SIMPLIFIED FUZZY LOGIC CONTROLLER BASED INDIRECT VECTOR CONTROL OF AN INDUCTION MOTOR DRIVE

MAX KARMAZIN

NOTE TO USERS

Simplified Fuzzy Logic Controller Based Indirect Vector Control of an Induction Motor Drive

by

Max Karmazin

A thesis submitted to the school of graduate studies in
partial fulfillment of the requirements for the degree of
Master of Engineering

Faculty of Engineering and Applied Science
Memorial University of Newfoundland

December 2003

St. John's Newfoundland Canada

Your file Votre référence
ISBN: 0-612-99084-2
Our file Notre référence
ISBN: 0-612-99084-2

Abstract

Modern motor controllers are based on the concept of machine intelligence (MI). These intelligent controllers have been employed increasingly in advanced electric motor drives and in particular, in present-day induction motor drives. These systems attempt to emulate the work of human logic, which is a very effective approach for applications with poorly defined mathematical model. Such intelligent systems grant high robustness and adaptive behaviour, as well as high efficiency, high torque to current ratio and high power to weight ratio. These systems are also called high performance drive systems (HPDS). This is why an intelligent control of electric motor drives and induction motors in particular, has been in the scope of attention of electrical engineers in recent years. Of three major types of MI controllers, the fuzzy logic controller (FLC) is used to overcome parameter variations, sudden speed or load changes and other non-linear disturbances. This work employs also a technique called field-oriented control of induction motor (IM) to achieve high performance of the drive controlled by the MI-based motor controller.

This thesis presents development and implementation of an integral field oriented intelligent control of an induction motor drive using fuzzy logic controller. An analytical model of an induction motor drive has been developed. Extensive simulations were carried out at rated speed, lower than rated speed and over the

rated speed. An induction motor slip gain tuning scheme is coupled with the developed simplified fuzzy logic controller to achieve high performance and reduce computational burden as well as complexity. Simulations and experimental data also assess the realisation of proposed combined controller. Conventional proportional-integral-derivative (PID) based controllers are used to compare and to confirm better performances and dynamic characteristics of the induction motor (IM) controlled by the developed controller.

A digital signal processing (DSP) board DS1102 and laboratory induction motor were used to successfully implement the complete vector control scheme. The test results agree with simulated results for different dynamic operating conditions. The efficacy of the proposed fuzzy logic based control scheme has been successfully established.

Acknowledgements

I would like to express my most sincere gratitude and appreciation to my principal supervisor, Professor M. A. Rahman for his encouragement, influence and advice throughout the preparation of this thesis. His wise guidance has helped me learn the peculiarities of modern engineering research and scientific study.

I would also like to express my thanks to my co-supervisor, Dr. M. Tariq Iqbal, for valuable discussions and useful suggestions. Help from Mr. Richard Newman, the technical staff of the Faculty of Engineering and Applied Science is acknowledged for their assistance. Special thanks should be given to Dr. T. S. Radwan and Mr. Casey Butt for their help and some useful ideas as well as discussions.

I would like also to take this opportunity to sincerely thank Dr. Rahman for his financial support, which made this work possible.

Finally, my dedication is due to my parents, Mr. Vasyl Karmazin and Tatiana Karmazin, and my wife Anna for their constant encouragement and support, without which it would not have been possible for me to complete this study. Also, my son Alexander should be noted separately for giving me inspiration and all the wonderful surprises he is capable of.

Contents

List of Figures

List of Symbols

v_a, v_b, and v_c	a, b and c phase voltages, respectively
i_a, i_b, and i_c	actual a, b and c phase currents, respectively
i_a^*, i_b^*, and i_c^*	command a, b and c phase currents, respectively
v_d	d-axis voltage
v_q	q-axis voltage
i_d	d-axis current
i_q	q-axis current
R_s	stator resistance
R_r	rotor resistance
L_s	stator self-inductance
L_r	rotor self-inductance
L_m	mutual inductance
X_{ls}	stator leakage reactance
X_{lr}	rotor leakage reactance
X_m	magnetizing reactance
ω_m	mechanical speed
ω_s	stator angular frequency
ω_r	actual rotor speed
ω_r^*	motor command speed
$\Delta\omega_r$	error between actual and command speeds ($\omega_r^* - \omega_r$)
ω_e	synchronous speed of rotating magnetic field

Δe	change in speed error
θ_r	rotor position
Ψ_{qs}	quadrature axis stator flux linkage
Ψ_{ds}	direct axis stator flux linkage
Ψ_{qr}	quadrature axis rotor flux linkage
Ψ_{dr}	direct axis rotor flux linkage
F_{qs}	quadrature axis stator flux linkage in terms of IM base frequency
F_{ds}	direct axis stator flux linkage in terms of IM base frequency
F_{qr}	quadrature axis rotor flux linkage in terms of IM base frequency
F_{dr}	direct axis rotor flux linkage in terms of IM base frequency
$\theta_r(0)$	initial rotor position
Θ_s	IM slip angle
P	differential operator (d/dt)
P	number of pole pairs
T_e	developed electromagnetic torque
T_L	load torque
J_m	rotor inertia constant
B_m	friction damping coefficient
V_B	dc bus voltage of inverter

Chapter 1

Introduction

1.1 General Introduction

More than one half (60%) of the total electrical energy produced in developed countries is converted into mechanical energy using electric motors, freeing the society from the burden of physical labour. Among many types of the motors, three-phase induction machines still enjoy the same unparalleled popularity as they did a century ago. An electrical torque developed in the rotor of an electric motor drives the mechanical rotating system at required speed. At least 90% of contemporary industrial drive systems employ induction motors using variable speed drives where accurate control and fast speed response are often crucial requirements.

In modern days, ac motor drives are used in a multitude of industrial and process applications requiring high performances. In high-performance drive systems, the motor speed should closely follow a specified reference trajectory regardless of any load disturbances, parameter variations, and model system uncertainties. The responses of the entire system are critically controlled through the intelligent

uses of advanced technologies and systems. The modern electric motor drive has to have high quickness of torque, position and speed responses, high energy conversion efficiency, response robustness to parameter variations and detuning, load and torque disturbances, high reliability, low weight and minimum cost.

Most of the motors are uncontrolled, but the share of adjustable speed induction motor drives that are fed from the power electronic converters is steadily increasing and day-by-day they are phasing out dc drives in favour of ac drives. Some sources estimate that more then 50 billion dollars could be saved annually on energy cost by replacing all uncontrolled motors with controlled ones. However, control of induction machines is a challenging task, much harder to implement than the control of dc motors. Two major difficulties are the necessity of providing adjustable-frequency voltage and the non-linearity and complexity of the analytical model of the motor, worsened by parameters uncertainty and instability.

1.1.1 General Review of Electric Motor Drives

In the last century direct current (dc) machines were often employed for a variety of variable speeds drives. The output torque and speed of a dc machine can be easily changed by controlling the armature current or voltage from a controlled rectifier, which results in good performance and response to the wide range of speed and torque change both in steady state and transients [1]. Nonetheless, the dc machines are structurally more complicated than the ac machines because they require considerable maintenance of its commutators, brushes and brush gears. The dc machines are also unsafe to use in explosive environments. These major disadvantages significantly limit the applications of dc machines in many areas. Alterna-

tively, construction of an ac machine, particularly the induction machine (IM) is robust, simple and relatively cheap. Induction motors have traditionally been considered the workhorses of industry, because of their low cost, raggedness, reliability, and reasonable efficiency. From the time of their invention ac motors have been favoured for the constant speed drives. Sufficient speed control by variable voltage and/or frequency supply was difficult until the power electronics did come on scene. Nowadays they can also have high performance applications for variable speed drives [2,3].

Despite positive features there are some limitations of the IM. It always runs at a lagging power factor because of the nature of induced rotor field voltage from the stator side. It cannot run at synchronous speed. The IM drive system always has a power loss due to slip. The real time implementation of the IM drives often needs accurate estimation of motor parameters and sophisticated modelling with relatively complex control circuitry. On the other hand, synchronous motor runs at synchronous speed, has no power loss due to slip and it's control is not that complex, because their field current can be controlled from the rotor side. The negative features of synchronous motor include the need of additional power supply for the dc field excitation, additional maintenance of brush gears and slip rings. The quest to develop an electric motor drive with the characteristics of the synchronous machine and as simple in construction and maintenance as induction machine as well as recent development of high-energy permanent magnet materials like neodymium-boron-iron (Nd-B-Fe), samarium-cobalt (Sm-Co) lead to creation of interior permanent magnet (IPM) motor. Its positive features include as high power to weight ratio as well high torque to current ratio, low noise, high efficiency and robustness. Unlike the wire-wound synchronous motor, the field excitation is

maintained by means of the permanent magnets, eliminating any need of extra power supply for the field windings, and associated slip rings and brush gear assembly. Therefore power loss in excitation windings, bringing the cost down and reliability up [4,5]. The cost and secure availability of new permanent magnets like Nd-B-Fe are concerns for wide spread acceptance of IPM motors.

Despite the fact that the operation principles of an induction motor are continued to be unchanged, great progress has been made on modern induction motor drives over the years, particularly in the last two decades. Contemporary induction motors are much lighter and smaller, with enhanced reliability and efficiency as well as an average life span of about 12 years.

Traditionally, motor control is handled by fixed gain proportional-integral (PI) and proportional-integral-derivative (PID) controllers. However, the fixed-gain controllers are very sensitive to parameter variations, load and system disturbances, temperature changes, etc. This means the controller parameters have to be constantly adjusted. The problem of change in parameters can be solved by several adaptive control techniques such as Model Reference Adaptive Control (MRAC), Sliding – Mode Control (SMC), Variable Structure Control (VSC), Self-tuning PI controllers, etc.

In recent years, controllers based on principles of machine intelligence (MI) have increasingly been employed in advanced induction machines. Machine intelligence systems try to emulate the human brain, and thanks to their robustness and adaptively changing control, they are particularly effective in applications characterized by complex of poorly defined mathematical models. Three types of MI controllers stand out: (a) neural network controller, (b) fuzzy logic controllers, and (c) neuro-fuzzy controllers, combining operating principles of the neural and fuzzy

controllers. In order to achieve high performance, field-oriented control of induction motor (IM) drive is employed in this thesis.

1.1.2 Survey of the Field Oriented Control

The concept of field orientation (also called vector control), proposed by Hasse [6] in 1969 using an indirect method, and by Blaschke [7] in 1971, using a direct method, constitutes one of the most important paradigms in the theory and practice of control of induction motors [8 - 11]. The objective of field orientation is to make the induction motor emulate the separately excited dc motor as a source of adjustable torque, in which the field flux and the armature current are orthogonal. For an ideal dc machine, the torque in vector expression is given as:

$$T_M = K_T I_a I_f \tag{1.1}$$

where I_a is armature current related to the torque and I_f is the field current related to the flux. These two variables are orthogonal or decoupled vectors. Since the flux component I_f is decoupled from its torque component I_a, the torque sensitivity remains at maximum in both the steady and transient states. The field oriented control of an ac motor is essentially transforming a dynamic structure of an ac motor model into that of a dc motor model.

Intelligent controllers of the induction motor are used because the decoupling characteristics of vector-controlled IM are adversely affected by the parameter changes in the motor.

1.2 Induction Motors

1.2.1 General

Due to simplicity and reliability the induction motors still have the unmatched popularity in industry. The large part of these machines has inaccessible squirrel-cage rotors with no moving contacts, brushes or slip rings of any kind. Additional ruggedness is provided by the "squirrel-cage" itself, which consists of conducting metal bars. It can run at high speeds and withstand heavy overloads. The wound-rotor induction machines are not widely used. They are employed in cases when existence and accessibility of rotor windings via slip rings or brushes gives some needed advantages. Simplest examples of such layouts are high power rated machines, that use rheostats connected to the rotor windings at start up, disconnecting them and shorting rotor winding after the operating speed is reached. This is done to reduce the rotor current at the start.

The slip power recovery and cascade systems, in which excess electric power is taken from the rotor and used in the supply line for speed control, are increasingly rare, and having been phased out by controlled squirrel-cage induction motor drives.

1.2.2 Construction of Three-phase IM

Basic subsystems of an induction machine are the stator and rotor. Most part of IM are made of a stationary stator and rotating rotor. The stator core is housed inside the motor frame, often ribbed outside for better cooling. It is formed by stacking thin (0.3 to 0.5 mm) electrical steel or soft-iron laminations, which are

screwed together. The rotor core, built around a metal shaft for mechanical energy transmission, is also made of iron laminations. At the end of the rotor the cooling fins are made by casting, on the other end often a cooling fan is placed. Insulating lacquer is used to cover individual laminations for eddy-current losses reduction. The bedplates, end shields and a terminal box are mounted onto the case. In the centre of the end shields bearings are installed to support the shaft of the rotor. Stator terminals that provide an access to the stator windings can be star- or delta-connected, which are insulated from one another.

A three-phase distributed winding of copper (Cu) or aluminium (Al) insulated conductors connected at the ends is placed in slots on the periphery of the core. Number of slots is evenly distributed amongst the phases. In the case of squirrel-cage rotor, rotor windings consist of uniformly spaced axial bars connected at both ends by end rings and slightly skewed with respect to the longitudinal axis of the motor. Small rotors under 50 cm diameter are usually produced by die-cast technology from aluminium. The shapes of the slots can be teardrop, deep bar or double cage. No insulation is made between the bars and the rotor slots. By making double-cage rotors a combination of good efficiency and high starting torque can be achieved. Another crucial feature is the width of an airgap. Small airgap is better for the mutual electromagnetic induction between rotor and stator windings, typically from 0.5 to 0.8 mm for motors in the range of 10 to 100 kW and from 0.35 to 0.50 for motors up to 10 kW, as examples.

An equivalent circuit of an induction motor drive model in d-q axis stationary reference frame is given on Fig. 1.1.

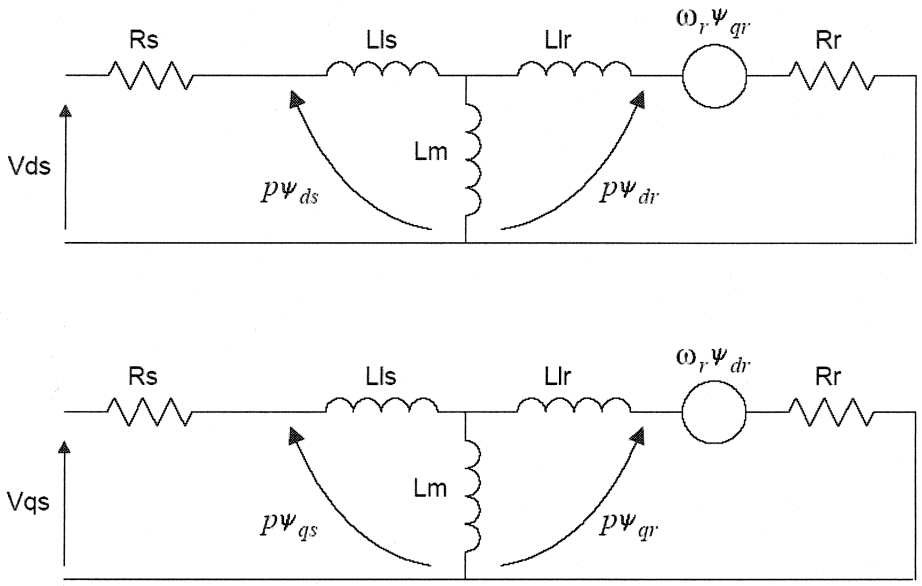


Fig. 1.1. The induction motor model d-q axis equivalent circuit in stationary reference frame.

The dynamic induction machine model in stationary reference frame is derived from the dynamic model in synchronously rotating reference frame. It is covered in chapter 2.2 describing mathematical modeling of the IM drive.

1.2.3 Classification of IM

Induction machines are distinguished by how well the inside of the motor is sealed from the ambient air. They can be on open-frame, partly closed and totally enclosed. Totally enclosed motors can work in extremely harsh environments and explosive atmospheres such as deep mines with high explosive gas concentration.

To meet the range of needs in the industry the squirrel-cage IMs are also classified into types characterized by their torque-speed curves. The evaluation of the effective rotor resistance is the most effective way of this classification. Four classes of machines are distinguished. Class A has low rotor resistance and because of that, high starting current and low starting torque. These machines have highest running efficiency at low slip. Class B machines are designed with higher rotor leakage inductance and have their starting current and torque lower then Class A machines, but better slip characteristics, therefore are considered for the major share of constant-speed industrial drives. Class C and Class D machines have higher rotor resistance and therefore higher starting torque and lower starting current. The recent high-efficiency machines are put into Class E category.

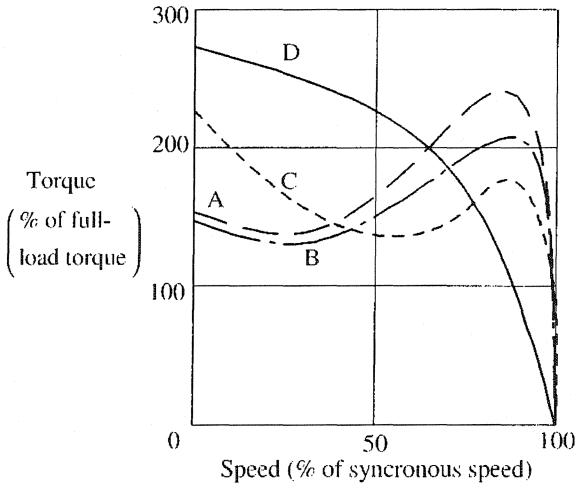


Fig. 1.2. Typical torque-speed curves for NEMA Design A, B, C, D motors.

1.3 Induction Motor Drives

1.3.1 Current State of Controllers for IM Drives

The induction motor (IM) drives have been the workhorses of the industry for many decades, used in the variety of applications and power-ranges. Therefore, the control of induction motor drives, especially of the most frequently used type – cage-rotor machines was developed greatly over the recent years. Control and estimation of IM drives is relatively more complex because of the complex dynamics of ac machines, relatively large parameter variations, need for variable-frequency converter power supplies and presence of harmonics. The controllers used in the motor drive systems can be broadly classified into main three categories such as: (a) fixed gain types, (b) adaptive types and (c) artificial intelligent types. The traditional controllers used in the industry are: (a) proportional-integral-derivative (PID) and (b) proportional-integral (PI) controllers. Also, now and then a type of controllers named pseudo-derivative-feedback (PDF) is used. However, the fixed-gain controllers are very sensitive to parameter variations, load disturbances, temperature changes, etc. This means that the controller parameters have to be constantly adopted. The problem of adaptation of parameters can be solved by several control techniques such as: (a) model reference adaptive control (MRAC), (b) sliding-mode control (SMC), (c) variable structure control (VSC), (d) self-tuning PI controllers, etc. The artificial intelligence types are represented by: (a) fuzzy logic controller (FLC), (b) artificial neural network controller, and (c) combined neuro-fuzzy controller.

1.3.2 Scalar Control of IM Drives

Currently, there are two areas for field oriented control of the induction motor. One of them is to improve the control schemes using optimal and state variable controls of induction motor with field oriented control [12 - 14]. The other method is focused on more accurate estimation of motor parameters, which is critical in field oriented control so that the rotor flux can be precisely calculated from these parameters. Two different ways exist to calculate the angle of the rotor flux: (a) the direct method (also called the flux feedback vector control), and (b) the indirect field oriented control (also called the flux feed-forward vector control) [15]. In the previous years more attention was given to the indirect method because stator flux sensor is not needed. Therefore, the reliability of the whole system is improved.

When using the direct method, the flux sensor is used to detect the amplitude and the position of rotor flux linkage. The type of sensor can be Hall effect transducer or search coils placed in a certain position in the stator. Comparing the amplitude of the reference flux and the actual flux the direct magnetising current (i_{ds}) is obtained. The quadrature axis (q-axis) current (i_{qs}), representing the torque component, is generated from the speed loop. The direct field oriented control method solemnly relies on the signals, generated from the air-gap fluxes. This method is working fine in the range of speed no more then 10% off (lower or higher) the rated speed, because of the flux measurement error. For many variable speed applications IM drives have to be employed in a range of speeds from zero to over rated values, so only indirect method can be used to calculate the rotor position and flux. In the speed region below rated speed, rotor flux can be calculated precisely from the stator current and rotor parameters. This approach is more relying on the knowledge of exact motor parameters than the direct method. To achieve

exemplary performance the rotor resistance, leakage inductance and the magnetising inductance must be precisely known. Unfortunately, induction motor parameters tend to vary because of the temperature, current and applied frequency changes. Consequently, using inaccurate parameters will lead to deterioration of control and dynamic performance of the drive would be lost. Many researches have concentrated on the motor parameters estimation [16 - 18] and there is huge progress made in this area of study.

A General speed control block diagram of IM drive is shown on Fig. 1.2. It consists of hierarchy of control loops added to it. Instead of voltage control, the converter may be current-controlled with direct or indirect voltage control in the inner loop. The induction machine dynamic model equations are non-linear and multivariable. Furthermore, there are coupling effects between the input and output variables. For example, both the torque and flux of an induction machine are functions of voltage and frequency. Machine parameters may vary with saturation and temperature adding further nonlinearity to the model of the drive.

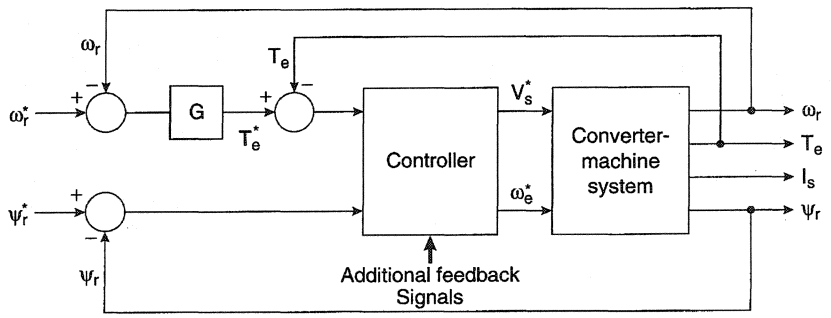


Figure 1.3. Block diagram of IM drive speed control.

In IM drive speed control scheme given above, all control and feedback signals can be equivalently considered as proportional to actual variables and dc.

The inner torque control may be optional, this loop is used for speed control. Usually, like in a dc machine, ac machine flux is held constant at the rated conditions to get the high torque per ampere and fast response. It is to be noted that any of the fluxes in rotor, stator or air gap can be considered for control, the control scheme in Fig. 1.2 considers rotor flux. Figure 1.2 presents the controller block, which can be of any type mentioned previously and will be discussed later in another chapter.

The most simple of the IM drive controls is the open loop constant volts/Hz control (CVH), the simplicity of which made it widely popular method used in the industry. To maintain the flux at a constant, typically rated level, the stator voltage should be adjusted in proportion to the supply frequency. For applications where speed has to be adjusted constantly, frequency control is inherent. Neglecting the stator resistance drop, voltage is made proportional to frequency so that the flux ψ remains constant.

$$\psi \approx \frac{V_S}{\omega_e^*} = \frac{1}{2\pi}\frac{V_S}{f} \tag{1.2}$$

A simple version of the CVH drive is shown in Figure 1.3. Optionally, a current limiter can be employed to reduce the output voltage of the inverter, when too high motor current is detected. The current measured in the dc link is more convenient as a feedback signal than the actual ac motor current.

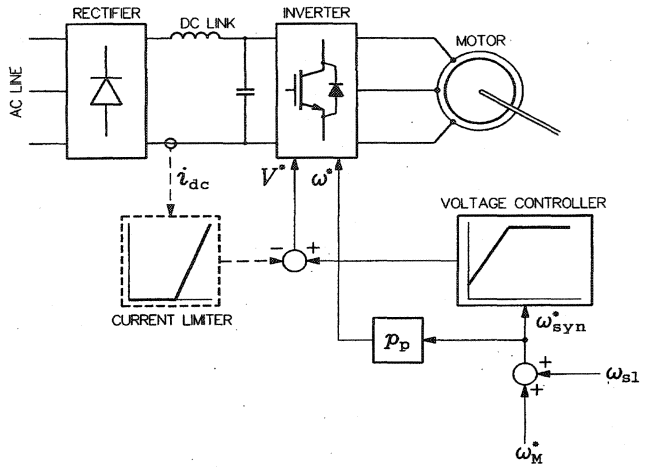


Figure 1.4. Basic CVH drive system.

This open loop volts/Hz control scheme can be improved by including slip regulation. In the common layout a PI controller is used to generate the slip command from the speed loop error. To generate the frequency command this slip is added to the feedback signal, it also generates the voltage command through the volts/Hz function generator block. The scheme can be thought over as an open loop torque control inside the speed control loop because slip is proportional to torque at constant rated flux and no feedback signal is used. The IM drive under CVH control works well at all speeds, and especially in the flux-weakening mode.

With the motor speed measured by the means of sensor or estimated, it can be controlled in the closed-loop scheme shown in Figure 1.4. The speed (angular velocity) is compared with the reference slip speed. The speed error signal is applied to a slip controller, usually of the PI type, which generates the reference slip speed. The slip speed must be limited for stability and over current prevention. Therefore, the slip controller's static characteristic exhibits saturation at a level lower than the critical slip speed. Adding the actual rotor speed to the generated

reference slip speed, the reference synchronous speed is obtained, which is used to generate the reference values of inverter voltage and frequency.

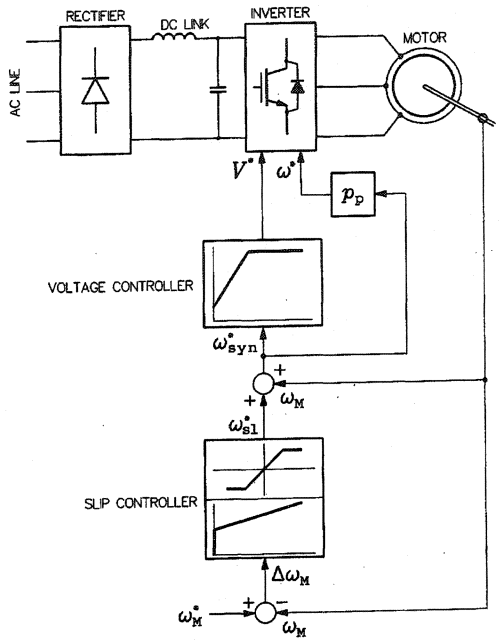


Figure 1.5. Scalar-controlled drive system with slip controller.

A typical general block diagram of a high-performance induction motor drive with speed control is shown in Figure 1.5. The speed of the motor is measured or estimated and compared with the reference signal. Usually, the speed is determined as a time derivative of the rotor angle, Θm. The speed error and reference signals are forwarded to the speed control system, which produces reference values of flux (stator, rotor, or air-gap) and torque of the motor. These, with other applicable signals from various parts of the drive, are employed in the flux and torque con-

trol system based on Field Oriented Control (FOC), Direct Torque Control (DTC), or Direct Self-Control (DSC) principle.

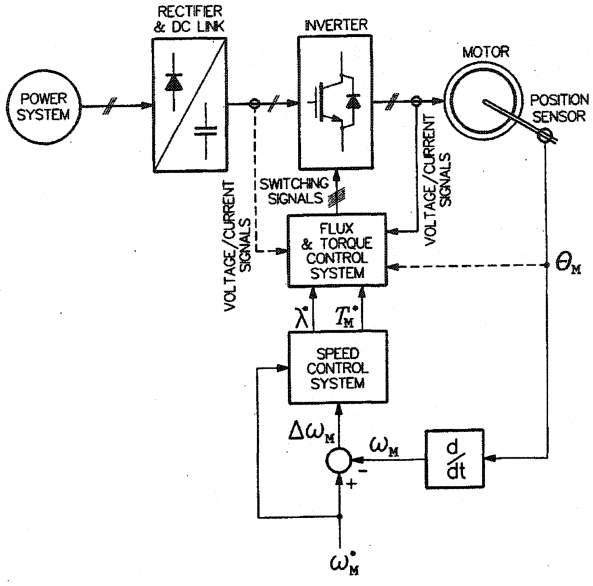


Figure 1.6. General block diagram of a high-performance adjustable speed drive.

However, it is often difficult to develop an accurate system mathematical model due to unknown load variation, unknown and unavoidable parameter variations due to saturation, temperature change and system disturbances. In order to overcome the above problems, the fuzzy-logic controller (FLC) is being used for motor control purpose.

1.3.3 Vector Control of IM Drives

As discussed above, because of inherent coupling effect scalar control methods have some drawbacks, such as sluggish response and system is easily prone to instability. Flux variation is always sluggish and control loop needs additional voltage to compensate it, reducing the torque sensitivity with slip increase and also increasing the response time for any type of inverters used.

Field-oriented control (also known as vector control) can improve the response time and resolve the discussed problems. In the vector control technique for IM drives, the magnitudes of the phase currents and the phase angle are controlled to provide high precision control of the motor. The vector control schemes that employ a current controlled voltage source inverter (VSI), provide the means of adjustable speed control for the IM drive that have fast response and follow command speed exactly and accurately.

The invention of control techniques developed in the 1970s by Hasse [6] and Blaschke [7] for ac drives brought a revival to the high-performance IM drives control. Due to a dc machine-like performance vector control is also called a decoupling, orthogonal, or transvector control. At the time of invention, though, implementation of vector control was strenuous due to technological limitations and complex nature. Because it demands corresponding feedback signal processing, especially in modern sensorless drives, and use of powerful DSP processing units is essential. Nowadays, with large scale of integrated circuit technologies, and advancing power electronics and microprocessor devises, the practical achievement of the vector control scheme for the IM drive is no longer a big issue. Undoubtedly, in

the future vector control will phase out scalar control and will be accepted as an industry standard.

Unlike dc machine control, in an ac machine both the magnitude and phase angle of the stator current need to be taken into account. This is completed by employing a time-varying vector that concurs to a sinusoidal flux wave moving in the airgap of the machine. From this cause the name of vector control is derived. The operating principle of vector control is based on exclusion of the coupling effect between the direct (d) and quadrature (q) axes. This can be achieved by adequate co-ordinate transformation, producing a separately excited dc motor-like control.

If the magneto motive force (MMF) wave of the stator current is referred to the vector corresponding to the flux wave, it becomes clear that only the perpendicular axis component of the stator current MMF wave contributes to the developed torque. The direct axis current is contributing to the flux magnitude. Naturally, it is suitable to define the stator current in a reference frame defined by the time-varying field, showing the close correspondence with dc machines. Such a analogy shows that the d-axis component of stator current in a IM drive is analogous to the field current in the dc machine and the quadrature axis component of stator current is analogous to the armature current of the dc motor. The distinctive characteristic of IM space vectors is that they rotate synchronously at the supply frequency.

Thus, when the reference quadrature axis current is controlled, it affects the actual quadrature axis current only, without affecting the flux. In the same manner, when the reference direct axis current is controlled, it controls the flux only and does not affect the other (quadrature axis) component of the current. This vector

orientation is a key feature under all operating conditions in a vector-controlled IM drive.

The vector control technique is classified into two big groups: (a) direct or feedback vector control and (b) indirect or feedforward vector control. The direct control relies on the direct sensing of the rotor (or stator) flux using flux sensors. The direct and quadrature axis flux components together with the command flux and torque and actual flux and torque, are employed to generate the main control parameters, the rotor flux as well as the direct and quadrature axis command currents. These dc currents, respectively proportional to the command torque and flux, are then converted to a stationary reference frame and used to generate phase current commands for the voltage source inverter. The correct alignment of direct axis current (i_{ds}) in the direction of rotor flux, and the quadrature axis current (i_{qs}) perpendicular to i_{ds} are crucial for control. The generation of a unit vector signal from feedback flux vectors gives the name to this method. Main features of direct control include stability, fast dc machine like transient response, and speed control in four quadrants without any additional control elements.

The second method of vector control is the indirect vector control. It is essentially the same as direct vector control, only the unit vector signals ($\cos\Theta_s$ and $\sin\Theta_s$) are generated in feedforward manner, by using sensors to find out the rotor position and stator currents. Using this information and the command speed signal the rotor reference frame, direct and quadrature axis command currents are calculated. These currents after the stage of transformation to a stationary reference frame are used to generate the phase current commands for the voltage source inverter. The speed signal from an incremental-position encoder is required in indirect vector control, because the slip signal locates the pole with respect to the rotor

direct axis in feedforward manner, which is moving at the rotor speed. An absolute pole position is not required for this technique. The indirect method of vector control is more sensitive to parameter variations than the direct method, and so the motor parameters must be known precisely.

The instantaneous current control of the voltage source inverter (VSI) is necessary for both direct and indirect vector control. In both of these schemes, a current controlled VSI is used to apply the produced command currents to the stator. This presumes the exact and accurate control of the VSI. The current controllers that produce minimal harmonics in the motor, as well as have low losses and fast response for high performance under dynamic conditions, are required. The performance of various current control schemes for the VSI fed induction motor drives and other drives have been investigated [19 - 21]. Each scheme has shown to have it's own disadvantages and advantages with regards to accuracy and dynamic response over changing speeds. The proposed work employs a simple sinusoidal band hysteresis current control algorithm for the VSI with closed-loop current control. Though its harmonics content is not optimum, and it would tend to saturate in part of the cycle, this scheme provides the fast response and good exactness in producing a stator current which tracks the command current within a hysteresis band, avoiding unnecessary complexity and easing up computational burden. In ac drives with the current source inverters (CSI), the required space vector of the stator current is generated by simultaneous adjustment of the dc-link current and inverter state selection.

As an alternative to rotor flux orientation systems, the stator and air-gap orientation can be used. The main drawback of such systems is complicated control algorithm, comparing to the rotor flux orientation. However, correct estimation of

the stator flux vector is easier than that of the rotor flux vector, while the air-gap vector can be measured directly.

1.3.4 Direct Torque and Flux Control of IM Drive

Direct torque control (DTC) of IM drive using voltage-fed PWM inverter drives was introduced by I. Takahashi and T. Noguchi [22]. This technique has very high performance comparable with vector-controlled techniques, but actually it is an advanced scalar control technique. The main feature of it is the direct control of the stator flux and torque of IM drive by inverter voltage space-vector selection (consecutive states) using a look-up table. The working principles of DTC can be described as follows: (a) stator flux of the machine is at all times a time integral of the stator electric motive force, so its magnitude is proportional to the stator voltage; (b) the developed torque is proportional to the sine of angle between the rotor and stator flux vectors; (c) the response of rotor flux alteration is slower than that of the stator flux.

As a result, this scheme directly controls both the developed torque and the magnitude of stator flux by suitable selection of space vectors of stator voltage, that is, consecutive inverter states.

The principle of the direct self-control (DSC) is different from DTC. The DSC scheme was proposed by Depenbrock in the year 1985 and originally it was developed for high-power IM drives using VSI. Low switching frequencies are required for this type of applications because of the technological specifications of GTO-based inverters. The main feature of this scheme is that inverter is forced to operate in a mode similar to a square-wave mode with sporadic zero states. When

the drive is required to run in a field weakening area with the speed higher than rated, zero states disappear, at all the other speeds the inverter operates in the square-wave mode.

The so called bang-bang controllers for torque and flux parameters are employed in the DTC drives, while integrators of stator EMFs are used in the DSC techniques, used in the feedback loop of the high-power IM drives. To improve the steady-state performance of the DTC and DSC systems, the space-vector PWM technique is used to determine a reference space-vector of stator voltage.

1.3.5 Adaptive and Intelligent Control of IM Drive

Instead of classical approach in speed-controlled drives employing typically controllers of the proportional-integral (PI) type, robustness and swiftness of the control can be enhanced greatly using a variable structure control (VSC) scheme, which is based on the idea of sliding mode of control. Leaving aside the exact details it can be mentioned that the main idea is to force the controlled variables to stay within the tolerance band by immediate reaction to a disturbance. This control scheme can be divided into two main types: (a) relay type, (b) relay-linear type. The adaptive control techniques can be generally classified as: (a) self-tuning control, (b) MRAC, (c) Sliding mode or variable structure control, (d) expert system control, (e) fuzzy control, (f) neural control, (g) combinations of above mentioned methods.

Contemporary VSI techniques and advancing power electronic technologies have led to the utilisation of microprocessors in the control of the IM drives. This allows the current advancements in computing power to be applied to IM drive

control, resulting in the implementation of complex control strategies, such as controllers based on the principles of machine intelligence (MI) have been employed in advanced high-performance drives (HPD). The primary idea of MI systems is the emulation of the way humans think they are very effective in applications characterized by complex or poorly defined mathematical models and because of their robustness and adaptivity they are exceptionally effective in high-performance IM drives. As already mentioned earlier, three types of controllers stand out: (a) neural controllers, (b) fuzzy controllers, (c) hybrid neuro-fuzzy controllers. The latter type combines operating principles of neural and fuzzy controllers. The latter are also likely to dominate control systems in the future. The proposed work is devoted to application of FLC in IM drive, therefore the literature review will focus on the development of this type of control technique.

Many researchers have reported their work on development of high performance induction motor drives, with the majority of the control techniques involving the IM fed by a voltage source or current source inverter. To illustrate it, the reader is directed to a typical indirect vector control scheme of high-performance drive shown previously in Figure 1.5.

Working with the voltage-fed inverter control for induction motors, R Ueda, T. Sonada, K. Koga, and M. Ichikawa [23] have shown, that in the typical drive performance with open loop volts/Hz control, the inherent machine coupling effect slows down the torque response. There is some amount of under-damping in the torque and flux responses, and drift in the flux signal for varying torque (i.e. stator current) is also noticeably big. Simulation studies indicated that at some regions of operation, the control system tends to be unstable.

A. B. Plunkett and D. L. Plette [24 - 25] have developed a control scheme where instead of controlling inverter voltage by the flux loop, the stator current was controlled. They have built a voltage-fed inverter drive with outer loop torque and flux control and hysteresis-band current control in the inner loop. Instead of using the constant rated flux they have programmed the flux to be a function of torque, which gave a huge efficiency improvement.

The era of vector control, as mentioned previously, started with the invention of direct or feedback method by Blaschke [7], and the invention of indirect or feedforward method by Hasse [6]. Other researchers concentrated on studying of ac drive operation under vector control [26 - 28].

Most of these works reported are investigating IM drive performance in the normal region of operation and while in flux-weakening mode are based on conventional PI and PID based speed controllers [29-35]. These controllers offer the advantages of simplicity of implementation in real time. However, all of them are very sensitive to parameter variations due to sudden changes in load, changes of command speed, saturation, temperature variations and other system uncertainties. Hence, it is rather difficult to tune the controller parameters precisely for an optimal implementation. As a result, these types of controllers are not suitable for high performance applications, and researchers have been prompted to develop adaptive control schemes for IM drive systems in which the controller parameters can be adapted in real time in response to system parameter variations and load changes.

T. M. Rowan, R. J. Kerkman, and D. Legatte [36] have reported on continuous on-line tuning of the IM drive and as a result have developed a better method based on the model referencing adaptive control (MRAC). The main feature of it is that the output of a reference model is compared with the output of an

adjustable or adaptive model until the errors between the two models are eliminated, irrespective of parameter variation and load disturbances. Such system is defined as a robust system. The model works successfully if machine parameters are considered as constant. However it does not work adequately if parameters tend to vary, gradually decreasing accuracy of control.

Brickwedde [37] has reported a research on microprocessor-based adaptive speed and position control for electrical drives. The shortcoming of his model for the induction motor is the computational burden that the algorithm imposes on the microcomputer. This limits the maximum operating speed of the drive.

In another works, Pillay [38-41] have proposed a digital signal processor (DSP) based hysteresis current controller scheme for a permanent magnet motor drive using a look-up table to generate the command current, which is not appropriate for a wide range of speed change. Although, the controller have proved to be effective, comprehensive tests at different dynamic operating conditions have not been done.

In the last decades sensorless control of the induction motors received much attention from researchers. The main focus of work was on vector control of IM drive without any speed sensors. The other methods require either a shaft-mounted speed encoder (often optical), or a speed sensor for the control circuit. In many cases, any additional speed sensor is undesirable because it increases cost and decreases reliability of the drive. By means of special relatively complex calculation techniques it is possible to produce the speed signal from machine voltages and current using a DSP board. The drawback of such systems is the need of exact motor parameters for satisfactory operating.

H. Kubota, K. Matsuse, and T. Nakano [42] developed a speed adaptive flux observer (Luenberger Observer) – an improved method of speed evaluation that operates using the main principle of speed adaptive flux observer. Basically, it uses a full or partial mathematical model and a feedback loop with already known machine parameters and variables. Lyapunov theorem was used to derive the speed adaptation algorithm. Despite the fact that observer improves accuracy of speed estimation, the parameters variation negatively effect the developed technique, producing large errors as the speed decreases.

Another work in the area of sensorless vector control was done by J. Holtz [43], he proposed to inject an auxiliary frequency carrier signal from the stator side of a custom-made rotor for the position and speed estimation. The previous techniques could not estimate the motor speed from the moment stator frequency became zero, the pure dc condition when machine torque and speed is zero. The drawback of this scheme is that the machine rotor slots are to be specially designed to get variation of magnetic saliency. The main type of problem that arises is that the accuracy of estimation is affected by the machine model parameter variation, the inertia variation error and the load torque variation error.

Further progress in control of torque-controlled IM drive with direct vector control was made by B. K. Bose and N. R. Patel [44]. The proposed control method uses sensorless stator flux orientation with neuro-fuzzy based controller. Fuzzy logic is used for motor parameter estimation.

As mentioned earlier, I. Takahashi and T. Noguchi [22], have introduced an advanced scalar control technique, known as direct self-control (DSC) or direct torque and flux control (DTC). This technique is found to be comparable with vec-

tor-control technique. It uses a look-up table to derive an inverter voltage space vector, therefore needs an exact motor drive parameters to be known beforehand.

G. Buja [45] and P. Vas [46,47] did another work on DTC for induction motor drives, the technique developed is fairly precise in speed control, but has the same drawbacks as mentioned before for this type of control.

K. Hong and K. Nam [48] developed a disturbance compensation scheme considering the speed measurement delay, where the estimation of the load torque and its compensation is made in a feedforward manner. This scheme can be used in high-inertia applications because of relatively big speed measurement delays, the inaccuracy of the model and the filter delay.

The comprehensive theoretical base was developed for sliding mode control (SMC) by U. Itkis [49]. He has shown that theoretically, a second-order system requires only an error signal and it's derivative as control inputs.

Researchers A. M. Khambadkone and J. Holtz [50] did significant research in the area of self-commissioning drives. Such works involve the precise measurement of machine parameters for the purpose of feedback signal estimation and control system tuning. The advantage of this type of controller is that it needs the exact machine parameters only at the start, not during the operating condition. This was done by observing the feedback loop response in real time.

The controllers of adaptive type show better performance and are not that sensitive to motor parameter variations and sudden load changes, compared to PI and PID type of controllers. The main drawback appears to be the high computational burden. For implementation of adaptive schemes a high power computer is needed. Due to the high number of system state variables these systems suffer from chattering, overshooting and steady state errors because of finite switching.

After L. Zadeh invented the principles of fuzzy logic control in 1965 [51,52], a number of researchers [53-55] have tried to apply the principles of fuzzy logic and various other types of intelligent controllers, such as artificial neural network (ANN), etc, for the control of the induction motor drives, at normal working conditions. They are often called artificial intelligence (AI) controllers, because they involve software programming where the computer mimics human thinking in the control of the motor [56,57].

In early stages of research the main problem for implementing the ANN type of control was the high computational burden. This is why the earlier reported works are mainly theoretical, and are based on simulations or low-speed experimental results [58 - 59]. As discussed before, these types of controllers can be made self-adaptive and do not need exact information on system parameters for precise operation, especially for systems where one parameter depends on the other or on the operating conditions. Intelligent controllers can be considered self-optimised and adaptive systems, also they do not need any advance information about any non-linearity in the system.

Significant amount of work has been reported on the use of artificial neural networks (ANN) for induction motor drives [58-65] and dc motor drives [66 - 68]. That work was concentrated on the use of the ANN in motor drive systems with the aim of achieving characteristics of adaptive controllers by exploiting the inherent non-linear input and output properties of the ANN.

One of the first works on applying a special control technique incorporating the use of DSP was made by Miki [69]. He reported the use of a current control method that enabled a multitude of patterns of space-voltage vectors generated by

the inverter, and developed a DSP-based software field-oriented induction motor system.

A group of researchers under the B. K. Bose [70] have proposed fuzzy control of vector-controlled induction motor, where fuzzy control is used to get robustness against parameter variation and load torque disturbance effects. Both rough and fine control look-up rule tables are used to improve the transient response and system settling time. The drawback of the proposed scheme is the large number of fuzzy rules, requiring huge computational burden.

S. A. Mir and D. S. Zinger [71] have reported a work on application of fuzzy logic controller for inverter fed IM. Because an IM operated with a direct self controller (DSC) shows a sluggish response, he used fuzzy logic to choose the switching states. Flux position, error in flux magnitude and error in torque are used as fuzzy state variables. Fuzzy rules are determined by observing the vector diagram of flux and currents. To improve the system performance at low speeds a fuzzy resistance estimator is proposed to eliminate the error due to the change in stator resistance. Any change in stator resistance of the induction machines causes an error in stator current. This error is utilised by the fuzzy resistance estimator to correct the stator resistance used by the controller to match the machine resistance.

The more general study on fuzzy logic application for intelligent control of a variable speed drive was made by two groups of researchers Y. Tang and L. Xu [59, 73] and J. Fonseca, J.L. Afonso, J. S. Martins, and C. Couto [72]. They applied a fuzzy logic for the intelligent control of a slip power recovery system. They also developed a direct fuzzy logic controller and an adaptive fuzzy controller, based on model reference adaptive control for the doubly excited machine and converter system. It was proven that compared with the field orientation control, the intelligent

control of the complex slip recovery system reduces costs and enhances robustness and desired performance.

A. Ibaliden and P. Goureau have experimented with the static and dynamic fuzzy controller for the speed control of an induction motor [74] by comparing it to the conventional PI controller. To improve the static fuzzy controller performances during parameter variations, a dynamic controller consisting of two fuzzy controllers is proposed.

Mao-Fu Lai and M. Nakano [75] also reported some initial work on the use of a phase-locked loop induction motor speed drive incorporating a fuzzy logic controller. Their system combines the relatively good speed regulation of the phase-locked loop techniques and the advantages of fuzzy logic to obtain a robust, fast and precise control of induction motor speed. Their experimental results showed that the combination of a fuzzy logic controller and the phase-locked loop in induction motor drive could achieve precise speed control with fast response.

Recently, researchers like L. A. Cabrera, M. Elbuluk and I. Husain [62] have reported the work on use of artificial neural networks for tuning the stator resistance of IM drive systems for DTC control. The parallel recursive prediction error and back propagation training algorithms are being used to tune the stator resistance, with on-line training, making DTC more robust and accurate. Experiment was conducted for three different neural-network configurations showing the efficiency of the tuning process.

Similar work was done by researchers in the area of ANN control for the other types and configurations of drives [58-67]. Recently, Rahman and Hoque [68] have proposed an on-line adaptive ANN based controller for permanent magnet motor drive. The back propagation-training algorithm has been used in two artifi-

cial neural networks: the output of one network gave the command signal, the output of the other network gave the estimated signal. Comparing the error between the two outputs, the weights and biases are updated. The combined off-line and on-line types of training have been used. The complete scheme incorporating analogous vector control circuit and hysteresis current controller circuit makes the system complex and also not as suitable for robust operation as compared to the fully digital drive system. Also, the proposed controller was not able to drive the motor above the base speed.

Approximately at the same time, B. Heber, L. Xu, and Y. Tang [73] have proposed a field orientation control of induction machines using fuzzy logic enhanced speed control. Trying to overcome the problem of field orientation detuning caused by parameter variations for indirect field orientation control, he presented fuzzy logic design approach that can meet the speed tracking requirements when detuning occurs. The proposed circuit uses rated fixed value of the quadrature axis current.

Cerruto has proposed a fuzzy adaptive control of induction motor drives [60]. A speed model reference adaptive control (MRAC) system for indirect field-oriented induction motor drives based on using fuzzy laws for the adaptive process and a neuro-fuzzy procedure to optimise the fuzzy rules. Variation of the rotor time constant is also accounted for by performing a fuzzy fusion of three simple compensation strategies. The drawback of the scheme is the high demand of computational burden and slightly sluggish response.

Further work was done by B. K. Bose and N. T. Patel [44]. They proposed a high-performance stator flux oriented speed sensorless direct vector-controlled induction motor drive with neuro-fuzzy based performance enhancement. The drive

incorporated previously developed features like stator resistance estimation by fuzzy logic, on-line search of flux for steady state efficiency improvement, zero-speed standstill start-up of the drive using machine current model equations, etc. The drive was implemented using a distributed DSP system. However, this method has some disadvantages as it thoroughly depends on the look-up table and, for implementation, it needs a powerful computational system to handle simultaneously both on-line ANN and fuzzy logic processes.

Although much research work has been reported on the use of ANN and fuzzy logic for induction motor drives [53-76], one of the best implementation of the neural networks was reported by J. Pinto, B. K. Bose and L. Silva [64]. They used neural network controller for flux vector synthesis of a stator flux oriented vector-controlled IM motor drive. The reported scheme was implemented using the space vector PWM and stator flux vector estimation. A feedforward ANN-based controller generates symmetrical PWM pulses in both undermodulation and overmodulation regions covering the range from dc (zero frequency) up to square-wave mode at 60 Hz. Also, a programmable cascaded low-pass filter that permits dc offset-free stator flux vector synthesis at very low frequency using the voltage model, implemented by hybrid neural network and a feedforward neural network. This type of layout permits faster implementation, and gives better transient performance when compared with a standard DSP-based controller. The drawback of the method is that it requires initial training for the controller, and to be implemented in industry needs a specially designed hardware, microchips as for example.

The successful real-time application of the fuzzy logic controller for the high performance IM drive system was made by M. A. Rahman, T. Radwan and N. Uddin [76]. They used fuzzy-logic controller based on the indirect vector control.

The fuzzy-logic speed controller is employed in the outer loop, implemented using a digital signal processor board. It was found to be more robust, and hence a suitable replacement of the conventional PI controller for the high-performance industrial applications. The drawback of this particular scheme is simplification based on assumption that the direct axis control current is zero. And, as an outcome, the motor draws the higher current during the rated conditions, as well as a large number of fuzzy rules increase the computational burden. Because of that, an experimental implementation was speed limited. Furthermore, the condition $i_d = 0$ is not a realistic constraint for an indirect motor control at rated load conditions. Hence, there exists a need to find means to completely utilise the proposed scheme as well as increase the precision of it by providing systematic research on the fuzzy logic control of the induction motor using direct axis current i_d not equal to zero approach. For the best results the drive controller is thought to incorporate the indirect vector control scheme. In a vector control scheme it is also important to choose the right current controller, which plays a critical role for drive performance.

1.4 Problem Identification and Thesis Objectives

Each of the systems presented above have some drawbacks. The solution requires careful consideration if the IM drive is to be practically implemented under high performance drive (HPD) standards. Thus, it is necessary to further develop control algorithms and approaches to produce this high standard of realisation in a practical manner. High performance drive (HPD) systems must provide fast and

accurate speed and torque responses, quick recoveries from sudden disturbances, and show insensitivity to parameter variations.

The main advantages of high performance induction motor drive include robustness, reliability, high power to weight ratio, high torque to inertia ratio, high load ability compared to the conventional motors. High performance drives (HPD) are used in all kinds of industrial applications as robotics, air conditioners, rolling mills, traction machines, spindle drives, etc. In spite of the multiple advantages of induction motor drive, it also has some well-known drawbacks. Saturation and armature reaction effects can affect the work of the induction motor as well, as other non-linearities. This makes the control of the IM for HPD applications an engineering challenge.

Consequently, the work presented here would develop a complete control of induction motor drive incorporating intelligent controller. For this purpose a fuzzy logic controller is proposed as a basic element. Therefore, this work presents the design of a specific FLC for the induction motor drive. The vector control scheme incorporating a speed controller and a current controller is used, because it decouples the torque and flux, thus providing faster transient responses and making the control task easier. The real-time implementation of the integral vector control scheme of induction motor drive incorporating fuzzy logic controller is made using a DSP controller board installed in PC, a one horse power induction motor unit, a voltage source inverter and other hardware and software for experimental purposes. For this type of IM drive controllers, a sampling frequency of 20 kHz was chosen due to the special simplified FLC layout, which overcomes the high computational burden attached to fuzzy logic, and is sufficient for experimental implementation of the scheme at all modes of operation. To be able to run the motor above as well as

below the rated speed, different modes of operation are considered in this work, especially the flux-weakening mode. Using fuzzy-logic based technique the operating speed range of the motor can be increased notably.

As the above literature survey clearly have shown that all of the above described controllers such as: fixed-gain proportional integral (PI) and proportional integral derivative (PID) controllers, model reference adaptive controllers (MRAC), variable structure controllers (VSC), sliding mode controllers (SMC) and self-tuning regulators (STR) require precise knowledge of system model parameters. Especially proportional-integral and proportional-integral-derivative controllers require the exact system information.

On the other hand, fuzzy logic controller (FLC) does not need any exact parameter information for the system mathematical model, it is self-adaptive to uncertainties and can handle drive system having non-linearity. However, FLC-based drive systems use relatively complicated mathematical algorithms that can impose high computational burdens. Because of that the best performance can only be guaranteed if the latest, most powerful personal computer systems are used. Hence, this work is also concerned about producing a control scheme of the maximum compatibility and minimal complexity using fuzzy logic. Furthermore, many researchers that work in this area use a simplified linear model of the IM drive by forcing the direct-axis current to zero ($i_d = 0$). This scheme produces a motor control that requires increased stator current to produce a given torque and is not quite accurate as an outcome. Therefore, this work includes the development and a practical implementation of the fuzzy logic controller scheme, which produces motor torque with the minimum possible stator currents using $i_d \neq 0$ approach. This characteristic property increases the effectiveness of the drive significantly. To verify

the theoretical results and computer simulations, practical laboratory implementation has been made.

1.5 Organisation of the Thesis

The introduction and literature survey of IM drive control techniques, as well as the problem identification and objectives of this thesis have been included in this chapter. The organisation of the remaining part of the thesis is as follows.

Chapter two covers the theoretical development of the induction motor (IM) mathematical model as well as the analysis and modelling of the PWM voltage source fed IM drive. The systematic mathematical formulations are presented.

The development of the fuzzy logic based speed controller for the IM drive is discussed in chapter three. It also covers a simplified fuzzy logic based speed controller that reduces complexity and computational burden significantly. Some specific notions developed include linguistic variables, simplified membership functions, type of fuzzification, rule evaluation and defuzzification. The complete Simulink model used for the drive simulation is given in attachment (Appendix B).

Computer simulation results for the developed drive scheme are presented in chapter 4. In order to predict the performances of the proposed controller the extensive simulations are carried out and simulation results presented.

The results of the real-time experimental implementation of the simplified FLC-based induction motor drive which have been implemented using a digital signal processor (DSP) based vector control of a laboratory 1 hp IM are presented in chapter five.

Chapter 6 presents the summary of the contribution of this work and the conclusions. Some suggestions for the future study and controller development are made.

All pertinent references and appendices are listed at the end of the thesis.

Chapter 2

Modeling and Analysis of the PWM VSI-Fed IM Drive

2.1 General Introduction

This chapter presents the development of the a complete current-controlled voltage source inverter (VSI) -fed induction motor (IM) drive using the direct and quadrature (d-q) axis mathematical model. Both speed and current controllers play an important role in the vector control scheme to make the drive follow the command speed accurately in changing conditions and quadrants of operation. To apply the precise stator currents to the stator of the induction motor through the mathematical model of a VSI a fixed-band hysteresis current controller was used.

To operate the vector control scheme precisely, a fuzzy logic based speed controller has been chosen. The results for FLC -based controller have been com-

pared with conventional PID –based controller. This will be developed and shown in chapter 3.

2.2 Mathematical Modeling of the IM drive

The method of mathematical modeling for the simulation purposes of induction machine is the subject of this chapter. The IM has three-phase windings, delta or wye form, distributed proportionally and sinusoidally and embedded in slots of the stator. A wound rotor machine winding is similar to that of the stator, but in a cage rotor machine, the rotor incorporates a cage structure with shortened by ring edges. With a fair amount of simplification, the induction motor can be looked upon as a transformer with three phases, where shorted secondary coil is rotating. It can also be said, that the coupling coefficients change continuously when rotor is moving. The mathematical model can be derived using differential equations with varying in time mutual inductances, but this model appears to be very complicated to use, because the three-phase rotor windings move with respect to the three-phase stator windings.

For analytical purposes, the three-phase induction machine can be represented as a two-phase machine, with two pairs of axes: (a) direct and quadrature stator axes, and (b) – direct and quadrature rotor axes. The three-phase stationary reference frame (as – bs – cs) variables can be transformed into two-phase stationary reference frame (d^s – q^s) variables using Parks [78] transformation, and then transform them to synchronously rotating reference frame (d^e-q^e), and back.

The stator circuit equations in a synchronously rotating (d^e-q^e) frame, using transformation for both stator and rotor variables proposed by Kron [79]:

$$v_{qs}^{\ s} = R_s i_{qs}^{\ s} + \frac{d}{dt}\Psi_{qs}^{\ s} \tag{2.0}$$

$$v_{ds}^{\ s} = R_s i_{ds}^{\ s} + \frac{d}{dt}\Psi_{ds}^{\ s} \tag{2.1}$$

In this equations quadrature and direct –axis stator flux linkages $\Psi_{qs}^{\ s}$ and $\Psi_{ds}^{\ s}$ are included. The superscript "s" refers to stationary reference frame, sub-scripts "ds" and qs" refer to the direct and quadrature stator quantities referred to stationary reference frame, respectively. When these equations are converted to the synchronously rotating reference frame using dynamic model (or Kron Equation, otherwise), we get:

$$v_{qs} = R_s i_{qs} + \frac{d}{dt}\Psi_{qs} + \varpi_e \Psi_{ds} \tag{2.2}$$

$$v_{ds} = R_s i_{qs} + \frac{d}{dt}\Psi_{ds} - \varpi_e \Psi_{qs} \tag{2.3}$$

The variables included in equations are in rotating form. For convenience, the superscript *e* has been dropped from now on from the synchronously rotating frame parameters. Speed EMF due to rotation of the axis is represented with the last terms of equations (2.2) and (2.3). That assumes, if the synchronous speed comes to zero, equations return to stationary form. Equations for rotor in stationary form, when rotor is not moving are [80,81]:

$$v_{qr} = R_r i_{qr} + \frac{d}{dt}\Psi_{qr} + \varpi_e \Psi_{dr} \tag{2.4}$$

$$v_{dr} = R_r i_{qr} + \frac{d}{dt}\Psi_{dr} - \varpi_e \Psi_{qr} \tag{2.5}$$

These equations have variables that are referred to the stator. Rotor is moving with rotor speed, the direct and quadrature axis are moving at the speed, which is the difference between synchronous and rotor speed. This speed is relative to the synchronously rotating frame. Incorporating this, equations are modified:

$$v_{qr} = R_r i_{qr} + \frac{d}{dt}\Psi_{qr} + (\omega_e - \omega_r)\Psi_{dr} \tag{2.6}$$

$$v_{dr} = R_r i_{qr} + \frac{d}{dt}\Psi_{dr} + (\omega_e - \omega_r)\Psi_{qr} \tag{2.7}$$

The equivalent circuit of the dynamic model (de – qe) that satisfies equations (2.2), (2.3) and (2.6), (2.7) is presented in Fig. 2.1.

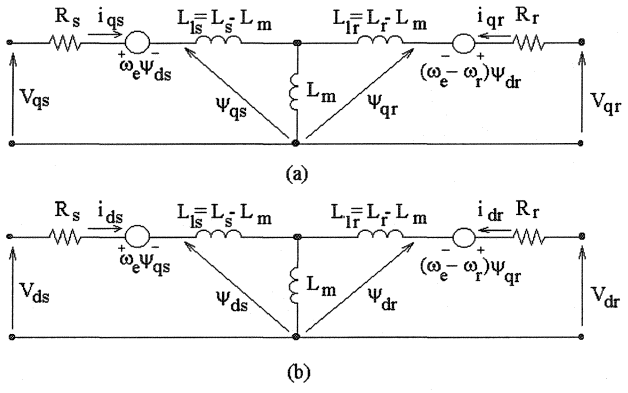


Fig. 2.1. The dynamic equivalent circuit (d-q) of an induction machine, (a) quadrature axis circuit, (b) direct axis circuit.

A special advantage of this model is that all the sinusoidal variables in stationary frame appear as dc quantities in synchronous frame. The flux linkages in terms of the currents can be written using an equivalent circuit shown on fig. 2.1 as in [77]:

$$\Psi_{qs} = L_{ls} i_{qs} + L_m (i_{qs} + i_{qr}) \tag{2.8}$$

$$\Psi_{qr} = L_{lr} i_{qr} + L_m (i_{qs} + i_{qr}) \tag{2.9}$$

$$\Psi_{qm} = L_m (i_{qs} + i_{qr}) \tag{2.10}$$

$$\Psi_{ds} = L_{ls} i_{ds} + L_m (i_{ds} + i_{qr}) \tag{2.11}$$

$$\Psi_{qr} = L_{lr} i_{dr} + L_m (i_{ds} + i_{dr}) \tag{2.12}$$

$$\Psi_{dm} = L_m (i_{qs} + i_{dr}) \tag{2.13}$$

Bringing together equations (2.2), (2.3), (2.6), and (2.7) given previously, the mathematical transient model for a three-phase wye-connected induction machine equations are given in matrix form as [11]:

$$\begin{bmatrix} v_{qs} \\ v_{ds} \\ v_{qr} \\ v_{dr} \end{bmatrix} = \begin{bmatrix} R_s + L_s p & \omega_s L_s & L_m p & \omega_s L_m \\ -\omega_s L_s & R_s + L_s p & -\omega_s L_m & L_m p \\ L_m p & L_m(\omega_s - \omega_r) & R_r + L_r p & L_r(\omega_s - \omega_r) \\ -L_m(\omega_s - \omega_r) & L_m p & L_r(\omega_s - \omega_r) & R_r + L_r p \end{bmatrix} \begin{bmatrix} i^e_{qs} \\ i^e_{ds} \\ i^e_{qr} \\ i^e_{dr} \end{bmatrix} \tag{2.14}$$

where, P is number of poles of the IM; p is differential operator (d/dt); v_d, v_q are d and q -axis stator or rotor voltages; i_d, i_q are d and q -axis stator or rotor currents; J_m is moment of inertia of the motor and load; ω_r is rotor speed in angular fre-

quency; ω_e is speed of the rotating magnetic field; B_m is friction coefficient of the motor; L_s, L_r are stator and rotor self-inductance; R_s, R_r are stator and rotor resistance per-phase; L_m is mutual or magnetizing inductance; T_e is electromagnetic developed torque; T_L is load torque; Θ_r is rotor angular position.

For a cage-rotor singly fed machine, $v_{qr} = v_{dr} = 0$. The dynamic machine model in stationary frame can be derived simply by substituting $\omega_e = 0$ in equations (2.2) - (2.7). Fig. 1.1 shows the correspondent equivalent IM circuits. In the stationary frame, the variables appear as sine waves in steady state with sinusoidal inputs.

The transformation angle, $\theta(t)$ between the stationary stator windings and the quadrature axis of the reference frame is given as (in elect. rad.):

$$\theta = \int_0^t \omega(t)dt + \theta\ (0) \tag{2.15}$$

Similarly, rotor position angle θ_r between the axis of the rotor and stator, for a rotor rotating with speed ωr is defined as:

$$\theta_r = \int_0^t \omega_r(t)dt + \theta_r(0) \tag{2.16}$$

The resulting angles are calculated as sum of integral and the initial values of these angles at the beginning of considered time. The stator voltage equations of the three phases of the IM can be defined as:

$$v_a = r_a i_a + \frac{d\psi_a}{dt} \tag{2.17}$$

$$v_b = r_b i_b + \frac{d\psi_b}{dt} \tag{2.18}$$

$$v_c = r_c i_c + \frac{d\psi_c}{dt} \tag{2.19}$$

which can be also given in matrix form , as:

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} r_a & 0 & 0 \\ 0 & r_b & 0 \\ 0 & 0 & r_c \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + p \begin{bmatrix} \psi_a \\ \psi_b \\ \psi_c \end{bmatrix} \tag{2.20}$$

where p is the time differential operator, $\frac{d}{dt}$. Considering a symmetrical three-phase induction machine, as before, phase voltages can be resolved into d-q-0 voltages using the below-given equation in the matrix form. In order to avoid the complexity of calculations, all of the equations can be transformed to the synchronously rotating rotor reference frame where the machine equations are no longer dependent on the rotor position. This is concluded using equations of Park's transformation [78]. First, the machine equations are transformed from the stationary a-b-c frame to the two-phase stationary d-q frame, then they are transformed from the stationary d-q frame to the synchronously rotating reference frame. This process can be reversed. Using x to represent the machine phase variables, voltage or current, the Park's transform gives the following equations:

$$\begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} = \begin{bmatrix} \mathrm{Cos}\,\theta_r & \mathrm{Sin}\,\theta_r & 1 \\ \mathrm{Cos}\left(\theta_r - \frac{2\pi}{3}\right) & \mathrm{Sin}\left(\theta_r - \frac{2\pi}{3}\right) & 1 \\ \mathrm{Cos}\left(\theta_r + \frac{2\pi}{3}\right) & \mathrm{Sin}\left(\theta_r + \frac{2\pi}{3}\right) & 1 \end{bmatrix} \begin{bmatrix} x_q \\ x_d \\ x_o \end{bmatrix} \tag{2.21}$$

The corresponding inverse Park relation to (2.21) is as following:

$$\begin{bmatrix} x_q \\ x_d \\ x_o \end{bmatrix} = \frac{2}{3} \begin{bmatrix} Cos\,\theta_r & Cos\left(\theta_r - \frac{2\pi}{3}\right) & Cos\left(\theta_r + \frac{2\pi}{3}\right) \\ Sin\,\theta_r & Sin\left(\theta_r - \frac{2\pi}{3}\right) & Sin\left(\theta_r + \frac{2\pi}{3}\right) \\ 0.5 & 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} \tag{2.22}$$

Added zero sequence component is given as x_o. An angle θ_r is the initial rotor position at time t = 0, corresponding to $\theta_r(0)$ which happens to be also the angle difference between the a-phase and q-axis. For balanced 3-phase, x_o is equal to zero, and it is also convenient to set $\theta_r(0) = 0$ so that the a-phase corresponds with the q-axis. Under these conditions, ignoring the zero sequence component, the transformation relations (2.21) and (2.22) can be further simplified as:

$$\begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{-1}{2} & \frac{-\sqrt{3}}{2} \\ \frac{-1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} x_q \\ x_d \end{bmatrix} \tag{2.23}$$

and inversely:

$$\begin{bmatrix} x_q \\ x_d \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & \frac{-1}{3} & \frac{-1}{3} \\ 0 & \frac{-1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} \tag{2.24}$$

In order to convert (or resolve) these variables to the rotating d-q frame we use following relations. To have a graphic guide Fig 2.2 can be used as a reference. Therefore, the quantities (of voltages or currents) in the stationary d-q frame can be converted to the synchronously rotating frame as follows:

$$\begin{bmatrix} x_q^r \\ x_d^r \end{bmatrix} = \begin{bmatrix} \cos\theta_r & -\sin\theta_r \\ \sin\theta_r & \cos\theta_r \end{bmatrix} \begin{bmatrix} x_q \\ x_d \end{bmatrix} \tag{2.25}$$

Again, converting (or resolving) the rotating frame parameters into the stationary frame, the inverse relation is:

$$\begin{bmatrix} x_q \\ x_d \end{bmatrix} = \begin{bmatrix} \cos\theta_r & \sin\theta_r \\ -\sin\theta_r & \cos\theta_r \end{bmatrix} \begin{bmatrix} x_q^r \\ x_d^r \end{bmatrix} \tag{2.26}$$

Otherwise, the two-axis stator voltages and currents are related to the three-phase representations by the following equation:

$$\begin{bmatrix} x_{qs}^{e} \\ x_{ds}^{e} \end{bmatrix} = \begin{bmatrix} -\sin w_e t & \cos w_e t \\ \cos w_e t & \sin w_e t \end{bmatrix} \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} x_{as} \\ x_{bs} \\ x_{cs} \end{bmatrix} \tag{2.27}$$

As presented in equation (2.14) speed is the variable and cannot be treated as a constant, it can be related to the mechanical and electrical torques as following:

$$T_e = T_L + J\frac{d\omega_m}{dt} = T_L + \frac{2}{P}J\frac{d\omega_r}{dt} \tag{2.28}$$

where variables employed are load torque, rotor inertia and mechanical speed.

By multiplying (2.3) by "j" and adding with the previous equation (2.2) a compact representation of machine equations in complex form can be achieved:

$$v_{qs} - jv_{ds} = R_s(i_{qs} - ji_{ds}) + \frac{d}{dt}(\Psi_{qs} - j\Psi_{ds}) + j\omega_e(\Psi_{qs} - j\Psi_{ds}) \tag{2.29}$$

or

$$v_{dqs} = R_s i_{qds} + \frac{d}{dt}\Psi_{qds} + j\omega_e\Psi_{qds} \tag{2.30}$$

where v_{qds}, i_{qds}, etc. are the complex vectors rotating with synchronous speed. In a like manner, (2.6) - (2.7) can be joined to give:

$$v_{qdr} = R_r i_{qdr} + \frac{d}{dt}\Psi_{qdr} + j(\omega_e - \omega_r)\Psi_{qdr} \tag{2.31}$$

From the above given equations (2.29) and (2.30), the steady – state equations can be derived by setting the time derivative components to zero:

$$V_s = R_s I_s + j\omega_e \Psi_s \tag{2.32}$$

$$0 = \frac{R_r}{S} I_r + j\omega_e \Psi_r \tag{2.33}$$

where the rms phasors replaced the complex vectors. The equations above can also be presented in the form of a complex synchronous frame dqs equivalent circuit, if the parameter R_m is neglected.

Using a known equation for torque in terms of maximum values of rotor current and magnetizing flux, the torque can be generally expressed in the vector form as following :

$$T_e = \frac{3}{2}\left(\frac{P}{2}\right)\overline{\Psi_m} \times \overline{I_r} \tag{2.34}$$

Resolving the variables into d - q components, the torque expression can be written as:

$$T_e = \frac{3}{2}\left(\frac{P}{2}\right)(\Psi_{dm} i_{qr} - \Psi_{qm} i_{dr}) \tag{2.35}$$

Some other equations for torque can be derived from the above using the d – q currents and fluxes relations, and using substitution:

$$T_e = \frac{3}{2}\left(\frac{P}{2}\right)(\Psi_{dm} i_{qs} - \Psi_{qm} i_{ds}) \tag{2.36}$$

$$T_e = \frac{3}{2}\left(\frac{P}{2}\right)(\Psi_{dm} i_{qs} - \Psi_{qm} i_{ds}) \tag{2.37}$$

$$T_e = \frac{3}{2}\left(\frac{P}{2}\right)L_m(i_{qs} i_{dr} - i_{ds} i_{qr}) \tag{2.38}$$

$$T_e = \frac{3}{2}\left(\frac{P}{2}\right)(\Psi_{dr} i_{qr} - \Psi_{qr} i_{dr}) \tag{2.37}$$

These equations give the complete model of electro-mechanical dynamics of an induction machine in synchronously rotating frame.

The equivalent two-phase representation of a three-phase induction machine can be given as in the following figure:

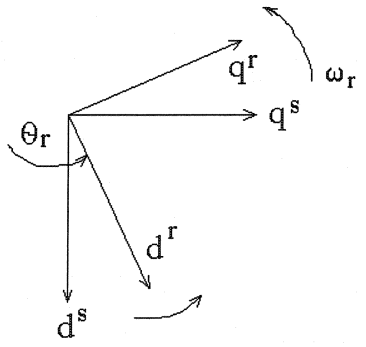


Fig. 2.2. Equivalent two-phase representation of a three-phase induction machine.

A three-phase machine is represented as equivalent two-phase machine as shown in the above figure, where $d^s - q^s$ correspond to stator direct and quadrature axis, and $d^r - q^r$ correspond, respectively, to rotor axis. The assumptions made so far are: (a) eddy current and hysteresis losses are negligible, (b) EMF induced is sinusoidal, (c) saturation is neglected, (d) there are no field current dynamics or coils imperfections, (e) stator resistance and load of the three phases are balanced.

2.3 Dynamic Model of the IM

For the purpose of computer simulations and transient analysis, the dynamic state-space equations of the induction machine are particularly important. The electrical variables employed in the model are picked as currents and fluxes. However, either currents or fluxes can be chosen. Usually, a rotating frame model is preferred in place of stationary reference frame. Defining the flux linkage variables in terms of base frequency of the induction machine as:

$$F_{qs} = \omega_b \psi_{qs} \tag{2.38}$$

$$F_{qr} = \omega_b \psi_{qr} \tag{2.39}$$

$$F_{ds} = \omega_b \psi_{ds} \tag{2.40}$$

$$F_{dr} = \omega_b \psi_{dr} \tag{2.41}$$

where ω_b is the base frequency of the induction machine. Using the above flux linkages in equations (2.2), (2.3) and (2.6), (2.7), the following dynamic relations can be found, taking into account that rotor is a squirrel cage type (thus $v_{qr}= v_{dr}=0$):

$$v_{qs} = R_s i_{qs} + \frac{1}{\omega_b}\frac{dF_{qs}}{dt} + \frac{\omega_e}{\omega_b} F_{ds} \tag{2.42}$$

$$v_{ds} = R_s i_{ds} + \frac{1}{\omega_b}\frac{dF_{ds}}{dt} + \frac{\omega_e}{\omega_b} F_{qs} \tag{2.43}$$

$$0 = R_r i_{qr} + \frac{1}{\omega_b}\frac{dF_{qr}}{dt} + \frac{(\omega_e - \omega_r)}{\omega_b} F_{dr} \tag{2.44}$$

$$0 = R_r i_{dr} + \frac{1}{\omega_b}\frac{dF_{dr}}{dt} - \frac{(\omega_e - \omega_r)}{\omega_b} F_{qr} \tag{2.45}$$

Using relations (2.8) – (2.13) to find another flux linkage expressions we have to multiply them by ω_b, and rearranging them using stator and rotor leakage reactance, currents can be found in terms of flux linkages as following:

$$i_{qs} = \frac{1}{X_{ls}}\left(F_{qs} - F_{mq}\right) \tag{2.46}$$

$$i_{ds} = \frac{1}{X_{ls}}\left(F_{ds} - F_{md}\right) \tag{2.47}$$

$$i_{qr} = \frac{1}{X_{lr}}\left(F_{qr} - F_{mq}\right) \tag{2.48}$$

$$i_{qr} = \frac{1}{X_{lr}}\left(F_{dr} - F_{mq}\right) \tag{2.49}$$

where, $X_{ls} = \omega_b L_{ls}$; $X_{lr} = \omega_b L_{lr}$; $X_m = \omega_b L_m$ and are stator and rotor leakage reactance, and magnetizing reactance, correspondingly; ω_b : motor angular electrical base frequency; ω_e : stator angular electrical frequency; ω_r : rotor angular electrical speed.

From equations of currents in terms of flux linkages (2.46) – (2.49), the following relations can be found:

$$F_{mq} = X_{ml}\left(\frac{F_{qs}}{X_{ls}} + \frac{F_{qr}}{X_{lr}}\right) \tag{2.50}$$

$$F_{md} = X_{ml}\left(\frac{F_{ds}}{X_{ls}} + \frac{F_{dr}}{X_{lr}}\right) \tag{2.51}$$

where

$$X_{ml} = \frac{1}{\left(\frac{1}{X_m} + \frac{1}{X_{ls}} + \frac{1}{X_{lr}}\right)} \tag{2.52}$$

Substituting equations for currents (2.46) – (2.49) into the dynamic equations for the induction machine (2.42) – (2.45), and re-arranging them in the state-space form gives Krause's model equivalent circuit [77]:

$$\frac{dF_{qs}}{dt} = \omega_b \left[v_{qs} - \frac{\omega_e}{\omega_b} F_{ds} + \frac{R_s}{X_{ls}} \left(F_{mq} + F_{qs} \right) \right] \tag{2.53}$$

$$\frac{dF_{ds}}{dt} = \omega_b \left[v_{ds} - \frac{\omega_e}{\omega_b} F_{qs} + \frac{R_s}{X_{ls}} \left(F_{md} + F_{ds} \right) \right] \tag{2.54}$$

$$\frac{dF_{qr}}{dt} = \omega_b \left[v_{qr} - \frac{(\omega_e - \omega_r)}{\omega_b} F_{dr} + \frac{R_r}{X_{lr}} \left(F_{mq} - F_{qr} \right) \right] \tag{2.55}$$

$$\frac{dF_{dr}}{dt} = \omega_b \left[v_{qr} - \frac{(\omega_e - \omega_r)}{\omega_b} F_{qr} + \frac{R_r}{X_{lr}} \left(F_{mq} - F_{dr} \right) \right] \tag{2.56}$$

Inserting (2.50) and (2.51) in differential equations (2.53) - (2.56) and re-arranging it, bringing similar terms together so that the derivative of each state would be the function of only other state variables and model inputs. In this case, the equations of an induction squirrel-cage motor become [82]:

$$\frac{dF_{qs}}{dt} = \omega_b \left[v_{qs} - \frac{\omega_e}{\omega_b} F_{ds} + \frac{R_s}{X_{ls}} \left(\frac{X_{ml}}{X_{lr}} F_{qr} + \left(\frac{X_{ml}}{X_{lr}} - 1 \right) F_{qs} \right) \right] \tag{2.57}$$

$$\frac{dF_{ds}}{dt} = \omega_b \left[v_{ds} - \frac{\omega_e}{\omega_b} F_{ds} + \frac{R_s}{X_{ls}} \left(\frac{X_{ml}}{X_{lr}} F_{dr} + \left(\frac{X_{ml}}{X_{lr}} - 1 \right) F_{ds} \right) \right] \tag{2.58}$$

$$\frac{dF_{qr}}{dt} = \omega_b \left[- \frac{(\omega_e - \omega_r)}{\omega_b} F_{dr} + \frac{R_r}{X_{lr}} \left(\frac{X_{ml}}{X_{ls}} F_{qs} + \left(\frac{X_{ml}}{X_{lr}} - 1 \right) F_{qr} \right) \right] \tag{2.59}$$

$$\frac{dF_{dr}}{dt} = \omega_b \left[- \frac{(\omega_e - \omega_r)}{\omega_b} F_{qr} + \frac{R_r}{X_{lr}} \left(\frac{X_{ml}}{X_{ls}} F_{ds} + \left(\frac{X_{ml}}{X_{lr}} - 1 \right) F_{dr} \right) \right] \tag{2.60}$$

Similarly, one of the torque equations (2.28) can be re-written as:

$$\frac{d\omega_r}{dt} = \left(\frac{p}{2J} \right) (T_e - T_L) \tag{2.61}$$

Using the equations (2.57) – (2.61) it is easy to implement a simple block modular dynamic representation of induction machine for computer simulations. The proposed approach gives the direct access to any machine variable, .e.g. flux linkages or current variables, which are used in current loops of control system to calculate the three-phase currents as well as the torque and speed signals. The out-put power of the IM can also be calculated based on that.

2.4 Indirect Vector Control of the IM Drive

Vector control, or field – oriented control is a strategy when the torque control of induction machine is modelled to resemble a dc machine torque production, when flux may be adjusted by controlling the field current, and torque may be controlled independently of flux by changing the armature current. For induction machine, some difficulties arise because of the interactions between the stator and rotor fields whose orientation are not held exactly at ninety electrical degrees, but may vary depending on operational conditions.

For field orientation purposes the direct vector control method relies on controlling stator voltage and is dependant on sensing the flux in the airgap, usually by Hall-effect devices. The alternative scheme, the indirect vector control, does not use any kind of sensors in the air-gap, simplifying the drive layout and increasing reliability and sometimes, precision.

Essentially, the vector control schemes are very close, except the indirect vector control scheme uses the unit vector signals generated in feedforward manner. The derivation of control equations for indirect vector control can be made using d-q equivalent circuit of induction machine shown in Fig. 2.1. Because the rotor is directed on the direct synchronous axis and $\omega_{sl} = \omega_e - \omega_r$, the following relation is straight-forward:

$$\Theta_e = \int \omega_e dt = \int (\omega_r + \omega_{sl}) dt = \Theta_r + \Theta_{sl} \tag{2.62}$$

The rotor pole position is not absolute, it is slipping with frequency ω_{sl} with respect to the rotor.

Using the machine equations (2.6) and (2.7) and eliminating the rotor currents (which are inaccessible) from that by means of the rotor flux linkage expressions the following relations can be found [84]:

$$\frac{d\psi_{dr}}{dt} + \frac{R_r}{L_r}\psi_{dr} - \frac{L_m}{L_r}R_r i_{ds} - \omega_{sl}\psi_{qr} = 0 \tag{2.63}$$

$$\frac{d\psi_{qr}}{dt} + \frac{R_r}{L_r}\psi_{qr} - \frac{L_m}{L_r}R_r i_{qs} + \omega_{sl}\psi_{dr} = 0 \tag{2.64}$$

where the slip speed is: $\omega_{sl} = \omega_e - \omega_r$.

To reduce the number of variables in equations by one it is desirable that the resultant rotor flux linkage would be aligned with the direct axis:

$$\psi_{qr} = 0 \tag{2.65}$$

or,

$$\frac{d\psi_{qr}}{dt} = 0 \tag{2.66}$$

Implementing this condition for equations (2.63) and (2.64) we get the following relations (substituting $\overline{\psi_r} = \psi_{dr}$):

$$i_{ds}L_m = \frac{L_r}{R_r}\frac{d\overline{\psi_r}}{dt} + \overline{\psi_r} \tag{2.67}$$

$$\omega_{sl} = \frac{L_m R_r}{\overline{\psi_r} L_r} i_{qs} \tag{2.68}$$

When the motor is operating at steady state, the rotor flux is usually constant, hence from equation (2.67) one obtains:

$$\overline{\psi_r} = i_{ds} L_m \tag{2.69}$$

Figure 2.3 shows a simple indirect field – oriented control scheme for a current controlled PWM-fed induction motor. In this figure 2.3, the base drive circuit includes the hysteresis current controller block, that takes the three phase command currents from the command current generator and outputs the logic variables NA, NB and NC used to control the VSI.

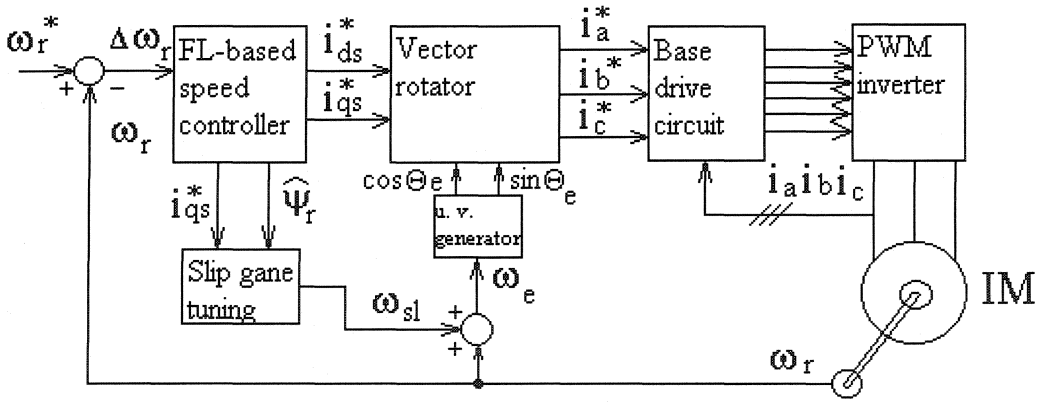


Fig. 2.3. Indirect vector control block diagram of IM drive with slip gain-tuning block.

To implement the vector control scheme, equations (2.62) - (2.68) have to be taken into consideration. Usually, the flux component of current i_{ds} * for the rotor flux desired is determined from (2.69) and kept constant in an open loop manner for simplicity. This approach produces some amount of error at the transients, because flux can be considered constant only at a steady state. This is why it is preferable to implement a flux estimation scheme for the drives that have multiple starts and stops in operation cycle. Fig. 2.4 shows the layout of a controller scheme that uses a slip gain tuning technique by means of fuzzy logic, which will be detailed later.

For the described indirect vector control the speed signal from the speed sensor is compulsory because the scheme locates the pole moving with speed ω_r in feedforward manner. In this case an absolute pole position on the rotor is not required the same as in the synchronous motor.

A dc machine –like vector controlled drive can be derived using equation (2.67), and the following equations are found as:

$$T_e = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_r}\psi_{dr}^{e} i_{qs}^{e} \tag{2.70}$$

$$\left(T_e - T_L\right) = \left(\frac{2}{P}\right) J \frac{d\omega_r}{dt} \tag{2.71}$$

If satisfied, the above conditional equations ensure the decoupling of the induction machine rotor equations. The extent to which decoupling is actually

achieved in the industrial applications actually depends on the accuracy of motor parameters. Because it is known that the values of rotor resistance and magnetising inductance vary more, than other parameters, the on-line parameter adaptive technique, such as fuzzy logic control principle is employed in this research.

2.5 Indirect Vector Control Slip Gain Tuning

In all operating conditions and quadrants of operation it is desirable that the control circuit operates with actual parameters of the IM to achieve robust decoupling control. The error in slip signal, which is a function of the machine's parameters, is a problem for an indirect vector control because of slip gain detuning. Especially often this problem appears at low speeds of drive operation.

The method used for slip gain tuning is the model referencing adaptive control (MRAC) principle discussed in [35]. The main idea is that the output of the reference block, which satisfies the tuned field orientation control and is the function of command currents, is compared with the adaptive model by means of fuzzy logic (FL) algorithms, and the resulting error is used to generate the slip gain constant. Consequently, slip gain tuning will take place when the output signal from the reference model matches the output signal from the adaptive model. The tuning realized in the proposed controller is based on the torque model, Fig. 2.4.

The slip gain coefficient Ks can be tuned so that the command torque would be exactly equal to torque produced by the induction machine at any moment of

time. The reference model output signal for the ideal vector control is given as following:

$$X^* = T_e^* = \frac{3}{2}\left(\frac{P}{2}\right)\frac{L_m}{L_r}\overline{\psi_r} * i_{qs}^* \tag{2.72}$$

Replacing flux term with current using (2.69), the relation can be re-written as:

$$X^* = T_e^* = \frac{3}{2}\left(\frac{P}{2}\right)\frac{L_m^{\,2}}{L_r} i_{ds} * i_{qs}^* \tag{2.73}$$

The actual torque is calculated from stationary frame variables for the slip gain control purposes using adaptive model relation (2.37):

$$X = T_e = \frac{3}{2}\left(\frac{P}{2}\right)(\Psi_{dm} i_{qs} - \Psi_{qm} i_{ds}) \tag{2.37}$$

Fuzzy logic is used in place of conventional P-I compensator to address the variation of inductances and variation of stator resistance at low speeds. The block scheme of indirect vector control with fuzzy logic gain slip tuning is shown on Fig. 2.4.

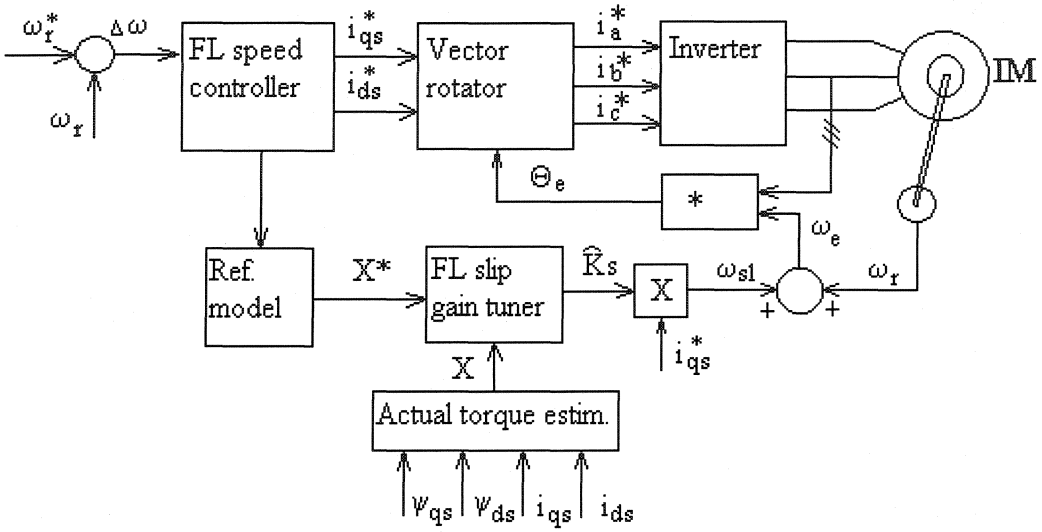


Fig. 2.4. Indirect vector control with fuzzy logic gain slip tuning block scheme.

The respective scale factors are chosen for the fuzzy logic controller to derive the per unit variables for manipulation. The controller generates the corrective incremental slip gain ΔK_s, using the brought together detuning error and it's derivative. The main idea behind this is to arrange a variable feedback control and convergence at any moment of operation. When the reference model and actual torque estimation signals match, this is an ideal condition and the slip gain K_s will be set to the right value, K_{so}. Ideally, a tuned system gives a better transient respond. There will be shown later in simulation and experimental sections.

2.6 Current Controller and the Voltage Source Inverter

During the control circuit operation, the vector rotator generates the a-b-c phase command currents, which are used by a current controller to command the voltage source inverter (VSI) and to produce the motor stator command currents. The outputs of the current controller implemented by means of the DSP are the firing pulses, which drive the inverter switches. Therefore, the feedback control is used for the control scheme with the VSI that forces the motor to follow the command speed.

The hysteresis current controller is used very often in industrial applications requiring high performance, because of its ease of implementation and simplicity while providing good satisfactory results. The current control principle for the VSI is used in this work and is based on the hysteresis controller with a fixed-band. The other popular type of hysteresis controller with sinusoidal varying band significantly increases switching frequency at the function zero crossings, therefore producing high losses

The control scheme generates the command currents using the error between the command speed and the actual rotor speed of the induction motor. This is made to force the actual machine stator current to follow the command current as close as possible, at the same time forcing the motor to follow the command speed.

The voltage source for the inverter is composed of a rectifier and dc link, incorporating a capacitor and inductor as shown in Fig. 2.5. The voltage source inverter can operate in both the PWM mode and in so-called square-wave mode, characterized by rectangular waveforms of the output voltage. The square-wave operation produces the highest voltage gain of the inverter, but the quality of output current is poorer than that in the PWM mode.

Another possible layout is the use of the current-source, where a controlled rectifier provides supply with closed-loop current control and the inductive dc link. The inverter differs from its voltage source opposite by the absence of freewheeling diodes, which are not necessary because the constant input current can never be negative. The output currents waveforms appear as the rippled sinusoids. The most common in practice are the two-level voltage source inverters, which are usually used in high-power drives applications. The dynamics of current phase control of these inverters is very good and they are more robust than voltage source inverters. The current quality in the square-wave mode of operation is poor, though. On the other hand, the voltage source inverters (VSI), in the PWM mode, allow control of both the frequency and magnitude of the output voltages, current source inverters are incapable of the magnitude control of output currents. This is why the VSI is used in this work.

Furthermore, the current controller forces the motor stator currents to follow the a-b-c phase command currents as closely as possible. These command currents are produced by comparison between the command speed and the actual speed of the motor for the purpose of generating a motor speed identical to the command speed. Therefore, under the speed control logic, the current controller forces the motor to follow the command speed closely.

The three-phase source inverter (VSI) is shown in Figure 2.5. The voltage source for the inverter is made up from a rectifier and the dc-link, usually composed of two elements: a capacitor and an inductor.

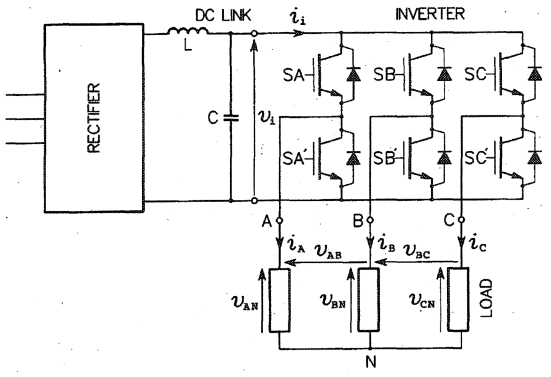


Fig. 2.5. Current controlled voltage source inverter for the IM drive diagram.

If an ac machine fed from the inverter operates as a motor, the average input current is positive. However, the instantaneous input current may assume negative values, absorbed by the dc-link capacitor, which, therefore, is necessary. The capacitor also serves a source of the high-frequency ac component of i_i, so that it is not drawn from the power system via the rectifier. In addition, the dc link capacitor smoothes and stabilizes the voltage produced by the rectifier. The optional dc-link inductor is less important, being introduced to provide an extra screen for the power system from the high-frequency current drawn by the inverter.

The operation of the current-controlled VSI can be described as follows: The errors between the actual motor currents and the command currents are used by the hysteresis current controller to produce the driving pulses for the transistors of the inverter. Each motor stator phase is connected to the corresponding leg of the three-phase VSI. The center point of the two equal-valued capacitors is considered the ground. The stator neutral is not connected to this assumed ground.

If referred to the Figure 2.5, the SA, SB and SC represent three binary logic variables of the three legs of the inverter, which determine the conduction state of

the inverter. When SA is 1, transistor 1 (T_1) is conducting and transistor 4 (T_4), is not, and when SA is 0, T_4 is conducting and T_1 is not. SB and SC have the same conduction principle in work.

The voltage source inverters are so controlled that one switch connects another of the input terminals to the output terminal, and potentials of all three-output terminals are always known. To avoid the short through, turn-on of a switch is delayed a little with respect to turn-off of the other switch. This delay has the order of few microseconds and is called a dead time or blanking time.

2.7 The Unconnected Neutral Effects

In the unconnected neutral layout the individual line-to-neutral voltages are dependent on each other, therefore each line current will be dependent not only on the corresponding inverter phase state, but also on the state of the other two phases. Thus, in general, if the neutral of the motor stator is not connected to the dc bus midpoint, the switching of any one phase depends on the states of the other two phases. This means that the current controller is dealing with an interchange of voltages and currents between the phases of the inverter, and there is a possibility that the actual current may not follow the command current accurately. Because of this the experiments have to be conducted with a full scale of measurements, and the obtained results have to be assessed accurately.

Chapter 3

Fuzzy Logic Speed Controller

3.1 General Information

As it was featured in detail in chapter two, it is often difficult to develop an accurate system mathematical model for an induction motor drive due to unknown load variation, unknown and unavoidable parameter variations arising out of saturation, temperature variations, and system disturbances. In order to overcome the above problems, recently, the fuzzy-logic controller (FLC) is used for induction motor control purpose.

In order to force the motor rotor to run with the command speed, industrial applications employ the PI and PID schemes. These two controller types are the most often used command current generating algorithms for controlling the current-controlled VSI. Usually, the speed error from command speed and actual speed of the motor is used to generate the necessary command torque. The reference torque is used to calculate the required two and then three axis currents for the induction motor in order to maintain the desired speed. According to the speed controller layout the voltage source inverter (VSI) is used to apply calculated currents to the

stator windings. The rotor speed is made to closely follow the reference speed by constantly adjusting the applied stator currents according to command signal.

The precision of the command torque calculations depends on the mathematical equations of the motor drive and the exact motor parameters, such as motor resistance and reactance. As mentioned previously, these parameters can vary during operation under different speed and loading conditions and therefore influence the air gap flux and reactance parameters. The effect of changing parameters is especially noticeable on low speeds of operation. For the PI and PID based control systems the motor parameters are assumed to be constant. However, the PI and PID changes in parameters affect the performance of induction motor drives with this type of controllers. Moreover, the conventional PI and notably PID controllers are sensitive to step changes of load disturbances and command speed. Thus, the effective control of the IM drive needs a complex structure for high performance applications. Because of the mentioned drawbacks of conventional controllers, the intelligent controllers, and, namely, fuzzy-logic controller (FLC) is considered for use with an induction motor control.

As compared to the conventional PI, PID, and their adaptive versions, the FLC has some advantages such as:

(1) it does not need any exact system mathematical model,
(2) it can handle non-linearity of arbitrary complexity, and
(3) it is based on the linguistic rules with an if-then general structure, which is the basis of human logic.

However, the application of FLC has faced some disadvantages during hardware and software implementation due to its high computational burden.

The FLC can handle non-linear functions of any arbitrary complexity, and as a matter of fact, the applied logic can be sufficiently modified and tuned. The main advantage the fuzzy logic controller has over conventional PI and PID controllers (including adaptive MRAC, SMC, etc.) is that it does not need the exact system parameters. In the real-world system these parameters, such as reactance or resistance, could be changing dynamically. Thus, the fuzzy logic controller does not need the exact system mathematical model. The use of fuzzy logic controller (FLC) for the purpose of reference speed, torque, slip and current control eliminates considerable amount of electric motor and drive system parameter variations problem. Thus it makes control more universal.

3.2 Fundamentals of Fuzzy Logic Control

The mathematical tool for the FLC is the fuzzy set theory introduced by Zadeh [52]. Fuzzy controllers work similar to human logic or human operators, who seldom process strictly numerical data. Instead, humans tend to issue and follow such inexact control commands as "more", "slowly", or "a bit to the left". With respect to signals in fuzzy control systems, fuzzy (linguistic) values, such as "low", "medium", or "high", are employed in the control strategy. Each fuzzy value represents a fuzzy set, which requires a specified range of crisp (numerical) values. Fuzzy logic can be looked upon as an expansion of Boolean logic that is intended to handle the concept of "partial truth". Such as the truth values lie between "completely true" and "completely false" – between 1 and 0 [53]. For instance, a statement in fuzzy logic may be true to a degree of 0.3, not just 1 or 0. This is called the "degree of belongingness" of the variable to the crisp value. A

membership grade which is the degree of belongingness of each crisp value in the set, is realised by the means of membership function. In practice, various membership functions are used, such as triangular, trapezoidal, or bell-shaped.

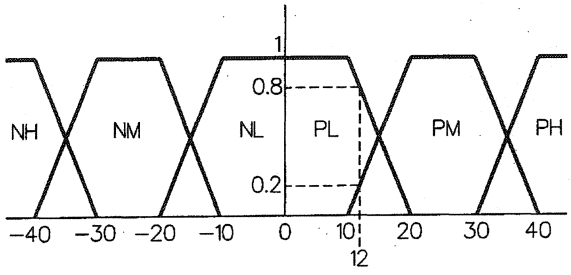


Figure 3.1. An example of fuzzy sets for torque signal of induction motor.

The concept of fuzzy variables and membership functions is illustrated in Figure 3.1 for the torque signal of an example motor, where torque is plotted on the horizontal axis. The whole torque range is divided into six overlapping fuzzy sets of numerical values. To each of these sets, a linguistic value is assigned. In this particular example trapezoidal membership functions are used.

A typical fuzzy control system is illustrated in Figure 3.2. The (*C*) and (*F*) symbols represent crisp and fuzzy values, respectively. Raw output signals from the controlled plant are pre-processed and applied to the fuzzyfier. The pre-processing includes determination of control errors, and it may involve calculation of these controlled variables, which cannot be measured directly. Based on predefined membership functions, the fuzzyfier assigns one or more of fuzzy values to each crisp variable received. The resulting fuzzy variables, with their membership grades assigned, are then forwarded to the inference engine, which is the part of the fuzzy controller that performs the main burden of computations. The

membership grades can also be thought of as "weights" or degree of belongingness of individual existing fuzzy inputs to the inference fuzzy engine.

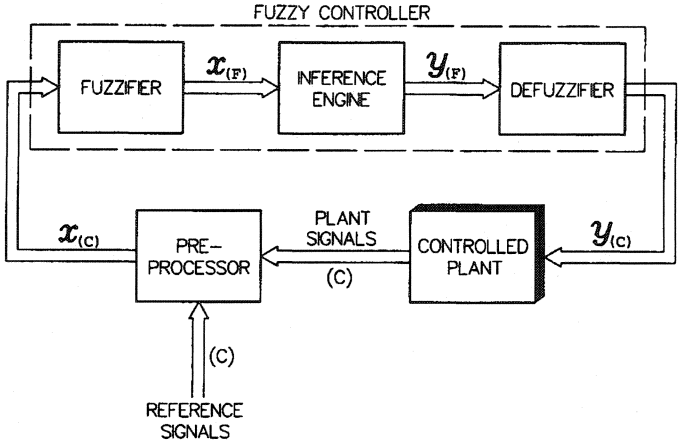


Figure 3.2. Block diagram of fuzzy control system.

The consolidated output set of the fuzzy inference engine is a combination of weighted fuzzy sets calculated from individual expert rules. The weight of a rule, formally called a firing strength, is a result of calculation from membership grades of the inputs involved in the process. With several inputs considered, the weight of the rule can, for example, be taken as a product of their individual membership degrees. The last part of the fuzzy controller employed in the process of output calculation, is the defuzzifier, it extracts a crisp value of the control signal from the output set receiving reference output for the controlled plant. In general, the fuzzy controller can be multidimensional, generating several control signals in a parallel manner, or interconnected, generating signals, that are functions of one another. With the progress of fuzzy logic theory, several defuzzification methods have been developed, the most common of them, so-called centroid technique

consisting in determination, using specific formulas, of the gravity centre of the output set of the inference engine.

To give an example of a typical fuzzy logic controlled system, the general structure of a fuzzy feedback control system is shown in Figure 3.3. The two inputs to the system, the loop error *E* and change in error *CE* signals are converted to the respective per unit signals *e* and *ce* by dividing by the respective scale factors, that is *e=E/GE* and *ce=CE/GC*, which are chosen beforehand by analysis and trial and error. Similarly, the output plant reference control signal *U* is derived by multiplying the per unit output by the scale factor *GU*, that is, *DU=du* GU*, and then summed to generate the output *U* signal.

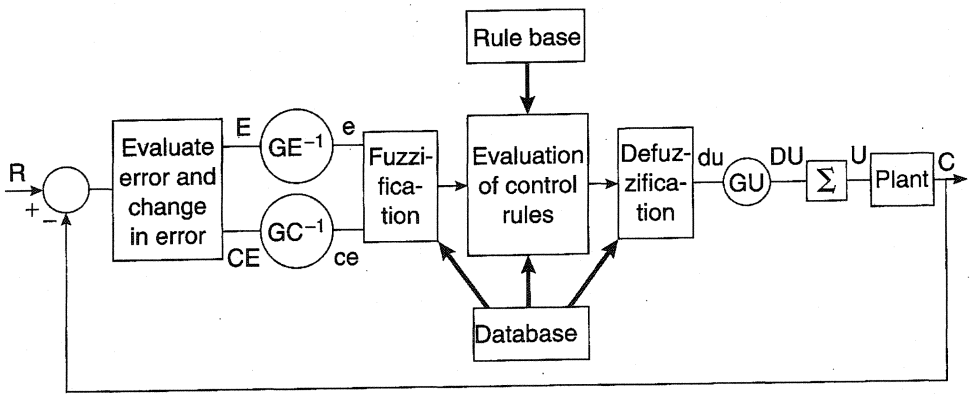


Figure 3.3. Structure of fuzzy control in controller feedback system.

An advantage of the proposed fuzzy control in terms of per unit variables is universal, the same control algorithm can be readily applied to all the plants of the same family. Besides, it becomes convenient and simple to design the fuzzy controller. The scale factors can be of constant or programmable types. Programmable scale factors can control the sensitivity of operation in different

regions of control. This type of programmable scale factor strategy can be applied in other similar control loops.

There are two methods of implementation for the fuzzy control in simulations and in real applications. The first involves continuos ongoing mathematical computation for the processes of fuzzification, evaluation of membership functions and control rules, and after that the defuzzification in real time. This is the generally accepted method. The drawback of this method is that it requires relatively big computational burden. If fuzzy controller operates with a large number of membership functions and rules, the computational burden can become intolerable. The second method is the look-up table method, where all the input and output static mapping computation (fuzzification, evaluation of control rules, and defuzzification) is done ahead of time and stored in the form of a large look-up table for real-time implementation. Instead of one look-up table, there may be hierarchical tables (e.g. coarse, medium, and fine). Look-up tables require large amounts of memory for precision control, but their execution may be fast as compared to the first approach.

3.3 Fuzzy Logic Control Implementation

According to the fuzzy logic theory, introduced by Zadeh [52], fuzzy sets play an important role in fuzzy interference engine. For a better interpretation, fuzzy sets are often represented graphically in the form of fuzzy membership functions. The fuzzy set A on the universe X is defined by a membership function, m_A from X to the real interval [0,1], which associates a number $m_A(x) \in [0,1]$ to

each element x of universe X. The membership function $m_A(x)$ represents the degree of belongingness of variable x to the fuzzy universe (set) A, in the other words, a biased value for the A-ness of variable x. For instance, the equation $m_A(x) = 0.3$ means x has A-ness of about 30%.

The simplest fuzzy set used in practice is a singleton function, which supports only one element - x_o. Mathematically, it can be described as a fuzzy set $F(x_o) = m(x)|_{x_o}$. Consequently, the fuzzy union is the combination of the fuzzy singletons of all the elements that form it. In this example, all the elements x constitute the universe X. Nevertheless, in fuzzy set theory, the boundaries of the fuzzy sets can be changed, making the use of fuzzy logic especially convenient in systems with poorly defined mathematical models.

In practice, there are numerous choices for the membership functions. The most popular choices used for fuzzy sets are functions related to linguistic variable zero (*ZE*) in the limits *X*=[-1, 1], which are the membership functions (MF) of linguistic value *ZE* (zero): (a) triangular, (b) Gaussian function, based on Gaussian equation, (c) trapezoidal and (d) singleton. The name given to the function represents the graphical appearance of it's fuzzy set.

Mathematically, these membership functions can be defined as:

a) singleton:
$$f(x) = \begin{cases} 1, & x = x_o \\ 0, & x \neq x_o \end{cases} \qquad (3.1)$$

b) triangular:
$$f(x;a,b,c)=\begin{cases}0, & x\le a\\ \dfrac{x-a}{b-a}, & a\le x\le b\\ \dfrac{c-x}{c-b}, & b\le x\le c\\ 0, & x\ge c\end{cases} \tag{3.2}$$

c) trapezoidal:
$$f(x;a,b,c)=\begin{cases}0, & x\le a\\ \dfrac{x-a}{b-a}, & a\le x\le b\\ 1 & b\le x\le c\\ \dfrac{d-x}{d-c}, & c\le x\le d\\ 0, & x\ge d\end{cases} \tag{3.3}$$

d) Gaussian function:
$$f(x;\sigma,c)=e^{\frac{-(x-c)^2}{2\sigma^2}} \tag{3.4}$$

The selection of fuzzy logic membership functions depends on the one's preference and practice, and is often changed during the experimentation. The designer has also an option of identifying his/her own fuzzy membership functions. In the above mentioned examples, one can see that the point 0 (zero) entirely belongs to the fuzzy set, while the numbers -1 and +1 do not. This is not necessarily the case for the other possible choices of fuzzy membership functions. Asymmetrical and closed (functions which are not open to the right or left) MFs can be synthesized using two sigmoidal functions and product sigmoidal (product of two sigmoid functions). Curves, which are based on polynomials, can be represented with several functions, including symmetrical polynomial curve open

to the left and it's mirror image, open to the right. Another type of these MFs is a function which is zero at both ends but has a rise in the middle of it.

In addition to the above-mentioned types of membership functions (MF), the user can easily produce any arbitrary MF.

The FLC implements human perception and experience through the membership functions and rules. Fuzzy inference is the complete process of formulating the mapping of the function from a given input to an output using fuzzy logic. The two basic types of fuzzy inference methods are emphasized: Mamdani and Sugeno types [85]. The main difference between these two methods is in the way the output is defined. The practice has shown that in the area of control applications Mamdani type fuzzy inference engine is the most preferred method. Thus, the proposed work also utilises this method for the fuzzy speed controller.

The output of the controller using the Mamdani method is defined by the centroid of a two-dimensional function. Fuzzy inference consists of three main steps, which are given as follows:

(a) fuzzification process,

(b) rule base evaluation process,

(c) defuzzification process.

The representation of MFs for fuzzy system can be made by mathematical functions, segmented strait lines (for simple triangular, quadrangle or trapezoidal shapes, etc.) and stored in memory look-up tables.

3.3.1 Fuzzification Process

At the beginning of fuzzy inference process, the inputs are taken and the appropriate membership functions are applied to them to conclude the degree to which they belong to each of the appropriate fuzzy sets using the fuzzy membership functions. This process of translating a numerical variable (real number, or crisp value) into a linguistic variable (fuzzy value) is called fuzzification process. In the controller, the input to the interference engine is a numerical value, which is limited to the set input universe and the output is a value that represents the input's fuzzy degree of membership to the linguistic set that meets the criteria. Practically, fuzzification of an input is obtained via equations of pre-set membership functions from a range of fuzzy sets of that input.

3.3.2 Rule Base Evaluation Process

The rule evaluation process is controlled by application of the number of conditional statements, such as "IF", "AND", "THEN", etc., where both the antecedent ("IF") and the consequent ("THEN") parts are expressed in linguistic form. The value of the output is calculated using a composition of the number of inputs through the fuzzy rule base matrix. The process of rule evaluation consists of two main processes: (a) first the application of the fuzzy operator (AND or OR) in the antecedent, (b) inference from the antecedent to the consequent. A typical fuzzy inference rule can be written as follows:

Rule Ri : If $\Delta\omega$ is Ai and Δe is Bi then Ti is Ci.

where speed error $\Delta\omega$ and change of speed error Δe are the input linguistic variables, torque Ti is the output linguistic variable and A_i, B_i, and C_i are the labels of linguistic variables $\Delta\omega$, Δe and T, respectively. If the antecedent is true to some degree of membership then the consequent is also true to that same degree of membership. The fuzzy operators used for fuzzy rules of inference engine are AND ($\cap$), OR ($\cup$) and NOT ($^{-}$), defined below:

(a) AND means classical intersection: $m_{A\cap B} = \min\{m_A(x), m_B(x)\}$

(b) OR means classical union: $m_{A\cup B} = \max\{m_A(x), m_B(x)\}$

(c) NOT means classical compliment: $\bar{m}_A = 1 - m_A(x)$.

Therefore, according to rule , the rule R_i, $m_{ci}(x)=\min\{m_{Ai}(x), m_{Bi}(x)\}$. The fuzzy rules can be established using the following approaches:

(a) from control engineering knowledge and expert experience,

(b) from a process model,

(c) from the model of human behaviour and learning process.

The Mamdani method implication is the most important and well established, used by researchers and works published in literature. The inference engine or the rule firing can of two basic types: (a) composition based and (b) individual rule based inferences.

In most part of the FLC usage the individual rule based inference is preferred because it is computationally efficient and does not need much computer memory. Composition based inference is equivalent to the individual rule based

inference in the case of Mamdani-type implication used to represent the meaning of the individual rules.

3.3.3 Defuzzification

The process that is completed by fuzzy inference engine and generates the final output is called the defuzzification process. A combination of consequents of each fuzzy rule is called the aggregation. The aggregation process must be evaluated, and it is another process, that has to be completed before the process of defuzzification can be completed.

Fuzzy set (combined output of each rule) is an input to the defuzzification process, producing a crisp value, which is a single, non-fuzzy number. Testing and combining of all the rules in the fuzzy inference system, the controller produces the resultant decisions. The aggregation process is called the process of combining the fuzzy sets that represent the output of each rule into a single fuzzy set.

The defuzzification process is the process that can be described as the reverse of the fuzzification process. The combined output of each rule (the result of aggregation process) is the input for defuzzification and a single number is the output of it, which is the union of all the outputs of individual rules that are validated or "fired". Various different defuzzification methods exist. The main methods used by researchers in their works recently are: (a) centre of sums method, (b) centre of largest area method, (c) centre of gravity method, (d) height method, etc. The centre of gravity method was used in this work, because it is a compromise in accuracy of the outcome and the computational burden imposed in calculating it. The centre of gravity method can be evaluated as:

$$I = \frac{\sum_{i=1}^{N} I_i m_{ci}(I_i)}{\sum_{i=1}^{N} m_{ci}(I_i)} \tag{3.5}$$

where N is the total number of rules, $m_{ci}(I_i)$ denotes the output membership grade for the i^{th} rule with the output subset of I_i, and I is the output fuzzy set.

3.4 Fuzzy Logic Controller for IM Drive

In the previous chapter it was discussed that at start-up, steady state and low speed command or speed/torque command change, reactance and resistance parameters change, saturation and armature reaction effects affect the operation of IM drive. Also, the d-axis stator current $i_d = 0$ technique is the source of error and therefore decreases efficacy of the drive. The consequence of even slight variation of parameters can produce significant distortion of the air gap flux and may cause the reactance parameters to vary with different operating conditions when the IM drive is controlled with the means of a conventional controller. These non-linearities can affect the performance of the drive at different dynamic operating conditions, which is unacceptable if the drive is used in some high-performance applications. The use of artificial intelligence based drive controllers, which are capable of handling non-linear functions of arbitrary complexity can overcome these complications.

A fuzzy logic controller has been proposed for an IM drive, which gives the following benefits when used in high performance systems:

1) The FLC does not need any information of the exact system mathematical model; therefore the system exact mathematical model is not required. This feature makes it one of the best approaches for control of an IM drive, where the exact system mathematical model is difficult to obtain because it is relatively complicated and includes dynamically varying parameters.

2) Because the FLC has self-adaptive capabilities, it is an exceptional tool for optimum control of the real-time IM drives, especially when resistance is changing with temperature, reactance parameters change according to the conditions of operation, inertia follows varying mechanical load and command speed may be changed on demand. The FLC can self-adapt to these changing circumstances and for this reason it can be used as a preferred controller for the IM drive. Moreover, because fuzzy logic is also based on linguistic control variables and rules, it can be used for an induction machine in order to simplify the control task for this non-linear system.

3) The practical realization of high-performance drive (HPD) controllers requires sophisticated, and often expensive, hardware, such as high-speed digital signal processors, current and speed sensors, and PCs. Therefore the cost of the unit can be relatively high. High cost of controlling hardware and software can be tolerated, if the area of implementation is crucially important. Keeping the cost down would allow broadening the use of robust high-performance, especially intelligent controllers in real-time applications. Even as the use of the proposed FLC also needs some

specialized hardware and software, it is less complex than other similar intelligent controllers, and therefore is the most competitive considering the performance requirements.

Taking into the account the above-described requirements and other specific challenges, a detailed fuzzy controller for the induction motor drive is designed in the remaining part of this chapter. Under the indirect vector control scheme the fuzzy logic controller is used in the outer loop to optimize the speed control, together with the current controller used in the inner loop. The selection of current controller and VSI was made based on analysis in chapter two.

3.4.1 FLC Structure for the IM Drive

As stated before, the motor dynamics can be described by the following equations:

$$T_e = J_m p\omega_r + B_m\omega_r + T_L \tag{3.6}$$

and,

$$T_e - T_L = J_m\left(\frac{p}{2J}\right)\frac{d\omega_r}{dt} \tag{3.7}$$

$$\frac{d\Theta_r}{dt} = \omega_r \tag{3.8}$$

where J_m is the rotor inertia constant, T_e is the electrical torque, T_L is the load torque, B_m is the friction damping coefficient, p is the differential operator and ω_r is rotor speed.

When modeling a system with unknown non-linear mechanical characteristics it is considered to use FLC because of it characteristics it has, capable of handling any non-linearity, as for example, an unpredictable load pattern change. The load can be modeled using the following equation as [69]:

$$T_L = A\omega_r^{\,2} + B\omega_r + C \tag{3.9}$$

where coefficients A, B and C are some arbitrary constants.

Using equations (3.6) and (3.7) it is possible to combine these expressions to obtain a single input for the system in continuous time domain form. This would make the control undertaking easier, and the resulting equation of an IM will be as following:

$$J_m \frac{d\omega_r}{dt} = T_e - (B_m + B)\omega_r - A\omega_r^2 - C \tag{3.10}$$

Employing a small signal model for IM system analysis one can see that a small incremental change ΔT_e of the electrical torque T_e produces an equivalent incremental change $\Delta\omega_r$ of the rotor speed ω_r. Rewriting equation (3.10) and using a small signal model variables the following relation is obtained:

$$J_m \frac{d(\Delta\omega_r)}{dt} = \Delta T_e - (B_m + B)(\Delta\omega_r) - A(\Delta\omega_r)^2 \tag{3.11}$$

By employing the finite differences instead of all the continuous variables in equation (3.11), the discrete time small signal model of the simplified induction machine with non-linear load applied in arbitrary pattern can be given as:

$$\Delta T_e(n) = \frac{-J_m}{t_s}\Delta e(n) + (B_m + B)\Delta\omega_r(n) + A\{\Delta\omega_r(n)\}^2 \tag{3.12}$$

This equation can be described in the form of function, where developed electric machine torque is shown to be a function of speed error and change of speed error:

$$T_e(n) = \int_{discrete}\Delta T_e(n) = f(\Delta e(n), \Delta\omega_r(n)) \tag{3.13}$$

where, $\Delta\omega_r(n) = \omega_r^*(n) - \omega_r(n)$ is the present sample of speed error, $\Delta e(n) = \Delta\omega_r(n) - \Delta\omega_r(n-1)$ is the change of speed error, $\Delta\omega_r(n-1)$ is the past sample of speed error, $\omega_r(n)$ is the present sample of actual speed, $\omega_r^*(n)$ is the present sample of command speed, t_s is the sampling time interval and "*f*" denotes the non-linear function relationship.

Ideally, a vector-controlled induction motor drive operates like a separately excited dc motor drive. Neglecting the armature reaction and field saturation, the developed torque is given by $T_e = K_t' I_a I_f$, where I_a is an armature current and I_f is field current.

With vector control of IM, i_{ds} is analogous to field current of dc machine I_f and i_{qs} is analogous to armature current I_a of a dc machine. Therefore, the torque can be expressed as [86]:

$$T_e = K_t \Psi_r i_{qs} \tag{3.14}$$

or, using the relation between the rotor flux and the direct stator current component, the next relation can be developed:

$$T_e = K_t' i_{ds} i_{qs} \tag{3.15}$$

where, Ψ_r : peak value of the sinusoidal space vector;

i_{qs} : stator quadrature current, torque component;

i_{ds} : stator direct current, field component.

This dc machine-like performance is only possible if i_{ds} is oriented (aligned) in the direction of flux Ψr and i_{qs} is established perpendicular to it, so when i_{qs}^* is controlled, it affects the actual i_{qs} current only, but does not affect the i_{qs} component of current. This vector orientation (of field orientation) of currents is crucial under all operating conditions in a vector-controlled drive.

The fuzzy logic (FL) algorithm of speed controller employed in the IM drive outer loop is based on estimation of two errors: speed error and change of speed error. This two linguistic variables are considered as an input to the system of accordingly interconnected FL blocks, and the output are the torque and field (quadrature and direct) current components, respectively of the stator current. The following functional relations can be established.

From (3.14), (3.15) the equation of torque as a function of synchronously rotating direct and quadrature stator and rotor currents is given as:

$$i_{qs}(n) = \int_{discrete} \Delta i_{qs}(n) = f(\Delta T_e(n), \Delta i_{ds}(n)) \tag{3.16}$$

and,

$$i_{ds}(n) = \int_{discrete} \Delta i_{ds}(n) = f(\Delta T_e(n), \Delta i_{qs}(n)) \tag{3.17}$$

From the above developed relation for command torque T_e, equations (3.16) and (3.17) are used to calculate the necessary quadrature and direct -axis currents to produce the required rotor speed ω_r. In the following stages of transformations, after the controller block according to the scheme layout, in the real-time the motor position signal and the output signals of the simplified FLC in terms of the command quadrature axis and direct axis currents i_q^* and i_d^* are utilized to calculate the motor command currents i_a^*, i_b^* and i_c^* by using Park's transformation.

When designing the block-schema of the controller, the first FL block (FLC #1) returns command torque Te signal as given in (3.13), which is used for command stator currents production in two next bocks FLC #2 and FLC #3 using equations (3.16) and (3.17), respectively. This controller scheme calculates the necessary quadrature and direct -axis currents using fuzzy logic to get the required rotor speed ω_r, as it is shown in Fig. 3.4. In real-time, this currents are transformed using the respectful formulae to get the motor command phase currents i_a*, i_b* and i_c* using Park's transformation.

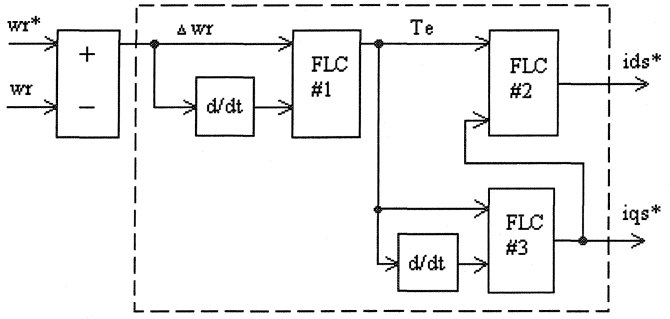


Fig. 3.4. Block diagram of proposed IM drive
speed controller using fuzzy logic blocks.

As seen in Fig. 3.4, FLC #3 actually uses a torque signal output of FLC #1 and it's derivative, which would be, in fact is a second derivative of change of speed error. This simplification is made to avoid excessive DSP processing time. Computer simulations have shown that there is very little effect of taking out the link input for direct stator current signal for FLC #3 and replacing it with torque signal input and change of torque signal input. The derivative blocks used can be replaced with time delay blocks, which is another way to get the required inputs. This notion would allow to further simplify the proposed scheme and shorten calculation burden, at the same time also secure the controller from uncertainties in the form of spikes in the output, which are the drawback of time derivative blocks, if the processed signal changes abruptly. The proposed changes and simplifications also have little effect on the output signal and therefore, were implemented readily. The more comprehensive list of changes and simplifications, as well as the relative effect of these changes are given in the next passage.

The block diagram of the proposed FLC based IM is shown in Fig. 3.5.

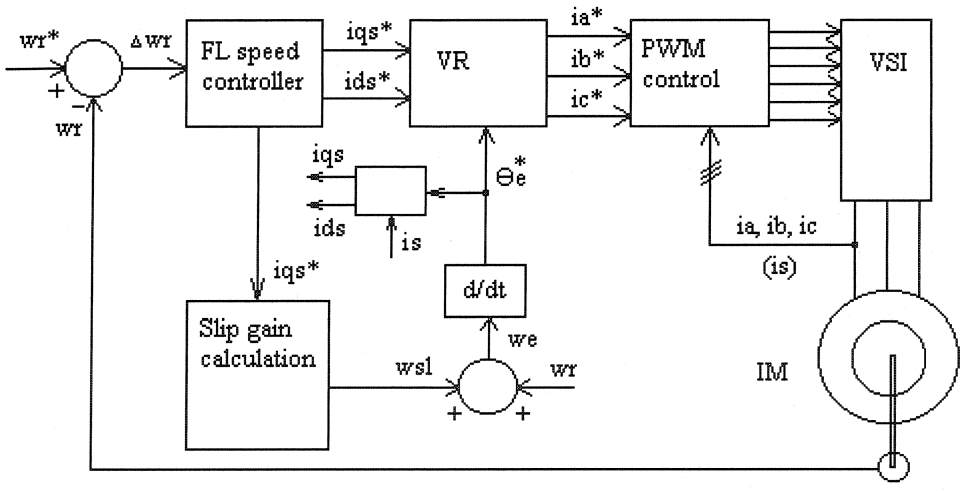


Fig. 3.5. Block diagram of proposed IM drive system incorporating slip gain tuning control.

It is to be noted that Fig. 3.5 incorporates the slip gain tuning in addition to the fuzzy logic controller. In this diagram VR represents a vector-rotating block.

3.4.2 Simplified FLC for IM Drive

As shown in 3.4.1 and Fig. 3.4, a simplification of the scheme is made at the start, replacing one of the closed inner loop inputs to block number three with a derivative of the second input. This simplifies the circuit and reduces computational burden significantly. Experimentation has proved that this manipulation has little effect on the output. Actually, it does not matter which of the two interconnected bocks is chosen for simplification, the outcome is similar. This simplification significantly reduces computational burden and lowers the computer memory to implement the proposed FLC scheme in real time.

The crucial part in developing the FLC for the IM is the definition of the input linguistic variables. The developed model, expressed by equations (3.13), (3.16) and (3.17) defines the input and output linguistic variables for the FLC of the IM drive. It also determines the layout of the controller and the number of blocks used in it. According to equation (3.13), the inputs of the first proposed FLC block are the current sample of speed error and the change of speed error, which is found as a difference between present and past samples of speed error. The change of speed error is found by means of the derivative block. Nonetheless, it has been observed that it is possible to replace the derivative block of the scheme with a time-delay block, which would provide a faster response and cleaner output. As mentioned, the drawback of derivative block implementation is the possibility of high spikes in the output as a consequence to abrupt changes of the input signal. By means of computer simulations it was observed that the effect of the interchange of

derivative blocks with time-delay blocks produces a faster response of the controller. The output signal for the studied range of drive operation is of better quality (less spikes) and the overall improvement of motor drive performance is observed together with the decrease of computational burden.

The use of the time-delay blocks instead of derivative blocks produces an FLC-based controller with acceptably robust response as well as the precisely accurate tracking of the command speed. It also allows raising the speed sensor sampling rate significantly.

Consequently, the simplified FLC makes a significant difference in a number of factors for real time implementation of the laboratory IM drive system with a fuzzy-based controller employed. The block diagram of the proposed FLC based IM drive incorporating slip gain tuning by means of a separate FL block is shown in Fig. 3.5. A Matlab Simulink implementation of the first controller block is given in Fig. 3.6.

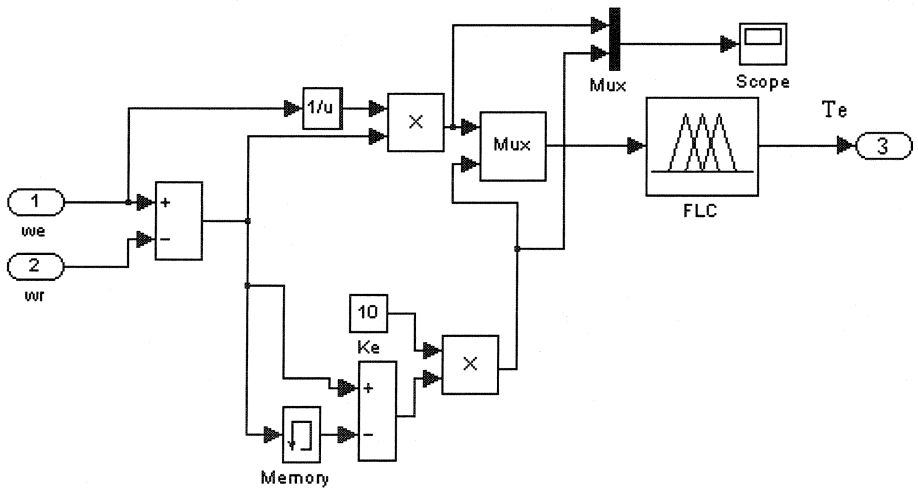


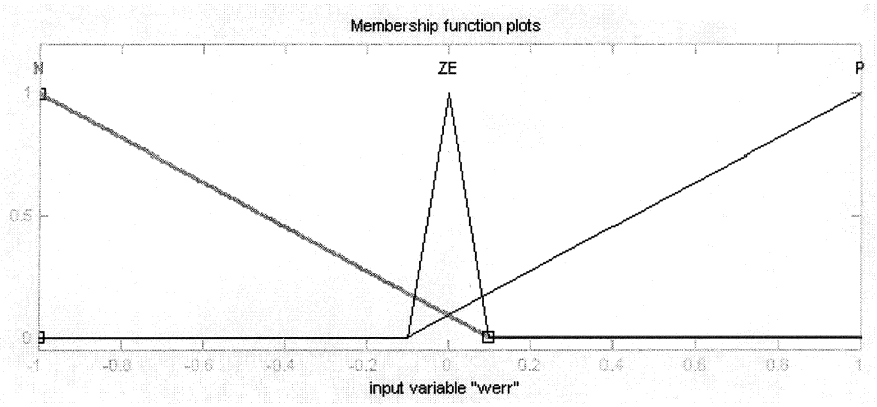
Fig. 3.6. A Matlab Simulink implementation of the 1st fuzzy logic block of FLC using memory block instead of derivative block.

The complete Matlab Simulink scheme of command current generator featuring three FLC blocks is given in Appendix B, Fig. B.2a.

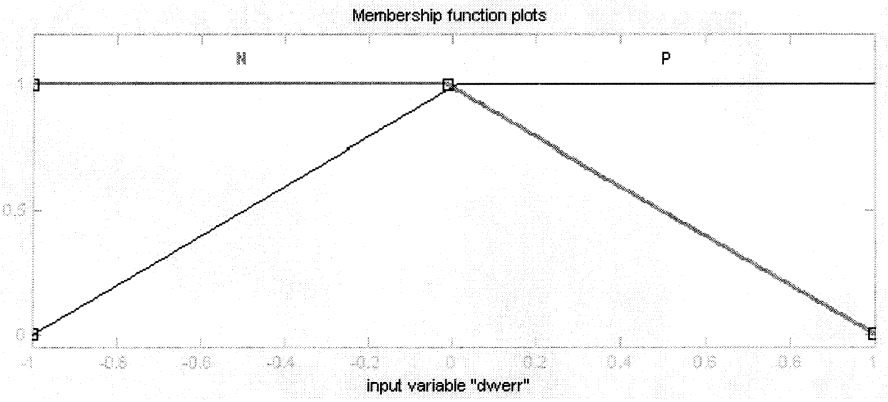
The next step is to choose the scaling factors, K_ω, K_e and K_i, which are chosen for fuzzification procedure and receiving the suitable output of the current command. The factors K_ω and K_e are chosen so that the normalized value of speed error and change of speed error, $\Delta\omega_{rn}$ and Δe_n, respectively, remain within the limits of $\pm$ 1. The factor for the output signal K_i is chosen so that the rated current is the output of the controller at all rated conditions. For implementation, the following values are taken as $K_\omega = \omega_r^*$ (command speed), $K_e = 10$ and $K_i = 10$ in order to get the optimum drive simulations and real-time performance.

The membership functions of $\Delta\omega_{rn}$ as well as i_{dn}^* and i_{qn}^* are chosen after selecting the scaling factors. Membership functions (MF) are important elements of the FLC. The following MFs are used for the input and output fuzzy sets of the first block for producing the torque signal of the developed controller, which are shown in Fig. 3.7. MFs for FL blocks producing direct and quadrature currents are similar to presented below. Triangular and trapezoidal functions are chosen to represent the membership functions for all the fuzzy sets with no exceptions. Triangular membership functions are used for the fuzzy sets ZE of the input vectors and all the fuzzy sets of the output vectors. The trapezoidal and triangular functions are used because of the ease of their mathematical representation. As a result it makes possible to simplify the implementation of fuzzy logic interference engine and to reduce computational burden for real-time operation.

(a)



(b)



(c)

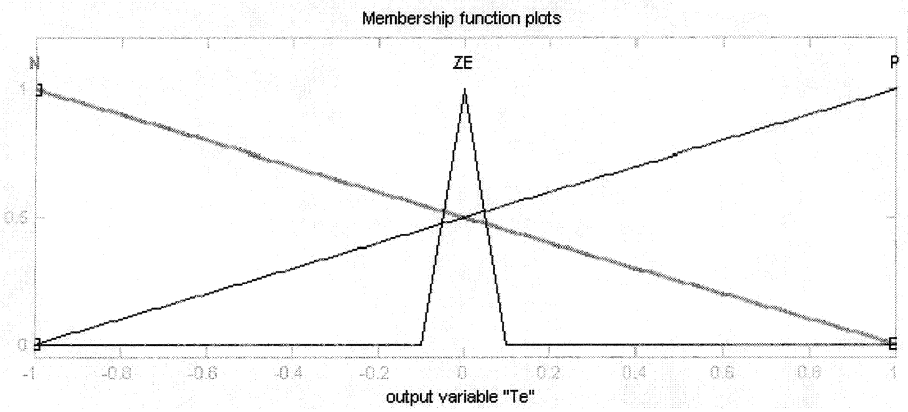


Fig. 3.7. Membership functions for: (a) speed error $\Delta\omega_m$, (b) change of speed error Δe_n , and (c) command torque T_e^*, implemented in Matlab Simulink.

Computer simulations and experimentation have proved that reduction of the number of fuzzy logic membership functions and fuzzy rules to experimentally verified minimum produces a negligible response on the motor speed and output torque. On the contrary, the large number of rules and membership functions increases the computational burden drastically. Thus, experimentally confirmed minimal number of membership functions for the two input vectors of $\Delta\omega_r(n)$ and $\Delta e(n)$ of the first FL block can be reduced to three and two, correspondingly. Minimal number of MFs for the output is three. The other two fuzzy blocks are simplified similarly. This returns a much-simplified FLC as compared with the non-simplified controller, thus lessening the computational burden adequately. Therefore, the computational requirement needed to implement the proposed FLC in real time is considerably lowered.

Further simplification can be achieved by taking out the second input of each block. This would also reduce the number of rules. It was found that despite the IM drive system keeps robust performance for the common start up and change of load procedures, some complications can arise if command signal or/and drive load are changing dynamically. Therefore, the further definitions should be made considering the drive system applications.

It is essential to mark the significance of this factor for IM drive system laboratory implementation, which was completed using a 200 MHz PIII computer with 128Mb RAM installed. Due to simplification procedures made to the fuzzy logic controller blocks a laboratory 1 hp induction motor was able to run in a wide range of speed, including higher than synchronous speeds, with a sampling frequency period T = 0.2e-3 sec.

The rules used of the proposed FLC algorithm differ somehow in three of the fuzzy logic blocks. Nevertheless, the core rules are left unchanged, which are as follows:

1) if $\Delta\omega_{rn}$ is P (positive), then T_e is P (positive).
2) if $\Delta\omega_{rn}$ is ZE (zero), then T_e is ZE (zero).
3) if $\Delta\omega_{rn}$ is N (negative), then T_e is N (negative).

Rules incorporating the second input are used for fine-tuning the performance of FLC and can be omitted if necessary. Mamdani type fuzzy inference and the centre of gravity method of defuzzification are used [85].

The exact fuzzy logic rules for the FLC1 block in Fig. 3.4 consist of six following linguistic expressions:

1) if $\Delta\omega_{rn}$ is N (negative) and Δe_n is N (negative), then T_e is ZE (zero)
2) if $\Delta\omega_{rn}$ is ZE (zero) and Δe_n is N (negative), then T_e is P (positive)
3) if $\Delta\omega_{rn}$ is P (positive) and Δe_n is N (negative), then T_e is P (positive)
4) if $\Delta\omega_{rn}$ is N (negative) and Δe_n is P (positive), then T_e is ZE (zero)
5) if $\Delta\omega_{rn}$ is ZE (zero) and Δe_n is P (positive), then T_e is ZE (zero)
6) if $\Delta\omega_{rn}$ is P (positive) and Δe_n is P (positive), then T_e is P (positive)

The fuzzy logic rules for the FLC2 block in Fig. 3.4 consist of the following linguistic expressions:

1) if ΔT_e is N (negative) and Δi_{qs} is N (negative), then i_{ds} is ZE (zero)

2) if ΔT_e is ZE (zero) and Δi_{qs} is N (negative), then i_{ds} is ZE (zero)

3) if ΔT_e is P (positive) and Δi_{qs} is N (negative), then i_{ds} is P (positive)

4) if ΔT_e is N (negative), then i_{ds} is N (negative)

5) if ΔT_e is P (positive), then i_{ds} is P (positive)

The fuzzy logic rules for the FLC3 block in Fig. 3.4 consist of the following linguistic expressions:

1) if ΔT_e is N (negative) and ΔdT_e is N (negative), then i_{qs} is ZE (zero)

2) if ΔT_e is P (positive) and ΔdT_e is N (negative), then i_{qs} is P (positive)

3) if ΔT_e is N (negative), then i_{qs} is N (negative)

4) if ΔT_e is ZE (zero), then i_{qs} is ZE (zero)

5) if ΔT_e is P (positive), then i_{qs} is P (positive)

The decision to use supplementary rules can be made based on the required sampling speed rate, required response time, available computational resources, etc. These rules evaluate the two inputs at the same time based on the "IF", "AND" and "THEN" logic operators for producing result.

In the scope of this work, the values of the constants, membership functions, fuzzy sets for the input and output variables and the rules were selected by experimental trial and error in order to achieve the best possible drive implementation.

3.5 Concluding Remarks

A novel interconnected layout scheme for the FLC IM drive control using the id $\neq 0$ technique and slip gain-tuning have been presented in this chapter. The fuzzy controller has self-adjustable capabilities and can be considered as a substitute for the traditional fixed gain PI and PID controllers or other types of modern controllers. By means of computer simulations and validating experimentation a minimum number of fuzzy rules and membership functions was found, which is three for both parameters. A specific simplified fuzzy logic controller is proposed for the IM motor drive as a high performance precise speed controller in order to deal with the real-time motor parameter variations, load and system disturbances problems. In order to validate the efficiency of proposed representation, a complete indirect vector control scheme of the drive is carried-out both in computer simulations and laboratory experiments. Computer simulations and experimental results as well as discussions are detailed in the next chapters.

Chapter 4

Computer Simulation of FLC -based Indirect Vector Control of the IM

4.1 Introduction

In order to foretell the performance of the proposed fuzzy logic based indirect vector controller for the IM drive with a slip tuning block, extensive computer simulations were made using the Matlab Simulink software, particularly with the Fuzzy Logic Toolbox [85]. To compare the proposed scheme with the other types of controllers, two other speed controllers have been implemented and tested. A conventional PI controller was used for the most of comparison tests, as well as a fuzzy logic based speed controller for indirect vector control of induction motor drive developed and tested recently by M.A. Rahman, T.S. Radwan and M.N. Uddin [77] for the purposes of controller characteristics comparison. The latest is a non-simplified FLC based controller operating under the i_d = 0 postulation, based on earlier works by M.A. Rahman and M.N. Uddin [63]. The computer simulations and laboratory experimental results of the proposed

simplified FLC based system with slip gain tuning are presented, as illustrated in Fig. 3.5. The nonlinear load model and sudden command speed change were used for simulation and experimentation purposes.

The VSI type inverter based on fast-switching IGBTs with a base drive unit has been employed in this work because of its suitable performance characteristics. To make the best use of the proposed inverter, a fixed-band hysteresis controller is used as the current controller due to its fast transient response, precisely accurate steady-state response and its robust characteristics over all the speed range when used for the purposes of motor control and particularly an IM drive.

The IGBT inverter complete circuits and the base drive circuit for the inverter switches are detailed in Appendix C. The details of the current control technique for the VSI were presented in Fig. 2.6 and part 2.6 of chapter two.

Matlab Simulink simulation model details are given in Appendix B. Computer simulation results have been presented and discussed below.

4.2 Simulation Results and Discussion

A drive system model described mathematically for simulation purposes is based on the laboratory experimental 1 hp, 4-pole IM parameters employed with 2 N-m as full rated load, fed by a current-controlled VSI. The high-performance drive capabilities have to include the ability of operation at different loading and changing command speed conditions, which are amongst the main criteria of the high performance control system. The other crucial feature is the ability to tolerate changing motor parameters.

Figure 4.1 (a) – (d) shows the induction motor speed response, command phase current and phase current in phase "a" of the motor, command quadrature -axis current, for the conventional PI -based IM drive with i_d equal to zero and no slip gain tuning, for the rated speed and no load conditions. The results show that the drive follows the command speed relatively fast with almost no overshoot or undershoot and with very low, almost none steady state error. The command q-axis and d-axis currents are very close and almost indistinguishable at the steady state. For this reason here and on the following graphs only the command quadrature current is shown, which also represents the command torque (since i_d has been forced to zero in this scheme). Conversely, this control current does not oscillate much, showing a tendency of minimization to some low constant value after start-up occurs and the drive reaches the speed close to the command at steady-state conditions. This does not appear to impact significantly the operational performance. Resulting graphs also show clearly that the actual current follows the command current conformably close.

On the next Figs. 4.2 (a)-(d) the drive system speed response, as well as the command phase current and actual phase current in phase "a", command quadrature -axis current for the PI -based IM drive with i_d equal to zero and no slip gain tuning at half load and rated speed conditions are shown (corresponding output torque, motor slip and corresponding command direct –axis current are not shown here). Once more it is seen that the drive follows the command speed with some overshoot and no undershoot, very low or almost zero steady state error. The command quadrature q-axis current, and hence the command torque, shows no oscillations at steady state. It starts from over 60 A, and settles at less than 2 A at the steady state. The actual phase current is close to sinusoidal and is relatively smooth.

Similar responses of the PI based IM drive with i_d equal to zero and no slip gain tuning (as in contrast to the proposed control topology with the slip gain tuning) are obtained for full load and rated speed conditions. The simulated graphs are shown in Figs. 4.3 (a)-(d). It is clearly seen, that under the full load and rated speed conditions, the phase current peak magnitude is approximately 3.5 Amperes, which overshoots by almost 10% of the rated current for the induction machine with the assumed parameters. This is due to the fact that the $i_d = 0$ approximation requires a higher stator command current than ideally with i_d not forced to zero. Once again, it is shown that the drive can follow the command speed smoothly with no overshoot and almost no steady state error. These results verify that the PI/i_d=0 based IM drive used for the purposes of comparison with the proposed simplified FLC based drive with slip gain tuning can follow the command rated speed at various loading conditions.

In order to evaluate drive performance in terms of response time, speed overshoot, transient, steady state errors and overall stability a step change of command speed is applied to the PI –based drive with i_d=0. This experiment is crucial because in high performance drive applications it is important to be able to step change command speeds in accordance with situational demands. This type of operation occurs relatively often. The responses of the drive including rotor speed, command phase current, q-axis command current, "a" -phase actual current, and start-up part together with steady-state actual phase current are shown in Fig. 4.5 (a) - (d), respectively, for a step change of command speed at half load conditions. Fig. 4.4 (a) - (d) and Fig. 4.6 (a) - (d) show correspondingly, the results of simulations of the speed step change applied with no load and full load conditions. The results of computer simulations show that the drive tracks the command speed without overshoot and steady-state error during a step change in reference speed,

such as a decrease of speed (from 188.5 rad./sec. to 150 rad./sec.) and after some time an increase of speed (from 150rad./sec. to 188.5rad./sec.). There is a small undershoot during a step decrease of speed (from 188.5rad./sec.to 150rad./sec.), but this undershoot is acceptable. There is also some overshoot at start up and when the rotor speed changes back to original value (from 150rad./sec. to 188.5rad./sec.), this overshoot is observed to increase slightly as the load increases. Again, quadrature -axis command current oscillates within permissible limits, the command torque oscillates significantly at start up, and the oscillations settle when the drive reaches the command speed. The torque oscillations do not appear to significantly impact drive performance. The results show that the PI -based/i_d=0 drive follows the step change of the command speed relatively close.

Another important feature of the high performance control system is the ability of the drive to withstand disturbances, namely, the most often occurring type – load disturbances. Change of load is a typical external disturbance and is a very common occurrence for the high performance drive. The rotor speed and the corresponding actual control current of the induction motor, quadrature -axis command current and an actual "a" -phase current including steady-state current responses for the PI -based i_d equals zero drive are shown in Figs. 4.7 (a)-(d), correspondingly, for a sudden increase of load at the rated speed. The motor is started with no load and at time t = 0.75 sec. the load is abruptly increased to full load. Therefore, this pattern models the sudden increase of load as a typical situation in industrial applications. It is observed that the drive speed is transiently reduced by exponential law (approximately by 5%), and after no load situation is restored, the command speed is quickly recovered with some low overshoot. The phase current response is very close to sinusoidal and follows the command current. The command quadrature current i_q^* corresponds to the pattern of changing

load. The machine torque oscillations are within permitted limits and do not appear to negatively affect drive performance.

The graphs for the simplified FLC based IM drive with fuzzy slip gain tuning featuring the speed response graph, the corresponding command phase current, command quadrature (q) -axis current and actual "a" -phase current are shown in Figs. 4.8 (a)-(d), respectively, for rated speed conditions and no load. The results show that the drive follows the command speed very quickly, 0.15 sec. faster than the best-tuned PI controller used for comparison purposes. There is no overshoot whatsoever as the steady state is reached and speed becomes equal to reference speed. Steady-state error is exceptionally close to zero. The quadrature command current i_q has a much lower magnitude (30% less) and no oscillations compared to the PI/i_d=0 -based drive, and the direct –axis current i_d is also very close to the quadrature component. This can be explained by the fact that, with i_d not forced to zero or some constant value, both q and d -axis components of current contribute to the command torque rather than just i_q alone. The machine torque oscillations are also considerably smaller and stabilize in the steady state faster for the FLC based scheme. Fewer fluctuations in the torque curve contribute to higher drive stability and faster transients, which is a substantial advantage.

The next Figs. 4.9 (a)-(d) show the rotor speed response, command phase current i_a, quadrature (q) -axis command current and actual phase current i_a, for the simplified FLC based IM drive with fuzzy slip gain tuning at half load and rated speed conditions. Over again, the drive follows the command speed with no overshoot or undershoot and any demonstrable steady state error. The distinguishable change this time is the short transient rippling of rotor speed at start up. Again, i_q is of lower magnitude than for the PI/i_d=0 -based drive (approximately 30% lower), and obviously, the time of start up is shorter. Current

i_d is observed to be approximately the same magnitude of i_q. The actual phase currents are closing in shape, although not purely sinusoidal.

Similar responses of the FLC based IM drive with fuzzy slip gain tuning at full load and rated speed conditions are shown in Figs. 4.10 (a)-(d). The maximum phase current magnitude does not change in computer simulations, and is kept under these conditions at approximately 0.9 Amperes (compare with the PI controller data of 2.3 Amperes steady state current at no load and reference speed conditions, and 3.5 Amperes at full load and reference speed conditions). The only distinguishable change is the increased time of start up current. The start up current changes from 0.34 sec. at no load to 0.43 sec. at full load. This is 35% better time if measured up to the PI controller used for comparison purposes, which showed the result of 0.53 sec. of start up time at no load conditions and more than 0.62 sec. start up time at the full load and reference speed conditions. It has to be mentioned that despite the better start up times and significantly lower steady state currents, the simplified FLC -based drive computer simulations showed that the actual start up currents are in the region of 20% higher than those for the PI –based controller. For assessment, the PI -based/i_d=0 scheme controller required a maximum stator current magnitude of 3.5 Amperes to follow the command speed at this load, which is the rated steady state current. Once again, it is shown that the FLC -based drive with fuzzy slip gain tuning can follow the command speed smoothly. The transient overshoot is virtually eliminated and steady state is achieved even more quickly. The actual phase currents are very close to sinusoidal. Therefore, the FLC -based drive with fuzzy slip gain tuning achieves the steady-state command speed faster than the PI -based/i_d=0 for an induction motor drive, with slightly higher peak phase start up currents, and considerably lower steady state current in all cases of load and command speeds. These computer simulation results confirm that the

FLC -based IM drive with slip gain tuning can quickly follow the command speed under different loading conditions.

Figure 4.4 (a) - (d) shows the responses of PI -based/i_d=0 scheme controller at no load condition. Figure 4.4 (a) shows the drive rotor speed. It can be seen that at this conditions the rotor speed reaches the reference speed after 0.5 sec. with small overshoot, then at 0.75 sec the reference speed drops from 188.5 rad/sec to 150 rad/sec, the actual rotor speed follows it with little undershoot and settles at time t = 1.4 sec. When the reference speed changes to original value of 188.5 rad/sec, the rotor speed follows it almost with no overshoot. Computer simulations show that the command phase current as well as the actual phase current in phase "a" corresponds to the rotor speed graph pattern, with actual stator phase current reaching 2.4 Amperes at the steady state. Figures 4.5 (a) - (d) and 4.6 (a) - (d) show the simulated results of a PI –based/i_d=0 scheme at half load and full load conditions employing the same pattern of reference speed change. As the load increases it is clearly seen that the overshoot with the reference speed increase and undershoot with speed decrease become larger. Transient times increase as well, correspondingly. The steady state actual phase current reaches 3.5 Amperes, which is the rated value for this induction motor drive at full load.

The responses of the simplified FLC based IM drive with fuzzy slip gain tuning, including speed, "a" -phase actual current, quadrature -axis command current and command phase current to a step change of command speed at no load conditions are shown in Fig. 4.11 (a) - (d), respectfully. The results undoubtedly show that the drive can follow the command reference speed quickly and accurately during a step decrease (from 188.5rad/sec to 150rad/sec) and step increase (from 150 rad/sec to 188.5 rad/sec), there are no visual overshoots or undershoots. The transient times are considerably lower than those of the PI controller. Again, the

actual phase current shows the above-mentioned features. At start up, the current is about 20% higher than that of PI controller, but at the steady state condition, computer simulations show that the phase current is way lower, just below 1 Ampere.

Figure 4.12 (a) - (d) show the speed response, command phase current, quadrature (q) -axis command current and actual phase current i_a, respectively, for the FLC -based IM drive with fuzzy slip gain tuning block to a step change of command speed at half load conditions. Figs 4.13 (a)-(d) show the same main responses to a step change of command speed at full load conditions. Once more, the above-mentioned figures show that the drive rapidly and follows the command reference speed. The transition time of start up and speed change is increased slightly, but there is still no overshoot or undershoot of speed, even under the full load conditions. One can also observe that the actual phase current follows command current, and the actual machine current increases at low speed especially under full load conditions. The results show that the simplified FLC based IM drive can follow the command speed under a step change of command speed and at different loading conditions.

Another important feature of the high performance control system is the ability of the drive to withstand disturbances inside the system, such as inductance or resistance change, for instance, or external disturbances. Change of load is a typical external system disturbance and is a very common occurrence for high performance drives in the industry. The speed and corresponding command reference phase current of the motor, quadrature (q) -axis command current and actual "a" -phase current responses for the FLC -based IM drive are shown in Figs. 4.13 (a)-(d), respectively, for a sudden increase of load at rated speed (188.5 rad./sec.). The motor is started with no load and at t = 1 sec the full load is

suddenly applied. It is observed that drive speed is reduced gradually, and only very slightly, and that the rotor speed runs lower, less than 1% of command speed and is recovered exceptionally quickly, under this comparatively sudden increase of load. Computer simulations also show that the actual current response is almost sinusoidal and follows command current. When compared to the PI –based controller, the resulting graphs for similar computer simulations are shown on Fig. 4.4 (a) - (d), Fig. 4.5 (a) - (d) and Fig. 4.6 (a) - (d) featuring the step command speed change with no load, half load and full load conditions, correspondingly. It can be observed that PI controller drive has higher transient time, overshoots and undershoots or speed, and much higher steady state actual phase current. The recovery of rotor speed when load is taken off is slow and gradual, with long transient.

Induction machine parameters such as resistance and/or inductances can be affected significantly by temperature change, saturation or other factors. Therefore, one of the most important criteria of the high performance control system for the IM drive is the ability to tolerate sudden or gradual motor parameter variations. The gradual change of stator resistance with temperature is a widespread phenomenon in this type of machines. The computer simulations of the simplified FLC based drive for doubled stator resistance at no load and rated speed conditions are given in Figs. 4.14 (a) and (b), respectively, where the speed and corresponding actual phase current responses are shown. The simulated responses for full load conditions are shown in Figs. 4.15 (a) and (b). In both cases it can be clearly seen that the drive follows the command with no visible speed disturbances even after a sudden change of armature resistance (resistance value is doubled).

Furthermore, the inertia of the electric motor may change at different loading conditions, so it is important to look into the response of the drive to inertia

variations. Figs. 4.16 (a) and (b) show computer simulations of speed and corresponding actual "a" -phase current responses, respectively, to a sudden doubling of machine rotor inertia under no load conditions at rated speed. Figs. 4.17 (a) and (b) show similar responses under the full load conditions. These results point out that the simplified FLC -based drive follows the rated command speed accurately and smoothly even when subjected to a sudden doubling of rotor inertia under different loading conditions. There is no visible effect on the rotor speed or actual machine phase current whatsoever.

Also, in high performance industrial applications changes in direct (d) and quadrature (q) -axis reactance parameters of the IM during operation can significantly affect drive functioning. This can be easily understood as one of the forms of the torque equation (2.35) for the induction machine contains both the direct (d) and quadrature (q) -axis inductance (or flux) parameters. Therefore, high performance drives must be able to respond quickly and accurately to variations in L_q and L_d. Figs. 4.18 (a) and (b) show the speed and corresponding actual "a" -phase current responses, respectively, to a sudden 50% decrease in L_q under no load and at rated speed. Figs. 4.19 (a) and (b) show similar responses under full load conditions. These results indicate that the simplified FLC based drive with fuzzy slip gain tuning follows the rated command speed accurately and precisely even when subjected to a sudden 50% decrease of L_q at the time 1.25 sec., under different loading conditions. It is virtually impossible to detect the point were inductance was abruptly decreased, which shows the great robustness as well as excellent characteristics of the proposed controller.

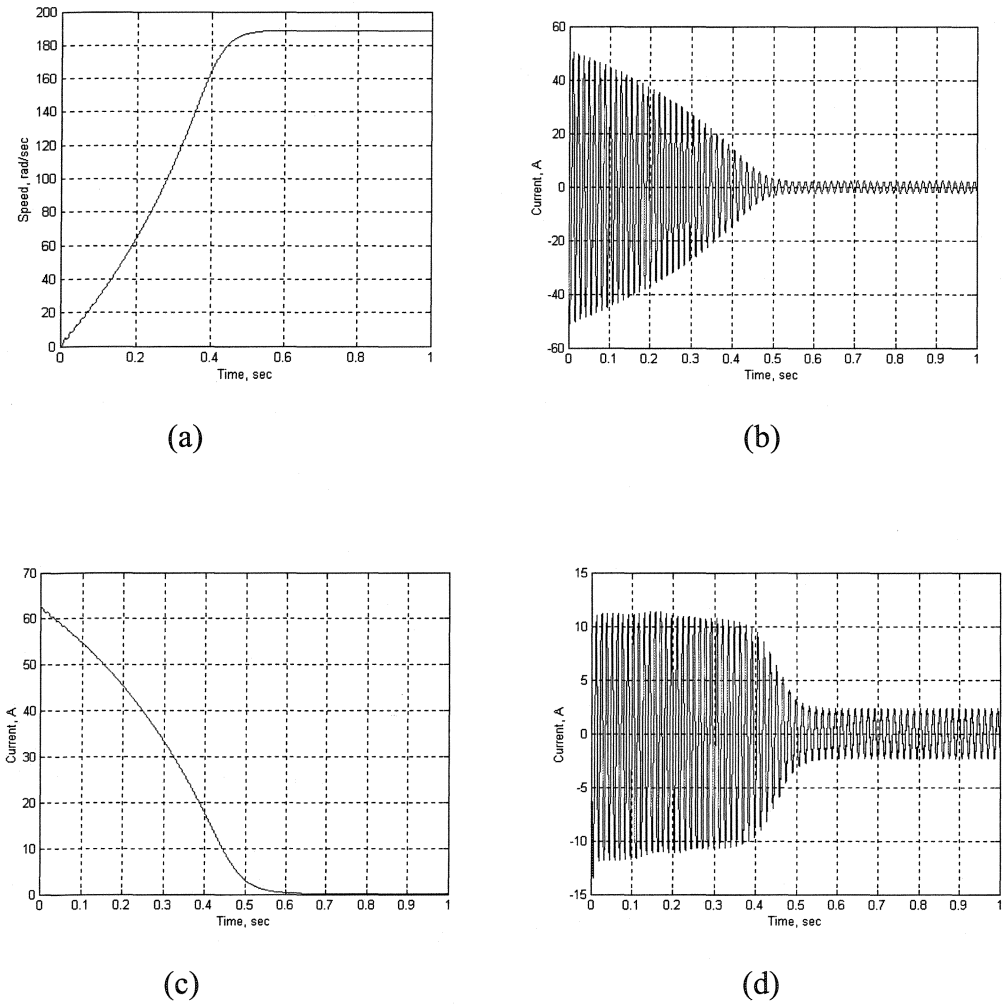


Fig. 4.1. Simulated responses of the PI based/i_d=0 IM drive: (a) speed, (b) command phase current, (c) q-axis command current and (d) actual phase current at no load and rated speed conditions.

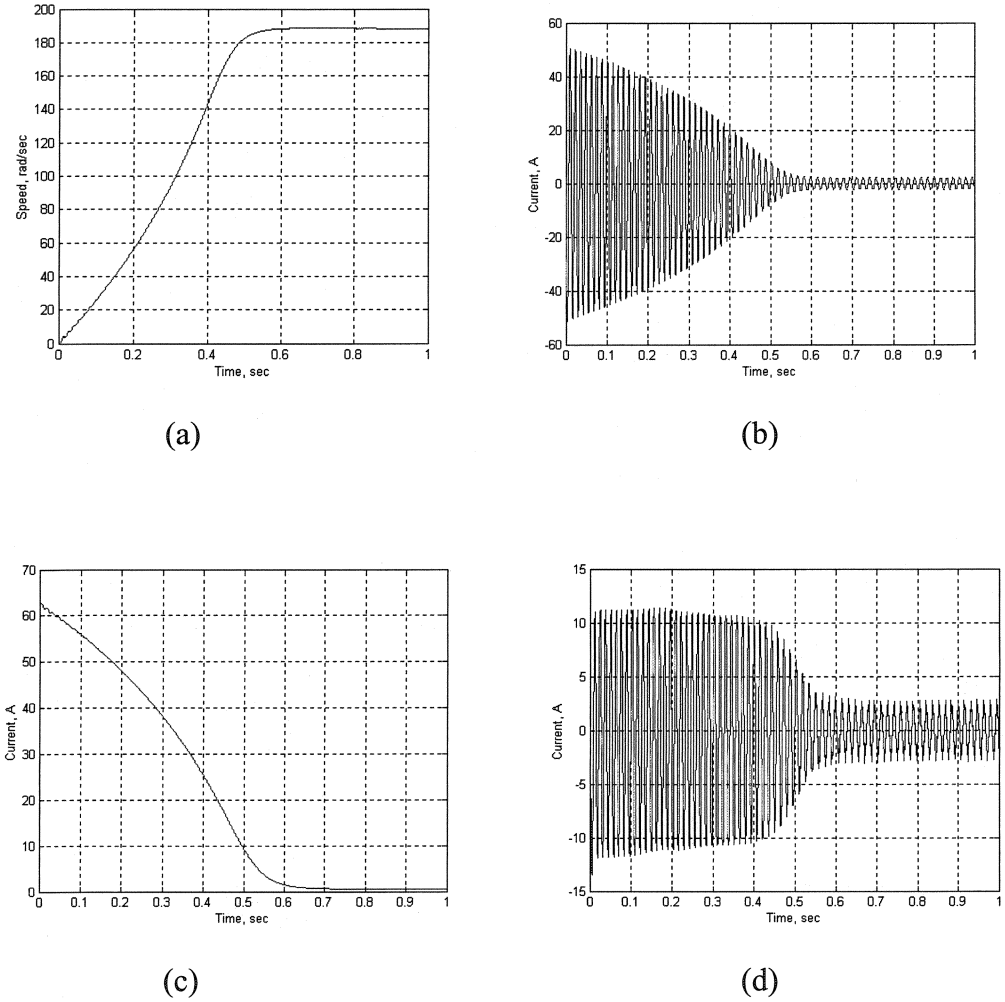


Fig. 4.2. Simulated responses of the PI based/i_d=0 IM drive: a) speed, (b) command phase current, (c) q-axis command current and (d) actual phase current at half load and rated speed conditions.

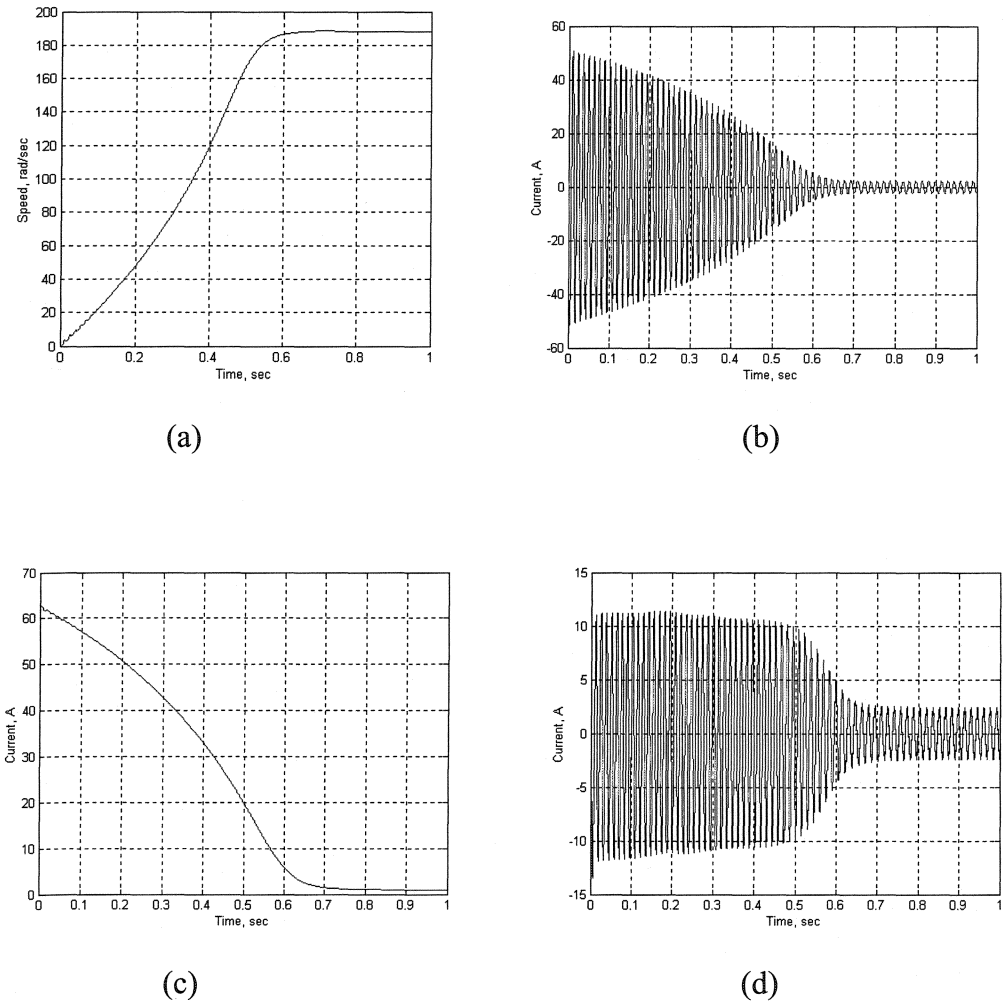


Fig. 4.3. Simulated responses of the PI based/i_d=0 IM drive: a) speed, (b) command phase current, (c) q-axis command current and (d) actual phase current at full load and rated speed conditions.

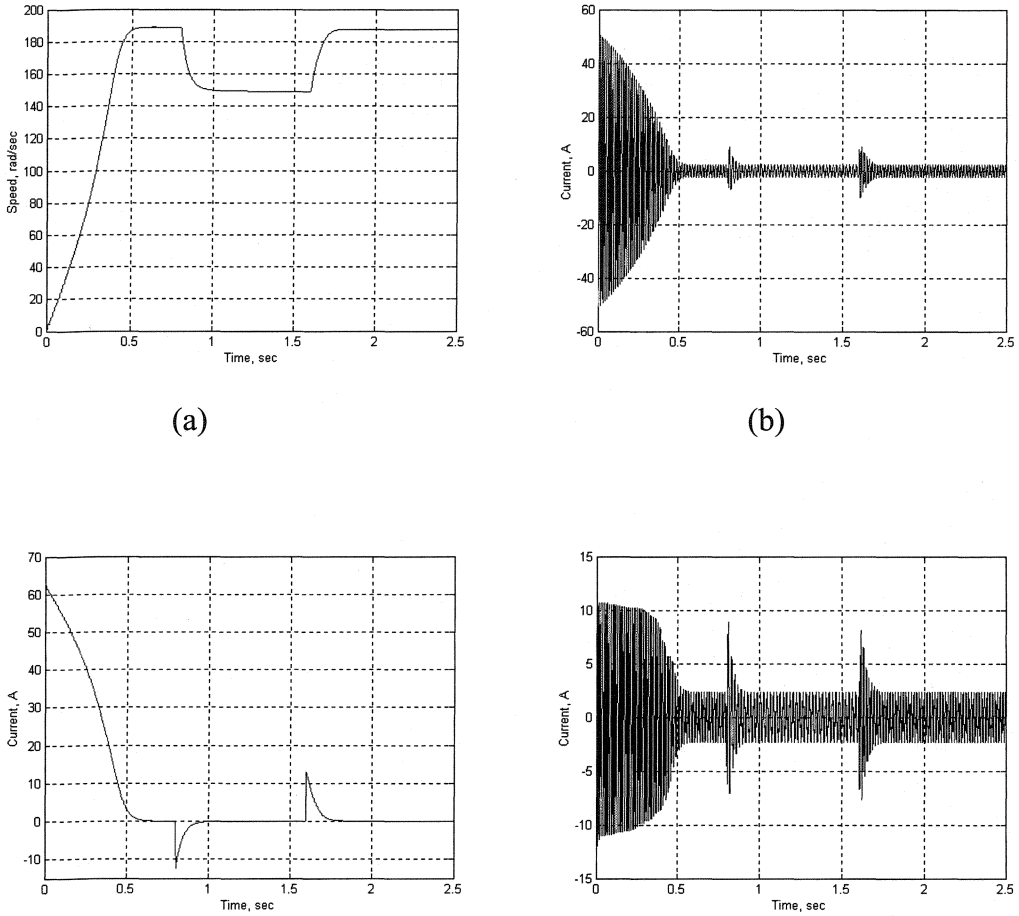


Fig. 4.4. Simulated responses of the PI based/i_d=0 IM drive: a) speed, (b) command phase current, (c) q-axis command current and (d) actual phase current for a step change of speed at no load conditions.

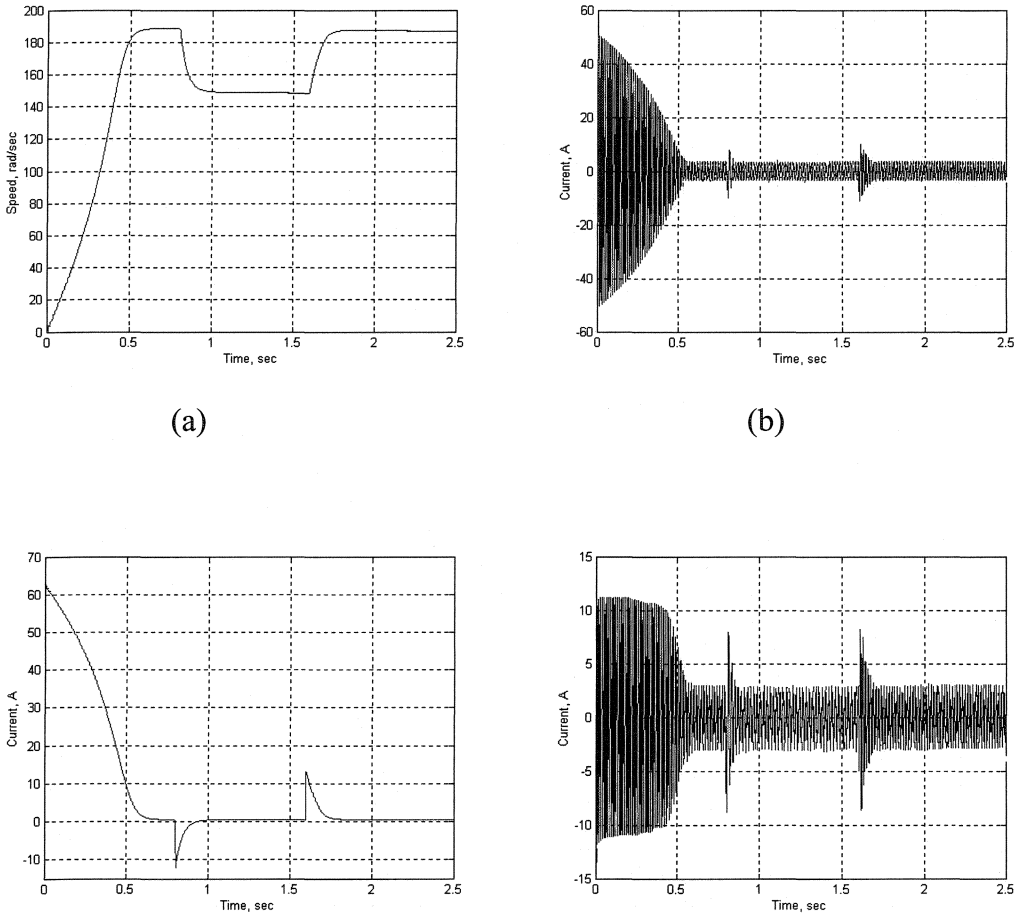


Fig. 4.5. Simulated responses of the PI based/i_d=0 IM drive: a) speed, (b) command phase current, (c) q-axis command current and (d) actual phase current for a step change of speed at half load conditions.

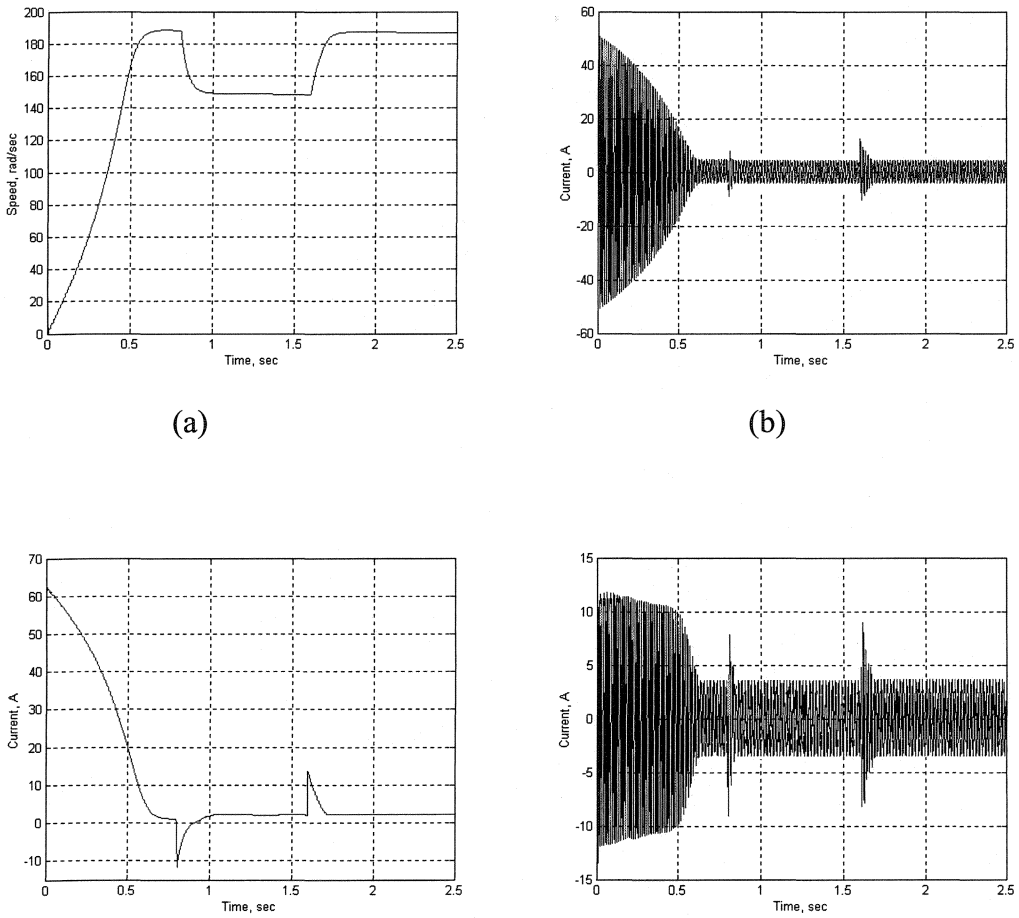


Fig. 4.6. Simulated responses of the PI based/i_d=0 IM drive: a) speed, (b) command phase current, (c) q-axis command current and (d) actual phase current for a step change of speed at full load conditions.

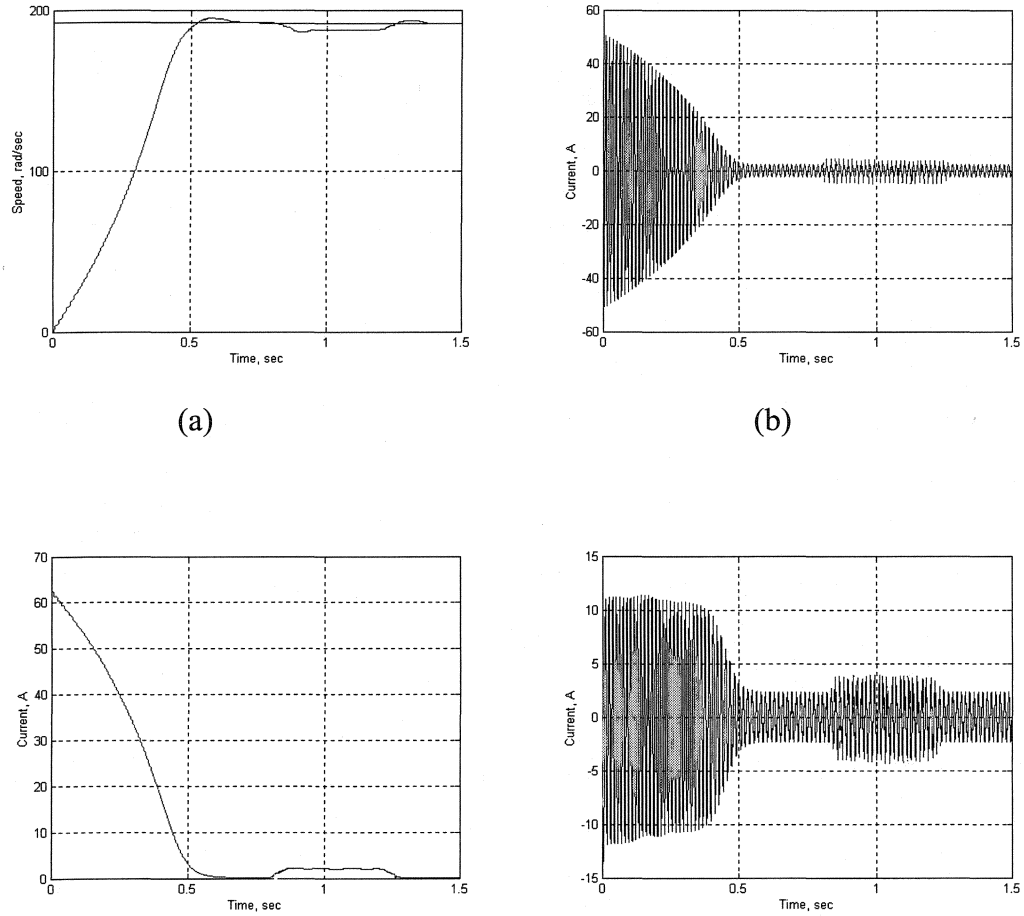


Fig. 4.7. Simulated responses of the PI based/i_d=0 IM drive: a) speed, (b) command phase current, (c) q-axis command current and (d) actual phase current for a sudden change of load (from zero to full load) at rated speed.

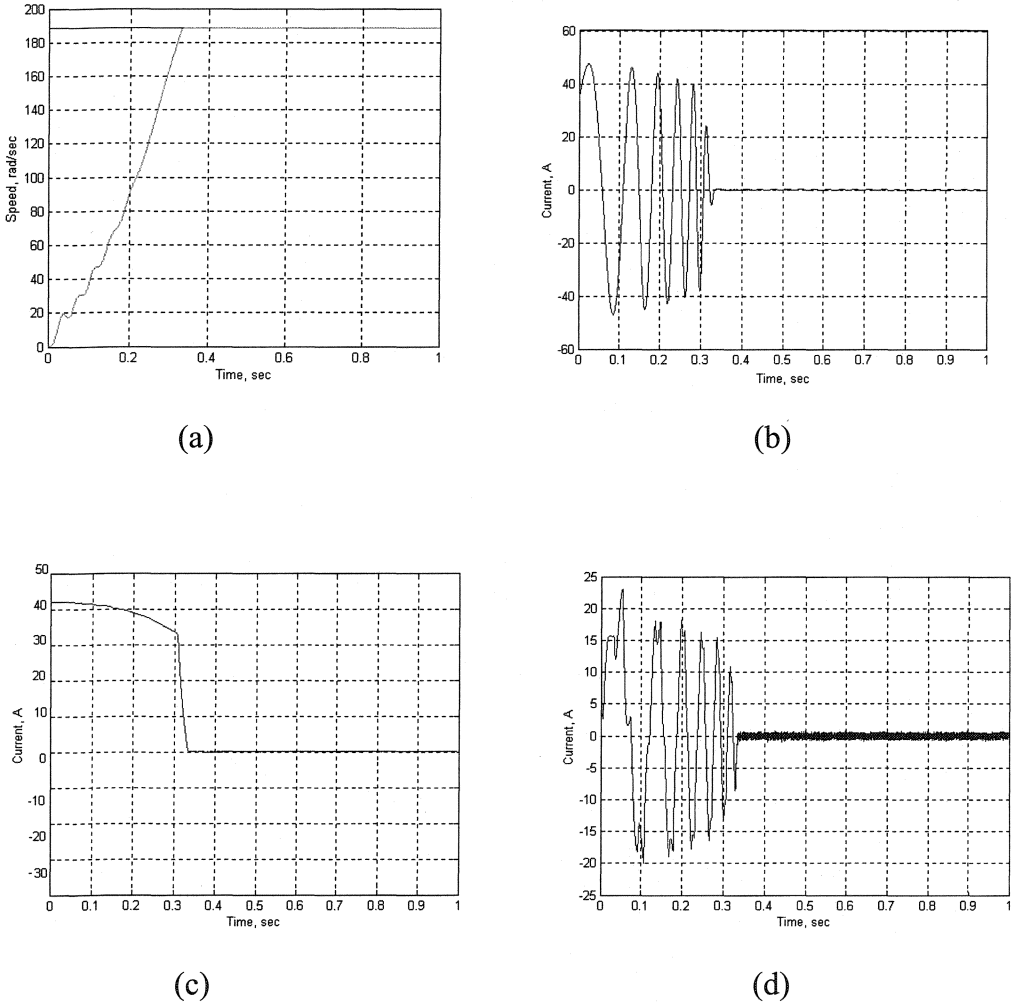


Fig. 4.8. Simulated responses of the simplified FLC -based IM drive: a) speed, (b) command phase current, (c) q-axis command current and (d) actual phase current at no load and rated speed conditions.

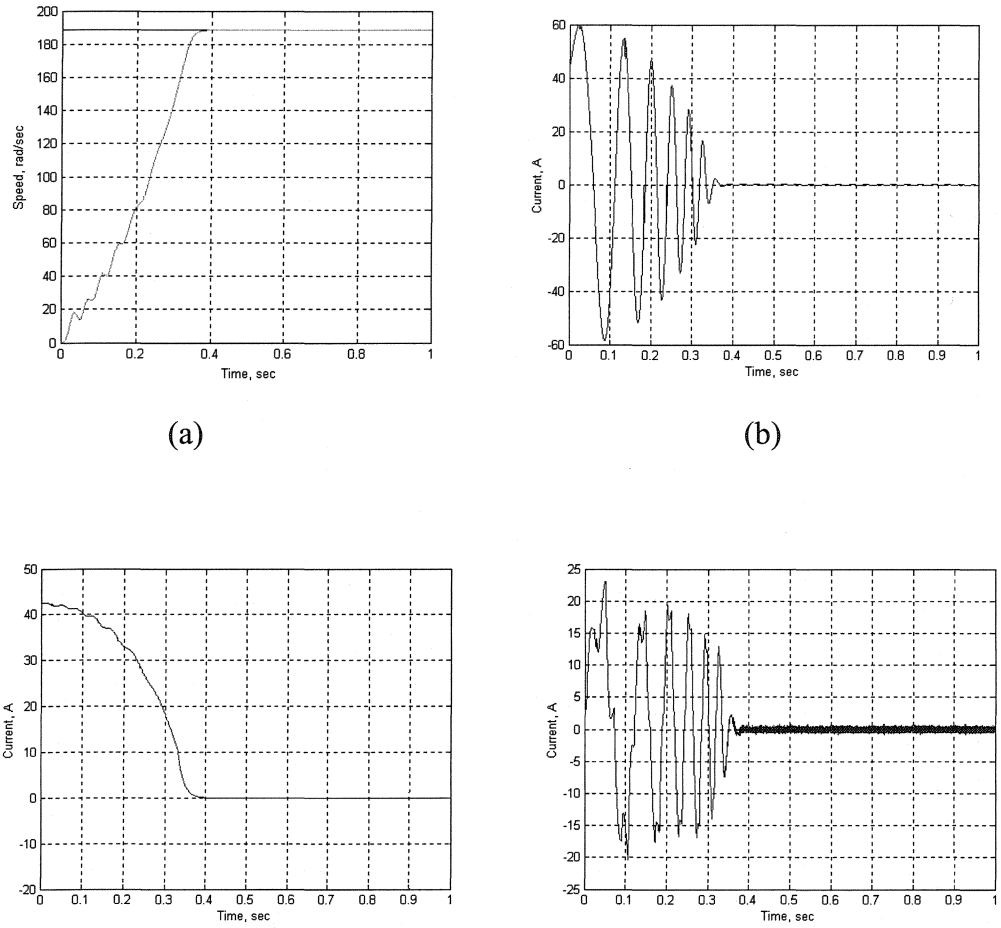


Fig. 4.9. Simulated responses of the FLC -based IM drive: a) speed, (b) steady-state actual phase current, (c) q-axis command currents and (d) actual phase current at half load and rated speed conditions.

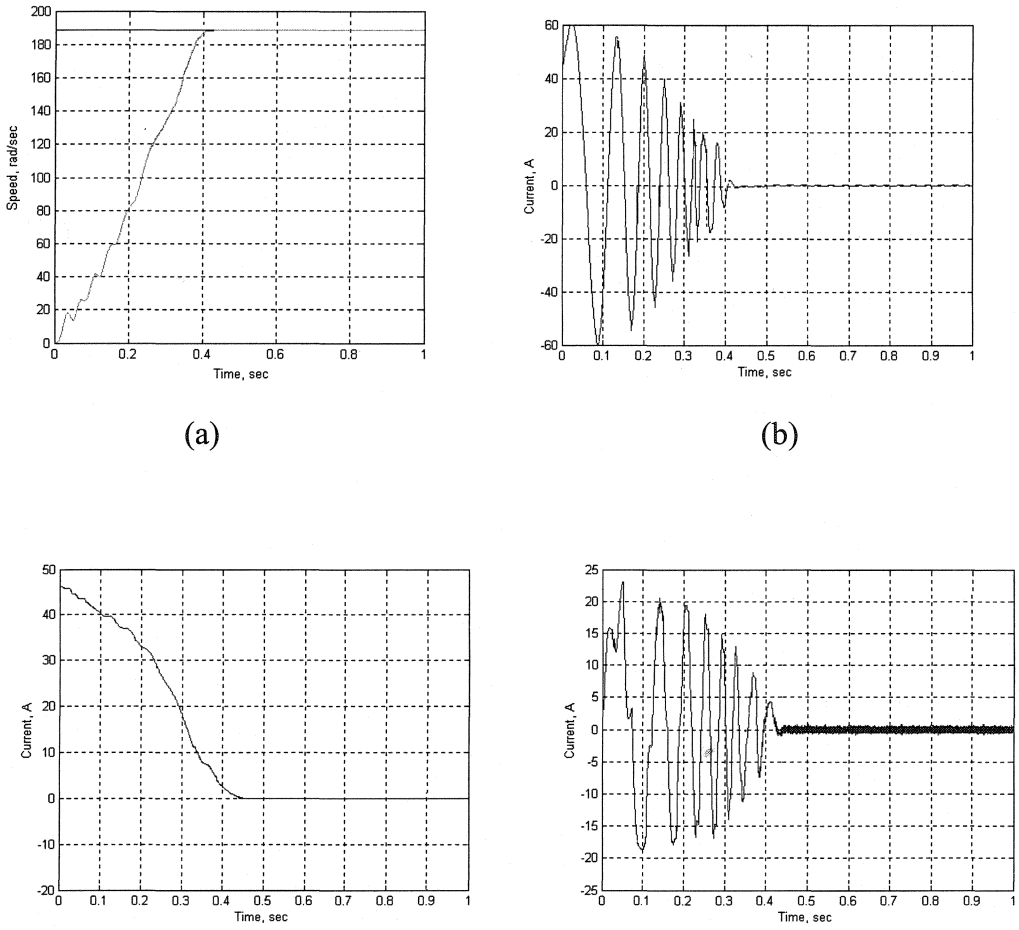


Fig. 4.10. Simulated responses of the FLC -based IM drive: a) speed, (b) steady-state actual phase current, (c) q-axis command current and (d) actual phase current at full load and rated speed conditions.

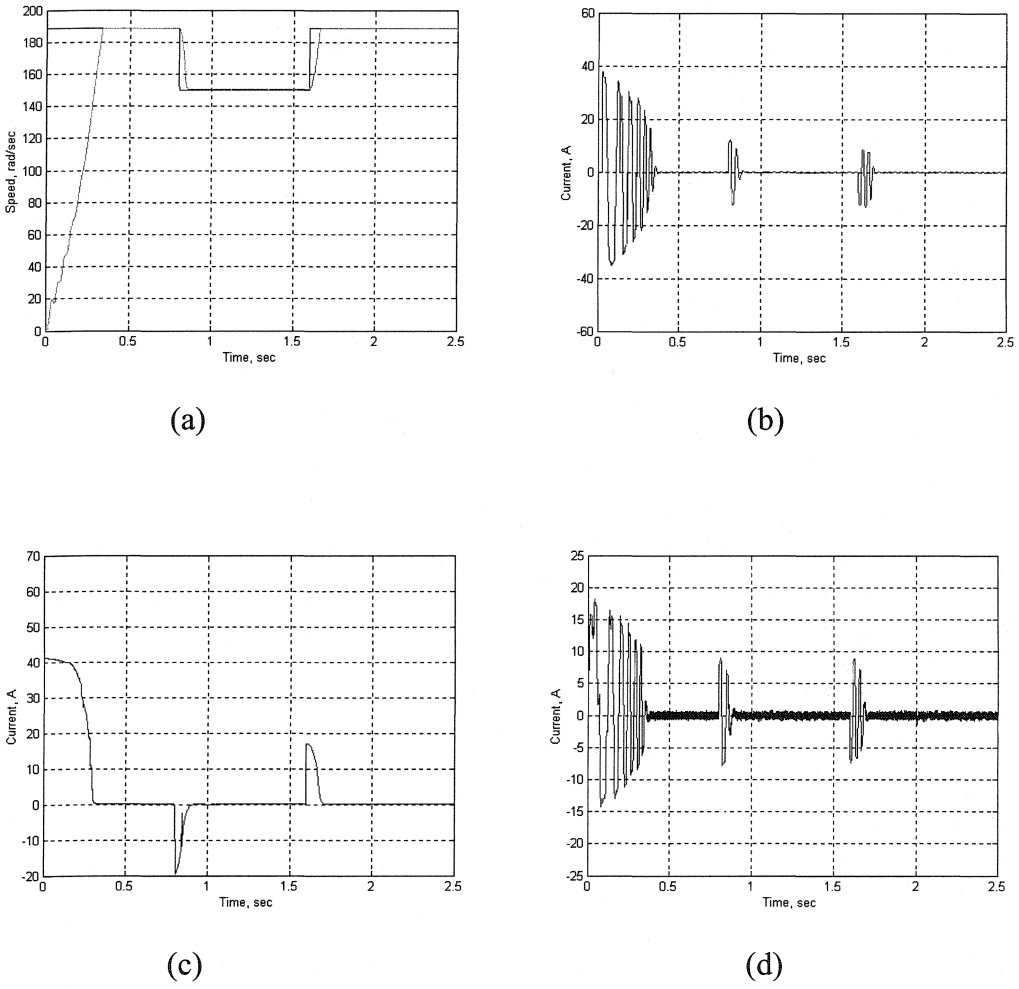


Fig. 4.11. Simulated responses of the simplified FLC -based IM drive: a) speed, (b) steady-state actual phase current, (c) q-axis command currents and (d) actual phase current for a step change of speed at no load conditions.

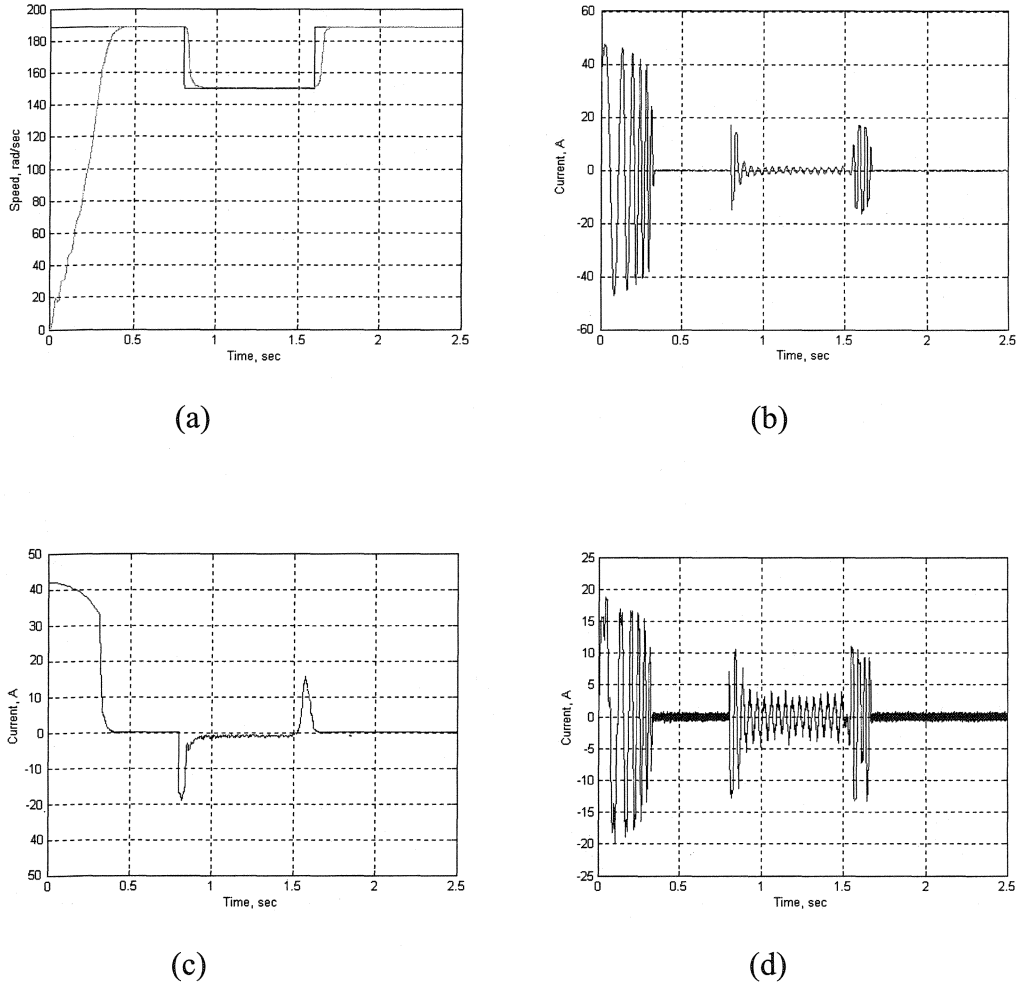


Fig. 4.12. Simulated responses of the simplified FLC -based IM drive: a) speed, (b) steady-state actual phase current, (c) q-axis command current and (d) actual phase current for a step change of speed at full load conditions.

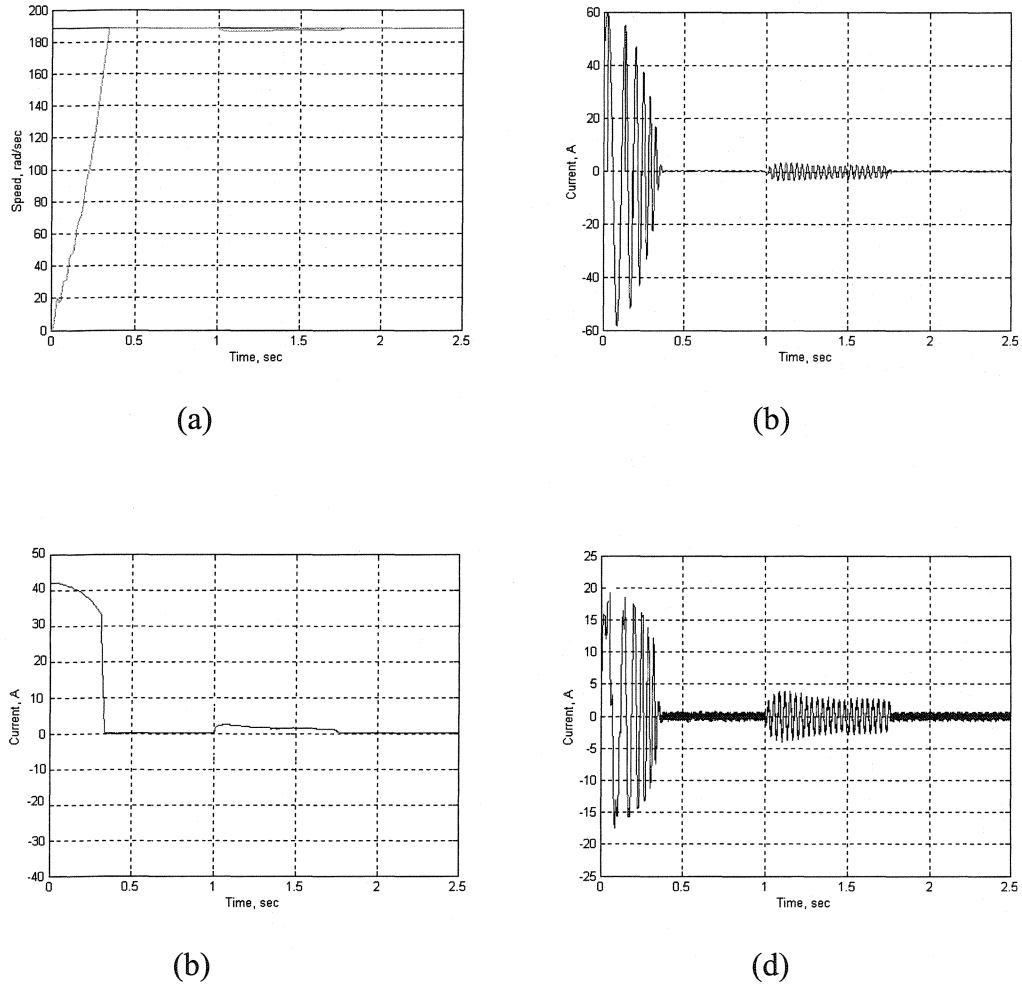
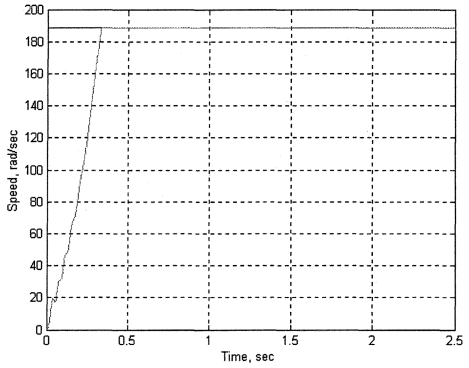
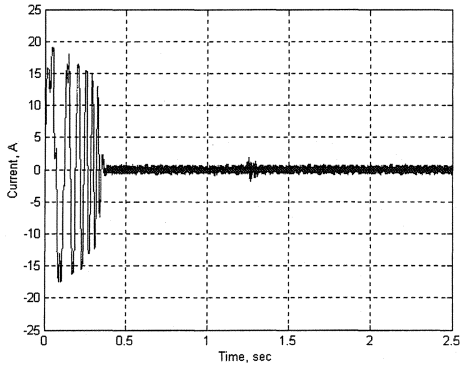


Fig. 4.13. Simulated responses of the FLC –based IM drive: a) speed, (b) command phase current, (c) q-axis command current and (d) steady-state actual phase current for a sudden change of load (from zero to full load) at rated speed.

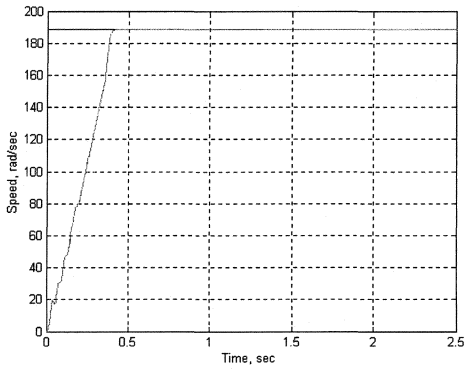


(a)

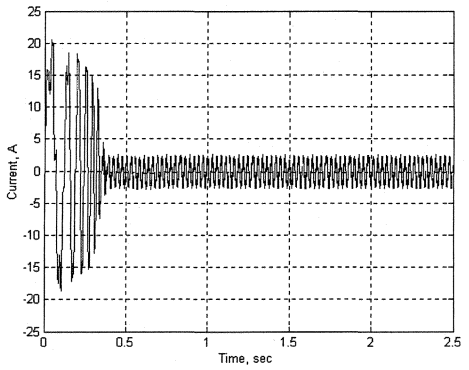


(b)

Fig. 4.14. Simulated responses of the FLC -based IM drive: a) speed, (b) steady-state actual phase current i_a for a sudden change of stator resistance (R to 2R) at no load and rated speed conditions.

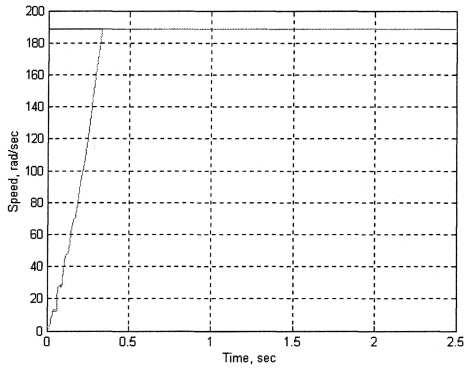


(a)

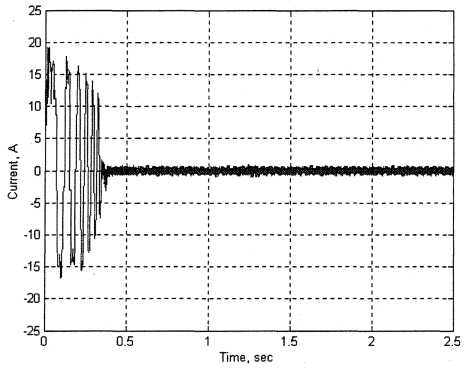


(b)

Fig. 4.15. Simulated responses of the FLC -based IM drive: a) speed, (b) steady-state actual phase current i_a for a sudden change of stator resistance (R to 2R) at full load and rated speed conditions.

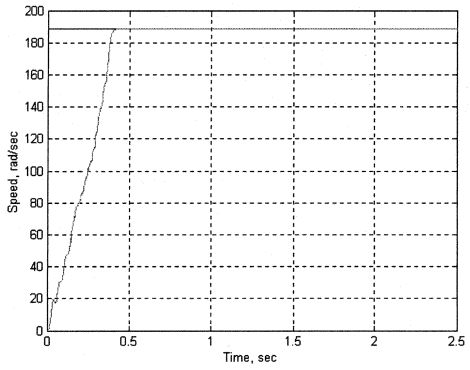


(a)

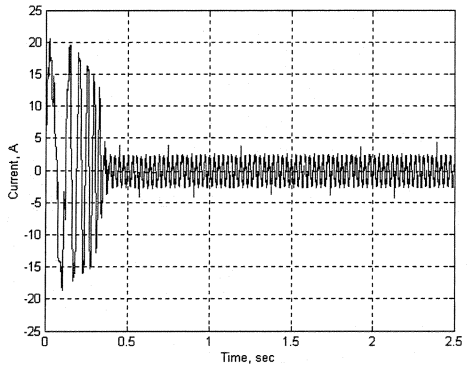


(b)

Fig. 4.16. Simulated responses of the FLC -based IM drive: a) speed, (b) steady-state actual phase current i_a for a sudden change of rotor inertia (J to 2J) at no load and rated speed conditions.

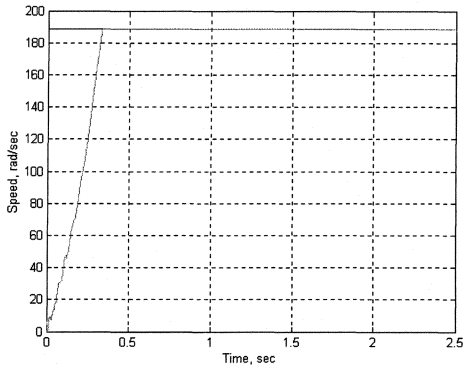


(a)

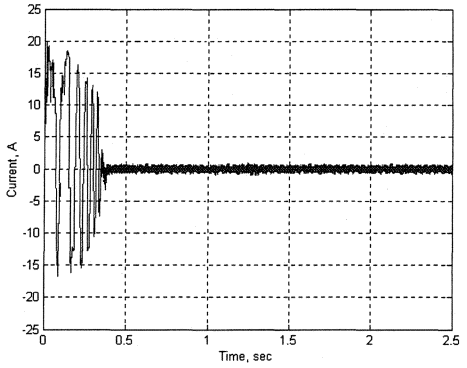


(b)

Fig. 4.17. Simulated responses of the simplified FLC -based IM drive: a) speed, (b) steady-state actual phase current i_a for a sudden change of rotor inertia (J to 2J) at full load and rated speed conditions.

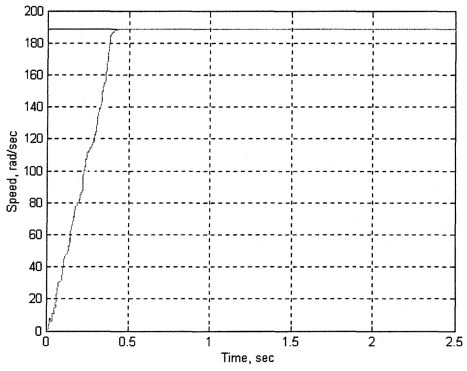


(a)

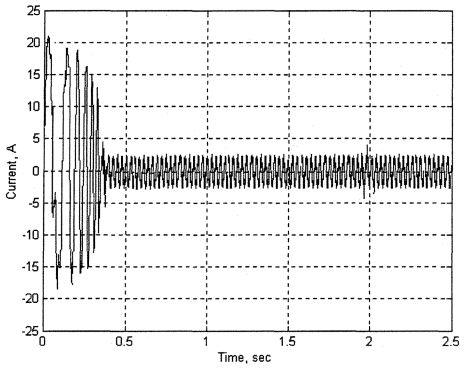


(b)

Fig. 4.18. Simulated responses of the simplified FLC -based IM drive: a) speed, (b) steady-state actual phase current i_a for a sudden 50% decrease of L_q at no load and rated speed conditions.



(a)



(b)

Fig. 4.19. Simulated responses of the simplified FLC -based IM drive: a) speed, (b) steady-state actual phase current i_a for a sudden 50% decrease of L_q at full load and rated speed conditions.

4.3 Concluding Remarks

Computer simulations of the proposed indirect vector control scheme of the IM drive incorporating the proposed simplified FLC have been presented in this chapter. The performance of the proposed drive has been found to be better than the same of the conventional PI controller. The FLC is found to be a suitable controller and accommodate itself to different operating conditions such as step change of command speed, load change and parameter variations, which have been proven by experiments. Therefore, the simplified FLC with i_d not equal zero for an induction machine can be an alternative for the conventional fixed gain PI or PID controllers or for more complex fuzzy logic schemes, incorporating the greater number of fuzzy rules and variables membership functions.

Despite the greater number of fuzzy rules or/and membership functions that smoothen the performance of the drive system and generally increase the preciseness of controller, these factors contribute greatly to the computational burden, on the other hand limiting and slowing down system performance. By careful simplification of the fuzzy inference engine employed in the presented control system blocks, high agility can be obtained together with minimizing of computational processes at the same time. The presented computer simulation results prove the high performance and robustness of the simplified FLC controller in an IM drive systems.

The proposed FLC scheme with i_d not equal to zero draws about 25% less current compared to the PI –based drive with $i_d = 0$, no command current oscillations were observed. This is because with i_d not equal zero model both q and d – axis components contribute to the command current rather than just i_q alone, which is a substantial advantage.

To confirm the efficiency of the proposed controller a laboratory practical realization of the complete indirect vector control scheme for the induction motor drive has been carried out in real-time as part of this work. Detailed procedures, experimental stages and results are described in the following chapter.

Chapter 5

Experimental Implementation of the FLC -based Vector Control of the IM Drive System

5.1 General Introduction

The major trends, successes and difficulties in implementation of high performance drive control systems based on fuzzy logic are reviewed earlier in chapter 1. One of the main disadvantages of fuzzy logic control is the high computational burden imposed on both hardware and software of the drive system. Because of that, look-up tables for fuzzy logic were often used previously, which limited the accuracy of the controller response. Also, because of both high-level hardware and software constraints, controllers of this type have limited use in expensive or scientific experimental devises. Apart from design of the novice fuzzy logic controller layout, significant attention has been given to subsequent simplification of the controller logic with the expectation of cutting on the computational burden, increasing the robustness, limiting the response time, etc. In this work, the fuzzy logic calculations are made on-line with the highest possible

sampling frequency. No look-up tables have been used. A practical real-time implementation of the FLC for induction motor drives has been presented. Solution to the practical implementation would open the new horizons to implementation of high precision and robust FLC systems for induction motor drives in industry. For that reason, the implementation problems discussed previously are overcome in this work by making the most of a simplified FLC along with an optional fuzzy logic block responsible for the slip gain-tuning scheme as part of the controller.

The complete simplified indirect vector FLC for IM drive with fuzzy slip gain tuning block is experimentally implemented and the results are presented in this chapter. The proposed real-time implementation technique gives better results than those of the conventional PI controller used for comparison purposes. The proposed drive has been implemented by the use of a PC connected to a current controlled voltage source inverter through the dSPACE DS1102 digital-signal-processor (DSP) controller board in a standard 1 hp laboratory induction machine. The detailed implementation of hardware and software programming as well as the various laboratory experimental results under different dynamic operating conditions are presented and discussed in this chapter in order to confirm the effectiveness of the proposed drive system.

5.2 Description of Experimental Set Up

5.2.1 Hardware Realization

The experimental hardware schematic used for the real-time implementation of the proposed IM drive system is shown in Fig. 5.1. A personal computer (PC)

with uninterrupted communication capabilities through dual-port memory holds a DSP board installed in it. The DSP board of DS1102 modification is based on a Texas Instrument (TI) TMS320C31 32-bit floating point digital signal processor, as a main computational device. There are also a set of on-board peripherals used in digital control systems supplemented to DSP board, such as analog to digital (A/D), digital to analog (D/A) converters, incremental encoder interfaces and other interfaces for peripherals devices connection. The DS1102 board has one 4 channel (2 12-bit and 2 16-bit each) A/D, one 4 channel (12-bit each) D/A converter and two 16-bit incremental encoders, used for various purposes. The DSP DS1102 board is also equipped with a supplementary TI-TMS320P14 16 bit micro controller, that acts as a slave processor and provides the necessary digital input/output (I/O) capabilities (in this work, the slave processor was used only for digital I/O configuration) and powerful timer functions, such as an input capture, output capture and the possibility of PWM waveform generation.

The actual motor armature currents were measured by the use of Hall-effect sensors, which have good frequency response. The received signals were fed to the DSP board through the corresponding A/D converter. Because the motor neutral not grounded scheme layout was used, only two phase currents are needed to be fed back, and therefore the third phase current is calculated from them. An absolute incremental encoder was used for the purpose of the rotor position angle measurement. In the laboratory experimental set-up, it was mounted on the rotor shaft by means of a flexible coupler, and fed to the DSP board through the encoder interface. The encoder generates 4096 pulses per revolution, which are fed to the incremental encoder interface of the board. By using a 4-fold pulse multiplication the output of the encoder is increased to as much as 4x4096 pulses per revolution in order to get a batter resolution. A common 24-bit position counter is used to count the encoder pulses. The counter is reset once per revolution by the index pulse generated directly from the encoder. The induction motor speed is calculated by

the means of use of measured rotor position angles using the mathematical method of numerical differentiation.

The fixed-band hysteresis controller was used as the current controller in order to implement the indirect vector control algorithm for induction motor drive. The speed controller is used to generate the command currents for the hysteresis current controller.

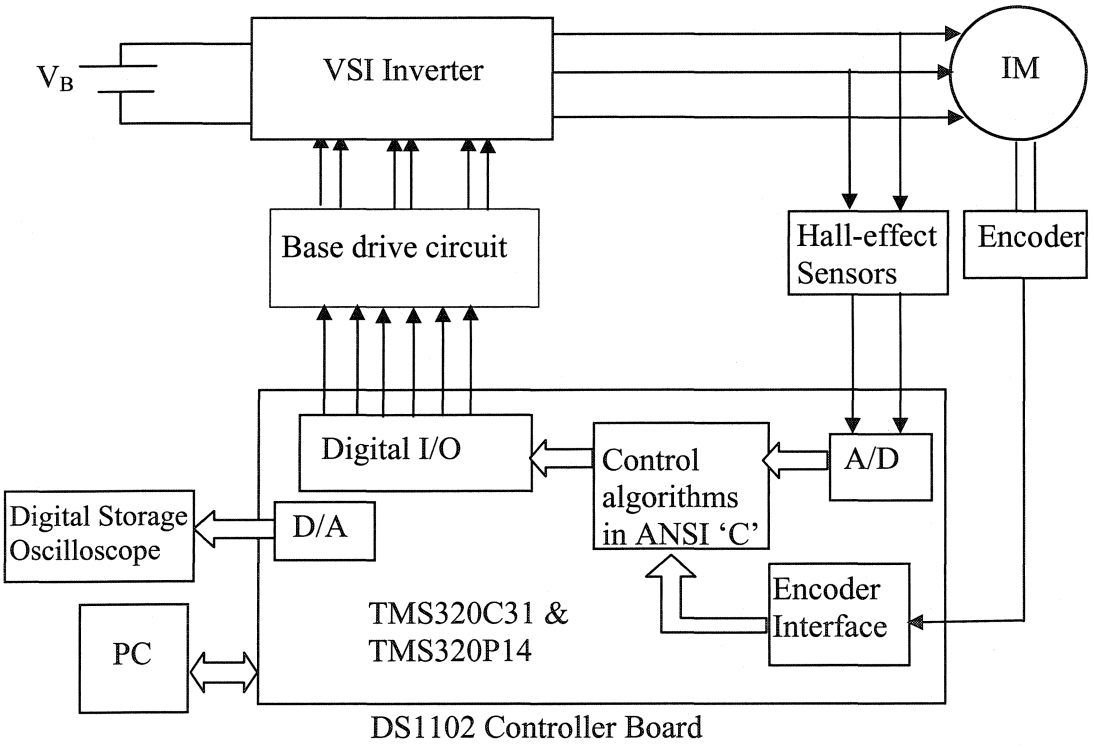


Fig. 5.1. Laboratory experimental hardware realization scheme for the FLC controlled IM drive system.

The command currents are compared to the corresponding actual motor currents in the hysteresis current controller and the appropriate logic signals are

generated. This signals act as firing pulses for the IGBT inverter switches. As a result, these six PWM logic signals are produced by the DSP board and from its outputs. They are then fed to a base drive circuit (BDC) for the insulated gate bipolar transistor (IGBT) or bipolar junction transistor (JCT) inverter power module. The control algorithms for the fuzzy logic speed controller and the hysteresis current controller are written in the high-level ANSI "C" programming language and implemented using the DS 1102 controller board. The program code also secures some input/output (I/O) functions for access and initialization of the external devises of the DSP board. For the given reasons, it can be said that the designed system is a fully digital system.

The base drive circuit is also used to provide an electrical isolation between the high power voltage supply and the low power logic micro controller circuits. It is also used to correct if required the voltage level so that it is sufficient to drive the inverter. The dc power supply for the inverter was obtained by rectifying the ac power through a specific rectifying unit.

For the purposes of experiments, in order to provide loading, the laboratory IM was coupled to a dynamometer via a belt. The dynamometer works as a mechanical load imposed on the motor. To capture the desired signals coming out through the D/A port of the DSP board, a digital storage oscilloscope was used. Also, using specialized grabber software, oscilloscope images were converted into data readable by Matlab software and images that were then used in the following part of the chapter. The experimental laboratory setup is shown in appendix 3, Fig. D.1.

5.2.2 Software Realization

Instead of the simpler method of ANSI "C" programming code generation by means of specialized Matlab software real time workshop (RTW), it was decided to develop a software program for the corresponding blocks of fuzzy logic

controller by hand, using the characteristic equations and flow charts. The developed software takes into account the simplified nature of FLC, the fuzzy slip gain tuning block, as well as the fixed-band hysteresis current controller for the IM drive in the ANSI "C" programming language. Because of the reduced computational burden of the simplified FLC scheme the sampling frequency used in experiments was 10 kHz, which is adequate for successful real-time implementation of the drive. The detailed software implementation is described below with the use of code parts and laboratory experimental results (full code is given in Appendix F at the end).

In order to access the on-board peripherals all the peripherals of the DSP board have to be initialized at the start. There are a number of macro functions that can be used, supplied with the DS1102 controller board. For example, the macro function *init()* initializes the D/A converter subsystem of the DS1102 for output. The other feature of this command is that it also resets the interrupt request bits and calibrates the 16-bit A/D converters. To initialize the DSP controller board TMS320C31's on-chip timers in order to generate timer interrupts at a predefined sampling rate as given in Table 6.1, the corresponding functions *timer0* and *timer1* are used. This is necessary for real-time experimental implementation of the IM drive control system.

Next, the I/O functions for the A/D converters and 16 bit I/O ports are initialized in the program [87, Appendix F].

As stated earlier, the highest possible 10 kHz sampling frequency was used throughout the entire experiments. The corresponding to these frequency intervals with which an interrupt service routine (ISR) is used to read the actual motor currents and rotor position angles is 100 μsec.

Table 6.1. Part of the code responsible for the DSP board peripheral initialization

```
void main()
{
 init(); /*init DAC mode, calibrate ADCs */
 init_slave_DSP_digital_i_o();
      /* initialize i/o ports for output*/
 *error=NO_ERROR;
     /* initialize overload error flag */
 dp_mem[0].f=0.0;
    /* init 1st dp-mem loc for float */
 dp_mem[1].f=0.0;
    /* init 2nd dp_mem loc for float */
 ds1102_inc_clear_counter(1);
    /* clear incr. encoder counter */
 start_isr_t0(TS);
    /* initialize sampling clock timer  */
 while(*error==NO_ERROR);
     /* background process  */
}
```

The details of initializing all the variables and constants used in the program are shown in Table 6.2 in the step-by-step manner.

Table 6.2. Part of the code responsible for reading the rotor position and motor currents.

```
void  isr_t0()
 { begin_isr_t0(*error);
      .
                .

   ds1102_ad_start();
   i_a = Ka*ds1102_ad(3);
   i_b = Kb*ds1102_ad(4);
   i_c = -(i_a+i_b);
   θr = INCG*ds1102_inc(1 or 2);
      .
      .
      .
 }
```

The DSP board DS 1102's A/D channels 3 and 4 are used to get the actual motor currents by means of the constants K_a and K_b, given in the Table 6.2. The actual value of these constants used for calculations depends on the type of Hall-effect sensors used in experimental set-up. In this work, the constants K_a, K_b has been set to be 11.49 and 10.93, respectively.

To get the actual rotor position corresponding to the incremental encoder value the incremental gain (INCG) constant is used. The INCG constant have been determined to be 13176.79306. To calculate the rotor speed the program is using the method of differentiating the present and past samples of the rotor position. According to the controller schematic, the command torque is generated by the first fuzzy logic block of the speed controller employing the speed error, which is an error between the reference speed and the present sample of actual speed. Therefore, the actual inputs to the first simplified fuzzy logic block of the controller are the speed error and change of speed error. There are three components of the simplified FLC, which are: (1) fuzzification, (2) fuzzy inference engine (rule base) and (3) defuzzification. The corresponding parts of the code in ANSI "C" language used for implementations of these components in real-time are given below.

Fuzzification is the process of conversion of a numerical crisp value into linguistic fuzzy variable. Before the input variables can be used for calculations they have to be normalized by bringing them into the range of operational values. The normalized values of the inputs $\Delta\omega_r$ and Δe are then passed for fuzzification. This stage of calculations uses pre-selected membership functions to convert the crisp values of $\Delta\omega_{rn}$ (cvalue1) and Δe_n (cvalue2) to their corresponding fuzzy values fvalue1 and fvalue3. The trapezoidal and triangular membership functions, which were developed and described in detail in chapter 2, are used to reduce the computational burden during the real-time implementation.

The part of the code written in ANSI "C" high-level programming language responsible for fuzzification is given in Table 6.3. Given code represents a singleton fuzzifier. In this part of the program code, the value NS is set to represent a number of fuzzy sets used in fuzzy inference engine, the corresponding *cvalue* is a crisp value, in particular, *cvalue1* represents the crisp value of the speed error and *clvalue2* represents the crisp value of the change of error. Respectively, *fvalue* is used as a fuzzy value for a particular input. The variable "*i*" denotes the i-th fuzzy set of a particular input, *bl* is the membership function bottom low value, *tl* is the function top low value, *th* is the function top high value, *bh* is the function bottom high value and *cent* is the centroid of a trapezoidal membership function.

Table 6.3: Part of the code responsible for fuzzification process.

```
for(i=0; i<NS1; i++)
  {
   fvalue[i] = 0;
    if ((cvalue1 >= bl_1[i]) && (cvalue1 < tl_1[i]))
     fvalue1[i] = (cvalue1 - bl_1[i])/(t_1[i] - bl_1[i]);
      else
       {
        if ((cvalue1 >= tl_1[i]) && (cvalue1 <= th_1[i]))
          fvalue1[i] = 1;
           else
             {
              if ((cvalue1 > th_1[i]) && (cvalue1 <=
bh_1[i]))
                fvalue1[i] = (cvalue1 – bh_1[i])/(th_1[i] -
b_h[i]);
              }
                   }
    }
   if (cvalue1 < cent_1[0])
        fvalue1[0] = 1;
        else
           {
           if (cvalue1 > cent_1[NS-1])
               fvalue1[NS-1] = 1;              }
```

In order to describe the triangular membership functions used in the fuzzy inference engine, the existing program code for trapezoidal functions should be modified only slightly by setting the *tl* value equal to *th* value (the "low" of the function is assumed to be at its left side, the "high" of the function - at the right side, correspondingly).

The fuzzy inference engine operates with the fuzzy rule base, which is composed of a number of conditional statement control rules ("if-then") that acquire the results of all applicable to the given set of values rules. The simplified FLC presented in this study computes the fuzzy values of the outputs using a limited number of membership functions and rules, as it was detailed in chapter 2. The next level of simplification is the use of only one input for fuzzy output calculations, which means there are no AND or OR statements included into the fuzzy rules. The part of the program code holding a routine for the fuzzy inference engine written in ANSI "C" language is presented Table 6.4. In this table, NS1 (number of fuzzy sets for the speed error, chosen to be equal to 3), NS2 (number of sets for the change of speed error, chosen equal to 2) and NS3 (number of fuzzy sets for the output, chosen to be 3) are the number of fuzzy sets for $\Delta\omega_{rn}$, and i_{qn}^{*}, respectively.

Table 6.4: The program code part for the fuzzy rule-base inference engine evaluation using two inputs.

```
for (i=0; i<NS3; i++)
                fvalue3[i] = 0.0;
    for (i=0; j<NS2; i++)
    {
        for (j=0; j<NS1; j++)
        {
    minval = fvalue2[i];
      if (minval > fvalue1[j]);
    k = RuleBase[i][j];
      if (minval > fvalue3[k]);
    fvalue3[k] = minval;
        }
    }
```

The fuzzy inference engine rule base (Rulebase[i][j]) is the two-dimensional matrix corresponding to the fuzzy file based matrix. The variable k is an integer that corresponds to the output fuzzy set given in the rule base matrix.

The next step of the calculation routine programmed in code is the defuzzification process. The input for the defuzzification procedure is a fuzzy set and the output is a single non-fuzzy (i.e., a crisp value) number. In order to reduce the computational burden for on-line implementation the following mathematical form is obtained using the center of gravity method [88]:

$$i = \frac{\sum_{j=1}^{NS} A_j * \mu_{C_j}(i) * Cent_j}{\sum_{j=1}^{NS} A_j} \tag{5.1}$$

where NS is the number of output fuzzy sets, A_j is the area , $Cent_j$ is the centroid of the jth output set C_j, $\mu_{C_j}(i)$ is the fuzzy value which scales the output set C_j.

In compliance to the formula given above (5.1), a routine for defuzzification has also been written as given below in Table 6.5.

Table 6.5: ANSI "C" code program part for the process of center of area defuzzification.

```
temp1 = 0;
cvalue3 = 0.0;
 for (i=0; i<NS3; i++)
  {
   temp2 = fvalue3[i]*areas_3[i];
   cvalue3 += temp2*cent_3[i];
   temp1 += temp2;
  }
 cvalue3 /= temp1;

TEC = cvalue2*KI/K_T ;
```

In Table 6.5, where variable areas_3[i] is an area, variable cent_3[i] is a centroid of the i-th fuzzy set of the output.

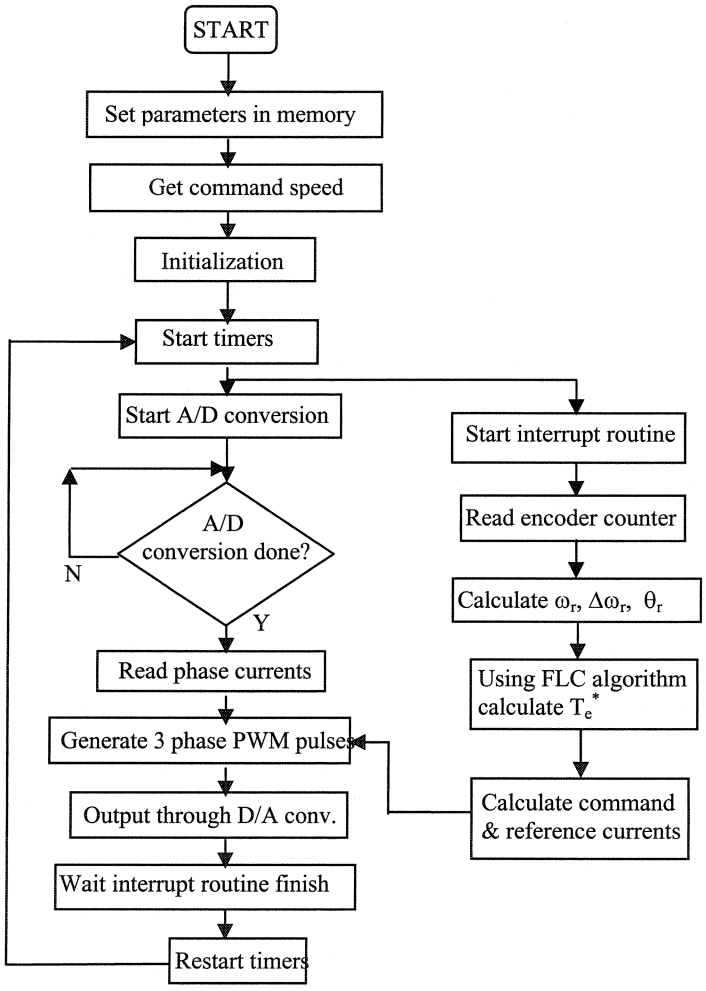


Fig. 5.3. Flow chart of the software for real-time implementation of the FLC based IM drive.

In the next lines, after finishing the fuzzy inference engine calculation procedures, the command currents i_a^*, i_b^* and i_c^* are generated from the command

torque T_e^* and the rotor position angle θ_r using the inverse Park's transformation and two additional fuzzy logic blocks. The program code modules, representing these two blocks are similar to given above example of fuzzy inference engine for the torque signal determination. The module structure of the code easily allows using similar parts of code responsible for fuzzy inference engine in any of the fuzzy logic blocks, according to the scheme layout, for example such as blocks responsible for production of i_q and i_d currents, or fuzzy slip gain tuning block.

The produced command phase currents i_a^*, i_b^* and i_c^* are compared to the actual motor currents in the hysteresis current controller, which provides the necessary driving pulses for the inverter switches. The hysteresis current controller algorithm is also written in the standard ANSI "C" language. The digital I/O ports are configured as output ports for the six PWM logic pulses, which are fed to the base, drive circuits of 3-phase voltage source. For the sake of simplicity, the steps 1-5 are summarized in the flow chart of Fig. 5.3.

The final step is the compilation of the program by the TI ANSI "C" compiler, after which it is downloaded to the DSP board DS 1102 using standard included dSPACE loader program - down1102.

5.3 Experimental Results and Discussion

A number of laboratory experimental tests were made to confirm the previously completed computer simulations of the proposed system to verify its robustness and efficacy. A series of experiments employing a IM drive system operating under conventional PI controller were made to compare and provide

evidence of better control characteristics, that comply with high performance drive requirements. Laboratory experiments were carried out under number of conditions, such as a sudden change of command speed, sudden change of load and also variations of parameters of the drive. They were also made under number of load conditions, ranging from no load condition to full load condition. The experimental results presented and commented below confirm the computer simulations results presented in chapter 4.

The experimental results of the speed and the corresponding phase current response for the IM drive system with the PI controller are shown in Fig. 5.3 (a) and (b), respectively. The drive was started with no load and the rated command reference speed (188.5 rad/sec) conditions.

The corresponding experiment with resulting speed response and actual "a" -phase current for the simplified FLC based IM drive with fuzzy logic slip gain block are shown in Figs. 5.4 (a) and (b), respectively, under same conditions of no load and rated speed (188.5 rad./sec.). When the results are compared, it is clearly seen that the drive based on the FLC reaches the command speed very promptly, with no observable overshoot or oscillation, and follows command speed accurately, with a steady-state error, which appears to be zero. The PI controller - based induction motor drive is slower to start up and has an overshoot of 12 rad/sec (114 rpm), before it settles down to the reference speed. The phase current in both cases is within the rated values of the machine, but under the FLC induction machine draws at least 25-30% less current. When compared to computer simulations results, this experimental phase current appears to be higher than one simulated, but as stated before, considerably lower than that of the PI –based IM drive under similar conditions.

The performances of the PI –based IM drive under the full rated load and rated speed conditions are shown in Fig. 5.5 (a) and (b), which represent the rotor speed and phase current responses, correspondingly. The analogues experiment results are presented in Fig. 5.6 (a) and (b). These figures show the speed response and actual phase current, respectively, for the simplified FLC based IM drive with fuzzy logic slip gain tuning at rated load and rated speed conditions. Again, the experimental results show that the FLC –based drive system follows the command speed without noticeable steady-state errors and achieves this quickly and with no observable overshoot or speed curve oscillation. Once more, the phase current is within the rated values of the induction machine, but this time it has increased compared to the no load conditions case. On the other hand, for the PI –based controller drive system the steady state error have been increased for the full load condition compared to the no load condition. The experimental results in Figs. 5.3 - 5.6 show that simplified FLC –based IM drive system is capable of following the reference command speed and maintain the load, drawing a low current at full load conditions.

The next experiment tests were carried out to investigate the drive performances with controllers we compare at the low reference speed conditions. The speed responses of the PI –based IM drive are shown in Fig. 5.7 (a) and (b), respectively, for the low command speed (90 rad./sec.). Figure 5.8 (a) - (b) shows the speed response, quadrature (q) -axis command current and actual phase current, respectively, for the simplified FLC based IM drive at light load and low speed (90 rad/sec) conditions. It can be seen that the drive system tracks the command speed punctually, without any visible overshoot and no noticeable steady-state error. As in computer simulations, i_q^* and i_d^* command currents appear to be very close to each other. This can be explained by similarity of fuzzy inference blocks used for

determining these two currents. The phase current remains within rated values of the induction machine. These results verify that the simplified FLC -based IM drive can achieve fast and follow the low command speed precisely without overshoot and steady state error, under various loading conditions. On the other hand, the PI –based IM drive system used for comparative reasons suffers from overshoot. Consequently, the efficacy of the simplified FLC –based system has been validated under different speed and loading conditions.

The responses of the PI –based IM drive and simplified FLC -based IM drive due to sudden changes in command speed at light load conditions are shown in Fig. 5.9 (a), (b) and (c), respectively. These graphs make obvious that the simplified FLC –based IM drive under consideration can follow the command speed quickly and accurately during a step increase (from 145 rad/sec to 188.5 rad/sec) and decrease (from 188.5 rad/sec back to 145 rad/sec) of speed under the light load conditions (with the load ranging from 25% to no more than a half of the rated load). There are no detectable overshoot or undershoot or steady-state errors. The PI –based IM drive system, however, has an overshoot of 12 rad/sec and also a big undershoot at the moment of change in command speed, which might be not acceptable for the high performance drive system. Whereas, the proposed simplified FLC –based IM drive is more robust and faster than the conventional PI –based system. The increase in phase current frequency with increase of the motor speed is illustrated in Fig. 5.9 (c). To conduct this experiment an ANSI "C" software program (dspeed.c) was used to achieve the real-time step change of command speed (included in Appendix E).

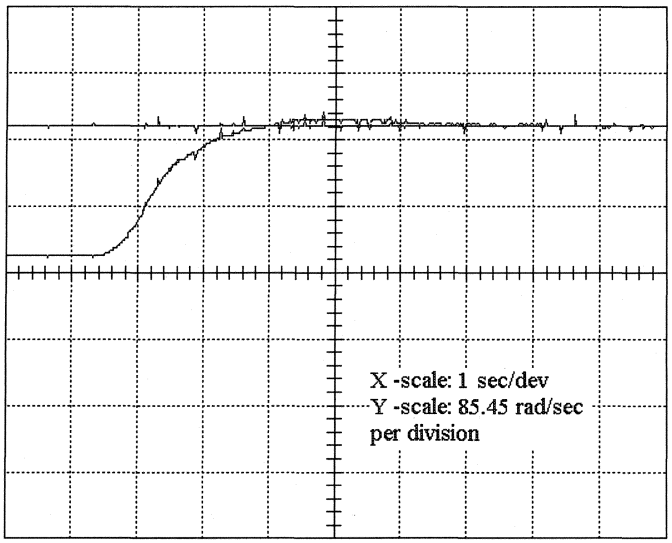
In industrial applications the sudden change of load is a common situation and the high performance drive has to be able to withstand and operate under these circumstances. The following graphs were taken to show the experimental

responses of the PI –based IM drive and simplified FLC based IM drive for a sudden increase of load at rated speed: a) speed, (b) q -axis command current and c) actual phase current at steady-state after the drive was loaded. Fig. 5.10 (a), (b) and (c) show the speed response, q and d -axis command current and actual phase current at steady-state after loading, for the PI –based IM drive and simplified FLC -based IM drive, respectively, for a sudden increase of load (from no load to full rated load) at rated command speed. The induction motor running at no load at some point of time was loaded to the maximum rated load. The graphs show that the drive is virtually insensitive to the load disturbance. There is no visible effect on the characteristic at the point of load application. Furthermore, i_d^* command current appears to be a scaled and only slightly different likeness of i_q^*. The phase current appears to be considerably lower (25-30%) than the PI –based IM drive current and certainly remains within rated machine parameters. The graphs also show that the induction motor speed in the case of PI –based controller drops by 14 rad/sec (134 rpm) at the point where load is applied. It is clearly evident that the simplified FLC –based system is more robust than the PI –based drive system in the case of sudden change of load at rated speed.

All the above mentioned results from the laboratory experiments confirm the efficacy and robustness of the simplified FLC -based IM drive. The tested simplified FLC –based IM drive system follows the command speed quickly and without overshoot and no obvious steady state error at different dynamic operating conditions such as suddenly changing speed or load. The nature of the PI controller is such that it has to be developed based on the available motor parameters and known operating conditions, such as reference speed and applicable load range. Therefore, it is not capable to of running the induction motor covering all possible operating ranges with unpredictable uncertainties like sudden load application,

parameter variations or other system disturbances. Also, the PI controller works best if it is individually tuned for each motor because of parameter variations. A particular developed model for one particular motor PI controller is not applicable to the other model or type of the motor without re-tuning. The proposed FLC - based controller does not have these drawbacks because it is insensitive to motor parameter variations over the wide range. The developed simplified FLC -based controller (using three interconnected fuzzy logic blocks) is capable of efficiently adjusting the output with the load or speed change, parameter variations and other system disturbances. The block of the controller responsible of the slip gain tuning further slightly improves the robust characteristics of the drive system. However, this additional block imposes extra calculation burden and can be left out of consideration if the computational resources are limited.

(a)



(b)

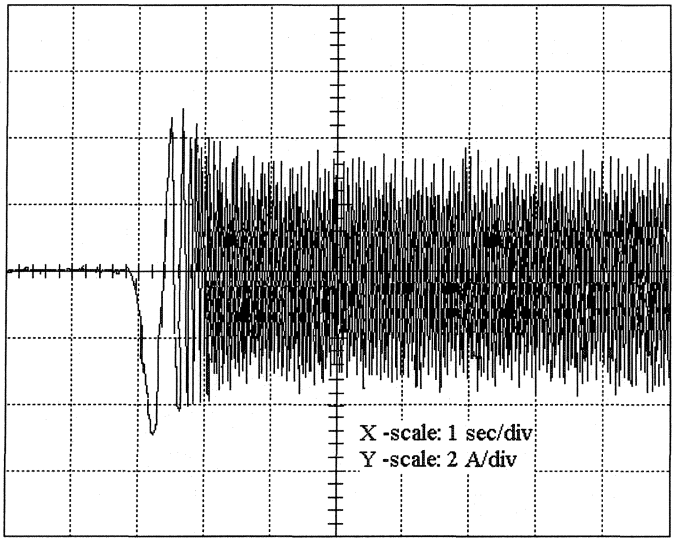
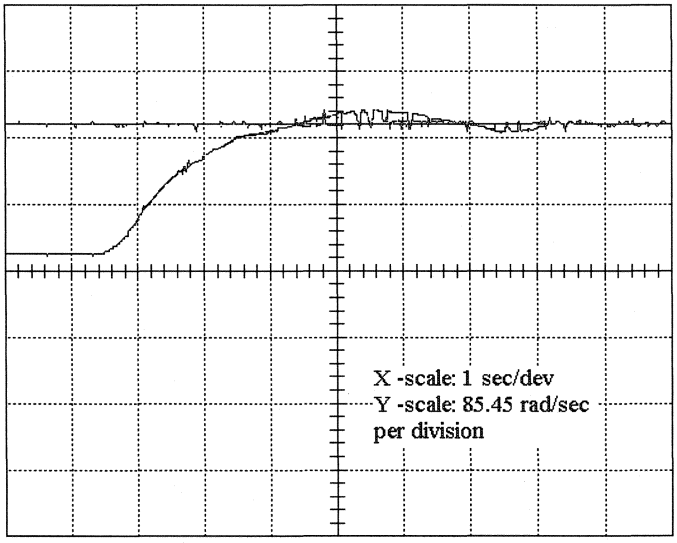


Fig. 5.3. Experimental responses of the PI -based IM drive: a) speed and (b) actual phase current with no load and rated reference speed (188.5 rad/sec) conditions.

(a)



(b)

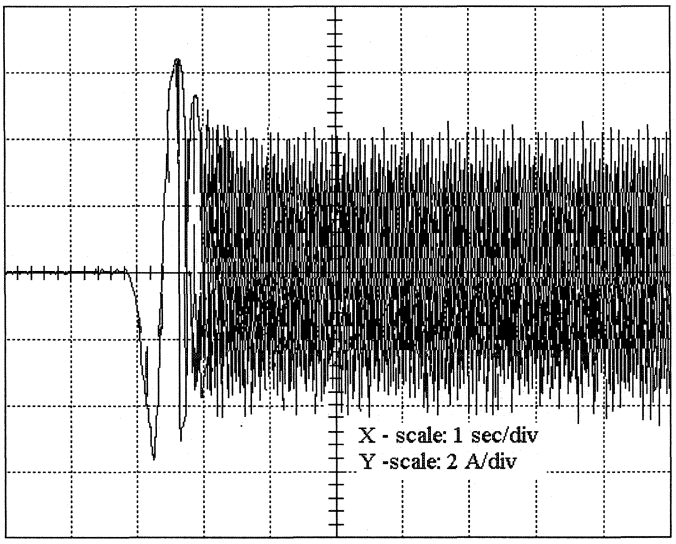
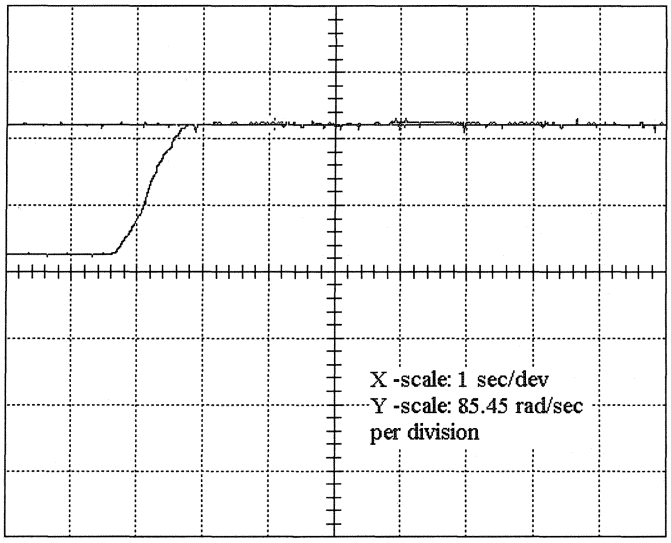


Fig. 5.4. Experimental responses of the PI -based IM drive: (a) speed and (b) actual phase current with full load and rated reference speed (188.5 rad/sec) conditions.

(a)



(b)

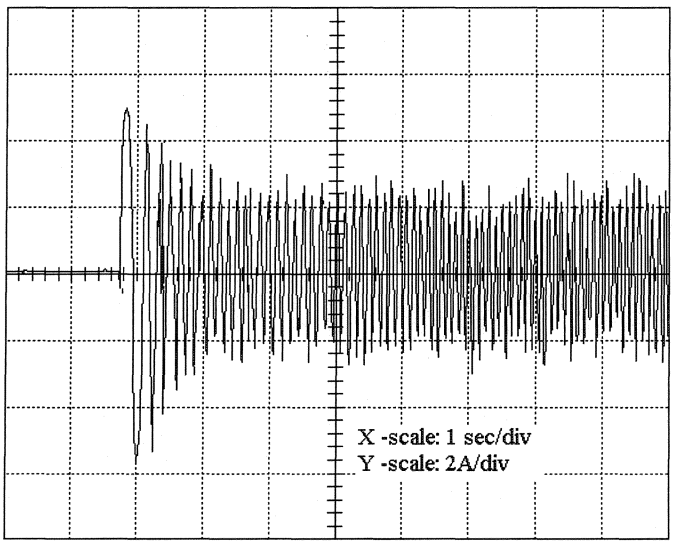
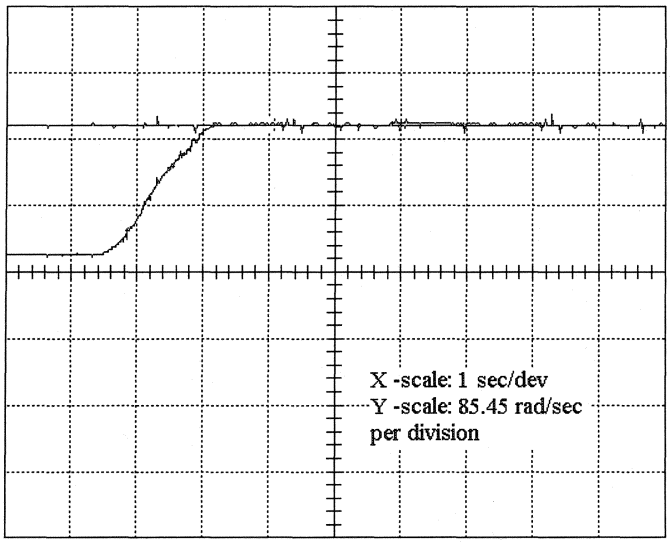
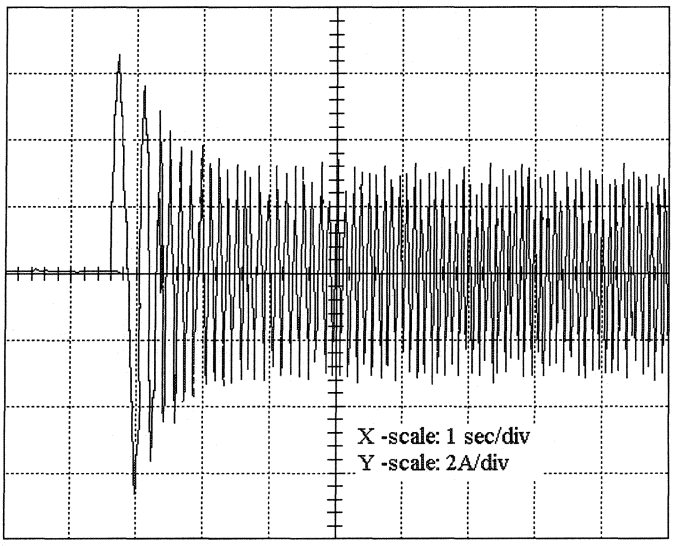


Fig. 5.5. Experimental responses of the simplified FLC -based IM drive: (a) rotor speed and (b) actual phase current with no load and rated reference speed (188.5 rad/sec) conditions.

(a)


X -scale: 1 sec/dev
Y -scale: 85.45 rad/sec
per division

(b)


X -scale: 1 sec/div
Y -scale: 2A/div

(c)

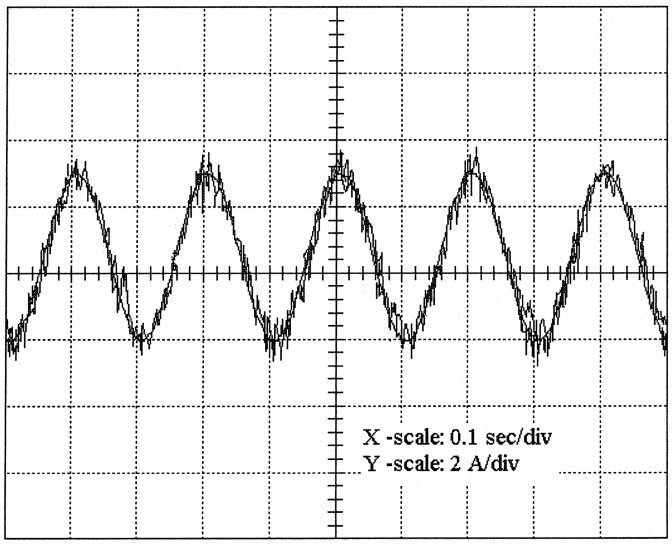
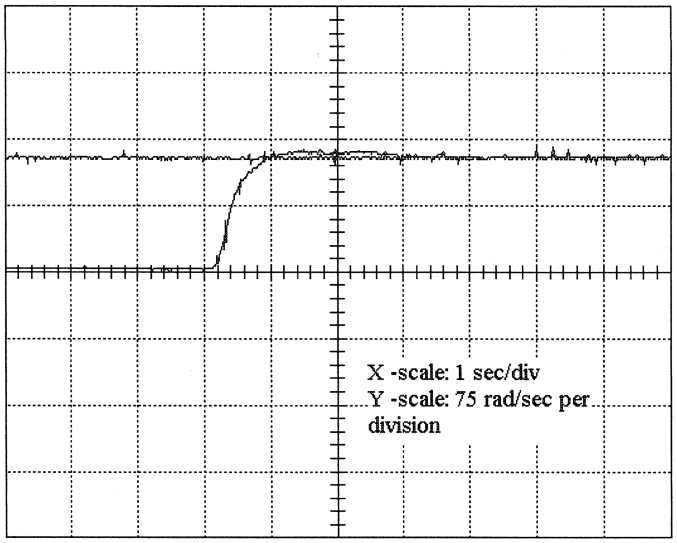


Fig. 5.6. Experimental responses of the FLC -based IM drive: a) speed, (b) actual phase current and (c) scaled steady state actual phase current with full load and rated reference speed (188.5 rad/sec) conditions.

(a)



(b)

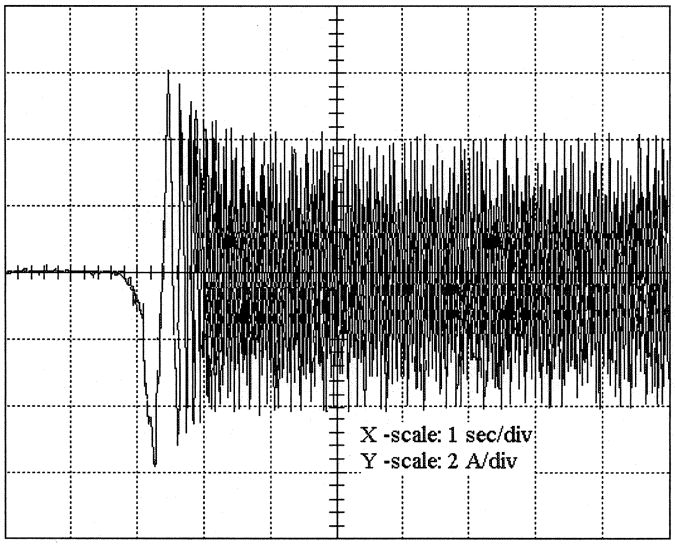
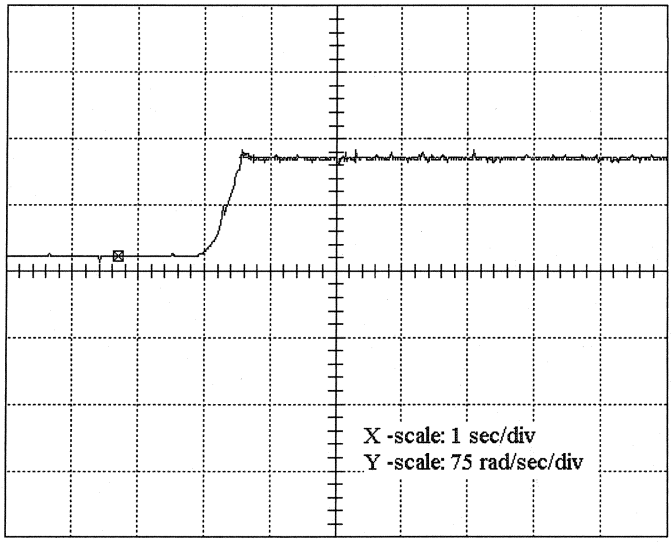
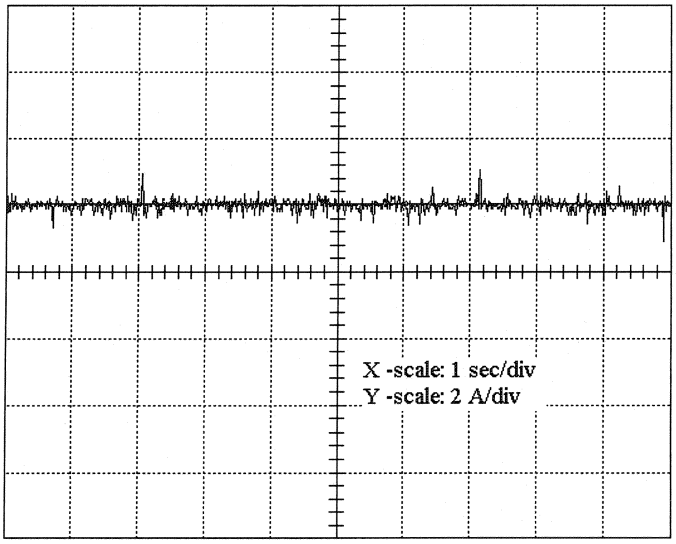


Fig. 5.7. Experimental responses of the simplified PI -based IM drive: (a) speed and (b) actual phase current at light load and low speed (90 rad/sec) conditions.

(a)


X -scale: 1 sec/div
Y -scale: 75 rad/sec/div

(b)


X -scale: 1 sec/div
Y -scale: 2 A/div

(c)

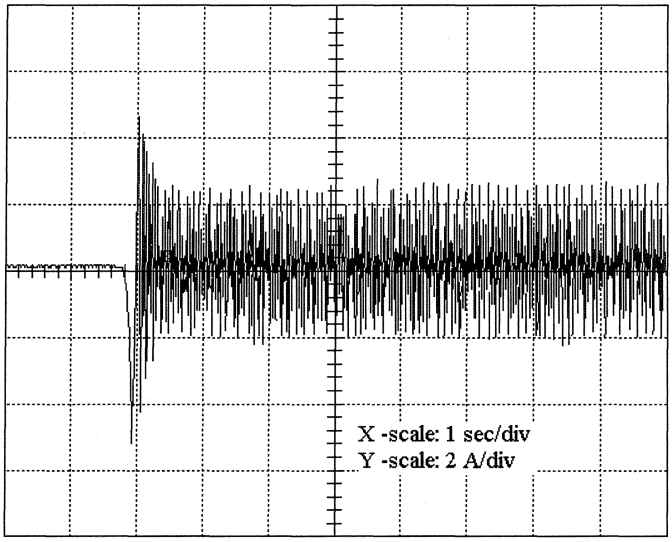
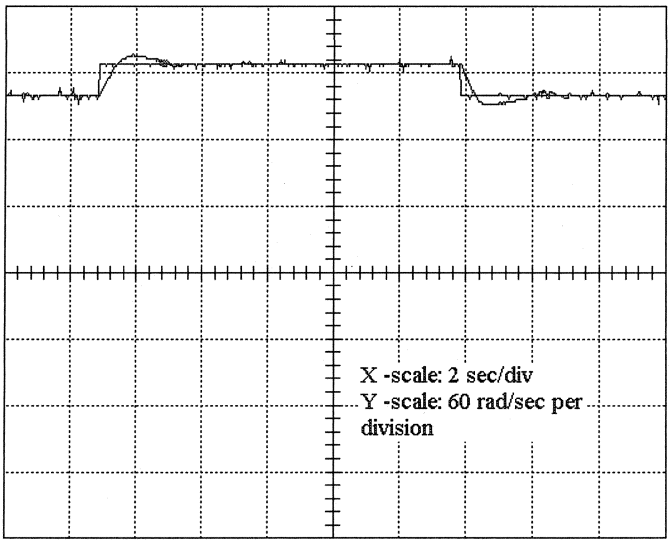
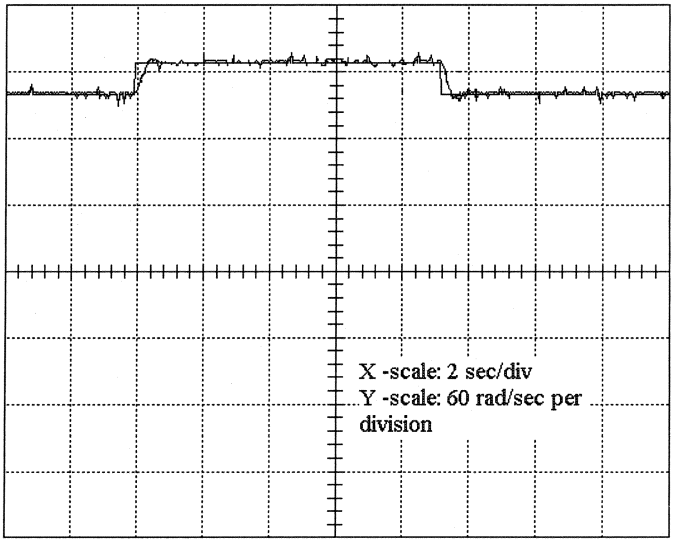


Fig. 5.8. Experimental responses of the simplified FLC based IM drive: (a) motor speed, (b) q and d -axis command currents at steady state and (c) actual phase current at light load and low speed (90 rad/sec) conditions.

(a)


X -scale: 2 sec/div
Y -scale: 60 rad/sec per division

(b)


X -scale: 2 sec/div
Y -scale: 60 rad/sec per division

(c)

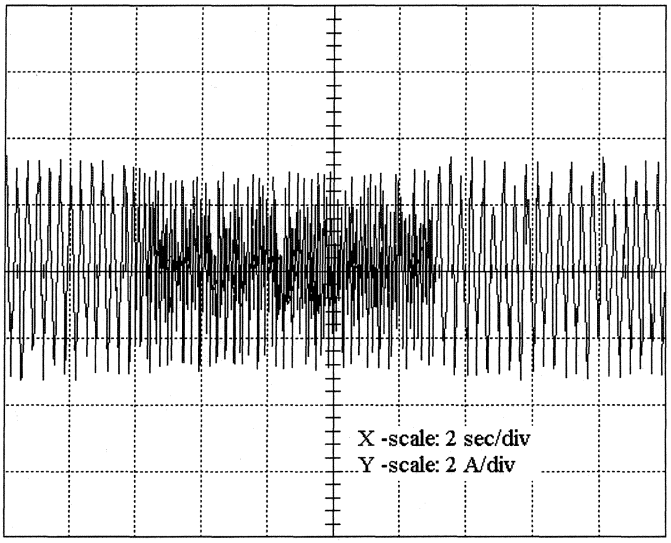
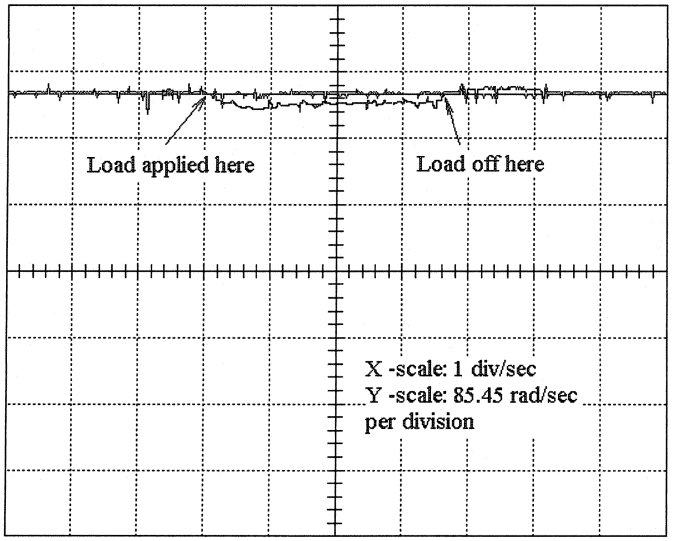


Fig. 5.9. Experimental responses comparison for the PI –based IM drive: (a) - rotor speed; and simplified FLC based IM drive: (b) rotor speed, and (c) actual phase current at light load conditions for sudden changes in command speed.

(a)



(b)

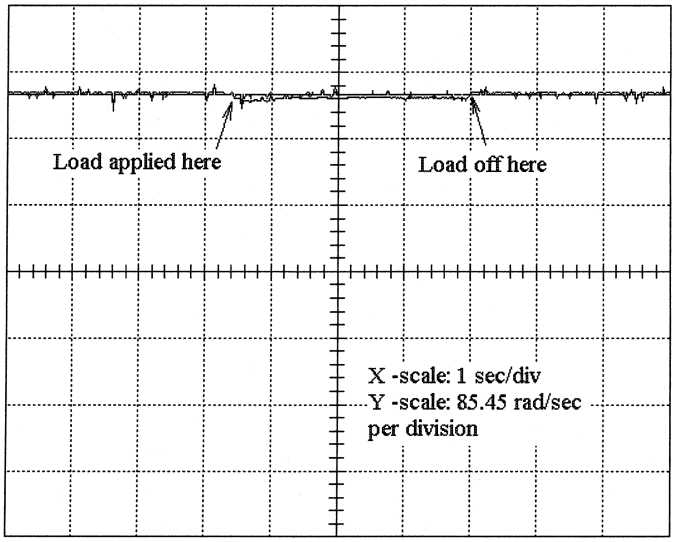


Fig. 5.10. Experimental rotor speed responses comparison of the PI –based (a) and simplified FLC based IM (b) drives for a sudden increase of load at rated speed.

5.4 Concluding Remarks

In this chapter a comprehensive DSP based real-time implementation procedure for the indirect vector control simplified FLC with optional fuzzy slip gain tuning block for the IM drive has been presented. The complete developed drive system has been implemented using a PC –based digital signal processor board, IGBT inverter and a standard 1 hp induction motor, through both hardware and software. In order to demonstrate the superiority of the proposed FLC a conventional PI controller was implemented in real time. The performance of the conventional PI –based controller and the proposed simplified FLC –based IM drive systems have been thoroughly examined at different dynamic operation conditions such as sudden change of command speed or load, as well as some major drive parameter variations. From the experimental results it is clearly seen that the FLC –based IM drive is more robust than the conventional PI –based controller for IM drive system. As a result, it is found that the PI controller can be replaced by the proposed FLC, especially for industrial applications. The developed drive is found to be robust for use in high performance industrial drive applications.

The FLC –based induction machine with i_d not equal to zero draws 15 to 30% less current than the PI –based IM drive and the FLC –based system with i_d = 0. When compared to computer simulations results, the experimental phase currents appear to be higher than simulated, but as stated before, considerably lower than that of the PI –based IM drive under similar conditions, which shows the great potential for industrial high-performance energy-efficient drive applications.

The main FLC –based drive system has advantages as compared to the PI controller –based drive system, which can be stated as: (a) faster start up; (b) less settling time; (c) zero overshoot or undershoot; (d) very low steady state error; (e) 15 to 25 % lower phase current driven. The main drawback limiting the use of this

type of controllers is the high cost of implementation because of high demand for computational power. Though, in high performance drive applications, where the cost of the drive and inverter can be comparable with the cost of specialized DSP equipment needed for FLC implementation, this type of controllers is reasonably good option to use. It is observed that the experimental results have validated the simulation results presented in chapter 4.

The proposed controller can be easily implement for application with other types of motors, such as dc machines or IPMSM drives for instance. A series of real-time experiments utilizing the proposed controller were made using the laboratory 1 hp IPMSM drive. The controller was modified by slightly tuning the output gains. The results achieved (not presented in this work) have been found to be comparable with published recently theoretical and experimental works of M.A. Rahman and C. Butt that employ the maximum torque per ampere strategy (MTPA) for IPMSM drive [89, 90]. This aspect of proposed controller application can be developed in the future works.

Chapter 6

Conclusions

6.1 General

At the beginning of electric drives development the major attention in industry was paid to the development of electric motor drives themselves, their physical characteristics and features. This led to the development of specific groups of drives with desired specifications. After design of drives has been improved to the point when no major changes or contributions could be made to the construction of machines, researchers turned to the development of drive controllers, as described in chapter 1. The design of robust controllers became particularly important in connection with great demand by the industry for the high-performance drives, which set drive requirements to a whole new level. This happened because in most cases, the performance of the drive depends on the robust performance of the speed controller. As discussed before, constant gain types of the speed controllers and their adaptable versions demand an accurate mathematical model of the drive system. They also typically require the exact knowledge of machine parameters, which cannot be constant under dynamic

operating conditions. Furthermore, a system can be an object of abrupt load changes or command speed changes. This makes the design of such controllers burdensome and inaccurate. The drawbacks of these systems are: they are susceptible to overshoot, undershoot, steady-state error and even instability in their applications for high performance IM drives with fixed gain type controllers. Also, conventional adaptive controllers used in modern industry require complex, and usually expensive, hardware and software for their implementation and experienced specialists for their tuning.

In the proposed work it has been proven both by means of computer simulations and laboratory experiments that the standard induction motor drive running with the developed simplified fuzzy logic controller and fuzzy slip gain tuning block can be employed in variable speed high-performance electric motor drives. It was demonstrated that the proposed system layout can overcome these problems by providing an indirect vector control scheme of adaptive speed controller which can accommodate parameter variations, system uncertainties and non-linearities as well as external load disturbances, while requiring a minimum of calculation power and complexity as well as relatively inexpensive hardware.

In chapter 2 the detailed analysis, mathematical and dynamic modeling of the indirect vector controlled pulse-width modulated (PWM) voltage source inverter (VSI) fed IM drive system have been given. Current controllers and the voltage source inverter are chosen and specified. The indirect vector control scheme as well as the development of block diagram of the proposed system including the slip gain tuning block, covered separately, are also detailed in this chapter. A specific mode of drive system operation and the effects of unconnected neutral are discussed at the end of this chapter.

In chapter 3, the step-by-step development of the fuzzy logic controller for the IM drive is given. The concepts of fuzzy logic theory, the explanation of fuzzy linguistic variables, fuzzy sets, fuzzy membership functions as well as the processes of fuzzification, fuzzy rule evaluation and defuzzification have been included. Also, the design of simplified FLC -based speed controller has been presented together with justification and analysis of simplifications made.

In chapter 4 the comparative results of computer simulations using Matlab Simulink software for both the conventional PI -based IM drive with $i_d^* = 0$ and the proposed simplified FLC -based IM drive with $i_d \neq 0$ incorporating a fuzzy slip gain tuning block operation are presented. The results of extensive simulations of compared systems in different modes of operation and disturbances are analyzed and discussed next. It is concluded that simulations have verified the proposed drive can outperform the conventional PI -based controller with regards to speed response, consumed current and general robustness, featuring very much reduced computational burden and no need for high-end hardware.

Chapter 5 gives the step-by-step real-time laboratory experimental implementation of the developed FLC –based indirect vector control system with additional fuzzy slip gain tuning block in the IM drive. The detailed description of experimental set-up, including both hardware elements such as the DSP DS-1102 board, incremental encoder, Hall-effect current sensors, base drive circuits, IGBT inverter, as well as software realization in ANSI "C" high-level programming language have been presented. Experimental examination of the proposed drive system performance was made under varying dynamic operating conditions including sudden reference speed or load change, abrupt system parameter variations, etc. The results validated the computer simulations presented in chapter 4 and verified that the simplified FLC –based IM drive demonstrates the superior

performance over the conventional PI –based IM drive. It can achieve and follow command speed faster than its rival, promptly and without overshoot or undershoot and with nearly zero steady-state error for a variety of operating conditions, whereas the PI controller based system suffers from overshoot, undershoot and steady-state error at a number of different operating conditions. Thus, the robustness of the controller has been verified.

Computer simulations and laboratory experiments were conducted with the reference speed ranging from 30% to 170% and load ranging from 0 to 150%. The proposed system has been found to be stable under different operating conditions of speed and loading. Further research concerning stability of the proposed system layout is beyond the scope of this work and can be considered for future works.

6.2 Major Contributions of this Work

The major contributions of this work are as following:

1. A new controller layout for the FLC -based controller has been developed for the IM drive incorporating additional fuzzy slip gain tuning block has been developed. The simplification method of this controller has been used to reduce computational burden and to maintain robustness and high performance standards at the same time. The performance of the proposed drive system has been proven both by extensive computer simulations as well as laboratory experimentation under various dynamic operating conditions.

2. For comparative reasons the proposed FLC drive with $i_d \neq 0$ has been simulated and experimentally implemented along with a conventional PI – based controller drive with using a $i_d^* = 0$ technique. These comparative simulations have illustrated the encouraging ability of the proposed controller to handle ever-changing motor dynamics as well as non-linear load disturbances better, over a wide speed range including speeds, higher than synchronous speed.

3. After the implementation codes in high-level ANSI "C" programming language have been written, the proposed simplified FLC has been implemented in real-time for a standard 1 hp IM available in the laboratory, using a PC –based digital signal processor controller board DS-1102. Therefore, it has been proved, that the simplified FLC –based IM drive has a superior performance as compared to the conventional PI –based controller drive. The performance, efficacy and robustness of the proposed drive have been verified by experimental investigation of the drive for various dynamic operating conditions such as sudden changes in command speed and loading, as well as abrupt system parameter changes.

6.3 Future Work

The presented work investigates the performance of the proposed interconnected fuzzy logic block scheme for the IM drive controller. Also, a series of computer simulations (not presented here) were made using the same controller

applied to the IPMSM drive. Some optimistic results were obtained. Therefore, the future scope of this work can be seen as an application of the developed simplified FLC scheme to a number of drive types, such as IM, IPMSM, SM, etc. Also, one of the directions of study could be further simplification of the proposed controller logic. The work in this direction could result in an universal controller for different electric machine types, with the standard hardware and software controller part applied to different machines. An approach like this would decrease the price of high-performance drives dramatically, if adopted by industry. The major parameter differences of the drive systems can be overcome by controller output gain self-tuning. Some form of adaptive technique, such as artificial neural networks or further FLC and neuro-fuzzy implementations, could be used for this process.

6.4 Conclusions

The main conclusions of this thesis are as follows:

- The proposed interconnected fuzzy logic block scheme layout can be used in high performance IM drive applications for a range of operating speeds at rated conditions. Additional fuzzy slip gain tuning block is also an asset, if the computational burden is not a major concern.

- The developed IM drive was found to be applicable for effective use for the motor speed control that would be accurate, precise and robust under different dynamic operating conditions, such as command speed change or load change, system uncertainties and/or parameter variations.

- The proposed drive provides a sufficient alternative to the conventional PI - based controller IM drives including the FLC –based IM drives, which are using the $i_d^* = 0$ technique. The developed FLC –based drive employing an intelligent adaptive control method including the $i_d^* \neq 0$ technique can be widely used in industrial IM drives with minimal complexity, and significantly reduced computational burden. Therefore, the proposed drive system meets the cost effective hardware requirements.

- The efficacy and robustness of the new control system layout has been proved both by extensive computer simulations and laboratory experiments.

References

[1] P. C. Sen, "Electric Motor Drives and Control – Past, Present and Future", *IEEE Trans. on Industrial Electronics*, vol. 37, no. 6, Dec. 1990, pp. 562-575.

[2] M. A. Rahman, "Permanent Magnet Synchronous Motors - A Review of the State of Design Art", *Proceedings of International Conference On Electric Machines, Athens*, 1980, pp. 312-319.

[3] M. A. Rahman, "Modern Electric Motors in Electronic World", *IEEE/IECON'93 Conference Record*, (Hawaii), pp. 644-648.

[4] K. Hasse, "On the Dynamic Behaviour of Induction Machines Driven by Variable Frequency and Variable Voltage Sources", *ETZ A, Bd. 89,* H.4, pp. 77-81, 1968.

[5] G. R. Slemon, *Electric Machines and Drives*, Addison-Wesley Publishing Company, 1992.

[6] K. Hasse, "Zur Dynamik Drehzahlegeregelter Antriebe mit Stromrichtergespeisten Asynchron-Kurzschlusslaufermaschinen," *Dissertation* Techn. Hochsch., Darmstadt, West Germany, 1969.

[7] F. Blaschke, "The Principle of Field Orientation as Applied to the New Transvector Closed-Loop Control System for Rotating Field Machines," *Siemens Review*, Vol. 34, May 1972, pp. 217-220.

[8] M. A. Rahman, T. S. Radwan, A. M. Osheiba and A. E. Lashine, "Analysis of Current Controllers for Voltage Source Inverters", *IEEE Trans. Ind. Electronics*, vol. 44, Aug. 1997, pp. 477-485.

[9] P. C. Sen, *Principles of Electric Machines and Power Electronics*, New York: Wiley, 1988.

[10] M. H. Rashid, *Power Electronics-Circuits, Devices and Applications*, Englewood Cliffs, NJ: Prentice Hall, 1996.

[11] B.K. Bose, *Power Electronics and AC Drives*, Prentice-Hall Inc., Englewood Cliffs, NJ, 1986, 402p.

[12] A. Bellini, G. Figalli and G. Ulivi, "A Microcomputer-Based Optimal Control System to Reduce the Effects of the Parametric Variations and Speed Measurement Error in Induction Motor Drives," *IEEE Trans. on Ind. Appli.*, Vol. IA-22, No. 1, Jan./Feb. 1986, p.p. 42-50.

[13] R. Lorenz, "Turning of Field-Oriented Induction Motor Controllers for High-Performance Applications," *IEEE Trans. on Ind. Appli.*, Vol. IA-22, No. 2, March/April 1986, p.p. 293-297.

[14] R. D. Lorenz and D. Lawson, "Performance of Feedforward Current Regulators for Field Orientated Induction Machine Controller," *IEEE Trans. on Ind. Appli.*, Vol. IA-23, No. 4, July/Aug. 1987, p.p. 597-602.

[15] T. Kume and T. Iwakane, "High-Performance Vector Controlled AC Motor Drives, Application and New Technologies," *IEEE Trans. on Ind. Appli.*, Vol. IA-23, No. 5, Sept./Oct. 1987, pp. 872-880.

[16] T. Matsuo and T. A. Lipo, "A Rotor Parameter Identification Scheme for Vector-Controlled Induction Motor Drives," *IEEE Trans. on Ind. Appli.*, Vol. IA-21, No. 4, May/June 1985, pp. 624-632.

[17] T. A. Lipo, "Flux Sensing and Control of Static AC Drives by the Use of Flux Coils," *IEEE Trans. on Magnetics*, Vol. MAG-13, Sept. 1977, pp. 1403-1408.

[18] L. A. Jones and J. H. Lang, "A State Observer for the Permanent-Magnet Sychronous Motors", *IEEE Trans. Ind. Electronics*, Vol. 36, No. 3, Aug. 1989, pp. 374-382.

[19] J. C. Hung, "Practical industrial control techniques", *in Proc. IEEE IECON'94*, pp. 7-14.

[20] D. M. Brod and D. W. Novotny, "Current Control of VSI-PWM Inverters", *IEEE Transactions on Industry Applications*, Vol. IA-21, No. 4, May/June 1985, pp. 562-570.

[21] M. N. Uddin, T. S. Radwan, G. H. George and M. A. Rahman, "Performance of Current Controllers for VSI-Fed IPMSM Drive", *IEEE Transactions on Industry Applications*, Vol. 36, No. 6, Nov./Dec. 2000, pp. 1531-1538.

[22] I. Takahashi and T. Noguchi, "A new quick response and high efficiency control strategy of an induction motor", *IEEE Trans. Ind. Appl.*, vol. 22, pp. 820-827, Sept./Oct. 1986.

[23] R. Ueda, T. Sonada, K. Koga, and M. Ichikawa, "Stability analysis in induction motor driven by V/f controlled general propose inverter", *IEEE Trans. Ind. Appl.*, vol. 28, pp. 472-481, March/April 1992.

[24] A. B. Plunkett, "A Current Controlled PWM Transistor Inverter Drive", *Conf. Rec. IEEE-IAS Annu. Meeting*, Oct. 1979, pp. 951-957.

[25] A. B. Plunkett and D. L. Plette, "Inverter-induction motor drive for transit cars", *IEEE Trans. Ind. Appl.*, vol. 18, pp. 26-37, 1977.

[26] J. Holtz and S. Stadtfeld, "A Predictive Controller for the Stator Current Vector of the AC Machines fed from Switching Voltage Source", *Proc. Int. Power Engineering Conf.*, Tokyo, Japan, Apr. 1983, pp. 1665-1675.

[27] I. Boldea and S. A. Nasar, *Vector Control of AC Drives*, CRC Press, NY, 1992.

[28] B. K. Bose (Ed.), *Power Electronics and Variable Frequency Drives*, IEEE Press, NY, 1996.

[29] E. P. Cornell and T. A. Lipo, "Modelling and design of controlled current induction motor drive system", *IEEE Trans. Ind. Appl.*, vol. 13, pp. 321-330, July/Aug. 1977.

[30] W. Leonhard, "Adjustable speed ac drives", *Proc. Of the IEEE*, vol. 76, pp. 455-471, 1988.

[31] R. W. De Doncker and D. W. Novotny, "The universal field oriented controller", IEEE IAS Annu. Meet. Conf. Rec., pp. 450-456, 1988.

[32] P. Jansen and R. D. Lorenz, "A physically insightful approach to the design and accuracy assessment of flux observers for field oriented induction machine drives", *IEEE IAS Annu. Meet. Conf. Rec.*, pp. 570-577, 1992.

[33] G. Kaufman, L. Garces, and G. Gallagher, "High performance servo drives for machine tool applications using ac motors", *IEEE IAS Annu. Meet. Conf. Rec.*, pp. 579-587, 1989.

[34] X. Xu, R. De Doncker, and D. W. Novotny, "A stator flux oriented induction machine drive", *IEEE Power Elec. Spec. Conf.*, pp. 870-876, 1988.

[35] C. Schauder, "Adaptive speed identification for vector control of induction motors without rotational transducers", *IEEE Trans. Ind. Appl.*, vol. 28, pp. 1054-1061, Sept./Oct. 1992.

[36] T. M. Rowan, R. J. Kerkman, and D. Legatte, "A simple on-line adaptation for indirect field orientation of the induction machine", *IEEE IAS Annu. Meet. Conf. Rec.*, pp. 579-587, 1989.

[37] A. Brickwedde, "Microprocessor-based adaptive speed and position control for electrical drives", *IEEE Trans. on Ind. Appl.*, vol. 21, pp. 1154-1161, Sept./Oct. 1985.

[38] P. Pillay and R. Krishnan, "Modeling of permanent magnet motor drives", *IEEE Trans. on Industrial Electronics*, vol. 35, no. 4, Nov. 1988, pp. 537-541.

[39] P. Pillay and R. Krishnan, "Modeling, Simulation and Analysis of Permanent Magnet Motor Drives, Part I: The Permanent-Magnet Synchronous Motor Drive", *IEEE Trans. on Industry Applications*, vol. 25, no. 2, March/April 1989, pp. 265-273.

[40] P. Pillay and R. Krishnan, "Modeling, Simulation and Analysis of a High Performance Vector Controlled Permanent Magnet Synchronous Motor Drive", *IEEE/IAS Annual Meeting Conference Record*, 1987, pp 253-261.

[41] P. Pillay and R. Krishnan, "Control Characteristics and Speed Controller Design for High Performance Permanent Magnet Synchronous Motor Drive", *IEEE Trans. on Power Electronics*, vol. 5, no. 2, April 1990, pp.151-159.

[42] H. Kubota, K. Matsuse, and T. Nakano, "DSP-based speed adaptive flux observer of induction motor", *IEEE Trans. Ind. Appl.*, vol. 29, pp. 344-348, March/April 1993.

[43] J. Holtz, "Sensoreless position control of induction motor – an emerging technology", *IEEE IECON Conf. Rec.*, pp. I1-I12, 1998.

[44] B. K. Bose and N. R. Patel, "A sensorless stator flux oriented vector controlled induction motor drive with neuro-fuzzy based performance enhancement", *IEEE IAS Annu. Meet. Conf. Rec.*, pp. 393-400, 1997.

[45] G. Buja et. al., "Direct torque control of induction motor drives", *ISIE Conf., Rec.*, pp. TU2-TU8, 1997.

[46] P. Vas, *Sensorless Vector and Direct Torque Control*, Oxford, NY, 1998.

[47] D. Fodor, J. Vass, and Z. Katona, "Embedded controller board for field-oriented AC drives", in *Proc. IEEE IECON'97*, New Orleans, LA, 1997, pp. 1022-1027.

[48] K. Hong and K. Nam, "A disturbance compensation scheme considering the speed measurement delay", *IEEE IAS Annu. Meet. Conf. Rec.*, pp. 403-409, 1996.

[49] U. Itkis, *Control Systems of Variable Structures*, Willey, NY, 1976

[50] A. M. Khambadkone and J. Holtz, "Vector controlled induction motor drive with a self-commissioning scheme", *IEEE Trans. Ind. Elec.*, vol. 38, pp.322-327, Oct. 1991.

[51] L. A. Zadeh, "Fuzzy sets", *Inform. Control*, vol. 8, pp. 338-353, 1965.

[52] L. A. Zadeh, "Outline of a New Approach to the Analysis of Complex System and Decision Processes", *IEEE Trans. on Syst, Man and Cybern.*, vol. SMC-3, 1973, pp. 28-44.

[53] L. H. Tsoukalas and R. E. Uhrig, *Fuzzy and Neural Approaches in Engineering*, Wiley, NY, 1997.

[54] T. Takagi and M. Sugeno, "Fuzzy identification of a system and its applications to modeling and control", *IEEE Trans. Syst. Man and Cybern.*, vol. 15, pp. 116-132, Jan./Feb. 1985.

[55] Y. F. Li and C. C. Lau, "Development of fuzzy algorithm for servo systems", *IEEE Control Syst. Magazine*, Apr. 1989.

[56] B. K. Bose, "Expert systems, fuzzy logic, and neural network applications in power electronics and motion control", *Proc. IEEE*, vol. 82, pp. 1303-1323, Aug. 1994.

[57] P. Vas, *Artificial-Intelligence-Based Electric Machines and Drives*. New York: Oxford Univ. Press, 1999.

[58] I. Miki, N. Nagai, S. Nishiyama, and T. Yamada, "Vector control of induction motor with fuzzy PI controller", in *IEEE IAS Annu. Rec.*, 1992, pp. 464-471.

[59] Y. Tang and L. Xu, "Fuzzy logic application for intelligent control of a variable speed drive", *IEEE Trans. Energy Convertion*, vol. 9, pp. 679-685, Dec. 1994.

[60] E.Cerruto, A. Raciti and A. Testa, "Fuzzy adaptive vector control of induction motor drives", *IEEE Trans. on Power Electronics*, vol. 12, Nov. 1997, pp. 1028-1039.

[61] C. H. Won, S. Kim, and B. K. Bose, "Robust position control of induction motor using fuzzy logic control", in *IEEE IAS Conf. Rec.*, Houston, TX, 1992, pp. 472-481.

[62] L. A. Cabrere, M. E. Elbuluk, I. Husain, "Tuning the Stator Resistance of Induction Motors Using Artificial Neural Network", in *IEEE Trans. on Power Electr.*, vol.12, no. 5, pp. 779-787, 1997.

[63] J. O. P. Pinto, B. K. Bose, L. E. B. Sliva, M. P. Kazmierkovski, "A neural-network-based space-vector PWM controller for voltage-fed inverter induction motor drive", *IEEE Trans. on Ind. Appl.*, vol. 36, no. 6, Nov./Dec. 2000, pp. 1628-1636.

[64] J. O. P. Pinto, B. K. Bose, L. E. B. Sliva, M. P. Kazmierkovski, "A stator flux oriented vector controlled induction motor drive with space vector PWM and flux vector synthesis by neural networks", *IEEE Trans. on Ind. Appl. Society Annual Meeting*, Rome/Italy, 2000, pp. 1605-1612.

[65] J. O. P. Pinto, B. K. Bose, L. E. Borges, M. P. Kazmierkovski, "A neural-network-based space-vector PWM controller for voltage-fed inverter induction motor drive", *IEEE Trans. on Ind. Appl. Society Annual Meeting*, Rome/Italy, 2000, pp. 1605-1612.

[66] A. Rubaai, D. Ricketts, and D. Kankam, "Experimental verification of a hybrid fuzzy control strategy for a high performance brushless DC drive", IEEE Trans. Ind. Appl., vol. 37, no. 32, pp. 503-512, Mar./Apr. 2001.

[67] C. Shiguo, D. G. Holmes and W. A. Brown, "Digital Control of a Servo System using Neural Network", *IEEE/IAS Annual Meeting Conference Record*, 1995, pp. 129-133.

[68] M. A. Rahman and M. A. Hoque, "On-Line Self-Tuning ANN Based Speed Control of a PM DC Motor", *IEEE/ASME Trans. on Mechatronics*, vol. 2, No. 3, Sept. 1997, pp. 169-178.

[69] I. Miki, O. Nakao, S. Nishiyama, "A new simplified Current Control Method for Field-Oriented Induction Motor Drives", in IEEE Trans. on Ind. Appl., vol. 27, no. 6, Nov./Dec. 1991.

[70] C. Y. Won, S. C. Kim, B. K. Bose, "Robust position control of induction motor using fuzzy logic control", in *IEEE-IAS Meet.*, 1992, pp. 472-481.

[71] S. A. Mir, D. S. Zinger, M. E. Elbuluk, "Fuzzy Controller for Inverter Fed Induction Machines", in *IEEE IAS Annu. Rec.*, 1992, pp. 464-471.

[72] J. Fonseca, J.L. Afonso, J. S. Martins, and C. Couto, "Fuzzy logic speed control of an induction motor", Microprocess. Microsyst., vol. 22, pp. 523-534, 1999.

[73] B. Heber, L. Xu, and Y. Tang, "Fuzzy logic enhanced speed control of an indirect field-oriented induction motor drive", *IEEE Trans. on Power Electr.*, vol.12, no. 5, pp. 772-778, 1997.

[74] A. Ibaliden and P. Goureau, "Fuzzy robust speed control of induction motor", in *Proc. ICEM'96, Pt. III*, Vigo, Spain, 1996, pp. 168-173.

[75] M. F. Lai, M. Nakano, G. C. Hsieh, "Application of fuzzy logic in the phase-locked loop speed control of induction motor drive", *IEEE Ttrans. on Ind. Electr.*, vol. 43, no. 6, pp. 630-639, 1996.

[76] M. N. Uddin, T. S. Radwan, M. A. Rahman, "Performances of Fuzzy-Logic-Based Indirect Vector Control for Induction Motor Drive", IEEE Trans. on Id. Appl., vol. 38, no. 5, Sept./Oct. 2002.

[77] P. C. Krause, *Analysis of Electric Machinery*, McGraw-Hill Inc., 1986.

[78] R. H. Park, "Two-reaction theory of synchronous machines – generalized method of analysis – Part 1", *AIEE Trans.*, vol. 48, pp. 716-727, July 1929.

[79] G. Kron, *Equivalent Circuits of Electric Machinery*, John Wiley, NY, 1951.

[80] C. M. Ong, *Dynamic Simulation of Electric Machinery*, Prentice Hall, NJ, 1998.

[81] J. Yen, R. Langari, *Fuzzy Logic*, Prentice Hall, NJ, 1999

[82] B. Ozpineci, L. M. Tolbert, "Simulink Implementation of Induction Machine Model – A Modular Approach", *IEEE International Electric Machines and Drives Conference*, June 1-4, 2003, Madison, Wisconsin, pp. 728-734.

[83] R. Krishnan, *Electric motor drives: modelling, analysis, and control*, Prentice Hall, NJ, 2001.

[84] *Fuzzy Logic Toolbox User Guide*, The Math Works Inc., 1997.

[85] S. Bolognani and M. Zigliotto, "Hardware and software effective configurations for multi-input fuzzy logic controllers", *IEEE Trans. Fuzzy Syst.*, vol. 6, pp. 173-179, Feb. 1998.

[86] *DS1102 Program Loader User's Guide*, dSPACE, Paderborn, Germany, Ver. 2, 1996

[87] H. T. Nguyen, M. Sugeno, R. Tong and R. R. Yager, *Theoretical Aspects of Fuzzy Control*, John Wiley & Sons, Inc., 1995.

[88] M.A. Hoque, Casey Butt and M.A.Rahman, "A Novel Approach for MTPA Speed Control of IPMSM Drive", *Proc. 2nd IEEE International Conference of Electrical & Computer Engineering*, Dhaka, Dec. 26-27, 2002, pp. 336-339.

[89] Casey Butt, M.A. Hoque and M.A. Rahman, "Fuzzy Logic Based Speed Controller for an Interior Permanent Magnet Synchronous Motor Drive Incorporating Maximum Torque per Ampere Mode Operation", *IEEE Industry Applications, 38th IAS Annual meeting*, Salt Lake City, Volume 1, 2003, pp. 499-506.

APPENDIX A

Induction Machine Parameters

IM used in real-time laboratory experiments is a standard 1 hp drive produced by English Company of Canada. It has the following parameters:

Type: D

Rated power: 1 hp

R.P.M. (full load): 1705

Number of phase: 3

Number of poles: 4

Rated current: 3.4 A

Rated frequency: 60 Hz

Rated input line to line voltage: 208 V

Stator resistance, R_s: 4.0 Ω

Rotor resistance, R_r: 1.142 Ω

Stator self-inductance, Ls: 0.368 mH

Rotor self-inductance, Lr: 0.368 mH

Mutual inductance, M: 0.349 mH

Inertia constant $J_m = 0.003$ Kg.m^2

Friction coefficient, B_m: 0.001 Nm/rad/sec

Temp. rise: 50 C

APPENDIX B

Matlab Simulink Model of IM Drive

B.1 General View

This appendix presents the details of the subsystem blocks for the Simulink model of the complete VSI fed IM drive, as described in chapter 2. When the system is modeled in Matlab Simulink, the most effective way to model it is to work referring exactly to the principle equations describing the IM drive system and their interpretations rather than base the individual components of the model strictly on the schematic of the vector control scheme. This is a better approach because the computations that are referred to the direct-quadrature reference frame are very much dependable on the use of mathematical transformation operations that cannot be implemented physically in real-world experimental systems. Therefore, Fig. B.1 presents the Matlab Simulink schematic of the complete current-controlled VSI-fed indirect vector control FLC – based IM drive system that was developed in this work. In the bottom of the figure are presented some of the number of types of speed and load step changes forms, programmed by means of specialized Simulink blocks.

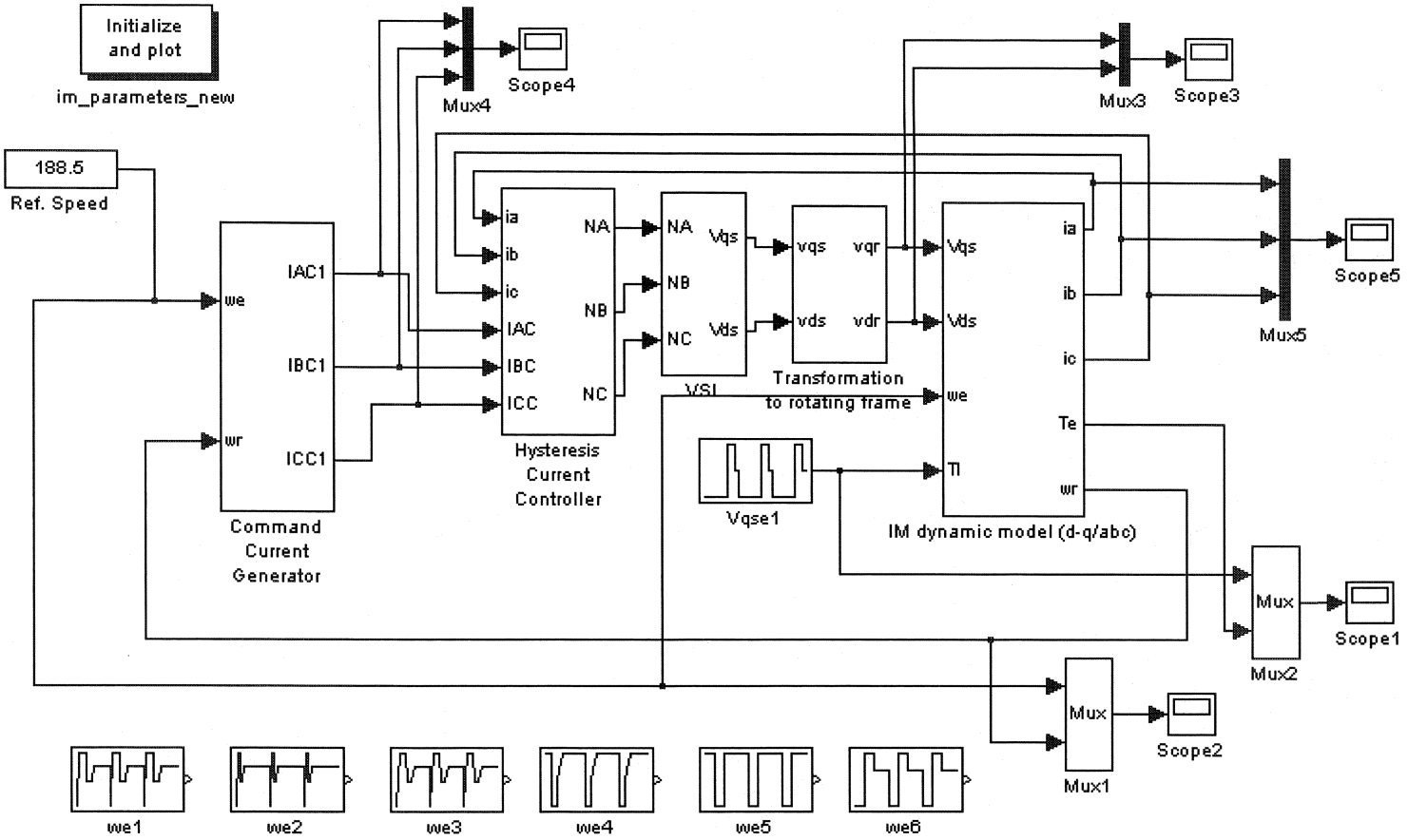


Figure B.1. Simulink model of the developed FLC –based IM drive.

In the following subsections of this appendix the purpose and design of each of the component blocks of the model in detail will be presented. The simplified FLC scheme only is addressed.

B.2 Command Current Generator

The command currents necessary to maintain the desired rotor speed of the induction motor are generated by the use of command current generator block To produce this signal, the necessary inputs are the rotor speed and rotor angular position. The detailed schematic of the command current generator block is shown in Figs. B.2a and B.2b.

The required command torque signal, which is necessary to achieve the desired command rotor speed, is calculated from the difference between the command speed signal and the actual speed signal of the rotor by the FLC. This command torque is then used to obtain the desired quadrature command current i_q^{r*} and direct command current i_d*. Specifically, it is seen that presented scheme uses i_q^{r*} current to attain i_d^{r*} and vice versa due to the system layout. The in-depth step-by-step process of development and simplification of FL algorithms including exact fuzzy membership functions and linguistic rules (blocks FLC1, FLC2 and FLC3 in Fig. B.2a are described in pages 92 and 93) is described in sections 3.4.1 and 3.4.2 of this thesis. Then, using the appropriate transformations presented in chapter 2, three phase currents i_a, i_b, and i_c are found next. Blocks responsible for these transformations are shown in Figs. B.2a and B.2b.

B.3 Current Controller

The following Simulink block in the depicted model represents the controller used for the voltage source inverter. The current controller takes the three phase command currents from the command current generator block and uses them as an input, afterwards it outputs the logic variables NA, NB and NC used to control the VSI. A fixed band hysteresis controller, as shown in Fig. B.3, is employed for this purpose.

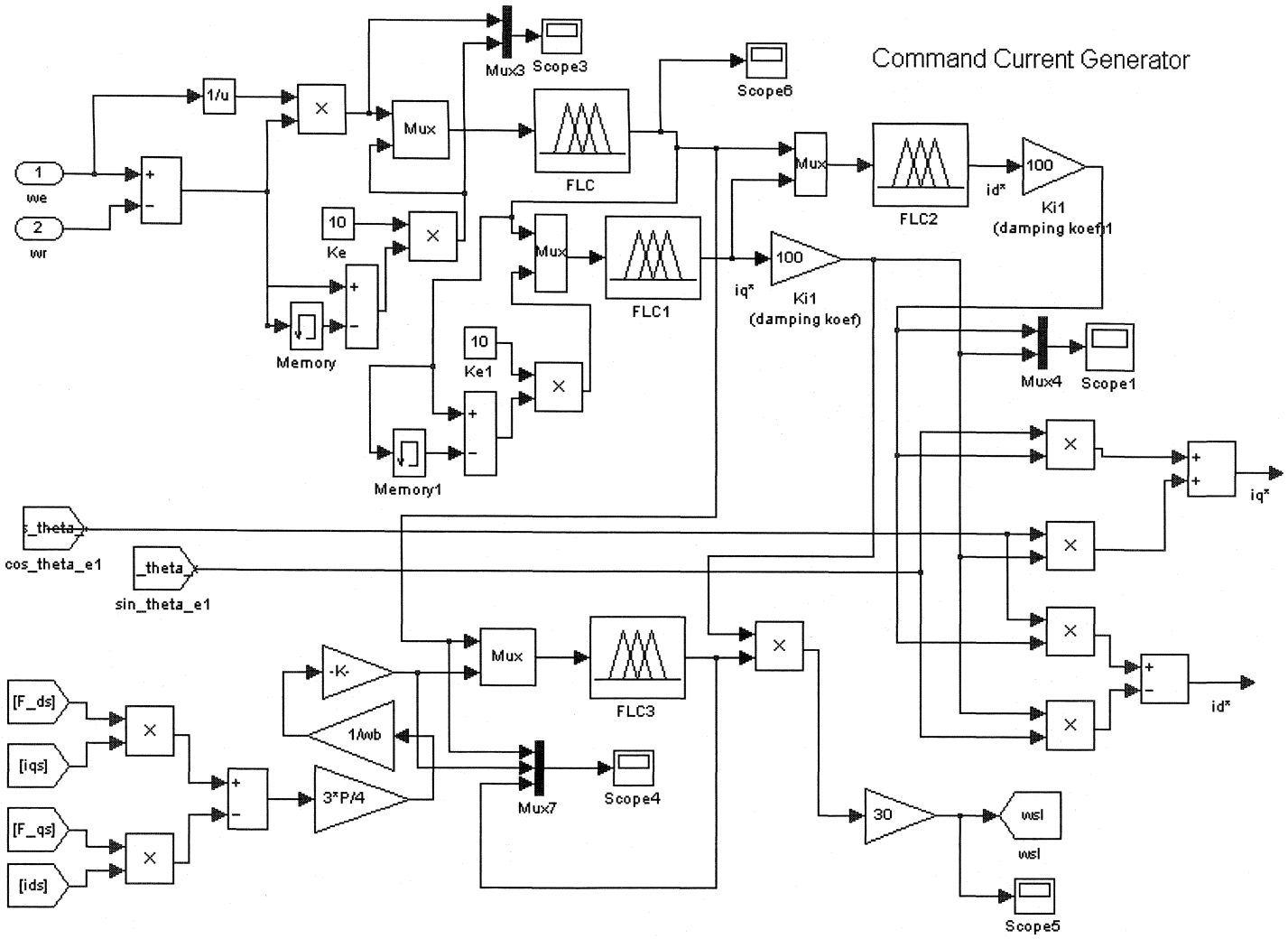


Figure B.2a. Command current generator block with fuzzy slip gain tuning.

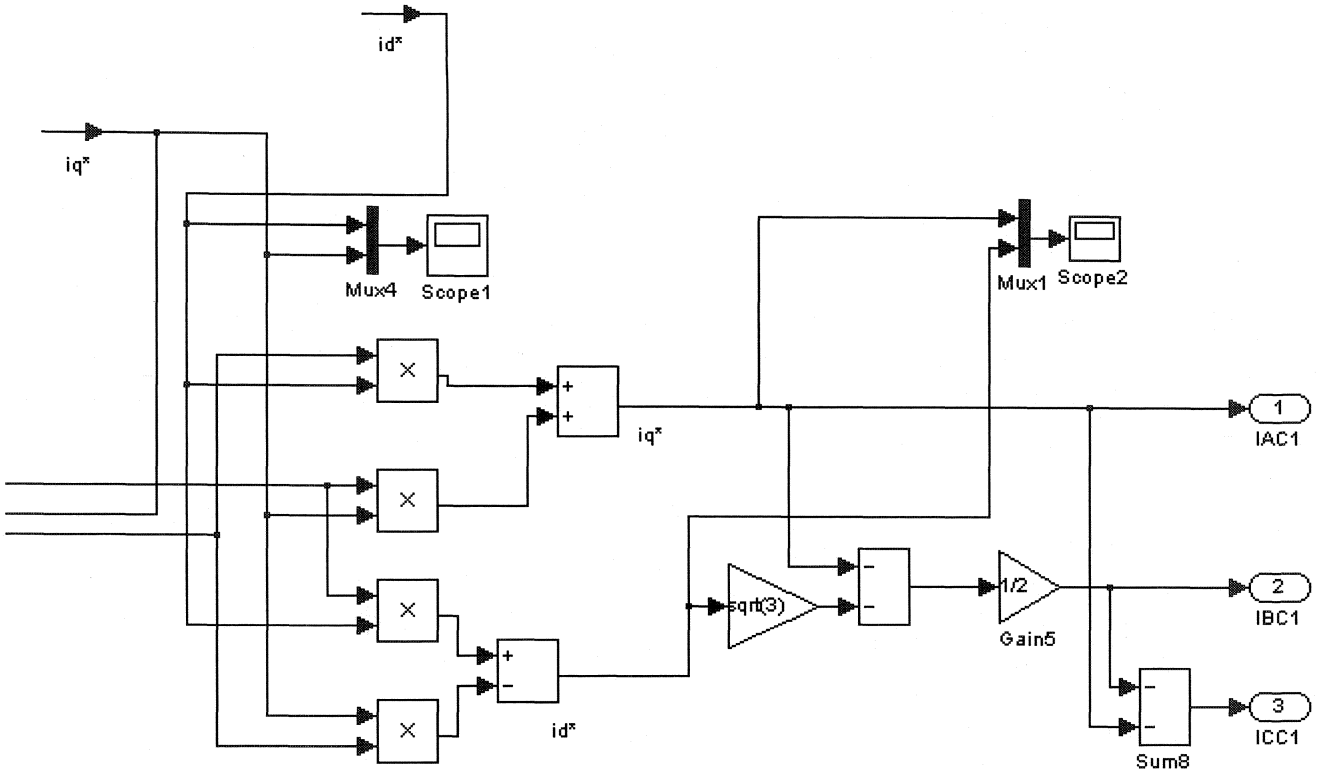


Figure B.2b. Command current generator block (continued).

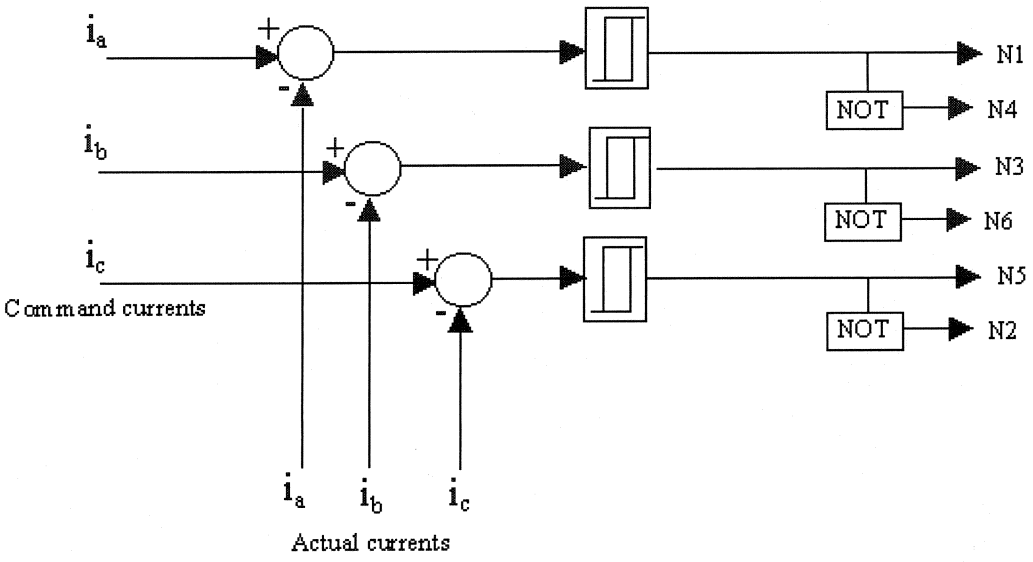


Fig. B.3. Hysteresis current controller scheme.

B.4 Voltage Source Inverter Model

The next block in the presented Matlab Simulink model represents the voltage source inverter and is designed according to the following equation:

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}\begin{bmatrix} NA \\ NB \\ NC \end{bmatrix} V_B \qquad \text{(B.1)}$$

Output signals of this block are the quadrature and direct v_q^* and v_d^* command voltages. This block model is shown in Fig. B.4.

B.5 Model of Transformation to Rotating Frame

The IM model equations require i_q^{r*} and i_d^{r*} currents to give the required outputs ω_r and θ_r. Therefore, v_q^* and v_d^* voltages, as obtained from the VSI, must be transformed to the rotating d-q frame in order to give the rotating voltages v_q^{r*} and v_d^{r*}. This is done, as shown discussed in chapter 2.

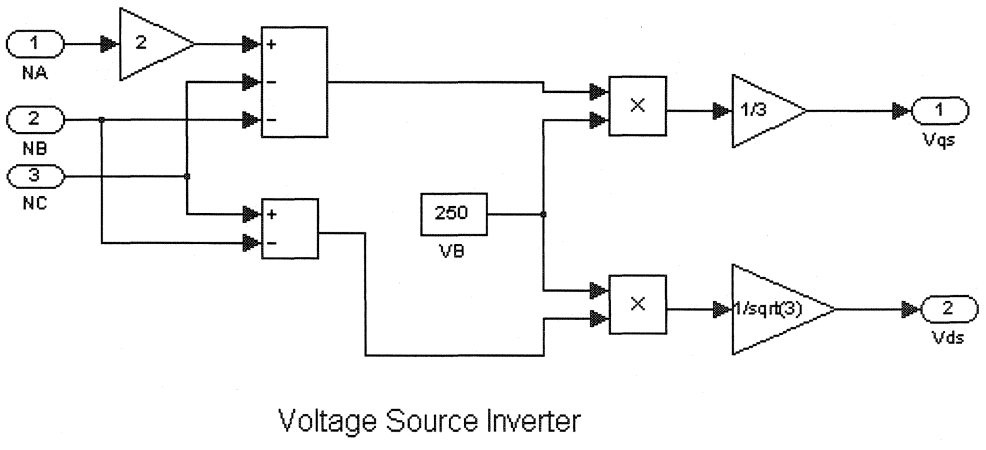


Fig. B.4. Voltage source inverter block scheme.

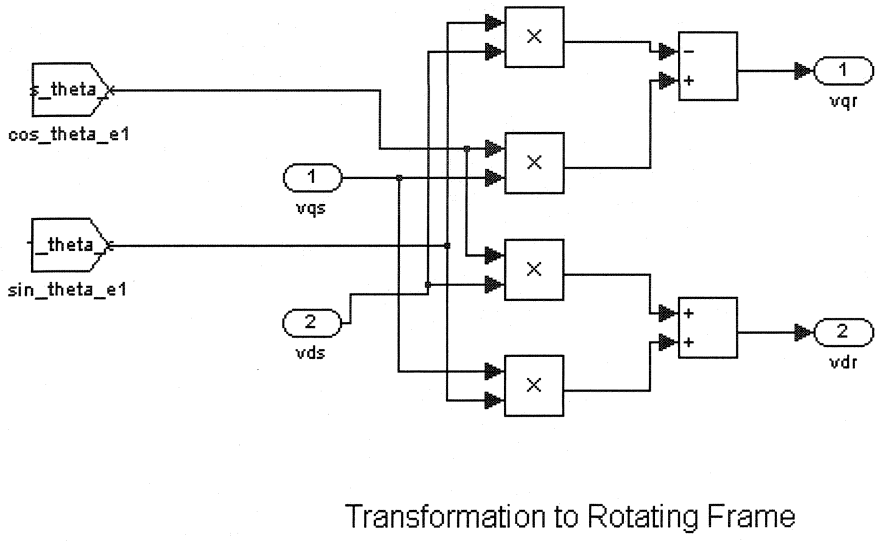


Fig. B.5. Transformation to rotating reference frame block.

B.6 IM Drive Model

This block given below (Fig. B.6) represents the actual induction motor. Inputs are the received earlier rotating command voltages v_q^{r*}, v_d^{r*} and the rotor speed ω_r. In addition, v_q^{r*} and v_d^{r*} must be converted to rotating currents i_q^{r*} and i_d^{r*}, respectively, for use in the equations governing the motor dynamics. This conversion is possible with the input of $\cos\theta_r$ and $\sin\theta_r$ data signals.

Because this block uses several functions to perform the described necessary calculations, it is therefore composed of a number of sub-blocks (namely, three), where each sub-block performs its specific tasks and produces the required output signal.

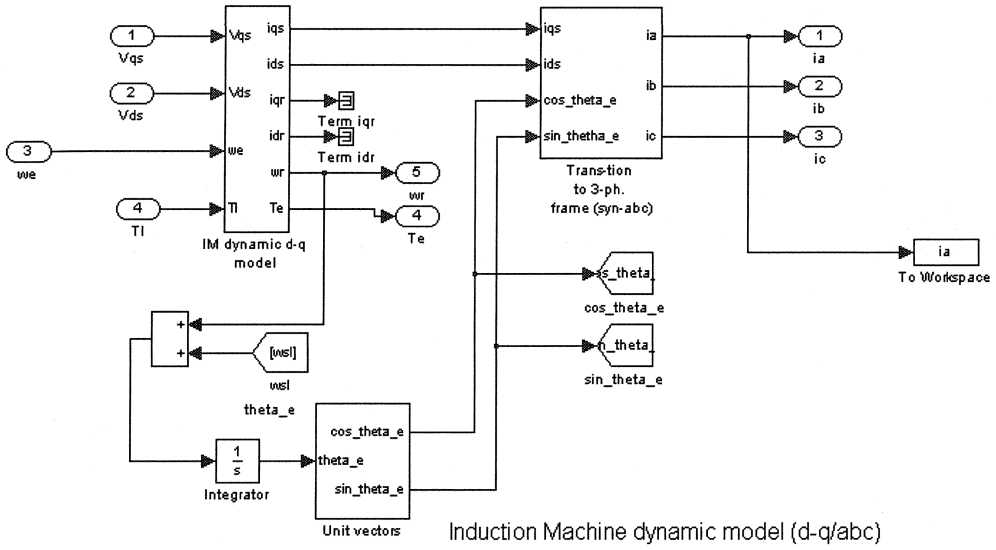


Fig. B.6. IM dynamic model block.

The first sub-block (Fig. B.7) calculates the actual motor currents i_q^r and i_d^r from the respective voltages v_q^{r*}, v_d^{r*} and ω_r. This is done via the IM equations, given in chapter 2 analyses. Machine torque and rotor speed signals are produced as well.

The second sub-block (Fig. B.8) converts quadrature and direct plane rotating currents i_q^r and i_d^r to the three phase currents i_a, i_b, and i_c.

The rotor exact position is found by integrating the sum of the rotor speed and slip speed, after that cos and sin of the received angle are found.

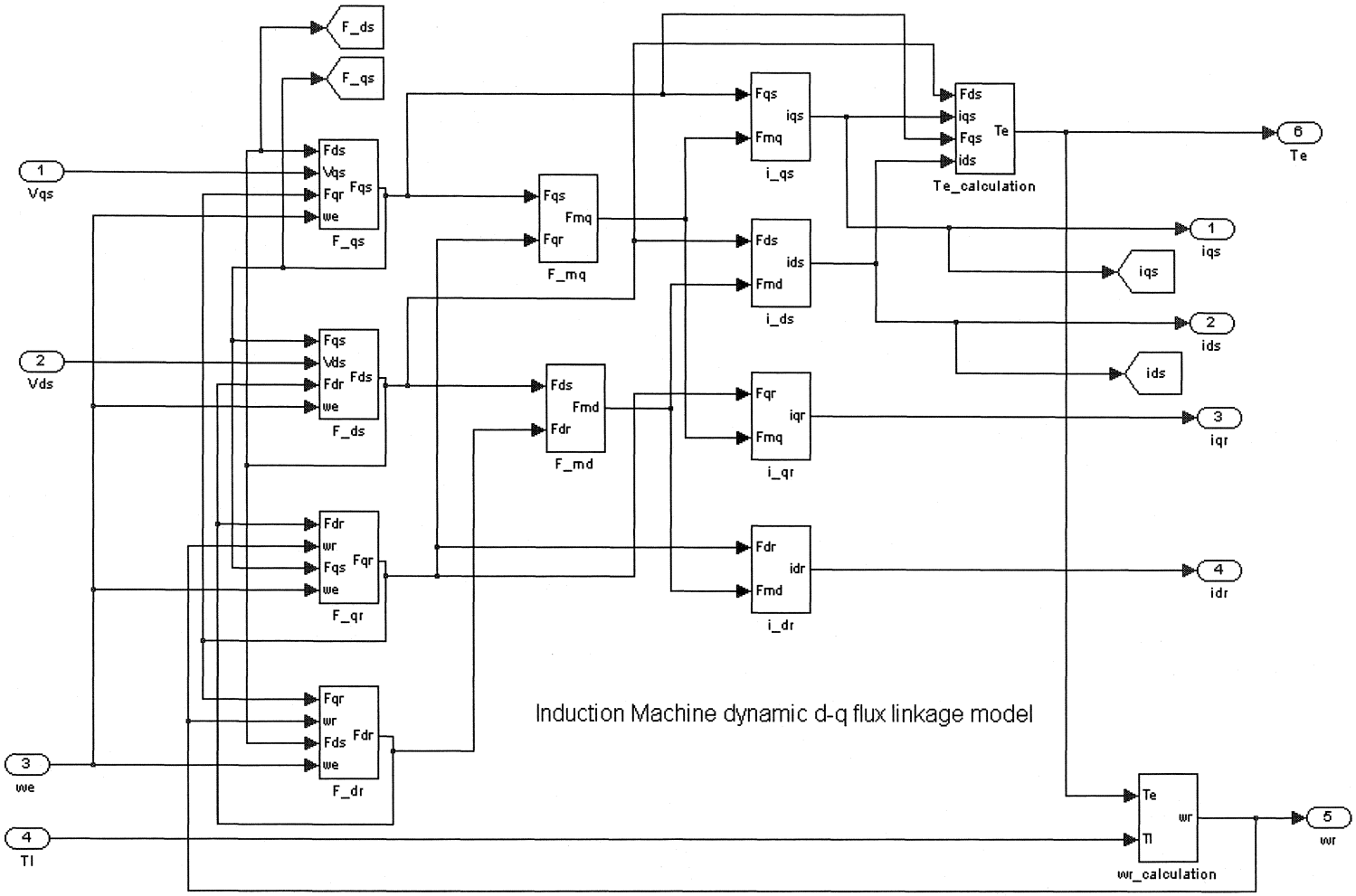
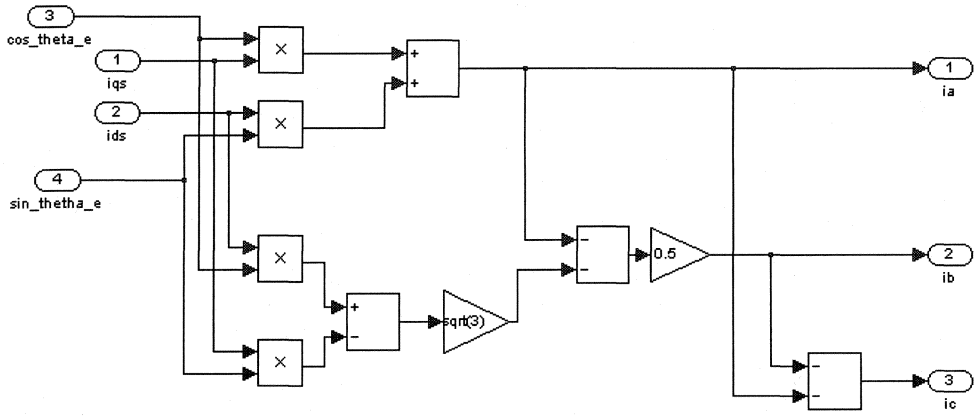


Fig. B.7. Calculation of IM q- and d-axis rotor and stator currents from q- and d-axis command voltages, torque and speed calculation.



Transformation to 3-phase Frame

Fig. B.8. Calculation of the actual stator currents from q- and d-axis currents block (qd to abc transformation).

APPENDIX C

IGBT Inverter Model

The main advantages of two types of transistors, namely the bipolar junction transistor (BJT) and metal-oxide-semiconductor field-effect transistor (MOSFET) are combined in an insulated gate bipolar transistor (IGBT). One of the specific features of IGBT transistor is that because the IGBT has high input impedance, like a MOSFET, the gate draws a small leakage current. An IGBT is a three terminal device consisting of a gate, collector and emitter terminal. Consequently, having the low conduction loses like a BJT, it does not have the second breakdown problem as the BJTs do. Despite the IGBT is a voltage-controlled device, same as a power MOSFET, its performance is closer to that of a BJT than a MOSFET though. . IGBTs maximum switching frequencies are higher than BJTs but slower than the power MOSFETs switching frequencies. The IGBT has lower switching and conduction losses than the MOSFET while sharing many of its attractive features such as ease of gate drive, high peak current capability and ruggedness. Current and voltage ratings of an IGBT can be up to 400 A, 1200 V with a switching frequency of up to 20 kHz.

Fig. C.1 presents the schematic of the IGBT inverter module with its snubber circuit. Even though the IGBT can handle both soft and hard type switching, a

specific snubber circuit has been used to limit the rate of change of voltage across the inverter legs because of the possible system instabilities or unpredictable transients provoked by internal or external system changes.

R = 25 Ω, 50 W

C = 450 μF, 450 V (dc)

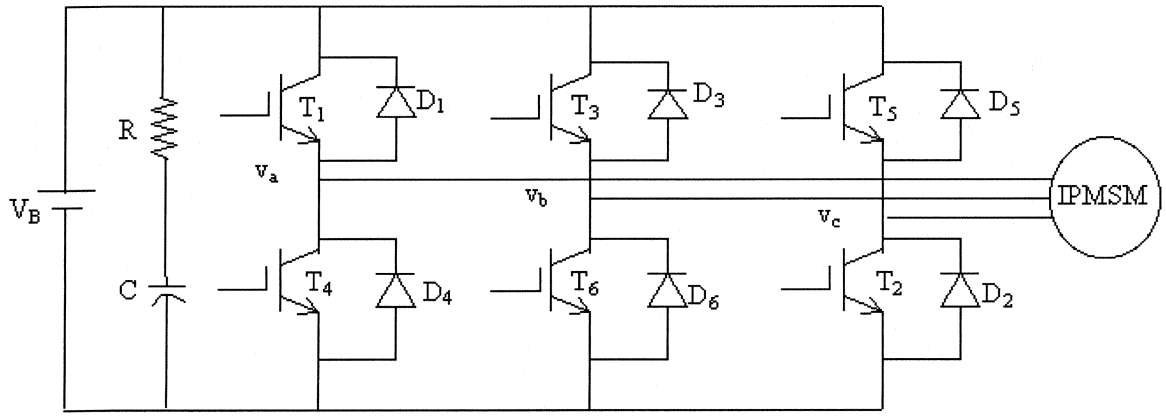


Fig. C.. Basic circuit scheme of an IGBT inverter module.

C. Base Drive Circuits Model

The actual real time circuit requires, that in order to operate the IGBTs as switches, the gate voltages must be of level that the IGBTs are into the saturation mode for low on-state voltages, because the main function of the base drives circuit is to generate six pulses having proper voltage levels for the six IGBTs of the VSI. The output signals of the digital I/O subsystem of the DSP board are pulses that have the magnitudes of 5 V. This voltage to is not sufficient to be used as gate sig-

nals to drive the IGBTs. Consequently, the sufficient amplification is required. In addition, isolation is needed between the logic circuits and the IGBTs because logic signals should be applied between the gates and the corresponding emitters of the IGBTs. This means that for the transistors of the upper legs (T1, T3 and T5), the ground of the logic pulses will not be common. Therefore, a base drive circuit is needed to provide an electric isolation and appropriate voltages levels to the gates of the IGBTs. The base drive circuit used for experiments and assembled in the Power Research Laboratory of engineering department of M.U.N., is presented in Fig. C.2a and C.2b. The SN7407N chip has been used as a level shifter that shifts the voltage level from +5V to +15V. The HP2531 chip is an optical coupler, which has been used to provide isolation between the logic circuit and the power circuit of the IGBT inverter. The IR2130 chip is the main driver, which provides six driving pulses for the six switches of the inverter. In order to provide +20V isolated power to the optical coupler chip and the driver at the same time, an isolated power supply has been built, which is shown in Fig. C.2b.

D. Matlab Simulink IM Drive Initialization Code

In order to run the Matlab Simulink IM drive model it has to be initialized with a simple program code, build in the model itself (im_parameters.m). This code contains the major preliminary induction machine parameters, such as resistances or inductance. It also holds the base frequency, reference speed, torque and their incremental values. When the simulation is complete, the program builds some major plots for the experiment and defines the axis for them.

Resistances and capacitor values:

$R_{18\text{-}23} = 22\ \Omega$, $R_{12\text{-}17} = 680\ \Omega$, $R_{5\text{-}10} = 3.3\ \Omega$, $R_{11} = 3.3\ k\Omega$

$C_{3\text{-}5} = 10\ \mu F$, $C_{6\text{-}11} = 180\ pF$, $C_{12} = 100\ \mu F$, $C_{13} = 47\ \mu F$

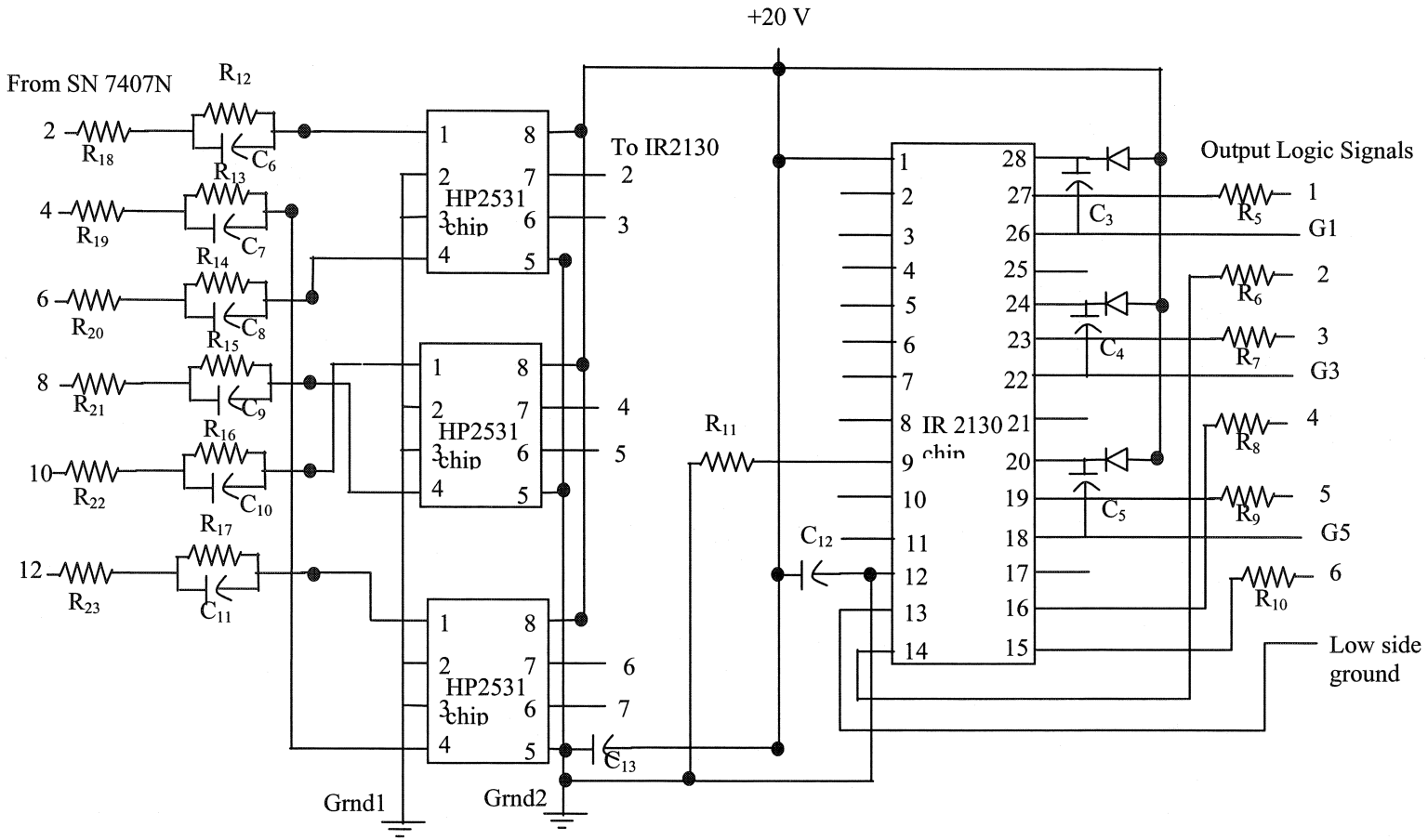
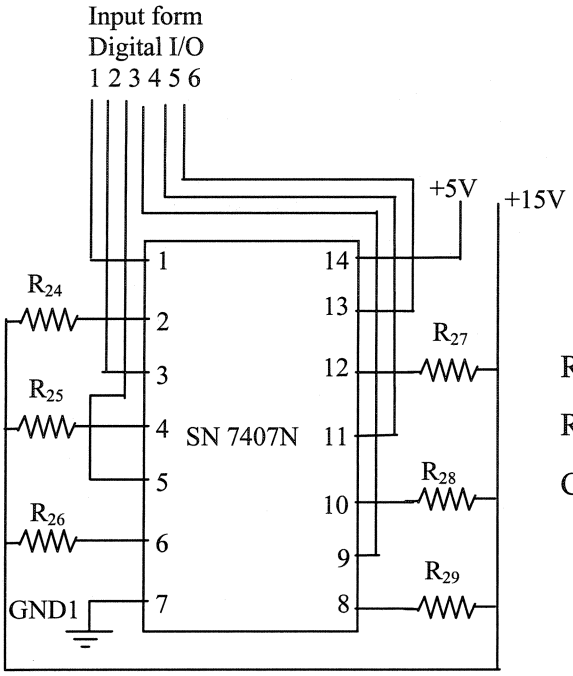


Fig. C.2a. Base drive circuits for the VSI.



Resistance and capacitor values:

$R_{24\text{-}29} = 1.5\ k\ \Omega$, $R_1 = 1.1\ k$, $R_4 = 3.3\ k\ \Omega$

$C_1 = 2200\ \mu F$, $C_2 = 10\ \mu F$

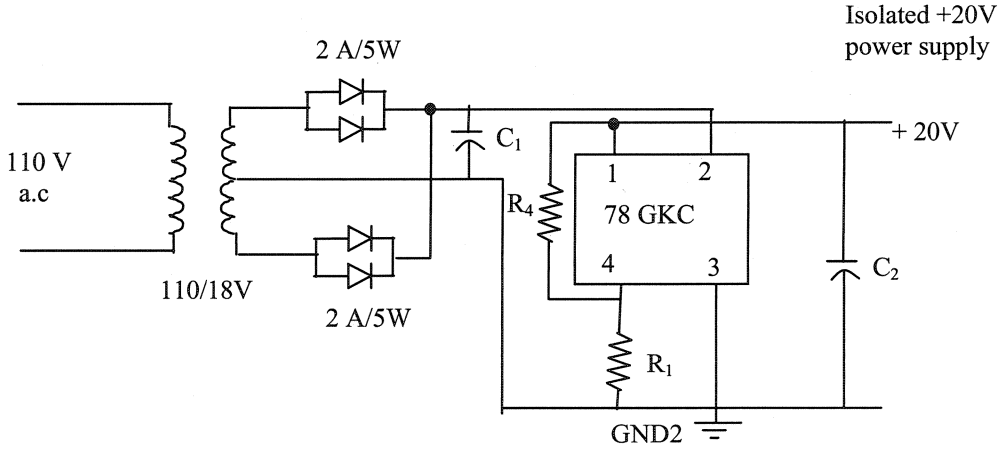


Fig. C.2b. Base drive circuits for the VSI (continued).

APPENDIX D

Matlab Simulink IM Drive Model Initialization Code

```
% Initialization of IM
% im_parameters.m sets up the system parameters for a simulation
% load machine parameters for simulation

Rr = 1.143;  % rotor resistance
Rs = 4;  % stator resistance
Lls = 0.0187;  % stator inductance
Llr = 0.0187;  % rotor inductance
Lm = 0.3489;  % magnetizing inductance
fb = 60;  % base frequency
P = 4;  % number of poles
J = 0.03;  % moment of inertia, Kg*m^2

Lr = Llr + Lm;
Tr = Lr/Rr;

% Impedance and angular speed calculations

wb = 2*pi*fb;  % base speed
Xls = wb*Lls;  % stator impedance
Xlr = wb*Llr;  % rotor impedance
Xm = wb*Lm;  % magnetizing impedance
Xm_star = 1/(1/Xls + 1/Xm + 1/Xlr);

%Vm = 120;  % kill this, not needed

Wr = 188.5;  % reference spead, kill_it
dWr = 100;  % change in ref. speed, delta
Tl_s = 2;
dTl = 1;

% Transfer to keyboard for simulation
disp('Using im_parameters.m to set up the simulation')
disp('Run simulation and type "return" for the plot')
keyboard
clf;
% from the block "to workspace":
plot(y(:,1),y(:,2))
grid on
title('Response to one volt step')
xlabel('time in sec')
ylabel('pu speed wr/wb')
%axis square  % for square plot
```

APPENDIX E

Dspeed.c Program

```
/* dspeed.c ************************************************************
*************************************************************************/

#include <stdlib.h>
#include <stdio.h>

#include <clib.h>          /* Host-DSP interface library include file */

#define DP_MEM_OFFS    0       /* use first dual-port memory address */

unsigned int board_index;
float speed,delta_speed;
void close_and_exit (int error_code)
{
   DSP_unregister_host_app();
   exit(error_code);
}


void write_dual_port_memory (UInt32 address, UInt32 value)
{
   int error;
   error = DSP_lock_board(board_index);
   if(error != DSP_NO_ERROR)
   {
         printf("Error: can't lock board  error = %d.\n\n",error);
         close_and_exit(5);
   }
   error = DSP_write_dual_port_memory(board_index, address, value);
   DSP_unlock_board(board_index);
   if(error != DSP_NO_ERROR)
   {
         printf("Error %d writing the DSP board's dual-port memory !\n\n",
              error);
         close_and_exit(5);
   }
}


void main (int argc, char *argv[])
```

```
{
  int error;

  printf("\n dspeed ``change speed reference from key board..."\n\n");

  if(argc != 2)
  {
        printf("Usage: dspeed board\n\n");
        exit(1);
  }

  error = DSP_register_host_app("dspeed");
  if(error != DSP_NO_ERROR){
        switch(error){
          case DSP_DEVICE_DRIVER_NOT_FOUND:
            printf("\nDevice Driver not installed.\n");
            break;
          case DSP_VXD_NOT_LOADED:
            printf("\nVirtual device driver not installed.\n");
            break;
          case DSP_NO_FREE_HOST_APP_IDX:
            printf("\nNo free host application index.");
            break;
#ifdef NET
          case DSP_NET_ERROR:
            printf("\nNetwork error.");
            break;
#endif
          default:
            printf("\nDSP_register_host_app: error code %d\n",error);
            break;
        }
        exit(1);
  }

  error = DSP_board_index(argv[1],&board_index);
  if(error != DSP_NO_ERROR)
  {
        printf("\nBoard %s not registered, error = %d.\n",
            argv[1],error);
        close_and_exit(2);
  }

  /* initialize output signal with 0.0 */
  write_dual_port_memory(DP_MEM_OFFS,
                                        DSP_cvt_ieee_to_ti((Float32) 0.0));

  /* set output signal to speed */
  do{
  printf("\n please enter the delta_speed in rpm...\n");
  scanf("%f", &delta_speed);
  speed=delta_speed*0.104719755;
  write_dual_port_memory(DP_MEM_OFFS,
```

```
                                        DSP_cvt_ieee_to_ti((Float32) speed));
    } while (delta_speed!=0.0);
    printf("Press RETURN to abort ...\n");
    rewind(stdin);
    getchar();

    /* reset output signal to 0.0 */
    write_dual_port_memory(DP_MEM_OFFS,
                           DSP_cvt_ieee_to_ti((Float32) 0.0));

    close_and_exit(0);
}
```

APPENDIX F

FLC_IM.c Program

```
/* flc_im.c */

 /* Preprocessors */
#include "brtenv.h"
#include "math.h"

#define TS   1.000e-4                    /* sampling period */
#define WRCC 90                           /* reference speed */
#define W_SYNC 188.5                     /* synchronous speed */
#define K_T  1.065                       /* K_T=1/kt */
#define KE   20                          /* change of error const. */
#define KI   35                           /* munltiplication constant */
#define SQR3 1.7321                      /* square root of 3 */
#define TCC  1000.                       /* TCC=1/N*TS */
#define PI   3.14159                     /* pi */
#define INCG 13176.79306                 /* encoder gain */
#define P    2                           /* number of pair pole */
#define NS1 3                            /* # of fuzzy sets for the speed error */
#define NS3 3                            /* # of fuzzy sets for the output */
#define TR 0.3216                        /* Rotor time-constant */
#define APRF 0.00730469                  /* absolute peak (max) rotor flux */
#define LM 0.3489                        /* mutual inductance, (H) */
#define NH 0
#define NL 1
#define PL 2
#define NCC 3
#define PM 4
#define PH 5

long int i, j, k, N=0, NA, NA1, NB, NB1, NC, NC1, IO;
double i_a, i_b, i_c,THI,TH, WR=0.0, DW2, DW1=0.0, TEC=0.0, IQC, IDC, test_start;
double IQSC, IDSC, IAC, IBC ,ICC,TH1=0.0,TH2=0.0,TH3,TH4,THE,WRC, delta_wrc;
double VAS, DE, minval, temp1, temp2;
double cvalue1, cvalue3, fvalue1[NS1], fvalue3[NS3];
double W_error;
double THE_SC=0.0, THE_R=0.0, W_SL=0.0;

/*Initialization for the speed error fuzzy sets */
  double tl_1[3] = {-1.0, 0.0, 1.0};            /* top low value */
  double th_1[3] = {-1.0, 0.0, 1.0};            /* top high */
  double bl_1[3] = {-1.0, -.1, -0.1};           /* bottom low */
  double bh_1[3] = {0.1, 0.1, 1.0};             /* bottom high */
  double cent_1[3] ={-0.66666, 0.0, 0.66666};          /* center of area */
  double areas_1[3] = {1.0 , 0.1, 1.0};         /* area */
```

```
/*Initialization for the control fuzzy sets */
 double tl_3[3] = {-1.0, 0.0, 1.0};              /* top low value */
  double th_3[3] = {-1.0, 0.0, 1.0};             /* top high */
  double bl_3[3] = {-1.0, -.1, -0.1};            /* bottom low */
  double bh_3[3] = {0.1, 0.1, 1.0};              /* bottom high */
  double cent_3[3] ={-0.66666, 0.0, 0.66666};          /* center of area */
  double areas_3[3] = {1.0 , 0.1, 1.0};          /* area */

 /* error flag for chkerror31 at last dual-port memory location */

 int *error = (int *)(DP_MEM_BASE+DP_MEM_SIZE-1);

/* timer0 interrupt service routine */

void  isr_t0()
{
 begin_isr_t0(*error);
 ds1102_ad_start();
 i_a = 11.49*ds1102_ad(3);
 i_c = 10.93*ds1102_ad(4);
 i_b = -(i_a+i_c);
 THI = ds1102_inc(2);
 TH = THI*INCG;
 test_start = dp_mem[1].f;
 /*test_start=ds1102_ad(1);*/
 if (test_start == 0) {
 i = ds1102_inc_read_index(2);
         if(i>0){
               ds1102_inc_clear_counter(2);
               if(TH>0) TH2 = TH2-2.*PI;
               if(TH<0) TH2 = TH2+2.*PI;
                  }
               }
   if(N>=10){
         TH1=TH2;
         TH2=TH;
         N=0;
         TH3=TH2-TH1;
         TH4=TH3;
         if(TH4<0.0) TH3=TH3+2.*INCG;
         WR=TH3*TCC;
        }
 delta_wrc=dp_mem[0].f;
 WRC = WRCC+delta_wrc;

 if( WR>1.1*WRC )
          WR=WRC;
 if( WR<0.0 )
          WR=WRC;
 N+=1;
 THE_R = P*TH;    /* conversion from mech. to elect. */
```

```
/* speed error and change of speed error calculation */
 DW2 = WRC-WR;
 DE = DW2-DW1;
 DW1 = DW2;

 cvalue1 = DW2/WRC;    /* crisp value for the speed error */

/* Fuzzification for the speed error */
for(i=0; i<NS1; i++)
  {
  fvalue1[i] = 0;
    if( (cvalue1 >= bl_1[i]) && (cvalue1 < tl_1[i]))
      fvalue1[i] = (cvalue1 - bl_1[i])/(tl_1[i] - bl_1[i]);
       else
        {
         if( (cvalue1 >= tl_1[i]) && (cvalue1 <= th_1[i]))
           fvalue1[i] = 1;
            else
              {
              if( (cvalue1 > th_1[i]) && (cvalue1 <= bh_1[i]))
                fvalue1[i] = (cvalue1 - bh_1[i])/(th_1[i] - bh_1[i]);
              }
                    }
    }
   if (cvalue1 < cent_1[0])
         fvalue1[0] = 1;
         else
            {
            if (cvalue1 > cent_1[NS1-1])
                fvalue1[NS1-1] = 1;
            }

/* Fuzzy rule base inference engine using one input */
 for (i=0; i<NS3; i++)
                fvalue3[i] = 0.0;
    for (j=0; j<NS1; j++)
     {
     minval = fvalue1[j];
     fvalue3[j] = minval;
     }

/* Defuzzification of the control output */
temp1 = 0;
cvalue3 = 0.0;
 for (i=0; i<NS3; i++)
  {
   temp2 = fvalue3[i]*areas_3[i];
   cvalue3 += temp2*cent_3[i];
   temp1 += temp2;
  }
 cvalue3 /= temp1;

TEC = cvalue3;
```

```
if(WR>.5*WRC){
        if(fabs(TEC)>3.) TEC=3.;
          }
 else {
     if((TEC>15.)||(TEC<12.)) TEC=12;
    }

 W_error = WRC - WR;

 IQC = TEC*K_T;
 IDC = IQC;

 W_SL = TEC*LM/TR/APRF;

        if (W_SL>30) W_SL = 30;

        THE_SC = THE_SC + TS*W_SL;
        THE = THE_R + THE_SC;

 IQSC=IQC*cos(THE)+IDC*sin(THE);
 IDSC=-IQC*sin(THE)+IDC*cos(THE);

 IAC=IQSC;
 IBC=-0.5*(SQR3*IDSC+IQSC);
 ICC=-(IAC+IBC);


/*************************************************************************
*         sinusoidal band hysteresis current controller          *
*************************************************************************/

 if(IAC>=0.0){
        if(i_a>1.2*IAC) NA=0;
        if(i_a<0.8*IAC) NA=1;
        }
       else NA=0;
    if(IAC<0.0){
            if(i_a<1.2*IAC) NA1=0;
            if(i_a>0.8*IAC) NA1=1;
            }
           else NA1=0;
 if(IBC>=0.0){
        if(i_b>1.2*IBC) NB=0;
        if(i_b<0.8*IBC) NB=1;
        }
       else NB=0;
  if(IBC<0.0){
          if(i_b<1.2*IBC) NB1=0;
          if(i_b>0.8*IBC) NB1=1;
          }
          else NB1=0;
 if(ICC>=0.0){
```

```
            if(i_c>1.2*ICC) NC=0;
            if(i_c<0.8*ICC) NC=1;
            }
           else NC=0;
        if(ICC<0.0){
                if(i_c<1.2*ICC) NC1=0;
                if(i_c>0.8*ICC) NC1=1;
                }
               else NC1=0;

   VAS=0.33333*(2*NA-NB-NC);


   ds1102_da(1,.2*IDC);
   ds1102_da(2,.2*IQC);

/*
   ds1102_da(1,.2*i_a);
   ds1102_da(2,.2*IAC);
*/

   ds1102_da(3,.003*WRC);
   ds1102_da(4,.003*WR);

   IO = NA+2*NC1+4*NB+32*NA1+64*NC+128*NB1;     /* Output signals from I/O */
   ds1102_p14_write_io_register(0,0,IO);
   host_service(1,0);
   end_isr_t0();
  }
/*************************************************************************/

void main()
{
 init();                        /*init DAC mode, calibrate ADCs */
 init_slave_DSP_digital_i_o();      /* initialize i/o port as output */
 *error=NO_ERROR;                /* initialize overload error flag */
 dp_mem[0].f=0.0;               /* init 1st dp-mem loc for type float */
 dp_mem[1].f=0.0;               /* init 2nd dp_mem loc for type float */
 ds1102_inc_clear_counter(1);       /* clear incrimental encoder counter */
 start_isr_t0(TS);              /* initialize sampling clock timer  */

 while(*error==NO_ERROR);          /* background process  */
}

/* initialization of digital I/O port as output */
 init_slave_DSP_digital_i_o()
 {
  ds1102_p14_pin_io_init(0x00e7);
  }
/********************** End ********************************************/
```

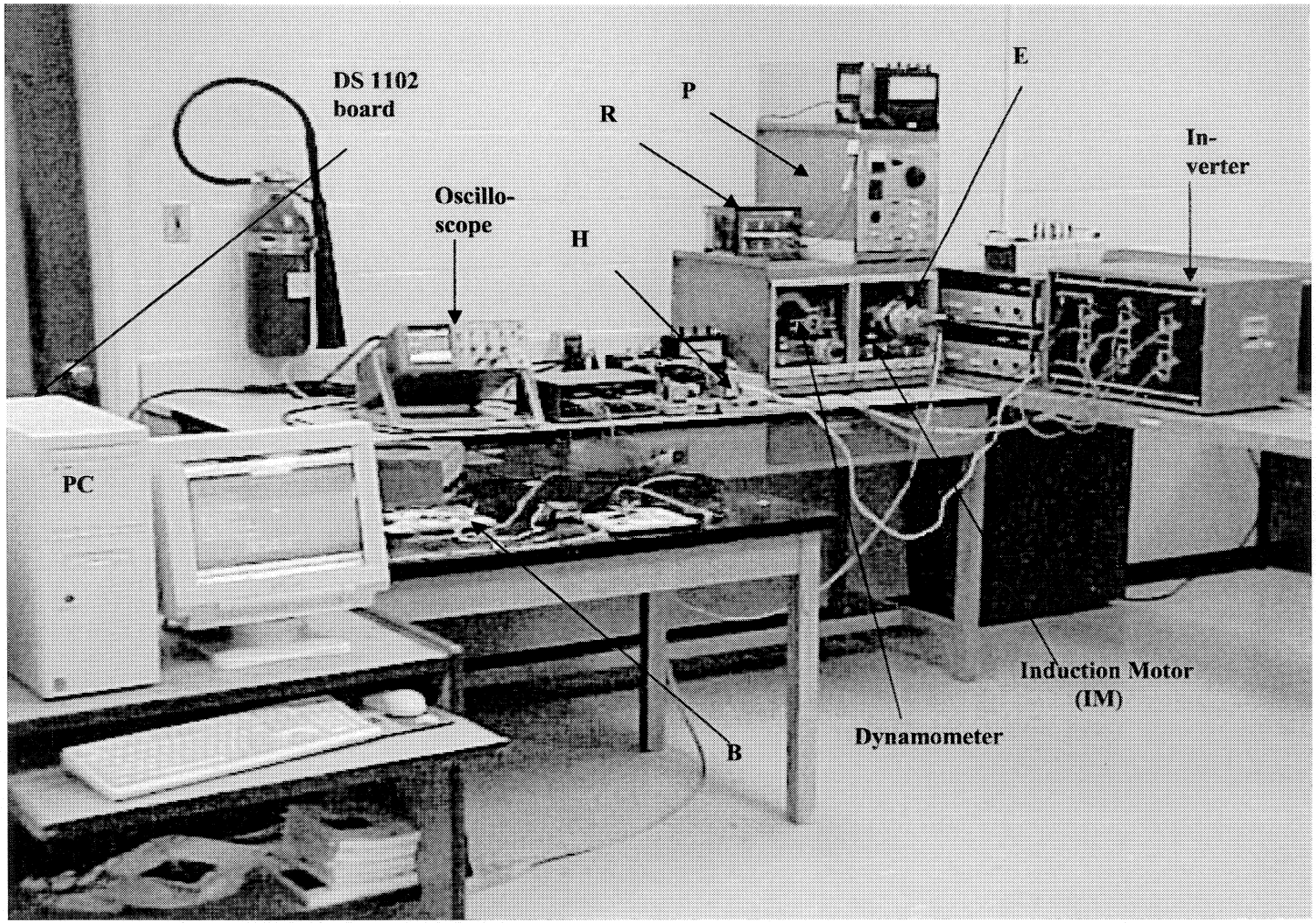


Fig. F.1. Experimental laboratory setup. Where letters represent the following: E – optical incremental encoder, H – Hall effect transducers, B – base drive circuit, P – power supply, R – rectifier, O – oscilloscope.