $\beta$ ) plane

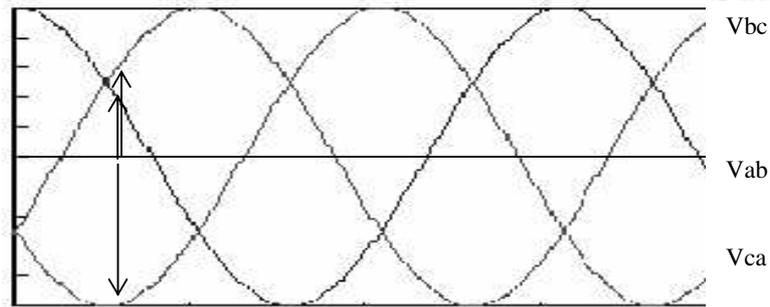


Fig.4.9- Output line voltages in the time domain

Let us consider the situation when the desired line-to-line output voltage vector  $\mathbf{V}$  is in sector 1 as shown in Fig.4.10. This vector could be synthesized by the pulse width modulation (PWM) of the adjacent SSV's  $\mathbf{V}_1$  (pnn) and  $\mathbf{V}_2$  (ppn), the duty cycle of each being  $d_1$  and  $d_2$ , respectively, and the zero vector ( $\mathbf{V}_7$  (nnn) /  $\mathbf{V}_8$ (ppp)) of duty cycle  $d_0$  :

$$d_1 \mathbf{V}_1 + d_2 \mathbf{V}_2 = \mathbf{V} = m \cdot V_s \cdot e^{i\theta} \quad (4.1)$$

$$d_1 + d_2 + d_0 = T \quad (4.2)$$

where,  $0 < m < 0.866$ , is the modulation index.

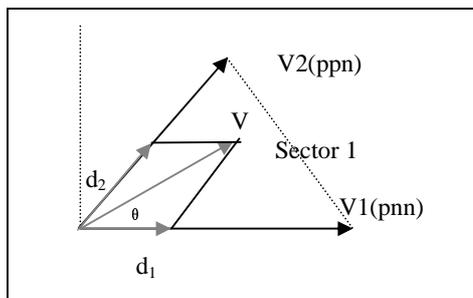


Fig.4.10- Synthesis of the required output voltage vector in sector 1

While determining the duty cycles  $d_1$ ,  $d_2$  and  $d_0$  in SVM techniques, the only difference is the choice of zero vectors and the sequence in which the vectors are applied within the switching cycles. One is free about selecting the given alternatives below in his SVM algorithm:

- 1-Choice of the zero vector – using  $V_7$  (ppp),  $V_8$  (nnn) or both,
- 2-Sequencing of the vectors
- 3-Splitting of the duty cycles of the vectors without introducing additional commutations.

Here four different SVM schemes are given roughly according to their repeating duty-cycle distribution:

- a-The right aligned sequence ( $d_0/2, d_1, d_2, d_0/2$ )
- b-Symmetric sequence ( $d_0/4, d_1/2, d_2/2, d_0/2, d_2/2, d_1/2, d_0/4$ )
- c-Alternating Zero Vector Sequence ( $d_1, d_2, d_0, d_2, d_1, d_0$ )
- d-Highest Current Not-Switched Sequence ( $d_1, d_2, d_0$ )

Among these SVM techniques the commonly preferred symmetric sequence, which has the lowest THD, has been implemented in this study [44,45]. Switching sequence is given in Fig.4.11.

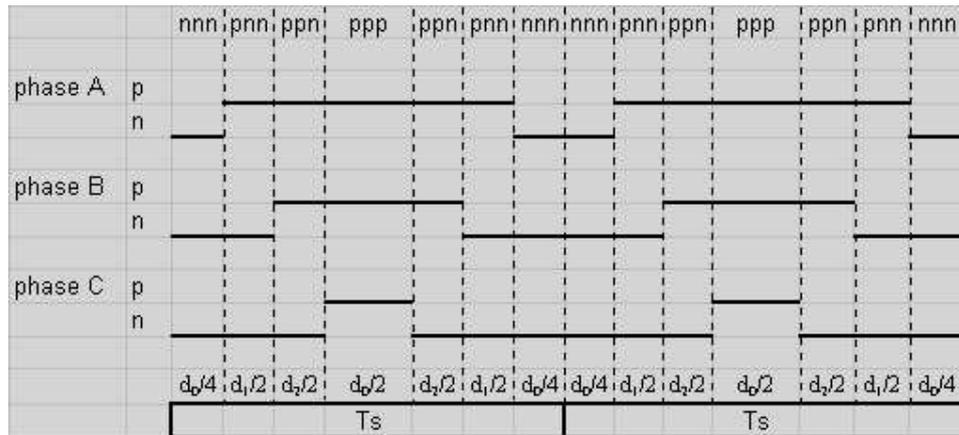


Fig.4.11- Phase gating signals in Sym. Seq. SVM

#### 4.4 SVPWM Application to the Static Power Bridge and Implementation Using DSP Platform

In the case of AC drive applications, sinusoidal voltage sources are not used as explained before. Instead, they are replaced by 6 power IGBTs that act as on/off switches to the rectified DC bus voltage. The aim is to create sinusoidal current in the coils to generate rotating field. Owing to the inductive nature of the phases, a pseudo-sinusoidal current is created by modulating the duty-cycle of the power switches. The switches (IGBT) shown in the Fig.4.12 are activated by signals (a, b,

c) and their complement values. Eight different combinations are available with this three-phase VSI including two zero states.

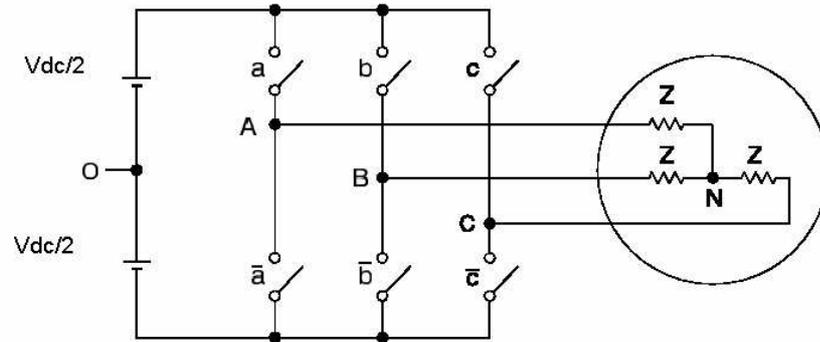


Fig.4.12- Power Bridge

It is possible to express each phase-to-neutral voltage for every switching combination of IGBTs as listed in Table 4.1.

Table 4.1. Power Bridge Output Voltages ( $V_{AN}$ ,  $V_{BN}$ ,  $V_{CN}$ )

A	B	C	$V_{AN}$	$V_{BN}$	$V_{CN}$
0	0	0	0	0	0
0	0	1	$-V_{dc}/3$	$-V_{dc}/3$	$2V_{dc}/3$
0	1	0	$-V_{dc}/3$	$2V_{dc}/3$	$-V_{dc}/3$
0	1	1	$-2V_{dc}/3$	$V_{dc}/3$	$V_{dc}/3$
1	0	0	$2V_{dc}/3$	$-V_{dc}/3$	$-V_{dc}/3$
1	0	1	$V_{dc}/3$	$2V_{dc}/3$	$V_{dc}/3$
1	1	0	$V_{dc}/3$	$V_{dc}/3$	$2V_{dc}/3$
1	1	1	0	0	0

In field-oriented control algorithm, the control variables are expressed in rotating frame. The current vector  $I_{sref}$  that directly controls the torque is transformed into a voltage vector by the inverse Park transform. This voltage reference is expressed in the ( $\alpha$ - $\beta$ ) frame. Using this transformation three-phase voltages ( $V_{AN}$ ,  $V_{BN}$ ,  $V_{CN}$ ) and the reference voltage vector are projected in the ( $\alpha$ - $\beta$ ) frame. The expression of the three phase voltages in the ( $\alpha$ - $\beta$ ) frame are given by general Clarke transformation equation:

$$\begin{bmatrix} V_{s\alpha} \\ V_{s\beta} \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} V_{AN} \\ V_{BN} \\ V_{CN} \end{bmatrix} \quad (4.3)$$

Since only 8 combinations are possible for the power switches,  $V_{s\alpha}, V_{s\beta}$  can also take finite number of values in the ( $\alpha$ - $\beta$ ) frame (Table 4.2) according to the IGBT command signals (a, b, c).

Table 4.2. Stator Voltages in ( $\alpha$ - $\beta$ ) frame and related Voltage Vector

A	B	C	$V_\alpha$	$V_\beta$	Vectors
0	0	0	0	0	V0
0	0	1	$-V_{dc}/3$	$-V_{dc}/\sqrt{3}$	V1
0	1	0	$-V_{dc}/3$	$V_{dc}/\sqrt{3}$	V2
0	1	1	$-2V_{dc}/3$	0	V3
1	0	0	$2V_{dc}/3$	0	V4
1	0	1	$V_{dc}/3$	$-V_{dc}/\sqrt{3}$	V5
1	1	0	$V_{dc}/3$	$V_{dc}/\sqrt{3}$	V6
1	1	1	0	0	V7

The eight voltage vectors re-defined by the combination of the switches are represented in Fig.4.13. Now, given a reference voltage (coming from the inverse

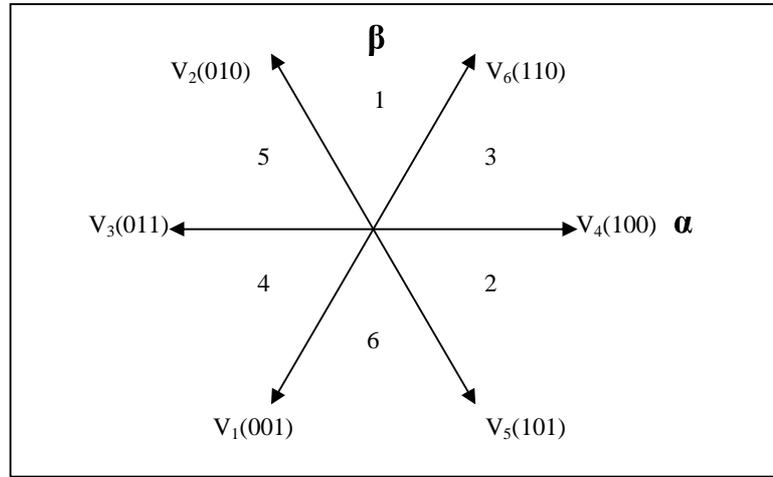


Fig.4.13 -Voltage Vectors

Park transform), the following step is used to approximate this reference voltage by the above defined eight vectors. The method used in approximating the desired stator reference voltage with only eight possible states of switches is to combine adjacent vectors of the reference voltage and modulate the time of application of each adjacent vector. In Fig.4.14, the reference voltage  $V_{sref}$  is in the third sector and the application time of each adjacent vector is given by:

$$T = T_4 + T_6 + T_0$$

$$V_{sref} = \frac{T_4}{T} \vec{V}_4 + \frac{T_6}{T} \vec{V}_6 \quad (4.4)$$

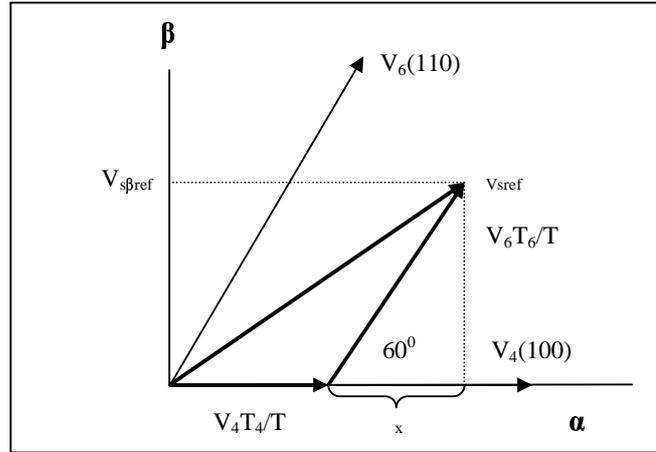


Fig. 4.14 - Projection of the reference Voltage Vector

The determination of the amount of times  $T_4$  and  $T_6$  is given by simple projections:

$$\begin{aligned}
 V_{s\beta\text{ref}} &= \frac{T_6}{T} \|\vec{V}_6\| \cos(30^\circ) \\
 V_{s\alpha\text{ref}} &= \frac{T_4}{T} \|\vec{V}_4\| + x \\
 x &= \frac{V_{s\beta\text{ref}}}{\text{tg}(60^\circ)}
 \end{aligned} \tag{4.5}$$

Finally, with the  $(\alpha\text{-}\beta)$  component values of the vectors given in the Table 4.2, the amount of times of application of each adjacent vector is:

$$\begin{aligned}
 T_4 &= \frac{T}{2V_{dc}} (3V_{s\alpha\text{ref}} - \sqrt{3}V_{s\beta\text{ref}}) \\
 T_6 &= \sqrt{3} \frac{T}{V_{dc}} V_{s\beta\text{ref}}
 \end{aligned} \tag{4.6}$$

The rest of the period spent in applying the null-vector. For every sector, commutation duration is calculated. The amount of times of vector application can all be related to the following variables:

$$\begin{aligned}
 X &= \sqrt{3}V_{s\beta\text{ref}} \\
 Y &= \frac{\sqrt{3}}{2}V_{s\beta\text{ref}} + \frac{3}{2}V_{s\alpha\text{ref}} \\
 Z &= \frac{\sqrt{3}}{2}V_{s\beta\text{ref}} - \frac{3}{2}V_{s\alpha\text{ref}}
 \end{aligned} \tag{4.7}$$

In the previous example for sector 3,  $T_4 = -Z$  and  $T_6 = X$ . Extending this logic, one can easily calculate the sector number belonging to the related reference voltage vector. The following basic algorithm helps to determine the sector systematically.

*If  $X > 0$  then  $A=1$  else  $A=0$*   
*If  $Y > 0$  then  $B=1$  else  $B=0$*   
*If  $Z > 0$  then  $C=1$  else  $C=0$*   
*Sector =  $A+2B+4C$*

Application durations of the sector boundary vectors are tabulated as;

Sector

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

*Saturations*

*If  $(t_1 + t_2) > PWMPRD$  then*  
 *$t_{1sat} = (t_1 / (t_1 + t_2)) * PWMPRD$*   
 *$t_{2sat} = (t_2 / (t_1 + t_2)) * PWMPRD$*

The third step is to compute the three necessary duty cycles as;

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

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

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

The last step is to assign the right duty cycle to the right motor phase (in other words, to the right CMPRx) according to the sector(see Fig.4.16). Table 4.3 depicts this determination.

Table 4.3- Assigned duty cycles to the PWM outputs

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

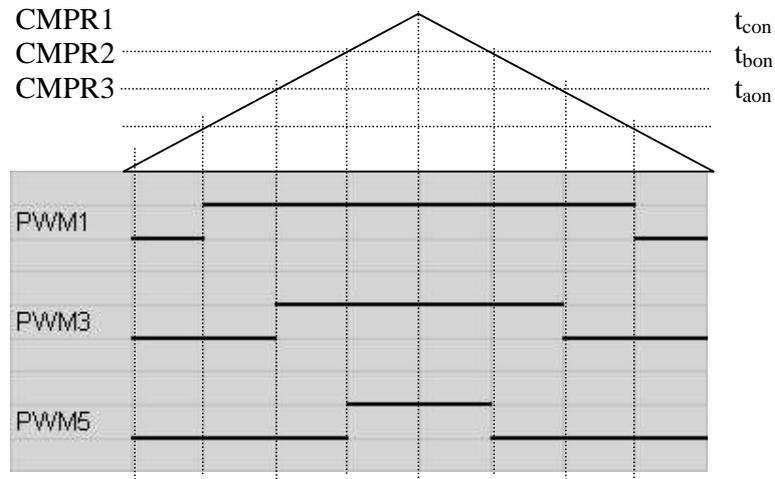


Fig.4.15- Sector 3 PWM Patterns and Duty Cycles (case shown in Fig.4.14)

#### 4.5 Event Manager Configuration of DSP for SVPWM

TMS320F/C2XX has specific peripherals in order to handle space vector module easily and optimally. Timer registers, compare registers, PWM outputs and PWM interrupts and programmable timer-ADC synchronization helps us for easy implementation. However, although these tools make design easy they are not user friendly for a beginner and rather complicated.

Timer is the base time of the PWM interrupts generation and overall control algorithm is synchronized with timer1, PWM underflow interrupt. During the excessively long time specifications for timer underflow the algorithm is run in an infinite loop. For the next period again timer1 underflow interrupt is extracted from this infinite loop. By this way PWM outputs and the overall control algorithm run synchronously. As an alternative, one may write an interface program for visualizing the software states without changing program instead of an infinite loop. Timer1 is configured in up-down counting mode to generate the symmetrical PWM patterns. The timer 1 control register T1CON is programmed in order to get a 50ns resolution: the pre-scalar clock of the timer is set to 1 giving the highest possible resolution. Two consecutive writes to T1CON are required to ensure the synchronization of the GP timers when T1CON [6] is used to enable GP timer 2 or 3:

- 1) Configure all other bits with T1CON [6] set to 0.
- 2) Enable GP timer 1 and, thus, GP timer 2 or GP timers 2 and 3, by setting T1CON[6] to 1. Otherwise the PWM outputs cannot be observed.

Configuration of T1CON is given below:

```
splk #PWMPRD,T1PER      ;Set PWM interrupt period
splk #0,T1CNT           ;Ignore Emulation suspend
splk #0A800h,T1CON      ;Up/Down count mode(must)
                        ;x/1 prescalar(optional)
                        ;Use own TENABLE
                        ;Disable Timer(for the first loading)
                        ;Internal Clock Source(must)
                        ;Reload Compare Register when T1CNT=0
                        (must)
                        ;Disable Timer Compare operation
```

The compare registers must be continuously reloaded with calculated duty cycle values ( $t_{aon}$ ,  $t_{bon}$ ,  $t_{con}$ ).

```
splk #0207h,COMCON      ;Disable for the first loading
                        ;Reload Full Compare when T1CNT=0(must)
                        ;Disable Space Vector
                        ;Reload Full Compare Action when T1CNT=0
                        ;Enable Full Compare Outputs
                        (must)
                        ;Disable Simple Compare Outputs(SC not
                        used)
                        ;Select GP timer1 as time base(must)
                        ;Full Compare Units in PWM Mode(must)
splk #8207h,COMCON      ;enable compare operation
```

Two consecutive writes to COMCON are required to ensure the proper operation of full compare units in the PWM mode:

- 1) Enable PWM mode without enabling compare operation.
- 2) Enable compare operation by setting COMCON[15] to 1 without changing any other bits.

The output of the Compare operation are not directly sent to the Output Logic but are previously transferred through the PWM Deadband on-chip circuit. Depending on the power bridge pre-driver used, the control register DBTCON has to be programmed. The dead-band unit is designed to assure that no overlap occurs between the turn-on periods of the upper and lower devices that are controlled by the two compare/PWM outputs associated with each full compare unit (see Fig.4.16).

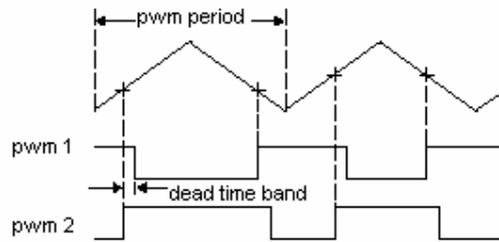


Fig.4.16- Dead time band

This assures that no overlap will occur under any condition. Although IGBT gate driver card provides sufficient dead time, we also added software dead time for reliability. However, at very low speed range effect of dead time bands at the output voltage becomes remarkable and must be compensated.

Bits in the full compare action control register (ACTR) control the action that takes place on each of the six compare output pins (PWMx/CMPx, x = 1–6) on a compare event. The polarity of the PWM pins is chosen in the Full Compare Action Control Register (ACTR) as follow:

```

ldp #DP_EV
splk #0666h,ACTR ;Bits 15-12 not used, no space vector
;PWM compare actions
;PWM5/PWM6 - Active Low/Active High
;PWM3/PWM4 - Active Low/Active High
;PWM1/PWM2 - Active Low/Active High

```

#### 4.6 Simulation and Experimental Results of SVPWM

The SVPWM algorithm implemented here by DSP is simulated before experimental works to verify its results. In the first simulation (Fig.4.17), SVPWM

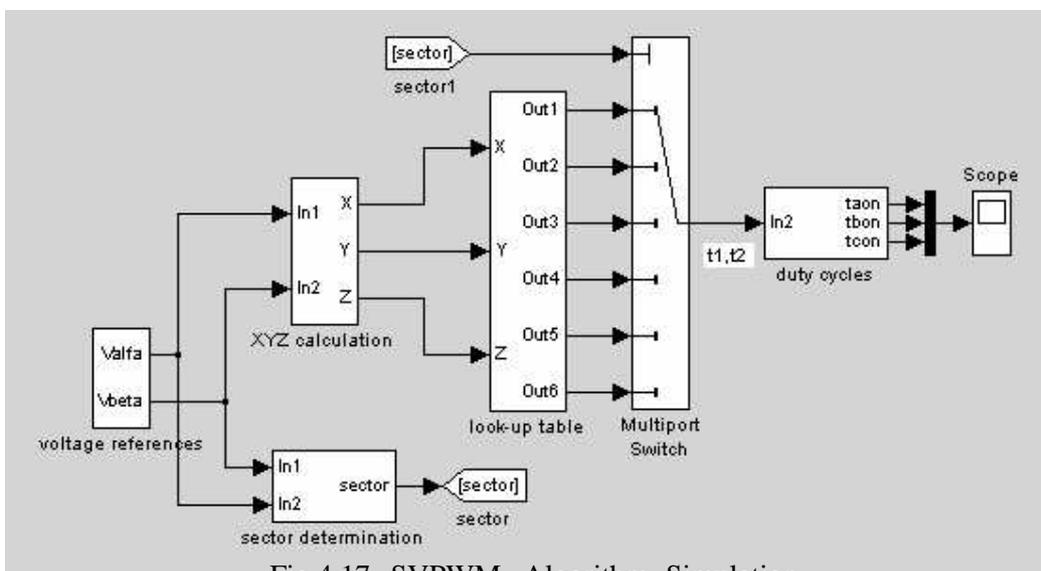


Fig.4.17 - SVPWM Algorithm Simulation

algorithm is simulated step-by-step and all the software variables in the algorithm are compared with the experimental DSP program outputs. It is shown that both of the results are the same, and correct.

In Fig.4.18. duty cycles of two PWM switches are shown ( $t_{aon}$ ,  $t_{bon}$ ,  $t_{con}$ ). In Fig.4.19 sector numbers of the rotating reference voltage vector is given. A careful reader will notice that the order of the sectors is the same as in Fig.4.13 of a vector

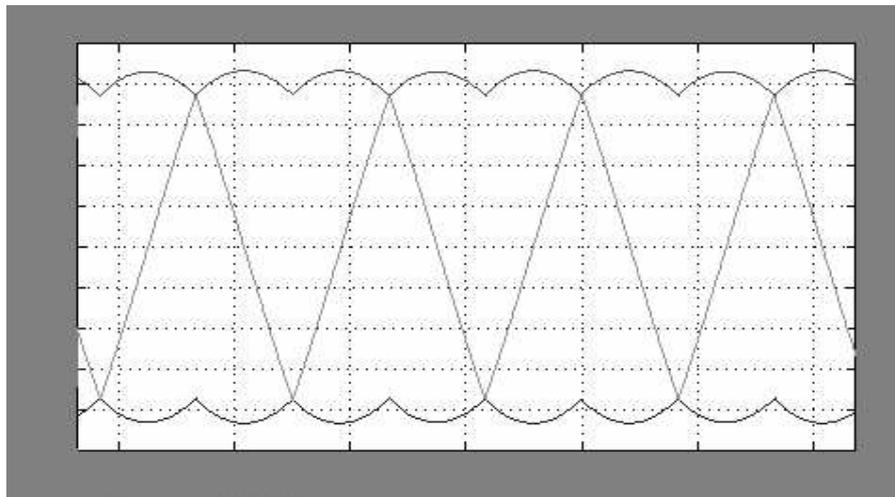


Fig.4.18- Simulated waveforms of duty cycles, ( $t_{aon}$ ,  $t_{bon}$ ,  $t_{con}$ )

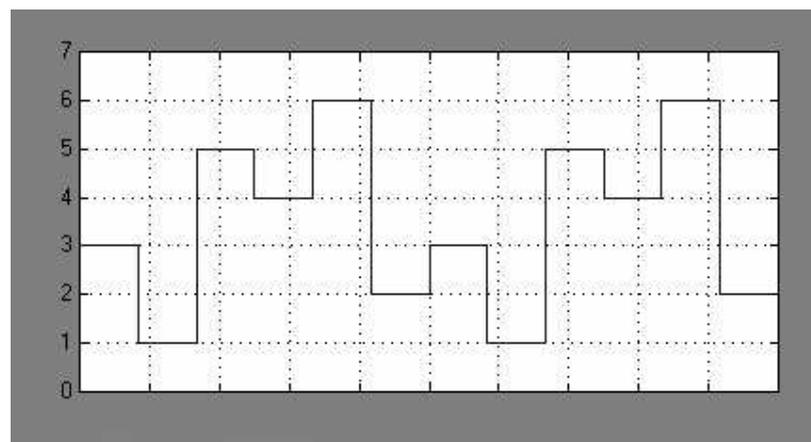


Fig.4.19- Sector numbers of voltage vector

rotating in the direction of counterclockwise. In Fig.4.20 durations of the to boundary sector vectors are shown. In Fig.4.21 projection vectors of the reference voltage vector on (a b c) plane are shown in time domain (see Fig.4.8).

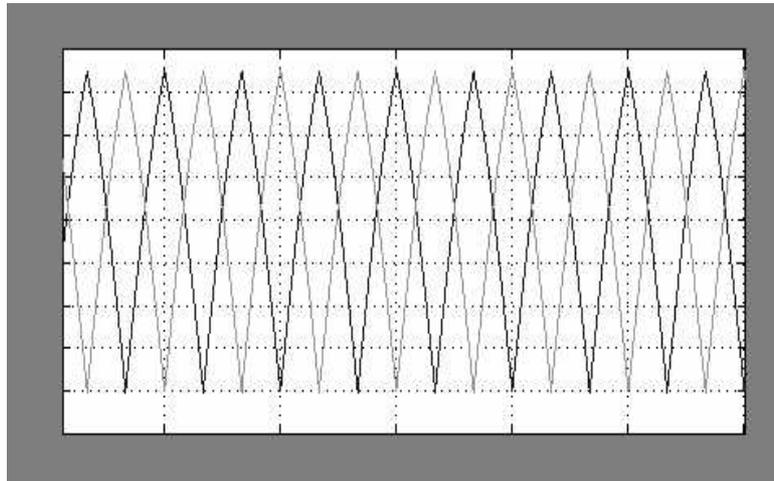


Fig.4.20- Duration of two sector boundary vectors ( $t_1, t_2$ )

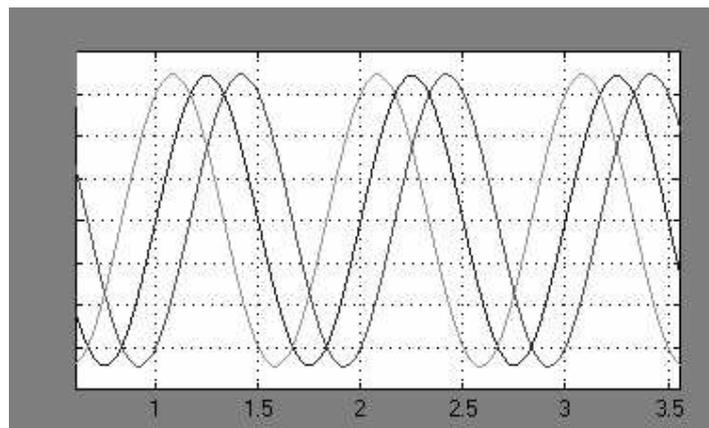


Fig.4.21- The projections of the  $V_a$ ,  $V_b$  and  $V_c$  of the reference voltage vector in the (a b c) plane  $-(X, Y, Z)$

In the second simulation, a straightforward SVPWM algorithm is implemented ignoring optimal conditions for practical applications. In this simulation one can observe line-to-line voltages in the form of frequent pulses and the sampled signal (reference voltage) for varying modulation constants (see Fig.4.22-2.24)

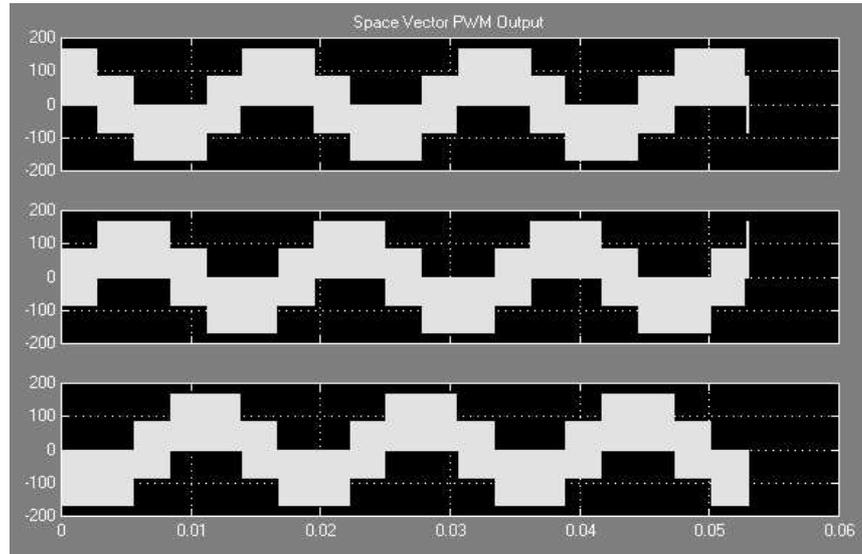


Fig.4.22- A typical line to line voltage output of SVPWM

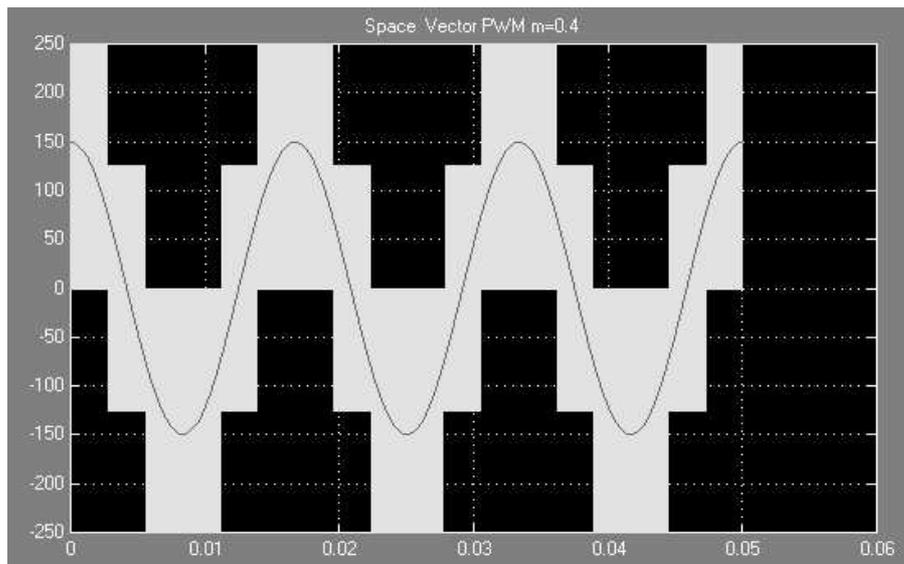


Fig.4.23- SVPWM output with the signal sampled ( $m=0.4$ )

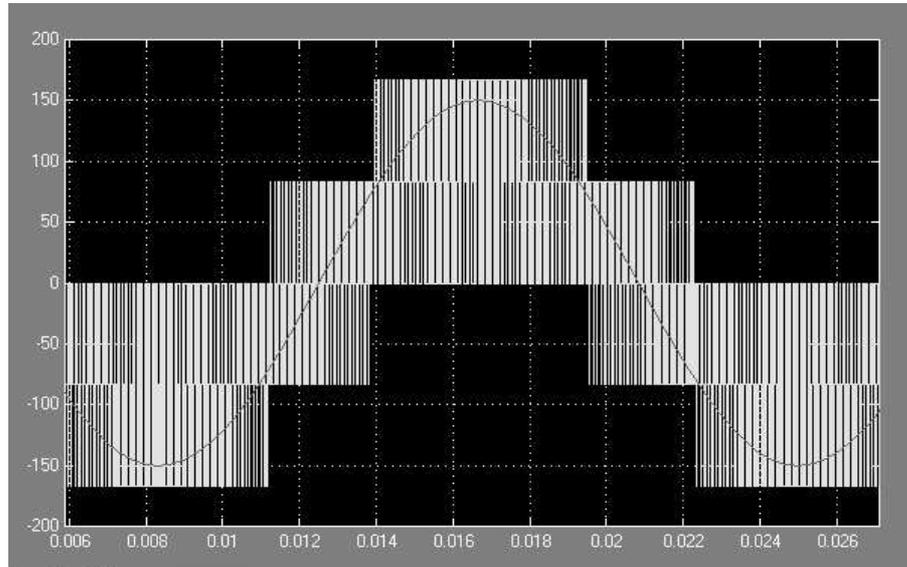


Fig.4.24- SVPWM output with the signal sampled ( $m=0.6$ )-Zoomed

The experimental outputs confirm the theoretical and simulation outputs. Given two reference voltage vectors associated with the reference currents and torque requirement SVPWM software parameters are observed and compared with the simulated ones. Fig.4.25 shows duty cycle of one of the PWM switches. The duty cycles is figured out by DAC outputs of the DSP processor.

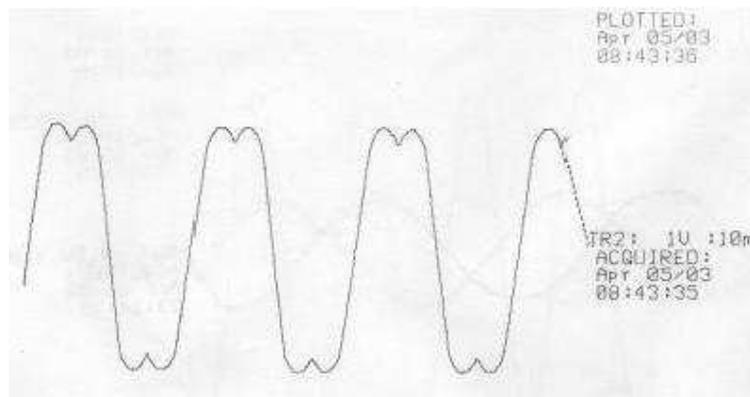


Fig.4.25- Duty cycle of PWM1

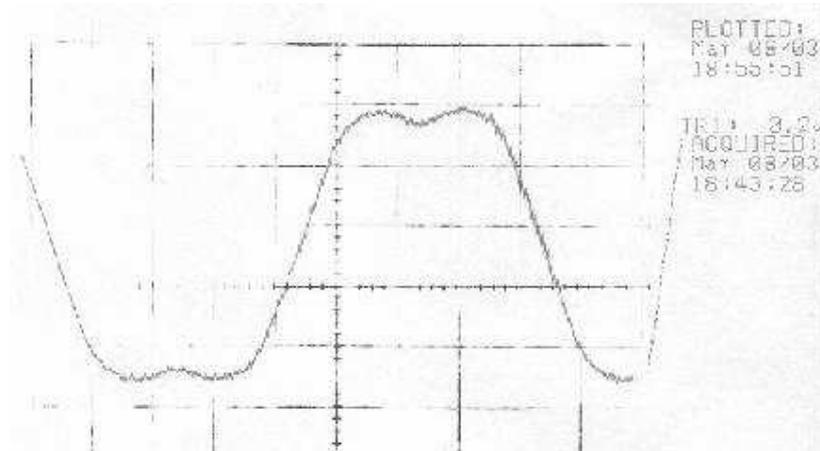


Fig.4.26- Low-pass filtered form of PWM1 pulses

A SVPWM designer must check the correctness of the six PWM outputs generated by this SVPWM module. A simple low-pass filter RC circuit may be used to filter out the high frequency components. The R and C values (or the time constant) are chosen for a desired cut-off frequency ( $f_c$ ) using the following equation:

$$\text{Time constant} = RC = 1/2\pi f_c$$

For example,  $R = 1.8 \text{ k.}$  and  $C = 100 \text{ nF,}$  gives  $f_c = 884.2 \text{ Hz.}$  This cut-off frequency has to be lower than the PWM frequency. This low-pass filter is connected to the PWM pins of the x24x/x240xEVM, the filtered version of the PWM signals are monitored by oscilloscope. The waveform shown on the oscilloscope should be the same as the one shown in Fig.4.25. In Fig.4.27 the sector number of the rotating reference voltage vector is shown (see Fig.4.19 simulation output).

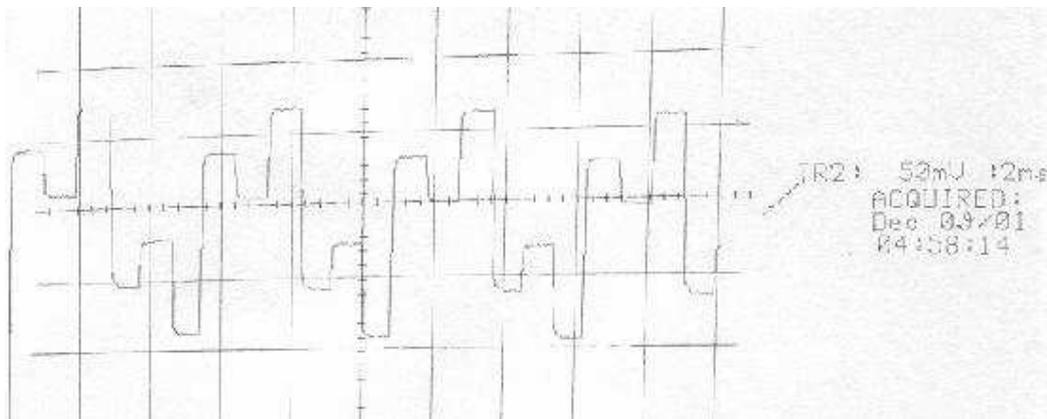


Fig.4.27-Sector number of the reference voltage

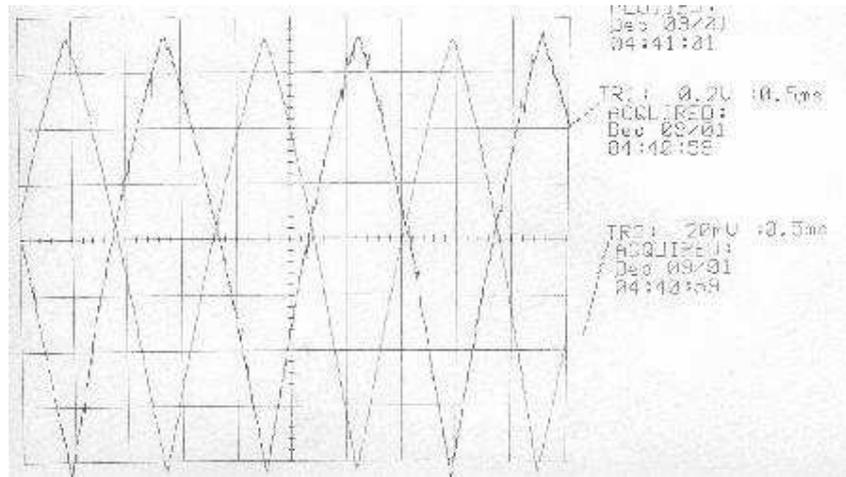


Fig.4.28- Duration of two boundary vectors ( $t_1, t_2$ )

Fig.4.28 is the experimental confirmation of simulation shown in Fig.4.20, duration of two boundary vector. Fig.4.29 is the experimental result of projection vectors in abc plain (X,Y in time domain- see Fig.4.21)

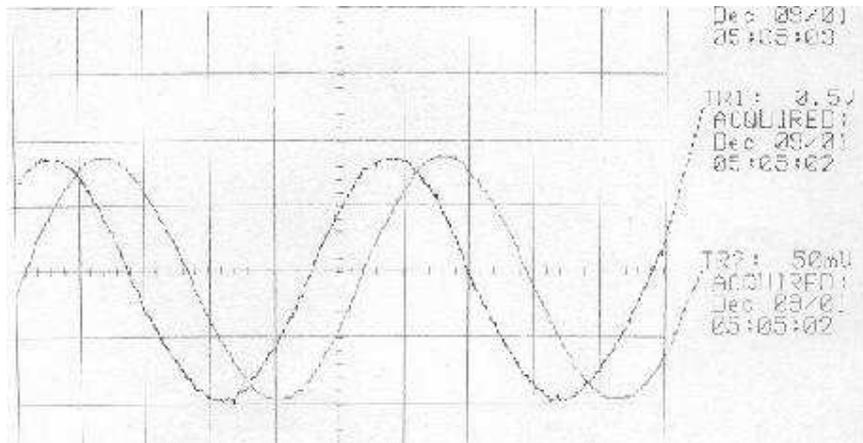


Fig.4.29- Projection vectors in abc plain (X,Y in time domain)

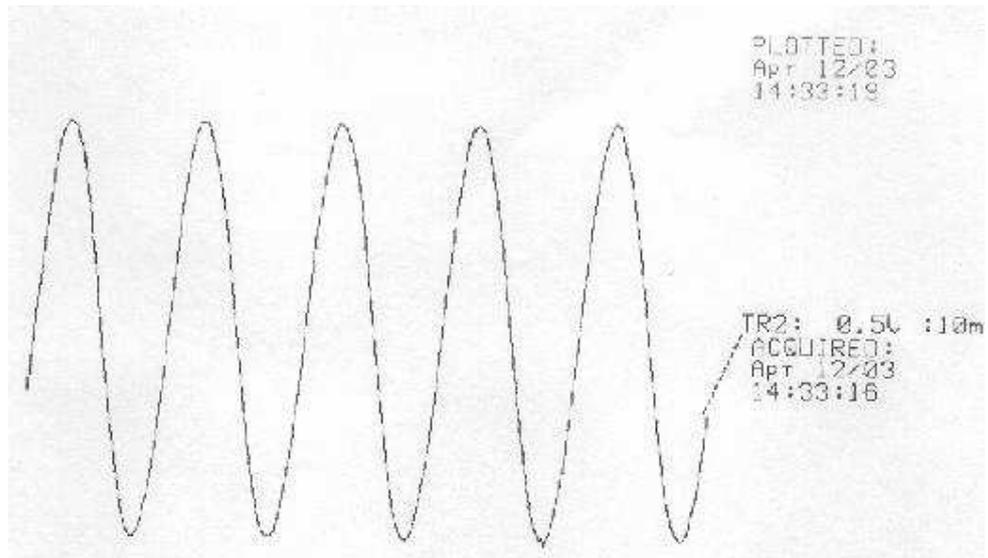


Fig.4.30- Typical phase current of an induction motor driven by SVPWM under heavy load conditions.

## CHAPTER 5

### KALMAN FILTER

#### 5.1 Sensorless Control

In controlling AC machine drives speed transducers such as tachogenerators, resolvers, or digital encoders are used to obtain speed information. Using these speed sensors has some disadvantages

- They are usually expensive,
- The speed sensor and the corresponding wires will take up space,
- In defective and aggressive environments, the speed sensor might be the weakest part of the system

Especially the last item degrades the system's reliability and reduces the advantage of an induction motor drive system. This has led to a great many speed sensorless vector control methods [46]. On the other hand, avoiding sensor means use of additional algorithms and added computational complexity that requires high-speed processors for real time applications. As digital signal processors have become cheaper, and their performance greater, it has become possible to use them for controlling electrical drives as a cost effective solution. Some relatively new fully digitized methods, used for speed sensorless field-oriented control, utilize this enhanced processing capacity [47]-[49].

Usually sensorless control is defined as a control scheme where no mechanical parameters like, speed and torque, are measured. Traditional vector control systems use the method of flux and slip estimations based on measurements of the phase currents and DC link voltage of the inverter but, this has a large error in speed

estimation particularly in the low-speed range. MRAS (model reference adaptive system) techniques are also used to estimate the speed of an induction motor [19]-[21]. These also have a speed error in low-speed range and settle to an incorrect steady-state value. In recent years, non-linear observers are used to estimate induction motor parameters and states [27-28], [31-32, 50].

## 5.2 Observers

All states are not available for feedback in many cases and one needs to estimate unavailable state variables. Estimation of unmeasurable state variables is commonly called observation. A device (or a computer program) that estimates or observes the states is called a state-observer or simply an observer. If the state-observer observes all state variables of the system, regardless of whether some state variables are available for direct measurement, it is called a *full-order state-observer*. An observer that estimates fewer than the dimension of the state-vector is called *reduced-order state-observer* or simply a *reduced-order observer*. If the order of the reduced-order state-observer is the minimum possible, the observer is called *minimum-order state-observer*.

Basically, there are two forms of the implementation of an estimator as open-loop and closed-loop. The difference between these two is a correction term, involving the estimation error, used to adjust the response of the estimator. A closed-loop estimator is referred to as an observer.

In open-loop estimators, especially at low speeds, parameter deviations have a significant influence on the performance of the drive both in steady state and transient-state. However, it is possible to improve the robustness against parameter mismatch and also signal noise by using closed loop observers

An observer can be classified according to the type of representation used for the plant to be observed. If the plant is deterministic, then the observer is a deterministic observer; otherwise it is a stochastic observer. The most commonly used observers are Luenberger and Kalman types [2]. The Luenberger observer (LO) is of the deterministic type, and the Kalman Filter (KF) is of the stochastic type. The basic Kalman filter is only applicable to linear stochastic systems, and for non-linear systems the extended Kalman filter (EKF) can be used, which can provide estimates of the states of a system or of both the states and parameters. The EKF is a recursive filter (based on the knowledge of statistics of both the state and noise created by

measurement and system modelling), which can be applied to non-linear time varying stochastic systems. The basic Luenberger observer is applicable to a linear, time-invariant deterministic system. The extended Luenberger observer (ELO) is applicable to non-linear time varying deterministic system. In summary it can be seen that both EKF and ELO are non-linear estimators and the EKF is applicable to stochastic systems and ELO is applicable to deterministic systems. The simple algorithm and the ease of tuning of the ELO may give some advantages over the conventional EKF. However, EKF being insensitive to parameter changes and used for stochastic systems (measurement and modeling noises taken into consideration) it is, therefore, commonly preferred in field-oriented control applications.

Various types of speed observers are discussed in literature, which can be used in high performance induction motor drives such as full-order adaptive state observer. In the full-order adaptive state observer the rotor speed is considered as a parameter, but in ELO and EKF the rotor speed is considered as state. When the appropriate observers are used in high-performance speed sensorless torque-controlled induction motor drives, stable operation can be obtained over a wide-speed range, including very low speeds [26-31], [51-54].

### 5.2.1 General Theory on Observers

An observer can be used to estimate states which cannot be measured, or where the measurements are corrupted by noise. If a system can be described in discrete time as:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) \end{aligned} \quad (5.1)$$

and the system is observable, i.e. the observability matrix,  $\mathbf{M}_o$ , has full rank 1, the states can be estimated by ( 5.2) where

$$\mathbf{M}_o = \begin{bmatrix} \mathbf{C}\mathbf{F} \\ \mathbf{C}\mathbf{F}^2 \\ \vdots \\ \mathbf{C}\mathbf{F}^n \end{bmatrix}$$

$$\begin{aligned} \hat{\mathbf{x}}(k+1) &= \mathbf{A}\hat{\mathbf{x}}(k) + \mathbf{B}\mathbf{u}(k) \\ \hat{\mathbf{x}}(k+1) &= \mathbf{A}\hat{\mathbf{x}}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{L}(\mathbf{y}(k+1) - \mathbf{C}\hat{\mathbf{x}}(k+1)) \end{aligned} \quad (5.2)$$

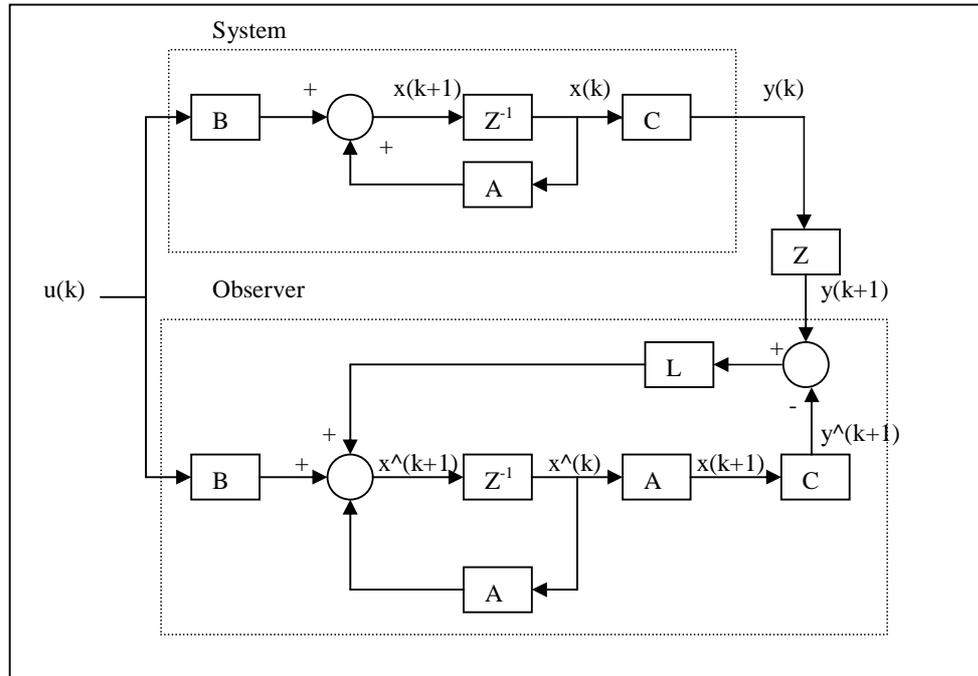


Fig.5.1 –Block diagram of an observer

Fig.5.1 shows the block diagram of the observer which is described by (5.2). The output vector,  $y$ , is used to calculate the current estimate of the state vector,  $x$ . The error of the observer is defined by:

$$\begin{aligned} e(k) &\triangleq x(k) - \hat{x}(k) \\ e(k+1) &\triangleq (A - LCA)e(k) \end{aligned} \quad (5.3)$$

where  $L$  is the observer gain

### 5.3 Kalman Filter

When applied to a physical system, the observer described in section 5.2, will be under the influence of two noise sources:

1. Process noise - i.e. thermic noise in a resistor, which is a part of the system.
2. Measurement noise - i.e. quantization noise.

Considering these two noise sources (5.1) can be rewritten as:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + G_v v(k) \\ y(k) &= Cx(k) + w(k) \end{aligned} \quad (5.4)$$

where  $v(k)$  is the process noise and  $w(k)$  is the measurement noise.

In the following,  $v(k)$  and  $w(k)$  will be regarded as zero mean, uncorrelated white noise sequences with covariances,  $V_1(k)$  and  $V_2(k)$ . The objective of the Kalman algorithm is to determine a gain matrix,  $\mathbf{L}$ , which minimizes the mean square of the error,  $e$ . This can be achieved with the algorithm described in Table 5.1, where:

$$\begin{aligned}\hat{x}(k|n) &\triangleq \mathbf{E}\{x(k) | y(1), y(2), \dots, y(n)\} \\ \mathbf{Q}(k+1) &\triangleq \mathbf{E}\{e(k+1) \cdot e^T(k+1)\}\end{aligned}\quad (5.5)$$

State estimate time update:

$$\hat{x}(k|k-1) = \mathbf{A}(k-1)\hat{x}(k-1|k-1) + \mathbf{B}(k-1)u(k-1)\quad (5.6)$$

Covariance Time update:

$$\mathbf{Q}(k) = \mathbf{A}(k-1)\mathbf{Q}(k-1)\mathbf{A}^T(k-1) + \mathbf{B}(k-1)\mathbf{V}_1(k-1)\mathbf{B}^T(k-1)\quad (5.7)$$

Kalman Gain Matrix:

$$\mathbf{L}(k) = \mathbf{Q}(k)\mathbf{C}^T(k) [\mathbf{C}(k)\mathbf{Q}(k)\mathbf{C}^T(k) + \mathbf{V}_2(k)]^{-1}\quad (5.8)$$

State estimate measurement update:

$$\hat{x}(k|k) = \hat{x}(k|k-1) + \mathbf{L}(k) [y(k) - \mathbf{C}(k)\hat{x}(k|k-1)]\quad (5.9)$$

Table 5.1-Discrete Kalman Filter

If anything but  $x$  kept constant, the covariance matrix will converge towards the solution to the discrete Riccati equation:

$$\mathbf{Q}(k) = \mathbf{A}(k)\mathbf{Q}(k)\mathbf{A}^T(k) + \mathbf{G}_v(k)\mathbf{V}_1(k)\mathbf{G}_v^T(k) - \mathbf{L}'(k)\mathbf{C}(k)\mathbf{Q}(k)\mathbf{A}^T(k)\quad (5.10)$$

where

$$\mathbf{L}'(k) = \mathbf{A}(k)\mathbf{Q}(k)\mathbf{C}^T(k) [\mathbf{C}(k)\mathbf{Q}(k)\mathbf{C}^T(k) + \mathbf{V}_2(k)]^{-1}\quad (5.11)$$

Since the variables in Riccati equation (5.10) are matrices, it is rather complicated to solve symbolically. There exist two special functions to solve algebraic Riccati equations. The function *care*( ) in Matlab can solve continuous-time algebraic Riccati equations and the function *dare*( ) can solve discrete-time algebraic Riccati equations whose general equation [55]:

$$\mathbf{E}^T \mathbf{X} \mathbf{E} = \mathbf{A}^T \mathbf{X} \mathbf{A} - (\mathbf{A}^T \mathbf{X} \mathbf{B} + \mathbf{S})(\mathbf{B}^T \mathbf{X} \mathbf{B} + \mathbf{R})^{-1} (\mathbf{A}^T \mathbf{X} \mathbf{B} + \mathbf{S}) + \mathbf{Q}$$

$$\mathbf{G} = (\mathbf{B}^T \mathbf{X} \mathbf{B} + \mathbf{R})^{-1} (\mathbf{B}^T \mathbf{X} \mathbf{A} + \mathbf{S}^T)$$

This general form is applied to the Kalman filter representation by redefining the elements in Riccati equation as:

$$\mathbf{A} = \mathbf{A}^T$$

$$\mathbf{B} = \mathbf{C}^T$$

$$\mathbf{Q} = \mathbf{G}_v \mathbf{V}_1 \mathbf{G}_v^T$$

$$\mathbf{R} = \mathbf{V}_2$$

$$\mathbf{E} = \mathbf{I}$$

$$\mathbf{S} = \mathbf{0}$$

The stationary covariance matrix,  $\mathbf{Q}$ , and the stationary gain matrix,  $\mathbf{L}'$  can be found by substituting:

$$\mathbf{Q} = \mathbf{X}$$

$$\mathbf{L}' = \mathbf{B}^T$$

This is used in the system in order to get a starting guess of the parameter.

*Note:*

The reason for using  $\mathbf{L}'$  rather than  $\mathbf{L}$  is that the equation corresponds to the closed discrete Kalman filter, displayed on Fig.5.2. However they will converge towards the same result.

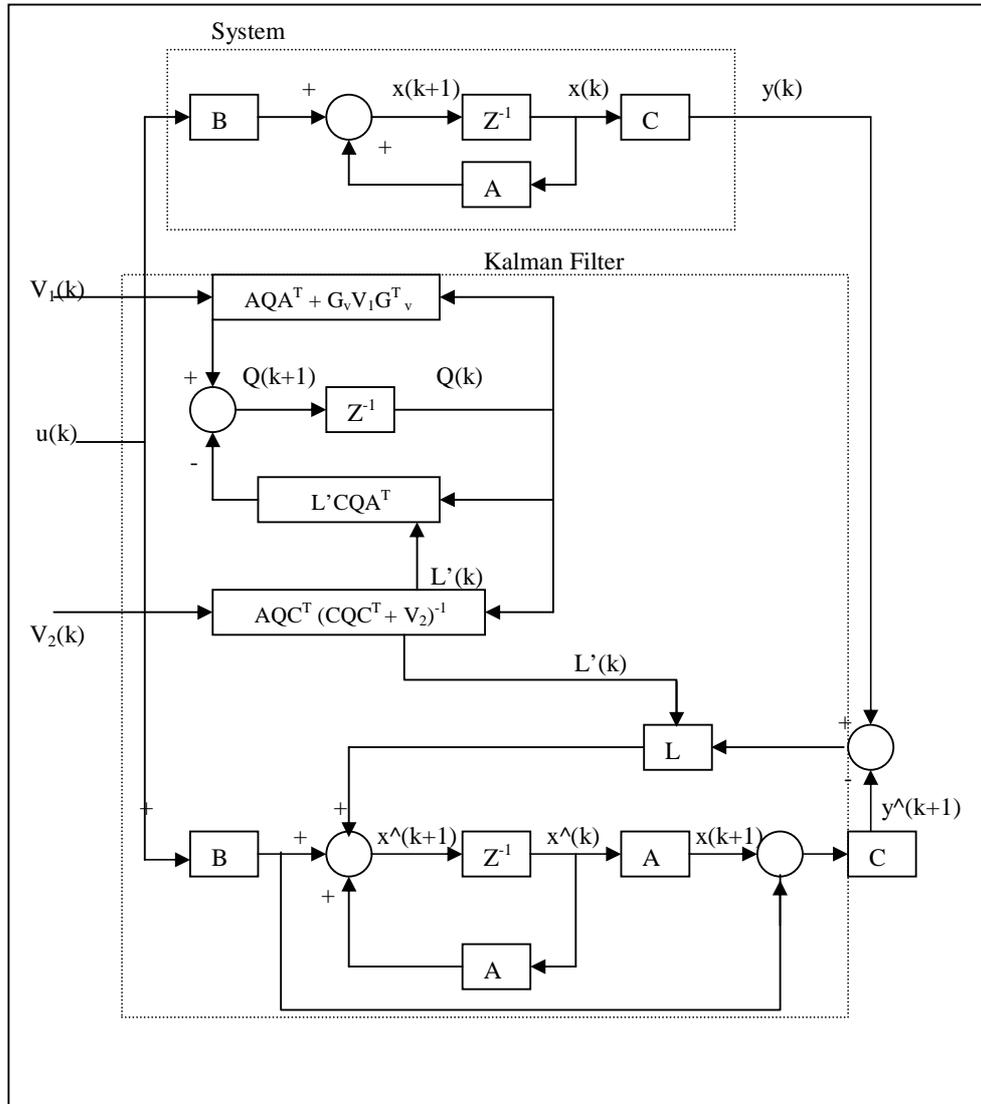


Fig.5.2 –Block Diagram of Kalman Filter

## **5.4 Extended Kalman Filter**

An Extended Kalman Filter is a recursive optimum state-observer that can be used for the state and parameter estimation of a non-linear dynamic system in real time by using noisy monitored signals that are distributed by random noise. This assumes that the measurement noise and system noise are uncorrelated. The noise sources take account of measurement and modeling inaccuracies. In the first stage of the calculations, the states are predicted by using a mathematical model (which contain previous estimates) and in the second stage; the predicted states are continuously corrected by using a feedback correction scheme. This scheme makes use of actual measured states, by adding a term to the predicted states (which is obtained in the first stage). The additional term contains the weighted difference of measured and estimated output signals. Based on the deviation from the estimated value, the EKF provides an optimum output value at the next input instant. In an induction motor drive the EKF can be used for the real-time estimation of the rotor speed, but it can also be used for state and parameter estimation. For this purpose the stator voltages and currents are measured (or the stator voltages are reconstructed from DC link voltage and the inverter switching signals) and, for example, the speed of the machine can be obtained by the EKF quickly and precisely [56].

### **5.4.1 Application of the Extended Kalman Filter**

In the present section the Extended Kalman Filter (EKF) is used for the estimation of the rotor speed of an induction motor. The EKF is suitable for use in high-performance induction motor drives, and it can provide accurate speed-estimates in a wide speed range including very low speeds as well [27-31], [51-54].

The main design steps for a speed sensorless induction motor drive implementation using the discretized EKF algorithm are as follows:

- Selection of the time-domain induction machine model,
- Discretization of the induction machine model,
- Determination of the noise and state covariance matrices,
- Implementation of the discretized EKF algorithm; tuning.

For the purpose of using an EKF for the estimation of the rotor speed of an induction machine, it is possible to use various machine models. For example, it is possible to use the equations expressed in the rotor flux-oriented reference frame, or in stator

flux-oriented reference frame. In order to avoid extra calculations and some non-linear transformations, stationary reference frame is preferred [48]. The main advantages of using the model in stationary reference frame are:

- Reduced computation time,
- Smaller sampling time,
- Higher accuracy,
- More stable behavior.

Thus, we have chosen stationary reference frame in our simulation and experimental implementation.

#### 5.4.2 Motor Model for EKF

The model for induction motor developed in stationary reference frame and used in the previous studies [2], [28] is given below:

$$\frac{d}{dt} \begin{bmatrix} i_{ds}^s \\ i_{qs}^s \\ \Psi_{dr}^s \\ \Psi_{qr}^s \\ w_r \end{bmatrix} = \begin{bmatrix} -\frac{K_R}{K_L} & 0 & \frac{L_m R_r}{L_r^2 K_L} & \frac{L_m w_r}{L_r K_L} & 0 \\ 0 & -\frac{K_R}{K_L} & -\frac{L_m w_r}{L_r K_L} & \frac{L_m R_r}{L_r^2 K_L} & 0 \\ \frac{L_m}{T_r} & 0 & -\frac{1}{T_r} & -w_r & 0 \\ 0 & \frac{L_m}{T_r} & w_r & -\frac{1}{T_r} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_{ds}^s \\ i_{qs}^s \\ \Psi_{dr}^s \\ \Psi_{qr}^s \\ w_r \end{bmatrix} + \frac{1}{K_L} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_{ds}^s \\ V_{qs}^s \end{bmatrix} \quad (5.12)$$

$$\begin{bmatrix} i_{ds}^s \\ i_{qs}^s \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_{ds}^s \\ i_{qs}^s \\ \Psi_{dr}^s \\ \Psi_{qr}^s \\ w_r \end{bmatrix} \quad (5.13)$$

where

$$\frac{K_L}{K_R} = \left( \frac{R_s}{\sigma L_s} + \frac{1-\sigma}{\sigma \tau_r} \right) = \left( \frac{1}{T_s} + \frac{1-\sigma}{T_r} \right) = \frac{R_s + R_r (L_m/L_r)^2}{L_s}$$

$L_r, L_s, L_m$  are rotor, stator and main inductances

$T_r, T_s$  are rotor and stator time constants

This model has a disadvantage; its order is higher. This will be a drawback when the EKF algorithm has to be implemented in real-time. One great advantage of this model, however, is that it does not need speed measurement, so neither the flux speed nor the rotor speed has to be known. The other is that, the flux model mentioned in Chapter 1 can be omitted, since this model also estimates the flux, and so the angle of the flux and any other parameter can be directly calculated.

It should be noted that in (5.12) it has been assumed that the rotor-speed derivative is negligible,  $dw_r/dt=0$ . Although the last row of the A [5x5] matrix in (5.12) corresponds to infinite inertia in reality it is not and the required correction is accomplished by the Kalman filter (by the system noise compensation, which also takes account of the computational inaccuracies ) [48]. If the load-torque is not known, the change of  $w_r$  cannot be found from the remaining states and control signals. This problem can be overcome by introducing the mechanical speed as a parameter rather than a state [21].  $w_r$  is assumed to be constant during the state estimate time update computation but it is included in covariance time update computation. The speed will, therefore, be estimated in the state estimate measurement update step. Furthermore, it should be noted that the effects of the saturation in magnetic paths of the machine have been neglected in the model. This assumption is justifiable. It can be shown that the EKF is not sensitive to changes in inductances, since changes in the stator parameters are being compensated by EKF. The application of (5.12) to the EKF will give not only the rotor speed, but also the rotor flux-linkage components (and consequently the angle and modulus of the rotor flux-linkage space-vector will also be known). This is useful for high performance field-oriented drive implementations. It is important to emphasize that the rotor speed has been considered as a state variable and the system matrix A is non-linear and it contains the speed,  $A=A(x)$ . The compact form of (5.12) and (5.13) are:

$$\frac{dx}{dt} = Ax + Bu \quad (5.14)$$

$$y = Cx \quad (5.15)$$

where:

$$A = \begin{bmatrix} -\frac{K_R}{K_L} & 0 & \frac{L_m R_r}{L_r^2 K_l} & \frac{L_m w_r}{L_r K_L} & 0 \\ 0 & -\frac{K_R}{K_L} & -\frac{L_m w_r}{L_r K_L} & \frac{L_m R_r}{L_r^2 K_l} & 0 \\ \frac{L_m}{T_r} & 0 & -\frac{1}{T_r} & -w_r & 0 \\ 0 & \frac{L_m}{T_r} & w_r & -\frac{1}{T_r} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (5.16)$$

$$\text{and } C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$x = [i_{ds}^s \quad i_{qs}^s \quad \psi_{dr}^s \quad \psi_{qr}^s \quad w_r]^T$  is the state vector,  $u$  is the input vector,  $u = [V_{ds}^s \quad V_{qs}^s]^T$ ,  $A$  is the system matrix, and  $C$  is the output matrix.

### 5.4.3 Discretized augmented machine model

The motor equations (5.14) and (5.15) are to be discretized for the digital implementation of EKF as:

$$x(k+1) = A_d x(k) + B_d u(k) \quad (5.17)$$

$$y(k) = C_d x(k) \quad (5.18)$$

$A_d$  and  $B_d$  matrices in the (5.17) are discretized system and input matrices, respectively. They are:

$$A_d = \exp[AT] \approx I + AT + \frac{(AT)^2}{2} \quad (5.19)$$

$$B_d \approx BT + \frac{ABT^2}{2} \quad (5.20)$$

where  $T$  is the sampling time. Note that the discrete output matrix  $C_d=C$  is defined in (5.16). When the last terms in (5.19) and (5.20) are ignored, then very short sampling-times, they require, are attainable to have a stable and accurate discretized model.

However, a better approximation is obtained with the given second-order series expansion at (5.19) and (5.20). In general to achieve an adequate accuracy, the sampling-time should be appreciably smaller than the characteristic time-constants of the machine. The final choice for this should be based on obtaining adequate execution time of the full EKF algorithm and also satisfactory accuracy and stability. The second-order technique obviously increases the computational time. If the second-order terms are neglected in (5.19) and (5.20) then the discrete form of (5.14) and (5.15) become:

$$\mathbf{x}(k+1) = \mathbf{A}_d \mathbf{x}(k) + \mathbf{B}_d \mathbf{u}(k) \quad (5.21)$$

$$\mathbf{y}(k) = \mathbf{C}_d \mathbf{x}(k) \quad (5.22)$$

where

$$\mathbf{A}_d = e^{\mathbf{A}T} \approx \mathbf{I} + \mathbf{A}T \quad (5.23)$$

$$\mathbf{B}_d = \int_0^T e^{\mathbf{A}\zeta} \mathbf{B} d\zeta \approx \mathbf{B}T \quad (5.24)$$

$$\mathbf{C}_d = \mathbf{C} \quad (5.25)$$

$$\mathbf{A}_d = \begin{bmatrix} 1 - T \frac{\mathbf{K}_R}{\mathbf{K}_L} & 0 & T \frac{\mathbf{L}_m \mathbf{R}_r}{\mathbf{L}_r^2 \mathbf{K}_L} & T \frac{\mathbf{L}_m \mathbf{w}_r}{\mathbf{L}_r \mathbf{K}_L} & 0 \\ 0 & 1 - T \frac{\mathbf{K}_R}{\mathbf{K}_L} & -T \frac{\mathbf{L}_m \mathbf{w}_r}{\mathbf{L}_r \mathbf{K}_L} & T \frac{\mathbf{L}_m \mathbf{R}_r}{\mathbf{L}_r^2 \mathbf{K}_L} & 0 \\ T \frac{\mathbf{L}_m}{\mathbf{T}_r} & 0 & 1 - T \frac{1}{\mathbf{T}_r} & -T \mathbf{w}_r & 0 \\ 0 & T \frac{\mathbf{L}_m}{\mathbf{T}_r} & T \mathbf{w}_r & 1 - T \frac{1}{\mathbf{T}_r} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.26)$$

$$\mathbf{B}_d = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{C}_d = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (5.27)$$

$$x(k) = \left[ i_{ds}^s(k) \quad i_{qs}^s(k) \quad \psi_{dr}^s(k) \quad \psi_{qr}^s(k) \quad w_r(k) \right]^T \quad (5.28)$$

$$u(k) = \left[ V_{ds}^s(k) \quad V_{qs}^s(k) \right]^T \quad (5.29)$$

By considering the system noise  $v(k)$  ( $v$  is the noise vector of states), being zero-mean white-Gaussian and independent of  $x(k)$  with a covariance matrix  $Q$ , the system model becomes:

$$x(k+1) = \mathbf{A}_d x(k) + \mathbf{B}_d u(k) + v(k) \quad (5.30)$$

By considering a zero-mean white-Gaussian measurement noise,  $w(k)$  (noise in the measured stator currents) which is independent of  $y(k)$  and  $v(k)$  with a covariance matrix  $R$ , the output equation becomes :

$$y(k) = \mathbf{C}x(k) + w(k) \quad (5.31)$$

#### 5.4.4 Implementation of the Discretized EKF Algorithm

##### 5.4.4.1 Determination of the noise and state covariance matrices

To be more specific, the goal of the Kalman filter is to obtain unmeasurable states (i.e. covariance matrices  $Q$ ,  $R$ ,  $P$  of the system noise vector, measurement noise vector, and system state vector ( $x$ ) respectively). In general, by means of noise inputs, it is possible to take computational inaccuracies, modeling errors, and errors in measurements into account in modeling the system. The filter estimation ( $\hat{x}$ ) is obtained from the predicted values of the states ( $x$ ) and this is corrected recursively by using a correction term, which is product of the Kalman gain ( $L$ ) and the deviation of the estimated measurement output vector and the actual output vector ( $y - \hat{y}$ ). The Kalman gain is chosen to result in the best possible estimated states.

Thus filtering algorithm contains basically two main stages, a prediction stage and a filtering stage. During the prediction stage, the next predicted values of the states  $x(k+1)$  are obtained by using a mathematical model (state variable equations) and also the previous values of the estimated states. Furthermore, the predicted-state covariance matrix ( $P$ ) is also obtained before the new measurements are made and for

this purpose the mathematical model and also the covariance matrix of the system ( $Q$ ) are used. In the second stage which is the filtering stage, the next estimated states,  $\hat{x}(k+1)$ , are obtained from the predicted estimates  $x(k+1)$  by adding a correction term  $L(y - \hat{y})$  to the predicted value. This correction term is a weighted difference between the actual output vector ( $y$ ) and the predicted output vector ( $\hat{y}$ ), where  $L$  is the Kalman gain. Thus the predicted state-estimate (and also covariance matrix) is corrected through a feedback correction scheme that makes use of actual measured quantities. The Kalman gain is chosen to minimize the estimation error variance of the states to be estimated. The computations are realized by using recursive relations.

The algorithm is computationally intensive, and the accuracy also depends on the model parameters used. A critical part of the design is to use correct initial values for the various covariance matrices. These can be obtained by considering the stochastic properties of the corresponding noises. Since these are usually not known, in most cases they are used as weight matrices, but it should be noted that sometimes simple qualitative rules can be set up for obtaining the covariance in the noise vectors. With advances in DSP technology, it is possible to implement an EKF conveniently in real time [48,49].

The system noise covariance matrix ( $Q$ ) is  $[5 \times 5]$ , and the measurement noise covariance matrix ( $R$ ) is  $[2 \times 2]$  matrix, so in general this would require the knowledge of 29 elements. However, by assuming that the noise signals are not correlated, both  $Q$  and  $R$  are diagonal, and only 5 elements must be known in  $Q$  and 2 elements in  $R$ . However, the parameters in  $\alpha$ - and  $\beta$ - axes are the same, which means that the first two elements of the diagonal are equal ( $q_{11}=q_{22}$ ), the third and fourth elements in the diagonal of  $Q$  are equal ( $q_{33}=q_{44}$ ), so  $Q=\text{diag}(q_{11},q_{11},q_{33},q_{33},q_{55})$  contains only 3 elements which have to be known. Similarly, the two diagonal elements in  $R$  are equal ( $r_{11}=r_{22}$ ), thus  $R=\text{diag}(r_{11}, r_{11})$ . It follows that in total only 4 noise covariance elements needs to be known.

$$Q = \begin{bmatrix} Qi_{ds}^s & 0 & 0 & 0 & 0 \\ 0 & Qi_{qs}^s & 0 & 0 & 0 \\ 0 & 0 & Q\Psi_{dr}^s & 0 & 0 \\ 0 & 0 & 0 & Q\Psi_{qr}^s & 0 \\ 0 & 0 & 0 & 0 & Qw_r \end{bmatrix} \quad R = \begin{bmatrix} Ri_{ds}^s & 0 \\ 0 & Ri_{qs}^s \end{bmatrix} \quad (5.32)$$

Starting values of the state vector  $x_0$  and the starting values of the noise covariance matrices  $Q_0$  and  $R_0$  are set together with the starting value of the state covariance matrix  $P_0$ , where  $P$  is the covariance matrix of the state vector. The starting state covariance matrix can be considered as diagonal matrix, where all elements are equal. The initial values of the matrices reflect the degree of knowledge of the initial states: the higher their value, the less accurate is any available information on the initial states. Thus the new measurement data will be more heavily weighted and the covariance speed of the estimation process will increase. However, divergence problem or large oscillations of the state estimates around the true value may occur when too high initial covariance values are chosen. A suitable selection allows us to obtain satisfactory speed convergence, and avoid divergence problems or unwanted oscillations.

The accuracy of the state estimation is affected by the amount of information that the stochastic filter can extract from its mathematical model and the measurement data processing. Some of the estimated variables, especially unmeasured ones, may indirectly and weakly be linked to the measurement data, so only poor information is available to the EKF. After deciding how to initialize the covariance matrices, the next step is prediction of the state vector.

### 1-Prediction of the state vector

Prediction of the state vector at sampling time  $(k+1)$  from the input  $u(k)$ , state vector at previous sampling time,  $x_{k|k}$ , by using  $A_d$  and  $B_d$  is obtained from

$$x_{k+1|k} = \mathbf{A}_d x_{k|k} + \mathbf{B}_d u(k) \quad (5.33)$$

$$x_{k+1|k} \triangleq \mathbf{F}(k+1, k, x_{k|k}, u(k)) \quad (5.34)$$

where

$$\mathbf{F} = \begin{bmatrix} \left(1 - T \frac{K_R}{K_L}\right) i_{ds}^s + T \frac{L_m R_r}{L_r^2 K_L} \psi_{dr}^s + T \frac{L_m w_r}{L_r K_L} \psi_{qr}^s + T \frac{1}{K_L} V_{ds}^s \\ \left(1 - T \frac{K_R}{K_L}\right) i_{qs}^s - T \frac{L_m R_r}{L_r^2 K_L} \psi_{dr}^s + T \frac{L_m w_r}{L_r K_L} \psi_{qr}^s + T \frac{1}{K_L} V_{qs}^s \\ T \frac{L_m}{T_r} i_{ds}^s + \left(1 - T \frac{1}{T_r}\right) \psi_{dr}^s - T w_r \psi_{qr}^s \\ T \frac{L_m}{T_r} i_{qs}^s + T w_r \psi_{dr}^s + \left(1 - T \frac{1}{T_r}\right) \psi_{qr}^s \\ w_r \end{bmatrix} \quad (5.35)$$

$$\mathbf{h} \triangleq \mathbf{C}_d \mathbf{x}_{k|k+1} = \begin{bmatrix} i_{ds}^s \\ i_{qs}^s \end{bmatrix} \quad (5.36)$$

The notation  $\mathbf{x}_{k+1|k}$  means that it is a predicted value at the (k+1)-th instant, and it is based on measurements up to k-th instant. In the following step of the recursive EKF computation, covariance matrix of prediction is computed.

## 2-Prediction covariance computation

The prediction covariance is updated by:

$$\mathbf{P}_{k+1|k} = \mathbf{M} \mathbf{P}_{k|k} \mathbf{M}^T + \mathbf{Q} \quad \text{where} \quad \mathbf{M} = \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \Bigg|_{\mathbf{x} = \hat{\mathbf{x}}_{k|k}} \quad (5.37)$$

with

$$\frac{\partial \mathbf{F}}{\partial \mathbf{x}} = \begin{bmatrix} 1 - T \frac{K_R}{K_L} & 0 & T \frac{L_m R_r}{L_r^2 K_L} & T \frac{L_m w_r}{L_r K_L} & T \frac{L_m}{L_r K_L} \psi_{qr}^s \\ 0 & 1 - T \frac{K_R}{K_L} & -T \frac{L_m w_r}{L_r K_L} & T \frac{L_m R_r}{L_r^2 K_L} & -T \frac{L_m}{L_r K_L} \psi_{dr}^s \\ T \frac{L_m}{T_r} & 0 & 1 - T \frac{1}{T_r} & T w_r & T \psi_{qr}^s \\ 0 & T \frac{L_m}{T_r} & T w_r & 1 - T \frac{1}{T_r} & T \psi_{dr}^s \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.38)$$

$$\frac{\partial \mathbf{h}}{\partial \mathbf{x}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (5.39)$$

In (5.38) there are 17 elements which are constant and 8 elements which are variable. Thus, in real time applications products involving the speed and the flux-linkages have to be computed. Next step is the computation of the Kalman filter gain matrix .

### 3-Kalman Gain Computation

The Kalman filter gain (correction matrix) is computed as;

$$\mathbf{L}_k = \mathbf{P}_{k|k-1} \mathbf{N}^T [\mathbf{N} \mathbf{P}_{k|k-1} \mathbf{N}^T + \mathbf{R}]^{-1} \text{ where } \mathbf{N} = \left. \frac{\partial h}{\partial x} \right|_{x = \hat{x}_{k|k-1}} \quad (5.40)$$

### 4-State Vector Estimation

The predicted state-vector is added to the innovation term multiplied by Kalman gain to compute state-estimation vector. The state-vector estimation (filtering) at time (k) is determined as:

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{L}_k (y_k - \hat{y}_k) \quad (5.41)$$

where

$$\hat{y}_k = \mathbf{C}_d \mathbf{x}_{k|k-1} \quad (5.42)$$

### 5-Estimation Covariance Computation

The last step is estimation covariance computation as;

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{L}_k \left. \frac{\partial h}{\partial x} \right|_{x = \mathbf{x}_{k|k-1}} \mathbf{P}_{k|k-1} \quad (5.43)$$

after all steps executed, set  $k=k+1$  and start from the step-1 to continue the computation recursively.

The EKF described here can be used for either steady-state or transient conditions of the induction machine for the estimation of the rotor speed. The speed estimation scheme requires the monitored stator voltages and stator currents. Instead of using monitored stator voltages, the stator voltage can also be reconstructed by using DC link voltage and inverter switching states, but especially at low speeds it is

necessary to have appropriate dead-time compensation and also the voltage drops across the inverter switches must be considered.

The tuning of the EKF involves an iterative modification of the machine parameters and covariances in order to yield the best estimate of the states. Changing the covariance matrices  $Q$  and  $R$  affect both the transient- and the steady-state operation of the filter. Also in the implementation of the EKF different  $Q$  and  $R$  matrices may be tried to detect the optimum cases which increase performance of the EKF. For constant  $Q$  and  $R$  values either steady-state or transient conditions have poor performance. If high accuracy is required for both conditions then an algorithm that switches to different covariance values at different operating points may be added to the main EKF algorithm (Noise Level Adjustment). This is also studied for both transient- and steady-state conditions and perfect results are obtained by this way. It should be noted about the following qualitative tuning rules:

- 1-) If  $R$  is large then  $L$  is small and transient performance is faster.
- 2-) If  $Q$  is large then  $L$  is large and transient performance is slower [2].

However, if  $Q$  is too large, or if  $R$  is too small instability may occur

## **5.5 State Estimation Simulations with EKF**

In this part, the state estimation performance of EKF is simulated. The simulation is implemented with Matlab/Simulink. In this simulation input voltages and measured currents in stationary reference frame are produced by FOC simulation which was implemented in Chapter 2. It is quite difficult to implement all matrix operations and overall computation using only Simulink. Thus, EKF algorithm is developed as a S-function and than inserted to Simulink in the form of S-function block. S-functions (system-functions) provide a powerful mechanism for extending the capabilities of Simulink. S-functions use a special calling syntax that enables you to interact with Simulink's equation solvers. This interaction is very similar to the interaction that takes place between the solvers and built-in Simulink blocks. The form of an S-function is very general and can accommodate continuous, discrete, and hybrid systems. As a result, nearly all Simulink models can be described as S-functions.

The most common use of S-functions is in creating custom Simulink blocks. You can use S-functions for a variety of applications, including:

- Adding new general-purpose blocks to Simulink,
- Incorporating existing C code into a simulation,
- Describing a system as a mathematical set of equations,
- Using graphical animations.

An advantage of using S-functions is that one can build a general purpose block that can be used many times in a model, varying parameters with each instance of the block [22]. The simulink model and S-function code is given in Appendix B.

In the simulation parameters of a 1-hp motor are used. Base excitation frequency is 60 Hz. The observable states in this model as mentioned in (5.28) are:  $\{ i_{ds}^s(k) \ i_{qs}^s(k) \ \psi_{dr}^s(k) \ \psi_{qr}^s(k) \ w_r(k) \}$ . In Fig.5.3 speed reversal of the motor

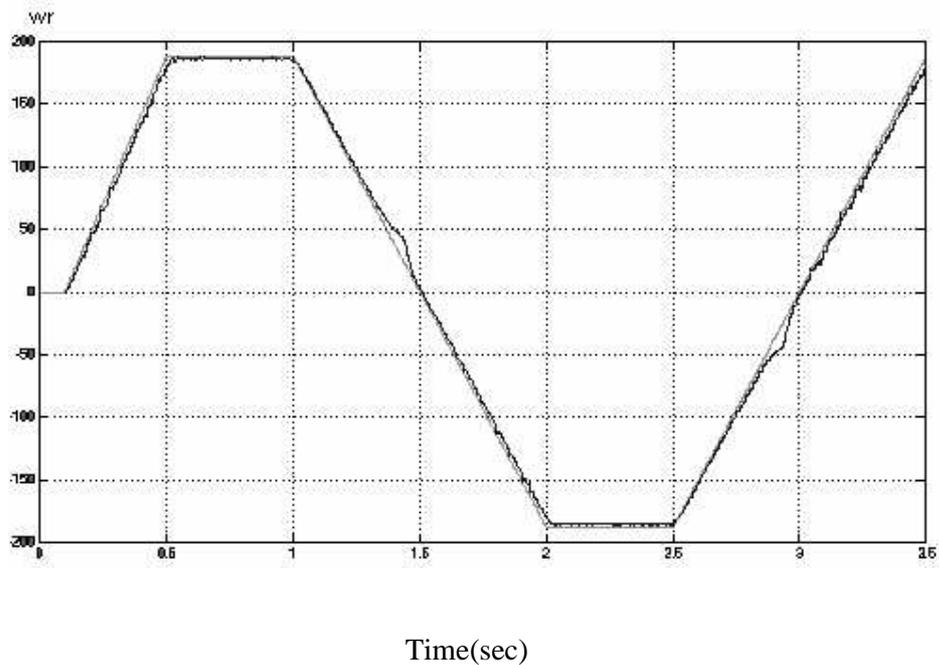


Fig.5.3 – High Speed, No-Load, Four Quadrant Speed Estimation with EKF  
(in  $(P/2)* [rad/sec]$ )

at no-load is given with reference speed. The estimated speed (jittery) and the reference speed (linear) are plotted together. Measurement and state covariances are chosen so that both the transient and steady state speed errors are optimized. One may

choose different covariances and obtain almost zero steady-state speed error with a poor transient speed estimation as shown in Fig.5.5 or vice versa.

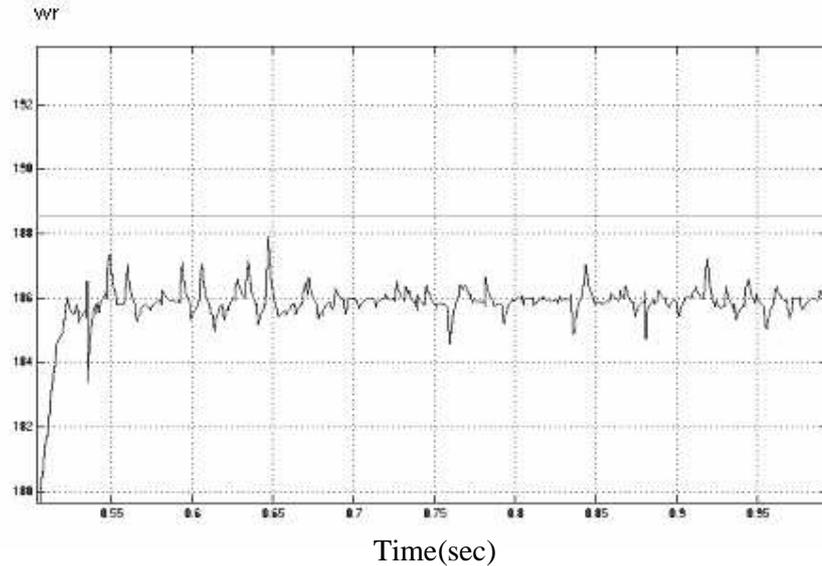


Fig.5.4 – (Fig.5.3-Zoomed at steady state) High Speed, No-Load, Four Quadrant Speed Estimation with EKF at Steady State (in  $(P/2)^*$  [rad/sec])

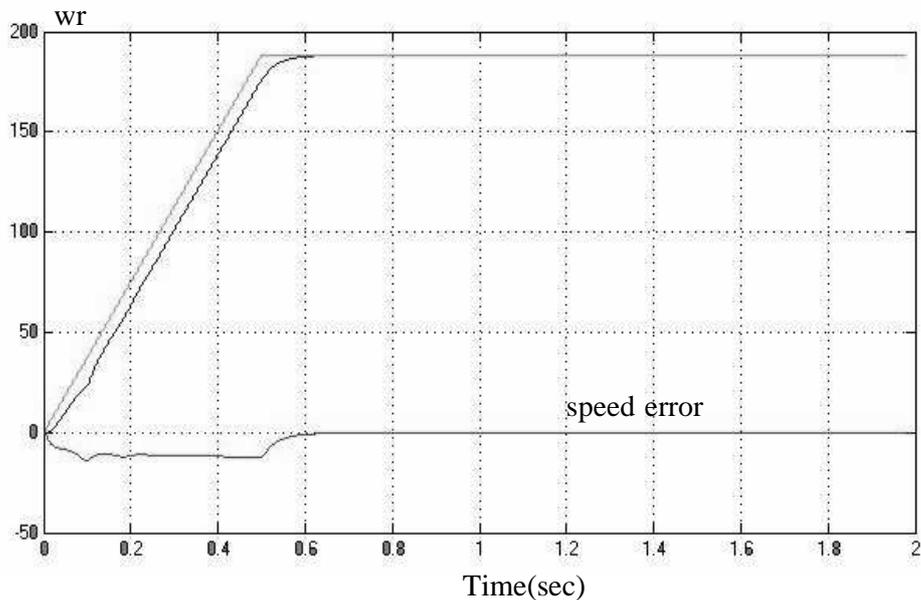


Fig.5.5- High Speed, No-Load, Speed Estimation with EKF – Steady State Performance Optimized (in  $(P/2)^*$  [rad/sec])

In the case of Fig.5.5 simulation, state covariance is decreased; the algorithm begins to behave such that the state space model gives more accurate estimates compared to measured values so it assigns less importance to the measurements. This causes a decrease in Kalman gain which reduces the correction speed of the currents. In the

extra time used for current correction the algorithm finds opportunity to decrease the steady-state error.

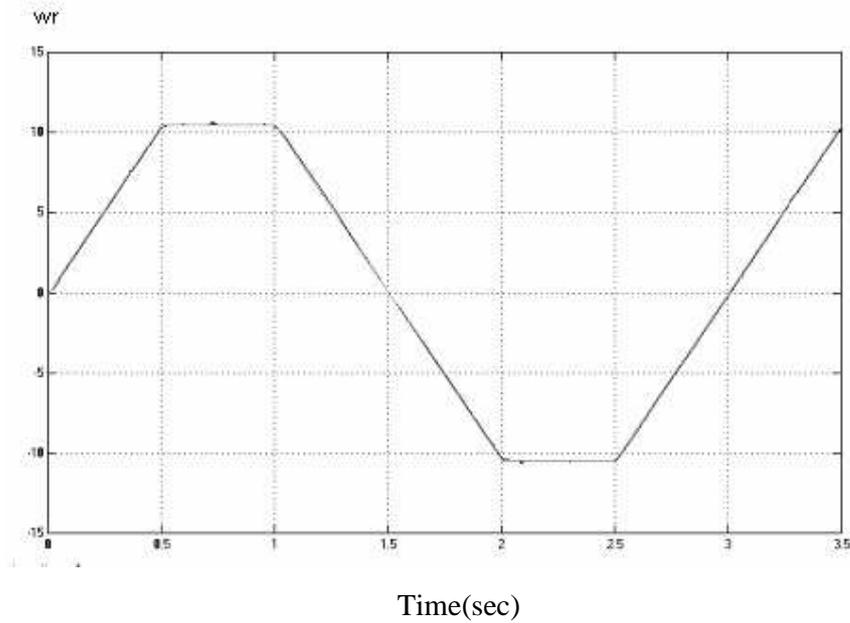


Fig.5.6 – Low Speed, No-Load, Four Quadrant Speed Estimation with EKF  
(in  $(P/2)* [rad/sec]$ )

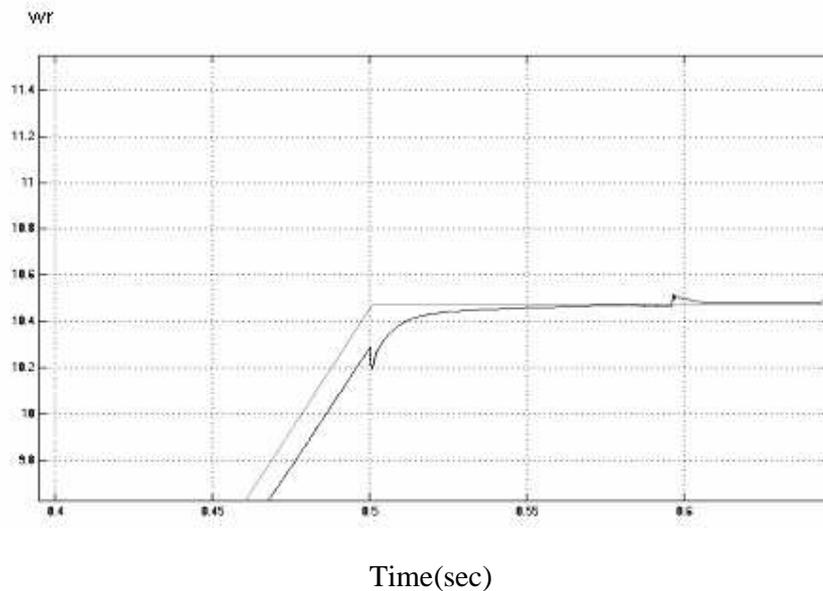


Fig.5.7 – (Fig.5.6-Zoomed) Low Speed, No-Load, Speed Estimation with EKF at  
Steady State to Transient State (in  $(P/2)* [rad/sec]$ )

Low speed estimation performance of the EKF is also quite satisfactory and close to reference speed as shown in Fig (5.6)-(5.7).

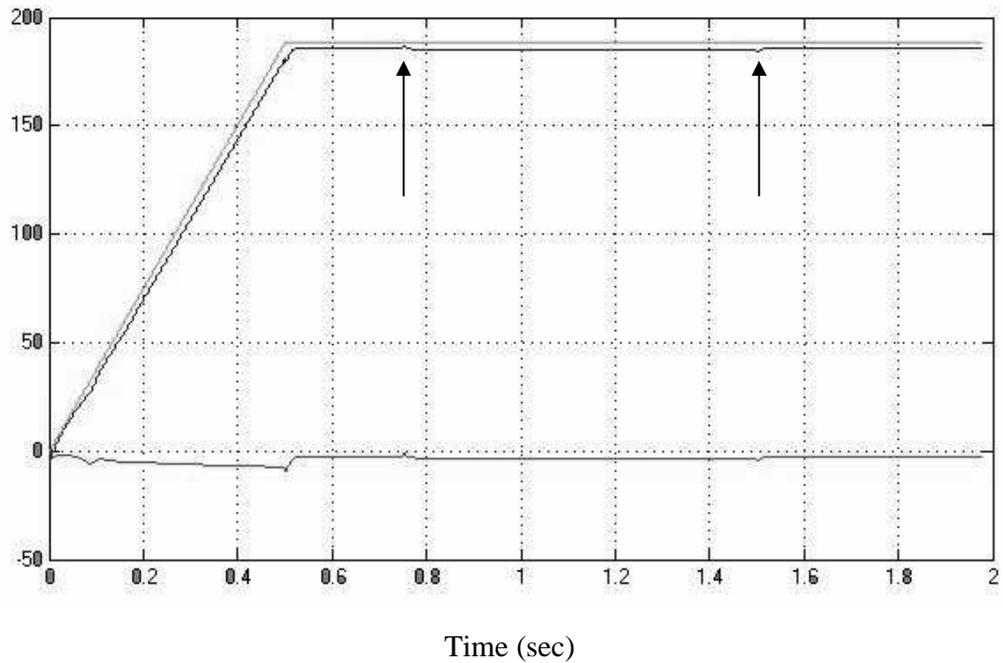


Fig.5.8- High Speed, Full-Load, Speed Estimation with EKF  
(in  $(P/2)* [rad/sec]$ )

In Fig.5.8 rated mechanical load is applied to the motor between 0.75-1.5 sec. to verify the performance of EKF under loaded conditions. As shown above EKF works properly even under fully loaded case. One may decrease steady-state error to very low levels with appropriate state covariances optimized for steady state.

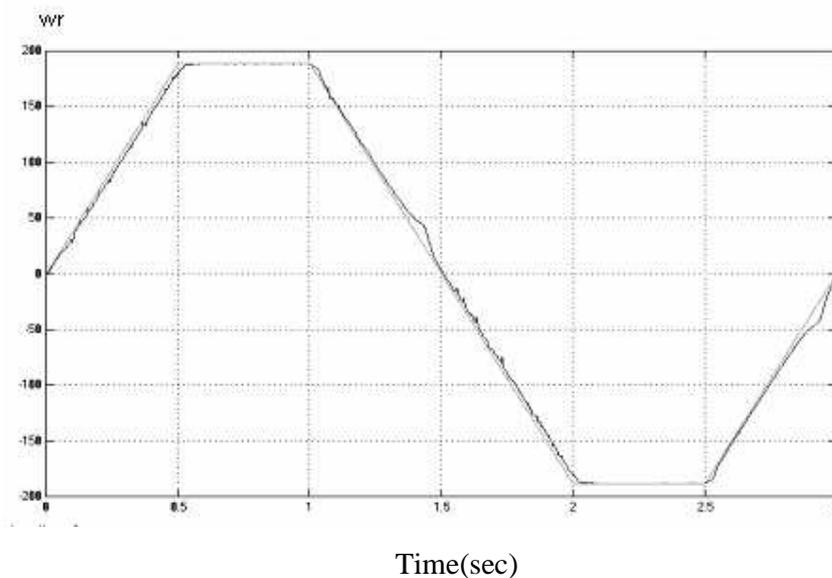


Fig.5.9- High Speed, No-Load, Speed Estimation using EKF with Adjustable  
Noise Level (in  $(P/2)* [rad/sec]$ )

In Fig.5.9 different from Fig.5.3 both the steady state and transient state errors are minimized individually with adjustable noise level technique (ANLT). In ANLT, different covariances are assigned for certain ranges of time both at steady state and transient-state by a basic switching logic speed-error is minimized separately in each range.

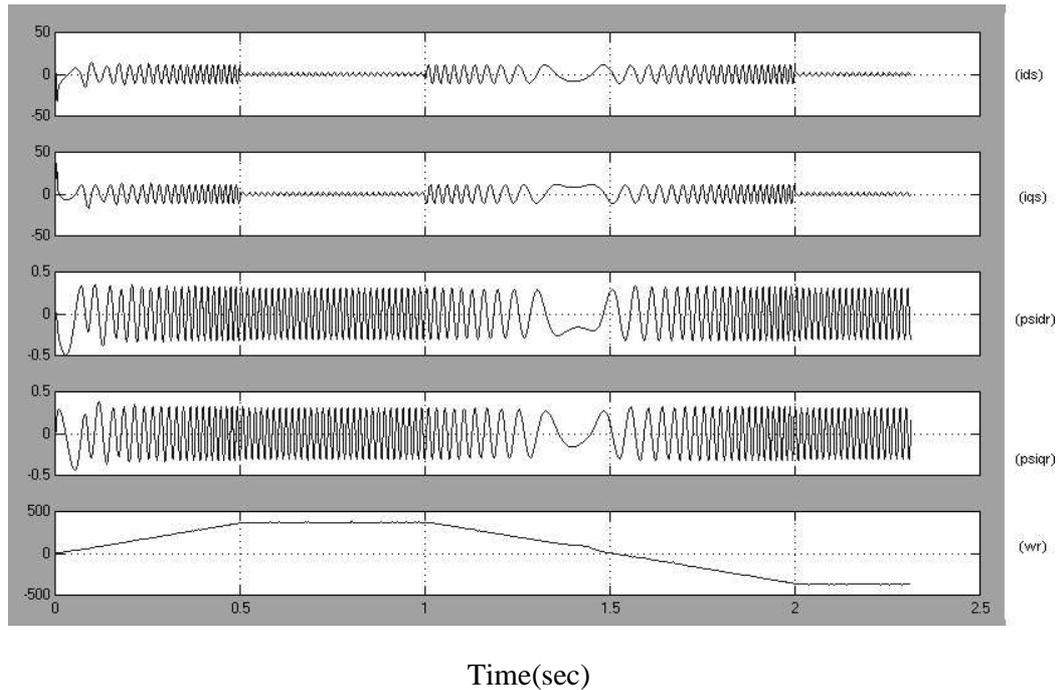


Fig.5.10- Estimated States in (5.28) Respectively at No Load

In Fig 5.10 all the states estimated by EKF are given together. The amplitude of the stator currents increases at transient states due to inertia of the motor and decrease to very low value at steady state as shown in Fig. 5.10 and 5.11. Note that, when the speed of the motor is close to zero, the frequency of the currents and fluxes decrease and become dc. This range is very problematic in induction motor FOC control due to extremely low frequency. The estimated speed waveform of EKF slightly deviates because of this reason. At low speeds performance of EKF is being affected negatively due to added negative effects of some other factors such as inaccurate parameter values, presence of voltage drops on the switches which are not accounted in the model, etc., as well. In Fig.5.12 dq-axis rotor fluxes and rotor flux magnitude are shown in enlarged form. The constant amplitude flux and smooth flux magnitude is vital for FOC as mentioned in Chapter 2. This demand is provided by EKF estimates as long as proper FOC technique is applied to the system.

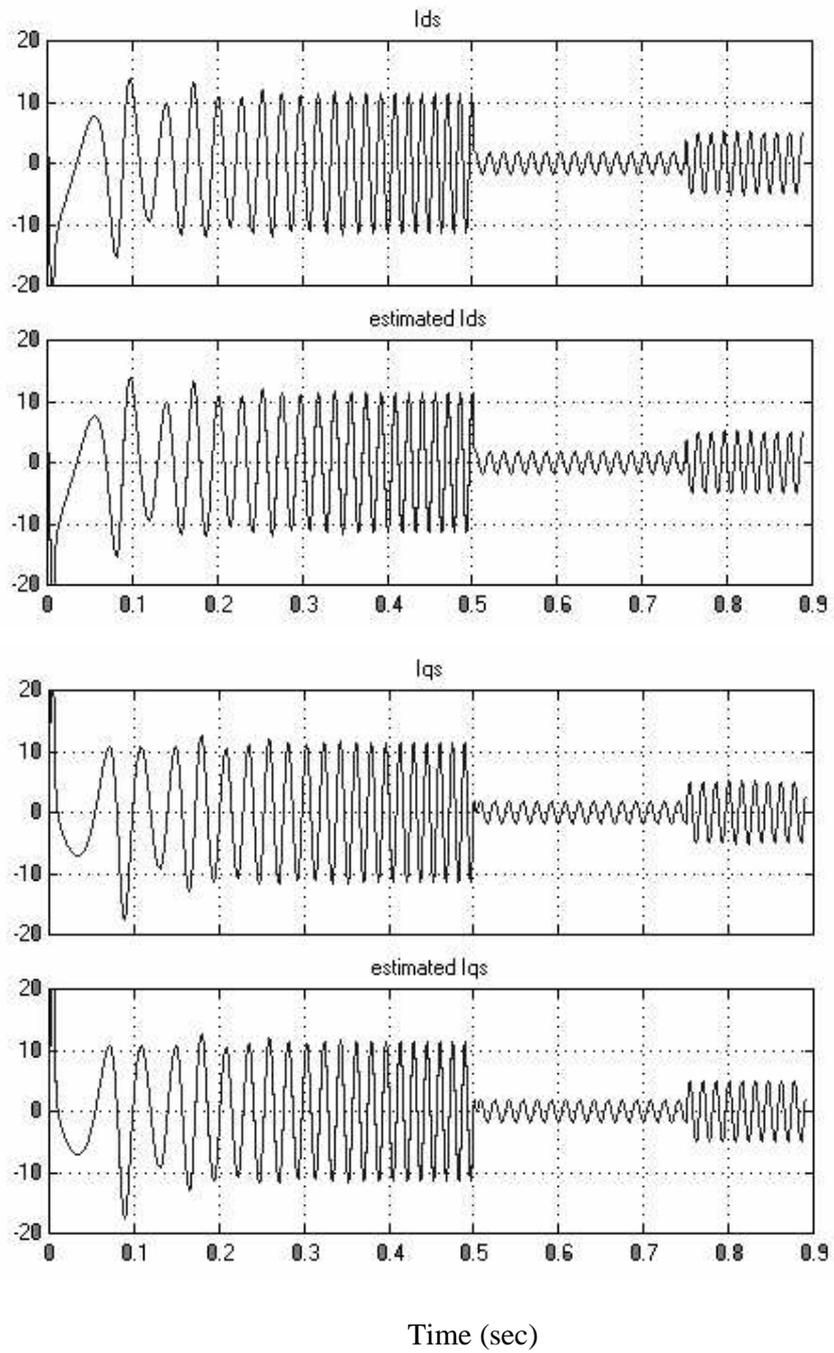


Fig.5.11- State I and II (dq-axis Stator Currents )

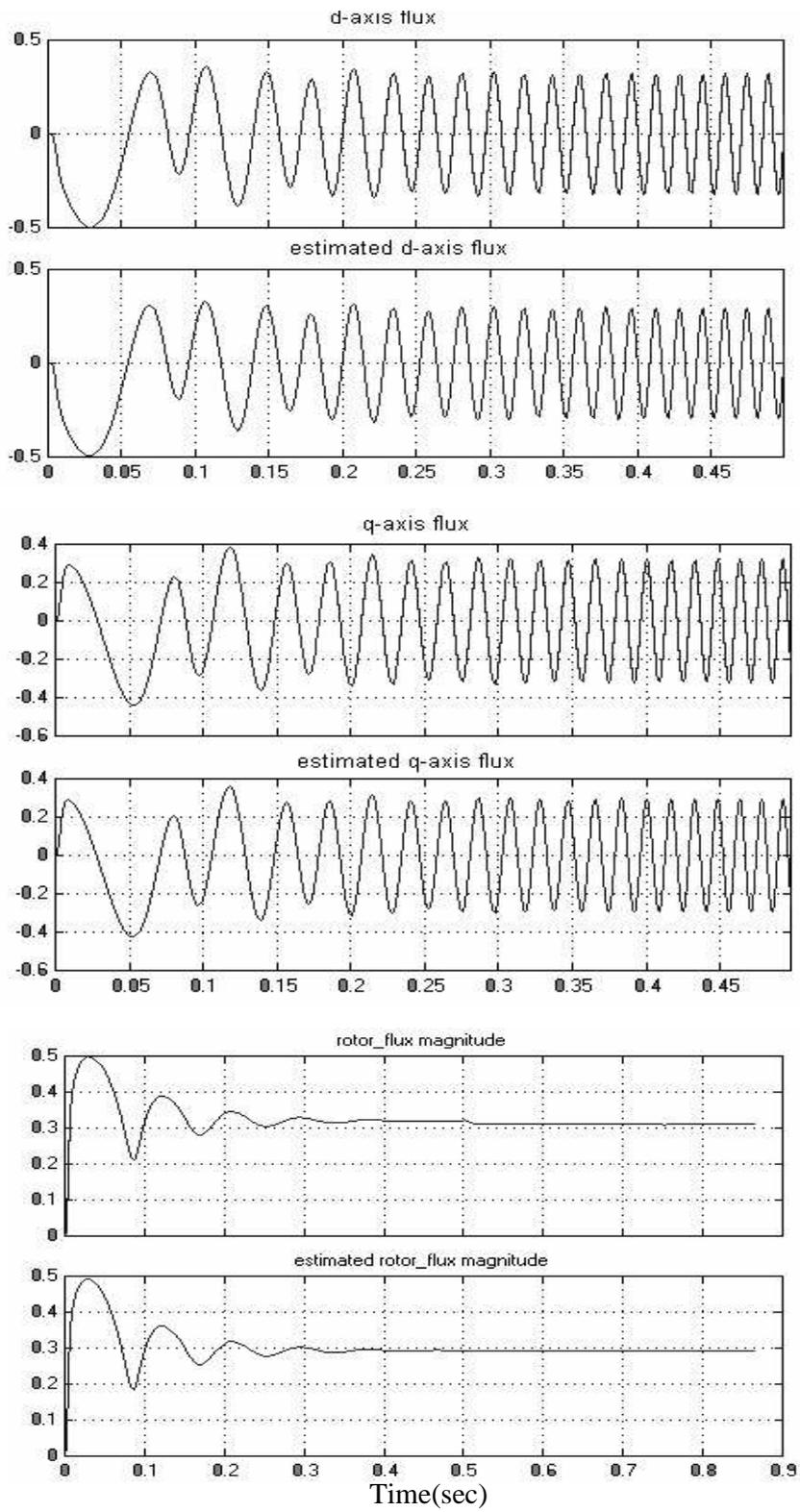


Fig.5.12- State III and IV (dq-axis Rotor Fluxes with their magnitude)

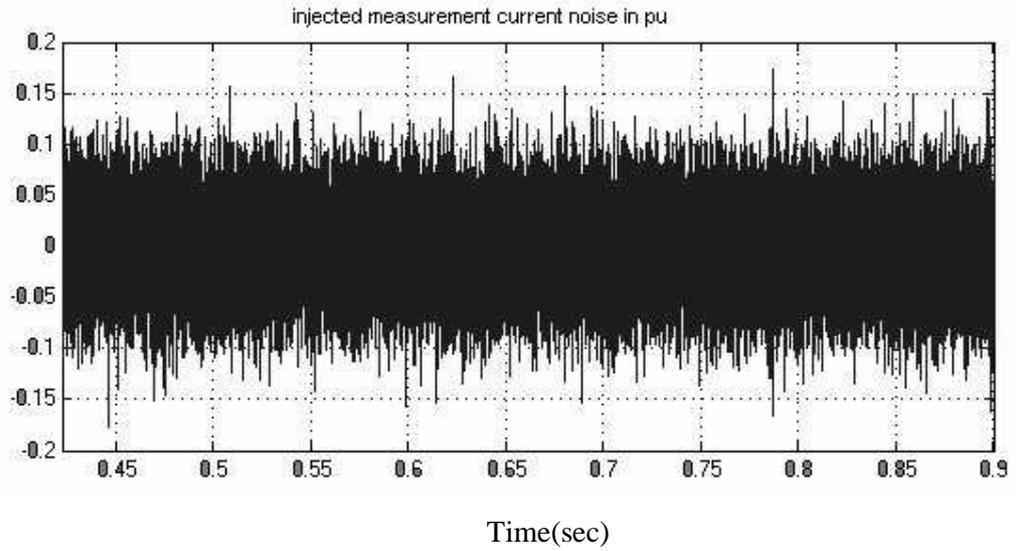


Fig.5.13- Injected noise to the stator currents in pu

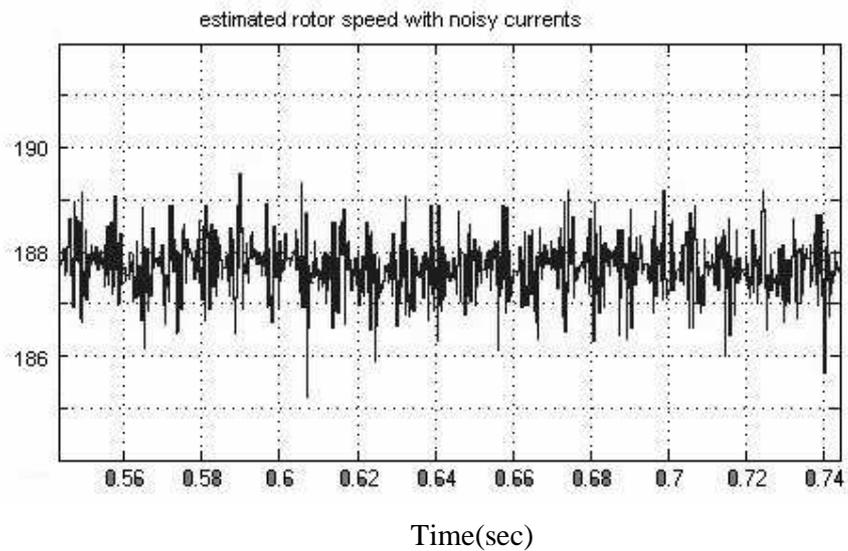


Fig.5.14- Estimated rotor speed with measured noisy current  
(in  $(P/2) * [rad/sec]$ )

In Fig. 5.13 the injected noise to the stator currents is shown. The noise is zero mean, white and Gaussian. The aim of the current injection is to observe the low pass filter characteristics of EKF. As shown in Fig. 5.14, the estimated speed is not affected too much from the injected noise. The speed estimation accuracy may be increased by increasing the measurement noise covariance under noisy conditions thus the system model will have more importance.

## 5.6 Unscented Kalman Filter

Unscented Kalman filter (UKF) is a novel estimation tool introduced by Julier and Uhlmann [34-35] to replace EKF in nonlinear filtering problems. As well-known, EKF is a simple solution derived by direct linearization of the state equation for extending the famous (linear) Kalman filter into nonlinear filtering area. Although it is straightforward and simple, EKF has well-known drawbacks[36]. These drawbacks include:

1. Instability due to linearization and erroneous parameters.
2. Costly calculation of Jacobian matrices.
3. Biasedness of its estimates.
4. Lack of analytical methods for suitable selection of model covariances.

UKF is proposed in order to overcome the first three of these disadvantages. The main advantage of UKF is that it does not need linearization in the computation of the state predictions and covariances. Due to this, its covariance and Kalman gain estimates are more accurate. This accurate gain, at the end, leads to better state estimates. In this study, UKF is introduced into the problem of speed and flux estimation of an induction motor. General simulation results are given and a brief comparison is made between speed estimation performances of UKF and EKF.

The filtering problem involved in this thesis is to find the best (in the sense of minimum mean square error (MMSE)) linear estimate of the state vector  $x_k$  of the induction machine which evolves according to the discrete-time nonlinear state transition equation

$$x_{k+1} = f(x_k, u_k) + w_k \quad (5.44)$$

where  $f(.,.)$  is the induction machine dynamics,  $x_k$  is the state of the induction machine at sampling instant  $k$ ,  $u_k$  is the known input to the induction machine at time  $k$  and  $w_k$  is the additive white process noise term representing modeling errors.

Also, it is assumed that we have a set of noisy measurements  $z_k$  which are related to the state vector of the induction machine by the linear relationship;

$$y_k = Cx_k + v_k \quad (5.45)$$

where  $C$  is the properly sized observation matrix and  $v_k$  is the white measurement noise related with the measuring device used. The additive white-noise vectors

$w_k$  and  $v_k$  are Gaussian and uncorrelated from each other with zero mean and covariances  $Q$  and  $R$ , respectively.

The state of the system is assumed to be unknown, and therefore, the aim of the estimation process is to find a MMSE estimate of the state  $\hat{x}_{k|k}$  which is given by

$$\hat{x}_{k|k} \stackrel{\Delta}{=} E\{x_k | Y^k\} \quad (5.46)$$

where  $Y^k \stackrel{\Delta}{=} \{y_1, y_2, \dots, y_k\}$  and  $E\{x|y\}$  denotes the expected value of the quantity  $x$ , given the information  $y$ . Also, traditionally, one calculates the error estimates given by the covariance matrix  $P_{k|k}$  defined as

$$P_{k|k} \stackrel{\Delta}{=} E\{[x_k - \hat{x}_{k|k}][x_k - \hat{x}_{k|k}]^T | Y^k\} \quad (5.47)$$

These direct definitions being too difficult to calculate, recursive forms are adopted for both the state and covariance estimates. The recursive update equations for them are given as;

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + L_{k+1} v_{k+1} \quad (5.48)$$

$$P_{k+1|k+1} = P_{k+1|k} - L_{k+1} P_{k+1|k}^v L_{k+1}^T \quad (5.49)$$

where the vectors  $\hat{x}_{k+1|k}$  (State Prediction),  $v_{k+1}$  (Innovation) and the matrices  $L_{k+1}$  (Kalman Gain),  $P_{k+1|k}$  (State Prediction Covariance), and  $P_{k+1|k}^v$  (Innovation Covariance) are dependent on the quantities  $\hat{x}_{k|k}$  and  $P_{k|k}$  with the following equations.

$$\hat{x}_{k+1|k} \stackrel{\Delta}{=} E\{f(x_k, u_k) | Y^k\} \quad (5.50)$$

$$P_{k+1|k} \stackrel{\Delta}{=} E\{[x_{k+1} - \hat{x}_{k+1|k}][x_{k+1} - \hat{x}_{k+1|k}]^T | Y^k\} \quad (5.51)$$

$$\hat{z}_{k+1|k} = C \hat{x}_{k+1|k} \quad (5.52)$$

$$v_{k+1} \stackrel{\Delta}{=} y_{k+1} - \hat{z}_{k+1|k} \quad (5.53)$$

$$P_{k+1|k}^v = C P_{k+1|k} C^T + R \quad (5.54)$$

$$K_{k+1} = P_{k+1|k}^{xy} (P_{k+1|k}^v)^{-1} \quad (5.55)$$

$$P_{k+1|k}^{xy} = P_{k+1|k} C^T \quad (5.56)$$

The quantities  $\hat{x}_{k+1|k}$  and  $P_{k+1|k}$ , which are called state prediction and prediction

covariance of the state, respectively. They are vital for the overall filter performance. (5.50) and (5.51) do not specify how these quantities are calculated. EKF assumes that errors in the state estimates are small enough to approximate (5.50) and (5.51) to their first order Taylor series. As a result,  $\hat{x}_{k+1|k}$  and  $P_{k+1|k}$  are calculated in EKF as follows;

$$\hat{x}_{k+1|k}^{\text{EKF}} = f(\hat{x}_{k|k}, u_k) \quad (5.57)$$

$$P_{k+1|k}^{\text{EKF}} = \nabla f_x P_{k|k} \nabla f_x^T + Q \quad (5.58)$$

where  $\nabla f_x$  denotes the Jacobian matrix of the function  $f$  with respect to the state  $x$ . This linearization in EKF frequently yields wrong results in the estimates of the covariance and thus the state. UKF solves the prediction problem by sampling the distribution of the state in a deterministic manner and then transforming each of the samples using the nonlinear state transition equation.

The  $n$ -dimensional random variable  $x_k$  with mean  $\hat{x}_{k|k}$  and covariance  $P_{k|k}$  is approximated by  $2n+1$  weighted samples or *sigma points* selected by the algorithm.

$$\chi_0(k|k) \stackrel{\Delta}{=} \hat{x}_{k|k} \quad W_0 \stackrel{\Delta}{=} \kappa / (n + \kappa) \quad (5.59)$$

$$\begin{aligned} \chi_i(k|k) &\stackrel{\Delta}{=} \hat{x}_{k|k} + (\sqrt{(n + \kappa)(P_{k|k} + Q)})_i \\ W_i &\stackrel{\Delta}{=} 1 / (2(n + \kappa)) \end{aligned} \quad (5.60)$$

$$\begin{aligned} \chi_{i+n}(k|k) &\stackrel{\Delta}{=} \hat{x}_{k|k} - (\sqrt{(n + \kappa)(P_{k|k} + Q)})_i \\ W_{i+n} &\stackrel{\Delta}{=} 1 / (2(n + \kappa)) \end{aligned} \quad (5.61)$$

for  $i=1, \dots, n$  where  $\kappa \in \mathfrak{R}$  is a free real number such that  $n + \kappa \neq 0$ ,  $(\sqrt{(n + \kappa)(P_{k|k} + Q)})_i$  is the  $i^{\text{th}}$  column of the matrix, square root of  $(n + \kappa)(P_{k|k} + Q)$ , and  $W_i$  is the weight associated with the  $i^{\text{th}}$  point. Given these set of samples, the prediction process is as;

1. Each sigma point is transformed through the process dynamics  $f$  ;

$$\chi_i(k+1|k) = f(\chi_i(k|k), u_k) \quad (5.62)$$

2. The state prediction is computed as;

$$\hat{x}_{k+1|k} = \sum_{i=0}^{2n} W_i \chi_i(k+1|k) \quad (5.63)$$

3. The prediction covariance is calculated as;

$$P_{k+1|k} = \sum_{i=0}^{2n} W_i [\chi_i(k+1|k) - \hat{x}_{k+1|k}] \cdot [\chi_i(k+1|k) - \hat{x}_{k+1|k}]^T \quad (5.64)$$

The equations (5.63) and (5.64) replace (5.50) and (5.51). The other UKF operations are the same as (5.52) to (5.56). Note that, operations in the new set of equations composed by (5.63), (5.64), (5.52) - (5.56) together with measurement updates given in (5.48) and (5.49) use only standard vector and matrix operations and need no approximations for both derivative and Jacobian. Also, the order of calculation is the same as that of EKF. In the next section, a detailed induction machine model used in the implementation of UKF is given.

### 5.6.1 Simulation Results

A number of simulations were carried out to verify the performance of the state estimation, particularly of the speed estimation with UKF. In Fig.5.15 – Fig.5.21, the state estimation performance of UKF is simulated and in Figs 5.22, 5.23, accuracies obtained from EKF and UKF are compared for the speed estimation. Fig.5.15 shows the actual state variables of the motor; stator currents, rotor fluxes and rotor speed at no-load in a high speed reversal scheme. Fig.5.16 shows corresponding estimated state variables with UKF under the same conditions. There are almost no differences between the actual and the estimated variables.

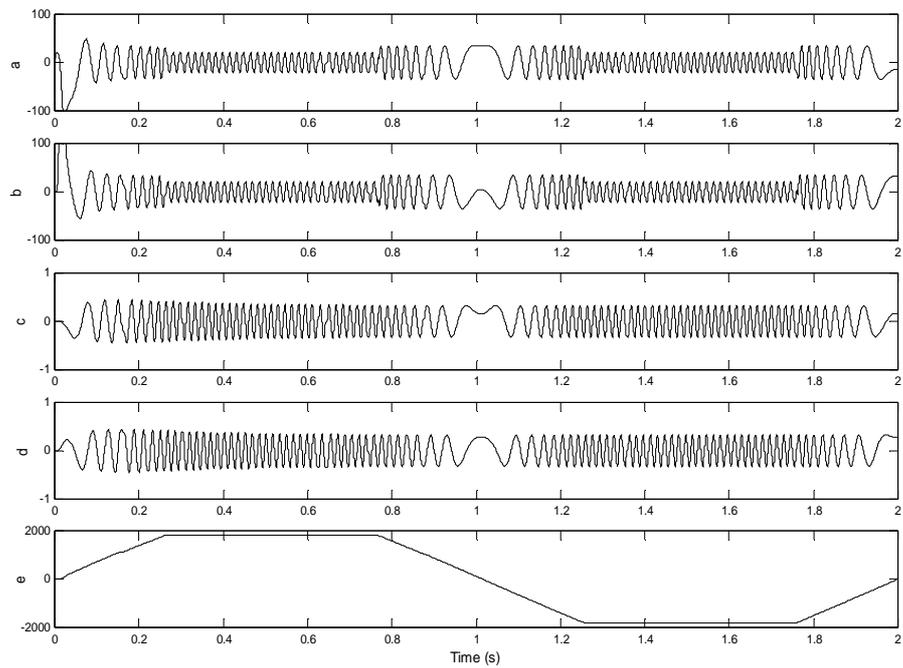


Fig.5.15- Induction motor actual states at no load four quadrant high speed reversal (a-b) d-q axis stator currents, (c-d) d-q axis rotor fluxes, (e) rotor speed.

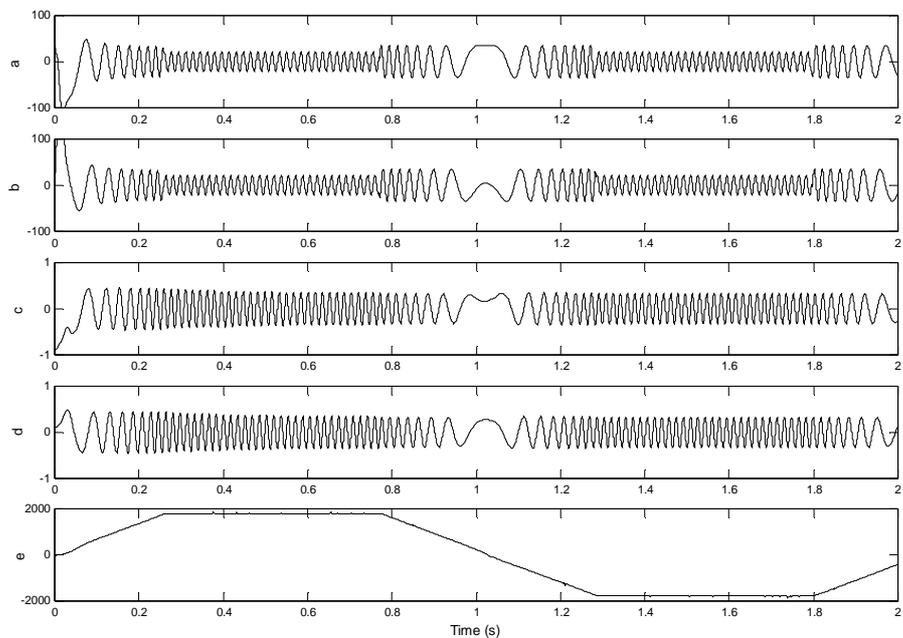


Fig. 5.16. Induction motor estimated states with UKF at no load four quadrant high speed reversal (a-b) estimated d-q axis stator currents, (c-d) estimated d-q axis rotor fluxes, (e) estimated rotor speed.

Fig.5.17 and Fig.5.18 illustrates magnified estimated speed waveforms at no-load in four quadrant high speed and low speed reversal schemes respectively. Both the high speed and low speed estimated waveforms confirm that UKF's performance is quite good in speed estimation for all quadrants without causing instability.

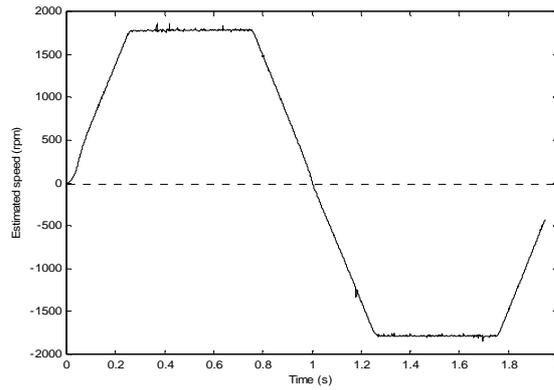


Fig. 5.17. Induction motor estimated speed at no-load four quadrant high speed reversal (*in rpm*).

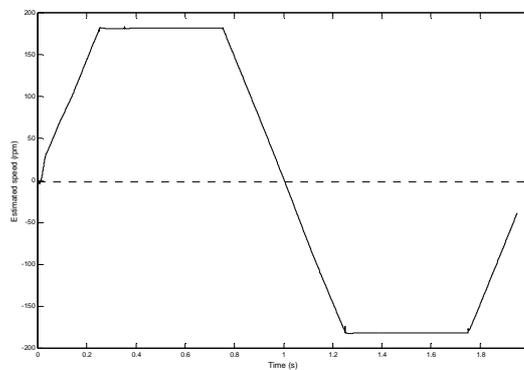


Fig. 5.18. Induction motor estimated speed at no-load four quadrant low speed reversal (*in rpm*)

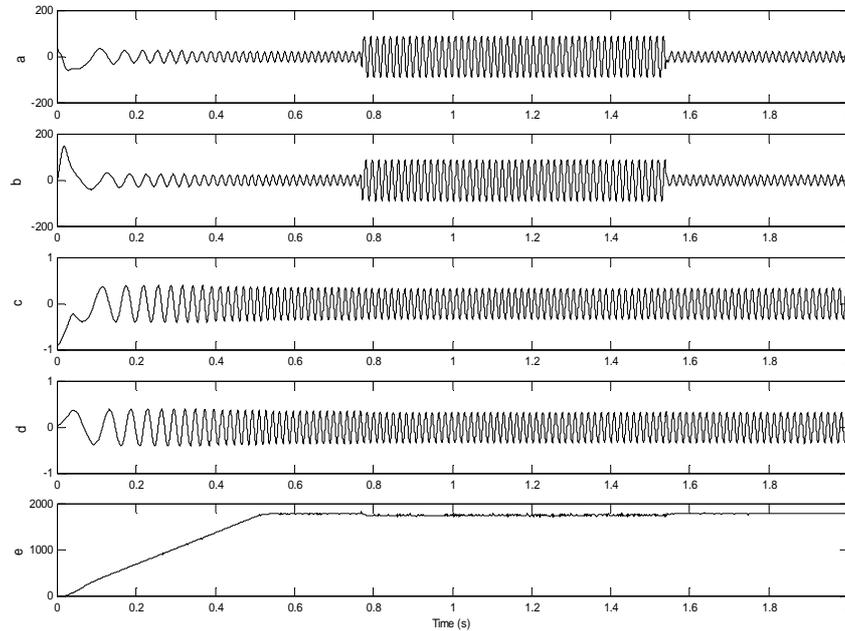


Fig. 5.19- Induction motor estimated states at %100 rated torque and speed (a-b) estimated d-q axis stator currents, (c-d) estimated d-q axis rotor fluxes, (e) estimated rotor speed (load torque applied between 0.75s-1.5s)

In Fig.5.19, estimated state variables of the induction motor are shown under %100 rated load torque and %100 rated speed conditions. Note that mechanical load is applied to the motor between only 0.75s and 1.5s. In addition to high performance at no-load, UKF gives quite satisfactory results under full-load condition. In Fig.5.20, and Fig.5.21, actual and estimated speed characteristics are given on top of each other for %100 and %10 rated torque and speed case. In the transient part of the waveforms, there appears a difference between the estimated and actual values which is the result of the fact that, in induction motor model, the speed is considered as a constant parameter and corrected only in the measurement updates of the UKF. In simulation tests, we also noticed that there usually exists a small steady-state error between the estimated and actual speed values but that seems to be at negligible levels.

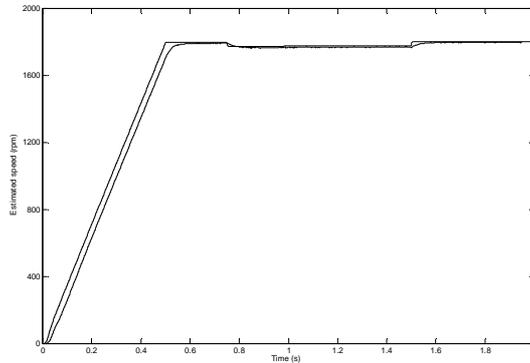


Fig. 5.20- Induction motor estimated speed at %100 rated torque and speed (load torque applied between 0.75s-1.5s)

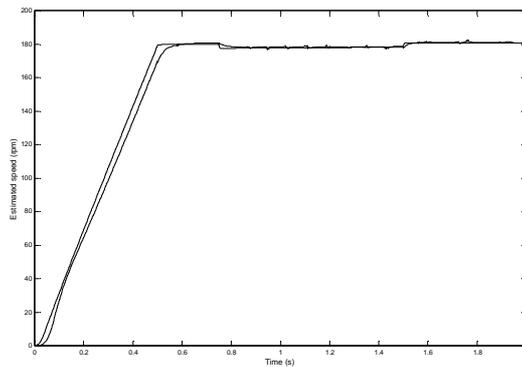


Fig. 5.21- Induction motor estimated speed at %10 rated torque and speed (load torque applied between 0.75s-1.5s).

In Fig.5.22 and 5.23, the speed estimation performances of UKF and EKF with identical parameters are compared at %100 rated torque and speed. Simulations of Fig.5.22 and 5.23 were carried out for different covariance values. Covariance values were selected so that the steady state performance in Fig.5.22 and the transient performance in Fig.5.23 is optimized. It is observed from the figures that, although the performances of the EKF and UKF are close to each other, UKF reduces the transient- and steady-state speed estimation errors by up to 10 rpm under rated conditions.

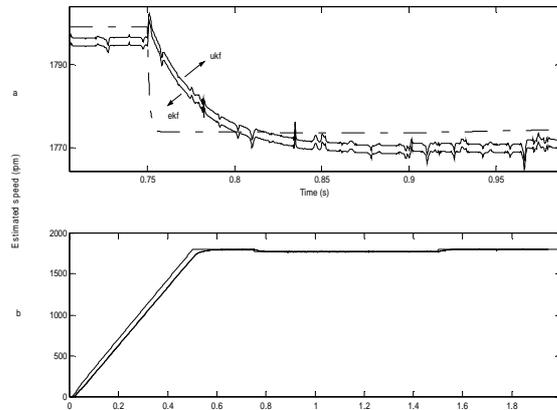


Fig. 5.22- (b) Induction motor estimated speed optimized for steady state performance at %100 rated torque and speed using EKF and UKF. (a) graphics in (b) zoomed at the mechanical loading initiation. (load torque applied between 0.75s-1.5s)

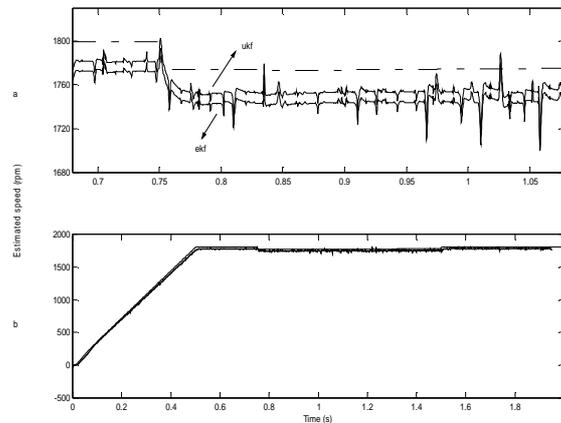


Fig. 5.23- (b) Induction motor estimated speed optimized for transient performance at %100 rated torque and speed using EKF and UKF. (a) graphics in (b) zoomed at the mechanical loading initiation. (load torque applied between 0.75s-1.5s)

It has been shown that UKF is as good as EKF at least in state observation, and it yields even slightly better speed estimation performance than EKF. This result encourages further study in the area to obtain better state estimation performances for nonlinear systems to overcome the well-known defects of EKF and other traditional nonlinear filtering techniques.

### 5.6.2 Experimental Results

In addition to computer simulations of the discussed estimators, EKF and UKF, the expected results are also confirmed with the experimental results. While obtaining the experimental results, the real time stator voltages and currents are processed in Matlab with the associated EKF and UKF programs.

Fig.5.24 shows estimations of states I&II (dq axis stator currents) made by EKF and the actual states I&II measured from the experimental setup. It may easily be noticed that the estimated states are quite close to the measured ones.

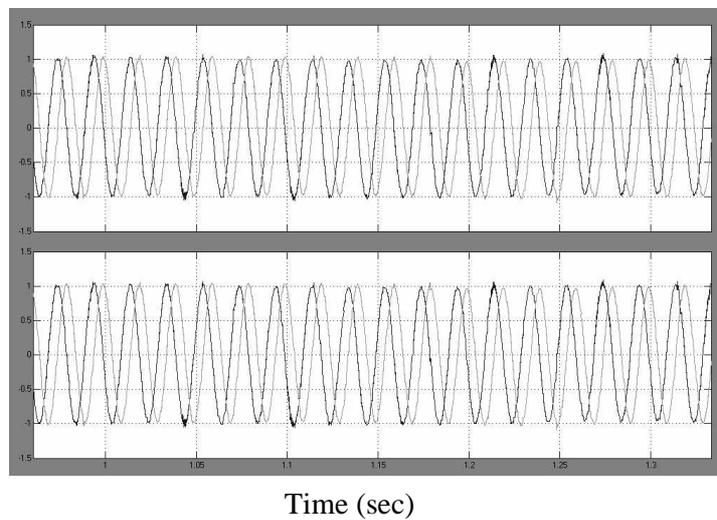


Fig. 5.24- The estimated states I and II (upper one\_ dq axis stator currents) by EKF and the measured states I&II (lower one)

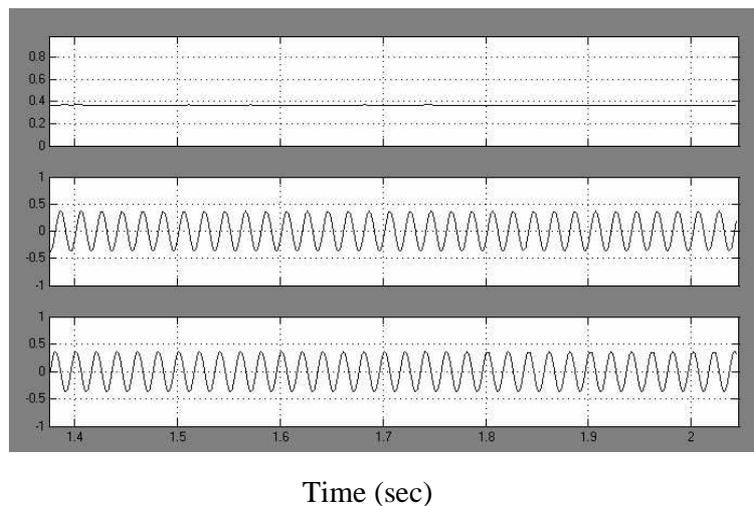


Fig. 5.25- The estimated states II and III by EKF (lower ones\_ dq axis rotor fluxes) and the magnitude of the rotor flux (upper one)

Fig.5.25 shows the estimated dq axis rotor fluxes in stationary reference frame. The magnitude of the rotor flux justifies that the estimated dq components of the rotor flux do not involve dc offset and orthogonal to each other.

In order to examine the rotor speed (state V) estimation performance of EKF experimentally under varying speed conditions, a trapezoidal speed reference command is embedded into the DSP code. As shown in Fig5.26, EKF rotor speed estimation successfully tracks the trapezoidal path.

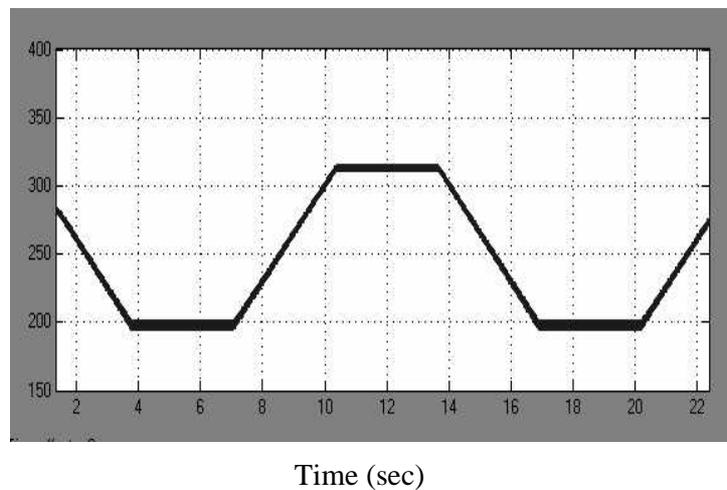
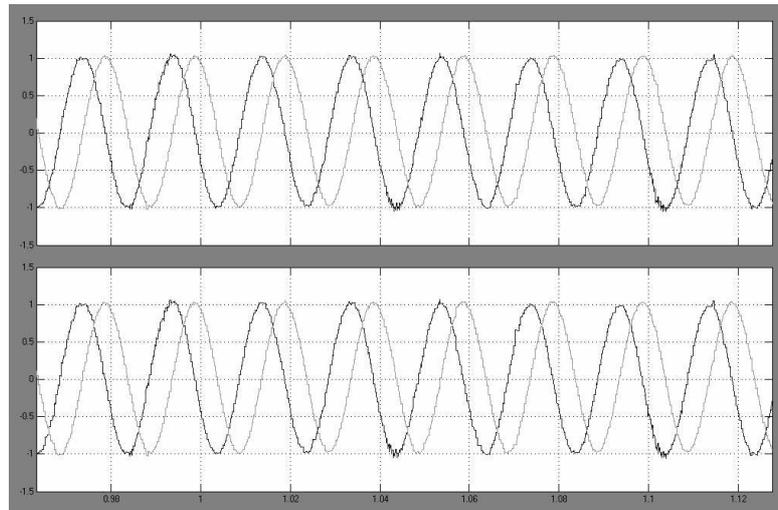


Fig.5.26- Rotor speed tracking performance of EKF obtained experimentally

The same states of the induction motor model estimated by EKF are also estimated by UKF. Fig.5.27 shows estimations of states I&II (dq axis stator currents) made by UKF and the actual states I&II measured from the experimental setup. One may easily notice that the estimated states are quite close to the measured ones. Fig.5.28 shows the estimated dq axis rotor fluxes in stationary reference frame by UKF. The magnitude of the rotor flux justifies that the estimated dq components of the rotor flux are estimated accurately.



Time (sec)

Fig. 5.27- The estimated states I and II (upper one\_dq axis stator currents) by EKF and the measured states I and II (lower one)

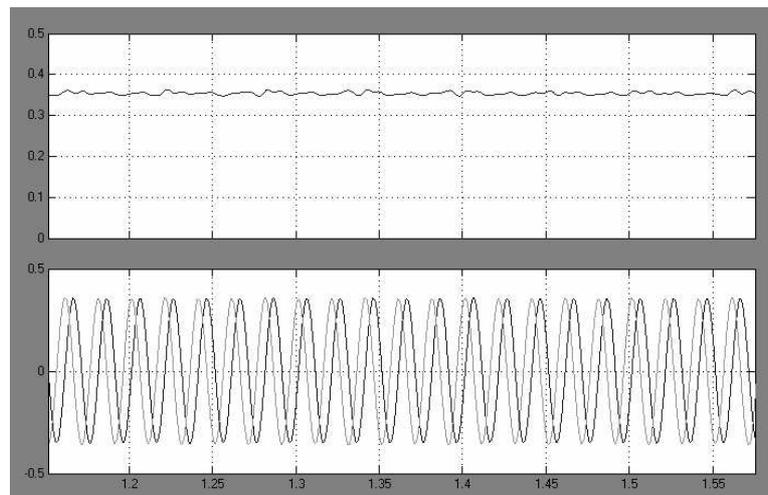


Fig. 5.28- The estimated states II and III by UKF (lower one\_dq axis rotor fluxes) and the magnitude of the rotor flux (upper one)

In order to compare both types of the observers, EKF and UKF, the covariance matrices regarding to both types have been initialized with the same entries under the same operating conditions. The estimated rotor speed waveforms, when plotted together as shown in Fig.5.29, confirm that the estimation accuracy of UKF is superior over EKF as claimed before when discussing the simulation results related to both observer design techniques. The simulation results were shown in Fig.5.22 and Fig 5.23.

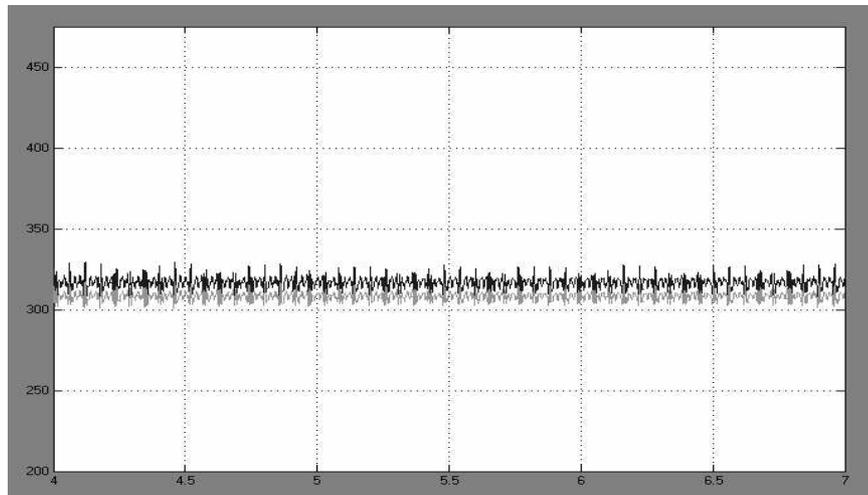


Fig.5.29. Rotor speed waveforms obtained experimentally by UKF (darker) and EKF (lighter) under the same experimental conditions (measured speed 314 rad/sec)

As expected from simulations in the Fig.5.22 and 5.23, the speed estimation accuracy of UKF is better than EKF under the same experimental conditions. The measured speed from the motor shaft is 314 rad/sec. The mean of the state estimation error in UKF is 2.65 rad/sec at steady state, and that in EKF is 5.8 rad/sec. This result shows that the estimates of EKF have serious bias problems compared to UKF. As discussed earlier, the derivative free algorithm of UKF without a linearity approximation contributes its estimates positively. Furthermore, the noise sampling feature of UKF is more realistic approach instead of assuming the noise directly as Gaussian. This property also makes its estimation accuracy better than EKF.

## **CHAPTER 6**

### **MODEL REFERENCE ADAPTIVE SYSTEMS**

#### **6.1 Adaptive Control**

Adaptive control may be defined in many ways. A possible definition of adaptive control is “a system that adapts itself to changes in the process”. Another definition, that is often used but probably too vague to be useful, is “a system which is designed from an adaptive point of view”. A more useful one is “a system that consists of a primary feedback that takes care of process signal variations and a secondary feedback that deals with process state changes. In this definition, the primary feedback is used as in non-adaptive control, and the secondary feedback makes the system adaptive. From this definition it is clear that process state variations give rise to adaptation of the system. The aim of reacting to state changes is to attempt to maintain a high system performance, even if the process states are unknown or varying [57]. In the literature there exist several adaptive control techniques. In this thesis Model Reference Adaptive System is applied to induction motor drive as a state observer.

#### **6.2 Model Reference Adaptive Systems**

Model Reference Adaptive System (MRAS) is one of the most popular adaptive control method used in motor control applications for tracking and observing system parameters and states [19-21],[15,18,22-25,58-64]. There exist a number of different model reference adaptive control techniques such as parallel model, series model, direct model and indirect model etc. MRAS used in this thesis is parallel

model MRAS that compares both the outputs of a reference model and adaptive model, and processes the error between these two according to the appropriate adaptive laws that do not deteriorate the stability requirements of the applied system. A generalized parallel MRAS scheme is shown in Fig.6.1 where the primary controller is used to obtain suitable closed-loop behavior, as in non-adaptive control schemes. However, because the process parameters are unknown or may vary with time, a fixed parameter setting for the primary controller, such that the closed-loop behavior is acceptable under all circumstances, cannot be found. In the MRAS technique, the desired process response to a command signal is specified by means of a parametrically defined *reference model*. An *adaptation mechanism* keeps track of the process output  $y_p$  and the model output  $y_m$  and calculates a suitable parameter setting such that difference between these outputs tends to zero. In addition to process output  $y_p$ , the process  $x_p$ , if available, and the process input  $u$  or the reference signal may be used by adaptation mechanism.

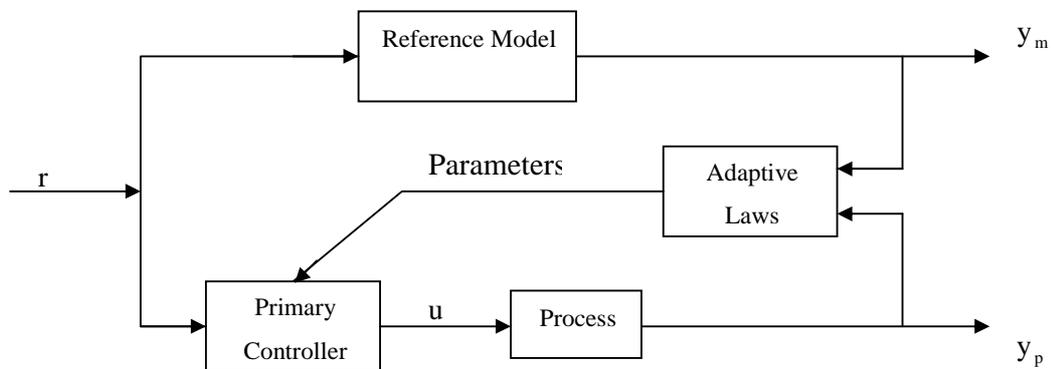


Fig.6.1- General parallel MRAS scheme

An important issue in MRAS is the design of adaptive laws. The first examples of adaptive law designs made use of sensitivity models, and later the stability theory of Lyapunov, and Popov's hyperstability theory, served as standard design methods, yielding a guaranteed stable adaptive system. (see *Appendix A*)

### 6.3 Introduction to MRAS practice in motor control applications

In a MRAS system, some state variables,  $x_d, x_q$  (e.g. back e.m.f components ( $e_{md}, e_{mq}$ ) reactive power components ( $q_{md}, q_{mq}$ ), rotor flux components ( $\Psi_{rd}, \Psi_{rq}$ ) etc.) of the induction machine, which can be obtained as sensed variables such as stator voltage and currents, are estimated in reference model. They are then compared with state-variables  $\hat{x}_d$  and  $\hat{x}_q$  estimated by using adaptive model. The difference between these state-variables is then used in adaptation mechanism, which outputs the estimated value of the rotor speed ( $\hat{\omega}_r$ ) and adjusts the adaptive model until satisfactory performance is achieved. Such a scheme is shown in Fig.6.2 where compact space-vector notation is used. However, Fig.6.3 corresponds to an actual implementation, and here components of the space vector are shown in detail.

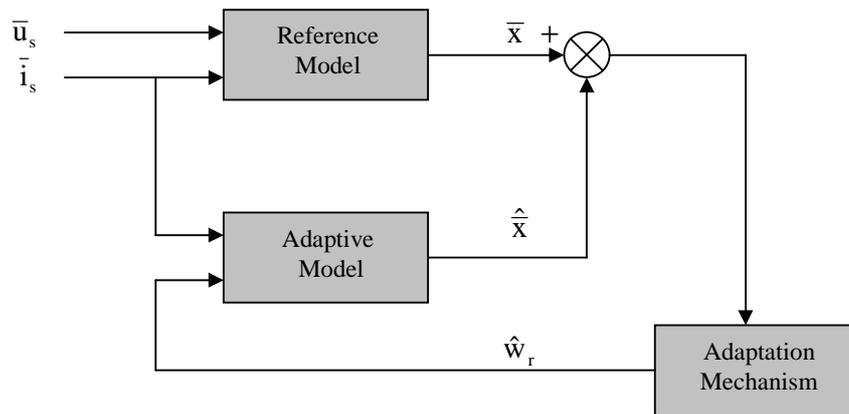


Fig.6.2- Generalized Model Reference Adaptive System

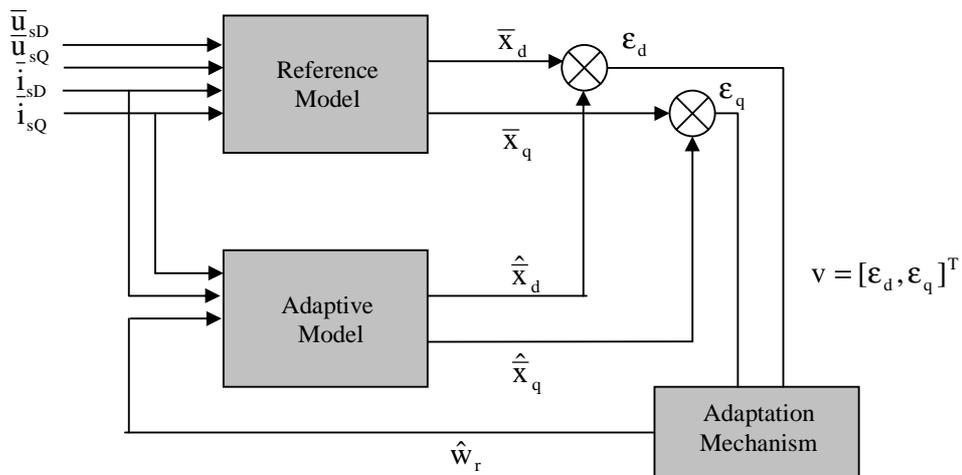


Fig.6.3- MRAS based speed estimator scheme using space vector

The appropriate adaptation mechanism can be derived by using Popov's criterion of hyperstability. This results in a stable and quick response system, where the differences between the state variables of the reference model and adaptive model are manipulated into speed tuning signal ( $\varepsilon$ ), which is then input to a PI-type of controller that outputs the estimated rotor speed. Two of the schemes will be discussed in the following sections: reactive power and back e.m.f errors are used as speed tuning signals. In these expressions  $\bar{i}_s$  and  $\bar{u}_s$  denote the stator voltage and stator current space vectors respectively in the stationary frame  $\bar{e}_m$  denotes the back emf space vector also in stationary reference frame as  $\Delta\bar{e}_m = \bar{e}_m - \hat{\bar{e}}_m$ . The symbol  $\hat{\phantom{x}}$  denotes the quantities estimated by the adaptive model. In addition to these classical MRAS schemes, artificial intelligence techniques assisted MRAS speed estimators are also discussed in the literature. They contain neither any mathematical adaptive model, nor any adaptation mechanism incorporated into the tuning of appropriate artificial intelligence based network (which can be a neural network, a fuzzy-neural network, etc.) [23-24, 60].

To improve the performance of the observers described in this section, various practical techniques are also discussed which avoid use of pure integrators. Pure integrators lead drift and initial condition problems in digital applications, so recent speed sensorless algorithms tend to avoid pure integrators. Most of the traditional vector control algorithms use low-pass filters instead of pure integrators, although they also cause serious problems at low speed range. Recent MRAS algorithms mentioned in this thesis avoid both pure integrators and low-pass filters. Reactive power scheme described below is robust to both stator and rotor resistance variations, and can even be applied at very low speeds (but not zero speed). Both of the observers (reactive power and back emf schemes) described below use monitored stator currents and stator voltages. In a voltage-source inverted-fed drive, however, it is not necessary to monitor the dc link voltage and the stator voltages since the latter can be reconstructed by using the inverter switching states.

#### 6.4. Application of Popov's hyperstability theorem and integral inequality

This part contains a short description of the selection of the appropriate adaptation mechanism, proves why there is a PI controller in the scheme described in MRAS schemes used in this thesis, and also shows the form of speed tuning signal to be used.

In general, a model referenced adaptive speed estimator system can be represented by an equivalent non-linear feedback system which comprises a feed-forward time-invariant linear subsystem as well as a feedback non-linear time-varying subsystem. This is shown in Fig. 6. 4.

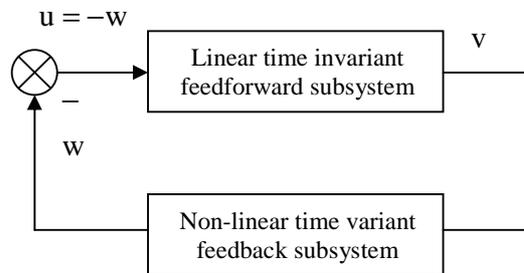


Fig.6.4- Equivalent non-linear feedback system

In Fig.6.4 the input to the linear time-invariant system is  $u$  (which contains the stator voltage and currents), its output is  $v$ , which is the speed-tuning signal  $v = [\varepsilon_d, \varepsilon_q]^T$ . The output of the non-linear time invariant system is  $w$ , and  $u = -w$ . The rotor speed estimation algorithm (adaptation mechanism) is chosen according to Popov's hyperstability theory, whereby the transfer function matrix of the linear time invariant system must be strictly positive real and the non-linear time-varying feedback system satisfies Popov's integral inequality, according to which  $\int v^T w dt \geq 0$  in the time interval  $[0, t_1]$  (see appendix B). Thus to obtain the adaptation mechanism, first the transfer function  $F(s)$  of the linear time-invariant feed-forward subsystem has to be obtained. It can be shown by lengthy calculations that in both of the schemes described in the following part this is strictly positive real. A possible proof uses the state-variable form of the error equation,  $dv/dt = Av-w$ , which is obtained by subtracting the state variable equations of the adjustable model from the state-variable equations of the reference model (where  $A$  is the state matrix).

The feed-forward path transfer matrix of the linear time-invariant subsystem shown in Fig.6.4 is  $F(s) = [sI-A]^{-1}$ , where  $I$  is an identity matrix. It follows from the derivation of the error state equation that  $w = [\hat{w}_r - w_r][-\hat{x}_q \ x_d]^T$  (where  $\hat{x}_d$  and  $\hat{x}_q$  are estimated by adaptive model), thus  $w$  is substituted into Popov's integral inequality,  $\int v^T w dt \geq 0$ , it can be shown that this inequality can be satisfied by letting  $\hat{w}_r = (K_p + K_i/p)\epsilon$ . In this equation  $1/p$  represents an integrator and  $\epsilon$  is the appropriate speed-tuning signal. In general, the state variables in the reference and adaptive models are  $x_d, x_q$  and  $\hat{x}_d, \hat{x}_q$ , respectively. Speed tuning signal is  $\text{Im}(\bar{x} \hat{x}^*)$ , where the asterisk denotes the complex conjugate. Speed-tuning signals are obtained from  $\text{Im}(\bar{e}_m \hat{e}_m^*)$  and  $\text{Im}(\bar{q}_m \hat{q}_m^*)$  where  $e_m$  and  $q_m$  represent back emf and reactive power, respectively. It can be seen that when a specific state-variable is used (on the outputs of the reference and adaptive models), then a corresponding speed tuning signal of a specific form is obtained by Popov's integral inequality. From the previous discussions it is noted that when the estimated rotor-speed with the adaptive model changes in such a way that the difference between the output of the reference model and the adaptive model is zero, then the estimated rotor speed is equal to the actual rotor speed. The error signal actuates the rotor-speed identification algorithm, which makes this error converge asymptotically to zero. The physical reason for the integrator (in PI controller) is that this ensures that the error converges asymptotically to zero.

In recent years several MRAS schemes are studied for vector control of ac drives without speed measuring sensors. Most of these schemes have low speed problems due to the low-pass filters or pure integrators. These schemes obtain speed tuning signal from state variables,  $x_d, x_q$  (e.g. active power components ( $p_d, p_q$ ), rotor flux components ( $\Psi_{rd}, \Psi_{rq}$ ), torque components ( $te_d, te_q$ ), voltage components ( $v_d, v_q$ ) etc.). In addition to these schemes, recently proposed schemes using back emf and reactive power enhanced the performance of the MRAS solutions excluding pure integrators in their algorithms [26]. Also reactive power MRAS model is truly robust to stator resistance changes. The details of these schemes will be given in the next section.

## 6.5 Back emf MRAS Scheme

In this MRAS scheme the back emf ( $e_m$ ) is used as speed tuning signal. When the back emf is used then the problems associated with the pure integrators in the reference model disappear, since in this case the reference model does not contain any integrator. Equations for an induction motor in the stationary frame can be expressed as:

$$V_s = R_s i_s + \sigma L_s \frac{di_s}{dt} + e_m \quad (6.1)$$

$$\frac{di_m}{dt} = w_r \otimes i_m - \frac{1}{T_r} i_m + \frac{1}{T_r} i_s \quad (6.2)$$

where  $w_r$  is a vector whose magnitude  $w_r$  is rotor electrical angular velocity, and whose direction is determined according to right hand system of coordinates as shown in Fig.6.5 “ $\otimes$ ” denotes the cross product of vectors respectively.

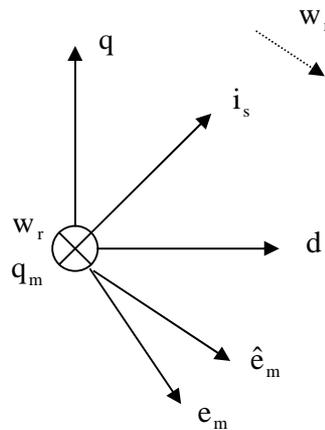


Fig.6.5- Coordinates in stationary reference frame

From (6.1) and (6.2),  $e_m$  and structure of MRAS can be derived as follows:

$$e_m = V_s - \left( R_s i_s + \sigma L_s \frac{di_s}{dt} \right) \quad (6.3)$$

$$e_m = L'_m \frac{di_m}{dt} \quad (6.4)$$

$$= L'_m \left( w_r \otimes i_m - \frac{1}{T_r} i_m + \frac{1}{T_r} i_s \right) \quad (6.5)$$

If we rewrite the equations above for the direct and quadrature-axis back emf in the following form:

$$\begin{aligned} e_{md} &= L_m \frac{di_{md}}{dt} = \frac{L_m}{L_r} \frac{d\psi_{rd}}{dt} \\ &= V_{sd} - \left( R_s i_{ds} + \sigma L_s \frac{di_{sd}}{dt} \right) \end{aligned} \quad (6.6)$$

$$\begin{aligned} e_{mq} &= L_m \frac{di_{mq}}{dt} = \frac{L_m}{L_r} \frac{d\psi_{rq}}{dt} \\ &= V_{sq} - \left( R_s i_{sq} + \sigma L_s \frac{di_{sq}}{dt} \right) \end{aligned} \quad (6.7)$$

If we use the counterelectromotive force (emf) vector  $e_m$  instead of rotor flux vector which was used in the previous MRAS schemes [8] for speed identification, then a new MRAS system is obtained. Fig.5.6 illustrates the new structure of the new MRAS for speed estimation. Two independent observers are configured to estimate the components of the counter-EMF vector, one based on (6.6) and the other based on (6.5) and (6.7). The observer based on (6.6) can be regarded as a reference model of the induction motor since (6.6) does not involve the quantity  $w_r$ , and the one based on (6.5) and (6.7) can be regarded as adjustable model because (6.5) and (6.7) do involve  $w_r$ . The error between the outputs of the two observers is then used to drive a suitable adaptation mechanism which generates the estimate  $\hat{w}_r$  for the adjustable model.

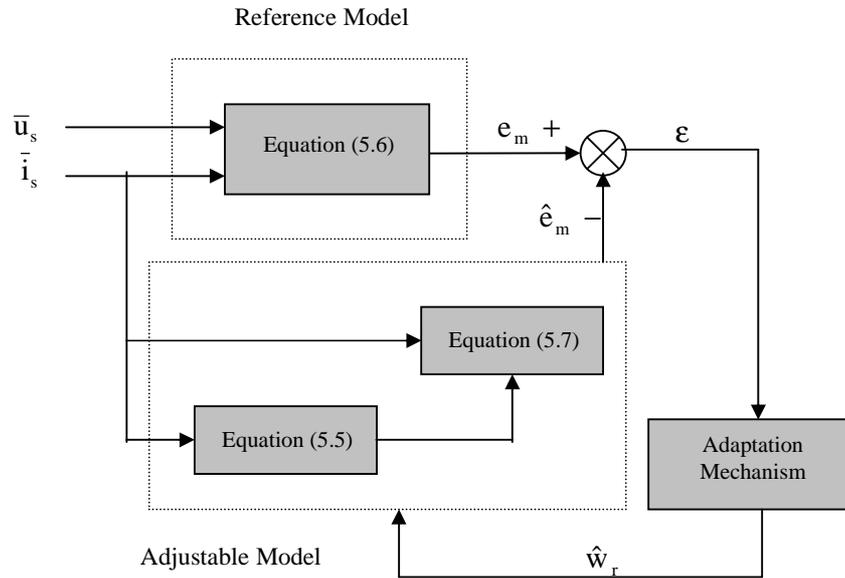


Fig.6.6- Structure of the MRAS system for speed estimation

When the scheme shown in Fig. 6.5 is employed in a speed-sensorless vector controlled drive, since the reference model does not contain pure integration a satisfactory performance can be obtained even at low speeds if an accurate value of the stator resistance is used. However, the stator resistance varies with temperature, and this affects the stability performance of the speed observer, especially at low speeds. A MRAS scheme which is intensive to stator resistance variation can be obtained by using such a speed-tuning signal, which is obtained from a quantity which does not contain the stator resistance. This is discussed later. On the other hand, some of the applications use parameter estimation or parameter tracking algorithms to compensate the error caused by parameter deviation. Thus, in literature there exist several on-line parameter estimation algorithms at real time. Parameter deviation effects on MRAS algorithms will also be discussed in this thesis.

### 6.5.1 Adaptation Mechanisms and Stability of MRAS

It is important to ensure that the system will be stable and the estimated quantity will converge to the actual value for the adaptation mechanism of MRAS algorithms. In general  $w_r$  is a variable; thus the models are linear time-varying systems. For the purpose of deriving an adaptation mechanism, however, it is valid to initially treat  $w_r$  as a constant parameter of the models. By differentiating both sides of (6.7), we get

$$\frac{de_m}{dt} = w_r \otimes e_m - \frac{1}{T_r} e_m + \frac{L_m}{T_r} \frac{di_s}{dt} \quad (6.8)$$

Here, letting  $\varepsilon = e_m - \hat{e}_m$ , and subtracting (6.8) for the adjustable model from (6.8) for the reference model, we obtain the following state error equation:

$$\begin{aligned} \frac{d\varepsilon}{dt} &= w_r \otimes \varepsilon - \frac{1}{T_r} \varepsilon - (\hat{w}_r - w_r) \otimes \hat{e}_m \\ &= A\varepsilon - W \end{aligned} \quad (6.9)$$

where

$$A = \begin{bmatrix} -\frac{1}{T_r} & -w_r \\ w_r & -\frac{1}{T_r} \end{bmatrix} = -\frac{1}{T_r} I + w_r J,$$

$$W = (\hat{w}_r - w_r) \otimes \hat{e}_m = (\hat{w}_r - w_r) \begin{bmatrix} -\hat{e}_{mq} \\ \hat{e}_{md} \end{bmatrix} \text{ and}$$

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

Since  $\hat{w}_r$  is produced by adaptation mechanism, (6.9) describes a nonlinear feedback system as shown in Fig.6.7. Hyperstability requires that the linear time-invariant forward-path transfer matrix be strictly positive and real, and that the nonlinear feedback including the adaptation mechanism satisfies Popov's criterion for hyperstability.

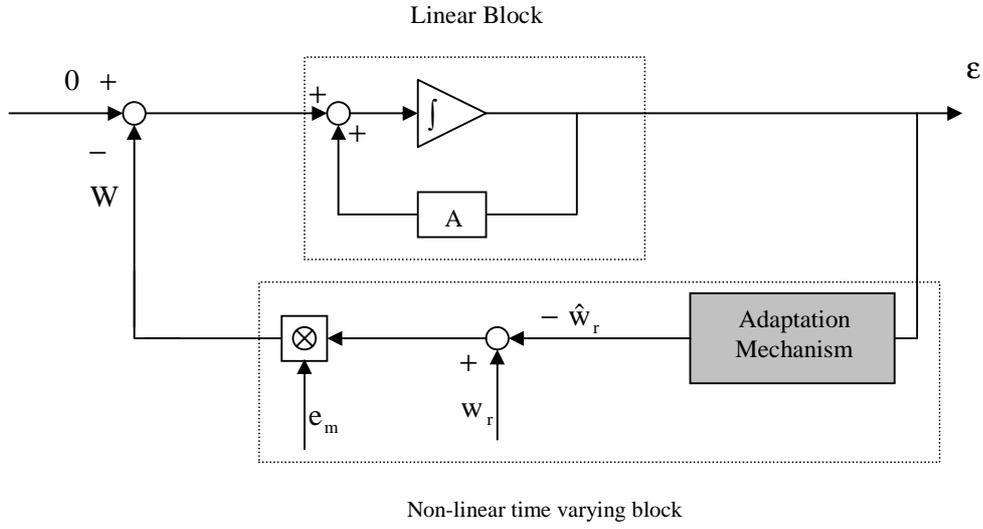


Fig.6.7- Equivalent nonlinear feedback system of MRAS

Popov's criterion requires that

$$\int_0^{t_1} \boldsymbol{\varepsilon} \bullet \mathbf{W} dt \geq -\gamma_0^2 \quad \text{for all } t_1 \geq 0 \quad (6.10)$$

where  $\gamma_0^2$  is a real positive constant and  $\bullet$  is dot product. Here, letting

$$\hat{\mathbf{w}}_r = \left( \mathbf{K}_p + \frac{\mathbf{K}_i}{p} \right) (\hat{\mathbf{e}}_m \otimes \boldsymbol{\varepsilon}) \quad (6.11)$$

and substituting for  $\mathbf{W}$  in inequality (6.10), (6.10) becomes

$$\begin{aligned} \int_0^{t_1} \boldsymbol{\varepsilon} \bullet \mathbf{W} dt &= \int_0^{t_1} \boldsymbol{\varepsilon} \bullet \{ (\hat{\mathbf{w}}_r - \mathbf{w}_r) \otimes \hat{\mathbf{e}}_m \} dt \\ &= \int_0^{t_1} (\boldsymbol{\varepsilon} \otimes \hat{\mathbf{e}}_m) \bullet \left( \mathbf{w}_r - \left( \mathbf{K}_p + \frac{\mathbf{K}_i}{p} \right) (\hat{\mathbf{e}}_m \otimes \boldsymbol{\varepsilon}) \right) dt \\ &\geq -\gamma_0^2 \end{aligned} \quad (6.12)$$

Using the following well known inequality:

$$\int_0^{t_1} \frac{d}{dt} f(t) \cdot f(t) dt \geq -\frac{1}{2} f^2(0) \quad (6.13)$$

it can be shown that inequality (6.12) is satisfied.

## 6.6. Reactive Power MRAS Scheme

In the previous part, back emf is used as tuning signal and the performance of the MRAS is proved to be perfect in the simulation. Since the reference model does not require pure integration, this system can achieve good performance even at low speeds, as long as the value of stator resistor is known precisely. The stator resistance, however, varies with the temperature of the stator. The stator resistance thermal variations affect the performance and stability of MRAS speed estimator, especially at low speeds as shown later in this thesis. Therefore, a speed identification system with low sensitivity to the stator resistance variations is necessary for applications of low-speed drives. Here, another approach to speed identification which is completely robust to stator resistance variations is proposed. This scheme can be represented in two different ways whose basics are the same.

First let us define a new quantity  $q_m$  as the cross product of the counter EMF vector  $e_m$  and the stator current vector  $i_s$ . That is,

$$q_m \triangleq i_s \otimes e_m \quad (6.14)$$

$q_m$  is a vector, whose direction is shown in Fig.4, and whose magnitude  $q_m$  represents the instantaneous reactive power maintaining the magnetizing current. Substituting the (6.6) and (6.7) for  $e_m$  in (6.14) noting that  $i_s \otimes i_s = 0$ , we have

$$q_m = i_s \otimes \left( v_s - \sigma L_s \frac{di}{dt} \right) \quad (6.15)$$

$$q_m = \frac{L_m^2}{L_r} \left( (i_m \bullet i_s) w_r + \frac{1}{T_r} i_m \otimes i_s \right) \quad (6.16)$$

Using (6.15) and (6.16) as the reference model and the adjustable model, respectively. An MRAS system can be drawn as in Fig.6.7, where proportional and integral (PI)

operations are utilized as the adaptation mechanism. From (6.15) and (6.16), it is evident that the speed estimation system of Fig.6.8 is completely robust to the stator resistance, besides requiring no integral calculation.

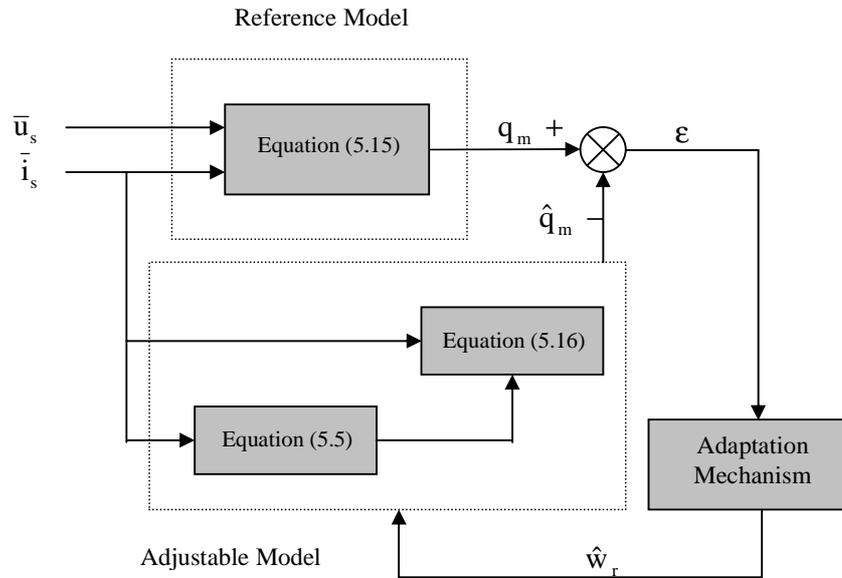


Fig.6.8- System structure of rotor speed observer using the tuning signal  $\text{Im}(\Delta \bar{e}_m \bar{i}_s)$

The information required for this module is stator voltage and stator current components in the d-q stationary reference frame. Two sets of equations are developed to compute reactive power of the induction motor in the reference and adaptive models. The reference model does not involve the rotor speed while the adaptive model needs the estimated rotor speed to adjust the computed reactive power to that computed from the reference model. Notice that the representation of complex number is defined for the stator voltages and currents in the stationary reference frame i.e.,  $v_s = v_{sd} + jv_{sq}$  and  $\bar{i}_s = i_{sd} + ji_{sq}$ .

### 6.6.1 Reference Model Continuous Time Representation

The back emf of the induction motor can be expressed in the stationary frame as follows:

$$\hat{e}_{md} = \frac{L_m}{L_r} \frac{d\psi_{rd}}{dt} = v_{sd} - R_s i_{sd} - \sigma L_s \frac{di_{sd}}{dt} \quad (6.17)$$

$$\hat{e}_{mq} = \frac{L_m}{L_r} \frac{d\psi_{rq}}{dt} = v_{sq} - R_s i_{sq} - \sigma L_s \frac{di_{sq}}{dt} \quad (6.18)$$

$$\bar{e}_m = e_{md} + j e_{mq} \quad (6.19)$$

The reactive power of the induction motor can be computed from cross product of stator currents and back emf vectors as follows:

$$q_m = \bar{i}_s \times \bar{e}_m = \bar{i}_s \times \left( v_s - R_s \bar{i}_s - \sigma L_s \frac{d\bar{i}_s}{dt} \right) = \bar{i}_s \times \bar{v}_s - \bar{i}_s \times \sigma L_s \frac{d\bar{i}_s}{dt} \quad (6.20)$$

where  $\bar{i}_s \times \bar{i}_s = i_{sd} i_{sq} - i_{sq} i_{sd} = 0$  and  $\sigma = 1 - \frac{L_m^2}{L_r L_s}$  (leakage coefficient) As a result

the reactive power shown in (6.20) can further be derived as

$$q_m = i_{sd} v_{sq} - i_{sq} v_{sd} - \sigma L_s \left( i_{sd} \frac{di_{sq}}{dt} - i_{sq} \frac{di_{sd}}{dt} \right) \quad (6.21)$$

### 6.6.2 Adaptive Model Continuous Time Representation

The estimated back emf computed in the adaptive model can be expressed as follows:

$$\hat{e}_{md} = \frac{L_m^2}{L_r} \frac{di_{md}}{dt} = \frac{L_m^2}{L_r} \left( -T_r \hat{\omega}_r i_{mq} - i_{md} + i_{sd} \right) \quad (6.22)$$

$$\hat{e}_{mq} = \frac{L_m^2}{L_r} \frac{di_{mq}}{dt} = \frac{L_m^2}{L_r} \left( -T_r \hat{\omega}_r i_{md} - i_{mq} + i_{sq} \right) \quad (6.23)$$

$$\hat{e}_m = \hat{e}_{md} + j \hat{e}_{mq} \quad (6.24)$$

where  $T_r = \frac{L_r}{R_r}$  is the rotor time constant,  $i_{md}, i_{mq}$  are computed from the following equations:

$$\frac{di_{md}}{dt} = -\hat{\omega}_r i_{mq} - \frac{1}{T_r} i_{md} + \frac{1}{T_r} i_{sd} \quad (6.25)$$

$$\frac{di_{mq}}{dt} = -\hat{\omega}_r i_{md} - \frac{1}{T_r} i_{mq} + \frac{1}{T_r} i_{sq} \quad (6.26)$$

Once the estimated back emf computed by (6.22)-(6.26), the estimated reactive power can be computed as follows:

$$\hat{q}_m = \bar{i}_s \times \hat{e}_m = i_{sd} \hat{e}_{mq} - i_{sq} \hat{e}_{md} \quad (6.27)$$

Then, the PI controller tunes the estimated rotor speed such that the reactive power generated by adaptive model matches that generated by reference model. The speed tuning signal is the error of reactive power that can be expressed as follows:

$$\varepsilon_{\Delta e} = \bar{i}_s \times (\bar{e}_m - \hat{e}_m) = q_m - \hat{q}_m \quad (6.28)$$

When this observer is used in a vector-controlled drive, it is possible to obtain satisfactory performance even at very low speeds. The observer can track the actual rotor speed with a bandwidth that is only limited by noise, so the PI controller gains should be as large as possible. The scheme is insensitive to stator resistance variations. The parameter  $T_r$  has a negligible influence on the operation of both of the overall MRAS vector control systems. If the MRAS successfully maintains nearly zero error, and if the same value of  $T_r$  is used in the MRAS adjustable models and in the function block for calculating  $w_{slip}$ , then we have the following relations:

$w_e = \hat{w}_e$  and  $T_r w_{slip} = \hat{T}_r \hat{w}_{slip}$  where variables without “^” are actual values, and ones with “^” represent the corresponding values used in the MRAS vector control systems. Thus, if  $T_r \neq \hat{T}_r$ , then  $w_{slip} \neq \hat{w}_{slip}$ , but  $w_o = \hat{w}_o$ , which is used for orienting the stator current vector. Therefore, complete field-orientation can be achieved even if the value of  $T_r$  is quite wrong. The error in the value of  $T_r$ , however, produces an error in the speed feedback, thus affecting the accuracy of the speed control as follows:

$$\varepsilon_w = \hat{w}_r - w_r = \left(1 - \frac{T_r}{\hat{T}_r}\right) w_{\text{slip}} \quad (6.29)$$

This also holds for the previous MRAS scheme. However, the accuracy of the speed estimation system discussed depends on the transient stator inductance and also referred magnetizing inductance. The latter quantity is not too problematic, since it does not change with temperature. Furthermore, deviations of  $T_r$  from its correct value produces a steady-state error in the estimated speed and this error become significant at low speeds.

### 6.6.3 Discrete time representation for microcontroller implementation

For implementation on DSP based system, the differential equations need to be transformed to difference equations. Due to high sampling frequency compared to bandwidth of the system, the simple approximation of numerical integration, such as forward, backward, or trapezoidal rules, can be adopted [65]. Consequently, the reactive power equations in both references an adaptive model are discretized as described in the next section.

#### 6.6.3.1 Reference Model

According to (5.21) reference model reactive power is given as :

$$q_m = i_{sd} v_{sq} - i_{sq} v_{sd} - \sigma L_s \left( i_{sd} \frac{di_{sq}}{dt} - i_{sq} \frac{di_{sd}}{dt} \right)$$

Using backward approximation:

$$q_m(k) = i_{sd}(k)v_{sq}(k) - i_{sq}(k)v_{sd}(k) - \sigma L_s \left( i_{sd}(k) \frac{i_{sq}(k) - i_{sq}(k-1)}{T} - i_{sq}(k) \frac{i_{sd}(k) - i_{sd}(k-1)}{T} \right) \quad (6.30)$$

And this equation can be further simplified as:

$$q_m(k) = i_{sd}(k)v_{sq}(k) - i_{sq}(k)v_{sd}(k) - \frac{\sigma L_s}{T} (i_{sd}(k-1)i_{sq}(k) - i_{sd}(k)i_{sq}(k-1)) \quad (6.31)$$

where T is the sampling time.

### 6.6.3.2 Adaptive Model

According to (6.27), reactive power in adaptive model is derived as :

$$\hat{q}_m = \bar{i}_s \times \hat{e}_m = i_{sd} \hat{e}_{mq} - i_{sq} \hat{e}_{md}$$

whose discrete-time representation is:

$$\hat{q}_m(k) = i_{sd}(k) \hat{e}_{mq}(k) - i_{sq}(k) \hat{e}_{md}(k) \quad (6.32)$$

In order to compute  $\hat{e}_{md}(k)$  and  $\hat{e}_{mq}(k)$  consider their continuous time representations

$$\hat{e}_{md} = \frac{L_m^2}{L_r} \frac{di_{md}}{dt} = \frac{L_m^2}{L_r} (-T_r \hat{w}_r i_{mq} - i_{md} + i_{sd}) \quad (6.33)$$

$$\hat{e}_{mq} = \frac{L_m^2}{L_r} \frac{di_{mq}}{dt} = \frac{L_m^2}{L_r} (-T_r \hat{w}_r i_{md} - i_{mq} + i_{sq})$$

which have the discrete-time representations as;

$$\hat{e}_{md}(k) = \frac{L_m^2}{L_r} (-T_r \hat{w}_r(k) i_{mq}(k) - i_{md}(k) + i_{sd}(k)) \quad (6.34)$$

$$\hat{e}_{mq}(k) = \frac{L_m^2}{L_r} (-T_r \hat{w}_r(k) i_{md}(k) - i_{mq}(k) + i_{sq}(k))$$

and  $i_{md}(k), i_{mq}(k)$  can be solved by using trapezoidal integration method which yields continuous time representation

$$\frac{di_{md}}{dt} = -\hat{w}_r i_{mq} - \frac{1}{T_r} i_{md} + \frac{1}{T_r} i_{sd} \quad (6.35)$$

$$\frac{di_{mq}}{dt} = -\hat{w}_r i_{md} - \frac{1}{T_r} i_{mq} + \frac{1}{T_r} i_{sq}$$

and discrete-time representation as;

$$i_{md}(k) = i_{md}(k-1) \left[ -\frac{T^2}{2} \hat{w}_r^2(k) + 1 - \frac{T}{T_r} + \left( \frac{T}{T_r} \right)^2 \right] - \quad (6.36)$$

$$i_{mq}(k-1) \hat{w}_r(k) \left[ T - \frac{T^2}{T_r} \right] + i_{sd}(k) \left[ \frac{T}{T_r} - \frac{T^2}{2T_r^2} \right] -$$

$$i_{sq}(k) \hat{w}_r(k) \left[ \frac{T^2}{2T_r} \right]_{123}$$

$$\begin{aligned}
i_{mq}(k) = & i_{mq}(k-1) \left[ -\frac{T^2}{2} \hat{w}_r^2(k) + 1 - \frac{T}{T_r} + \left( \frac{T}{T_r} \right)^2 \right] - \\
& i_{md}(k-1) \hat{w}_r(k) \left[ T - \frac{T^2}{T_r} \right] + i_{sq}(k) \left[ \frac{T}{T_r} - \frac{T^2}{2T_r^2} \right] - \\
& i_{sd}(k) \hat{w}_r(k) \left[ \frac{T^2}{2T_r} \right]
\end{aligned} \tag{6.37}$$

#### 6.6.4 Per unit, discrete time representation

For the sake of generality, the per unit concept is used in all equations. However, for the simplicity the same variables are also used in the per unit representations.

##### 6.6.4.1 Reference Model

Dividing (6.30) by base power of  $V_b I_b$ , then its per unit representation is as follows:

$$\begin{aligned}
q_m(k) = & i_{sd}(k) v_{sq}(k) - i_{sq}(k) v_{sd}(k) - \\
& K_1 (i_{sd}(k-1) i_{sq}(k) - i_{sd}(k) i_{sq}(k-1))
\end{aligned} \tag{6.38}$$

Rearranging (6.38) to have the one in (6.39);

$$q_m(k) = i_{sd}(k) (v_{sq}(k) - K_1 i_{sq}(k-1)) - i_{sq}(k) (v_{sd}(k) + K_1 i_{sd}(k-1)) \text{ pu} \tag{6.39}$$

where  $K_1 = \frac{\sigma L_s I_b}{T V_b}$ ,  $V_b$  is base voltage, and  $I_b$  is base current.

##### 6.6.4.2 Adaptive Model

Dividing (6.34) by base voltage  $V_b$ , then yields

$$\hat{e}_{md}(k) = K_2 (-K_3 \hat{w}_r(k) i_{mq}(k) - i_{md}(k) + i_{sd}(k)) \text{ pu} \tag{6.40}$$

$$\hat{e}_{mq}(k) = K_2 (-K_3 \hat{w}_r(k) i_{md}(k) - i_{mq}(k) + i_{sq}(k)) \text{ pu}$$

where  $K_2 = \frac{L_m^2 I_b}{L_r T_r V_b}$ ,  $K_3 = T_r w_b = \frac{L_r w_b}{R_r}$  and  $w_b = 2\pi f_b$  is base electrical angular velocity. Similarly, dividing (6.36) and (6.37) by base current  $I_b$ , then yields

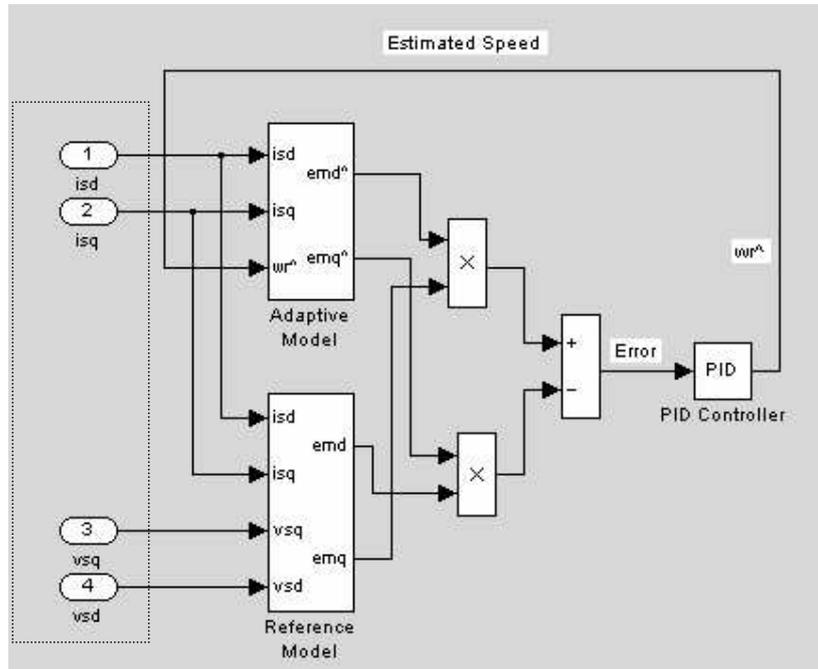
$$\begin{aligned} i_{md}(k) &= i_{md}(k-1) \left[ -K_4 \hat{w}_r^2(k) + K_5 \right] - \\ &\quad i_{mq}(k-1) \hat{w}_r(k) K_6 + i_{sd}(k) K_7 - i_{sq}(k) \hat{w}_r(k) K_8 \\ i_{mq}(k) &= i_{mq}(k-1) \left[ -K_4 \hat{w}_r^2(k) + K_5 \right] - \\ &\quad i_{md}(k-1) \hat{w}_r(k) K_6 + i_{sq}(k) K_7 - i_{sd}(k) \hat{w}_r(k) K_8 \end{aligned} \quad (6.41)$$

where

$$\begin{aligned} K_4 &= \frac{w_b^2 T^2}{2}, \quad K_5 = 1 - \frac{T}{T_r} + \left( \frac{T}{T_r} \right)^2, \quad K_6 = w_b \left( T - \frac{T^2}{T_r} \right), \quad K_7 = -\frac{T}{T_r} + \left( \frac{T}{T_r} \right)^2 \\ \text{and } K_8 &= \frac{w_b T^2}{2} \end{aligned}$$

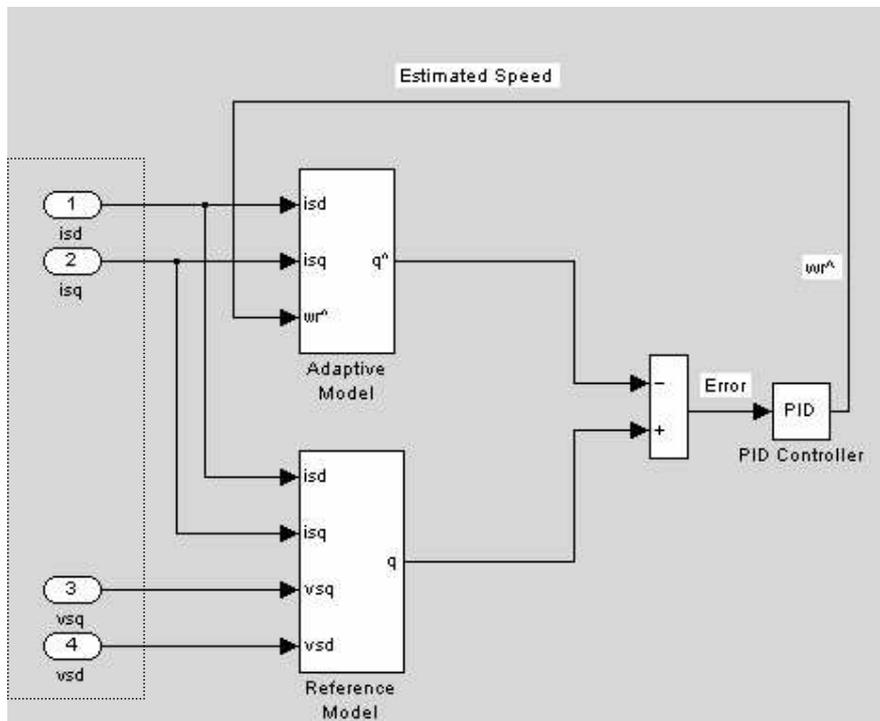
## 6.7 Simulation of the MRAS Schemes

In this thesis both the back emf and the reactive power schemes are studied in detail. In addition to the studies related with the theoretical base of the models, the simulations of the models are implemented to confirm the theoretical results using Matlab – Simulink. In these simulations the voltage and current outputs of induction machine model are used as the inputs of MRAS schemes. Two independent observers are configured to estimate the components of back emf and reactive power. The observer that does not involve rotor speed is called reference model, and the other observer including rotor speed is called adaptive or adjustable model. The error between the outputs of the two observers is then used to derive a suitable adaptation mechanism which generates the estimated speed for the adaptive model as shown in Fig.6.9 and Fig. 6.10. In Fig.6.9 the adaptive model is configured based on (6.3)-(6.5) and similarly the reference model is configured according to equations (6.6)-(6.7). In Fig.6.10 the adaptive model is configured according to (6.15) and the reference model is configured according to (6.16).



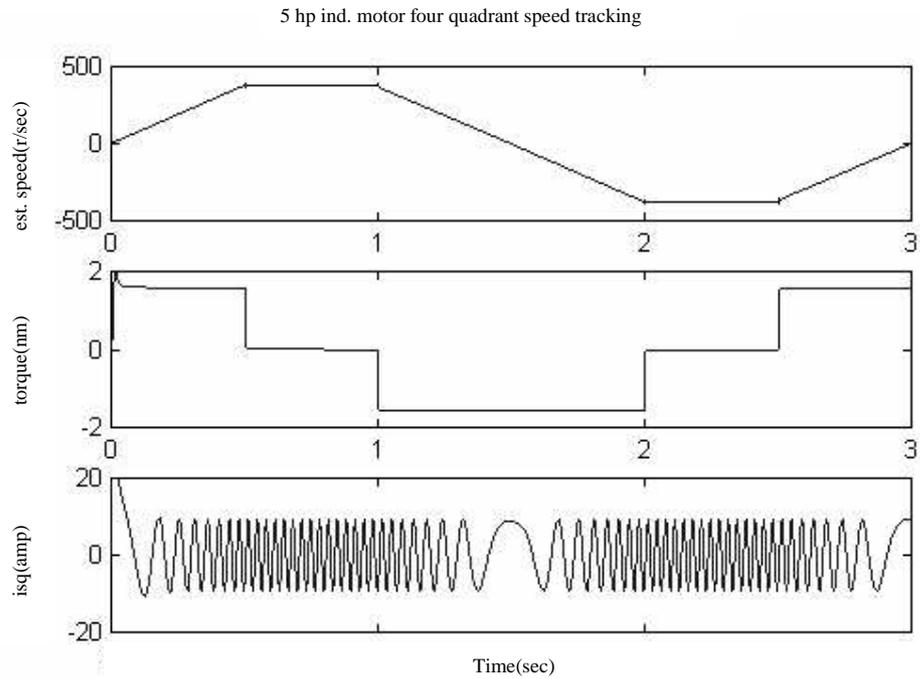
Motor Model

Fig .6.9- The Simulink model of back emf MRAS scheme

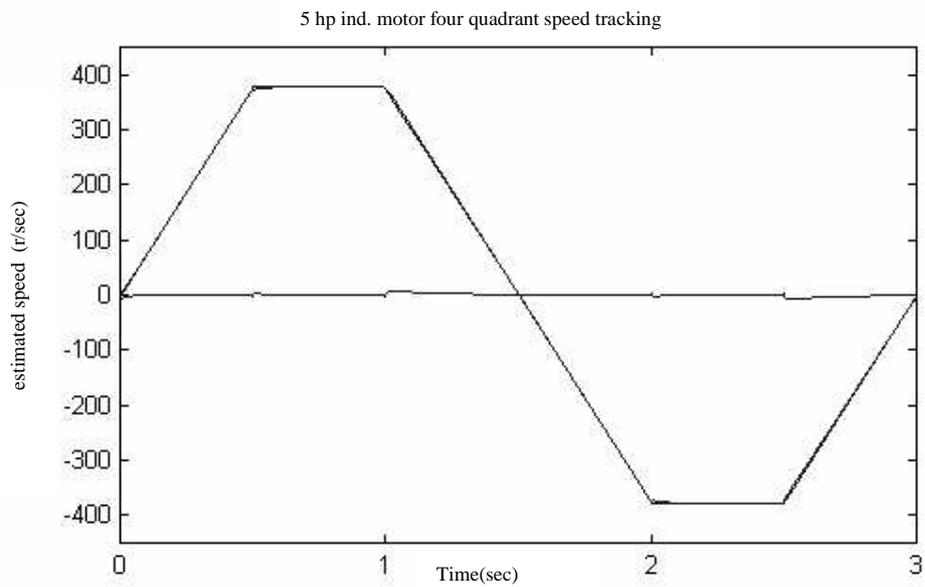


Motor Model

Fig.6.10- The Simulink model of reactive power MRAS scheme

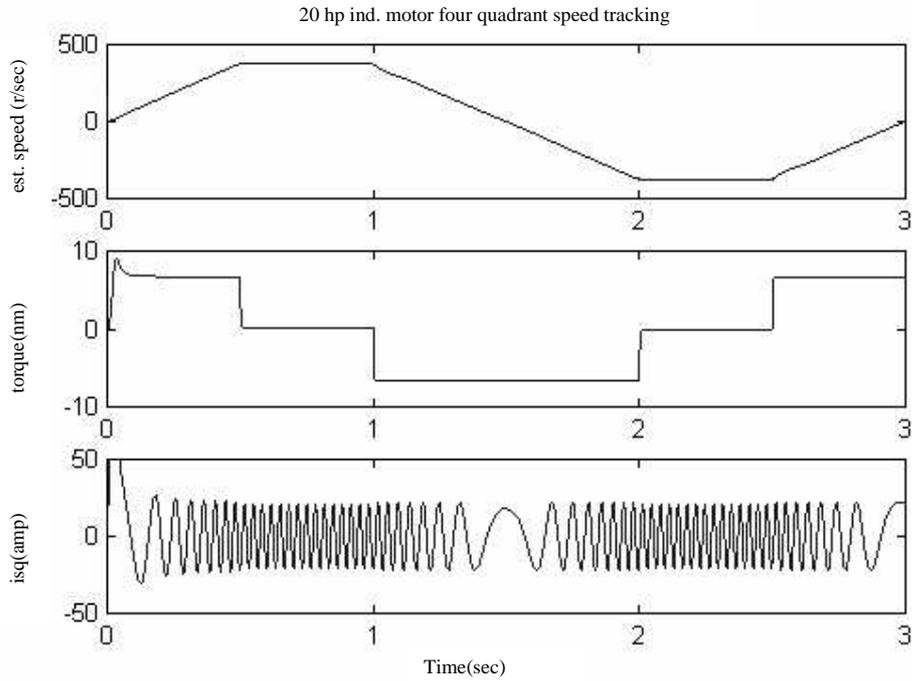


(a)

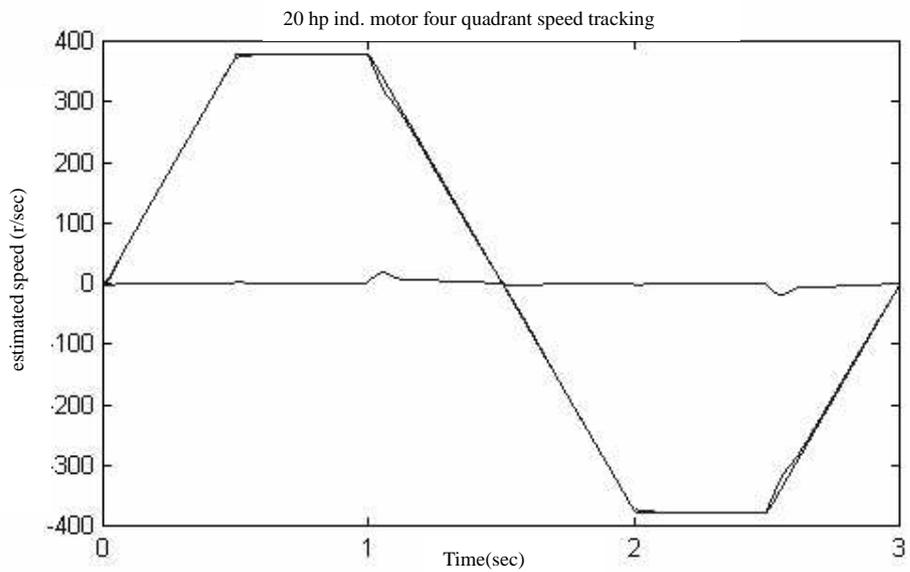


(b)

Fig. 6.11- Four-quadrant speed reversal of 5 hp induction motor using reactive power MRAS scheme at no\_load up to rated speed (a) estimated speed, produced torque due to inertia (J), q-axis stator current (b) estimated speed, speed error (difference between the actual and estimated speed)

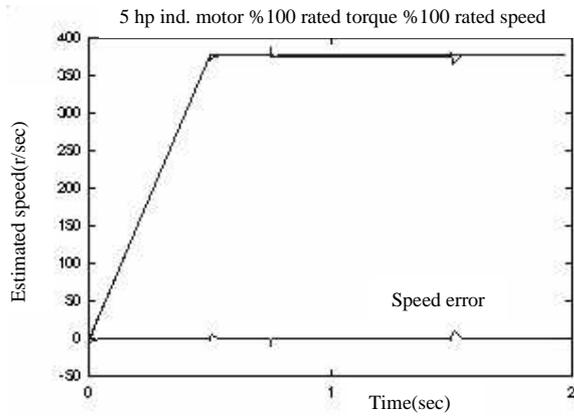


(a)

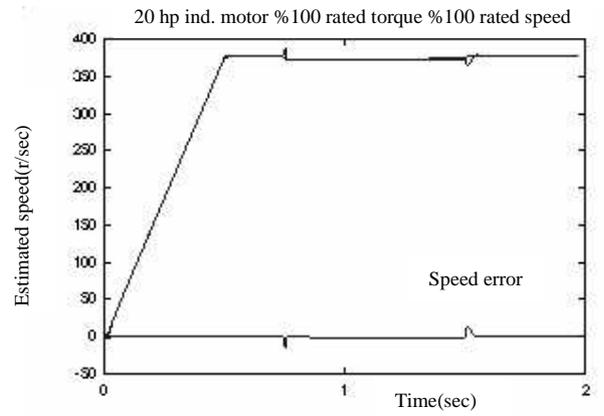


(b)

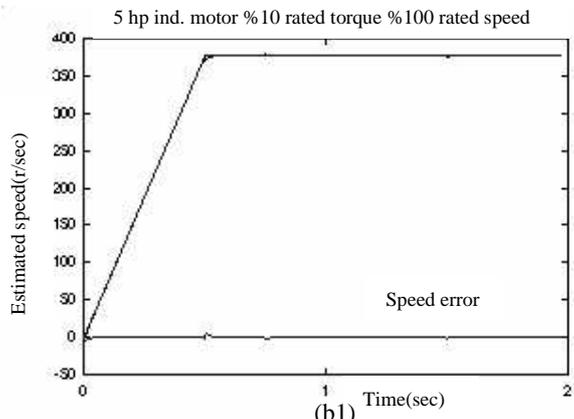
Fig. 6.12- Four-quadrant speed reversal of 20 hp induction motor using reactive power MRAS scheme at no\_load up to rated speed (a) estimated speed, produced torque due to inertia (J), q-axis stator current (b) estimated speed, speed error (difference between the actual and estimated speed)



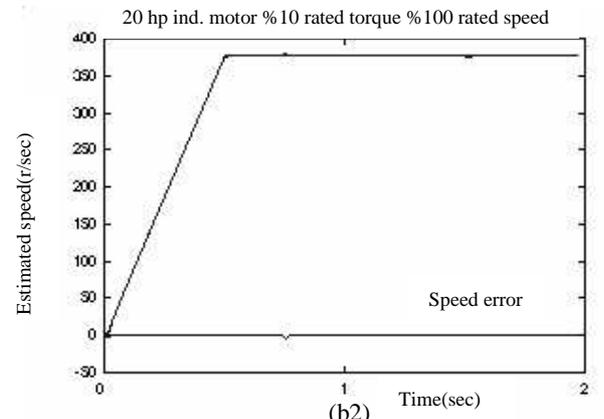
(a1)



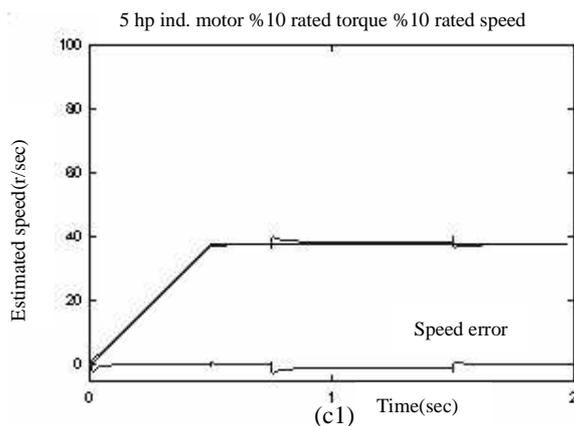
(a2)



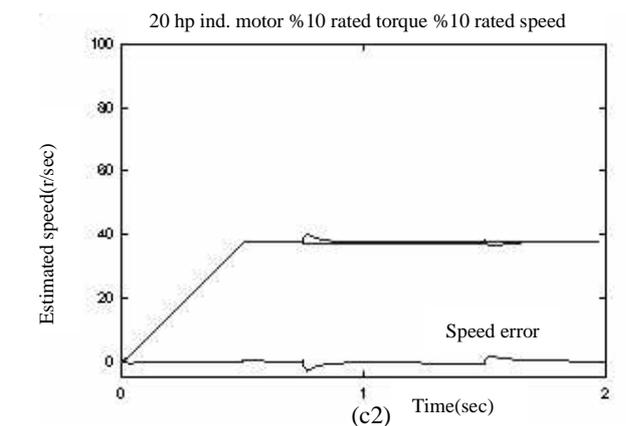
(b1)



(b2)

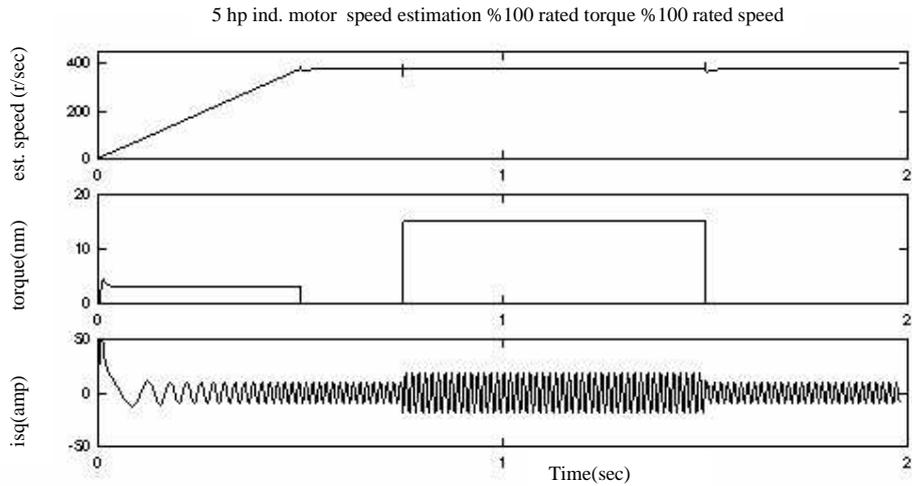


(c1)

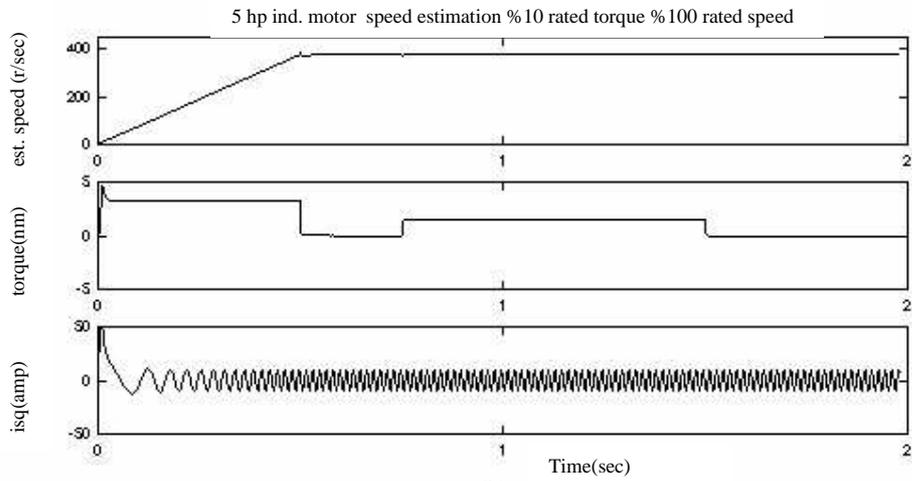


(c2)

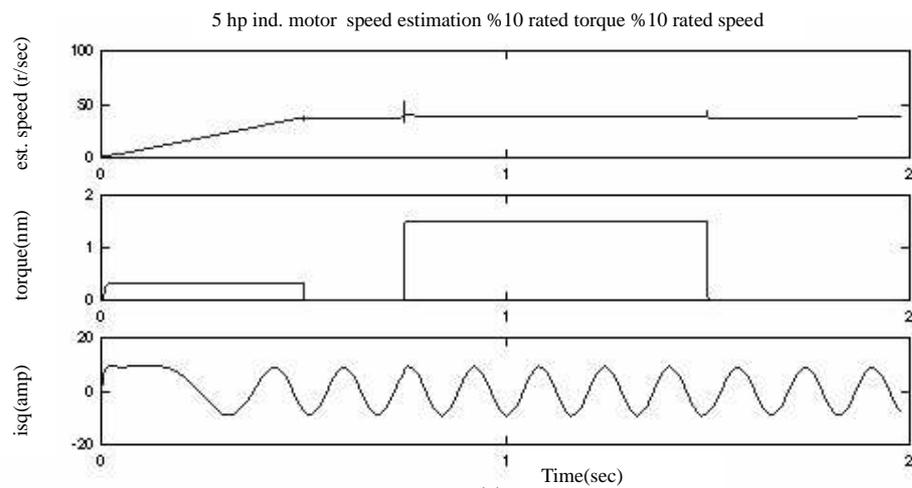
Fig.6.13- 5hp induction motor speed est. (a1) %100 rated torque - %100 rated speed  
 (b1) %10 rated torque-%100 rated speed (c1) %10 rated torque - %10 rated speed  
 20 hp induction motor speed est. (a2) %100 rated torque - %100 rated speed  
 (b2) %10 rated torque-%100 rated speed (c2) %10 rated torque - %10 rated speed  
 note: load torque applied between 0.75s to 1.5s in all figures



(a)

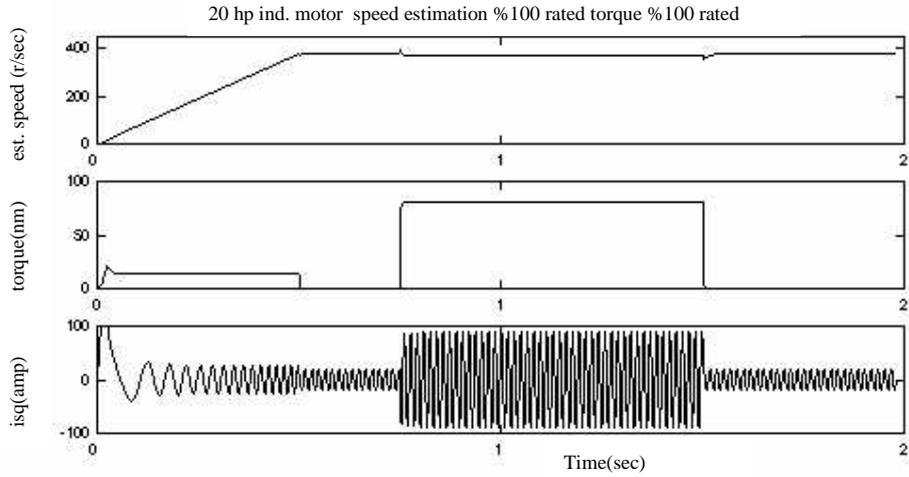


(b)

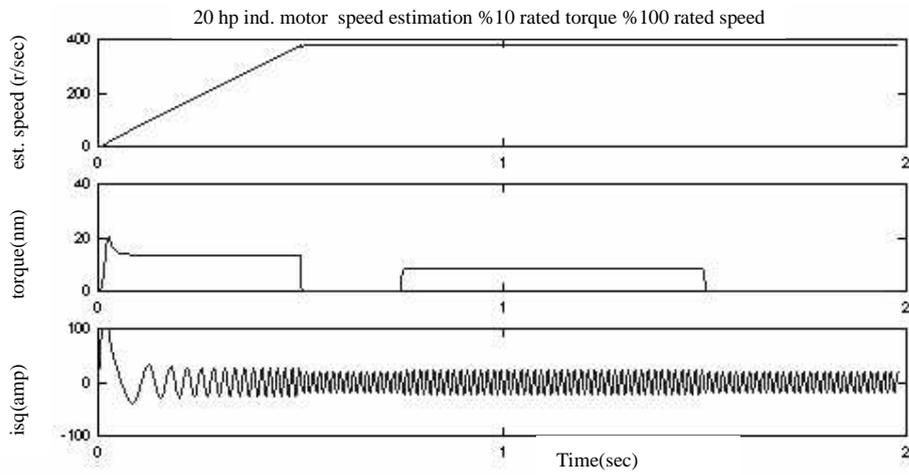


(c)

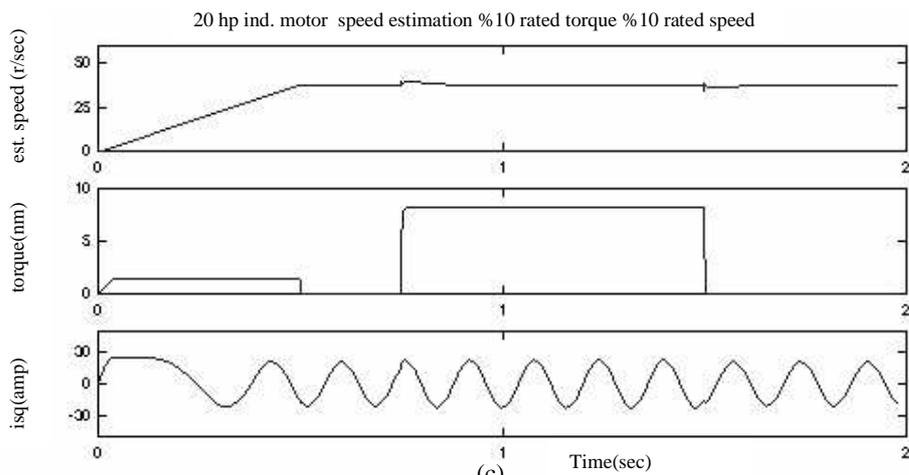
Fig.6.14- 5hp induction motor estimated speed speed using reactive power MRAS scheme, applied torque and stator q-axis current (a)%100 rated torque, %100 rated speed (b)%10 rated torque, %100 rated speed (c)%10 rated torque , %10 rated speed  
note: load torque applied between 0.75s –1.5s



(a)

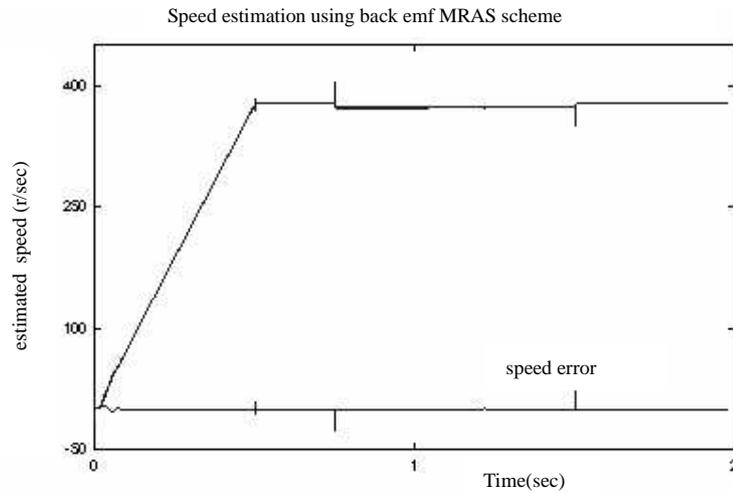


(b)

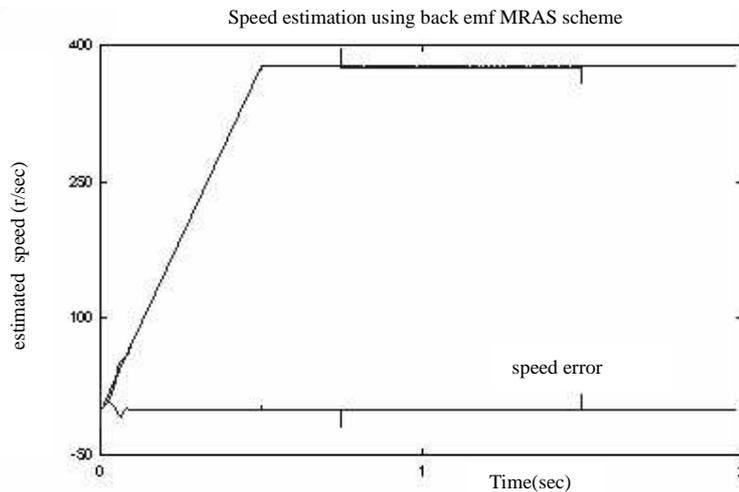


(c)

Fig.6.15- 20 hp induction motor estimated speed using reactive power MRAS scheme, applied torque and stator q-axis current (a)% 100 rated torque, % 100 rated speed (b)% 10 rated torque, % 100 rated speed(c)% 10 rated torque , % 10 rated speed  
note: load torque applied between 0.75s –1.5s



(a)



(b)

Fig.6.16- A typical estimated speed using back emf MRAS scheme (a) 20hp ind. motor % 100 rated torque % 100 rated speed (b) 5hp ind. motor % 100 rated torque % 100 rated speed

*note: load torque applied between 0.75s-1.5s*

Fig.6.11 and Fig.6.12 show simulation results for four-quadrant speed tracking performance of the reactive power MRAS scheme with 20 hp and 5 hp induction motors. The simulations here are implemented under no-load conditions. Fig.6.11 (a) and Fig.6.12 (a) includes estimated speed in rad/sec, generated electromagnetic torque during acceleration and deceleration due to inertia term (J) in nm., and stator q axis current in amp.

In Fig.6.11 (b) and Fig.6.12 (b), more accurate speed estimation and speed error (difference between the actual speed and estimated speed) are shown. No-load performance of the speed observer is very high as seen in these figures even at very low speed range. Since there does not exist any immediate transient speed change due to mechanical loading, speed outputs obtained in the simulation are very smooth with negligible speed errors.

In Fig.6.13, speed estimations of 5 hp and 20 hp induction motors are simulated under varying-load conditions. In Fig.6.13 (a1) and (a2) %100 rated torque is applied at %100 rated speed, in Fig. 6.13 (b1) and (b2) %10 rated torque is applied at %100 rated speed and in Fig.6.13 (c1) and (c2) %10 rated torque is applied at %10 rated speed. Heavy loading causes higher transient speed errors due to high instantaneous speed changes where light-loading speed errors are much smaller for both %100 rated speed and %10 rated speed conditions in percentage. In all of the situations transient speed tracking at starting and at steady-state, the speed estimation performances are quite high.

In Fig.6.14 and 6.15 the generated electromagnetic torque and q axis stator current are shown with estimated speed for both 5 hp and 20 hp induction motors at the same loading conditions given above. Fig.6.16 shows typical speed estimations for both of the motors under full-load condition. They are observed using back emf MRAS scheme. The speed observation performances of two MRAS schemes are almost the same except low-speed range, the near zero speed. The back emf scheme becomes highly dependent on the PID parameters particularly at low speed range, therefore, there may exist even instability problems which is never seen in reactive power scheme. So, reactive power scheme is superior to back emf scheme not only in immunity against the parameter deviation (stator resistance), but also in very low speed performance as well. Furthermore, transient speed changes in back emf scheme have higher overshoot when compared to reactive power scheme. Therefore, reactive power scheme is much better than all other MRAS schemes including pure integral and back emf scheme, thus, recommended for speed observation in ac drives.

## 6.8 Experimental Results

A state estimator made by using MRAS has been tested experimentally as well. The experimental data; the real time stator voltages and currents are obtained from the setup is processed by Matlab in the host computer where the associated MRAS program is running. The outputs of the processing are displayed in Figs.6.17, 6.18. Gains of PI can be changed to improve the settling time, overshoot, rise-time, etc of the speed waveform while the system is going through the transient-state. The steady-state accuracy of MRAS meet the expectations and quite successful. Also, Fig. 6.18 shows the speed tracking performance of the back emf MRAS scheme. It is seen that this tracking performance of the speed estimator seems to be quite satisfactory.

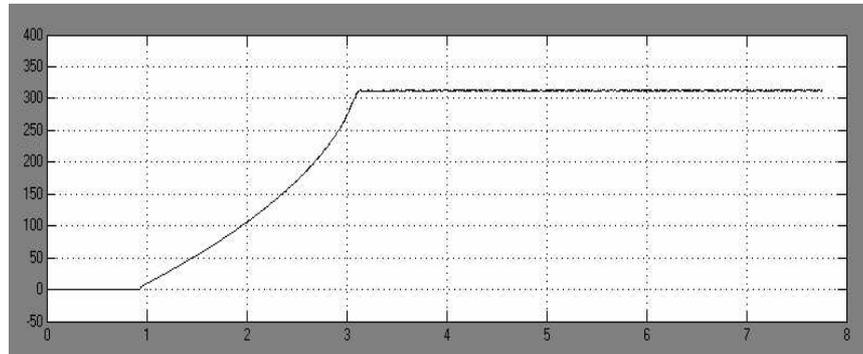


Fig.6.17 –Rotor speed estimated by MRAS experimentally at no-load by back emf scheme (*measured ss. speed: 314 rad/sec*)

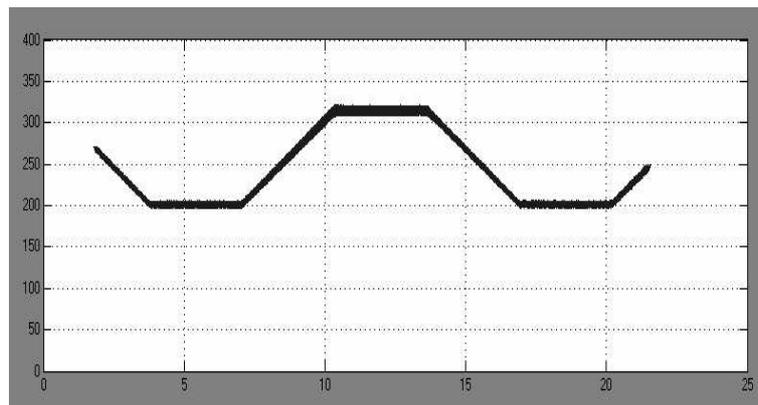


Fig. 6.18- Speed tracking of the back emf MRAS scheme.

The simulations and experimental works show the great promise of the studied MRAS schemes. However, due to equipment limitations these methods are not tested over a wide speed and torque ranges. Further, the experimental work is required to implement these techniques in the entire torque speed range of the induction motor.

## CHAPTER 7

### FLUX AND SPEED OBSERVERS FOR SENSORLESS DIRECT FIELD ORIENTATION

In this chapter, observers configured for direct field-orientation (DFO) are investigated. The field orientation is implemented in two ways as discussed in Chapter 2; Direct Field-Orientation and Indirect Field-Orientation. The basic difference of these methods underlies in the manner of detecting the synchronous speed. In IFO, the slip-angle is computed and added to the rotor speed to find the synchronous speed. Therefore, one must calculate the slip-angle and estimate the rotor angle. In the current model employed in the IFO, d q-axes stator currents and precise rotor time-constant are needed to find slip angle (or slip speed). Afterwards, adding these two angles will give the synchronous angle (see chapter 2 for detail). In the literature, a number of the algorithms are proposed to calculate the rotor angle (or rotor speed). Thus, previously proposed observers (designed by using MRAS and/or EKF techniques) are employed in IFO due to their rotor-speed estimation property. On the other hand, in DFO, the synchronous speed is computed from the ratio of dq-axes fluxes. Therefore, one must estimate the fluxes if sensorless control without hall-effect sensor is required. Flux estimator used in this chapter can compute both the synchronous speed and the rotor speed. Since the induction motor model applied to EKF in Chapter 5 estimates the rotor fluxes, it can also be applied to DFO. Furthermore, MRAS algorithm explained in Chapter 6 may be added to flux-observer used in a DFO for speed control. The different combinations of these observers can be implemented in both field-orientation methods.

## 7.1. Flux Observer

The flux observer module defined in this chapter is used for computing dq-stationary-axis fluxes and rotor-flux angle. The inputs of the observer are dq-stationary-axis currents and voltages. The logic underlying this flux observer is basically an advanced voltage model approach [66] in which integration of the back-emf is calculated with a different method. The well-known disadvantages of this model are parameter dependency (i.e.  $R_s$ ) at low-speed and drifting off of the integral of the sensed variables. These problems are compensated with a closed-loop in this flux observer. Basically, the fluxes obtained by current model are compared with those obtained by the voltage model then the error is fed to a PID block to obtain compensating voltages those are added to sensed stator voltages. There exist several algorithms in the literature which correct the voltage model with reference to the current model, or the current model with reference to the voltage model according the range in which one of these models is superior to other. In this flux observer, the voltage model is corrected by the current model through a basic PI block. In the end, the stator fluxes are used to obtain rotor fluxes and rotor flux angle. The overall observer structure is given below:

### Continuous Time:

The rotor flux dynamics developed by current model in synchronously rotating reference frame ( $w=w_e$ ) can be shown as:

$$\frac{d\psi_{dr}^{e,i}}{dt} = \frac{L_m}{T_r} i_{ds}^e - \frac{1}{T_r} \psi_{dr}^{e,i} + (w_e - w_r) \psi_{qr}^{e,i} \quad (7.1)$$

$$\frac{d\psi_{qr}^{e,i}}{dt} = \frac{L_m}{T_r} i_{qs}^e - \frac{1}{T_r} \psi_{qr}^{e,i} + (w_e - w_r) \psi_{dr}^{e,i} \quad (7.2)$$

where, superscript (i) represents current model dynamics and (e) represents synchronous frame.  $w_e$  is synchronous speed and  $w_r$  is the electrical rotor speed in rad/sec.

In rotor field-orientation, the main goal is to align the rotor-flux vector to the d-axis stator current, thus q- axis rotor-flux is regarded to be zero. That is:

$$\psi_r^{e,i} = \psi_{dr}^{e,i} \text{ and } \psi_{qr}^{e,i} = 0$$

Thus, (7.1) and (7.2) can be simplified to:

$$\frac{d\psi_{dr}^{e,i}}{dt} = \frac{L_m}{T_r} i_{ds}^{e,i} - \frac{1}{T_r} \psi_{dr}^{e,i} \quad (7.3)$$

$$\psi_{dr}^{e,i} = 0 \quad (7.4)$$

Then the rotor flux-linkages are transformed into the stationary reference frame by the inverse park transformation:

$$\psi_{dr}^{s,i} = \psi_{dr}^{e,i} \cos(\theta_{\psi_r}) - \psi_{qr}^{e,i} \sin(\theta_{\psi_r}) = \psi_{dr}^{e,i} \cos(\theta_{\psi_r}) \quad (7.5)$$

$$\psi_{qr}^{s,i} = \psi_{dr}^{e,i} \sin(\theta_{\psi_r}) + \psi_{qr}^{e,i} \sin(\theta_{\psi_r}) = \psi_{dr}^{e,i} \sin(\theta_{\psi_r}) \quad (7.6)$$

where  $\theta_{\psi_r}$  is the rotor-flux angle superscript (s) represents the stationary reference frame. The stator fluxes are obtained from (7.5) and (7.6) as:

$$\psi_{ds}^{s,i} = L_s i_{ds}^s + L_m i_{dr}^s = \left( \frac{L_s L_r - L_m^2}{L_r} \right) i_{ds}^s + \frac{L_m}{L_r} \psi_{dr}^{s,i} \quad (7.7)$$

$$\psi_{qs}^{s,i} = L_s i_{qs}^s + L_m i_{qr}^s = \left( \frac{L_s L_r - L_m^2}{L_r} \right) i_{qs}^s + \frac{L_m}{L_r} \psi_{qr}^{s,i} \quad (7.8)$$

where  $L_s$ ,  $L_r$  the stator and rotor self-inductance, respectively, and  $L_m$  refers to the magnetizing inductance. The same quantities will be obtained with voltage model as;

$$\psi_{ds}^{s,v} = \int (\mathbf{u}_{ds}^s - i_{ds}^s R_s - \mathbf{u}_{comp,ds}) dt \quad (7.9)$$

$$\psi_{qs}^{s,v} = \int (\mathbf{u}_{qs}^s - i_{qs}^s R_s - \mathbf{u}_{comp,q_s}) dt \quad (7.10)$$

where  $R_s$  is the stator resistance and parenthesis in the integration is back emf with compensated voltages. The superscript (v) indicates that the equations are developed by considering the voltage model. After calculating the stator fluxes with voltage model in (7.9) and (7.10), they are compared with the stator fluxes calculated by

current model in (7.7) and (7.8). Then the error is fed to a PI block to obtain compensated voltages.

$$\mathbf{u}_{\text{comp,ds}} = \mathbf{K}_p (\boldsymbol{\psi}_{\text{ds}}^{s,v} - \boldsymbol{\psi}_{\text{ds}}^{s,i}) + \mathbf{K}_I \int (\boldsymbol{\psi}_{\text{ds}}^{s,v} - \boldsymbol{\psi}_{\text{ds}}^{s,i}) dt \quad (7.11)$$

$$\mathbf{u}_{\text{comp,qd}} = \mathbf{K}_p (\boldsymbol{\psi}_{\text{qs}}^{s,v} - \boldsymbol{\psi}_{\text{qs}}^{s,i}) + \mathbf{K}_I \int (\boldsymbol{\psi}_{\text{qs}}^{s,v} - \boldsymbol{\psi}_{\text{qs}}^{s,i}) dt \quad (7.12)$$

Once the stator fluxes obtained, rotor-flux vector is reconstructed in stationary frame by using (7.13) and (7.14) :

$$\boldsymbol{\psi}_{\text{dr}}^{s,v} = -\left(\frac{\mathbf{L}_s \mathbf{L}_r - \mathbf{L}_m^2}{\mathbf{L}_m}\right) \mathbf{i}_{\text{ds}}^s + \frac{\mathbf{L}_r}{\mathbf{L}_m} \boldsymbol{\psi}_{\text{ds}}^{s,v} \quad (7.13)$$

$$\boldsymbol{\psi}_{\text{qr}}^{s,v} = -\left(\frac{\mathbf{L}_s \mathbf{L}_r - \mathbf{L}_m^2}{\mathbf{L}_m}\right) \mathbf{i}_{\text{qs}}^s + \frac{\mathbf{L}_r}{\mathbf{L}_m} \boldsymbol{\psi}_{\text{qs}}^{s,v} \quad (7.14)$$

Then the rotor-flux angle based on the voltage model is calculated as:

$$\theta_{\psi_r} = \tan^{-1} \left( \frac{\boldsymbol{\psi}_{\text{qr}}^{s,v}}{\boldsymbol{\psi}_{\text{dr}}^{s,v}} \right) \quad (7.15)$$

## 7.2. Open-Loop Speed Estimator

The open-loop speed estimator employed in this FOC structure is a well-known method based on stationary reference frame. The disadvantage of this method is the parameter sensitivity as in the case of all open-loop estimators [67]. However, the structure of this algorithm is quite easy when compared to the advanced estimation techniques. The mathematical base of the estimator is given below that can easily be extracted from induction machine equations.

$$\boldsymbol{\psi}_{\text{dr}}^s = \mathbf{L}_r \mathbf{i}_{\text{dr}}^s + \mathbf{L}_m \mathbf{i}_{\text{ds}}^s \quad (7.16)$$

$$\boldsymbol{\psi}_{\text{qr}}^s = \mathbf{L}_r \mathbf{i}_{\text{qr}}^s + \mathbf{L}_m \mathbf{i}_{\text{qs}}^s \quad (7.17)$$

The rotor currents can be expressed as:

$$\mathbf{i}_{\text{dr}}^s = \frac{1}{\mathbf{L}_r} (\boldsymbol{\psi}_{\text{dr}}^s - \mathbf{L}_m \mathbf{i}_{\text{ds}}^s) \quad (7.18)$$

$$\mathbf{i}_{\text{qr}}^s = \frac{1}{\mathbf{L}_r} (\boldsymbol{\psi}_{\text{qr}}^s - \mathbf{L}_m \mathbf{i}_{\text{qs}}^s) \quad (7.19)$$

The rotor voltages can be expressed as:

$$0 = R_r i_{dr}^s + w_r \psi_{qr}^s + \frac{d\psi_{dr}^s}{dt} \quad (7.20)$$

$$0 = R_r i_{qr}^s - w_r \psi_{dr}^s + \frac{d\psi_{qr}^s}{dt} \quad (7.21)$$

where  $R_r$  is the rotor resistance. Substituting (7.18) and (7.19) into (7.20) and (7.21), rotor currents are extracted from the rotor flux dynamics as:

$$\frac{d\psi_{dr}^s}{dt} = -\frac{1}{T_r} \psi_{dr}^s + \frac{L_m}{T_r} i_{ds}^s - w_r \psi_{qr}^s \quad (7.22)$$

$$\frac{d\psi_{qr}^s}{dt} = -\frac{1}{T_r} \psi_{qr}^s + \frac{L_m}{T_r} i_{qs}^s + w_r \psi_{dr}^s \quad (7.23)$$

Since we know the rotor fluxes from the previous flux estimation module, one can calculate the rotor flux magnitude and angle:

$$\psi_r^s = \sqrt{(\psi_{dr}^s)^2 + (\psi_{qr}^s)^2} \quad (7.24)$$

$$\theta_{\psi r} = \tan^{-1} \left( \frac{\psi_{qr}^s}{\psi_{dr}^s} \right) \quad (7.25)$$

Then the synchronous speed,  $w_e$ , can be calculated by the derivative of (7.25) as:

$$w_e = \frac{d\theta_{\psi r}}{dt} = \frac{(\psi_{dr}^s)^2}{(\psi_r^s)^2} \left( \frac{\psi_{dr}^s \frac{d\psi_{qr}^s}{dt} - \psi_{qr}^s \frac{d\psi_{dr}^s}{dt}}{(\psi_r^s)^2} \right) \quad (7.26)$$

Then, (7.22) and (7.23) are substituted into (7.26)

$$w_e = \frac{d\theta_{\psi r}}{dt} = w_r + \underbrace{\frac{1}{(\psi_r^s)^2} \frac{L_m}{T_r} (\psi_{dr}^s i_{qs}^s - \psi_{qr}^s i_{ds}^s)}_{w_{slip}} \quad (7.27)$$

Finally, rotor speed is calculated as:

$$\omega_r = \omega_e - \frac{1}{(\psi_r^s)} \frac{L_m}{T_r} (\psi_{dr}^s i_{qs}^s - \psi_{qr}^s i_{ds}^s) \quad (7.28)$$

### 7.3. Experimental Results

The performance of the flux estimator and open-loop speed estimator are tested in our experimental setup. It is shown that both estimators work properly. The experimental results follow the same sequence with the equations to have ease in tracing the outputs state step by step.

The stator currents and voltages are quite close to a pure sinusoidal waveform due to highly precise approximation of SVPWM. Thus, the outputs of the flux estimator are expected to be sufficiently close to sinusoidal and the magnitude of the flux is smooth enough for torque control operations. In Fig.7.1, typical input stator current and stator voltage waveforms are given those are obtained during the experiments.

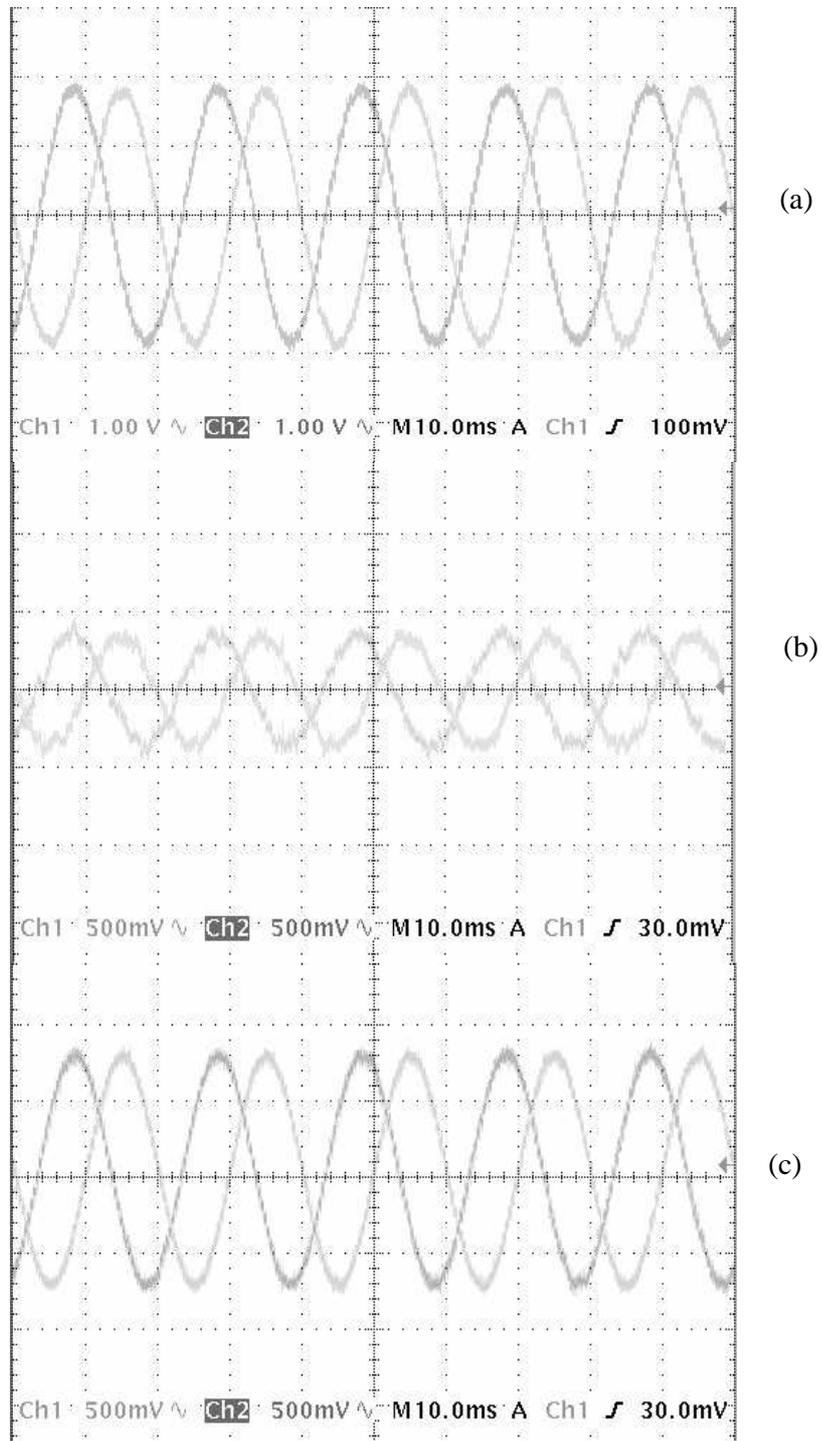


Fig.7.1- (a) stator phase currents under heavy load conditions (b) stator phase currents under no-load condition (c) stator phase voltages

The dq-axes rotor fluxes in stationary frame obtained from current model ((7.5), (7.6)) are shown in Fig.7.2.

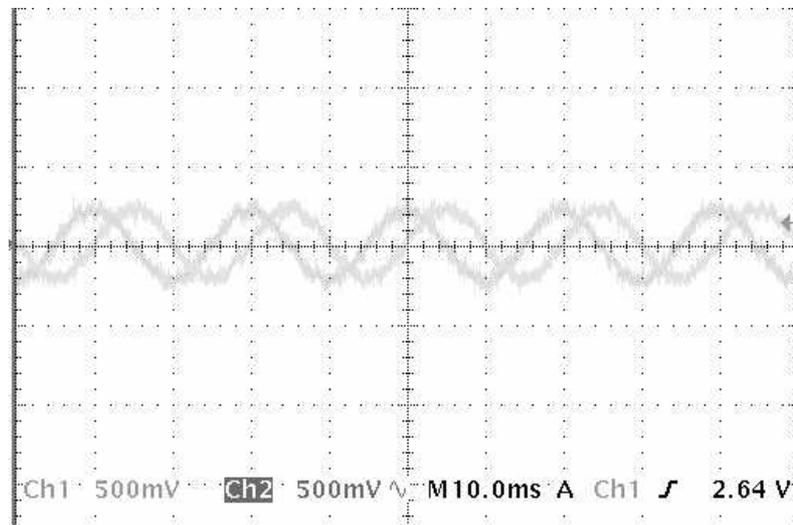


Fig.7.2- dq-axes rotor fluxes in stationary frame obtained from current model

The dq-axes stator fluxes in stationary frame obtained from the current model rotor flux estimates (( 7.7),(7.8)) are shown in Fig.7.3

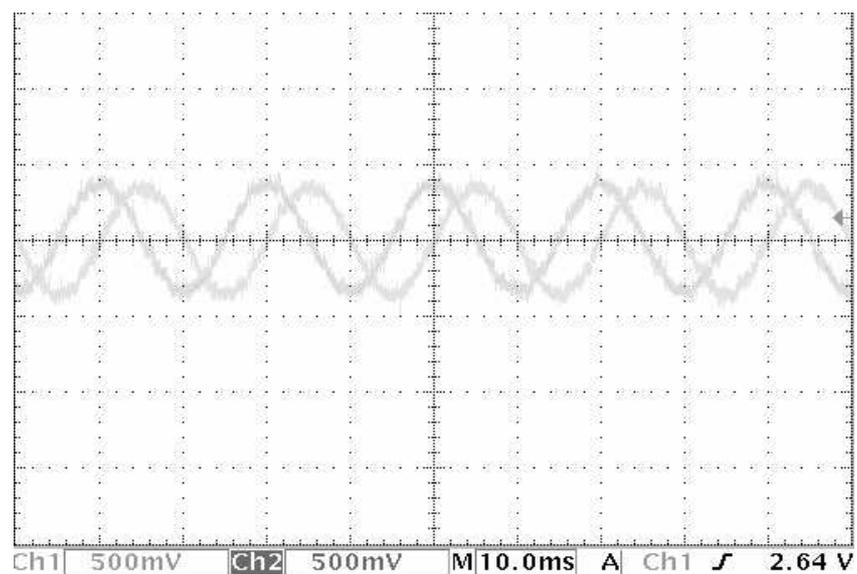


Fig.7.3- dq-axis stator fluxes in stationary frame obtained from the current model

The dq-axes stator fluxes in stationary frame obtained from the voltage model ((7.9),(7.10)) considering the compensating voltage are shown in Fig.7.4. The back emfs with added compensating voltages ((7.9),(7.10)) are shown in Fig.7.5.

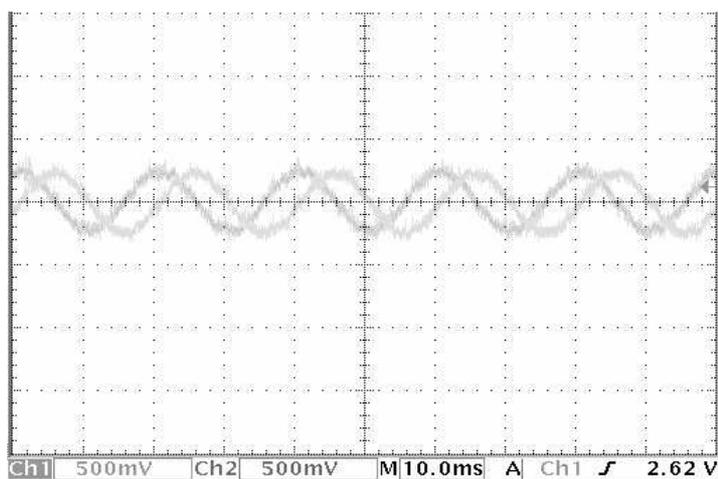


Fig7.4-The dq-axes stator fluxes in stationary frame obtained from the voltage model

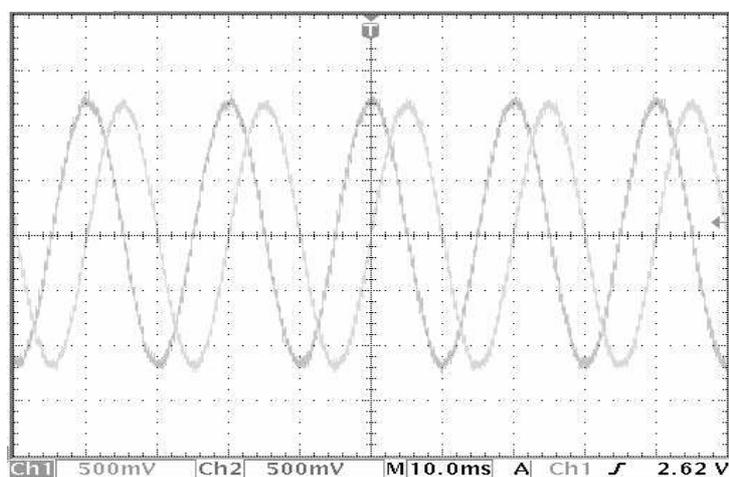


Fig.7.5- Back emfs with added compensating voltages

Finally, the outputs of the flux observer module, dq-axes stationary frame rotor fluxes reconstructed by the voltage model ((7.13),(7.14)), are shown in Fig.7.6 and the magnitude of the this rotor flux is shown in Fig.7.7.

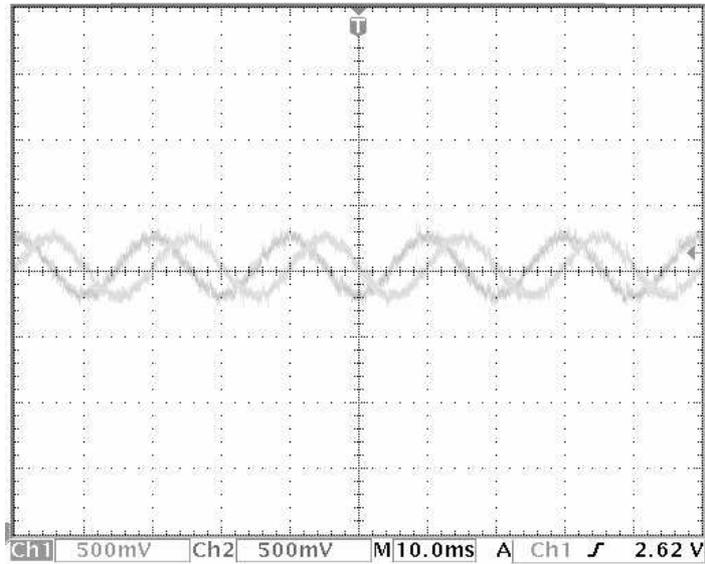


Fig.7.6 - dq-axes stationary frame rotor fluxes reconstructed by the voltage model

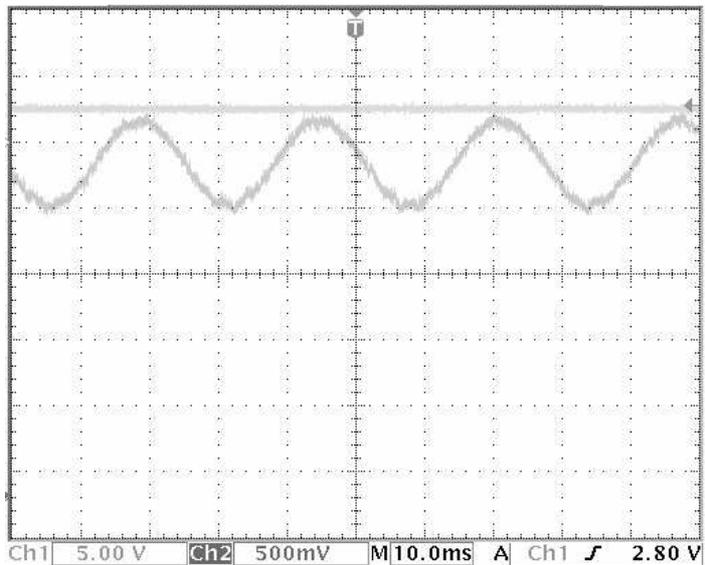


Fig.7.7. q-axis stationary frame rotor flux reconstructed by voltage model with rotor-flux magnitude

The smooth flux magnitude in Fig.7.7 guarantees a fast and constant torque response as in the case of dc motors. Another output of the flux estimator module, rotor-flux angle (7.15), is shown in Fig.7.8.

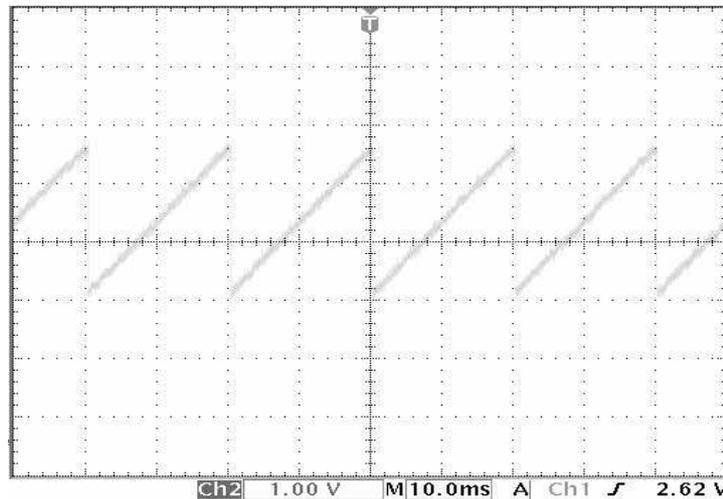


Fig.7.8- Rotor-flux angle based on the voltage model

The rest of the figures illustrate the estimated speed that is the output of open-loop speed estimator and the associated reference speed.

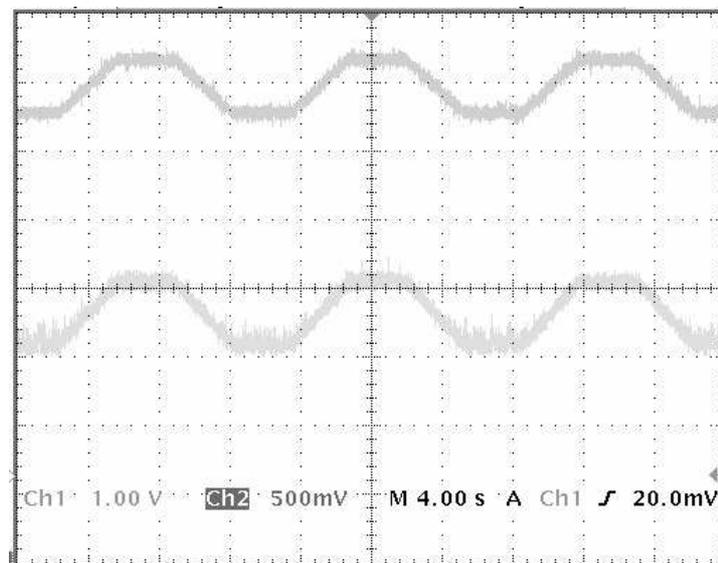


Fig.7.9- Reference speed (upper one) and estimated speed (lower one)  
(trapezoid limits are 0.27 pu to 0.4 pu)

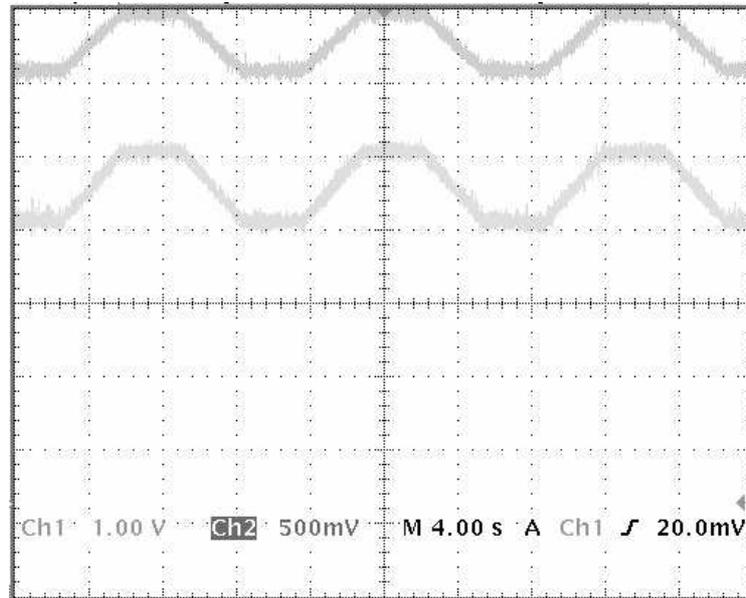


Fig.7.10- Reference speed (upper one) and estimated speed (lower one)  
(trapezoid limits are 0.4 pu to 0.5 pu)

## **CHAPTER 8**

### **THE HARDWARE & SOFTWARE**

In this chapter, the hardware configuration of experimental setup and the software organization will be summarized. The hardware configuration of the project is basically the combination of an asynchronous motor, a motor drive and a microprocessor. The software of the project involves assembly code of the FOC and state observers in modular strategy.

#### **8.1 Hardware Overview**

##### **8.1.1 The Motor**

The experimental setup of this thesis is as shown in Fig.8.1. While testing the setup, different motor sizes are used, but in the actual experimental stage 3kW squirrel cage induction motor (Siemens make) is used. In order to obtain motor parameters, classical no-load and locked-rotor tests are carried out on the motor. To get a rough starting guess of the parameters used in the FOC algorithm, steady-state model of the induction motor is employed as shown in Fig.8.2. In the real time applications, motor drives are expected to obtain motor parameters at the beginning with injected signals and on-line estimation of the parameters are embedded to FOC algorithm. These methods are skipped in this work and considered as future work. In this thesis, different from on-line parameter estimation, closed-loop observers (e.g. EKF) are expected to compensate the parameter deviation effects regarding the parameter errors as system noise.

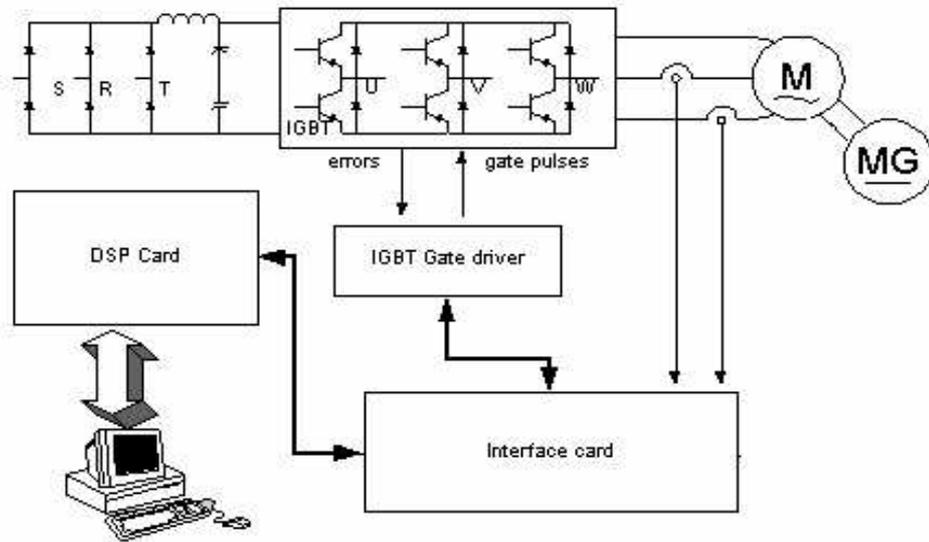


Fig.8.1 – Overall hardware configuration of the experimental setup

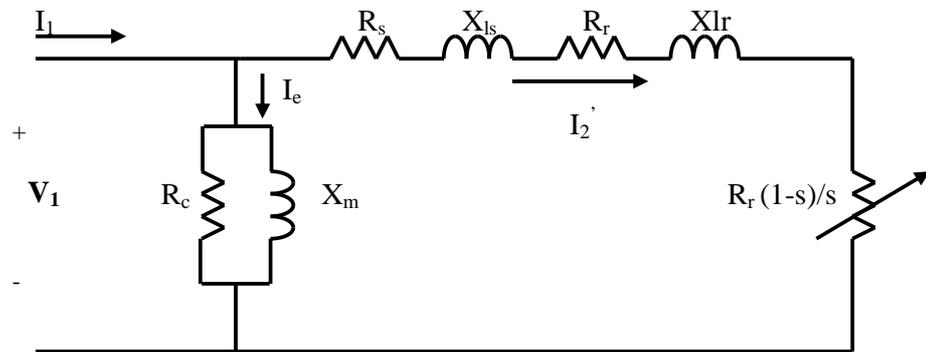


Fig.8.2 - Approximate per Phase Equivalent Circuit for an Induction Machine

$R_s$  in the equivalent circuit of Fig.8.2 is obtained by dc-test,  $R_c$  and  $X_m$  are determined by no-load and the rest of the parameters are determined by locked-rotor test. The stator resistance of each stator winding can be measured independently by applying a dc-current to one phase as shown in Fig. 8.3.

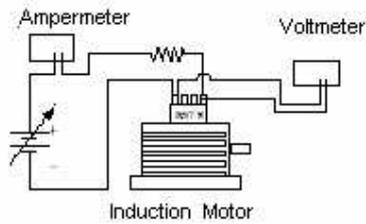


Fig.8.3 – Diagram of dc measurement

The stator resistance is measured on the motor terminals by applying a current through a resistor and measuring the corresponding voltage, or without a resistor applying low-level dc voltage. To obtain a more accurate measurement result, one must get several numbers of measured data and take the average of these data for each phase.

The leakage-reactances  $x_{ls}$ ,  $x_{lr}$  and the rotor resistance are determined when the motor speed is set to zero, i.e.  $s=1$ . Since the magnetizing branch elements are large enough compared to the rest of the equivalent components, these are neglected in this test. It is further assumed that leakage reactances are equal to each other according to IEEE test standards. Since stator resistance is measured and leakage reactances are assumed to be equal, rotor resistance can easily be calculated from the measured data. The measurements are done around the rated current of the motor and than the average of the measurements are computed to obtain more approximate parameters.

When the motor is running without load, the slip will be close to zero. Thus, the variable slip resistance will be very large. Therefore, in the no-load test one may consider the magnetizing branch as the approximate circuit of the motor model. The no-load data are measured around the rated voltage, and magnetizing branch elements are calculated around the rated voltage of the motor. The calculated motor parameters and ratings of the motor are given in Table-8.1

Table 8.1- Motor Parameters

$R_s$	(stator resistance)	2,0 ohm
$R_r$	(referred rotor resistance)	1,78 ohm
$R_c$		522 ohm
$X_m$	(magnetizing reactance)	65,25 ohm
$X_{ls}, X_{lr}$	(leakage reactances)	2,68 ohm
$L_s, L_r$	(stator & rotor induc.)	0,215 H
$L_{ls}, L_{lr}$	(leakage induc.)	0,0085 H
$L_m$	(magnetizing induc.)	0,207 H
$T_r$	(rotor time const.)	0,12

### 8.1.2. The Motor Drive

The drive circuit used here has been developed in another work [40] but modified to suit to the requirements of this work. The drive mainly includes a rectifier, dc-link circuit and an inverter. The rectifier used in this drive is Semikron-SKD28 that consists of six uncontrolled diodes. The rated current of the rectifier is 28 A and the rated operating voltage is 1300 V. During the tests, the three-phase voltage is supplied over an autotransformer to the rectifier.

In the dc-link circuit, the rectified voltage is a smooth dc filtered by dc-link capacitors. The filter is made of two 1000 $\mu$ F capacitors connected in series. In addition to them a resistor of 1W, 0.41 Mohm is connected across each capacitor to balance the voltage on them. The dc link voltage is supplied to the capacitors through a relay system as shown in the Fig. 8.4.

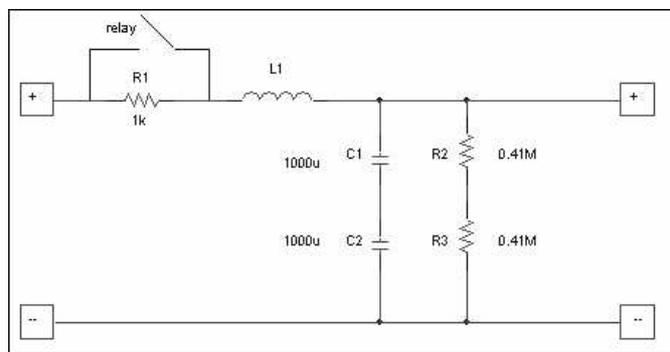


Fig.8.4 – Dc\_link Circuit

At the beginning, the capacitors are charged to a certain level through a 15 W resistor to prevent the inrush current at starting. When the capacitors are charged to predefined level, the relay disconnects the resistor. One can change the relay on-off voltage level by adjusting the controlling potentiometer on the interface card. Furthermore, by adding a manual switch, on-off state of the relay can be controlled manually by this switch. As a future work at this point, a dynamic braking circuit (freewheeling path controlled by a switch) may be added to avoid from over-charging the capacitors while the motor is slowing down rapidly.

The inverter on the drive is Semikron\_Semitrans IGBT module (SKM 40 GD 123 D). The rated value of  $V_{ce}$  in this IGBT package is 1200V and  $I_c$  is 40/30 A depending on the case temperature. The switching rise time of the IGBT switches is 55 ns and the switching fall time of the switches is 40 ns. This package may be used for applications at switching frequencies above 15 kHz. IGBTs in this module are triggered by a gate drive card, Semikron six IGBT driver (SKHI 60 H4). The gate drive card provides short-circuit protection for all six IGBTs in the full bridge. Short circuit protection scheme is based on the collector-emitter voltage of the devices. It switches off all IGBTs at once and gives an alarm in case a fault is detected. In our setup, these error outputs of the gate-drive card are used for fast hardware interrupt. The interlock circuit blocks simultaneous turning on of IGBTs of the same arm. One IGBT cannot be turned on before the gate charge of the other IGBT is completely removed.

The output of the inverter is connected to the motor through current sensors to acquire information about currents in real-time. Overall diagram of the inverter is shown in Fig.8.5.

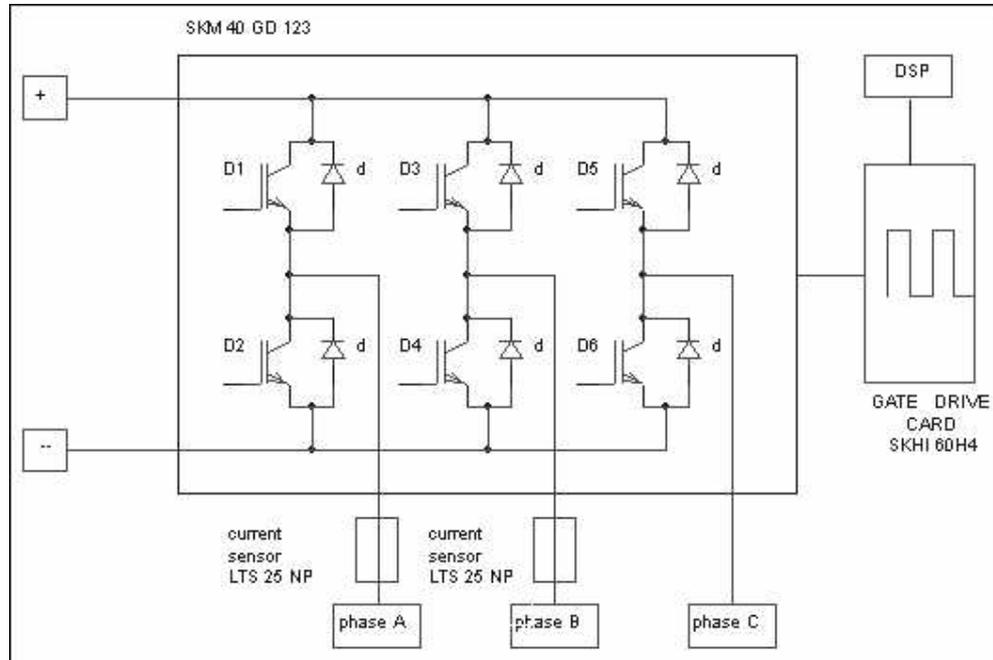


Fig.8.5 Inverter Circuit

### 8.1.3 The DSP

In order to run the real-time control algorithm and create PWM signals, Texas Instruments' (TI) TMS320 processor is used in this work. Texas Instruments' TMS320 family consists of fixed-point, floating-point, multiprocessor digital signal processors (DSPs). TMS320 DSPs have an architecture designed specifically for real-time signal processing. The F/C240 is a member of the 'C2000 DSP platform, and is optimized for control applications. The 'C24x series of DSP controllers combine this real-time processing capability with controller peripherals to create a suitable solution for vast majority of control system applications. The following characteristics make the TMS320 family a suitable choice for a wide range of processing applications:

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

TMS320F240 version of this family is the one used in this application. It uses a 16-bit word length along with 32-bit registers for storing intermediate results, and has two hardware shifters available to scale numbers independent of the CPU.

The 'C24x DSP controllers take advantage of an existing set of peripheral functions which includes:

- Timers,
- Serial communications ports (SCI, SPI),
- Analog-to-digital converters (ADC),
- Event manager,
- System protection, such as low-voltage detection and watchdog timers.

To function as a system manager, a DSP must have robust on-chip I/O and other peripherals. The event manager of the '240 is application-optimized peripheral unit, coupled with the high-performance DSP core, enables the use of advanced control techniques for high-precision and high-efficiency full variable-speed control of motors. Included in the event manager are special pulse-width modulation (PWM) generation functions, such as a programmable dead-band function and a space vector PWM state machine for 3-phase motors that provides quite a high efficiency in the switching of power transistors. Three independent up/down timers, each with its own compare register, support the generation of asymmetric (non-centered) as well as symmetric (centered) PWM waveforms.

#### **8.1.4 Interface Card**

In order to convey information back and forth between the power stage and DSP an interface card has been designed. Moreover, suitable signal amplification, signal filtering and hardware protection properties are added to this interface card (see Appendix C).

The dc-link voltage is sensed with a voltage sensor (LV25\_P) on the interface card. The insulation property of the voltage sensor is quite sufficient to protect the digital circuit and low voltage analog circuit from high voltage part. The dc-link voltage value is sensed to re-build the phase voltages in the control software with the

information of duty cycles of the IGBTs. This sensor is not necessary for the closed-loop FOC algorithms unless the precise voltage information is required. Since the reference voltage values obtained from feedback information and used as input of SVPWM can be considered as actual dq-voltages. Especially, at low speed range, the voltage drops on the switches become significant and the reference voltages do not reflect the actual voltage values due to this voltage drop. In this case one must embed a voltage drop compensation module to the control algorithm or use a voltage sensor. Another aim of the voltage sensor is to sense the overcharge on the dc-link capacitors. If the voltage level exceeds the predefined limit that is determined by the user, a comparator gives an error signal. This error signal is used for immediate hardware interrupt and all the IGBTs are set to off-state. Finally, to discard the power resistance, the aim of which is to prevent the in-rush current at starting, the voltage level information is needed to operate the relay across this resistance. The voltage sensor also provides this voltage information whether it exceeds the adjusted voltage level or not. If this mechanism is employed then the relay will operate automatically after the start command in a very short time.

The other sensed variables are stator currents using current-sensor on the interface card. For this purpose LTS 25-NP current transducers are used. These sensors are capable of sensing AC, DC and mixed current waveforms. The sensor has multi-range current sensing options depending on the pin connections. The sensors use hall-effect phenomena to sense the current. They have excellent accuracy and very good linearity in the operating range. The output of these sensors is between 0-5V and unipolar. Since the ADCs on the DSP board cannot sense the negative voltage and requires signal between 0-5V, our current sensors eliminate extra hardware, and software modules due to its mentioned properties. Normally, one must add offset to the AC current signals to compensate the negative parts and then subtract this amount in the software. Furthermore, the current signals must be normalized between the 0-5V range using amplifiers before the ADCs. All of these procedures cause extra uncertainty that affects the accuracy of the sensed information.

In case of noisy phase currents, optional low-pass filters are placed on the interface card with 1kHz cut-off frequency. However, at high frequency range above 50Hz these filters may cause serious phase lagging problem. The outputs of the current transducers are also used to provide over-current protection. After determining

the over-current limit, the potentiometers in the protection circuit are set to this critical limit. In case of over-current problems a comparator gives error signal to set the IGBTs into off-state.

In addition, the PWM signals generated by DSP are amplified to make them compatible with the gate drive card inputs. For this purpose, six PWM signals are adjusted to 15V individually without any other change. Finally, all the errors, gate drive card errors, over-voltage error, over-current error, and an external error are *OR* gated. The single error output is assigned to control PWM-OFF circuit to set the all IGBTs to off-state in case of any fault.

The pictures of the interface card, DSP and inverter are given below.

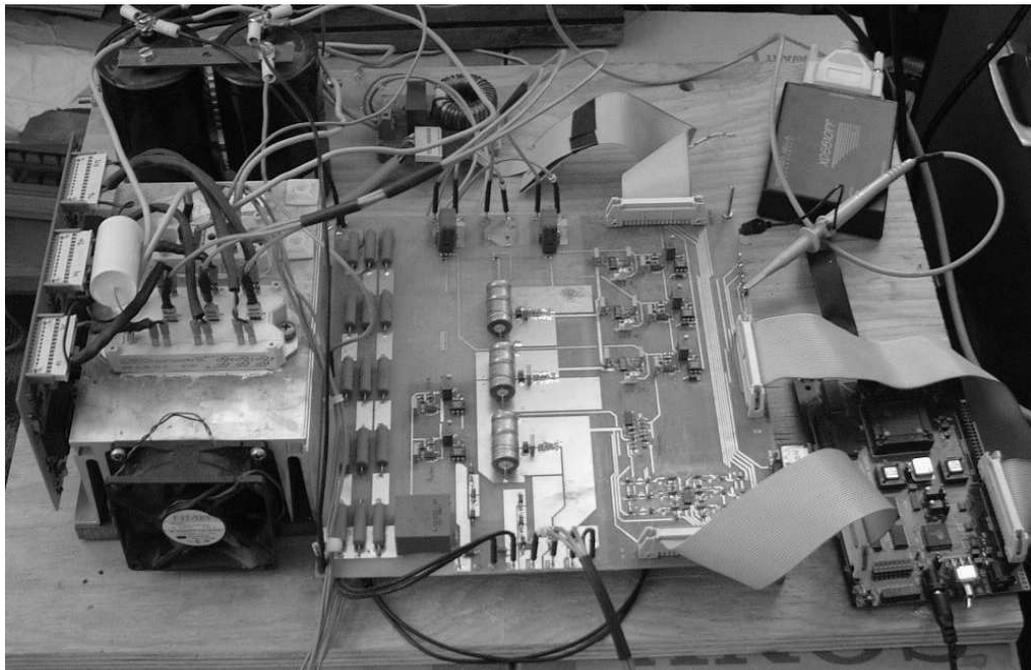


Fig.8.6- Experimental setup (Interface card, DSP and inverter)

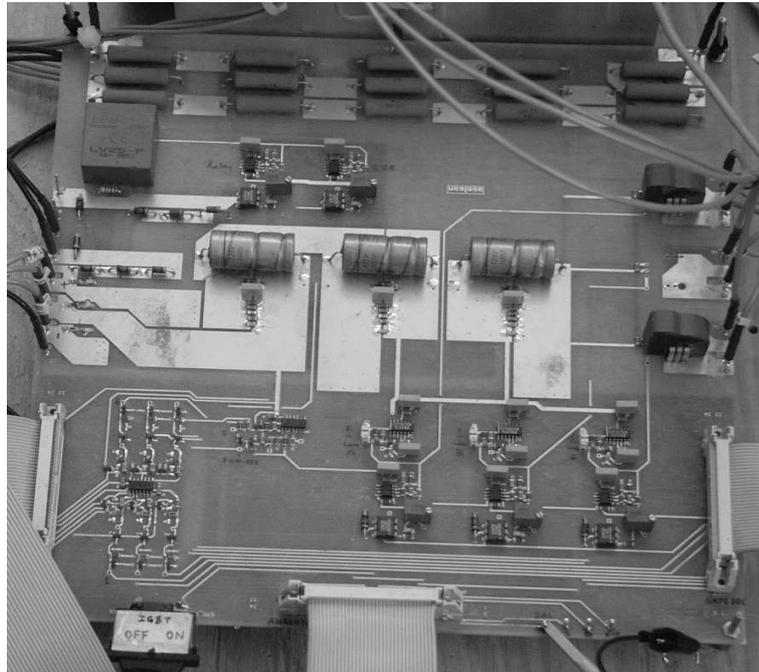


Fig. 8.7- Interface card

## 8.2 Software Overview

In this part general software flowchart will be explained. Furthermore, software modules and specific fixed-point numerical methods will be analyzed. The experimental outputs of the each module will be monitored to show that the modules run properly.

### 8.2.1 Software Organization

Overall algorithm of this project may be divided into two: initialization and the run time module as shown in Fig.8.8. The initialization module defines and initializes the software variables, constants and specific registers. Moreover, some of the look-up tables employed in the algorithm may also be addressed in this part. Initialized registers in the initialization module are watchdog timer registers, event manager registers, auxiliary register addressing, serial communication registers, clock registers. Some of these registers may be redefined in the specific modules if the modular algorithm is used. Among the registers above initialization of the serial communication registers is optional. The software constants and uninitialized software variables may be defined in this part or in the initialization of the each module.

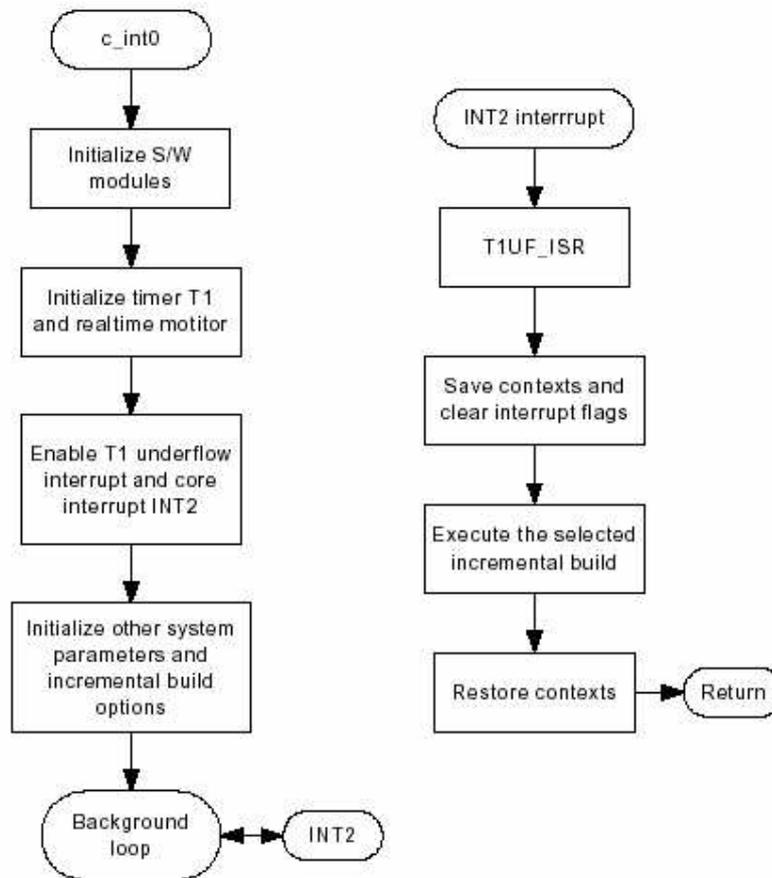


Fig.8.8- Software Flowchart

The code of the DSP TMS320 consists of many sections. The smallest unit of an object file is called a *section*. A section is a block of code or data that will ultimately occupy contiguous space in the memory map. Each section of an object-file is separate and distinct. Object files always contain three default sections:

- .text section** usually contains executable code.
- .data section** usually contains initialized data.
- .bss section** usually reserves space for uninitialized variables.

In addition, the assembler and linker allow you to create, name, and link named sections that are used like the *.data*, *.text*, and *.bss* sections.

There are two basic types of sections:

**Initialized sections** contain data or code. The *.text* and *.data* sections are initialized; named sections created with the *.sect* assembler directive are also initialized.

**Uninitialized sections** reserve space in the memory map for uninitialized data. The *.bss* section is uninitialized; named sections created with the *.usect* assembler directive are also uninitialized. In the initialization of the main code both of these sections may be used. Uninitialized sections reserve space in TMS320 memory; they are usually allocated into RAM. These sections have no actual contents in the object file; they simply reserve memory. A program can use this space at runtime for creating and storing variables.

Uninitialized data areas are built by using the *.bss* and *.usect* assembler directives. The *.bss* directive reserves space in the *.bss* section. The *.usect* directive reserves space in a specific uninitialized named section. Each time you invoke the *.bss* directive, the assembler reserves more space in the *.bss* section. Each time you invoke the *.usect* directive, the assembler reserves more space in the specified named section. While developing the code in modular strategy, one may name the variables according to the modules they are used. This helps to the code developer to classify his variables in certain clusters. Initialized sections contain executable code or initialized data. The contents of these sections are stored in the object file and placed in TMS320 memory when the program is loaded.

The initialization of the algorithm may involve both of the sections. The run-time module does not involve uninitialized section unless it is necessary. One may need to put *.text* section to the initialization of the algorithm to set the variables to zero or predefined constants. The initialization of the code will be called once at the beginning of the program. Thus the starting address of the initialization must be defined clearly and the code processing must start from this point. For this purpose DSP code has a specific addressing indicator. The symbol *\_c\_int0* is defined as the program entry point and the execution starts from this point. One must add necessary options to linker command if this automatic entry point is used.

Run time code is mainly composed in the *.text* section. Run time code may be generated either in the form of modules or one block of extended code. The modularity technique is preferred to debug the code in an easy way. Furthermore, the modules may be tested individually before building up the main code. In motor control applications in which the PWM technique is employed, the run time modules including overall control algorithms, real-time variable sensing, and final switching commands are synchronized with the sampling period as shown in Fig.8.9. In addition, synchronization of the PWM period is important for current sensing

accuracy. To get the current samples at the end of the PWM pulses will eliminate the switching frequency harmonic noise. This fact is out of the scope so the details will be skipped.

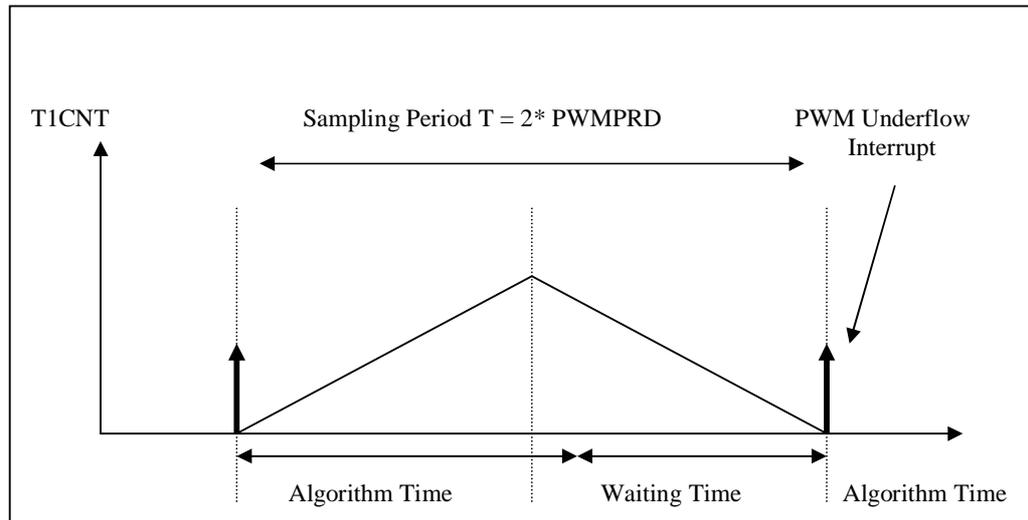


Fig.8.9- Overall FOC Algorithm Timing

The run-time code will be executed before  $2 * PWMPRD$  and then the program enters in an infinite loop. The program will leave the loop with PWM underflow interrupt that is active when the timer1 counter is zero. By following the timer1 counter which is programmed to continuous up down counting with a period of  $2 * PWMPRD$ , the main code is synchronized with the PWM pulses. One may develop a user interface code instead of infinite loop and communicate with the program adjust some of the coefficients such as PID and assign the DAC outputs to different software variables.

### 8.2.2 Base Values and PU model

Since the TMS320F240 is a fixed point DSP, PU (per unit) model of the motor is used to increase the accuracy and simplify the overall system. In a fixed point system, 16 bit words are assigned to variables and some of the bits are used for integer part of the number and some for floating part. To increase the accuracy, one must reserve as many bits as possible to floating part. This is also possible with PU system. If the PU parameters and variables are properly chosen, the integer part will be expected to be at most  $\pm 1$  and the number of the bits reserved for the floating part will be maximized.

While choosing the PU values, one may select ratings of the motor or multiple of the nominal values as base depending on the operating conditions. One must be careful about the transient regions, since the amount of the states may exceed the unity in transient state. Thus, the multiple of the nominal values may be adjusted according to the transient state peak values to limit the base values. However, this will give harm to the accuracy of the steady state computations. Due to this reason, while testing the performance of the observers, we run the system under light load conditions. The base quantities are calculated as shown below:

$$\begin{aligned}
 I_b &= K_1 * I_n \\
 V_b &= K_1 * V_n \\
 \omega_b &= 2\pi f_n \\
 \psi_b &= \frac{V_b}{\omega_b}
 \end{aligned} \tag{8.1}$$

where  $I_b$ ,  $V_b$ , are the maximum values of the phase nominal current and voltage;  $\omega_b$  is the electrical nominal rotor flux speed;  $\psi_b$  is the base flux and subscript  $n$  is expresses the nominal quantities of the same variable. The real time quantities are implemented in to the control thanks to the PU quantities, which are defined as follows:

$$\begin{aligned}
 i &= \frac{I}{I_b} \\
 v &= \frac{V}{V_b} \\
 \psi &= \frac{\Psi}{\Psi_b}
 \end{aligned} \tag{8.2}$$

### 8.2.3 Fixed-Point Arithmetic

In binary format, a number can be represented in signed magnitude, where the left-most bit represents the sign and the remaining bits represent the magnitude. Two's complement is an alternative form of representation used in most processors, including the TMS320. The representation of a positive number is the same in two's complement and in signed magnitude. Thus the first bit gives idea whether the number represented in signed magnitude is positive or negative.

In fixed-point operations  $Q_k$  format is used to represent the floating numbers. For example, in  $Q_{12}$  format, first bit is assigned for sign and three bits are assigned to represent the integer part and the rest of the 16 bits is assigned for floating part of the number. In  $Q_{15}$ , one bit is assigned for sign and the rest is assigned for floating part. Thus one can represent the number in the range of -1 to 1 excluding these to integers. So if the variables are normalized in a proper way those never exceed unity,  $Q_{15}$  format provides the best accuracy. If one uses numbers greater than unity then, the  $Q_k$  format must be rearranged to represent those numbers. For example, using  $Q_{12}$  format the numbers between the  $-7.999$  to  $+7.999$  with a less accuracy in the floating part since the floating part may be represented with fewer numbers when compared to the  $Q_{15}$  format. The resolution of the  $Q_{15}$  format is  $0.0000305$  ( $1/2^{15}$ ) where  $Q_{12}$  format is  $0.000244$  ( $1/2^{12}$ ). If the selected base values are in nominal values, than drive control quantities will not be greater than four times the nominal values (for the most part). In this case one may use  $Q_{12}$  format. On the other hand, if the value of the variables are guaranteed to be less than unity by choosing proper base quantities,  $Q_{15}$  format is preferred . The generalized representation of  $Q_k$  is given below:

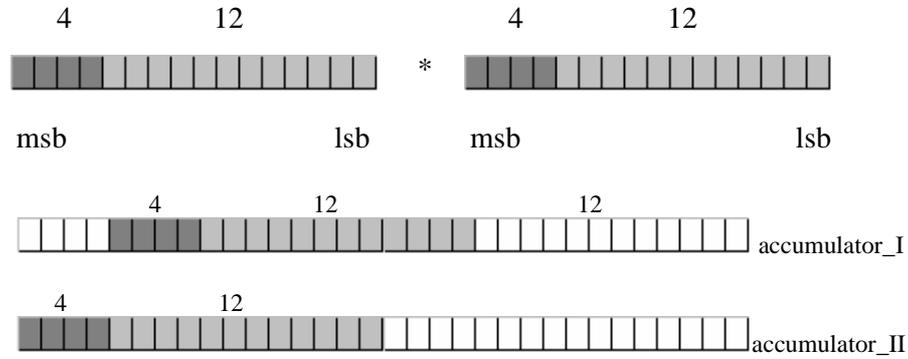
$$Z = -b_{15-k} * 2^{15-k} + b_{14-k} * 2^{14-k} + \dots + b_0 + b_{-1} * 2^{-1} + b_{-2} * 2^{-2} + \dots + b_{-k} * 2^{-k} \quad (8.3)$$

where  $k$  represent the number of bits representing floating part and  $b$  is the binary (0,1) quantities. Here are some examples to clarify the concept:

$$\begin{aligned} \pi &= 3.14159265 = 0011.0010\ 0100\ 0011 \quad \text{in } Q_{12} \text{ format (3243H)} \\ &= 011.0\ 0100\ 1000\ 0111 \quad \text{in } Q_{13} \text{ format (6486H)} \\ 0.0045 &= 0000.0000\ 0001\ 0010 \quad \text{in } Q_{12} \text{ format} \\ &= 0.000\ 0000\ 1001\ 0110 \quad \text{in } Q_{15} \text{ format} \end{aligned}$$

The summation of the numbers in  $Q_k$  format is the same as binary system summation. However, the multiplication has some tricks. In the multiplication of the  $Q_k$  format, the result is in the form of:  $Q_n * Q_m = Q_{n+m}$ . Since the multiplication operation is done in the 32 bit accumulator,  $Q_{n+m}$  may exceed to  $Q_{30}$  without any problem.

For example,  $Q_{12} * Q_{12} = Q_{24}$



After the multiplication  $Q_{24}$  is stored to the accumulator as shown in accumulator\_I. The right bits of the shaded part is neglected in accumulator\_I. Before storing to a 16-bit microprocessor word, the number in the accumulator must be left shifted as in the case of accumulator\_II and than the high word in the accumulator must be stored the associated word. In the multiplication of  $Q_k$  format  $n$  and  $m$  above may be any number between 0-15. However, 16-bit result must be aligned to either high or low word of the accumulator in the form of any  $Q_{15}$  format. The SXM must be set if any of the numbers in  $Q_k$  format is negative.

#### 8.2.4 FOC Software Modules

In the literature several vector control structures are proposed in which recently developed algorithms and modules can be adjusted. One of the classical sensorless closed-loop field-orientation structure block diagrams are shown in Fig.8.10. Although the major components of these structures remain the same, the difference underlies in some specific modules only. In this thesis we concentrated on the observer block to estimate either rotor-speed or rotor-flux. In this section, basic modules are defined and associated theoretical background is explained.



Background:  $pid\_out(k) = pid\_out(k-1) + (Kp + Ki T) e(k) - (Kp) e(k-1)$

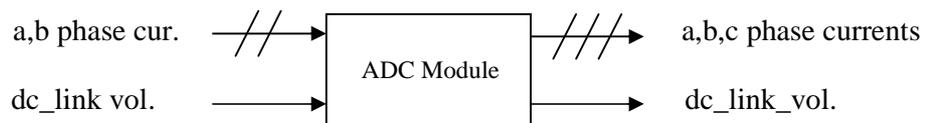
where

e: error between the inputs

Kp, Ki: proportional and integral coefficients respectively

T: sampling time

## 2- ADC



Description: ADC is a driver module that converts analog inputs into digital representations with programmable gains and offsets. In the study we also sensed the dc\_link voltage. Conversions are triggered by timer 1 underflow.

Inputs: ia, ib analog phase currents, analog dc\_link\_voltage (Q15)

Outputs: ia, ib digital phase currents, digital dc\_link\_voltage (Q15)

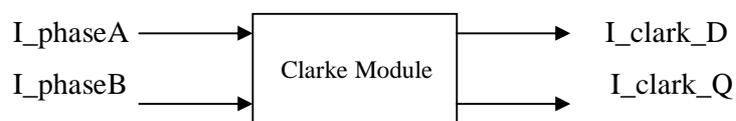
Background: ADC pins accept 0-5V only. Since the outputs of sensors on the interface card are between 0-5V, the voltage and current values are sensed directly.

Phase currents are embedded to the software according to the formula below:

$$I\_phaseA = I\_phaseA\_ADC * I\_A\_gain + I\_A\_offset$$

Since the sensed current values are unipolar, negative part is represented between 0-FFFF H and positive part is represented between 0-7FFF H in hex. 0-5V dc\_link voltage is represented between 0-7FFF H in hex. Due to Q15 conversion, XOR operation causes offset about 200H. Thus one must take into account this offset.

## 3-CLARK TRANSFORMATION



Description: It converts balanced three phase quantities into balanced two phase quadrature quantities.

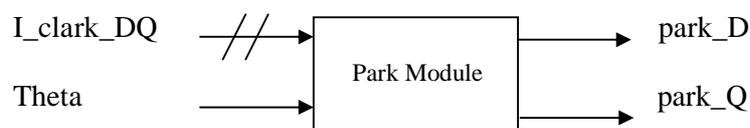
Inputs: ia, ib digital phase currents (Q15)

Outputs: clark\_D, clark\_Q (Q15)

Background:  $clark\_D = I\_phaseA$

$$clark\_Q = (I\_phaseA + 2 * I\_phaseB) / \sqrt{3}$$

#### 4-PARK TRANSFORMATION



Description: It converts vectors in balanced 2-phase orthogonal stationary system into orthogonal rotating reference frame.

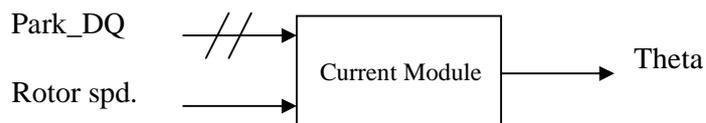
Inputs: clark\_D, clark\_Q (Q15)

Outputs: park\_D, park\_Q (Q15)

Background:  $park\_D = clark\_D * \cos\theta + clark\_Q * \sin\theta$

$$park\_Q = -clark\_D * \sin\theta + clark\_Q * \cos\theta$$

#### 5-CURRENT MODEL



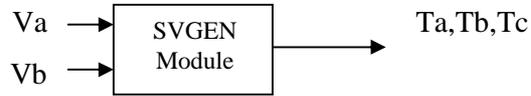
Description: It estimates the rotor flux position based on three inputs. These are the quadrature (isq) and direct (isd) axis components of the stator current in the orthogonal reference frame (output of PARK transform) and the rotor mechanical speed.

Inputs: park\_D, park\_Q, speed (Q15)

Outputs: theta (Q15)

**Background:** Theoretical background is given in chapter 2.

## 7-SVGEN



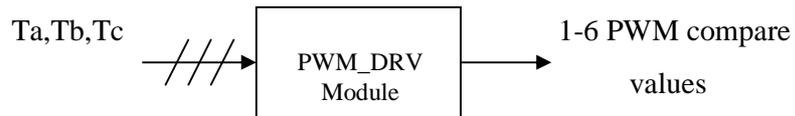
Description: It calculates the appropriate duty ratios needed to generate a given stator reference voltage using space vector PWM technique.

Inputs:  $V_q, V_d$  (Q15)

Outputs:  $T_a, T_b, T_c$  (Q15)

Background: Theoretical background is given in Chapter 4.

## 8-PWM DRIVE



Description: It uses the duty ratio information and calculates the compare values for generating PWM outputs. The compare values are used in the full compare unit in 24x/24xx event manager (EV). This also allows PWM period modulation.

Inputs:  $T_a, T_b, T_c$  (Q15)

Outputs: PWM1-6 compare values (Q15)

Background: Theoretical background is given in Chapter 4.

## 9-STATE OBSERVER

Description: It gives speed, flux information from sensed current and voltage values

Inputs:  $i_d, i_q, V_d, V_q$  (Q15)

Outputs: rotor speed, flux (Q15)

Background: Theoretical background is given in Chapters 5, 6, 7.

## **10-DAC**

Description: This module sends any assigned software variable to DAC outputs. Since DAC register format is 12-bit, in this module Q15 format is converted to equivalent 12-bit representation. This module allows monitoring four different software variables at the same time.

Inputs: any four-software variable (Q15)

Outputs: 0-5V equivalent analog representation of the variables.

## **11-Voltage Calculation**

Description: This software module calculates three phase voltages applied to the 3-ph motor (i.e., induction or synchronous motor) using the conventional voltage-source inverter. Three phase voltages can be reconstructed from the DC-bus voltage and three switching functions of the upper power switching devices of the inverter. In addition, this software module also includes the clarke transformation that converts three phase voltages into two stationary dq-axis voltages.

Inputs: three switching functions, dc\_link\_voltage (Q15)

Outputs: 3-phase voltages and 2-phase voltages (Q15)

Background: Theoretical base of this module is partially given in Chapter 4.

A complete state-observer performance test assembler code is given in Appendix D.

## CHAPTER 9

### CONCLUSIONS and FUTURE WORKS

#### 9.1 Conclusions

The focus of thesis has been the state estimation techniques for FOC of induction machine. Different state estimators utilized to observe dq-axis stator and rotor flux estimation, rotor speed, synchronous speed, rotor flux angle.

First, generalized dynamic mathematical model of the induction motor is studied in different reference frames. Induction motor mathematical model developed in stationary frame is investigated. Furthermore, analysis on this model is made including stability analysis of the model, transfer function determination and step response of the model. Next, using this motor model an indirect field orientation control is simulated and studied to gain familiarity with FOC. In addition, SVPWM algorithm used in the experiment is simulated and the results are confirmed experimentally.

As an estimator, Kalman filtering technique investigated to observe dq-axis rotor fluxes and rotor speed. For this purpose, appropriate mathematical model of induction machine is studied and discretized for real-time applications. Several simulations are illustrated to examine the performance of EKF. In addition, a novel Kalman filtering technique is introduced to observe the same states with the same motor model. These two techniques are proved to be appropriate for induction motor

state estimation with simulations and experimental results. A comparative approach is made to determine the superior technique.

An adaptive state observer, MRAS is tested on two different schemes to observe rotor speed. The stability analysis and continuous and discrete models of each scheme are investigated. The high performance of these schemes is shown in simulations and experimental results.

In addition, a flux observer with voltage model-current model combination is implemented. The outputs of this observer were fed to a open-loop speed estimator. Using these observers, dq-axis rotor-stator fluxes, rotor flux angle and rotor speed are estimated satisfactorily in the experiments.

There exist a number of criteria to determine the observer used in the FOC of induction motor in practice. If an observer, based on Kalman filter, is to be implemented then the microprocessor used in the system must be fast enough and it must have sufficiently large program memory. However, this is not a strict requirement for MRAS, since the computational burden of MRAS is relatively low compared to the Kalman filter.

Furthermore, one must know the noise content of the experimental setup to select the appropriate observer. If the sensed currents include high amount of measurement noise, then using MRAS will not be adequate choice since the low pass filter characteristics of the MRAS is worse than that of EKF. In addition, uncertainties related to the parameter deviation can be modeled by the user in EKF. Whereas MRAS is expected to compensate these uncertainties by the aid of it's closed-loop structure. Therefore, for the practical cases where the uncertainties are significant, EKF will provide better state estimations.

Either the number of the states or the number of the parameters to be estimated by the observer is another criterion to select the true observer. One can estimate the whole state vector by using an observer based on EKF. MRAS, however, provides the estimation of only one state or one parameter instantaneously. Therefore this criterion depends on the requirements of FOC algorithms. One may use MRAS not as a state observer but an online parameter tuning tool that tunes different state observers.

In addition to criteria mentioned above, mathematical complexity and difficulty of both the EKF and UKF are significant disadvantages when compared to the MRAS. Because, the mathematics used in MRAS is not as complicated as in Kalman filtering techniques. Thus, it is easier to analyze and deal with MRAS.

Finally, in this work it is shown that, instead of EKF using UKF will avoid costly Jacobian matrix calculations and will give better results under the same operating conditions.

## **9.2 Future Work**

The simulations and experimental works show the great promise of the studied methods in this thesis. However, due to equipment limitations these methods are not tested over a wide speed and torque ranges. Further, the experimental work is required to implement these techniques in the entire torque speed range of the induction motor.

Using the three different speed frames (arbitrary, stationary, synchronous) discussed in Chapter 3, equivalent circuit models for air-gap flux, stator flux and rotor flux may also be derived.

In addition to the implemented SVPWM algorithm in Chapter 4, over-modulation techniques may also be implemented in order to utilize entire dc link voltage.

A reduced-order motor model may be applied to the both EKF and UKF to decrease the computational burden. Furthermore, these observers may be used for on-line parameter estimation.

To solve the stability problems of MRAS, different methods may be used such as gain scheduling. MRAS may also be reconfigured to estimate the parameters of the motor in real time.

## REFERENCES

- [1] B. K. Bose, "Power Electronics and AC Drives" , Prentice Hall, 1986
- [2] P.Vas "Sensorless Vector and Direct Torque Control" New York: Oxford University Press,(1998)
- [3] I. Boldea and S. A. Nasar, "Vector Control of AC Drives", CRC Press, 1992.
- [4] D. W. Novotny and T. A. Lipo, "Vector Control and Dynamics of AC Drives", Oxford University Press Inc., Oxford, New York, 1997.
- [5] R.D. Doncker and D.W. Novotny "The universal field oriented controller", IEEE Tran. IA, vol. 30 no .1, pp. 92-100, January, 1994
- [6] R.. Lorenz "Tuning of the field oriented induction motor controller for high performance applications", IEEE Tran. IA, vol. 22, pp. 293-297, 1986
- [7] K.B. Nordin and D.W. Novotny "The influence of motor parameter deviations in feed forward field orientation drive systems" IEEE Tran. IA, vol. 21, pp.1009-1015, July 1985
- [8] R.Krishnan, F.C.Doran "Study of parameter sensitivity in high performance inverter-fed induction motor drive systems", IEEE Tran. IA, vol. 23,pp.623-635, 1987
- [9] P.L. Jansen and R.D.Lorenz "Transducer less position and velocity estimation in induction an salient ac machines", IEEE Tran. IA, vol. 31, no.2, pp.240-247, Mar/April, 1995
- [10] R.Gabriel, W.Leonhard and C. Nordby " Field oriented control of standard AC motor using microprocessor", IEEE Tran. IA, vol.16, no.2, pp.186.192, 1980
- [11] H.Jun and W.Bin " New integration Algorithms for Estimating Motor Flux over a Wide Speed Range" IEEE Tran. IE, vol. 13, no.5, pp.969-977, 1998
- [12] P.L. Jansen and R.D.Lorenz "Observer based direct field orientation: Analysis and comparison of alternative methods", IEEE Tran. IA, vol. 30, no.4, pp.945-953, Jul/Aug, 1994

- [13] G.C. Versghese and S.R. Sanders “Observers for flux estimation in the induction machine” IEEE Tran. IE, vol. 35, no.1, pp.85-94, 1998
- [14] Lennart Harneffors “Design and analysis of general rotor-flux oriented vector control system” IEEE Tran. IE, vol. 48, no.2, pp.383-390, 2001
- [15] G. Yang and T. Chin “ Adaptive-Speed identification scheme for a vector controlled speed sensorless inverter induction motor drive” ,IEEE Tran. IA, vol. 29, no.4, pp.820-825, 1993
- [16] B.Robyns, B.Frederique, “A methodology to determine gains of induction motor flux observers based on a theoretical parameter sensitivity analysis”, IEEE Tran. PE, vol. 15, no.6, pp.983-995, 2000
- [17] A.M. Trzynadlowski, “The Field Oriented Principle in Control of induction Motors”, Kluwer Academic Publisher, Norwell, MA, 1994
- [18] M.Elbuluk,N.Langovsky and D.Kankam “Design and Implementation of a Closed Loop Observer and Adaptive Controller for Induction Motor Drives” IEEE Tran. IA vol. 34 no. 3 pp. 435-443, May/June 1998
- [19] H.Tajima, Y.Hori “Speed Sensorless Field Orientation Control of the Induction Machine” IEEE Tran. IA vol. 29, no. 1, pp. 175-180, Feb.1993
- [20] Colin Schauder “Adaptive Speed Identification for Vector Control of Induction Motor without Rotational Transducers” IEEE Tran. IA vol. 28, no. 5, pp. 1054-1061,Oct.1992
- [21] F.Z.Peng, T.Fukao “Robust Speed Identification for Speed Sensorless Vector Control of Induction Motors” IEEE Tran. IA vol. 30, no. 5, pp.1234-1239, Oct.1994
- [22] L.Zhen and L. Xu “Sensorless Field Orientation Control of Induction Machines Based on Mutual MRAS Scheme” IEEE Tran. IE vol. 45 no. 5, pp. 824-831, October 1998
- [23] M.Elbuluk, L.Tong and I.Husain “Neural Network Based Model Reference Adaptive Systems for High Performance Motor Drives and Motion Control” IEEE Tran. IA vol. 38, no. 3 pp. 879-886 May/June 2002
- [24] K.K.Shyu, H.J. Shieh and S.S. Fu “Model Reference Adaptive Speed Control for Induction Motor Drive Using Neural Network ”, IEEE Tran. IE vol. 45, no. 1 pp. 180-182, Feb. 1998
- [25] C.Attainese, Alfonso Damiano “Induction Motor Drive Parameters Identification” IEEE Tran. PE vol. 13, no. 6, pp. 1112-1123,Nov. 1998

- [26] H.Sugimoto and S.Tamai “Secondary Resistance Identification of an Induction Motor Applied Model Reference Adaptive System and its Characteristics” *IEEE Tran. IA* vol. 23, no. 2 pp. 296-303 March/April 1987
- [27] H.W.Kim and S.K.Sul “ A New Motor Speed Estimator using Kalman Filter in Low Speed Range”, *IEEE Tran. IE* vol. 43, no. 4, pp. 498-504, Aug. 1996
- [28] Y.R.Kim, S.K.Sul and M.H.Park “Speed Sensorless Vector Control of Induction Motor Using Extended Kalman Filter”, *IEEE Tran. IA* vol. 30, no.5 pp. 1225-1233, Oct. 1994
- [29] L.C.Zai, C.L.De Marco, T.A.Lipo “An Extended Kalman Filter Approach to Rotor Time Constant Measurement in PWM Induction Motor Drives” *IEEE Tran. IA* vol. 28, no.1, pp. 96-104, Jan/Feb 1992
- [30] P.Vas "Parameter Estimation, Condition Monitoring and Diagnosis of Electrical Machines" New York: O.University Press,(1993)
- [31] D.Atkinson, P.Acarnley, J.W.Finch “Observers for Induction Motor State and Parameter Est.” *IEEE Tran. IA* vol.27, no. 6, pp.1119-1127 ,Dec. 1991
- [32] L.Salvatore, S.Stassi and L.Tarchioni “A New EKF Based Algorithm for Flux Estimation in Induction Machines” *IEEE Tran. IE* vol. 40, no. 5, pp. 496-504, Oct. 1993
- [33] B.Akın, U.Orguner and A.Ersak “State Estimation of Induction Motor Using Unscented Kalman Filter” *IEEE CCA Conference, Istanbul, 2003*
- [34] S.Julier and J.K.Uhlmann, “A new extension of the Kalman filter to non linear systems,” Available: <http://www.robots.ox.ac.uk>.
- [35] S.Julier, J.K.Uhlmann, and H.F.Durrant-Whyte, “A new method for the nonlinear transformation of means and covariances in filters and estimators,” *IEEE Trans. Automatic Control*, vol.45, pp. 477-482, March 2000
- [36] S.Julier, J.K.Uhlmann, and H.F.Durrant-Whyte, “A new approach for filtering nonlinear systems,” Available : <http://www.robots.ox.ac.uk>
- [37] K. Hurst, T. Habetler, G. Griva and F.Profumo “Zero-Speed Tachless IM Torque Control: Simply a matter of Stator voltage Integration”, *IEEE Tran. IA* vol. 34, no. 4, pp. 790-794, 1998
- [38] M.Shin, D. Hyun, S.Cho and S.Choe “An Improved Stator Flux Estimation for Speed Sensorless Stator Control of Induction Motors”, *IEEE Tran. IA* vol.15, no.2, pp. 312-317, 1993

- [39] Ong, Chee-Mun., "Dynamic Simulation of Electric Machinery", N.J, Printice Hall, 1998
- [40] B.Mutluer, "Vector Control of Asynchronous Motors" M.S Thesis EE Dept. METU Dec. 1999
- [41] J.Holtz, "Pulsewidth Modulation-A Survey", IEEE Tran. IE, vol .39 no .5, pp. 410-419, Dec. 1992
- [42] N.Celanovic, D.Boroyevich "A Fast Space Vector Modulation Algorithm for Multilevel Three Phase Converters". IEEE Tran. IA, vol .37 no .2, pp. 637-641, April 2001
- [43] H.V. Broeck, H.C. Skudelny, G.V.Stanke "Analysis and Realization of a Pulsewidth Modulator Based on Voltage Space Vectors" IEEE Tran. IA, vol .24 no .1, pp. 142-150, Jan.1988
- [44] Texas Instruments, "SVPWM with TMS320C2XX", Literature number: SPRA 524, 2001
- [45] Analog Devices, "Implementing Space Vector Modulation with the ADMC300", Literature number: AN300-17
- [46] J.Holtz, "Sensorless Control of AC Machines" IEEE Press Book, 1996
- [47] L.B.Brahim and A.Kawamura "A Fully Digitized Field –Oriented Controlled Induction Motor Drive Using Only Current Sensors", IEEE Tran. IE, vol .39 no .3, pp. 241-249, June 1992
- [48] Sensorless Control with Kalman Filter on TMS320 FP DSP, TI, 1997
- [49] M.Marwali, A.Keyhani, W.Tjanaka "Implementation of Indirect Vector Control on an Integrated Digital Signal Processor - Based System" IEEE Tran. En.Conv. vol. 20, Oct.1998
- [50] S.W.Matthew, W.Dunnigan and W.Williams "Modeling and Simulation of Induction Machine Vector Control with Rotor Resistance Identification" IEEE PE Tran. vol.12, no. 13, pp. 495-506, May 1997
- [51] K.Hurst, T.G.Habetler "A Self Tuning Closed Loop Flux Observer for Sensorless Torque Control of Standard Induction Machines" IEEE Tran. PE vol.12, no. 5, pp.807-815 ,Sep.1997
- [52] M. Ouhrouche, "EKF-Based Estimation of Rotor Flux, Speed and Rotor Resistance in Cage Induction Motor Sensorless Drive" IASTED International Conference Modeling and Simulation (Pennsilvania) 2000

- [53] G.Soto, E.Mendes, A.Rezak “Extended Reduced Order Observer for Rotor Flux and Rotor Resistance in Induction Machines” ICEM (Istanbul) 1992
- [54] L.Salvatore, S.Stasi, F.Cupertino “Speed Sensorless Vector Control of Induction Motors using EKF” ICEM (Istanbul) 2000
- [55] MATLAB User’s Guide, 1998
- [56] T.Lund, “Sensorless Control of Induction Machine” M.S Thesis EPE Dept. T.D.U Aug. 2001
- [57] H.Butler "Model Reference Adaptive Systems" London: Printice Hall,(1992)
- [58] M.Liaw, K.Hsiang and J.Lin “A Discrete Adaptive Field Oriented Induction Motor Drive” IEEE Tran. PE vol. 7 no. 2, pp. 411-419, April 1992
- [59] S.Bowes and J.Li “New Robust Adaptive Control Algorithm for High-Performance AC Drives” IEEE Tran. IE vol. 47, no. 2 pp. 325-336, Apr.2000
- [60] A.Consoli, E. Cerruto, A.Raciti and A.Testa “Fuzzy Adaptive Vector Control of Induction Motor Drives” IEEE Tran. PE vol. 12, no. 6, pp. 1028-1039 Nov. 1997
- [61] X.Yu, M.Dunnigan and B.Williams “A Novel Rotor Resistance Identification Method of an Indirect Rotor Flux Oriented Controlled Ind. Mach. System” IEEE Tran. PE vol. 17 no. 3, pp. 353-364, May. 2002
- [62] J.Moreira and T. Lipo “A New Method for Rotor Time Constant Tuning in Indirect Field Oriented Control” IEEE Tran. PE vol. 8, no. 4, pp. 626-631, Oct. 1993
- [63] H.Sugimoto and S.Tamai “Secondary Resistance Identification of an Induction Motor Applied Model Reference Adaptive System and its Characteristics” IEEE Tran. IA vol. 23, no. 2 pp. 296-303 March/April 1987
- [64] T.Rowan, R.Kerkman and D.Leggate “A Simple On-Line Adaptation for Indirect Field Orientation of an Induction Machine” IEEE Tran. IA vol. 27, no. 4, pp. 720-727 July/August 1991
- [65] Texas Instruments, “Digital Motor Control Software Library”, Literature number: SPRU 485 August 2001
- [66] C. Lascu, I. Boldea, and F. Blaabjerg, “A modified direct torque control for induction motor sensorless drive” IEEE Trans. Ind. Appl., vol. 36, no. 1, pp. 122-130, Jan./Feb. 2000.
- [67] Texas Instruments, “ACI3-4 3-phase Induction Motor Control”, App. Note, 15 Nov., 2000

## Appendix A

### S-function M-file:

```
function [sys,x0]=kalm(t,x,u,flag)
%This s;function is used to simulate a kalman filter in
%field reference frame

global Q a11 a13 a14 a31 a34 a33 b11 a15 a25 a35 a45 b53;
global d_Fa Gm Cm L W V V1 Ts B Gv w n ;
global L11 L12 L21 L22 L31 L32 L41 L42 L51 L52 L;
if flag == 0
    % Initialization
    %Motor parameters

    Rs=Rs_;
    LIs=LIs_;
    Rr=Rr_;
    LIr=LIs_;
    xm = xm_
    L0= xm_;

    %Initial conditions

    Wr=0;
    We=0;
    phiqr=0;
    phidr=0;
    %The state noise
    V_ix=(V_ix_)^2;
    V_iy=(V_ix_)^2;
    V_phix=(V_phix_)^2;
    V_phiy=(V_phix_)^2;
    V_wr=(V_wr_);
    %The output noise
    W_ix=(W_ix)^2;
    W_iy=(W_ix)^2;

    %Sampling time

    Ts=200e-6;

    %State space model for the motor

    Lr=L0+LIr;
    Ls=L0+LIs;
    Taur=Lr/Rr;
    sigma=1-L0^2/(Ls*Lr);

    a11=-(Rs/(sigma*Ls)+(1-sigma)/(sigma*Taur));
```

```

a13=L0/(sigma*Ts*Lr*Taur);
a14=L0/(sigma*Ts*Lr);
a31=L0/Taur;
a33=-1/Taur;
b11=1/(sigma*Ts);
a15=L0/(sigma*Ts*Lr);
a25=-L0/(sigma*Ts*Lr);

A=[a11 We a13 a14*Wr
-We a11 -a14*Wr a13
a31 0 a33 (We-Wr)
0 a31 (Wr-We) a33];

B=[b11 0 ; 0 b11 ; 0 0; 0 0];
C =[1 0 0 0; 0 1 0 0];

F=eye(4,4)+Ts*A;

G=B*Ts;

d_F12=[a15*phiqr
a25*phidr
-phiqr
phidr ];

d_F=[F d_F12
0 0 0 0 1];

Gm=[G
0 0];

Cm=[C [0;0]]

V=[
V_ix 0 0 0 0
0 V_iy 0 0 0
0 0 V_phix 0 0
0 0 0 V_phiy 0
0 0 0 0 V_wr
];

W=[
W_ix 0
0 W_iy
];

%Discrete system matrices
%Extended model
%Jacobian matrix
%Total Jacobian matrix

%The discrete Riccati equation

%[Q,L,G,RR]=dare(d_F',Cm',V,W);
[Q,L,G,RR]=care(A',C',diag([V_ix V_iy V_phix V_phiy]),W);
Q=[Q,zeros(4,1);zeros(1,5)];
Q(5,5)=V_wr;

```

```

%Q=V

L=G'
n=0; %Counter
x0 =[0 0 0 0 0];
sys =[0,5,5,5,0,0];
elseif flag==2%Update
n=n+1;
U=[u(1);u(2)];
Y=[u(3);u(4)];
We=u(5);

%Extended A matrix

A=[a11 We a13 a14*x(5) a15*x(4)
-We a11 -a14*x(5) a13 a25*x(3)
a31 0 a33 (We-x(5)) -x(4)
0 a31 (x(5)-We) a33 x(3)
0 0 0 0 0];
d_F=eye(5)+Ts*A;

%State vector time update

x_1=[d_F(1:4,1:4)*x(1:4) ;x(5)]+Gm*U;
%x_1=d_F*x+Gm*U;

%Covariance time update
Q_1=d_F*Q*d_F'+V;
%Kalman gain update
L=Q_1*Cm'*inv(Cm*Q_1*Cm'+W);

%State vector measurement update
x=x_1+L*(Y-Cm*x_1);
%Covariance measurement update
Q=Q_1-L*Cm*Q_1;

%Uncomment to log innovation of Kalman
gains
%(makes execution slower)

%L11(n)=L(1,1);
%L12(n)=L(1,2);
%L21(n)=L(2,1);
%L22(n)=L(2,2);
%L31(n)=L(3,1);
%L32(n)=L(3,2);
%L41(n)=L(4,1);
%L42(n)=L(4,2);
%L51(n)=L(5,1);
%L52(n)=L(5,2);

sys=x;
elseif flag==3
sys=x;
elseif flag==9
save final
sys=[];
end

```

## Appendix B

As with Lyapunov's method, an adaptive law designed using hyperstability theory is guaranteed to be stable. In the hyperstability approach the designer has to propose an adaptive law, and with the aid of hyperstability theory one can check whether this law gives a stable result. In general, a model reference adaptive speed estimator system can be represented by an equivalent non-linear feedback system which comprises a feed-forward time invariant linear subsystem and a feedback non-linear time varying subsystem. The first part normally contains the reference model, and its output is the error signal to be used in the adaptation. The second part contains the adaptive laws and has an output  $W$ . This division is illustrated in Fig.A .

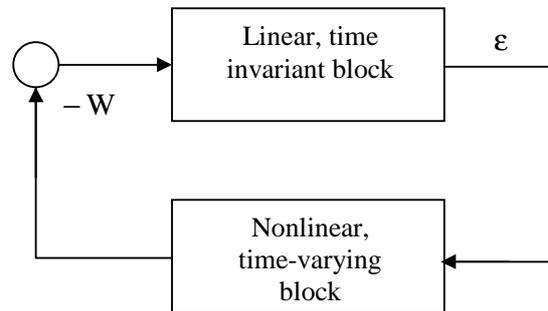


Fig A. Division of error equation into time invariant linear part and a time varying nonlinear part

Usually, the input  $-W$  of the linear block equals the multiplication of the parameter error  $\theta$  and the signal vector  $\xi$  used in the adaptation:  $-W = \theta^T \xi$ . Hyperstability theory guarantees an asymptotically stable system if both the linear and nonlinear parts satisfy a positivity condition. A controllable, linear system with input  $u$  and output  $y$  :

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= C^T x \end{aligned}$$

with a transfer function :

$$H(s) = \frac{Y(s)}{U(s)} = C^T (sI - A)^{-1} B$$

is said to be *positive real* (PR) if  $\text{Re}[H(s)]$  equal or greater than zero. Hence the real part of the transfer function can never become negative as long as the real part of  $s$  is larger than or equal to zero. According to hyperstability theory the linear part  $H$  must be strictly positive real (SPR) which means that the real part of  $H(j\omega)$  is larger than zero for all  $\omega > 0$  thus Nyquist diagram of  $H(j\omega)$  must lie in the right half of the complex plane, including the imaginary axis. This implies that the number of poles and zeros in  $H(s)$  differs at most by 1, and the phase shift is never larger than  $90^\circ$ . The nonlinear part must satisfy Popov's integral inequality, which states that a positive constant  $\gamma_0^2$  exists such that :

$$\int_0^{t_1} \varepsilon \bullet W dt \geq -\gamma_0^2 \quad \text{for all } t_1 \geq 0$$

This requirement is also denoted the passivity requirement. Observing nonlinear part as an electrical network, the inequality can be shown to state that the amount of energy output by the nonlinear system is never larger than the sum of the incoming energy and the energy stored in the system. The energy in the system depends on the external input of power and on the power generation in the system

$$\frac{d}{dt} [\text{stored energy}] = [\text{ext. power input}] + [\text{int. power generation}]$$

Considering  $\varepsilon$ , the input, as voltage and  $W$  as output, current, the external power input equals  $\varepsilon W$ . If the internal power generation is negative, the system is said to be dissipative or strictly passive. If the internal power generation is less than or equal to zero, the system is passive. Strict passivity is equivalent to SPR and asymptotic stability. The main result using positivity and passivity concepts is that any parallel combination of passive blocks is also passive. A feedback combination of two passive blocks in which at least one is strictly passive. This is of great interest in hyperstability theory, in which an SPR (and hence strictly passive) linear block is connected to a passive nonlinear block in a feed back configuration. This combination is strictly passive (and hence asymptotically stable).

## APPENDIX C

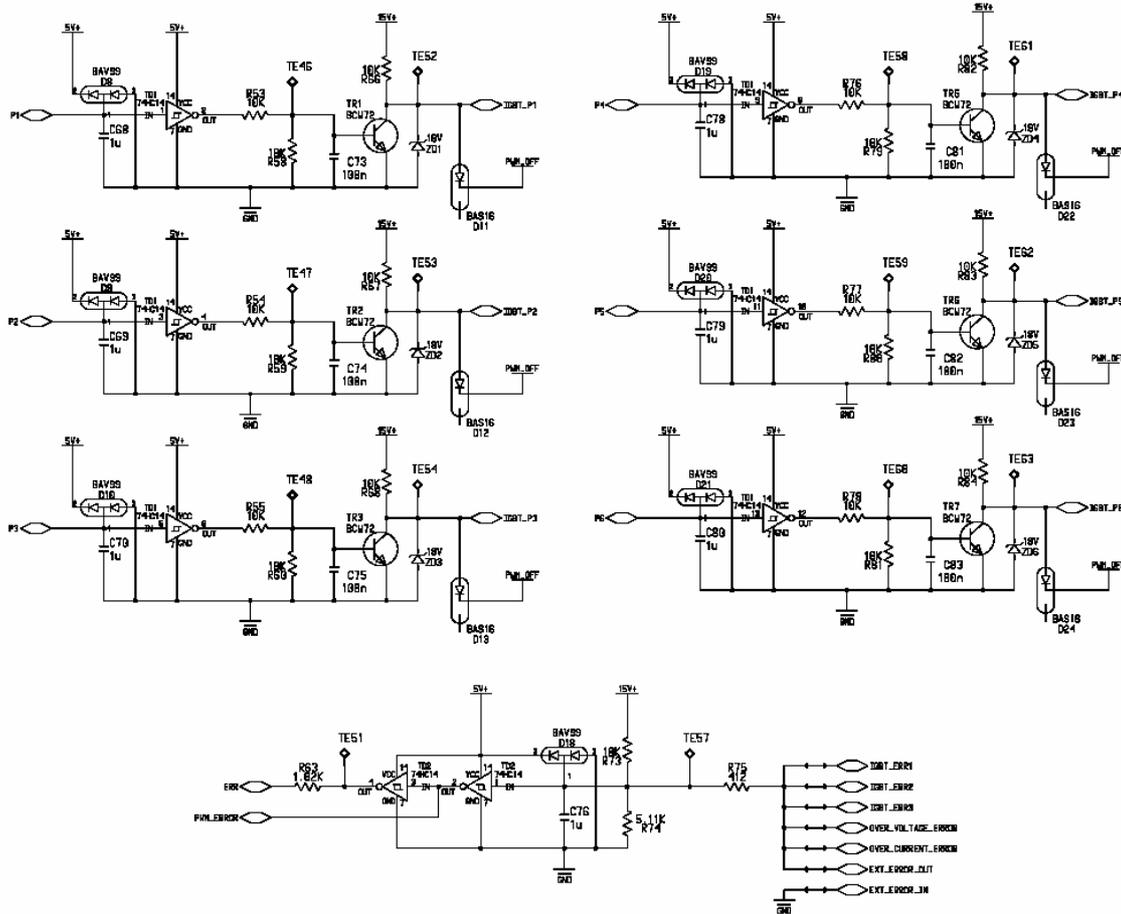


Fig. C.1- Amplification unit of PWM pulses with hardware interrupt cct. on the interface card

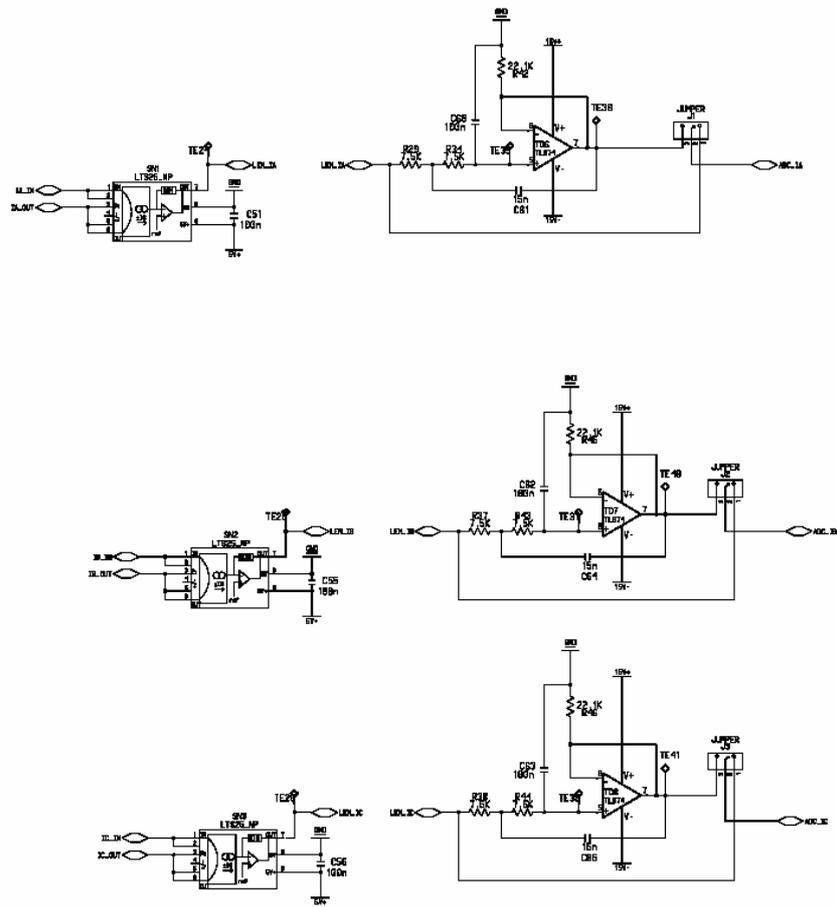


Fig. C.2- Current sensing unit and low-pass filters on the interface card



## Appendix D

```
=====
;
; Description:      Code of Observer Test
;
;=====
; History:
;-----
; 15.06.2003
;=====

;*****
;                               SYSTEM OPTIONS
;*****

PWM_PERIOD .set 100           ; PWM period in uS (10KHz)
TlPER_     .set PWM_PERIOD*10 ; *1000nS/(2*50nS)
;*****

;-----
; External references
;-----

    .include    "x24x_app.h"

    .ref  SYS_INIT

    .ref  RAMP_GEN, RAMP_GEN_INIT
        .ref  rmp_gain, rmp_offset, rmp_freq
        .ref  step_angle_max
        .ref  rmp_out

    .ref  I_PARK, I_PARK_INIT
        .ref  ipark_D, ipark_Q, theta_ip
        .ref  ipark_d, ipark_q

    .ref  DAC_VIEW_DRV, DAC_VIEW_DRV_INIT
        .ref  DAC_IPTR0, DAC_IPTR1, DAC_IPTR2, DAC_IPTR3

    .ref  SVGEN_DQ, SVGEN_DQ_INIT
        .ref  Ualfa, Ubeta
        .ref  Ta, Tb, Tc
```

```

.ref FC_PWM_DRV,FC_PWM_DRV_INIT
.ref     Mfunc_c1,Mfunc_c2,Mfunc_c3,Mfunc_p
.ref     n_period

.ref CLARKE, CLARKE_INIT
.ref  clark_a, clark_b
.ref  clark_d, clark_q

.ref PARK, PARK_INIT
.ref  park_d, park_q, theta_p
.ref  park_D, park_Q

.ref ILEG2_DCBUS_DRV, ILEG2_DCBUS_DRV_INIT

.ref  Ch_sel, Imeas_a_gain, Imeas_b_gain, Vdc_meas_gain

.ref  Imeas_a_offset, Imeas_b_offset, Vdc_meas_offset
.ref  Imeas_a, Imeas_b, Imeas_c, Vdc_meas

.ref PHASE_VOLTAGE_CALC
.ref  PHASE_VOLTAGE_CALC_INIT
.ref  Mfunc_V1, Mfunc_V2
.ref  Mfunc_V3, DC_bus
.ref  Vphase_A,Vphase_B,Vphase_C
.ref  Vdirect,Vquadra

.ref OBSERVER, OBSERVER_INIT
.ref  obs_1, obs_2
.ref  obs_3, obs_4

.ref TRAPEZOIDAL_GEN, TRAPEZOIDAL_GEN_INIT
.ref  trap_min, trap_max, trap_dly_max
.ref  trap_out

.ref PID_REG3_ID,PID_REG3_ID_INIT
.ref  id_fdb,id_ref
.ref  ud_out,e_d

.ref PID_REG3_IQ,PID_REG3_IQ_INIT
.ref  iq_fdb,iq_ref
.ref  uq_out,e_q

```

```

;=====
; V E C T O R   T A B L E
;=====

        .sect "vectors"
        .def  _c_int0

RESET   B   _c_int0           ; 00
INT1    B   PHANTOM           ; 02
INT2    B   T1_PERIOD_ISR     ; 04
INT3    B   PHANTOM           ; 06
INT4    B   PHANTOM           ; 08
INT5    B   PHANTOM           ; 0A
INT6    B   PHANTOM           ; 0C

;=====
; M A I N   C O D E   -   s t a r t s   h e r e
;=====

        .text
_c_int0:
        CALL  SYS_INIT
        CALL  FC_PWM_DRV_INIT
        CALL  DAC_VIEW_DRV_INIT
        CALL  RAMP_GEN_INIT
        CALL  I_PARK_INIT
        CALL  SVGEN_DQ_INIT
        CALL  ILEG2_DCBUS_DRV_INIT
        CALL  CLARKE_INIT
        CALL  PARK_INIT
        CALL  PHASE_VOLTAGE_CALC_INIT
        CALL  ACI_FE_INIT
        CALL  ACI_SE_INIT
        CALL  TRAPEZOIDAL_GEN_INIT
        CALL  PID_REG3_IQ_INIT
        CALL  PID_REG3_ID_INIT

;-----
;   Variables initialization
;-----
        LDP   #n_period
        SPLK  #T1PER_,n_period; initialize the PWM period to 10kHz

;-----
; System Interrupt Init.
;-----
        ;Event Manager
        POINT_EV
        SPLK  #0000001000000000b,IMRA ;Enable T1 Underflow Int

        SPLK  #0FFFFh,IFRA           ; Clear all Group A interrupt flags
        SPLK  #0FFFFh,IFRB           ; Clear all Group B interrupt flags
        SPLK  #0FFFFh,IFRC           ; Clear all Group C interrupt flags

```

```

POINT_PG0

SPLK #0000000001000010b,IMR ;En Int lvl 3,7 (T2 ISR)
      ;5432109876543210

SPLK #0FFFFh, IFR          ;Clear any pending Ints
EINT                               ;Enable global Ints
POINT_B0

;=====
MAIN:          ;Main system background loop
;=====

M_1
  NOP
  NOP
  NOP
  CLRC XF
  B      MAIN
;=====

;=====
; Routine Name: T1_PERIOD_ISR   Routine Type: ISR
;=====

T1_PERIOD_ISR:
;Context save regs
  MAR  *,AR1 ;AR1 is stack pointer
  MAR  *+      ;skip one position
  SST  #1, *+  ;save ST1
  SST  #0, *+  ;save ST0
  SACH *+      ;save acc high
  SACL *       ;save acc low

  POINT_EV
  SPLK #0FFFFh,IFRA          SETC  XF
  SETC  SXM                  ; set sign extension mode
  CLRC  OVM                  ; clear overflow mode

```

```

;=====
;Start main section of ISR
;=====

; Current leg measurement, Ileg2drv module
LDP #Imeas_a_gain
SPLK #4000H,Imeas_a_gain ;Q13-->2
SPLK #4000H,Imeas_b_gain ;Q13-->2
SPLK #2000H,Vdc_meas_gain

CALL ILEG2_DCBUS_DRV

LDP #Imeas_a
LACC Imeas_a,2 ;4.22amp rms 6amp peak
SACL Imeas_a
LACC Imeas_b,2
SACL Imeas_b

; Clarke module
LDP #clark_a
BLDD #Imeas_a,clark_a
BLDD #Imeas_b,clark_b

CALL CLARKE

; PARK module
LDP #park_d
BLDD #clark_d,park_d
BLDD #clark_q,park_q
BLDD #rmp_out,theta_ip

CALL PARK

; Ramp generation module

LDP #id_fdb
BLDD #park_D,id_fdb
SPLK #trap_out,id_ref

CALL PID_REG3_ID

LDP #iq_fdb
BLDD #park_Q,iq_fdb
SPLK #3500h,iq_ref

CALL PID_REG3_IQ

ldp #trap_min
splk #02500h,trap_min
splk #04000h,trap_max
splk #8,trap_dly_max

CALL TRAPEZOIDAL_GEN

; Inverse-Park module

```

```

LDP    #theta_ip
BLDD  #rmp_out,theta_ip
BLDD  #ud_out,ipark_D
splk  #uq_out,ipark_Q

CALL   I_PARK

                                           ; Space-Vector DQ module
LDP    #Ualfa
BLDD  #ipark_d,Ualfa
BLDD  #ipark_q,Ubeta

CALL   SVGEN_DQ
                                           ; PWM driver
LDP    #Mfunc_c1
BLDD  #Ta,Mfunc_c1
BLDD  #Tb,Mfunc_c2
BLDD  #Tc,Mfunc_c3
SPLK  #07FFFh,Mfunc_p

CALL   FC_PWM_DRV

LDP    #DC_bus
BLDD  #Ta,Mfunc_V1
BLDD  #Tb,Mfunc_V2
BLDD  #Tc,Mfunc_V3
BLDD  #Vdc_meas,DC_bus

CALL   PHASE_VOLTAGE_CALC

LDP    #Vdirect
LACC  Vdirect,1
SACL  Vdirect
LACC  Vquadra,1
SACL  Vquadra

LDP    #obs_1
BLDD  #Vdirect,obs_1
BLDD  #Vquadra,obs_2
BLDD  #clark_d,obs_3
BLDD  #clark_q,obs_4

CALL   OBSERVER

```

```

*****
*                               DAC_LIST                               *
*****

```

```

LDP      #DAC_IPTR0
SPLK    #clark_d,DAC_IPTR0
LDP      #DAC_IPTR1
SPLK    #clark_q,DAC_IPTR1
LDP      #DAC_IPTR0
SPLK    #Vphase_A,DAC_IPTR2
LDP      #DAC_IPTR1
SPLK    #Vphase_B,DAC_IPTR3

CALL     DAC_VIEW_DRV

```

```

;=====
;End main section of ISR
;=====

```

```

;Context restore regs
END_ISR:

```

```

POINT_PG0

MAR     *, AR1      ;make stack pointer active
LACL   *-          ;Restore Acc low
ADDH   *-          ;Restore Acc high
LST    #0, *-      ;load ST0
LST    #1, *-      ;load ST1
EINT
RET

```

```

;=====
; I S R - PHANTOM
;=====
PHANTOM B PHANTOM

```

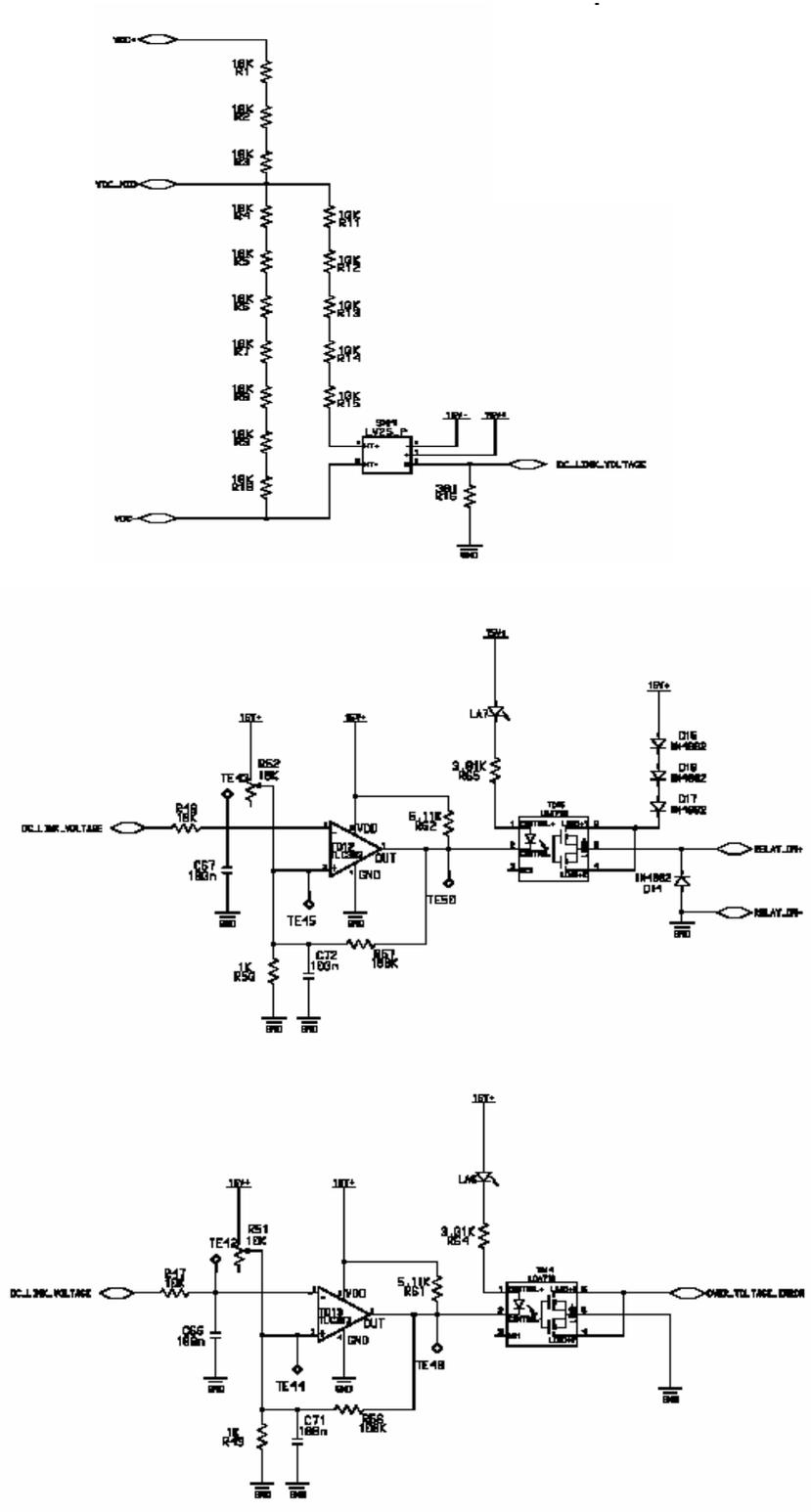


Fig.C.4- Dc link voltage sensing unit and over-voltage error generating unit on the interface card

## Appendix D

```
=====
;
; Description:      Code of Observer Test
;
;=====
; History:
;-----
; 15.06.2003
;=====

;*****
;                               SYSTEM OPTIONS
;*****

PWM_PERIOD .set 100           ; PWM period in uS (10KHz)
TlPER_    .set PWM_PERIOD*10   ; *1000nS/(2*50nS)
;*****

;-----
; External references
;-----

    .include    "x24x_app.h"

    .ref  SYS_INIT

    .ref  RAMP_GEN, RAMP_GEN_INIT
        .ref  rmp_gain, rmp_offset, rmp_freq
        .ref  step_angle_max
        .ref  rmp_out

    .ref  I_PARK, I_PARK_INIT
        .ref  ipark_D, ipark_Q, theta_ip
        .ref  ipark_d, ipark_q

    .ref  DAC_VIEW_DRV, DAC_VIEW_DRV_INIT
        .ref  DAC_IPTR0, DAC_IPTR1, DAC_IPTR2, DAC_IPTR3

    .ref  SVGEN_DQ, SVGEN_DQ_INIT
        .ref  Ualfa, Ubeta
        .ref  Ta, Tb, Tc
```

```

.ref FC_PWM_DRV,FC_PWM_DRV_INIT
.ref     Mfunc_c1,Mfunc_c2,Mfunc_c3,Mfunc_p
.ref     n_period

.ref CLARKE, CLARKE_INIT
.ref  clark_a, clark_b
.ref  clark_d, clark_q

.ref PARK, PARK_INIT
.ref  park_d, park_q, theta_p
.ref  park_D, park_Q

.ref ILEG2_DCBUS_DRV, ILEG2_DCBUS_DRV_INIT

.ref  Ch_sel, Imeas_a_gain, Imeas_b_gain, Vdc_meas_gain

.ref  Imeas_a_offset, Imeas_b_offset, Vdc_meas_offset
.ref  Imeas_a, Imeas_b, Imeas_c, Vdc_meas

.ref PHASE_VOLTAGE_CALC
.ref  PHASE_VOLTAGE_CALC_INIT
.ref  Mfunc_V1, Mfunc_V2
.ref  Mfunc_V3, DC_bus
.ref  Vphase_A,Vphase_B,Vphase_C
.ref  Vdirect,Vquadra

.ref OBSERVER, OBSERVER_INIT
.ref  obs_1, obs_2
.ref  obs_3, obs_4

.ref TRAPEZOIDAL_GEN, TRAPEZOIDAL_GEN_INIT
.ref  trap_min, trap_max, trap_dly_max
.ref  trap_out

.ref PID_REG3_ID,PID_REG3_ID_INIT
.ref  id_fdb,id_ref
.ref  ud_out,e_d

.ref PID_REG3_IQ,PID_REG3_IQ_INIT
.ref  iq_fdb,iq_ref
.ref  uq_out,e_q

```

```

;=====
; V E C T O R   T A B L E
;=====

        .sect "vectors"
        .def  _c_int0

RESET   B   _c_int0           ; 00
INT1    B   PHANTOM           ; 02
INT2    B   T1_PERIOD_ISR     ; 04
INT3    B   PHANTOM           ; 06
INT4    B   PHANTOM           ; 08
INT5    B   PHANTOM           ; 0A
INT6    B   PHANTOM           ; 0C

;=====
; M A I N   C O D E   -   s t a r t s   h e r e
;=====

        .text
_c_int0:
        CALL  SYS_INIT
        CALL  FC_PWM_DRV_INIT
        CALL  DAC_VIEW_DRV_INIT
        CALL  RAMP_GEN_INIT
        CALL  I_PARK_INIT
        CALL  SVGEN_DQ_INIT
        CALL  ILEG2_DCBUS_DRV_INIT
        CALL  CLARKE_INIT
        CALL  PARK_INIT
        CALL  PHASE_VOLTAGE_CALC_INIT
        CALL  ACI_FE_INIT
        CALL  ACI_SE_INIT
        CALL  TRAPEZOIDAL_GEN_INIT
        CALL  PID_REG3_IQ_INIT
        CALL  PID_REG3_ID_INIT

;-----
;   Variables initialization
;-----
        LDP   #n_period
        SPLK  #T1PER_,n_period; initialize the PWM period to 10kHz

;-----
; System Interrupt Init.
;-----
        ;Event Manager
        POINT_EV
        SPLK  #0000001000000000b,IMRA ;Enable T1 Underflow Int

        SPLK  #0FFFFh,IFRA           ; Clear all Group A interrupt flags
        SPLK  #0FFFFh,IFRB           ; Clear all Group B interrupt flags
        SPLK  #0FFFFh,IFRC           ; Clear all Group C interrupt flags

```

```

POINT_PG0

SPLK #0000000001000010b,IMR ;En Int lvl 3,7 (T2 ISR)
      ;5432109876543210

SPLK #0FFFFh, IFR           ;Clear any pending Ints
EINT                               ;Enable global Ints
POINT_B0

;=====
MAIN:           ;Main system background loop
;=====

M_1
    NOP
    NOP
    NOP
    CLRC XF
    B     MAIN
;=====

;=====
; Routine Name: T1_PERIOD_ISR   Routine Type: ISR
;=====

T1_PERIOD_ISR:
;Context save regs
    MAR *,AR1 ;AR1 is stack pointer
    MAR *+     ;skip one position
    SST #1, *+ ;save ST1
    SST #0, *+ ;save ST0
    SACH *+    ;save acc high
    SACL *     ;save acc low

    POINT_EV
    SPLK #0FFFFh,IFRA           SETC     XF
    SETC  SXM                   ; set sign extension mode
    CLRC  OVM                   ; clear overflow mode

```

```

;=====
;Start main section of ISR
;=====

; Current leg measurement, Ileg2drv module
LDP #Imeas_a_gain
SPLK #4000H,Imeas_a_gain ;Q13-->2
SPLK #4000H,Imeas_b_gain ;Q13-->2
SPLK #2000H,Vdc_meas_gain

CALL ILEG2_DCBUS_DRV

LDP #Imeas_a
LACC Imeas_a,2 ;4.22amp rms 6amp peak
SACL Imeas_a
LACC Imeas_b,2
SACL Imeas_b

; Clarke module
LDP #clark_a
BLDD #Imeas_a,clark_a
BLDD #Imeas_b,clark_b

CALL CLARKE

; PARK module
LDP #park_d
BLDD #clark_d,park_d
BLDD #clark_q,park_q
BLDD #rmp_out,theta_ip

CALL PARK

; Ramp generation module

LDP #id_fdb
BLDD #park_D,id_fdb
SPLK #trap_out,id_ref

CALL PID_REG3_ID

LDP #iq_fdb
BLDD #park_Q,iq_fdb
SPLK #3500h,iq_ref

CALL PID_REG3_IQ

ldp #trap_min
splk #02500h,trap_min
splk #04000h,trap_max
splk #8,trap_dly_max

CALL TRAPEZOIDAL_GEN

; Inverse-Park module

```

```

LDP    #theta_ip
BLDD  #rmp_out,theta_ip
BLDD  #ud_out,ipark_D
splk  #uq_out,ipark_Q

CALL   I_PARK

                                           ; Space-Vector DQ module
LDP    #Ualfa
BLDD  #ipark_d,Ualfa
BLDD  #ipark_q,Ubeta

CALL   SVGEN_DQ
                                           ; PWM driver
LDP    #Mfunc_c1
BLDD  #Ta,Mfunc_c1
BLDD  #Tb,Mfunc_c2
BLDD  #Tc,Mfunc_c3
SPLK  #07FFFh,Mfunc_p

CALL   FC_PWM_DRV

LDP    #DC_bus
BLDD  #Ta,Mfunc_V1
BLDD  #Tb,Mfunc_V2
BLDD  #Tc,Mfunc_V3
BLDD  #Vdc_meas,DC_bus

CALL   PHASE_VOLTAGE_CALC

LDP    #Vdirect
LACC  Vdirect,1
SACL  Vdirect
LACC  Vquadra,1
SACL  Vquadra

LDP    #obs_1
BLDD  #Vdirect,obs_1
BLDD  #Vquadra,obs_2
BLDD  #clark_d,obs_3
BLDD  #clark_q,obs_4

CALL   OBSERVER

```

```

*****
*                               DAC_LIST                               *
*****

```

```

LDP      #DAC_IPTR0
SPLK    #clark_d,DAC_IPTR0
LDP      #DAC_IPTR1
SPLK    #clark_q,DAC_IPTR1
LDP      #DAC_IPTR0
SPLK    #Vphase_A,DAC_IPTR2
LDP      #DAC_IPTR1
SPLK    #Vphase_B,DAC_IPTR3

CALL    DAC_VIEW_DRV

```

```

;=====
;End main section of ISR
;=====

```

```

;Context restore regs
END_ISR:

```

```

POINT_PG0

MAR     *, AR1      ;make stack pointer active
LACL   *-          ;Restore Acc low
ADDH   *-          ;Restore Acc high
LST    #0, *-      ;load ST0
LST    #1, *-      ;load ST1
EINT
RET

```

```

;=====
; I S R - PHANTOM
;=====
PHANTOM B PHANTOM

```