

**UNIVERSITÉ DU QUÉBEC À CHICOUTIMI**

**MÉMOIRE PRÉSENTÉ  
COMME EXIGENCE PARTIELLE  
DE LA MAÎTRISE EN INGÉNIERIE**

**PAR  
FARHAD HAGHGOEIAN**

**Modèle temps réel d'un estimateur de vitesse à base de  
réseaux de neurones pour une machine asynchrone**



### **Mise en garde/Advice**

Afin de rendre accessible au plus grand nombre le résultat des travaux de recherche menés par ses étudiants gradués et dans l'esprit des règles qui régissent le dépôt et la diffusion des mémoires et thèses produits dans cette Institution, **l'Université du Québec à Chicoutimi (UQAC)** est fière de rendre accessible une version complète et gratuite de cette œuvre.

Motivated by a desire to make the results of its graduate students' research accessible to all, and in accordance with the rules governing the acceptance and diffusion of dissertations and theses in this Institution, the **Université du Québec à Chicoutimi (UQAC)** is proud to make a complete version of this work available at no cost to the reader.

L'auteur conserve néanmoins la propriété du droit d'auteur qui protège ce mémoire ou cette thèse. Ni le mémoire ou la thèse ni des extraits substantiels de ceux-ci ne peuvent être imprimés ou autrement reproduits sans son autorisation.

The author retains ownership of the copyright of this dissertation or thesis. Neither the dissertation or thesis, nor substantial extracts from it, may be printed or otherwise reproduced without the author's permission.

**UNIVERSITY OF QUEBEC AT CHICOUTIMI**

**TRESIS PRESENTED  
AS REQUIRED PART OF  
MASTER OF ENGINEERING**

**BY  
FARHAD HAGHGOEIAN**

**Real-Time Model of Artificial Neural Networks Based  
Speed Estimator for an Induction Machine Drive**

## RÉSUMÉ

La commande vectorielle, connue aussi sous le nom de commande par orientation de flux, permet de contrôler indépendamment le flux et le couple d'une machine asynchrone. Avec ce découplage, la machine asynchrone se comporte alors comme une machine à courant continu qui nous permet d'obtenir de hautes performances.

La commande vectorielle présente l'inconvénient de nécessiter l'emploi d'un capteur de vitesse ou de position. Ce qui impose un surcoût et augmente la complexité des montages. Le fonctionnement sans capteur est l'un des centres d'intérêt des chercheurs à l'heure actuelle. La vitesse est estimée à partir des grandeurs statoriques mesurables, à savoir les courants et les tensions et en utilisant un modèle de la machine. Ce modèle dépend des paramètres structurels de la machine et il est donc clair que les performances du pilotage sans capteur dépendent de la robustesse de la procédure d'estimation choisie.

Dans ce travail, les avantages et les inconvénients des différents schémas de contrôle vectoriel sans capteurs d'une machine asynchrone sont étudiés. Notre choix s'est fixé sur la technique de Modèle de Référence (MRAS) basée sur la force contre électromotrice. Et l'utilisation de réseaux de neurones comme estimateur de vitesse. C'est ce qui constitue la contribution scientifique de notre

travail. Les réseaux de neurones sont des systèmes qui analytiques qui peuvent résoudre un problème où la solution ne peut pas être formulée explicitement. Ce sont des systèmes non algorithmiques, non numériques et parallèles de traitement d'informations. Le système neuronal est entraîné en ligne pour réaliser une application en temps réel. Ce genre d'entraînement ne nécessite pas un calcul préalable et rend le système flexible et robuste dans une large gamme d'opération. Finalement, nous croyons que ce travail est une contribution importante dans ce domaine de recherche.

## ABSTRACT

Thanks to the theory of Vector Control (VC), high performance speed and torque response are achieved from the induction machines nowadays. Driven by a VC controller, an ac machine behaves similar to separately excited dc machine in which the torque and flux are controlled independently. The most important drawback in using this theory is the need to mount the speed/position sensor in the closed-loop configuration which results to several economical and technical problems.

Controlled ac drives without mechanical sensors for speed or motor shaft position have the attraction of low cost and high reliability. The estimation of rotor speed is based exclusively on measured terminal voltages and currents. So, the performance of the controller is depended to the robustness of the speed estimation. A variety of sensorless controlled ac drive schemes are available for practical application. High-performance vector control requires a flux vector estimator in addition. The robustness of a sensorless ac drive can be improved by adequate control structures and by parameter identification techniques.

In this work the advantages and drawbacks of the different schemes for sensorless vector control of the induction machines are studied and finally a back *emf* based Model Reference Adaptive Systems (MRAS) technique using a

recurrent Neural Network as the speed estimator is chosen which is believed to be a new and robust method. Neural networks are analytical systems that address problems whose solution have not been explicitly formulated. A neural network is an information processing system that is non-algorithmic, non-digital, and intensely parallel which makes it a suitable choice for using as the mathematical technique in simulating our estimator. The system is trained online to achieve a real-time application. Such a kind of training has no need to pre-computations and makes the system flexible and robust in a wide range of operation. It is believed that the work is an important contribution in this area of research.

## **ACKNOWLEDGEMENTS**

This work is done thanks to the financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC).

It gives me a great pleasure to acknowledge all people who were involved directly or indirectly in making this work a success. It is a pleasure to convey my sincere thanks to my supervisor, Professor Mohand A. Ouhrouche for the invaluable guidance during each stage of this research.

The author would also like to express his appreciation to the colleagues, especially Mr. Jogendra S. Thongam for his sincere helps.

Finally, I would like to record my deep gratitude to my wife, Shahrzad for her consistent encouragement and support.



## TABLE OF CONTENTS

RÉSUMÉ	iii
ABSTRACT	v
ACKNOWLEDGEMENT	vii
TABLE OF CONTENTS	viii
LIST OF FIGURES	xii

### CHAPTER 0

Introduction . . . . .	1
0.1 Background . . . . .	2
0.1.1 Electric motors . . . . .	2
0.1.2 Adjustable Speed Drives. . . . .	2
0.1.3 Adjustable Speed Drives for Induction Machine . . . . .	3
0.2 Objectives . . . . .	5
0.3 Methodology . . . . .	6
0.4 Contribution . . . . .	7

### CHAPTER 1

Induction Machine Modeling . . . . .	8
1.1 Introduction . . . . .	9
1.2 Induction Machine Construction . . . . .	10
1.3 Current Equations . . . . .	11
1.4 Induced Voltages . . . . .	12
1.5 Running Operation . . . . .	13
1.5.1 Slip Speed . . . . .	13
1.5.2 Electrical model equivalent . . . . .	14

<b>1.6</b>	<b>Representation of Three-Phase Induction Machine . . . . .</b>	<b>16</b>
1.6.1	Representation of three-phase variables . . . . .	16
1.6.2	Complex variable model of three-phase induction motor . .	17
<b>1.7</b>	<b>Space Vector Representation of Three-Phase Induction Machine . . . . .</b>	<b>19</b>
1.7.1	The three reference frames . . . . .	19
1.7.2	Induction motor model in arbitrary reference frame . . . . .	21
1.7.3	Induction motor model in stator reference frame . . . . .	22
1.7.4	Induction motor model in rotor reference frame . . . . .	23
<b>1.8</b>	<b>Two-Phase Induction Machines. . . . .</b>	<b>23</b>

## **CHAPTER 2**

<b>Vector Control of Induction Machine . . . . .</b>	<b>27</b>
<b>2.1 Introduction . . . . .</b>	<b>28</b>
<b>2.2 Induction Machine Control Techniques . . . . .</b>	<b>30</b>
2.2.1 Pole Changing . . . . .	30
2.2.2 Line Voltage Control . . . . .	31
2.2.3 Line Frequency Control . . . . .	32
2.2.4 Constant-Slip Frequency Control . . . . .	34
2.2.5 Closed-Loop Control . . . . .	35
2.2.5.1 Constant V/f Operation . . . . .	35
2.2.5.2 Constant Slip Frequency . . . . .	36
2.2.5.3 Transit Drive System . . . . .	36
2.2.6 Other Speed Control Systems . . . . .	37
<b>2.3 Scalar Control against Vector Control . . . . .</b>	<b>38</b>
2.3.1 Scalar Control . . . . .	38
2.3.2 Vector Control Principals . . . . .	39
2.3.3 Using Vector Control Technique in the Arbitrary Frame. . .	42
2.3.4 Indirect Vector Control . . . . .	47
2.3.5 Direct Vector Control . . . . .	51

## **CHAPTER 3**

<b>Speed Estimation . . . . .</b>	<b>53</b>
-----------------------------------	-----------

<b>3.1</b>	<b>Speed Sensorless Control . . . . .</b>	<b>54</b>
<b>3.2</b>	<b>Speed Estimation Techniques . . . . .</b>	<b>55</b>
3.2.1	Model Reference Adaptive Systems . . . . .	55
3.2.2	Adaptive State Observer Speed Estimation Techniques . . .	56
3.2.3	Extended Kalman Filter Observers . . . . .	57
3.2.4	Extended Luenberger Observers. . . . .	58
3.2.5	Artificial Neural Networks . . . . .	59
3.2.5.1	Introduction . . . . .	59
3.2.5.2	History . . . . .	61
3.2.5.3	Capabilities . . . . .	62
3.2.5.4	Basic principles. . . . .	62
3.2.5.5	Architecture. . . . .	65
3.2.5.6	The ability of network to learn . . . . .	68
3.2.5.7	Validation of ANNs. . . . .	70
3.2.5.8	Learning algorithm . . . . .	71
3.2.6	Choosing the neural network model. . . . .	73

## **CHAPTER 4**

<b>Real-Time Neural Network-Based Sensorless Drive . . . . .</b>	<b>75</b>
<b>4.1 Introduction . . . . .</b>	<b>76</b>
<b>4.2 Simulink Implementation: A Modular Approach . . . .</b>	<b>77</b>
<b>4.3 MRAS Speed Estimation Using Neural Networks . . . .</b>	<b>78</b>
4.3.1 Drawbacks of Flux Based MRAS . . . . .	78
4.3.2 Back EMF based MRAS . . . . .	79
4.3.3 Neural Networks as the Adaptation Mechanism . . . . .	82
4.3.4 Implementation of the Estimator in Simulink . . . . .	84
4.3.4.1 PI with Speed Regulation Block . . . . .	86
4.3.4.2 Rotor Flux Decoupling . . . . .	86
4.3.4.3 $d,q$ to $a,b,c$ . . . . .	87
4.3.4.4 Inverter . . . . .	87
4.3.4.5 Induction Motor . . . . .	89
4.3.4.6 Neural Network Based Speed Estimator . . . . .	91
4.3.5 Real-Time Simulation . . . . .	94
4.3.5.1 Real-Time Systems . . . . .	94
4.3.5.2 Real-Time Systems Requirements . . . . .	95
4.3.5.3 Future development: HIL Implementation . . . . .	97
4.3.4 Simulation Results . . . . .	100

**CHAPTER 5**

<b>Conclusion . . . . .</b>	<b>110</b>
<b>5.1 Contribution . . . . .</b>	<b>111</b>
<b>5.2 Future Topics . . . . .</b>	<b>112</b>

**APPENDIX A**

<b>A Conference Article . . . . .</b>	<b>113</b>
---------------------------------------	------------

**APPENDIX B**

<b>Glossary of Symbols . . . . .</b>	<b>119</b>
<b>A.1 Symbols . . . . .</b>	<b>120</b>
<b>A.2 Subscripts . . . . .</b>	<b>121</b>

**APPENDIX C**

<b>References . . . . .</b>	<b>122</b>
-----------------------------	------------

## LIST OF FIGURES

<b>Figure 0-1.</b> ASD-Motor Circuit	<b>3</b>
<b>Figure 1-1.</b> A three-phase squirrel-cage induction motor	<b>9</b>
<b>Figure 1-2.</b> Y-connected stator and $\Delta$ -connected stator	<b>11</b>
<b>Figure 1-3.</b> Induction Machine Equivalent Circuits	<b>15</b>
<b>Figure 1-4.</b> Thevenin Equivalent Circuits of Machine	<b>16</b>
<b>Figure 1-5.</b> Stator reference frame ( $\alpha, \beta$ ), rotor reference frame ( $D, Q$ ) and arbitrary reference frame ( $d, q$ )	<b>19</b>
<b>Figure 1-6.</b> Location of reference frame axes relative to magnetic axes	<b>21</b>
<b>Figure 1-7.</b> Induction Machine Synchronous $d, q$ Axis Equivalent Circuits	<b>24</b>
<b>Figure 2-1.</b> Block Diagram of a field controlled DC machine	<b>28</b>
<b>Figure 2-2.</b> Open-loop Voltage Controller	<b>31</b>
<b>Figure 2-3.</b> Closed-loop Voltage Controller	<b>32</b>
<b>Figure 2-4.</b> Open-loop Input Voltage & Frequency Controller	<b>33</b>
<b>Figure 2-5.</b> Required Voltage Variation in V/f controller	<b>34</b>
<b>Figure 2-6.</b> Constant Slip Controller	<b>34</b>
<b>Figure 2-7.</b> Closed-loop Speed Control Employing Slip Frequency Regulation and Constant V/f Operation	<b>35</b>
<b>Figure 2-8.</b> Closed-loop Speed Control with Constant Slip Frequency	<b>36</b>
<b>Figure 2-9.</b> Speed Control Scheme for Transit Drive Systems	<b>37</b>
<b>Figure 2-10.</b> Rotor Flux and Current Vectors in Steady states	<b>38</b>
<b>Figure 2-11.</b> Torque-Speed Characteristics of Induction Motor	<b>39</b>
<b>Figure 2-12.</b> Vector Control Frame Conversion	<b>41</b>
<b>Figure 2-13.</b> Flux Vector in a general frame and arbitrary frame	<b>43</b>
<b>Figure 2-14.</b> Indirect Vector Control Architecture	<b>48</b>
<b>Figure 2-15.</b> Tracking Rotor Speed Reference	<b>49</b>

<b>Figure 2-16.</b> Tracking Rotor Flux Reference	<b>50</b>
<b>Figure 2-17.</b> Stator Current Vector Control Components	<b>50</b>
<b>Figure 2-18.</b> Direct Vector Control Architecture	<b>52</b>
<b>Figure 3-1.</b> Adaptive Estimator Structure	<b>55</b>
<b>Figure 3-2.</b> MRAS Speed Estimation Scheme	<b>56</b>
<b>Figure 3-3.</b> A typical model of neuron	<b>63</b>
<b>Figure 3-4.</b> Fully connected network with one hidden layer	<b>65</b>
<b>Figure 3-5.</b> Comparing Different Neural Networks: a) Feed forward NN, b) Fully recurrent NN, c) Jordan NN, d) Elman NN	<b>67</b>
<b>Figure 3-6.</b> Block diagram of unsupervised learning, and supervised learning	<b>69</b>
<b>Figure 3-7.</b> Back-propagation architecture	<b>72</b>
<b>Figure 4-1.</b> Coordinates in stationary reference frame	<b>80</b>
<b>Figure 4-2.</b> Coordinates in Stationary Reference Frame	<b>81</b>
<b>Figure 4-3.</b> Structure of Speed Estimator Using NN	<b>84</b>
<b>Figure 4-4.</b> Induction Field Oriented Control with AC Regulator Using NN	<b>85</b>
<b>Figure 4-5.</b> PI Block	<b>86</b>
<b>Figure 4-6.</b> $d,q$ to $a,b,c$ Frame Converter	<b>87</b>
<b>Figure 4-7.</b> PWM Inverter	<b>88</b>
<b>Figure 4-8.</b> Inverter Block	<b>89</b>
<b>Figure 4-9.</b> Induction Motor Digital Model	<b>90</b>
<b>Figure 4-10.</b> Induction Motor	<b>91</b>
<b>Figure 4-11.</b> Jordan Neural Networks	<b>93</b>
<b>Figure 4-12.</b> Embedded System controlling a Plant	<b>98</b>
<b>Figure 4-13.</b> Conversion of Simulink model to real-time simulation	<b>100</b>
<b>Figure 4-14.</b> No-load Operation Response of Speed estimation System	<b>102</b>
<b>Figure 4-15.</b> Speed Estimation for Load Changes	<b>104</b>
<b>Figure 4-16.</b> Full-loaded Machine Speed Estimation	<b>106</b>
<b>Figure 4-17.</b> Speed Estimation in Combined Operation	<b>108</b>

**CHAPTER 0**  
**INTRODUCTION**

## **0.1 Background**

### **0.1.1 Electric motors**

Motors, as a main parent of converting the electric power to mechanical power, play an important role in industrial manufacturing and in many other applications. In the early years, dc motors used to be the most widely used electrical motor in industry requiring variable speed due to their ease of control. Speed control of dc motors is almost priceless and also easy, so the majority of induction machines were used in constant speed applications when dc ones used to be used everywhere a variable speed was needed. However, dc motors have some disadvantages like brush erosion, maintenance requirements, environmental effects, complex structure and power limits.

On the other hand, induction motors are simple, rugged, small in size, almost maintenance-free and are one of the cheapest motors available at all power ratings [1], as they possess a wide range of speeds compare to the dc motors. The main obstacles to use induction motors are the costly conversion equipments, the complexity of signal processing [2].

### **0.1.2 Adjustable Speed Drives**

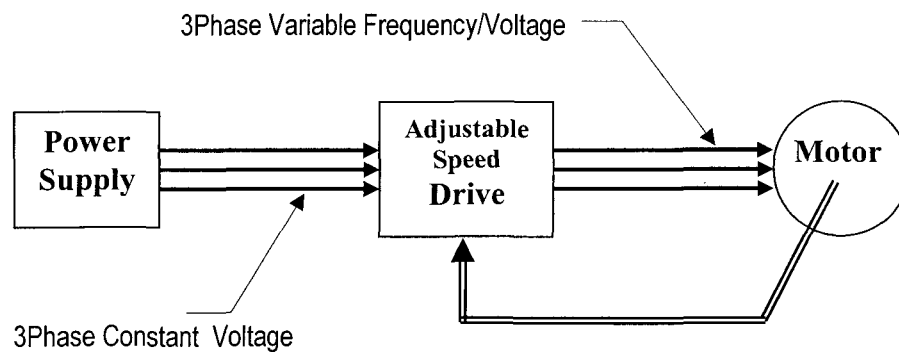
An Adjustable Speed Drive (ASD) is an electronic device connected between the source of power and the motor (Figure 0-1) for both AC and DC motors.



Although ASDs are used to change motor speed, they also may be used for other purposes, such as:

- To accelerate or decelerate machines in a smooth and predictable fashion.
- To provide motor direction reversal on a regular basis to satisfy the needs of a certain processes.
- To regulate conveyor speeds to comply with varying load requirements.

However, the most common use of ASDs is to vary the speed.



**Figure 0-1. ASD-Motor Circuit**

### 0.1.3 Adjustable Speed Drives for Induction Machines

The electromechanical equation describing the steady state behavior of an induction motor is highly non-linear, time varying and coupled [3], so the simple linear function of the DC motors ASDs can not be used for changing the speed of induction motors.

As a very important step, control schemes have been developed which provide a feasible approach of speed control to induction motors [4]. Blaschke, Hasse, and Leonhard developed a vector control theory to simplify the structure of speed control used to drive like dc motors by using coordinate transformation. Using such a controller, an induction motor behaves similar to a DC motor with an independent control of the parameters flux and torque. Around ten years later, Takashahi [5] in Japan and also Depenbrock [6] in Germany introduced Direct Torque Control theory which is another solution to be used in induction motors ASDs. This technique is manufactured by ABB [7] in mid 90s under the name of *ACS600 DTC Drive System*.

In recent years the Vector Control theory has been used in more and more works due to the progress in high speed electronic devices. The rotor speed knowledge is necessary to orient the injected stator current vector and to establish speed loop feedback in this way. Shaft position encoders and tachogenerators can be used for this aim, but using these applications has their own problems; The cost of speed sensor is in the same range as the cost of motor itself. The mounting of the speed sensor to the motor is also an obstacle in some applications [8], as the reliability and noise immunity are the other disadvantages of using the speed sensors.

Since the late 1980s, speed sensorless control methods of induction machine, in which the shaft speed is estimated (based on the measurement of stator voltage and current) instead of measuring have been studied [9]. There are

different ways for implementation a speed sensorless control system. Using the extended Kalman filter method proposed in [10] or the optimized version proposed in [11] is a solution. Though using EKF is consequent, but requires too much computation which limits the control bandwidth, and is sensitive to parameter variations [12]. In the middle of 1990s researchers started using Neural Networks in the induction motors speed control drives. In [13] a neural network was used as a part of the observer just to emulate the rotor flux in a field control system with speed encoder, as in [14] a neural network had been used to estimate the speed of the motor instead of a speed encoder, and it has been continued until now. The major advantage of using the Neural Networks is that they can estimate the nonlinear functional relationship between the inputs and outputs by learning from the training data. It's not necessary to determine the shape of the nonlinear function beforehand [15]. On the other hand, in these systems the parallel calculations helps us to overcome the time limitations, as the system has more immunity of noise and changes in the load and also motor parameters changes during work.

## **0.2 Objectives**

The main goal of this research work consists in estimating the speed of the induction motor by using the measurable variables of the motor -voltage and current- in the adaptive speed drive. Using Neural Networks to impart the

advantages above in a real time adaptive structure improves the ability of the system in simulating both of the steady state and transient responses. Online training of the system helps us to get rid of the pre-calculations required in offline training networks.

Finally, the Neural Network based speed estimator is replaced as a speed detector and the output estimated speed is used as the measured speed fed back to the control block to complete the structure of the induction motor drive. The robust speed estimation helps the drive having better performance.

### **0.3 Methodology**

The methodology upon which the present work is based consists of the steps given below:

- I. Study the induction motor modeling and simulating the chosen model in the software Matlab/Simulink on PC platform.
- II. Using the Space Vector theory in an indirect field oriented controller and simulating the whole system.
- III. Study of the sensorless speed estimation methods and designing the real-time Neural Network based estimator to be run in *Matlab/Simulink*. The conversion of the *Simulink* model to real-time simulation is done using RT-LAB<sup>TM</sup> with a focus on the issue of real-time sampling rate and high-fidelity.

## 0.4 Contribution

In this work we've achieved the application of an artificial neural network for estimating the speed of an induction motor based on delayed samples of stator currents and voltages. The main contribution of this study has been using a novel method of training the network, based on an integrator-free MRAS scheme for speed identification, without any need to pre-computations. This structure has increased the performance of the system, especially for the low speeds which is often the drawback of the usual MRAS scheme.

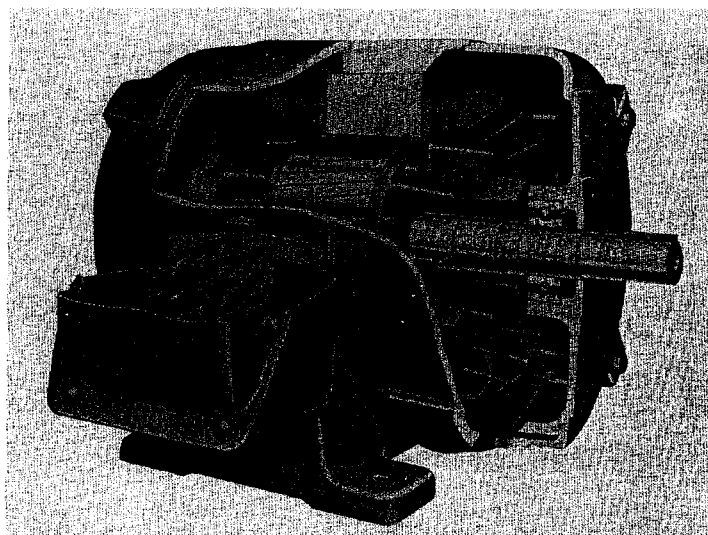
# **CHAPTER I**

## **INDUCTION MACHINE MODELING**

## 1.1 Introduction

The induction machine is the most commonly used of all rotating electrical machines. It is seldom desirable as a generator, but many types of motors are used in a wide variety of applications – from fractional horsepower two-phase control motors, to 45000 horsepower polyphase motors for wind tunnels, and tremendous numbers of induction motors are used in many home applications. [16]

The induction motor normally has a stator winding that is excited from an external a-c source. Its rotor usually consists of a laminated structure with skewed slots in which a conducting material has been cast, which produces a solid, short-circuited cylinder commonly called a squirrel-cage rotor. For special purposes the rotor can be wound, and when constructed in this fashion, the windings are insulated from the slots and connected to slip rings. Figure 1-1 is representative of



**Figure 1-1.** A three-phase squirrel-cage induction motor

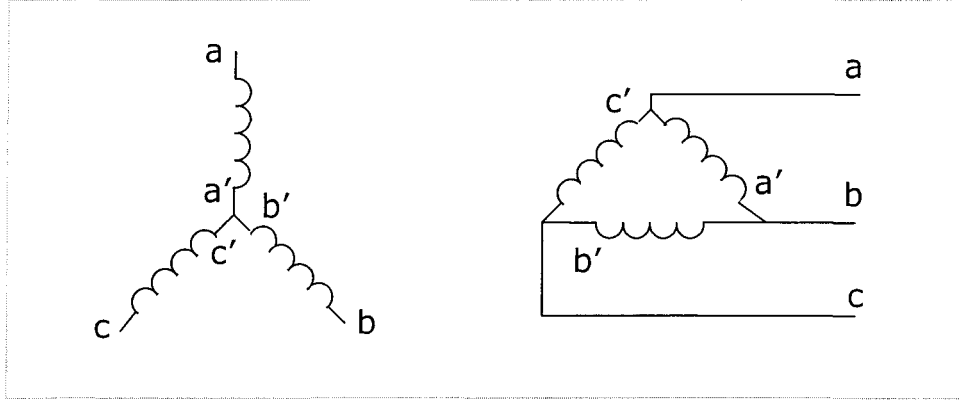
an integral horsepower, three-phase squirrel-cage induction motor.

## 1.2 Induction Machine Construction

Induction motors consist of two basic electrical assemblies: the wound stator and the rotor assembly. The induction motor derives its name from currents flowing in the secondary member (rotor) that are induced by alternating currents flowing in the primary member (stator). The combined electromagnetic effects of the stator and rotor currents produce the force to create rotation.

A three-phase winding is put in slot cuts on the inner surface of the stator, while the rotor may be any of two types of *squirrel-cage* or *wound-rotor*. The first type consists of Aluminum or Copper bars embedded in the rotor slots, shorten at the ends. The wound-rotor winding ones have the form of the stator winding. It is obvious that squirrel-cage induction machine is simpler, more economical and more rugged than the wound-rotor induction machines. The axes of the coils are  $120^\circ$  (electrical) apart. The ends of the three-phase may be connected in a wye (Y) or a delta ( $\Delta$ ) form as shown in figure 1-2 [17]. Unlike dc machines, induction machines have a uniform air gap. If balanced three-phase current flow through these windings, a rotating magnitude field of constant amplitude and speed will be produced in the air gap and will induce current in the rotor circuit to produce torque.





**Figure 1-2.** Y-connected stator (left) and  $\Delta$ -connected stator (right)

### 1.3 Current Equations

When a current flows through a phase coil, produces a sinusoidally distributed mmf wave, whose amplitude and direction depend on the instantaneous value of that current. The three phases would produce similar sinusoidally mmf waves displaced by  $2\pi/3$  electrical radians in space from each other, so a balanced three-phase current flowing will present

$$i_a = I_m \cos(\omega t) \quad (1-1)$$

$$i_b = I_m \cos(\omega t - 2\pi/3) \quad (1-2)$$

$$i_c = I_m \cos(\omega t + 2\pi/3) \quad (1-3)$$

Flowing through the respective phase windings, each of these currents produces a sinusoidally distributed mmf wave in the space, having a peak located along its axis. So, each mmf wave could be presented by a space vector along the axis of its phase with magnitude proportional to the instantaneous value of the

current. The angular velocity of the rotating wave would be equal to  $2\pi f$  radian per seconds, so the revolutions per minute (rpm)  $n$  of the traveling wave in a  $p$ -pole machine for a frequency  $f$  cycle per seconds for the current would be

$$n = \frac{2}{p} f \frac{\pi}{3} = \frac{2\pi f}{3p} \quad (1-4)$$

## 1.4 Induced Voltages

The rotating magnetic field in the air gap of the machine discussed before in section 1.1 induces voltages in the phase coils. Expression for these induced voltages can be obtained from Faraday's law of induction.

Considering the phase coils as full-pitch coils of  $N$  turns, the flux linkage will be maximum at zero radians and equal to zero at  $\pi/2$  radians. The flux linkage ( $\psi_a$ ) will vary, following the equation 1-5:

$$\psi_a(\omega t) = N\phi_p \cos(\omega t) \quad (1-5)$$

where  $\Phi_p$  is the flux linking each phase. Therefore, the voltage induced in phase coil  $aa'$  is obtained from Faraday's law as:

$$e_a(\omega t) = \omega N\phi_p \sin(\omega t) \quad (1-6)$$

Thus, the voltages induced in the three phase coils would be:

$$e_a = E_m \sin(\omega t) \quad (1-7)$$

$$e_b = E_m \sin(\omega t - 2\pi/3) \quad (1-8)$$

$$e_c = E_m \sin(\omega t + 2\pi / 3) \quad (1-9)$$

And the rms value of the induced voltage is:

$$E_{rms} = 4.44 f N \phi_p \quad (1-10)$$

## 1.5 Running Operation

### 1.5.1 Slip Speed

The speed at which the magnetic field rotates is the synchronous speed of the motor and is determined by the number of poles in the stator and the frequency of the power supply:

$$N_s = \frac{120 f_s}{P} \quad (1-11)$$

in the equations above  $N_s$  stands for synchronous speed,  $f$  is the frequency and  $P$  is the number of the poles.

Synchronous speed is the absolute upper limit of motor speed. If the rotor turns exactly as fast as the rotating magnetic field, then no lines of force are cut by the rotor conductors, and torque is zero. When running, the rotor always rotates slower than the magnetic field. The rotor speed is just slow enough to cause the proper amount of rotor current to flow, so that the resulting torque is sufficient to overcome windage and friction losses, and drive the load [18]. The difference in rotor speed versus stator speed in a motor is called *slip*, which is given by

$$S = N_s - N_r \quad (1-12)$$

defining  $g$  as given in (1-13), the slip could be presented as in equation (1-14)

$$g = \frac{N_s - N_r}{N_s} \quad (1-13)$$

$$S = gN_s \quad (1-14)$$

The frequency of the rotor currents is also different from that of the stator currents and is given by

$$f_r = gf_s \quad (1-15)$$

this frequency would be zero for the blocked rotor, and will be equal to the frequency of stator in the synchronous speed. Hence, the asynchronous machine could be called a frequency converter.

### 1.5.2 Electrical model equivalent

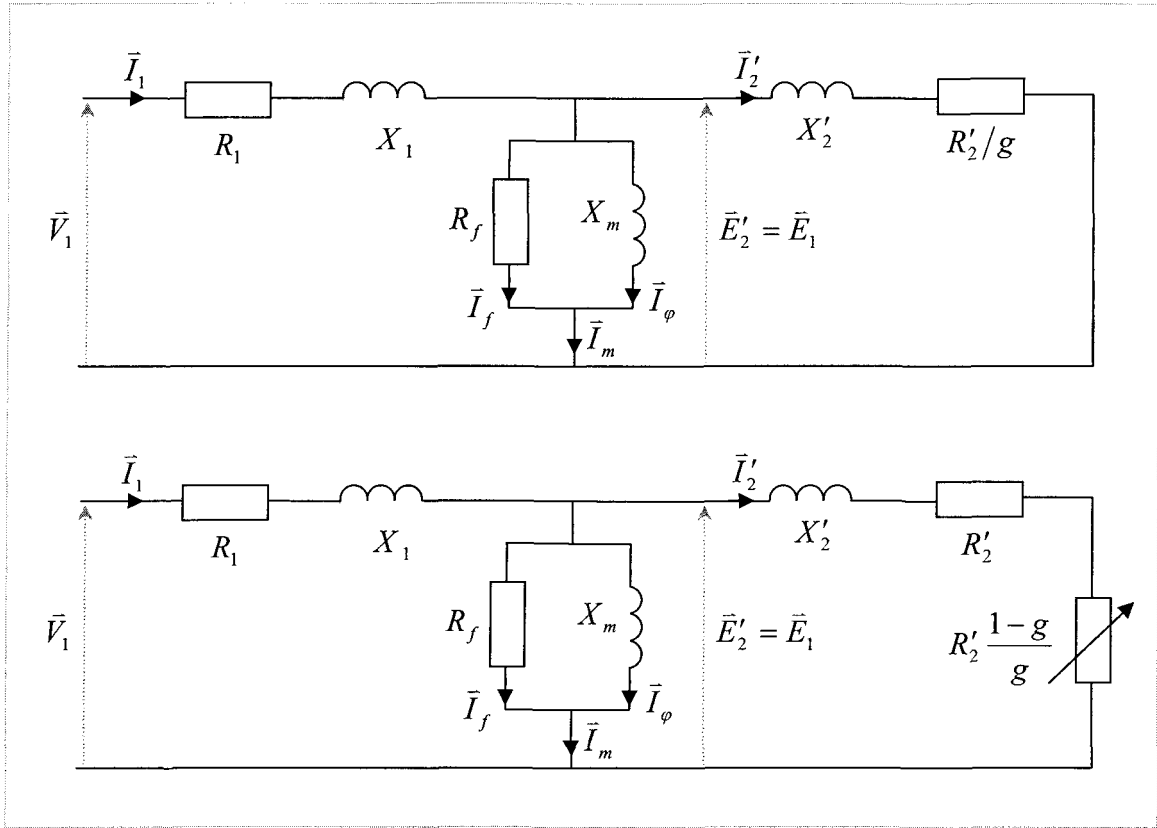
The electrical model equivalent schema for the asynchronous machine is given [19] in figure 1-3. The electromagnetic torque of the machine is obtained from (1-16)

$$T_{em} = \frac{P_{em}}{\omega_s} \quad (1-16)$$

Or using the recent parameters

$$T_{em} = \frac{3}{\omega_s} \frac{R'_2}{g} (I'_2)^2 \quad (1-17)$$

From the equivalent schema of the induction machine given above, the Thevenin equivalent circuit could be drawn as in figure 1-3 in which



**Figure 1-3.** Induction Machine Equivalent Circuits

$$\vec{V}_{th} = \vec{V}_1 \frac{X_m}{\sqrt{R_1^2 + (X_1 + X_m)^2}} \quad (1-18)$$

In the case of  $R_1^2 \ll (X_1 + X_m)^2$ ,

$$\vec{V}_{th} \approx \frac{X_m}{X_1 + X_m} \vec{V}_1 = K_{th} V_1 \quad (1-19)$$

The Thevenin impedance is

$$Z_{th} = \frac{jX_m(R_1 + jX_1)}{R_1 + j(X_1 + X_m)} \quad (1-20)$$

If  $R_1^2 \ll (X_1 + X_m)^2$ ,

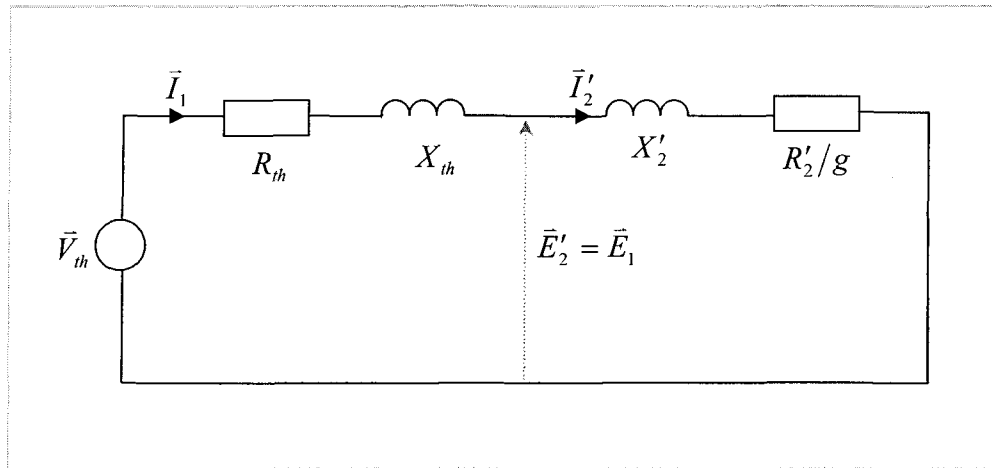
$$R_{th} \approx \left( \frac{X_m}{X_1 + X_m} \right)^2 R_1 = K_{th} R_1 \quad (1-21)$$

and since  $X_1 \ll X_m$ ,

$$X_{th} \approx X_1 \quad (1-22)$$

So, the rotor current is obtained from (1-23)

$$\bar{I}'_2 = \bar{V}_{th} \frac{1}{\left( R_{th} + \frac{R'_2}{g} \right) + j(X_{th} + X'_2)} \quad (1-23)$$



**Figure 1-4.** Thevenin Equivalent Circuits of Machine

## 1.6 Representation of Three-Phase Induction Machine

### 1.6.1 Representation of three-phase variables

The extensive amount of coupling between the six circuits makes the analysis of the machine a formidable task and a way to simplify this helps the analysis. The

approach to simplification involves the topic of “*Reference frame theory*” and constitutes an essential aspect of machine analysis [20].

Considering a balanced three-phase current flowing through the three-phase windings with the equations for the currents given in (1-1) to (1-3), defining the parameter  $a$  as:

$$a = e^{j2\pi/3} \quad (1-24)$$

we will have:

$$a^2 = e^{j4\pi/3} = e^{-j2\pi/3} = a^* \quad (1-25)$$

where  $*$  denotes the complex conjugate.

Now if we define:

$$\vec{I}_{s,abc} = \sqrt{\frac{2}{3}}(i_{sa} + a.i_{sb} + a^2.i_{sc}) \quad (1-26)$$

then:

$$\vec{I}_{s,abc}^* = \vec{I}_{s,abc}^{-1} = \sqrt{\frac{2}{3}}(i_{sa} + a^2.i_{sb} + a.i_{sc}) \quad (1-27)$$

Quantities defined in the manner of equation (1-26) are called *complex space vector* which should not be confused with *complex phasors* which are normally used to represent sinusoidally alternating quantities in stationary states.

### 1.6.2 Complex variable model of three-phase induction motor

The voltage equations for the stator and rotor of the induction motor is given below

$$\vec{V}_{s,abc} = R_s \vec{I}_{s,abc} + \frac{d\vec{\psi}_{s,abc}}{dt} \quad (1-28)$$

$$\vec{V}_{r,abc} = R_r \vec{I}_{r,abc} + \frac{d\vec{\psi}_{r,abc}}{dt} = 0 \quad (1-29)$$

in the equations above

$$\begin{bmatrix} \psi_{sa} \\ \psi_{sb} \\ \psi_{sc} \\ \psi_{ra} \\ \psi_{rb} \\ \psi_{rc} \end{bmatrix} = \begin{bmatrix} L_s & M_s & M_s & M_{sr} \cos \theta_r & M_{sr} \cos \theta_r^- & M_{sr} \cos \theta_r^+ \\ M_s & L_s & M_s & M_{sr} \cos \theta_r^+ & M_{sr} \cos \theta_r & M_{sr} \cos \theta_r^- \\ M_s & M_s & L_s & M_{sr} \cos \theta_r^- & M_{sr} \cos \theta_r^+ & M_{sr} \cos \theta_r \\ M_{sr} \cos \theta_r & M_{sr} \cos \theta_r^+ & M_{sr} \cos \theta_r^- & L_r & M_r & M_r \\ M_{sr} \cos \theta_r^- & M_{sr} \cos \theta_r & M_{sr} \cos \theta_r^+ & M_r & L_r & M_r \\ M_{sr} \cos \theta_r^+ & M_{sr} \cos \theta_r^- & M_{sr} \cos \theta_r & M_r & M_r & L_r \end{bmatrix} \begin{bmatrix} I_{sa} \\ I_{sb} \\ I_{sc} \\ I_{ra} \\ I_{rb} \\ I_{rc} \end{bmatrix} \quad (1-30)$$

in which

$$\theta_r^- = \theta_r - \frac{2\pi}{3} \quad (1-31)$$

$$\theta_r^+ = \theta_r + \frac{2\pi}{3} \quad (1-32)$$

Considering the representation of machine equations in complex variable form, by multiplying 2<sup>nd</sup> and 3<sup>rd</sup> row of expanded form of equations by  $\underline{a}$  and  $\underline{a}^2$  respectively --like as in section 1.6.1-- we'll obtain the general form

$$\vec{f}_{s,abc} = \sqrt{\frac{2}{3}} (f_{sa} + \underline{a} f_{sb} + \underline{a}^2 f_{sc}) \quad (1-33)$$

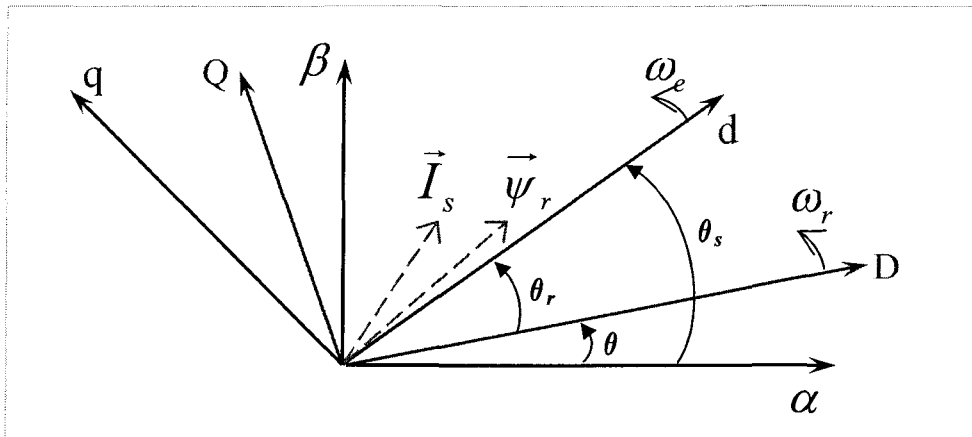
that  $f$  is any of  $V, I, \psi$ .



## 1.7 Space Vector Representation of Three-Phase Induction Machine

### 1.7.1 The three reference frames

While the complex vector approach to writing the machine equations results in a compact form, the essential sinusoidal coupling between the stator and rotor circuits with rotor position remain [20]. This coupling can be eliminated if these two series of equations are referred to a common frame. This frame could be non-rotating which is called stationary (stator) reference. Alternatively, the frame can be made to rotate with the same angular velocity as the rotor circuits which is called rotor reference frame. Also, one of the complex vectors could be taken to rotate these axes themselves synchronously with. This case is called arbitrary (freely rotating) reference frame. The three different frames are represented in figure 1-5.



**Figure 1-5.** Stator reference frame ( $\alpha, \beta$ ), rotor reference frame ( $D, Q$ ) and arbitrary reference frame ( $d, q$ )

Variables along the  $a, b, c$  stator axes in figure 1-2 can be referred to the  $d, q$  axes as shown below in the figure 1-5.

Transformation from the frame  $a, b, c$  to  $d, q$  is done via

$$\begin{bmatrix} f_d \\ f_q \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos \theta & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) \\ -\sin \theta & -\sin(\theta - \frac{2\pi}{3}) & -\sin(\theta + \frac{2\pi}{3}) \end{bmatrix} \begin{bmatrix} f_a \\ f_b \\ f_c \end{bmatrix} \quad (1-34)$$

on the contrary

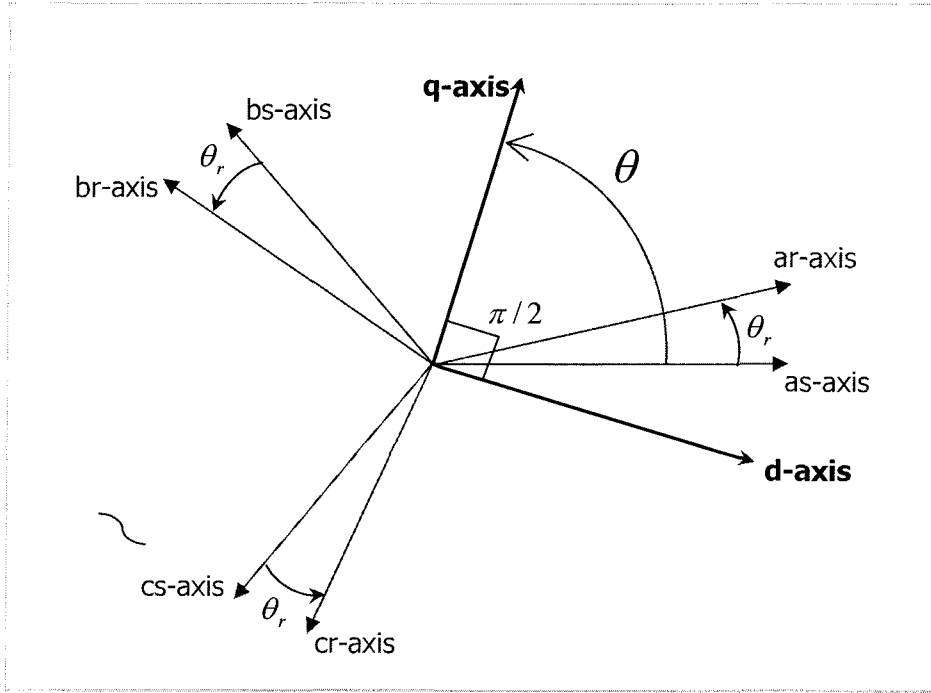
$$\begin{bmatrix} f_a \\ f_b \\ f_c \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos \theta & -\sin \theta \\ \cos(\theta - \frac{2\pi}{3}) & -\sin(\theta - \frac{2\pi}{3}) \\ \cos(\theta + \frac{2\pi}{3}) & -\sin(\theta + \frac{2\pi}{3}) \end{bmatrix} \begin{bmatrix} f_d \\ f_q \end{bmatrix} \quad (1-35)$$

Transformation from the frame  $a, b, c$  to  $\alpha, \beta$  is done via

$$\begin{bmatrix} f_\alpha \\ f_\beta \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} f_a \\ f_b \\ f_c \end{bmatrix} \quad (1-36)$$

and on the contrary

$$\begin{bmatrix} f_a \\ f_b \\ f_c \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ \frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} f_\alpha \\ f_\beta \end{bmatrix} \quad (1-37)$$



**Figure 1-6.** Location of reference frame axes relative to magnetic axes

### 1.7.2 Induction motor model in arbitrary reference frame

Specifying stator variables and rotor variables referred to the rotor frame with  $ss$  and  $sr$  subscripts respectively, stator variables and rotor variables referred to the rotor frame with  $sr$  and  $rr$  subscripts respectively, and the stator variables and rotor variables referred to the arbitrary frame with  $s$  and  $r$  respectively, the converting relations are

$$f_s = f_{ss} \exp(-j\theta_s) \quad (1-38)$$

$$f_r = f_{rr} \exp(-j\theta_s) \quad (1-39)$$

From (1-28) and (1-38)

$$\vec{V}_s e^{j\theta_s} = R_s \vec{I}_s e^{j\theta_s} + \frac{d}{dt} (\vec{\psi}_s e^{j\theta_s}) \quad (1-40)$$

and so

$$\vec{V}_s = R_s \vec{I}_s + \frac{d\vec{\psi}_s}{dt} + j \frac{d\theta_s}{dt} \vec{\psi}_s \quad (1-41)$$

Similarly from (1-29) and (1-39)

$$\vec{V}_r e^{j\theta_r} = R_r \vec{I}_r e^{j\theta_r} + \frac{d}{dt} (\vec{\psi}_r e^{j\theta_r}) \quad (1-42)$$

and so

$$\vec{V}_r = R_r \vec{I}_r + \frac{d\vec{\psi}_r}{dt} + j \frac{d\theta_r}{dt} \vec{\psi}_r \quad (1-43)$$

Recalling figure 1-5

$$\theta_r = \theta_s - \theta \quad (1-44)$$

which gives

$$\frac{d\theta_r}{dt} = \frac{d\theta_s}{dt} - \frac{d\theta}{dt} = \omega_e - \omega_r \quad (1-45)$$

So, in the arbitrary frame the voltage equations for the stator and rotor could be rewritten from (1-41), (1-43) and (1-45) as below

$$\vec{V}_s = R_s \vec{I}_s + \frac{d\vec{\psi}_s}{dt} + j \omega_e \vec{\psi}_s \quad (1-46)$$

$$\vec{V}_r = R_r \vec{I}_r + \frac{d\vec{\psi}_r}{dt} + j(\omega_e - \omega_r) \vec{\psi}_r \quad (1-47)$$

### 1.7.3 Induction motor model in stator reference frame

As in the stator frame  $\theta_s$  equals to zero,

$$\vec{V}_s = R_s \vec{I}_s + \frac{d\vec{\psi}_s}{dt} \quad (1-48)$$

$$\vec{V}_{rs} = R_r \vec{I}_{rs} + \frac{d\vec{\psi}_{rs}}{dt} - j\omega_r \vec{\psi}_{rs} \quad (1-49)$$

#### 1.7.4 Induction motor model in rotor reference frame

In rotor frame  $\theta_r$  is equal to zero, and then the speeds  $\omega_s$  and  $\omega_r$  would be the same and

$$\vec{V}_{sr} = R_s \vec{I}_{sr} + \frac{d\vec{\psi}_{sr}}{dt} + j\omega_r \vec{\psi}_{sr} \quad (1-50)$$

$$\vec{V}_r = R_r \vec{I}_r + \frac{d\vec{\psi}_r}{dt} \quad (1-51)$$

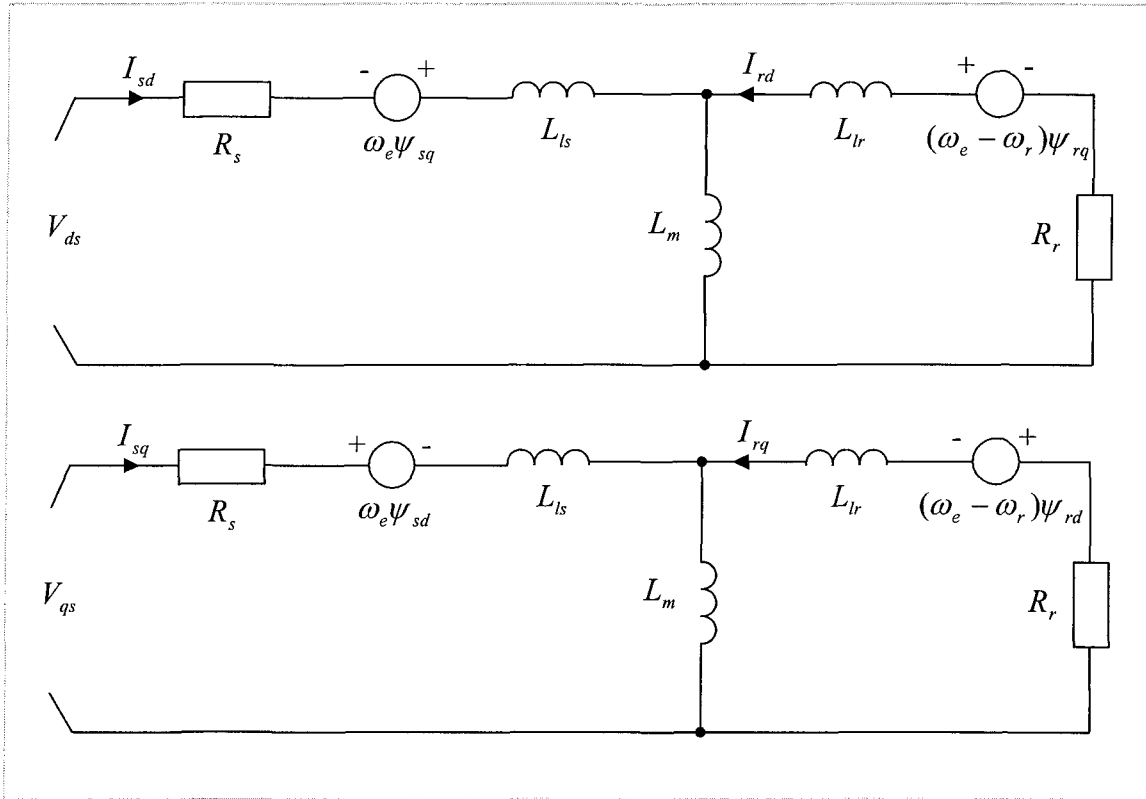
### 1.8 Two-Phase Induction Machines

Two-phase induction machines are widely used as single-phase induction motors in home and industrial applications. The analysis of the unsymmetrical two-phase machines has been accomplished by employing revolving-field theory [21] in the past, but reference-frame theory analysis could be used to analyze this group of machines essentially the same as of the three-phase ones.

Considering the model of the machine in a  $d,q$  frame with rotor flux and the stator current like variables of state

$$\frac{d}{dt} \begin{bmatrix} I_{sd} \\ I_{sq} \\ \psi_{rd} \\ \psi_{rq} \end{bmatrix} = A(\omega_e, \omega_r) \begin{bmatrix} I_{sd} \\ I_{sq} \\ \psi_{rd} \\ \psi_{rq} \end{bmatrix} + B \begin{bmatrix} V_{sd} \\ V_{sq} \end{bmatrix} \quad (1-52)$$

The induction machine equivalent circuits in an arbitrary synchronously rotating  $d,q$  axis reference frame are shown in Figure 1-7 [22] where  $\omega_e$  is the synchronous frequency of the reference frame, so for the stationary frame model it's equal to zero.



**Figure 1-7** Induction Machine Synchronous  $d,q$  Axis Equivalent Circuits

Expanding (1-52) by using the equations for the variables current and flux linkage and adding the variable of speed, the model of the machine in arbitrary frame could be presented by

$$\frac{d}{dt} \begin{bmatrix} I_{sd} \\ I_{sq} \\ \psi_{rd} \\ \psi_{rq} \\ \omega_r \end{bmatrix} = \begin{bmatrix} -\left(\frac{R_s}{\sigma L_s} + \frac{R_r(1-\sigma)}{\sigma L_r}\right) I_{sd} + \omega_e I_{sq} + \frac{R_r L_m}{\sigma L_s L_r^2} \psi_{rd} + \frac{L_m \omega_r}{\sigma L_s L_r} \psi_{rq} \\ -\omega_e I_{sd} - \left(\frac{R_s}{\sigma L_s} + \frac{R_r(1-\sigma)}{\sigma L_r}\right) I_{sq} - \frac{L_m \omega_r}{\sigma L_s L_r} \psi_{rd} + \frac{R_r L_m}{\sigma L_s L_r^2} \psi_{rq} \\ \frac{R_r L_m}{L_r} I_{sd} - \frac{R_r}{L_r} \psi_{rd} + (\omega_e - \omega_r) \psi_{rq} \\ \frac{R_r L_m}{L_r} I_{sq} - (\omega_e - \omega_r) \psi_{rd} - \frac{R_r}{L_r} \psi_{rq} \\ \frac{p^2 L_m}{J L_r} (I_{sq} \psi_{rd} - I_{sd} \psi_{rq}) - \frac{F}{J} \omega_r - \frac{p}{J} T_r \end{bmatrix} + \frac{1}{\sigma L_s} \begin{bmatrix} V_{sd} \\ V_{sq} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (1-53)$$

in which  $\sigma$  is the coefficient of dispersion,  $\omega_e$  is the synchronous frequency of the reference frame,  $p$  is the number of the pole pair,  $F$  is the damping coefficient and  $J$  is the total rotor inertia.

In rotor frame,  $\omega_e$  is equal to  $\omega_r$  and so the model of induction motor is given by

$$\frac{d}{dt} \begin{bmatrix} I_{sD} \\ I_{sQ} \\ \psi_{rD} \\ \psi_{rQ} \\ \omega_r \end{bmatrix} = \begin{bmatrix} -\left(\frac{R_s}{\sigma L_s} + \frac{R_r(1-\sigma)}{\sigma L_r}\right) I_{sD} + \omega_r I_{sQ} + \frac{R_r L_m}{\sigma L_s L_r^2} \psi_{rD} + \frac{L_m \omega_r}{\sigma L_s L_r} \psi_{rQ} \\ -\omega_r I_{sD} - \left(\frac{R_s}{\sigma L_s} + \frac{R_r(1-\sigma)}{\sigma L_r}\right) I_{sQ} - \frac{L_m \omega_r}{\sigma L_s L_r} \psi_{rD} + \frac{R_r L_m}{\sigma L_s L_r^2} \psi_{rQ} \\ \frac{R_r L_m}{L_r} I_{sD} - \frac{R_r}{L_r} \psi_{rD} \\ \frac{R_r L_m}{L_r} I_{sQ} - \frac{R_r}{L_r} \psi_{rQ} \\ \frac{p^2 L_m}{J L_r} (I_{sQ} \psi_{rD} - I_{sD} \psi_{rQ}) - \frac{F}{J} \omega_r - \frac{p}{J} T_r \end{bmatrix} + \frac{1}{\sigma L_s} \begin{bmatrix} V_{sD} \\ V_{sQ} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (1-54)$$

And finally in stator frame,  $\omega_e$  is equal to zero and the model of induction motor is given by

$$\frac{d}{dt} \begin{bmatrix} I_{s\alpha} \\ I_{s\beta} \\ \psi_{r\alpha} \\ \psi_{r\beta} \\ \omega_r \end{bmatrix} = \begin{bmatrix} -\left(\frac{R_s}{\sigma L_s} + \frac{R_r(1-\sigma)}{\sigma L_r}\right) I_{s\alpha} + \frac{R_r L_m}{\sigma L_s L_r^2} \psi_{r\alpha} + \frac{L_m \omega_r}{\sigma L_s L_r} \psi_{r\beta} \\ -\left(\frac{R_s}{\sigma L_s} + \frac{R_r(1-\sigma)}{\sigma L_r}\right) I_{s\beta} - \frac{L_m \omega_r}{\sigma L_s L_r} \psi_{r\alpha} + \frac{R_r L_m}{\sigma L_s L_r^2} \psi_{r\beta} \\ \frac{R_r L_m}{L_r} I_{s\alpha} - \frac{R_r}{L_r} \psi_{r\alpha} - \omega_r \psi_{r\beta} \\ \frac{R_r L_m}{L_r} I_{s\beta} + \omega_r \psi_{r\alpha} - \frac{R_r}{L_r} \psi_{r\beta} \\ \frac{p^2 L_m}{J L_r} (I_{s\beta} \psi_{r\alpha} - I_{s\alpha} \psi_{r\beta}) - \frac{F}{J} \omega_r - \frac{p}{J} T_r \end{bmatrix} + \frac{1}{\sigma L_s} \begin{bmatrix} V_{s\alpha} \\ V_{s\beta} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (1-55)$$



**CHAPTER 2**

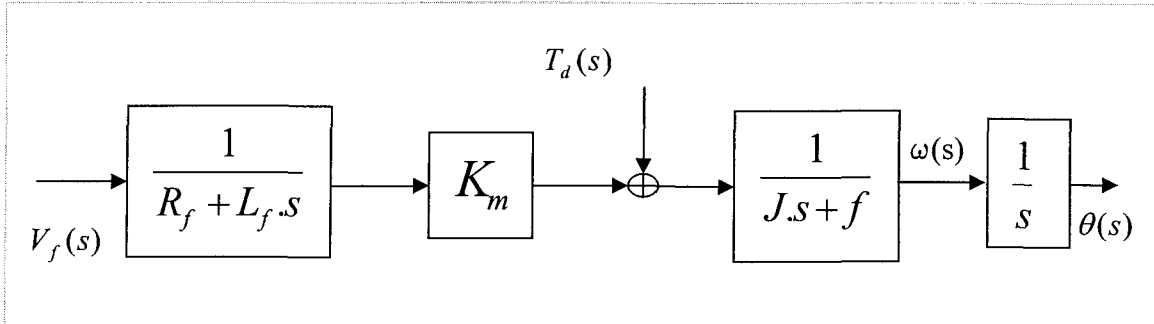
**VECTOR CONTROL OF**

**INDUCTION MACHINE**

## 2.1 Introduction

In earlier years, DC machines used to be the first choice of use in variable speed uses in industry due to their ease of control, and the majority of induction machines were used in constant speed drivers or limited speed range applications; because the transfer function of DC machines, and independence of torque and flux let us control the speed as field-controlled or armature-controlled with a linear control system [23]:

$$\dot{X} = A.X + B.U \quad (2-1)$$



**Figure 2-1.** Block Diagram of a field controlled DC machine

On the other hand, induction machines are robust, economical, and low maintenance which gives them outstanding potential for helping to fulfill the world's needs, but the non-linear behaviors of them make the control process more complicated. A useful form of representing the machine's electrical model is the state-space model, which is a linearized form of the model. It's assumed that the

mechanical dynamics are slow in comparison to the electrical dynamics; hence the speed is treated as a time-varying parameter in the model. Thus the model is split to mechanical and electrical subsystem. The state space form of the induction machine could be written similar to what is presented for dc machines as in below, though it's easier to understand if we use the form which is already presented in section 1-8.

$$\begin{cases} \dot{X}(t) = A(t).X(t) + B.u(t) \\ y(t) = C.X(t) \end{cases} \quad (2-2)$$

where

$$A(t) = \begin{bmatrix} -\left(\frac{R_s}{\sigma L_s} + \frac{R_r L_m^2}{\sigma L_s L_r^2}\right) & 0 & \frac{R_r L_m}{\sigma L_s L_r^2} & \frac{L_m}{\sigma L_s L_r} \omega_r(t) \\ 0 & -\left(\frac{R_s}{\sigma L_s} + \frac{R_r L_m^2}{\sigma L_s L_r^2}\right) & -\frac{L_m}{\sigma L_s L_r} \omega_r(t) & \frac{R_r L_m}{\sigma L_s L_r^2} \\ R_r \frac{L_m}{L_r} & 0 & -\frac{R_r}{L_r} & -\omega_r(t) \\ 0 & R_r \frac{L_m}{L_r} & -\omega_r(t) & -\frac{R_r}{L_r} \end{bmatrix} \quad (2-3)$$

$$X(t) = \begin{bmatrix} i_{sd} \\ i_{sq} \\ \psi_{rd} \\ \psi_{rq} \end{bmatrix} \quad (2-4)$$

$$B(t) = \begin{bmatrix} \frac{1}{\sigma L_s} & 0 \\ 0 & \frac{1}{\sigma L_s} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (2-5)$$

$$u(t) = \begin{bmatrix} v_{sd} \\ v_{sq} \end{bmatrix} \quad (2-6)$$

$$y(t) = \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} \quad (2-7)$$

$$C = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix} \quad (2-8)$$

## 2.2 Induction Machine Control Techniques

An induction machine is essentially a constant-speed machine as long as connected to a constant voltage and constant frequency power supply. In this case the operating speed is very close to the synchronous speed, and changes in the torque makes small changes in the speed. It is suitable for use in constant-speed drive systems, while in variable speed uses the controllers are more complicated than the traditional dc motor speed controllers. There are various methods for controlling the speed of an induction motor which are shortly discussed in the following.

### 2.2.1 Pole Changing

The first simple control techniques, could be considered as the *star-delta starters* (to limit the starting current) for fixed speed drivers and *pole changers* and *stator-tap changers* for multi speed drives [24][25]. The introduction of static power converters has allowed the use of more sophisticated control techniques,

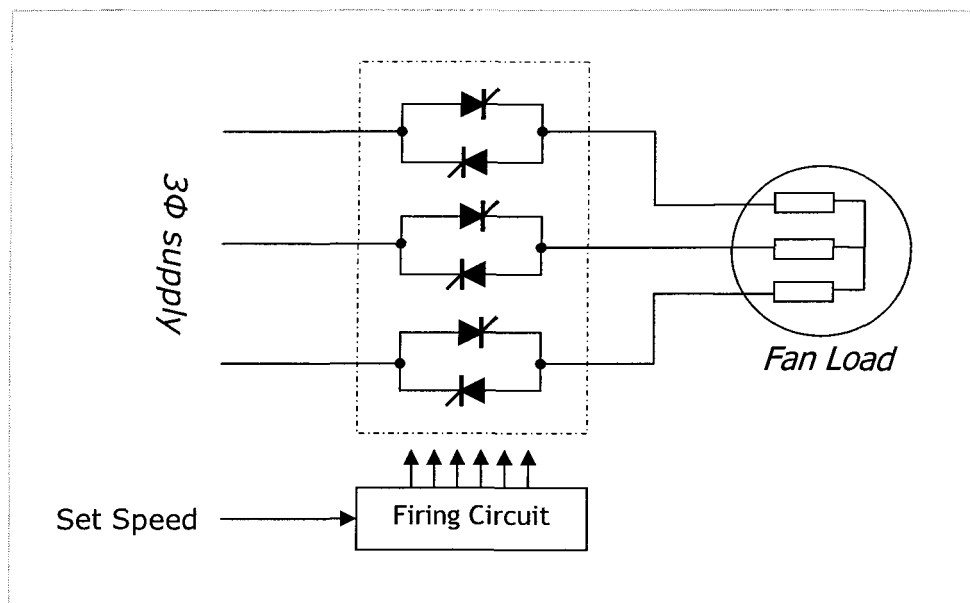
which let the induction machines operate in a wide speed range. As the operating speed is close to the synchronous speed, the speed of an induction machine could be changed by changing the number of the poles regarding equation (1-11) which is rewritten here as (2-9)

$$N_s = \frac{120f_s}{P} \quad (2-9)$$

This can be done by changing the coil connections of the stator winding. It's obvious that in this method the speed could be changed only in discrete steps.

### 2.2.2 Line Voltage Control

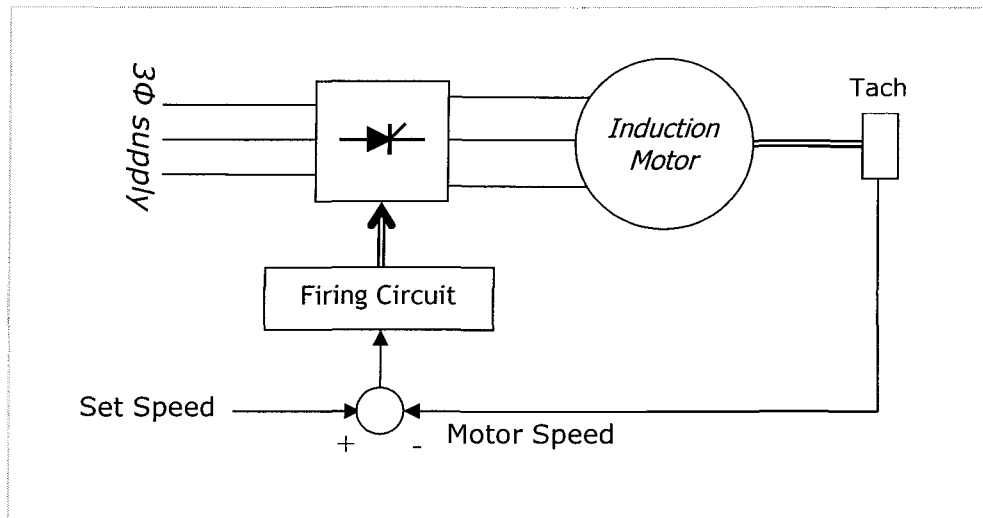
This method is useful for small motors with fan loads. The torque developed in an induction machine is proportional to the square of the terminal



**Figure 2-2.** Open-loop Voltage Controller

voltage. In figure 2-2 firing the thyristors at a particular firing angle, provides a particular terminal voltage for the motor.

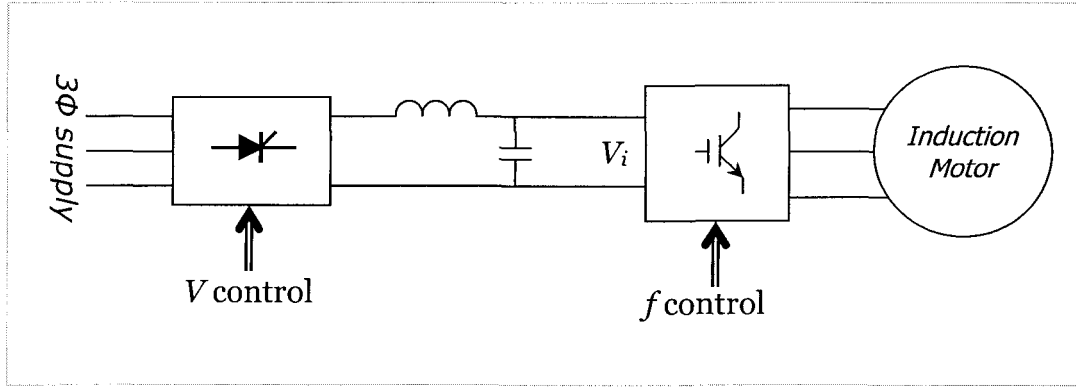
The open-loop controller may be changed to a closed-loop controller (figure 2-3) if a precise speed control is desired. In such a controller, the difference between the set speed and the motor speed would correct the firing angle of the thyristors. The voltage controllers are simple, although inefficient for all purposes.



**Figure 2-3.** Closed-loop Voltage Controller

### 2.2.3 Line Frequency Control

Figure 2-4 shows the block diagram of an open-loop speed controller in which the supply frequency of the induction motor is varied. Recall the equations (1-5) and (1-6) presenting the flux linkage and the coil voltages for each phase which were used to find the *rms* value of the induced voltage as given in (1-10)



**Figure 2-4.** Open-loop Input Voltage & Frequency Controller

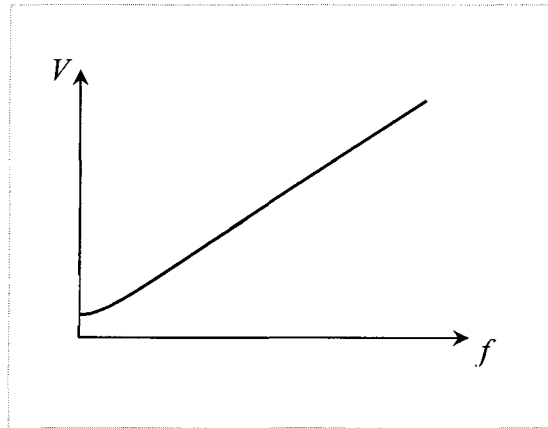
which is repeated as (2-10)

$$E_{rms} = \frac{\omega N \phi_M}{\sqrt{2}} = 4.44 f N \phi_M \quad (2-10)$$

Now in the motor model (figure 1-3), if the drop of voltage on  $R_l$  and  $X_l$  is small comparing to the terminal voltage –which happens in higher frequencies- it's possible to conclude from the recent equation that

$$\phi_P = k \frac{V_m}{f} \quad (2-11)$$

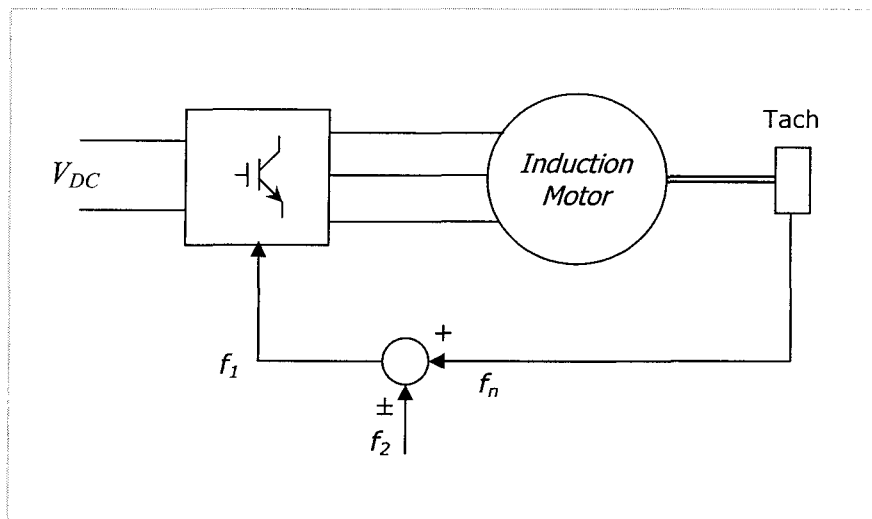
As the drop voltage on  $R_l$  and  $X_l$  is comparable to the terminal voltage the equation (2-11) is not valid any more. The ratio of  $V/f$  could be increased as shown in figure 2-5 to maintain this drawback.



**Figure 2-5.** Required Voltage Variation in V/f controller

#### 2.2.4 Constant-Slip Frequency Control

In the figure 2-6 the signal  $f_n$  represents a frequency related to the motor speed. Consider a signal of  $f_2$  added (or subtracted) to the rotor frequency ( $f_n$ , the same slip frequency) so that the stator frequency ( $f_1$ ) is kept constant.



**Figure 2-6.** Constant Slip Controller

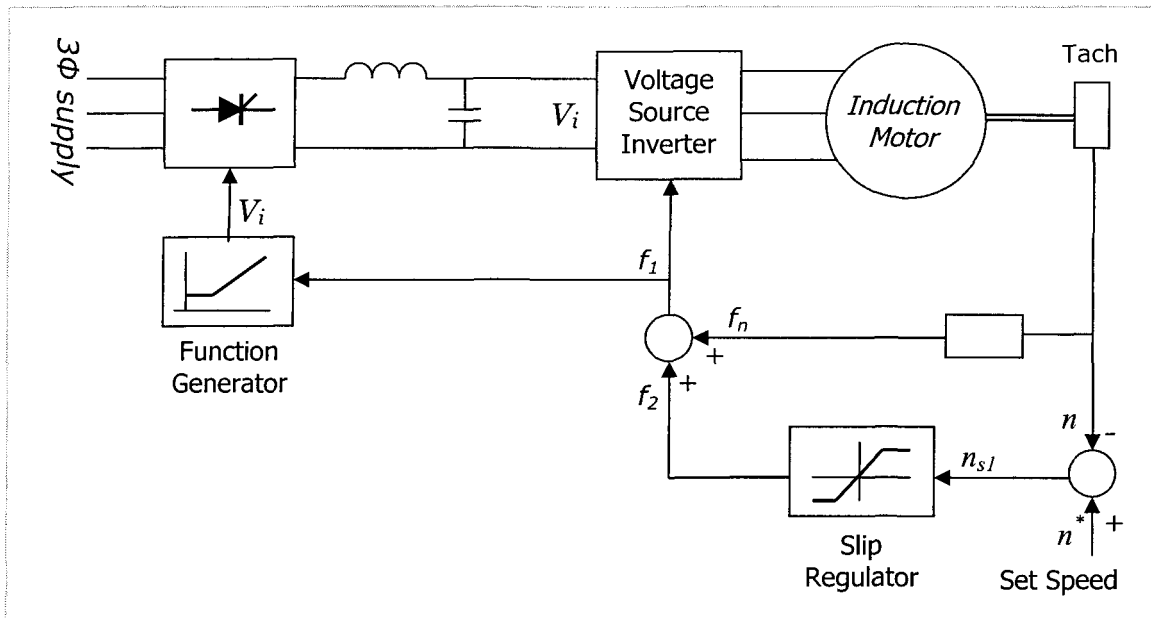


## 2.2.5 Closed-Loop Control

Industrial drives mostly operate as closed-loop feedback systems. Different structures exist for such a controller.

### 2.2.5.1 Constant V/f Operation

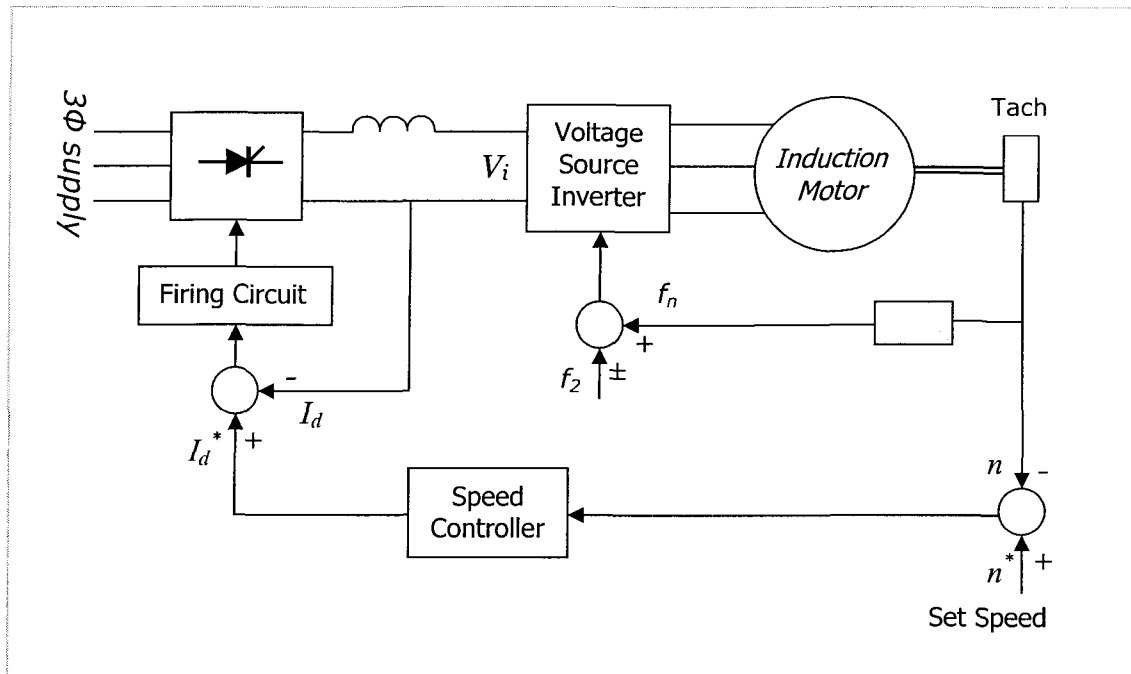
Figure 2-7 shows the block diagram of a *constant Volt/Hertz* operation in which the difference between the set speed ( $n_r^*$ ) and the actual speed ( $n_r$ ) represents the slip speed, hence the slip frequency. This slip frequency is added to the  $f_n$  (which represents the motor speed as pointed in section 2.2.4) to generate the stator frequency  $f_1$ . The function generator provides a signal such as a voltage obtained to keep the  $V/f$  as a constant.



**Figure 2-7.** Closed-loop Speed Control Employing Slip Frequency Regulation and Constant V/f Operation

### 2.2.5.2 Constant Slip Frequency

Another possibility for a simple speed control system is to keep the slip frequency constant and control the speed by controlling the dc link current  $I_d$  and hence the magnitude flux of the motor, using a current source inverter as is shown in the block diagram of figure 2-8.

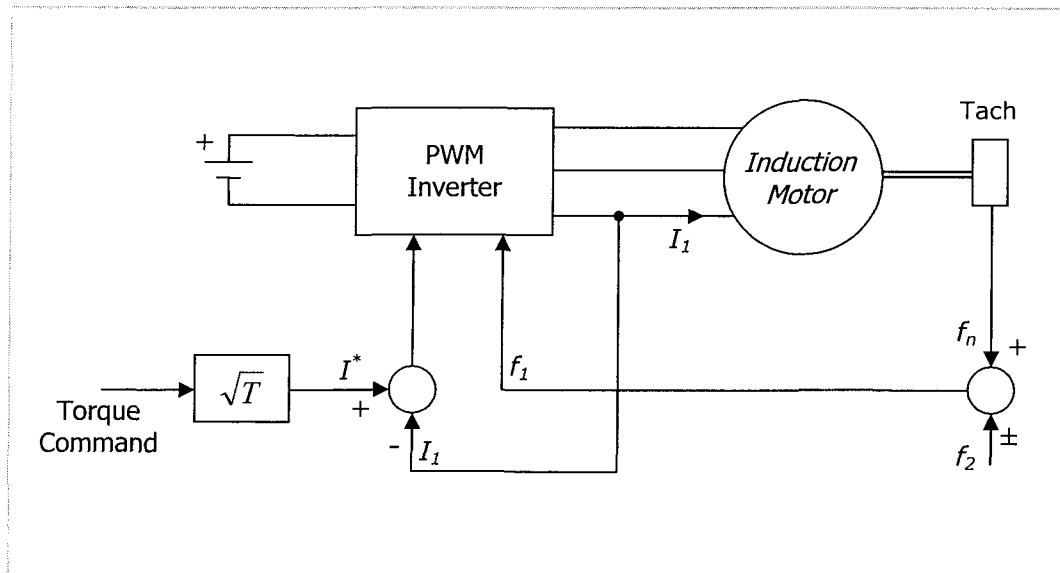


**Figure 2-8.** Closed-loop Speed Control with Constant Slip Frequency

### 2.2.5.3 Transit Drive System

A typical control scheme of a transit drive system is shown in figure 2-9. This method is normally used in traction applications like transit vehicles, in which the torque is controlled directly. As the voltage available for a transit system is a fixed dc supply, a pulse width modulated (PWM) inverter is

considered to vary the needed voltage [26]. In the configuration of figure 2-9 if the slip frequency is kept constant, the torque varies as the square of the stator current. The torque command is producing the reference current whose difference with the actual current will control the output voltage of the PWM inverter.



**Figure 2-9.** Speed Control Scheme for Transit Drive Systems

### 2.2.6 Other Speed Control Systems

More control systems have been presented to control the speed of an induction motor, such as the *rotor resistance control* (in both open-loop and closed-loop configurations) or the *rotor slip energy recovery* (in open-loop operation, torque-speed characteristics for firing angles, and closed-loop operation) which are not the subject of this work.

## 2.3 Scalar Control against Vector Control

### 2.3.1 Scalar Control

In scalar control methods the flux and torque are strongly coupled. A change in the voltage magnitude, current magnitude or frequency will result in a change in both torque and flux magnitude. This results in poor dynamic performances or oscillation. Scalar order is mainly based on using the average values of the parameters. In order to study this method for the implementation, note the model of the machine in steady states. As in steady states we have [27]

$$\vec{V}_r = 0 \quad (2-12)$$

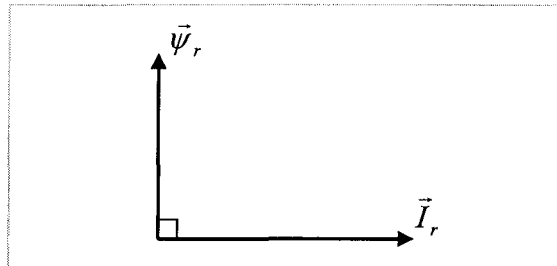
so the equation (1-47) will results to

$$R_r \vec{I}_r = -j(\omega_e - \omega_r) \vec{\psi}_r = -jg\omega_e \vec{\psi}_r \quad (2-13)$$

or

$$\vec{\psi}_r = j \left( \frac{R_r}{g\omega_e} \right) \vec{I}_r \quad (2-14)$$

Figure 2-10 shows the flux and current vectors situation for an induction



**Figure 2-10.** Rotor Flux and Current Vectors in Steady states

machine in steady states. The electromagnetic torque will be given by

$$T_e = p \Im m(\vec{\psi}_r \vec{I}_r^*) = p I_r \psi_r \quad (2-15)$$

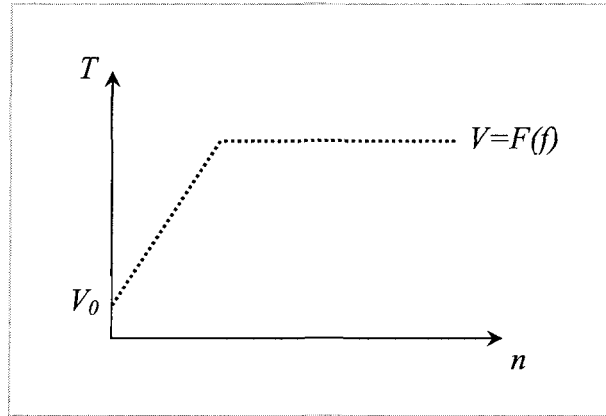
so from (2-14) and (2-15)

$$T_e = p \frac{S \omega_e}{R_r} \psi_r^2 \quad (2-16)$$

in which  $S \omega_e$  is the angular speed of the rotor and

$$S \omega_e = \omega_e - \omega_r \quad (2-17)$$

From section 2.2.3 and the figure 2-5 results the graph of figure 2-11 as the relationship between torque and rotor speed.



**Figure 2-11.** Torque-Speed Characteristics of Induction Motor

### 2.3.2 Vector Control Principals

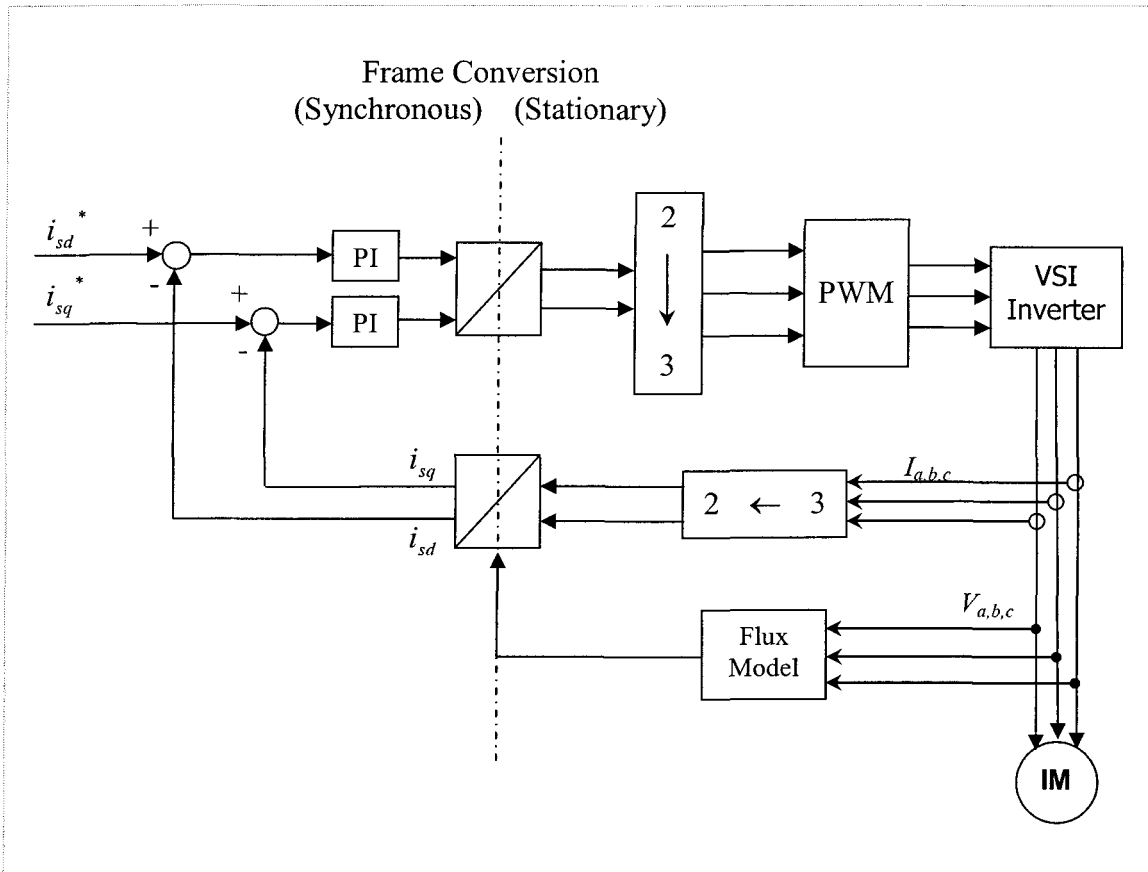
Denomination of *Vector controllers* refers to as these controllers basically control both of the phase and amplitude of the ac excitation. The vector control of currents and voltages results in control of the spatial orientation of the

electromagnetic fields in the machine and has led to the term *field orientation* [28]. Virtually, all induction machines vector controllers employ a  $90^\circ$  spatial angle between the critical flux and MMF which are called oriented systems.

The VC (also known as field oriented control) was introduced By Blaschke in early 1970's [4][28] which allowed decoupled control of the machine flux and torque producing currents. This technique allowed an inverter driven induction machine to achieve a high performance, at least the same as a DC machine achieving fast, near step changes in machine torque [20]. It was the technique known as Direct Field Oriented Control which considers that the space vector of rotor flux is known by the amplitude and position (either by measurement or estimation). Later, Hasse [29] proposed the technique of Indirect Field Oriented Control in which speed of the slip is used to compile the position of the rotor flux vector. However the compilation of the position of the rotor flux vector utilizes the resistance of the rotor that varies itself as a function of frequency and temperature. This may affect the controller and cause performance degradation.

In field oriented control (FOC), decoupled control of flux and torque is achieved by splitting the current into two DC components rotating in a reference frame aligned with the selected flux. The direct axis current acts to control the flux magnitude, while the quadrature axis current controls the torque magnitude. This is somehow similar to the control of DC machine which has a field winding (flux magnitude here) and an armature winding (torque produced here).

The synchronously rotating reference frame can be aligned with either the stator flux [30], the air-gap flux [31], or the rotor flux [4][32][33]. Figure 2-12 shows the general frame conversion architecture of VC.



**Figure 2-12.** Vector Control Frame Conversion

Schemes for obtaining the rotor flux position can be divided to two categories known as Direct and Indirect Field Oriented Control.

*Direct FOC* usually indicates that the rotor flux is directly evaluated, either by using direct flux measurement, with *Hall Effect* transducers or search coil

installed in the machine, or utilizing a flux observer. *Indirect FOC* on the other hand, usually implies that the instantaneous slip frequency is calculated from the two axis currents. As the slip frequency gives the speed of the rotor flux relative to the rotor mechanical speed, the slip is then integrated and added to the measured mechanical position. This gives a more accurate estimate of the flux position at low speeds and allows operation of the machine throughout the entire speed range.

The technique of Vector Control has become established as the main technique for high performance flux and torque control of induction machine drives.

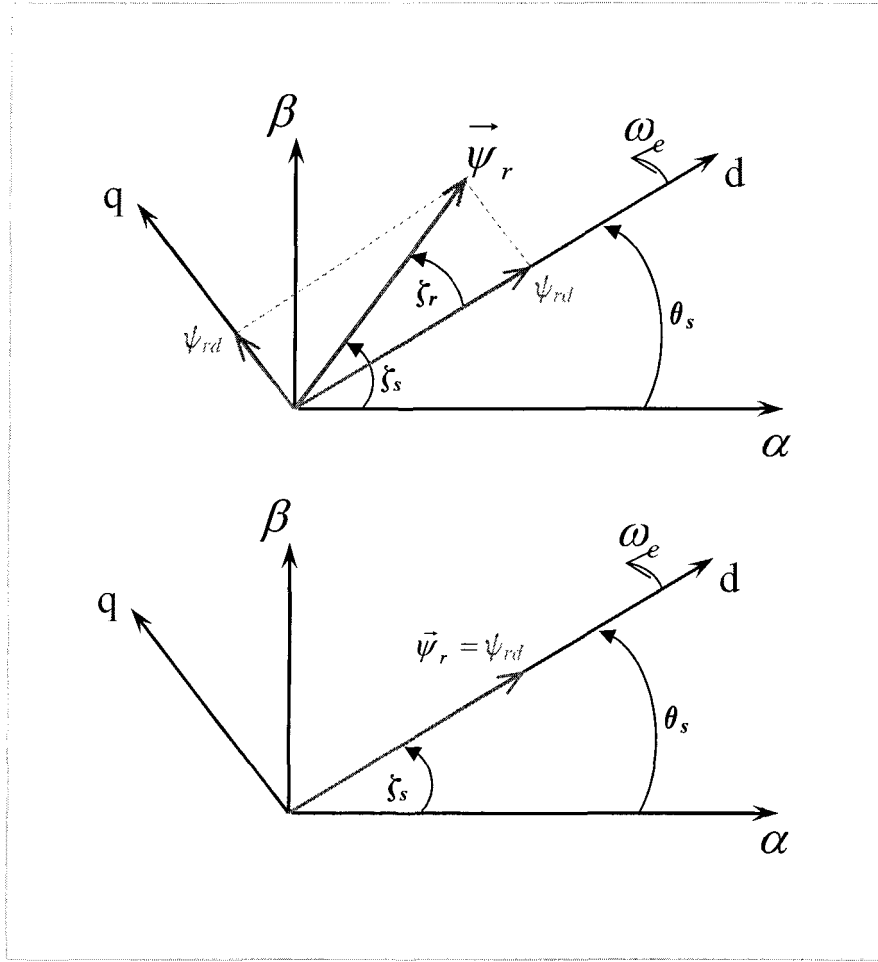
### 2.3.3 Using Vector Control Technique in the Arbitrary Frame

Following the arbitrary frame we'll repeat the equation (1-53) which was used to present the induction machine model as (2-18) in the following

$$\frac{d}{dt} \begin{bmatrix} I_{sd} \\ I_{sq} \\ \psi_{rd} \\ \psi_{rq} \\ \omega_r \end{bmatrix} = \begin{bmatrix} -\left(\frac{R_s}{\sigma L_s} + \frac{R_r(1-\sigma)}{\sigma L_r}\right) I_{sd} + \omega_e I_{sq} + \frac{R_r L_m}{\sigma L_s L_r^2} \psi_{rd} + \frac{L_m \omega_r}{\sigma L_s L_r} \psi_{rq} \\ -\omega_e I_{sd} - \left(\frac{R_s}{\sigma L_s} + \frac{R_r(1-\sigma)}{\sigma L_r}\right) I_{sq} - \frac{L_m \omega_r}{\sigma L_s L_r} \psi_{rd} + \frac{R_r L_m}{\sigma L_s L_r^2} \psi_{rq} \\ \frac{R_r L_m}{L_r} I_{sd} - \frac{R_r}{L_r} \psi_{rd} + (\omega_e - \omega_r) \psi_{rq} \\ \frac{R_r L_m}{L_r} I_{sq} - (\omega_e - \omega_r) \psi_{rd} - \frac{R_r}{L_r} \psi_{rq} \\ \frac{p^2 L_m}{J L_r} (I_{sq} \psi_{rd} - I_{sd} \psi_{rq}) - \frac{F}{J} \omega_r - \frac{p}{J} T_r \end{bmatrix} + \frac{1}{\sigma L_s} \begin{bmatrix} V_{sd} \\ V_{sq} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2-18)$$

The change in the situation of the flux vector in the arbitrary frame is shown in figure 2-13.





**Figure 2-13.** Flux Vector in a general frame (up) and arbitrary frame (bottom)

As the flux vector is rotating simultaneously with the chosen frame ( $d, q$ ) it's clear that the  $q$  axis rotor flux is zero, therefore

$$\vec{\psi}_r = \psi_{rd} \quad (2-19)$$

and the voltage equation (1-51) will become as

$$0 = R_r + \frac{d\psi_{rd}}{dt} \quad (2-20)$$

Rearranging the flux equation for induction machines to give the direct axis rotor current in terms of the rotor flux and stator current

$$i_{rd} = \frac{\psi_{rd} - L_m i_{sd}}{L_r} \quad (2-21)$$

From the two recent equations

$$\frac{d\psi_{rd}}{dt} + \frac{R_r}{L_r} \psi_{rd} = \frac{R_r L_m}{L_r} i_{sd} \quad (2-22)$$

which shows the rotor flux magnitude is related to the direct axis stator current by a first order differential equation the two recent equations, so can be controlled by controlling the direct axis stator current.

Considering the recent equations, the matrix equation of the induction motor model given in (2-18) would be presented as given in below

$$\frac{d}{dt} \begin{bmatrix} I_{sd} \\ I_{sq} \\ \psi_r \\ 0 \\ \omega_r \end{bmatrix} = \begin{bmatrix} -\left(\frac{R_s}{\sigma L_s} + \frac{R_r(1-\sigma)}{\sigma L_r}\right) I_{sd} + \omega_e I_{sq} + \frac{R_r L_m}{\sigma L_s L_r^2} \psi_r \\ -\omega_e I_{sd} - \left(\frac{R_s}{\sigma L_s} + \frac{R_r(1-\sigma)}{\sigma L_r}\right) I_{sq} - \frac{L_m \omega_r}{\sigma L_s L_r} \psi_r \\ \frac{R_r L_m}{L_r} I_{sd} - \frac{R_r}{L_r} \psi_r \\ \frac{R_r L_m}{L_r} I_{sq} - (\omega_e - \omega_r) \psi_r \\ \frac{p^2 L_m}{J L_r} (I_{sq} \psi_{rd}) - \frac{F}{J} \omega_r - \frac{p}{J} T_r \end{bmatrix} + \frac{1}{\sigma L_s} \begin{bmatrix} V_{sd} \\ V_{sq} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2-23)$$

The induction machine voltage equation for arbitrary synchronous frame direct axis is

$$0 = R_r I_{rd} + \frac{d\psi_{rd}}{dt} - (\omega_e - \omega_r) \psi_{rq} \quad (2-24)$$

In the rotor oriented reference frame the  $q$  axis rotor flux is equal to zero, therefore

$$0 = R_r I_{rd} + \frac{d\psi_{rd}}{dt} \quad (2-24)$$

The flux equations are given in (2-25) to (2-28)

$$\psi_{sd} = L_s I_{sd} + L_m I_{rd} \quad (2-25)$$

$$\psi_{sq} = L_s I_{sq} + L_m I_{rq} \quad (2-26)$$

$$\psi_{rd} = L_r I_{rd} + L_m I_{sd} \quad (2-27)$$

$$\psi_{rq} = L_r I_{rq} + L_m I_{sq} \quad (2-28)$$

where

$$L_s = L_{ls} + L_m \quad (2-29)$$

$$L_r = L_{lr} + L_m \quad (2-30)$$

Rearranging (2-27) to give the direct axis rotor current in terms of rotor flux and stator current gives

$$I_{rd} = \frac{\psi_{rd} - L_m I_{sd}}{L_r} \quad (2-31)$$

Submitting (2-31) to (2-24) then gives

$$\frac{d\psi_r}{dt} + \frac{R_r}{L_r} \psi_{rd} = \frac{R_r L_m}{L_r} I_{sd} \quad (2-32)$$

which shows that the rotor flux magnitude is related to the direct axis stator current by a first order differential equation. Thus the rotor flux magnitude could be controlled if the direct axis stator current is controlled.

In the steady states the rotor flux is directly proportional to the stator axis current as

$$\psi_{rd} = L_m I_{sd} \quad (2-32)$$

As locking the phase of the reference system such that the rotor flux is entirely in d-axis, the equation (2-28) is equal to zero, therefore

$$I_{rq} = -\frac{L_m}{L_r} I_{sq} \quad (2-33)$$

On the other hand, voltage equation for q-axis is given as below

$$0 = R_r I_{rq} + (\omega_e - \omega_r) \psi_{rd} \quad (2-34)$$

*Slip* equation (2-17) then results

$$S\omega_e = -\frac{R_r I_{rq}}{\psi_{rd}} \quad (2-35)$$

By using equation (2-33), and considering the direction of the flux vector shown in figure 2-12

$$S\omega_e = -\frac{R_r L_m I_{sq}}{L_r \psi_r} \quad (2-36)$$

The torque equation of induction motor is

$$T_e = p \frac{L_m}{L_r} \vec{\psi}_r \times \vec{I}_s \quad (2-37)$$

It's easy to find that in the arbitrary frame

$$T_e = p \frac{L_m \psi_r I_{sq}}{L_r} \quad (2-38)$$

The stator variables are obtained by using the  $d,q$  to  $a,b,c$  transformation matrix given in below

$$\begin{bmatrix} f_a \\ f_b \\ f_c \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos \theta & -\sin \theta \\ \cos(\theta - \frac{2\pi}{3}) & -\sin(\theta - \frac{2\pi}{3}) \\ \cos(\theta + \frac{2\pi}{3}) & -\sin(\theta + \frac{2\pi}{3}) \end{bmatrix} \begin{bmatrix} f_d \\ f_q \end{bmatrix} \quad (2-39)$$

### 2.3.4 Indirect Vector Control

With a constant flux, the main machine variables could be given from the equations (3-32), (2-33) and (2-36) as in the following

$$I_{sd}^* = \frac{\psi_r}{L_m} \quad (2-40)$$

$$I_{sq}^* = \frac{1}{p} \frac{L_r}{L_m} \frac{T_e}{\psi_r} \quad (2-41)$$

$$S\omega_e^* = \frac{L_m}{\tau_r} \frac{I_{sq}^*}{\psi_r^*} \quad (2-42)$$

In the equations above the quantities are commanded or estimated in drive control are denoted by asterisk (\*). The time constant  $\tau_r$  is obtained from

$$\tau_r = \frac{L_r}{R_r} \quad (2-43)$$

As it's discussed in section 2.3.2 with this intention, the exact knowledge of the rotor flux position is imperative. The angle  $\theta_s$  which determines the rotor flux angle is calculated by integrating on the expression of the slip speed.

$$\omega_e = \omega_r + S\omega_e \quad (2-44)$$

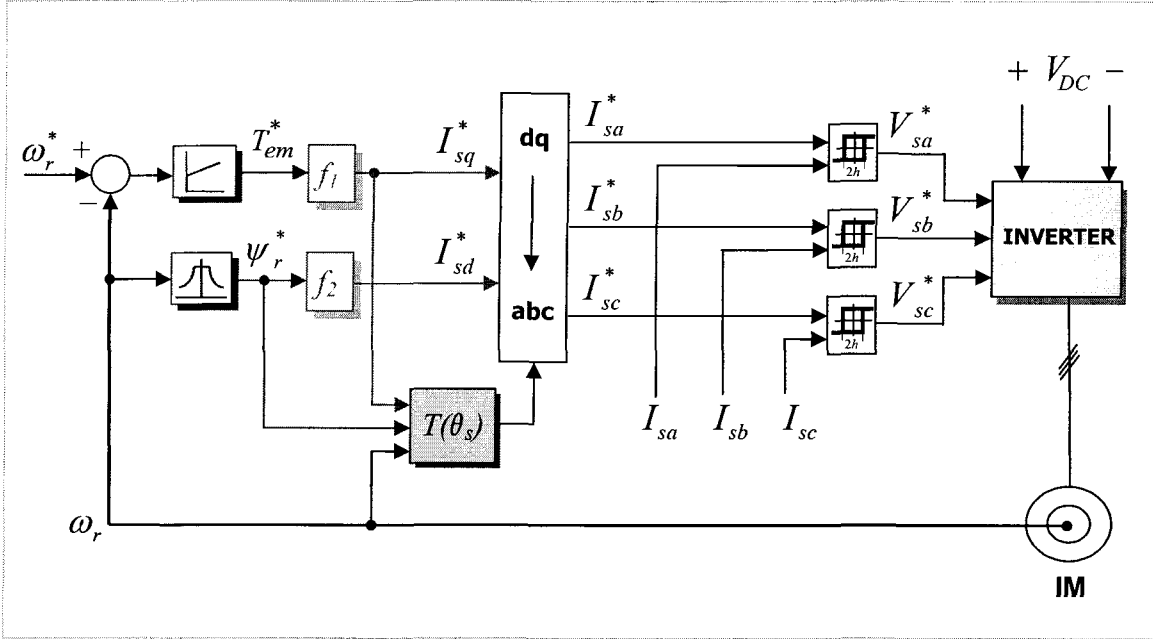
$$\theta_s = \int \omega_e dt \quad (2-45)$$

Submitting (2-42) and (2-44) to (2-45) gives

$$\theta_s^* = \int \left( \omega_r + \frac{R_r L_m I_{sq}^*}{L_r \psi_r^*} \right) dt \quad (2-46)$$

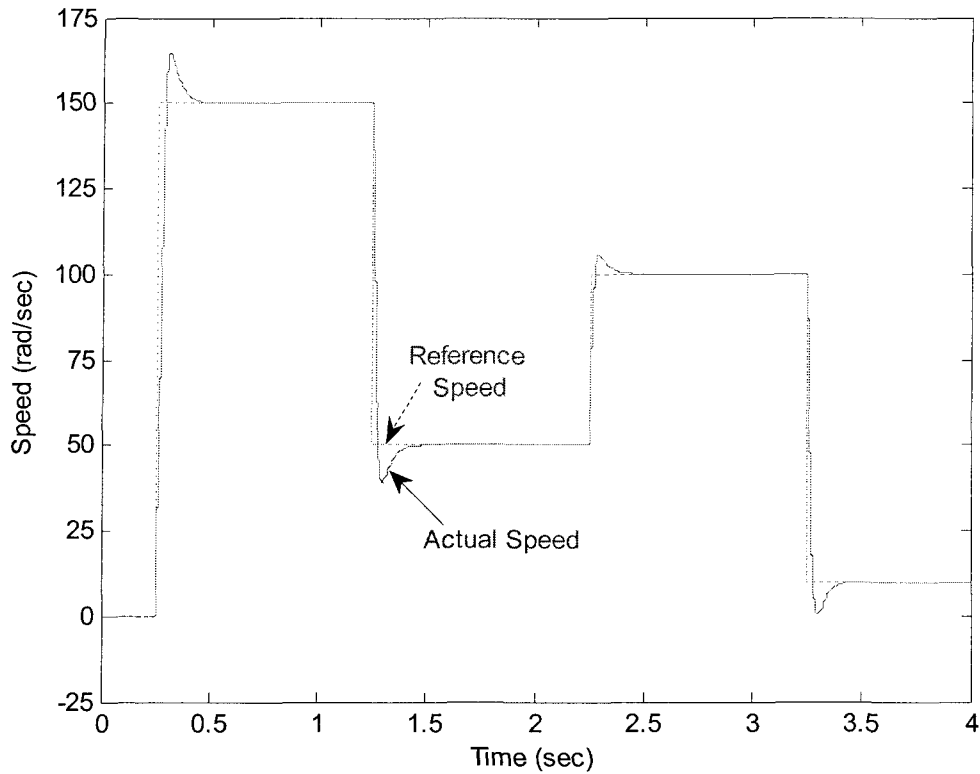
The principal schema of the Indirect Vector Control (or IFOC) is given in figure 2-14 in which the function blocks  $f_1$  and  $f_2$  are presented by the equations (2-41) and (2-40) respectively. Since this method relies on knowledge of the machine parameters such as  $L_m$  and  $\tau_r$  the real values of which may be changing as the operation condition changes, consideration should be given to the design to the effects of parameters variations.

As an example, the indirect vector control system given in figure 2-14 is implemented in *Simulink* and the results for the real-time simulation for a constant



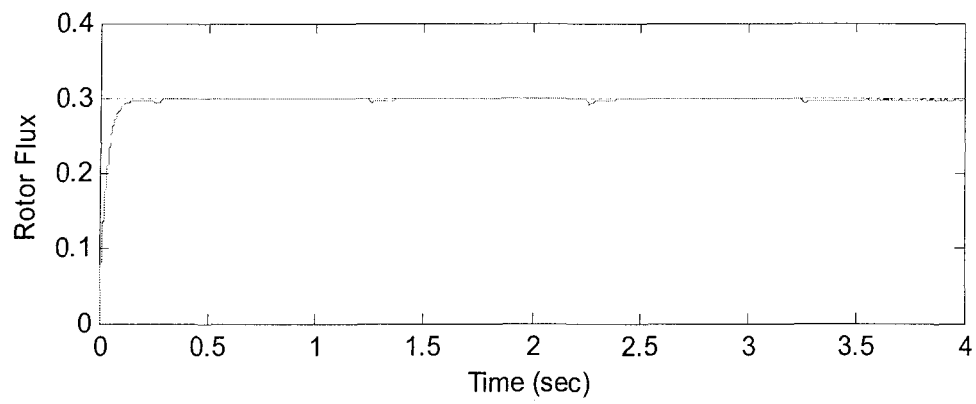
**Figure 2-14.** Indirect Vector Control Architecture

flux operation of the machine are presented in the following. More information about the real-time simulation and Simulink software will be presented and studied in chapter 4. Just to have an idea, simulating the vector control architecture of figure 2-14 results the tracking rotor speed which is compared with the reference speed in figure 2-15.

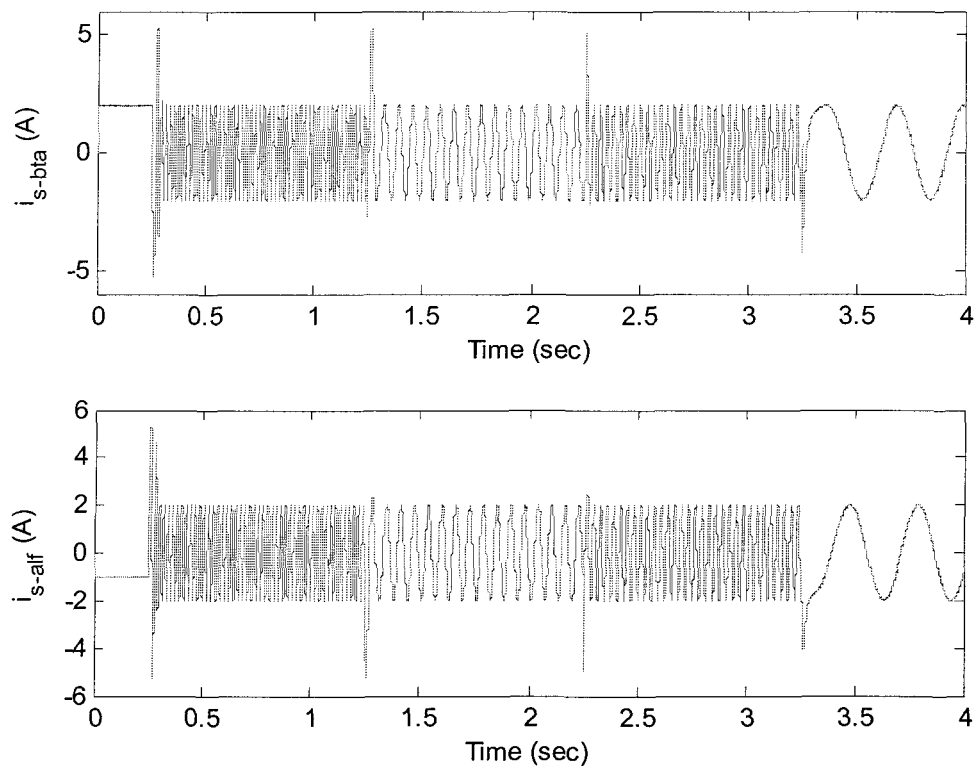


**Figure 2-15.** Tracking Rotor Speed Reference

The tracking rotor flux of the schema and the stator current space vector  $\alpha, \beta$  components for this operation are given in figure 2-16 and 2-17 respectively.



**Figure 2-16.** Tracking Rotor Flux Reference



**Figure 2-17.** Stator Current Vector Control Components



### 2.3.5 Direct Vector Control

In the case of Direct Vector Control the angle is measured or estimated, unlike in the case of Indirect VC which it was calculated. Determining this angle could be either from the measurement from the air gap flux or from terminal currents and voltages. In the latter case angle and magnitude of the rotor flux can be calculated by

$$\vec{\psi}_s = \int (\vec{V}_s - R_s \vec{I}_s) dt \quad (2-47)$$

$$\vec{\psi}_r = \frac{L_m}{L_r} (\vec{\psi}_s - L'_s \vec{I}_s) \quad (2-48)$$

where

$$L'_s = L_s - \frac{L_s^2}{L_r} \quad (2-49)$$

The principal schema of the Direct Vector Control (or DFOC) is given in figure 2-18 in which the function blocks  $f_1$  and  $f_2$  are presented by

$$I_{sd}^* = (\psi_r^* - \psi_r) \left( k_{p\psi} + \frac{k_{i\psi}}{S} \right) \quad (2-50)$$

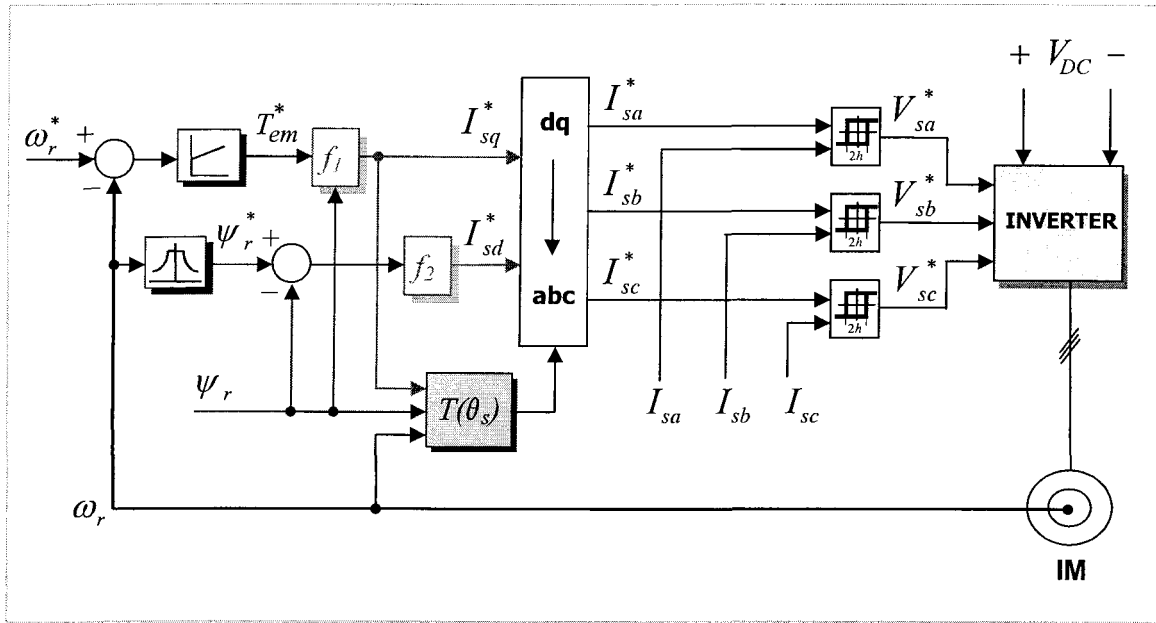
$$I_{sq}^* = \frac{1}{p} \frac{L_r}{L_m} \frac{T_e}{\psi_r} \quad (2-51)$$

and the rotor flux position  $\theta_s$  is given by

$$\theta_s = \tan^{-1} \left( \frac{\psi_{r\beta}}{\psi_{r\alpha}} \right) \quad (2-52)$$

Although Direct Vector Control may be relatively insensitive to the variations (depending on the actual implementation) of the rotor parameters, the

performance there may be sluggish at low speed operation due to inaccurate knowledge of the stator resistance, integration drift, etc. Refer to [34] for more details.



**Figure 2-18.** Direct Vector Control Architecture

# **CHAPTER 3**

## **SPEED ESTIMATION**

### 3.1 Speed Sensorless Control

Speed estimation methods have aroused a great interest in the recent years among induction motor control researches. A sensorless drive is a possibility in which the control technique is based on a mathematical representation of the motor, estimating the feedback speed by feeding the model with measurements of stator voltage and current. Speed sensorless control techniques first appeared in 1975 [35] and now are considered as an important area.

From the drive system's point of view, elimination of the speed sensors and the associated measurement connections has the advantages of lower cost (the speed/position detectors costs are sometimes in the range of the price of the motor itself), ruggedness, noise immunity,... which increases the reliability of system. Accurate speed information which is necessary for a high-performance and high-precision controller can be estimated from the values of stator current and voltage that can be easily measured. The performance of such drives can be comparable to a tuned sensed field oriented control.

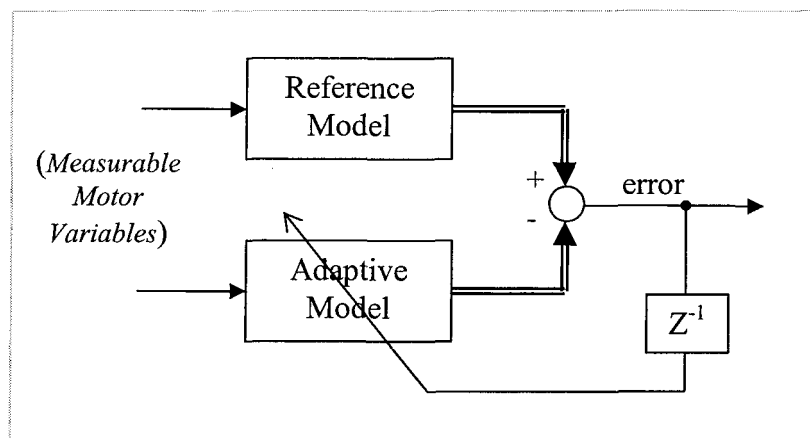
Since 1980s the concept of rotor speed estimation has been studied extensively where the instantaneous stator currents and voltages were used to estimate the rotor speed of an induction motor such as in *Model Reference Adaptive Systems* and *Extended Kalman Filters*. Some of the more common in use methods are studied in the following.

## 3.2 Speed Estimation Techniques

### 3.2.1 Model Reference Adaptive Systems

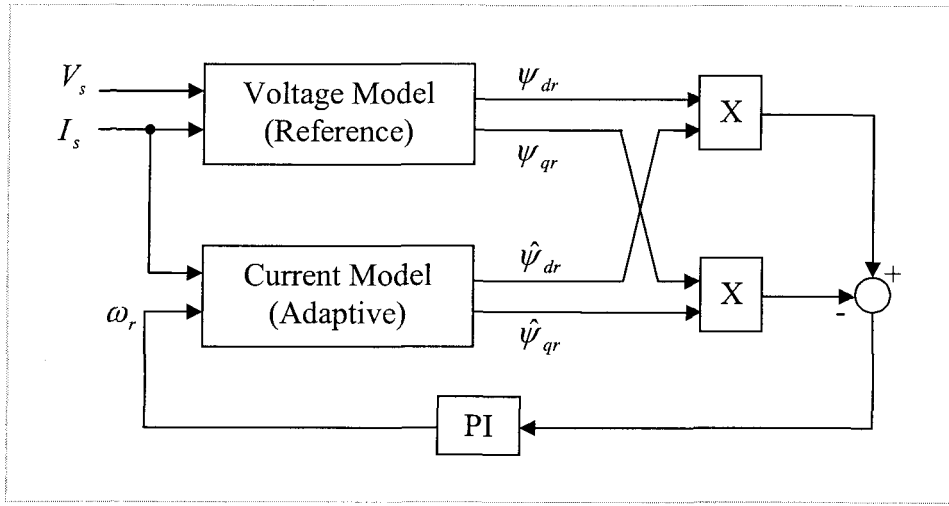
One of the most commonly used techniques for speed sensorless control is known as MRAS (Model Reference Adaptive System) which is often approached to rotor flux based technique [36][37], though it's possible to use other parameters such as Electromotive Force [38] which is used in this work, as it will later be explained in this chapter.

The observer is constructed from two induction machine models for the used parameter. The first model uses two measurable values and is used as the reference model. The second one which is the adaptive model, uses the equation that needs the output that is supposed to be controlled as an input as is shown in figure 3-1. The adaptation mechanism is derived to minimize the error of the two outputs of the models.



**Figure 3-1.** Adaptive Estimator Structure

Figure 3-2 shows the scheme of a flux based MRAS speed estimation. The measurable parameters used to feed the reference (voltage model) are the stator current ( $I_s$ ) and voltage ( $V_s$ ), while the estimated speed is used in adaptive (current model). The error between the fluxes obtained from these two models would be zero in the ideal estimation.



**Figure 3-2.** MRAS Speed Estimation Scheme

Another alternative MRAS scheme [39] utilizes the error in torque producing current to adapt estimation, as in [40] it is claimed that this scheme offers superior performance to the rotor flux MRAS scheme.

### 3.2.2 Adaptive State Observer Speed Estimation Techniques

This commonly adopted Adaptive Speed Observer (ASO) is first presented in 1993 [41], then various other works were presented to overcome the problems

[42][43][44]. The state-space model of induction machine in stationary frame is utilized to form an observer where the model is given by the equations in (2-2), while the state observer is given by

$$\hat{\dot{X}}(t) = A(t).\hat{X}(t) + B.u(t) + G(\hat{i}_s - i_s) \quad (3-1)$$

where  $G$  is the observer gain matrix, which is chosen so that the poles of the observer are proportional to those of the induction machine.

### 3.2.3 Extended Kalman Filter Observers

*Kalman Filtering* technique is a further extension of the adaptive stator observer theory which solves the problem of observers gain matrix selection by utilizing the *Riccati* difference equation [45]. It's a recursive mean square state estimator, which is capable to produce optimal estimates of states that are not measurable [46]. This provides the minimum mean square error for estimation when measurement noise and process disturbances are modeled as white noise [47]. The Kalman filter can be extended to allow joint state and parameter estimation which is termed the Extended Kalman Filter (EKF) and can be used to estimate the motor parameters [48] and be developed for the estimation of the rotor speed [10][11][49].

In EKF technique, the induction machine model is augmented with speed as an additional state, resulting in the non-linear model

$$x(k+1) = f(x(k), u(k), k) + w(k) \quad (3-2)$$

$$y(k) = h(x(k), k) + v(k) \quad (3-3)$$

where the parameters  $x$  ,  $y$  ,  $u$  ,  $v$  ,  $w$  are the state vector, the output vector, the input vector, noise matrix of the state model, and noise matrix of output model respectively.

The main advantage of using EKF is the robustness; no comprehensive parameter sensitivity study is in the literature. However the problems of stator and rotor resistance estimation are addressed, as in [10] the sensitivity of the EKF to rotor time constant is highlighted. This shows the dependence of the speed estimator on the rotor time constant/slip relationship similar to the MRAS and ASO schemes. The main drawback with EKF can be considered the complexity of the algorithm. This computational complexity is specially a problem while using in a real-time system which has to be adjusted online.

#### **3.2.4 Extended Luenberger Observers**

The Extended Luenberger Observer (LSO) [50][51] is an extension of the adaptive state observer. As with the Extended Kalman Filter, the system model is augmented with an additional state representing the parameter to be estimated. This is linearized resulting in the same system model used in EKF. However, in this case the gain matrix is calculated utilizing a pole placement technique to arbitrarily locate the poles of the observer.



Comparison with the EKF [52] shows that whilst similar performance is obtained between the two observers, though the torque estimation for the ELO is noisy and has to be filtered to produce a useful signal.

### **3.2.5 Artificial Neural Networks**

#### **3.2.5.1 Introduction**

Although the computers are faster than brain for many tasks, they are significantly slower for everyday tasks such as visual and voice recognition, or conversing in a natural language. It's because brain has  $10^{10}$  neurons, each of which has complicated behavior and interact in complicated ways, whereas computers have less transistors, each of which have simple behavior and interact in simple ways [53]. In fact, the brain can be considered as a highly complex, nonlinear, and parallel computer (information processing system) [54]. Many concepts inspired from the behavior of a neuron in biological nervous systems are used in Neural Network computing (brain-like computations). Interest in Artificial Neural Networks (ANNs) in many fields has grown rapidly over the last decade, and discussions on them are occurring everywhere nowadays. Their promise seems very bright as nature itself is the proof that this kind of thing works. A neural network is an information processing system that is non-algorithmic, non-digital, and intensely parallel. These specifications make it to be considered a robust estimation method while a non-linear function is present, and the

parameters are not exactly known [55]. The ability of this technique in addressing the problems whose solutions have not been explicitly formulated has proven in different areas. They can and have started to become very attractive for different control tasks.

Since early 1990s, there have been some investigations into the application of neural networks to power electronics and ac drives, including speed estimation [56][57]. In [58] a NN speed observer is compared with Luenberger and Kalman Observers and the results show it appreciated as reliable for sensorless induction motor control due to its algorithm simplicity and its robustness. Using ANNs in model has the advantages of extremely fast parallel computing, immunity from input harmonic ripples, and fault tolerance characteristics. There are different methods of using a NN-based estimator, with different algorithms of training the system. The training can be done offline or online, with a combination of the previous methods of speed estimation. For example, it's possible to use an EKF to train the estimator [11][59] which is more suitable with offline use according to the computation time. Another solution is using the NN speed estimator in a MRAS-based speed sensorless control [60][61][62] which has good results and is fast enough in online training too. A simple scheme for such a system is the same in figure 3-2 by replacing the Adaptive Model block with a NN-based speed estimator. It's also possible to use neural networks as the controller itself like what is presented in [63] and [64] which is not subject of this work as it's not going to solve the problems of need to the speed sensor.

The scheme used in this work is a little different which will be clearly explained in chapter 4.

### 3.2.5.2 History

The first step toward Artificial Neural Networks came in 1943 when Warren McCulloch, a neurophysiologist, and a young mathematician, Walter Pitts, wrote a paper on how neurons might work (“*A Logical Calculus of Ideas Immanent in Nervous Activity*”). They modeled a simple neural network with electrical circuits. Reinforcing this concept of neurons and how they work was a book written by Donald Hebb. The *Organization of Behavior* was written in 1949 [65]. As computers advanced into their infancy of the 1950s, it became possible to begin to model the rudiments of these theories concerning human thought. In 1950s and 1960s a group of researchers combined the biological and psychological insights to produce the first ANNs. Unfortunately, the earlier successes caused people to exaggerate the potential of neural networks, particularly in light of the limitation in the electronics then available. This combined with some fears, caused respected voices to critique the neural network research. The result was to halt much of the funding. This period of stunted growth lasted through 1981.

In 1982 several events caused a renewed interest on ANNs after John Hopfield presented a paper [66] to the National Academy of Sciences. With clarity and mathematical analysis, he showed how such networks could work and what they could do. Soon, interest and so funding was flowing once again. During

1980s and 1990s activity in the field of ANNs increased exponentially. Dedicated networks of biological or microelectronic neurons could provide the computational capabilities described for a wide class of problems having combinatorial complexity [67].

### **3.2.5.3 Capabilities**

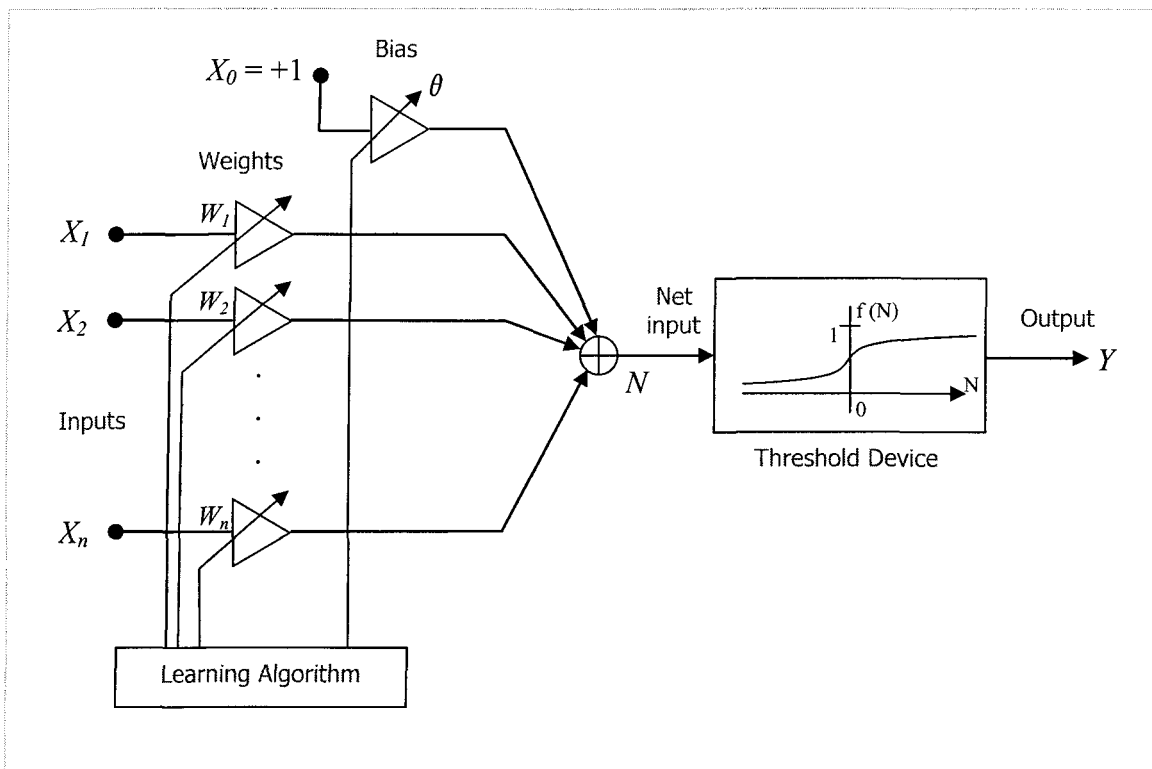
Basically, most applications of neural networks fall into the following categories, though imaginative researchers are devising new uses for ANNs daily.

- Prediction: Use input values to predict some outputs in future.
- Classification: Use input patterns to determine the classification.
- Data association: Similar to Classification, but it also recognizes data that contains errors.
- Data conceptualization: Analyze the inputs so that grouping relationships can be inferred.
- Data filtering: Smooth an input signal, or noise reduction.
- Process modeling and control: Creating a neural network model for a physical plant then using that model to determine the best control settings for the plant.

### **3.2.5.4 Basic principles**

A neural network is a powerful data modeling tool that is able to capture and represent complex input/output relationships. It's an information processing

system that is non-algorithmic, non-digital, and intensely parallel. It consists of a number of very simple and highly interconnected processors called neurodes, or like their biological pattern, neural cells in the brain, neurons. The neurons are connected by a large number of weighted links, over which signal can pass [68]. Real neurons have a finite dynamic range from nil response to the full firing rate, which is modeled by a non-linear, leveling off at 0 and 1. The additional bias term that determines the spontaneous activity of the neuron in the absence of inputs is modeled by a threshold value [69]. This non-linearity plays an essential role in the behavior of neural networks. The entire model of the neuron is given below in figure 3-3.



**Figure 3-3.** A typical model of neuron

The *transfer function* of a neuron would consist of two steps; first, the neuron computes the weighted input receiving along its input connection:

$$N_j = \sum_{j=1}^n w_{ij} \cdot x_j + \theta_j \quad (3-4)$$

The second step consists of converting the net input to an activation level.

This can be done by using:

- Threshold function:

$$f(N) = \begin{cases} 1 & \text{if } N \geq 0 \\ 0 & \text{if } N < 0 \end{cases} \quad (3-5)$$

- Piecewise-linear function:

$$f(N) = \begin{cases} 1 & \text{if } N \geq 1/2 \\ N + \frac{1}{2} & \text{if } -1/2 < N < 1/2 \\ 0 & \text{if } N \leq -1/2 \end{cases} \quad (3-6)$$

- Sigmoid function:

$$f(N) = \frac{1}{1 + \exp(a \cdot N)} \quad (3-7)$$

where  $a$  is the *slope parameter* of the sigmoid function.

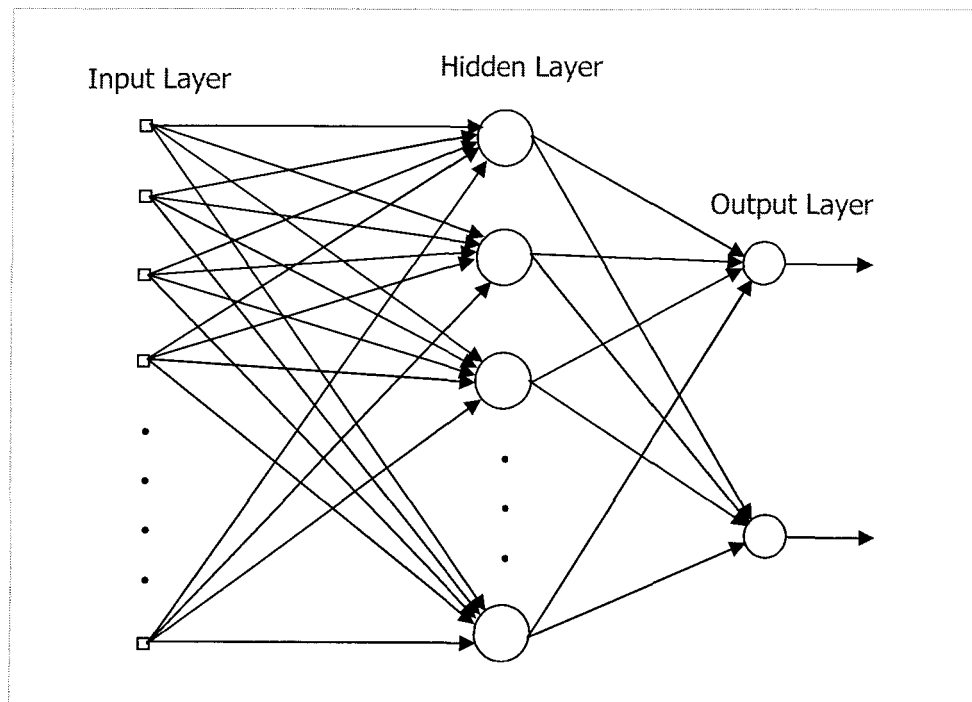
The activation function defines in equations (3-5), (3-6), and (3-7) range from 0 to +1. It's desirable to have the activation function range from -1 to +1, in which case the activation function assumes an asymmetric form with respect to

the origin [54]. Specifically, for a sigmoid we may use the *hyperbolic tangent function*, defined by:

$$f(N) = \tanh\left(\frac{N}{2}\right) = \frac{1 - \exp(-N)}{1 + \exp(-N)} \quad (3-8)$$

### 3.2.5.5 Architecture

A layered neural network is a network of neurons organized in the form of layers. If there are additional neurons to source nodes (input layer) and output(s) (output layer) we call them hidden neurons. By adding one or more hidden layers, the network is capable to extract higher-order statistics. The structure of such a network is shown in figure 3-4.



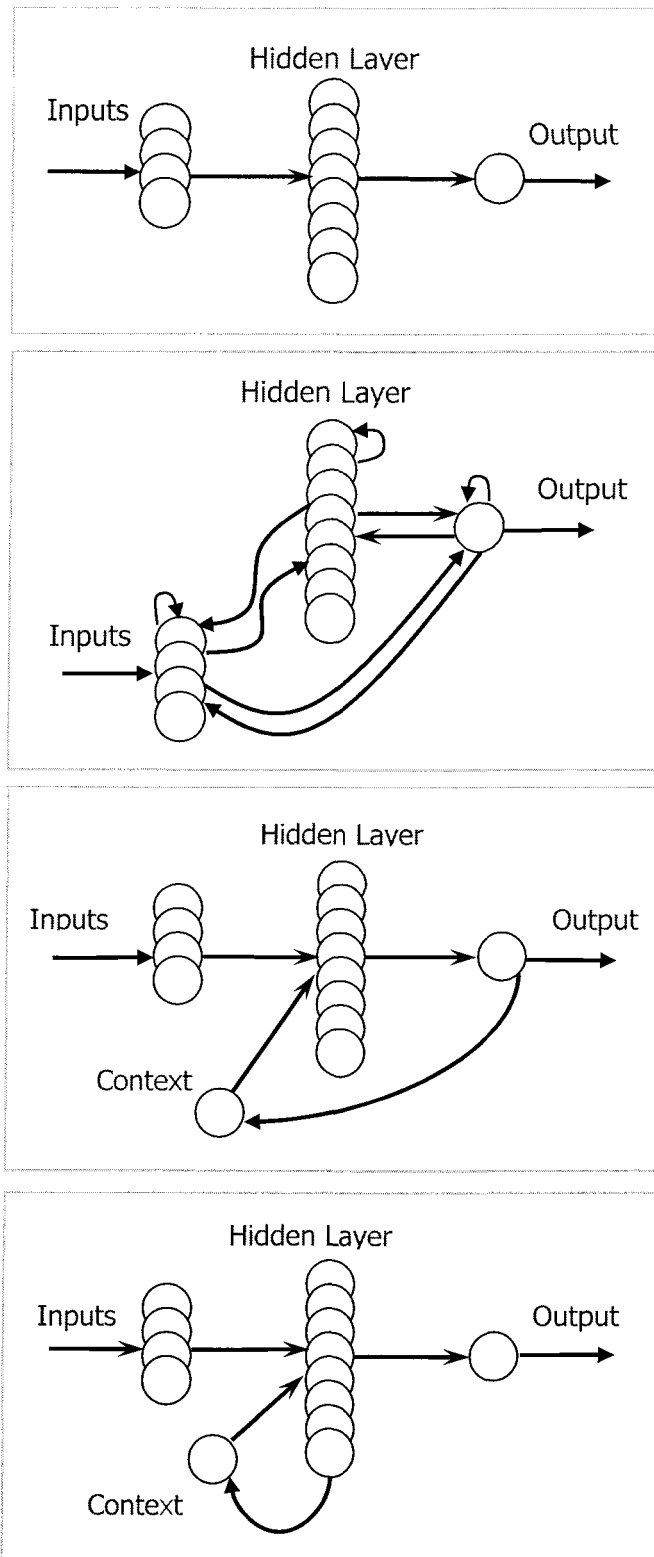
**Figure 3-4.** Fully connected network with one hidden layer

The structure above is the simplest feed forward network with one hidden layer. It's possible to have more than one hidden layer, as the partially connected networks exist in which some of the communication (synaptic) links are missing. There are other types of networks with one or more closed loops in the network topology which are called recurrent neural networks (RNN). They are developed to deal with the time dependent or time-lagged signals and are usable for the problems where the dynamics of the considered process is complex and the measured data is noisy. Specific groups of the units get the feedback signals from the previous time steps and these units are called *context* unit [70].

Recurrent networks are normally used in situations when the current information is time-dependent over short time. As the sequence of inputs is important, we need to somehow store a record of the prior inputs to produce the answer in next steps. A recurrent network uses feedback from one or more of its units at time step  $k-1$ , say  $y(k-1)$  are part of the inputs used in selecting the next set of outputs  $y(k)$ . Fully recurrent networks, as their name suggests, provide two-way connections between all processors in the neural network.

A Jordan recurrent network [71] is a network with feedback from the output units back to a set of context units. This form of recurrence is a compromise between the simplicity of a feed-forward network and the complexity of a fully recurrent neural network because it still allows the popular back propagation training algorithm (described in the following) to be used.





**Figure 3-5. Comparing Different Neural Networks:**

a) Feed forward NN, b) Fully recurrent NN, c) Jordan NN, d) Elman NN

Elman [72] is another one, allowing feedback from the hidden units to a set of additional inputs called context units.

These are the most common networks in use and figure 3-5 gives a simple idea about the difference of the structures in a general structure with one hidden layer and one output [73].

### 3.2.5.6 The ability of network to learn

Learning is defined [74] as: *“A process by which the free parameters of a neural network are adapted through a continuing process of simulation by the environment in which the network is embedded. Learning strategy is determined by the manner in which the parameter changes take place”*.

In other words learning process is changing the matrix of synaptic weights. What we do in practice, is to change the synaptic weights of each inter-connection in the network to update it, and then study the result:

$$w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n) \quad (3-9)$$

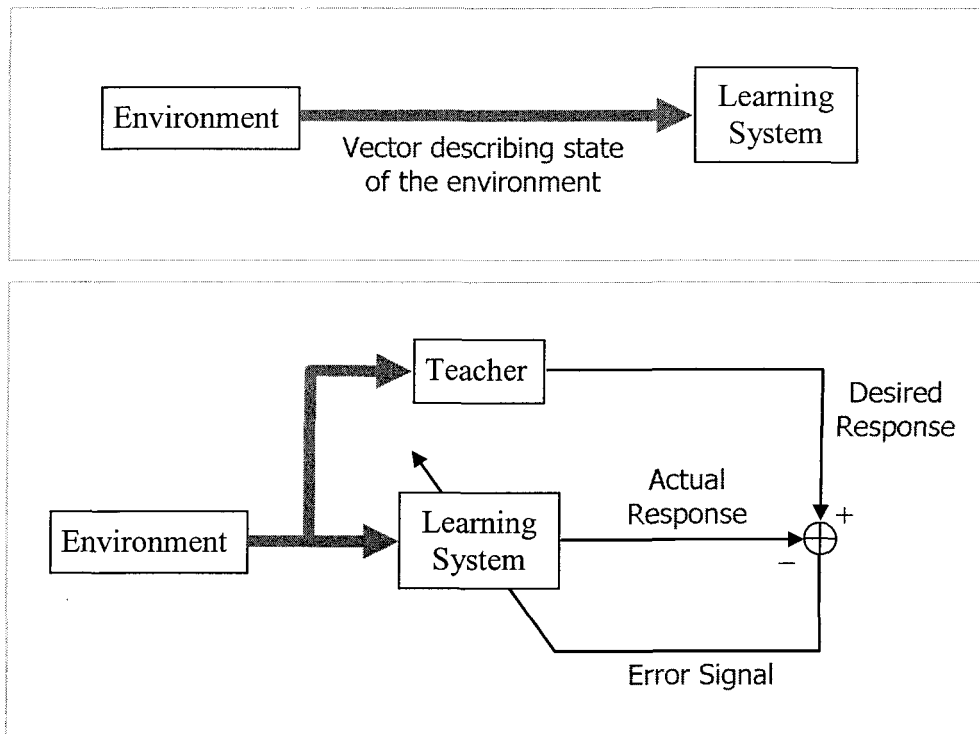
There are different types of training networks. We can categorize them in two distinct sorts, unsupervised and supervised. In unsupervised (self-organization) an output unit is trained to respond to clusters of pattern within the input. In supervised (active) learning the availability of an external teacher is essential. Two kinds of supervised training follow these steps:

- Batch Training:
  - I. Initialize the weights.

- II. Process all training data.
- III. Update the weights.
- IV. Go to II.

- Incremental Training:

- I. Initialize the weights.
- II. Process all training data.
- III. Update the weights.
- IV. Go to II.



**Figure 3-6.** Block diagram of unsupervised learning (top), and supervised learning (bottom)

### 3.2.5.7 Validation of ANNs

When a network is trained, the measure of performance that is optimized is usually the mean square error of the outputs. It is easily computed by summing the squared differences between what a predicted variable should be versus what it actually is, then dividing by the number of the components has gone into the sum. For any input, the output neurons take on an activation level determined by the input and the network. The error for the single presentation of training pattern at  $p_{th}$  step is calculated from [75]:

$$E_p = \frac{1}{m} \sum_{j=0}^{m-1} (T_{pj} - O_{pj})^2 \quad (3-10)$$

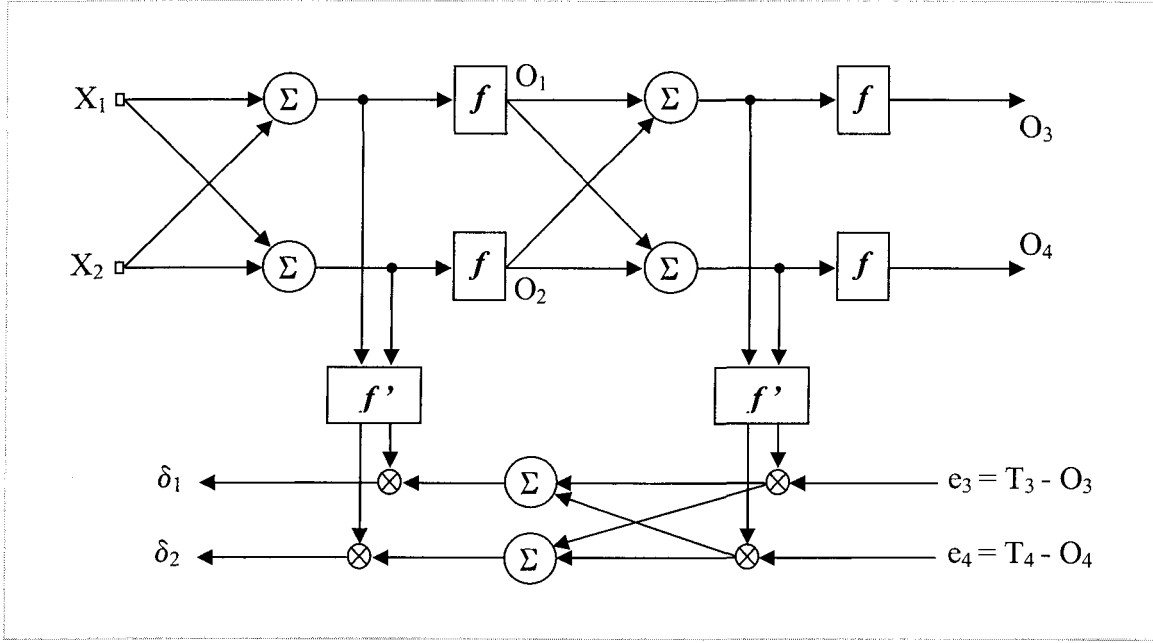
where  $T_{pj}$  means the target activation of the  $j_{th}$  neuron and  $O_{pj}$  denotes the output of neuron  $j$  in the output layer. If there are  $n$  presentations in the epoch, the error for the epoch would be:

$$E = \frac{1}{n} \sum_{p=0}^{n-1} E_p \quad (3-11)$$

As it would be foolhardy to train a network then immediately place it into service, a process of validation is necessary. For this reason, there are different methods offered to test and validate neural networks, all based on limitation the error amount with different definitions.

### 3.2.5.8 Learning algorithm

As we are not studying neural networks learning methods in general let's just talk about back-propagation [76] which we have already used in this work. Back-propagation is a powerful general purpose learning algorithm. A back-propagation network with a single hidden layer of processing elements can model any continuous function to any degree of accuracy (given enough processing elements in the hidden layer). There are literally hundreds of variations of back-propagation in the neural network literature, and all claim to be superior to basic back-propagation in one way or the other. Indeed, since back-propagation is based on a relatively simple form of optimization known as gradient descent, mathematically astute observers soon proposed modifications using more powerful techniques such as conjugate gradient and Newton's methods. However, basic back-propagation is still the most widely used variant. The basic back-propagation algorithm consists of three steps (see Figure 3-7). The input pattern is presented to the input layer of the network. These inputs are propagated through the network until they reach the output units. This forward pass produces the actual or predicted output pattern. As back-propagation is a supervised learning algorithm the desired outputs are given as part of the training vector. The actual network outputs are subtracted from the desired outputs and an error signal is produced. This error signal is then the basis for the back-propagation step, whereby the errors are passed back through the neural network by computing the contribution of each



**Figure 3-7.** Back-propagation architecture

hidden processing unit and deriving the corresponding adjustment needed to produce the correct output. The connection weights are then adjusted and the neural network has just “learned” from an experience. The following calculations should be done to train the network:

$$\Delta w_{ij} = \eta \cdot \delta_{p,j} \cdot O_{p,i} \quad (3-12)$$

$$w_{ij}[n+1] = w_{ij}[n] + \Delta w_{ij} \quad (3-13)$$

where  $w_{ij}[n]$  stands for the weight from neuron  $i$  to neuron  $j$  at  $n_{th}$  step,  $\delta_{p,j}$  is the value of the error factor for the  $p_{th}$  neuron in layer  $j$  and  $O_{p,i}$  is the value of output for  $p_{th}$  neuron in layer  $i$ . During learning, the outputs of a supervised network come to approximate the target values given the inputs in the training set.

In the case of multilayer networks we can train the hidden layers by propagating the output error back through the network layer by layer, adjusting weights at each layer.

Recurrent back propagation is, as the name suggests, a back propagation network with feedback or recurrent connections. Particularly, the feedback is limited to either the hidden layer units (Elman NN) or the output units (Jordan NN). In either configuration, adding feedback from the activation of outputs from the prior pattern introduces a kind of memory to the process. Thus adding recurrent connections to a back propagation network enhances its ability to learn temporal sequences without fundamentally changing the training process. Recurrent back propagation networks will, in general, perform better than regular back propagation networks on time-series prediction problems.

### **3.2.6 Choosing the neural network model**

The combination of topology, learning paradigm (supervised or unsupervised learning), and learning algorithm define a neural network model. There is a wide selection of popular neural network models; some are optimized for fast training, others for fast recall of stored memories, and some others for computing the best possible answer regardless of training or recall time. The best possible model for a given application or data mining function depends on the data and the function required.

In the following chapter the information about the structure of ANN we have used in this work, chosen algorithm for training and the advantages of them will be presented.



**CHAPTER 4**

**REAL-TIME NEURAL NETWORK**

**BASED SENSORLESS DRIVE**

## 4.1 Introduction

Real-time simulation technique is widely used by high-tech industries nowadays, particularly automotive and aeronautics industries (aircraft flight control, satellite control, etc), as the main tool for rapid prototyping of complex engineering systems in a cost-effective and secure manner, while reducing the time-to-market. It's a powerful analytical tool that allows for prediction of system behavior in response to operator actions and events via the use of real-time and archived data. Virtual testing of operator actions prior to implementation can reveal potential problems, hence reducing human errors and the risk of service power system interruptions. Real-time simulation provides an environment that is effective for training, assistance and verification the designed systems. Compared to traditional training methods, it makes the simulation an ongoing process. With real-time simulation, not only it's possible to perform analysis using real-time system parameters, it can also simulate "what if" scenarios simply by taking action using the online system model with minimal risks.

In this chapter what is done in this work using the techniques discussed in the past chapters in order to simulate the real-time speed estimation in the vector controlled induction machine drive is explained. As a short definition the aim is presenting a real-time neural network-based sensorless speed estimation driver for the induction machine which is simulated in the powerful simulator of *Matlab/Simulink*.

The reasons for using the induction machines are discussed in section 1.1. Being a sensorless let the system be aware of the problems of using the speed or position sensor pointed in section 3.1. Neural network based vector control meets the robustness of the controller for the non-linear behavior of the machine cared in 2.3.2 and 3.2.5. As the network is training online, it does not need any pre-computations and is independent to the machine parameters and their undergoing the changes during working.

## **4.2 Simulink Implementation: A Modular Approach**

When an electrical machine is simulated in circuit simulators like PSpice [77], the steady state model is usually used, while the transient behavior has the same importance in induction machine drive studies. The advantage of using Simulink over the circuit simulators is that the transients of electrical machines and drive controls is much more easy in it; as long as the equations are known, any drive or control algorithm can be modeled in Simulink. On the other hand, although the neural networks are presented as the unknown functions but the mathematical computation algorithms are known, so it's possible to simulate the neural network. However, the equations by themselves are not enough and the experiences with differential equation solving and optimizing the models are required.

There are some ready-to-use Simulink induction machine models in the literature [78], which are usually black-box like ones using S-functions with no

internal details [79][80]. S-functions are software source codes (written in *C*, *Fortran* or *Ada*) for Simulink blocks, which normally run faster than discrete Simulink blocks. The problem with using this technique is the limitation in power and ease. Using the *accelerator* functions could be considered if the fast running of the model in a critical need. Also *Simulink Power System Blockset* may be used as another approach. A fundamental problem is that in most cases Power System Blockset models need stiff solvers with a variable step size. However, *Mathwork's Real-Time Workshop* supports only fixed-step integrators. To obtain stability, these would require very small step sizes which cannot be achieved in real time because only very simple models using linear components can be simulated with a fixed-step solver in real-time. This blockset also uses S-functions and has some difficulties in working with other blocks of Simulink.

Another way is using the discrete blocks, which is used in [81] for flux and torque ripple reduction in Direct Torque Control strategy. We've used this method with MRAS technique in Vector Control strategy.

## **4.3 MRAS Speed Estimation Using Neural Networks**

### **4.3.1 Drawbacks of Flux Based MRAS**

It has been proved that an MRAS scheme is very effective in identifying motor speed. The classic MRAS scheme, used by numerous of the speed identification works is shown in figure 3-2. It is based on two different models

producing rotor flux and using the error between the two models results to adapt the system. The main problem in this strategy is that the flux based MRAS require pure integration of sensed variables which leads to problems with initial conditions and drift. To avoid these problems, the pure integrator must be replaced with a high gain low-pass filter. This replacement causes the instability of identification at low speed, which results to weak performance of speed sensorless vector control in common. Here, we use the MRAS scheme presented in [38] which works with two *back emf* outputs obtained from the reference and adaptive models and does not require any integrator. Only differentiators exist in this scheme, which helps system to have a very good performance even in low speeds as long as the stator resistance is known.

#### **4.3.2 Back EMF based MRAS**

Different methods for the MRAS speed estimation have been used during this research; at the beginning the flux was used as the output parameter of the two blocks of reference and adaptive model to adapt the estimator. The drawbacks of this method are enumerated in last sections.

The next structure was using *back emf* as the output parameter and the inputs which were used to adapt the ANN speed estimator, and feeding the ANN with terminal voltage and current of the stator as the inputs. The results of this structure were acceptable, and an article based on this work was presented in the

AIA '2005 conference, Innsbruck, Austria. This article is attached at the end of this thesis as appendix A.

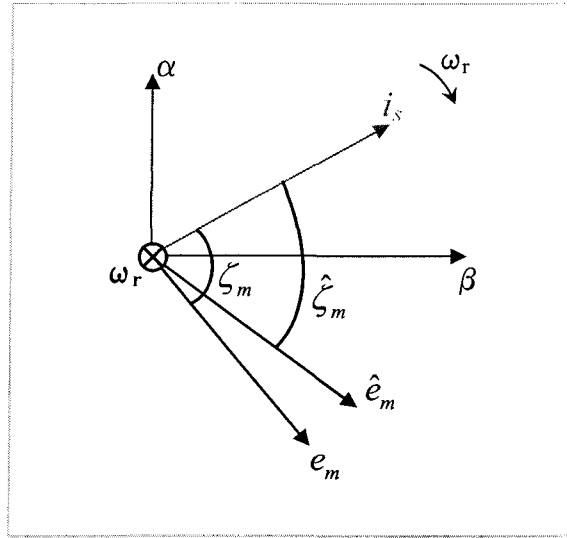
The final structure which has had the best results, is based on using back *emf* in reference and adaptive models, with the same time, an ANN as speed estimator, which is explained in the following.

Induction machine equations can be expressed in the stationary  $\alpha, \beta$  frame as the following

$$\vec{v}_s = R_s \vec{i}_s + \sigma L_s \frac{d\vec{i}_s}{dt} + \vec{e}_m \quad (4-1)$$

$$\frac{d\vec{i}_m}{dt} = \vec{\omega}_r \otimes \vec{i}_m - \frac{1}{T_r} \vec{i}_m + \frac{1}{T_r} \vec{i}_s \quad (4-2)$$

in which  $\vec{\omega}_r$  is defined as a vector whose magnitude is equal to the rotor angular velocity  $\omega_r$  and its direction is determining according to a right-hand system of coordinates as shown in figure 4-1. In this figure  $i_s$  stands for stator current vector,



**Figure 4-1.** Coordinates in stationary reference frame

$e_m$  is the counter EMF vector, and “ $\otimes$ ” denotes the exterior direction from the page for the defined rotor speed vector. Choosing the stationary  $\alpha, \beta$  frame for work has the advantage of freedom from knowing the exact flux angle.

The counter electromotive force  $e_m$  vector is obtained from

$$\vec{e}_m = L'_m \frac{d\vec{i}_m}{dt} \quad (4-3)$$

in which  $L'_m$  is the equivalent mutual inductance. Submitting (4-3) in (4-2) results

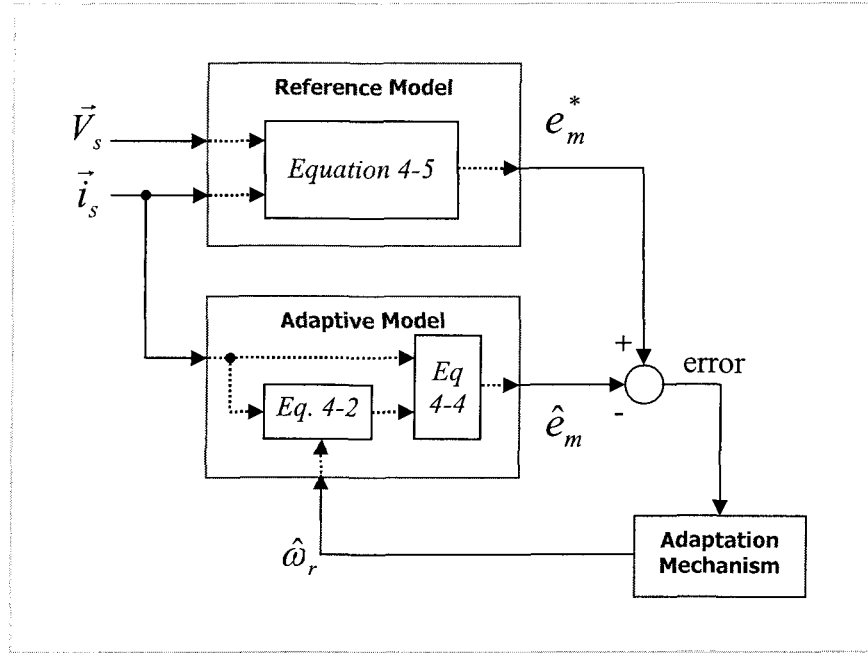
$$\vec{e}_m = L'_m \left( \vec{\omega}_r \otimes \vec{i}_m - \frac{1}{T_r} \vec{i}_m + \frac{1}{T_r} \vec{i}_s \right) \quad (4-4)$$

On the other hand, from (4-1) and (4-2)

$$\vec{e}_m = \vec{v}_s - \left( R_s \vec{i}_s + \sigma L_s \frac{d\vec{i}_s}{dt} \right) \quad (4-5)$$

Using the two recent equations for back *emf* (counter electromotive force) vector in the MRAS scheme for speed estimation results the structure of system as illustrated in figure 4-2.

Two independent observers are used in the figure above to calculate the components of back *emf* vector based on measurable and estimated variables. The one in top is regarded as reference model since doesn't involve  $\omega_r$  or any unknown variable quantity.



**Figure 4-2.** Coordinates in Stationary Reference Frame

#### 4.3.3 Neural Networks as the Adaptation Mechanism

It may be a question that why only MRAS technique is not used, while adding the Neural Networks to the system makes it more complicated. The explanation is that as it's noticed in [36], the system must be stable during the adaptation. In order to meet this term, the speed is considered constant during the adaptation procedure which is an extra supposition. Using the Neural Networks does not need this term and results to a better performance.

On the other hand, the combined method has some more advantages comparing with other works. For example, in [82] a three-layer NN was used with offline training. This technique gives a fairly good estimate of speed and is robust to parameter variation. However, it needs offline training which means pre-



calculations are needed. Also, the estimator should be trained sufficiently with various patterns to reach a good performance. In [83], a two-layer NN was used to estimate induction motor speed with online training, which did not have those problems, but it lied more in the realm of adaptive control than NN. A drawback was that the estimated speed value was not obtained as the output, but at one of the weights. Moreover, only one weight was adjusted in the online training. Therefore, it would be very sensitive to the parameters variation and system noises. The chosen method to use back *emf* based MRAS structure with NN estimator does not face any of these problems.

In order to study the mechanism, defining  $\alpha, \beta$  as stator fixed reference, back *emf* vector can be written as:

$$e_m = e_{m\alpha} + j e_{m\beta} \quad (4-6)$$

the equation (4-5) presented in vectoriel form can be expressed as:

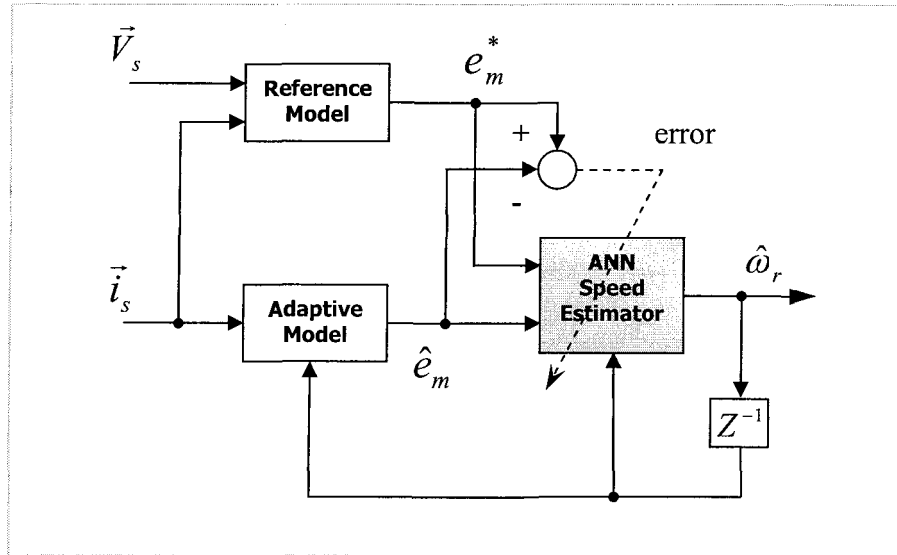
$$\begin{bmatrix} e_{m\alpha} \\ e_{m\beta} \end{bmatrix} = \begin{bmatrix} v_{s\alpha} \\ v_{s\beta} \end{bmatrix} - (R_s + \sigma L_s \cdot p) \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} \quad (4-7)$$

where  $p$  stands for  $d/dt$ . Also from (4-4) the adaptive model equation is written as:

$$\begin{bmatrix} e_{m\alpha} \\ e_{m\beta} \end{bmatrix} = \begin{bmatrix} -\frac{R_r}{L_r} & -\omega_r \\ \omega_r & -\frac{R_r}{L_r} \end{bmatrix} \begin{bmatrix} I e_{m\alpha} \\ I e_{m\beta} \end{bmatrix} + \frac{L_m^2}{L_r^2} R_r \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} \quad (4-8)$$

Using the two recent equations, and considering Neural Networks as the suitable adaptation utility, completes block diagram of the system will be such as

shown in figure 4-3. The structure of the ANN is the well known structure of Jordan recurrent network which is already shown in figure 3-5.



**Figure 4-3.** Structure of Speed Estimator Using NN

#### 4.3.4 Implementation of the Estimator in Simulink

The block diagram of the complete simulation system is given in figure 4-4. It has been preferred to use the Indirect Field Oriented Control method according to its ease of use and the good results.

The inputs and output units will be explained while presenting the results, but before it the simulation blocks are explained in the following.

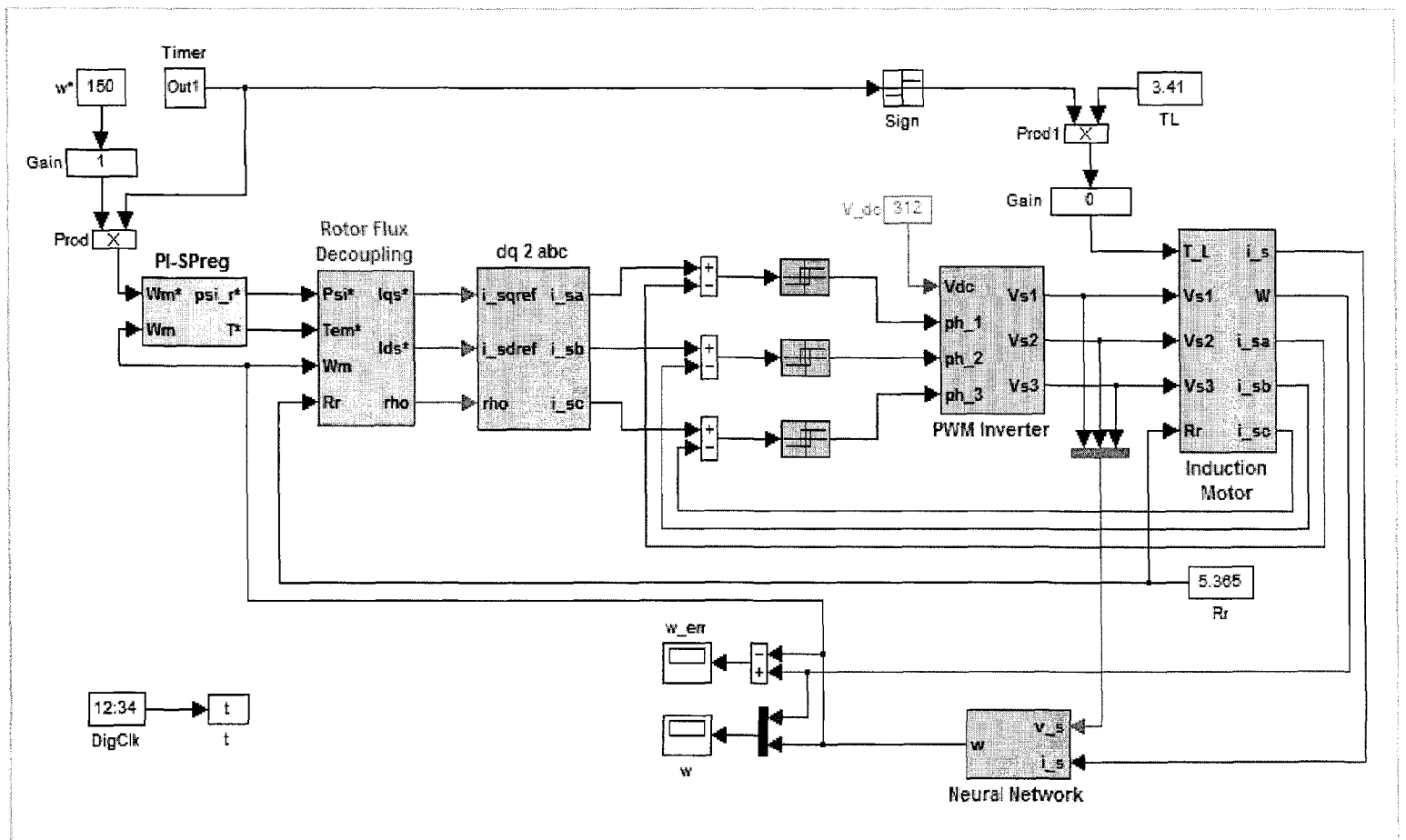
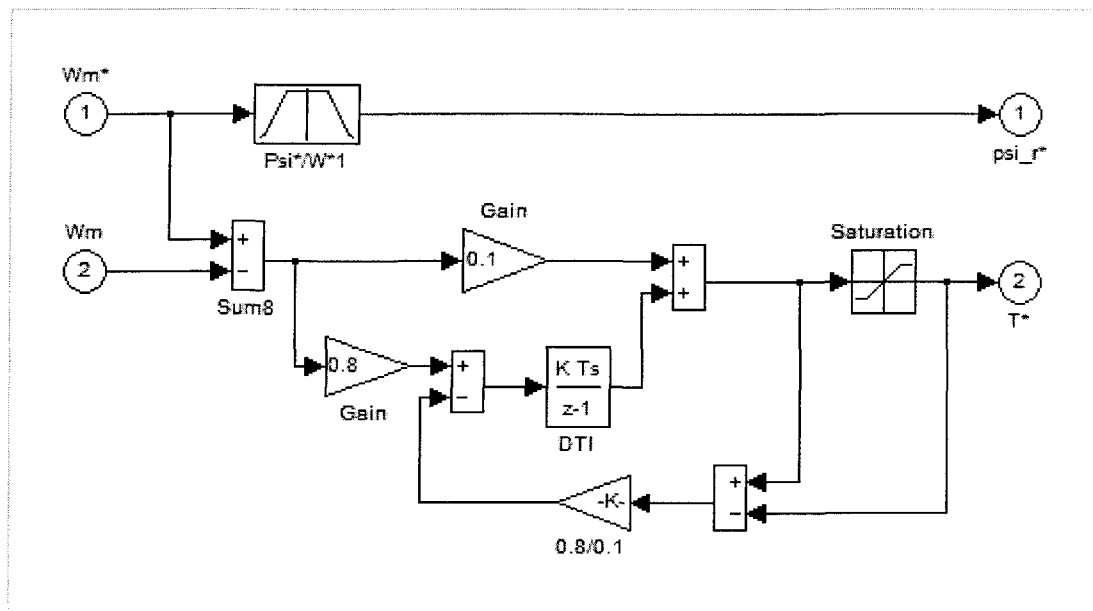


Figure 4-4. Indirect Field Oriented Control with AC Regulator Using NN

#### 4.3.4.1 PI with Speed Regulation Block

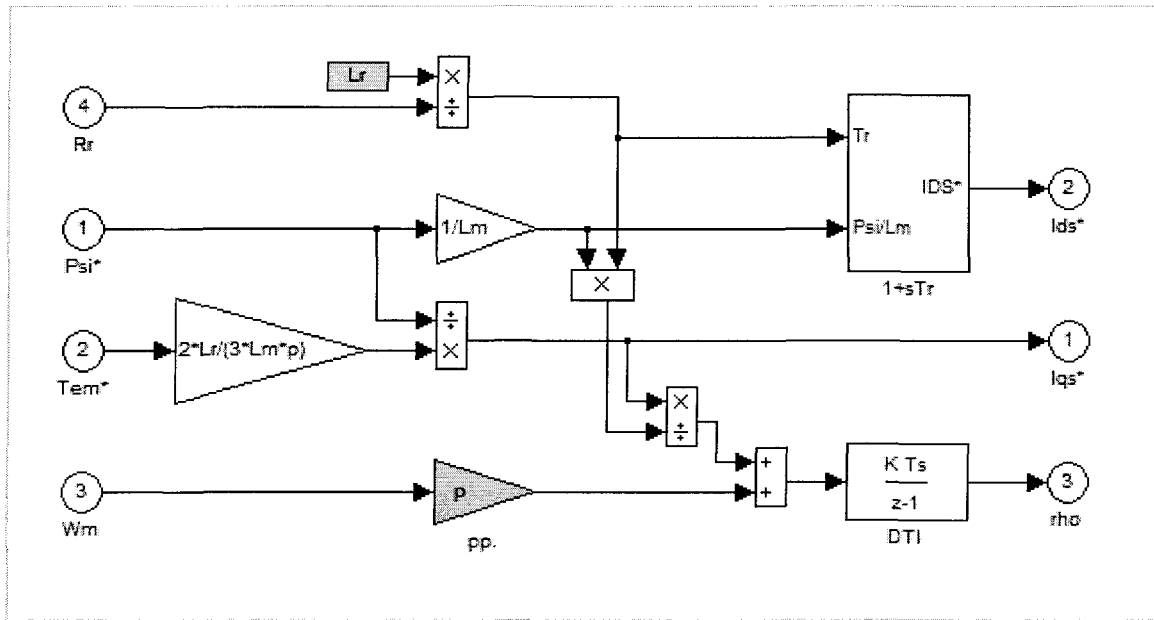
The well-known Proportional-Integral (PI) controller enables exact control for stationary operation by regulating the flux and speed. It integrates the error between the feedback and reference parameters to generate the variable values which are fed to the PWM to produce the gate signals. The second parameter ( $T^*$ ) is producing the wanted  $I_{qs}$  in the following block.



### Figure 4-5. PI Block

#### 4.3.4.2 Rotor Flux Decoupling

This block is generating the outputs following the equations (2-40), (2-41) and (2-42). In the constant flux condition which we are working, the output of the block which is denoted by  $I+sT_r$  is equal to the second input of the block which is presented in (2-40).



**Figure 4-6.**  $d,q$  to  $a,b,c$  Frame Converter

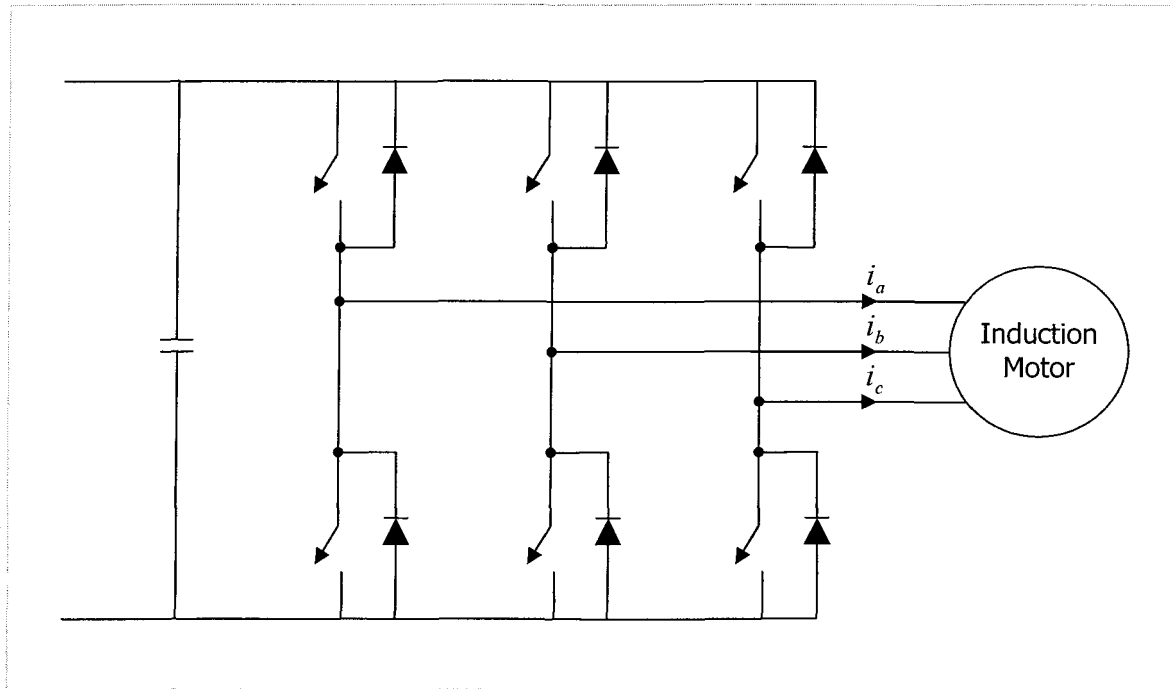
#### 4.3.4.3 $d,q$ to $a,b,c$

The frame conversion  $d,q$  to  $a,b,c$  is done within this block by following the well-known equation (1-35). The calculated current for each phase is compared to the corresponding phase current measured from the induction motor model with a specific threshold in a hysteresis function to switch the three-phase voltage inverter.

#### 4.3.4.4 Inverter

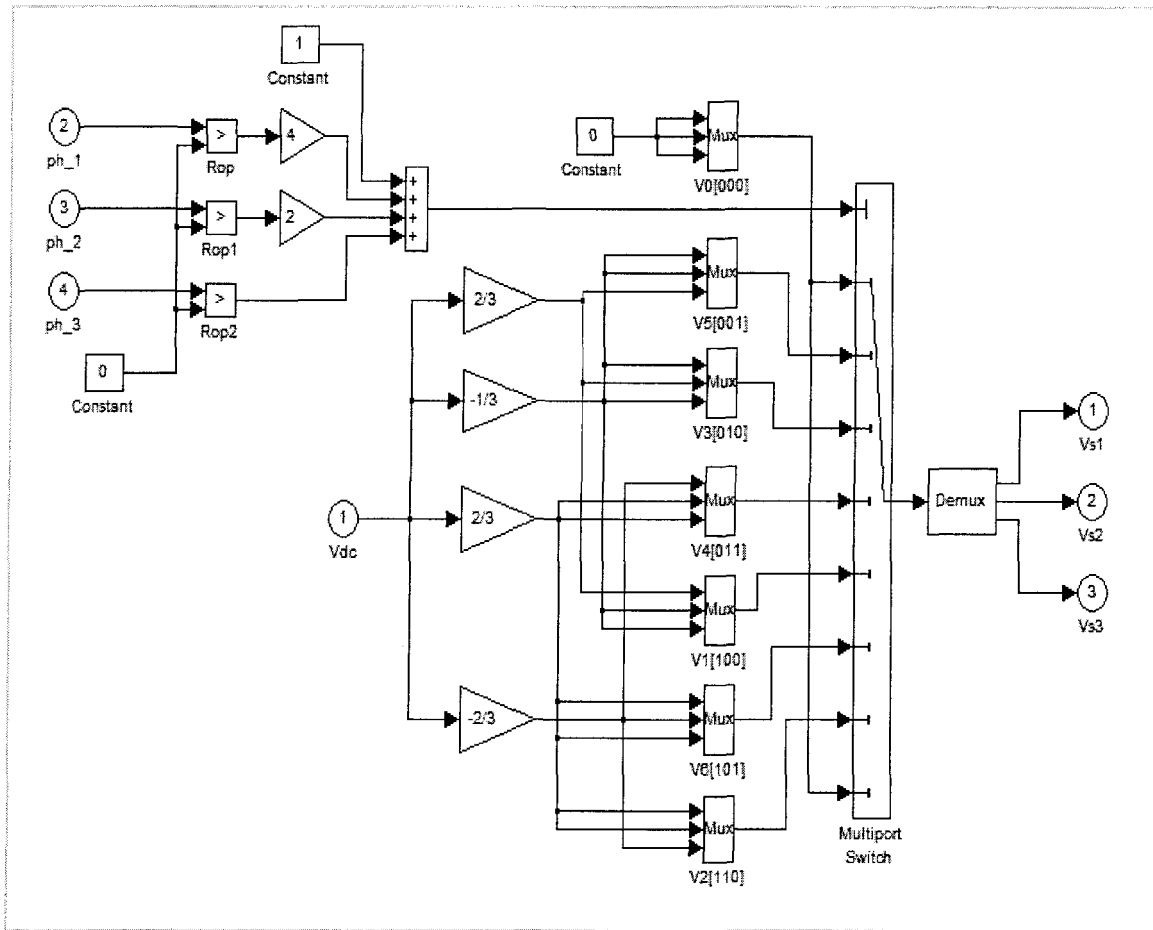
Figure 4-7 shows the schematic of the Pulse Width Modulation drive assuming a three-phase utility input [84]. As a brief review, a PWM inverter controls both the frequency and the magnitude of the voltage output. One of the advantages of PWM is that the signal remains digital all the way from the

processor to the controlled system; no digital to analog conversion is necessary. Also by keeping the signal digital, noise effects are minimized.



**Figure 4-7.** PWM Inverter

The PWM inverter block in Simulink is shown in figure 4-8.



**Figure 4-8. Inverter**

#### 4.3.4.5 Induction Motor

The induction machine model is based on what is presented in chapter 1. The output of the model is in fact performance of the equation (1-53) including the transformation between the frames which is already explained in section 2.3.2. As the model is fully digital, the regular integrators are replaced with *DPZiOP* which represent the “Discrete Zero-Pole with Initial Output” blocks in Simulink. The discrete state space of such a block is given by

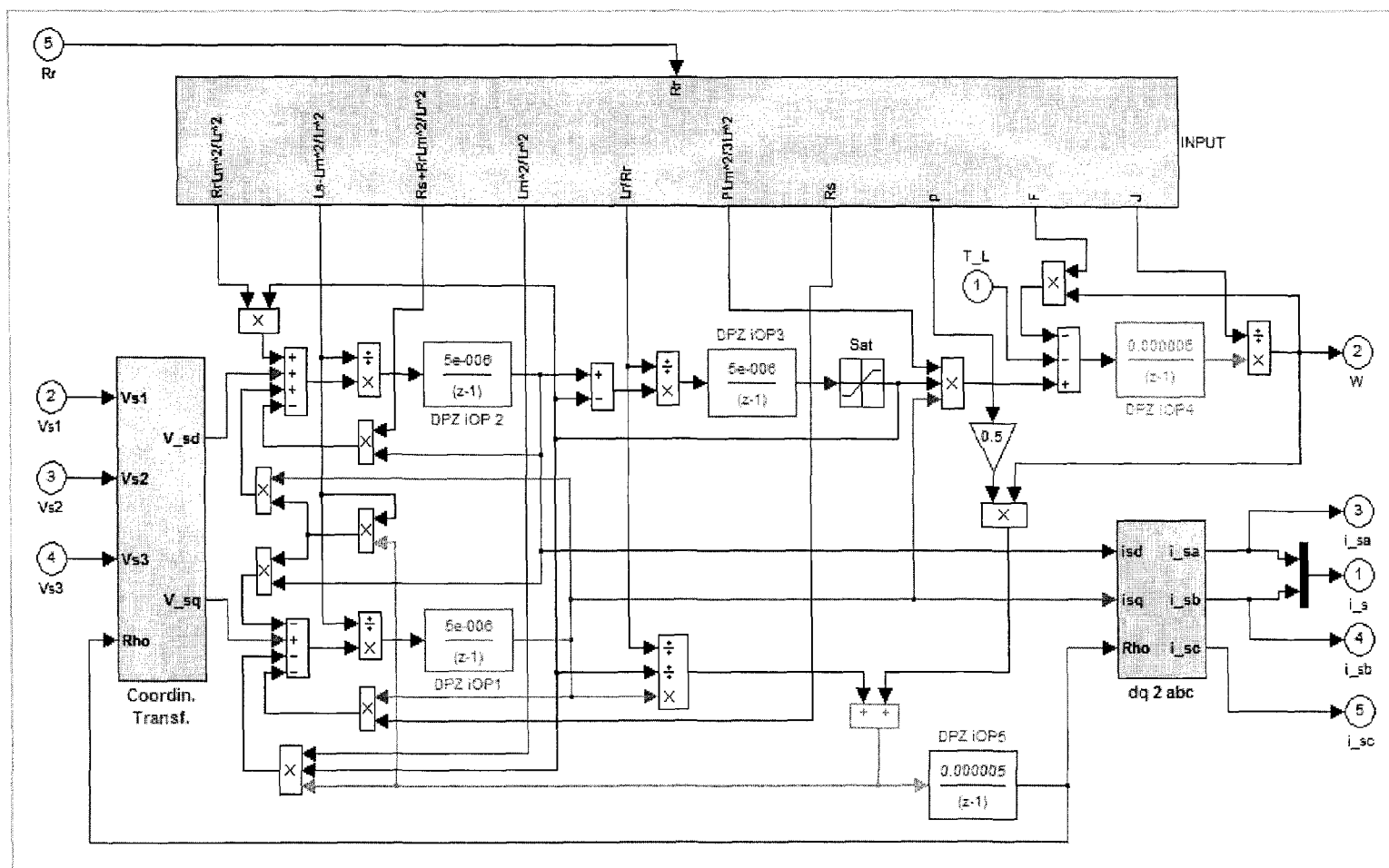


Figure 4-9. Induction Motor Digital Model

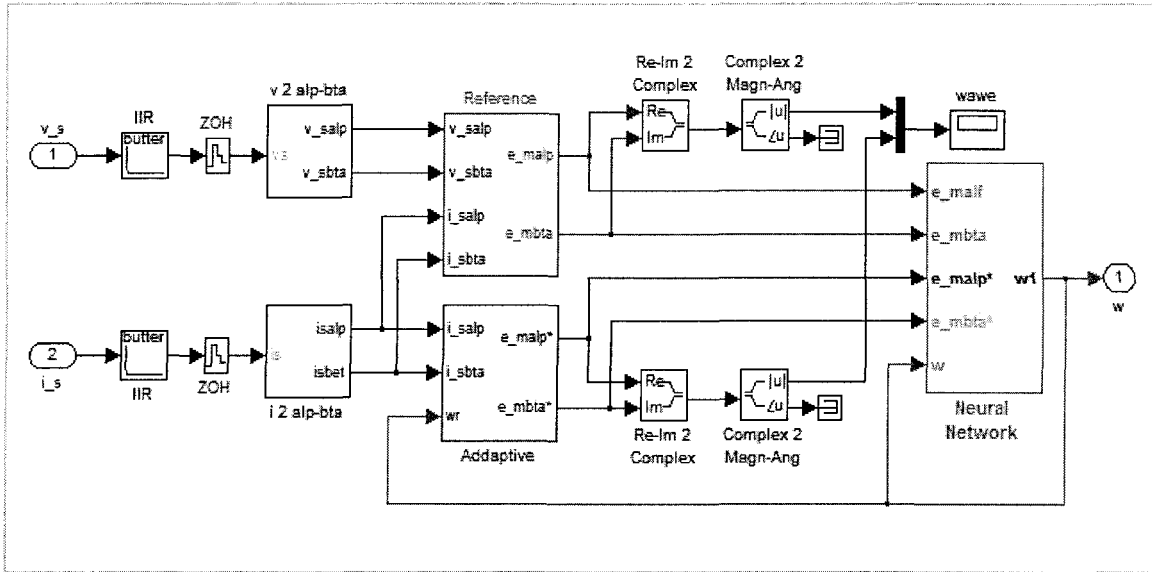


$$x(n+1) = Ax(n) + Bu(n) \quad (4-9)$$

$$y(n) = Cx(n) + Du(n) \quad (4-10)$$

#### 4.3.4.6 Neural Network Based Speed Estimator

According to the explanation of this part in section 4-3 this block must be clearly known. In fact, figure 4-10 shows what is already presented in figure 4-3 in *Simulink* discrete blocks. Only two low pass *Butterworth* [85] digital filters are added on the input. The filters are order 3 with band pass edge frequency of 0.01 which is obtained by try and error in order to filter the noise of the input signals of current and voltage.



**Figure 4-10.** Induction Motor

The structure of Neural Network the well known Jordan recurrent network shown in figure 3-5. The inputs are the two pairs of reference and adaptive back

emf blocks output in the stationary  $\alpha$ - $\beta$  frame. So, the multilayer neural network consists of four scalar inputs of the reference values of back emf ( $e_{m\alpha}$ ,  $e_{m\beta}$ ) and the adaptive values of them ( $e_{m\alpha}^*$ ,  $e_{m\beta}^*$ ), one hidden layer consisting of eight neurons, and one output which is the estimated speed. The estimated speed is used for completing the feedback loop, thus giving rise to a Jordan Neural Network. Implementation of this NN is given in figure 4-11 in which there are four nodes in the input level, eight units in hidden layer (one hidden layer) and one output unit.

Training the network is done online which makes the system be known as real-time. This online training is done by continuously updating of the weights using back-propagation method. To start the training all the weights are initially randomize from -0.5 to 0.5 as  $\alpha$  is set equal to 0.1 and  $\gamma$  is set equal to 0.5 (these two recent parameters are obtained by try and error in several tests to have the best result. Theoretically they can take any amount between 0 and 1 with no exact formula [86]). After that, in each step the estimated speed is used in adaptive model and the error between two models would replace to repeat the calculation. So, the training is completely done online (real-time) with no pre-calculation necessity.

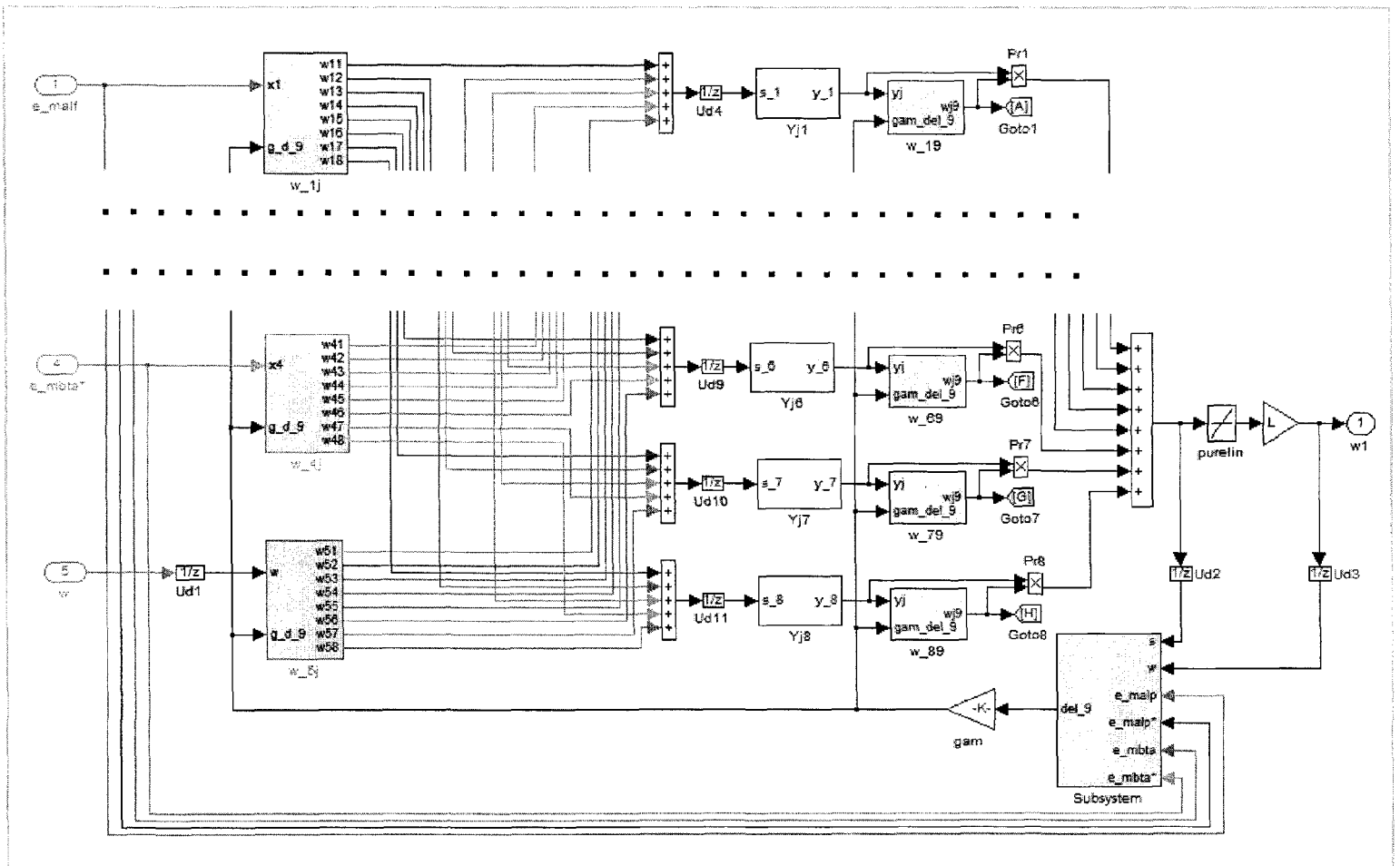


Figure 4-11. Jordan Neural Network

The adjustment of the weights in hidden layer is achieved by using the error of back *emf*. Note that though back *emf* is a vector, but as using the amplitude of it conduces to acceptable result, it's used in order to simplify the calculation. The weights are updated using the equations (3-12) and (3-13) for back-propagation algorithm. Hyperbolic tangent sigmoid transfer function (*tansig*) is chosen as the activation function (discussed in section 3.2.5.4) whose mathematical representation is given in equation (3-8). Recalling equation (3-12) in training algorithm, hard limit transfer function (*hardlim*) is selected for calculation of  $\delta$  which is represented by equation (3-5).

### **4.3.5 Real-Time Simulation**

#### **4.3.5.1 Real-Time Systems**

Real-time can refer to the events simulated by a system (for example, a computer) at the same speed they would occur in real life. As an example, in graphic animation, a real-time program would display the objects moving across the screen at the same speed they would actually move. In reference to embedded systems, real-time denotes the required stability of the embedded system. In a real-time system, the embedded device is given a predetermined amount of time, such as *1ms*, *5ms*, or *20ms* to read input signals, such as sensors, to perform all necessary calculations, such as control algorithms, and to write all outputs, such as control actuators, and control fuel flow. Most general-purpose systems are not

real-time because they take a few seconds, or even minutes, to react. Thanks to the speed of new processors, it's possible nowadays to implement a real-time operation in simulation and control systems.

Typically, a real-time system consists of a controlling system and a controlled system. In this type of systems the correctness of the system not only depends on the logical results of the computations, but also depends on the time at which the results are produced. The controlling system interacting with a real-world plant is based on the information obtained from various sensors and inputs that measure the actual state of the plant. The information presented to the controller must be consistent with the actual state of the environment, or the actions of the controlling system can be disasters.

#### **4.3.5.2 Real-Time Systems Requirements**

In most of the real-time systems, severe consequences result if the timing and logical correctness requirements of the system are not satisfied. The requirements on a real-time-capable system can be attributed to three basic concepts: data throughput, responsiveness and determinism. Data throughput is the speed which the data is recorded and processed with. Responsiveness denotes the time-delay between the occurrence of an event and the availability of the corresponding output data. Deterministic system behavior requires that the input data occurring at a determined rate can be processed online without any data being lost [87]. The system must react to the events in a predictable manner under all

operating conditions and be able to complete the corresponding calculations within the guaranteed response time.

A real-time system must be able to handle time constraints which are commonly divided to two categories: hard real-time systems and soft real-time systems. In soft real-time systems, time correctness is important but not critical while in hard real-time systems it is critically important and may not be sacrificed for other gains. Computer control in hard real-time systems is becoming common in a wide range of uses, especially in safety critical systems. Timing is no longer a node internal issue since each node has to synchronize its actions with other nodes regarding their time limitations [88].

Therefore, being real-time does not mean for a system, the data is processed as quickly as it's possible according to the used equipments. It rather means that the defined time limits are observed, whereas the absolute limitation values depend on each individual application. The goal of a real-time operation system is to manage the resources of a system in accordance with the above mentioned retirements [89][90].

Though the discrete blocks (containing the induction motor model, driver, etc.) used in the speed estimator implementation in Simulink are all fully digital models, the step time for each part and also the needed delay blocks (to be added between the blocks) in the model must be chosen according to above. Setting the step time so small may make the system behavior very similar to the real (analog) environment, but as hardware implementing of the system is technologically

limited by the hardware (such as CPU time). So, the step time can be increased until the results have not been affected by this. Finally, the step time value of  $5\mu sec$  is what is used in the following results, but there is no matter if we change it in a range of 10 times (smaller or larger) which is nowadays a possible to reach for the hardware-in-the-loop implementation.

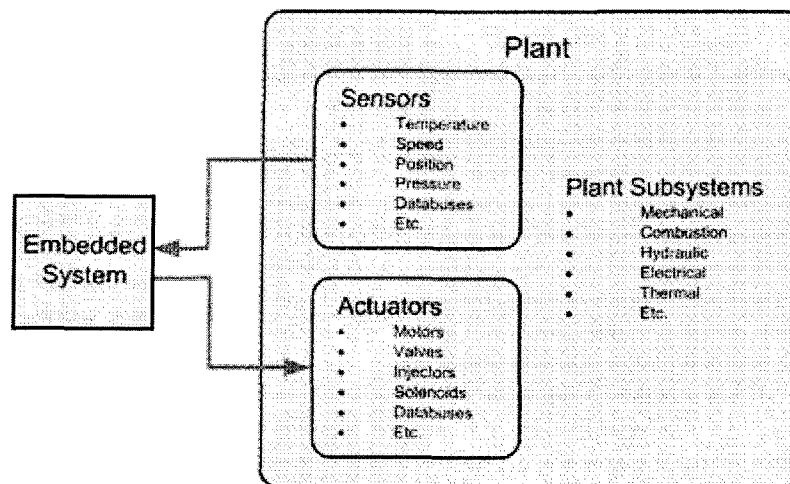
#### **4.3.5.3 Future development: HIL Implementation**

Embedded systems are designed to control complex plants such as land vehicles, satellites, spacecrafts, Unmanned Aerial Vehicles, aircrafts, weapon systems, marine vehicles, and jet engines. They generally require a high level of complexity within the embedded system to manage the complexity of the plant under control [91]. The future goal for this work could be the implementation of real-time control applied on the station model using an actual station governor, giving so-called *Hardware-in-the-Loop Simulation (HIL)*.

Hardware-in-the-Loop simulation is a technique that is used increasingly in the development and test of complex real-time embedded systems in a comprehensive, cost-effective, and repeatable manner [92][93]. It is most often used in the development and testing of embedded systems, when those systems can not be tested easily, and repeatable in their operational environment. The purpose of HIL simulation is to provide an effective platform for developing and testing real-time embedded systems. HIL simulation provides an effective platform by adding the complexity of the plant under control to the test platform.

The complexity of the plant under control is included in test and development by adding a mathematical representation of all related dynamic systems. These mathematical representations are referred to as the *plant simulation*.

General Architecture for an Embedded System controlling a Plant is given in figure 4-12.



**Figure 4-12.** Embedded System controlling a Plant

In fact, the metric of development and test efficiency is typically a formula that includes the following factors: cost, duration and safety. Cost of the approach will be a measure of the cost of all tools and effort. The duration of development and test affects the time-to-market for a planned product. The safety factor and duration are typically equated to a cost measure. Specific conditions that warrant the use of HIL simulation include the following:

- Tight development schedules

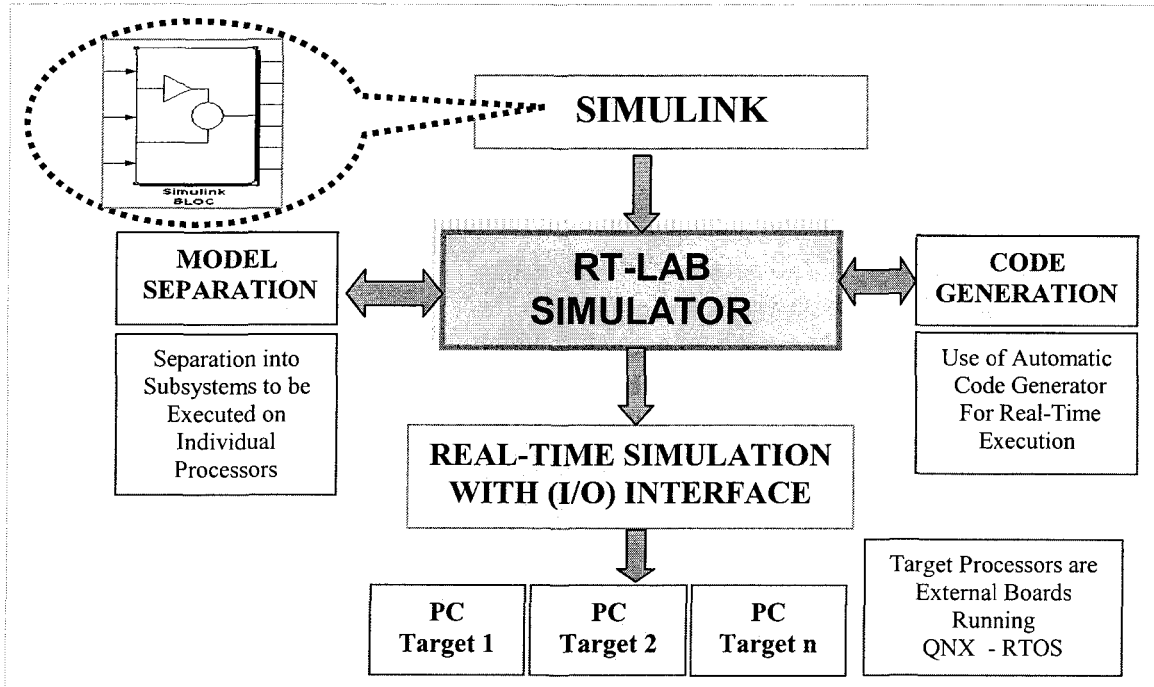


- High-burden-rate plant
- Early process human factors development

HIL simulation requires the development of a real-time simulation that models some parts of the embedded system under test and all significant interactions with its operational environment.

As for the next step of this research work, it's considered to implement the current fully-digital real-time simulation in a HIL system. It's believed that the results which are obtained from the offline simulation in the *Matlab/Simulink* package can be repeated in such a platform according to what is observed in current work. The *Electric Machines Identification and Control Laboratory (EMICLab-UQAC)* is equipped to the RT-Lab simulator package [94], and the hardware equipments which allow the research and development in this area. RT-Lab real-time, distributed PC-based simulation platform is optimized to run *Simulink* and the *Power Systems Blockset* in real-time, with efficient fixed-step solvers, on PC cluster. The simulation model provides all the process signals in real-time that are next converted by D/A modules and supplied to the controller as voltages. The control signals are produced by the controller and supplied via A/D converters to the simulation model [95]. The simulation hardware can be used for fully digital simulation or for hardware-in-the-loop applications where either the simulated electric plant or simulated controller is connected to external hardware by different types of I/O available in Simulink from extensive I/O library.

The structure of the RT-LAB based real-time simulation for the Simulink model is given below in figure 4-13.



**Figure 4-13.** Conversion of Simulink model to real-time simulation

#### 4.3.6 Simulation Results

The parameters of the induction motor used for the simulation are given below in Table 4-1.

Real-Time digital simulation is carried out using *Matlab/Simulink* in order to verify the accuracy of the estimation algorithm in addition to observing the response of the sensorless vector control drive system. Simulation is carried out

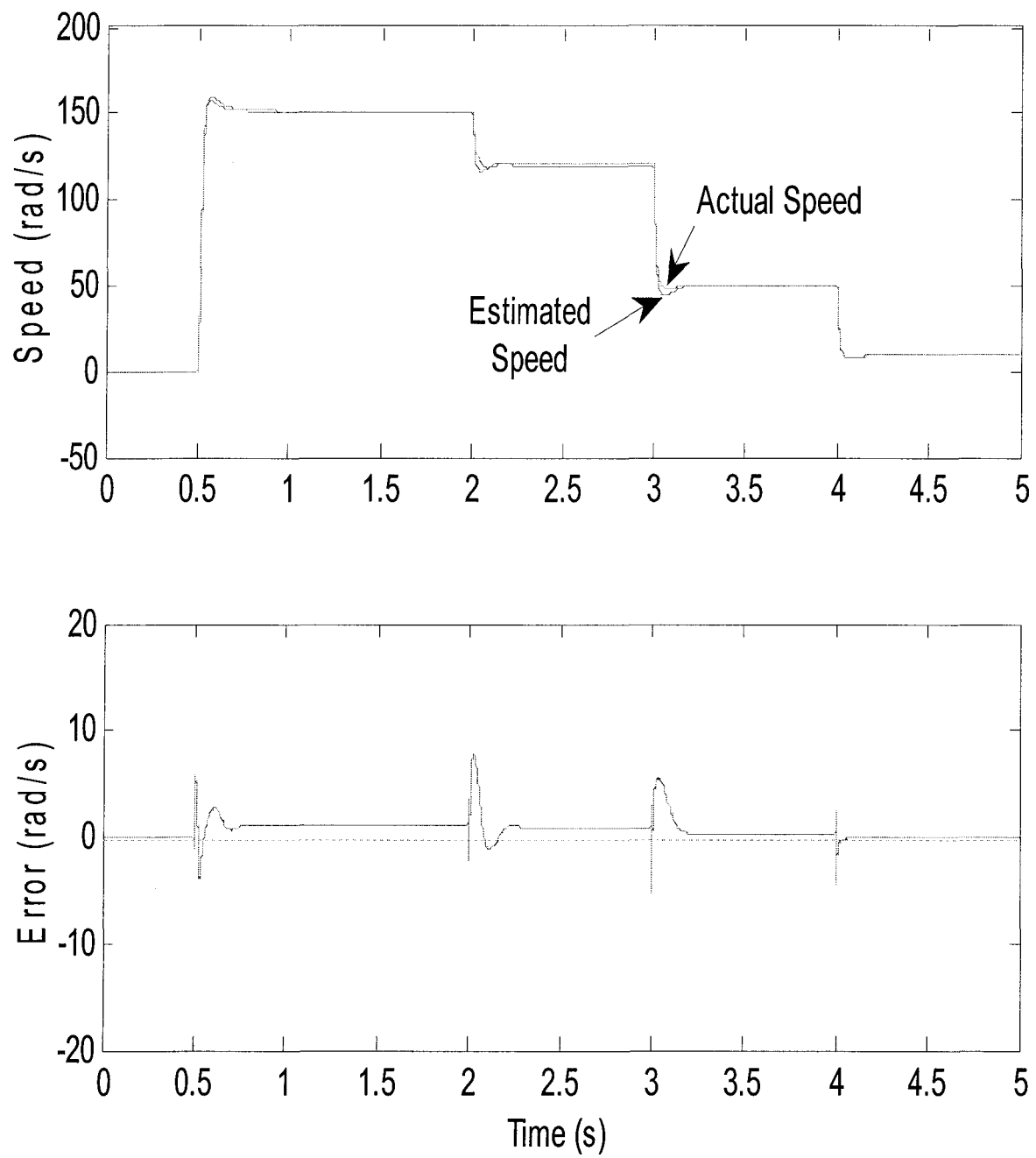
Related Power	$Pr$	500 W
Line-Line Voltage	$Vr$	220 V
Related Torque	$T$	3.41 N m
Number of Poles	$P$	4
Stator Resistance	$Rs$	4.495 $\Omega$
Rotor Resistance	$Rr$	5.365 $\Omega$
Stator Inductance	$Ls$	165 mH
Rotor Inductance	$Lr$	162 mH
Magnetizing Inductance	$Lm$	149 mH
Rotor Moment of Inertia	$J$	.00095 Kg m <sup>2</sup>

**Table 4-1.** Induction Motor Parameters

for different operating conditions of the motor drive to study the performance of the NN speed estimator.

First, the machine is operated at no load. Figure 4-14 shows the real speed, estimated speed, and the speed estimation error for this operation. At  $t = 0.5$  sec the machine is accelerated from zero to  $150 \text{ rad/s}$  and then, decelerated in steps to  $120 \text{ rad/s}$ ,  $50 \text{ rad/s}$  and  $10 \text{ rad/s}$  at  $t = 2.0 \text{ sec}$ ,  $t = 3.0 \text{ sec}$  and  $t = 4.0 \text{ sec}$  respectively.

Studying the response shows that there is an error of less than 1% in the estimated speed in the steady states. However, trying to decrease it (by changing the training parameters) causes some instability in the output. The error of the estimation is completely acceptable, in both steady states and during the changes, noting that

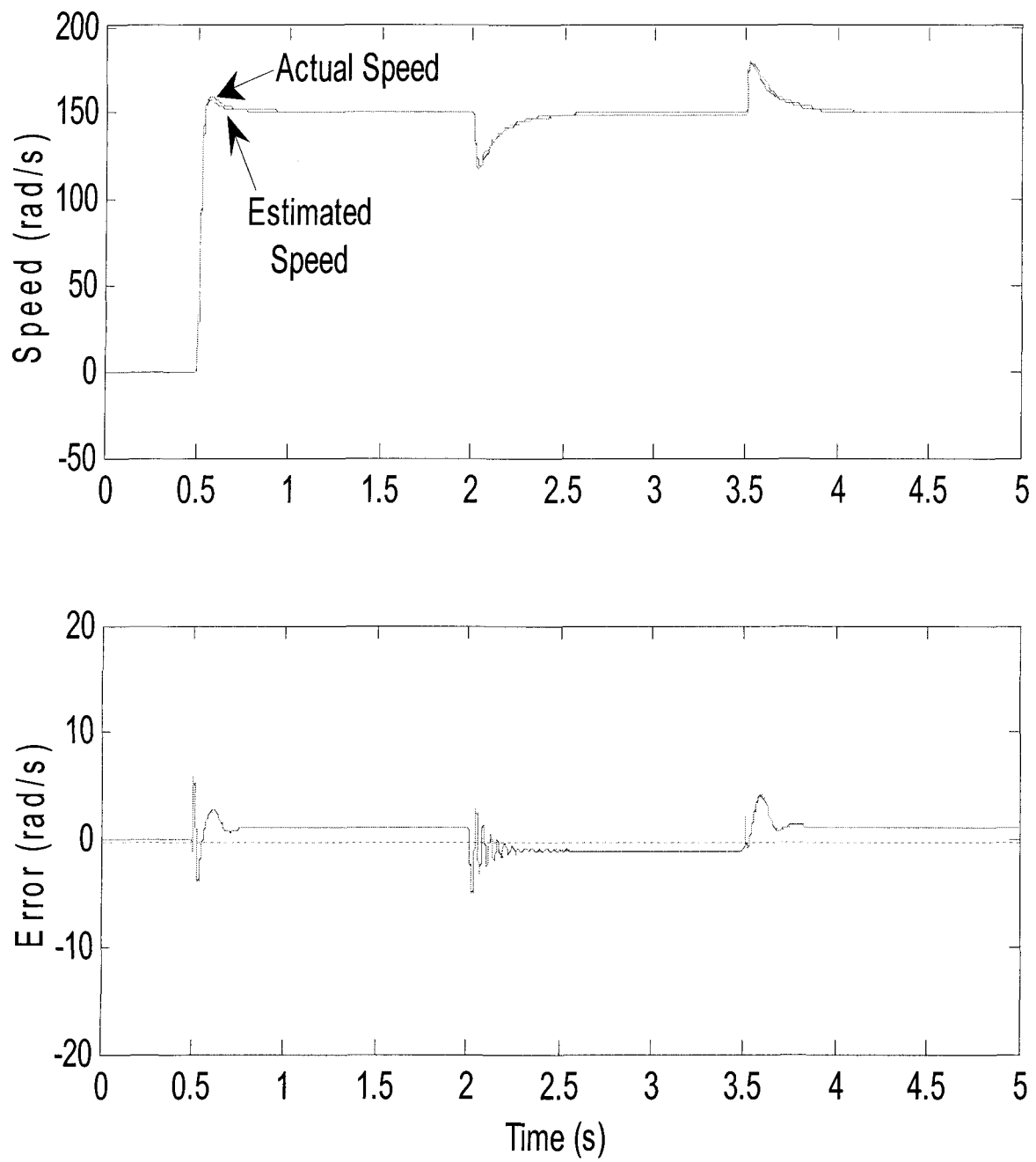


**Figure 4-13.** No-load Operation Response of Speed estimation System

the step function normally causes the most instability in systems. Also note as it's been explained in section 4.3.1 the performance of the estimator in low speeds is very good, unlike the MRAS schema which works based on two rotor flux produce.

In next operation, the performance of the estimator while the machine is loaded and unloaded is studied. The machine is first accelerated to  $150 \text{ rad/s}$  at  $t = 0.5 \text{ sec}$  and then full load is applied at  $t = 2.0 \text{ sec}$  and then, the load is again fully removed at  $t = 3.5 \text{ sec}$ .

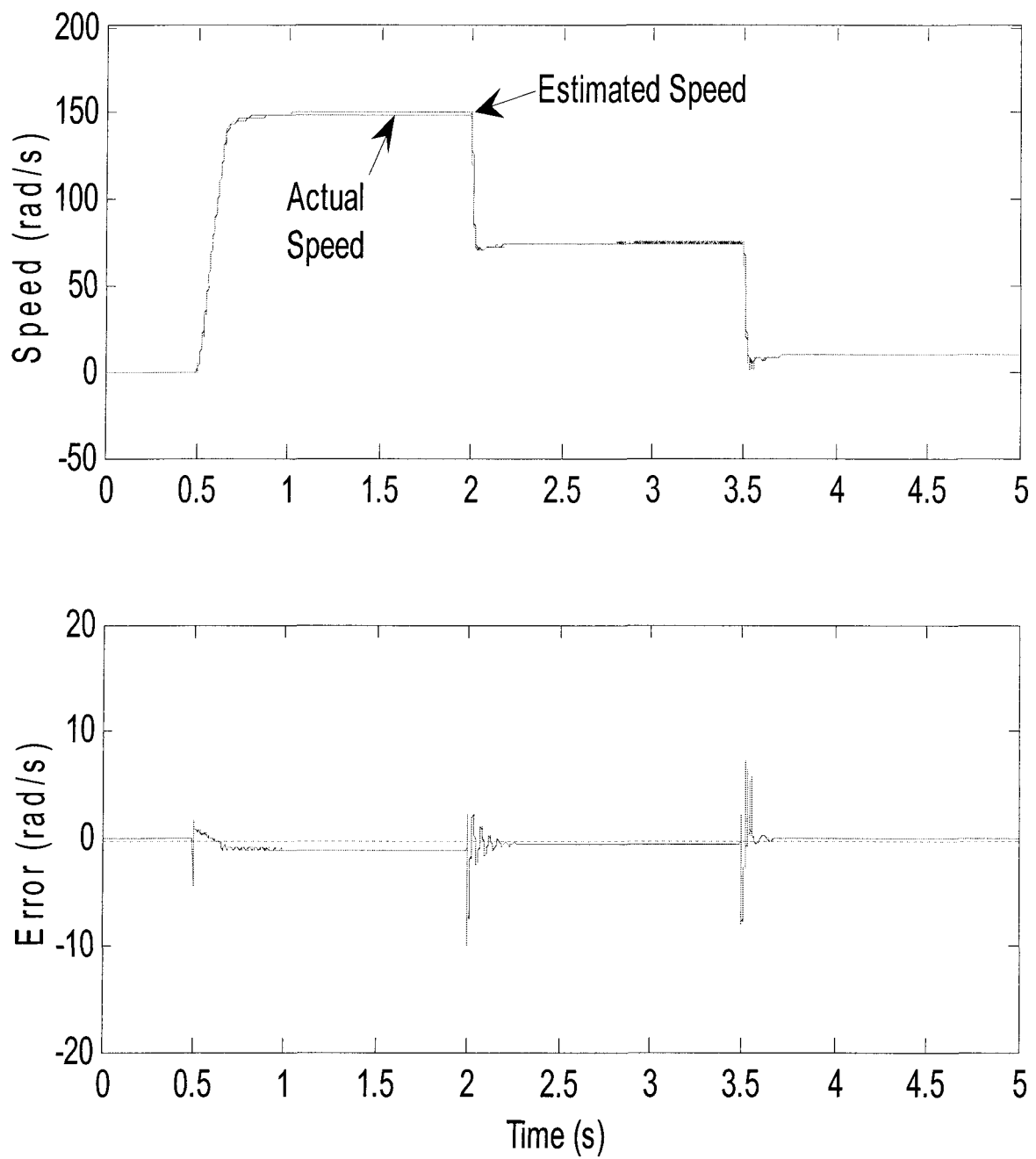
The speed estimated speed and the speed estimation error are shown in figure 4-15. It is observed that the estimated speed tracks the actual speed very well with a small error of less than 1% in steady states. On the other hand the transient time is short enough for such a system. No oscillation in result is seen during different tests.



**Figure 4-14.** Speed Estimation for Load Changes

As another operation, the performance of the fully loaded drive system at various operating speeds is studied. The fully loaded motor is started under load at  $t = 0.5\text{sec}$  to the speed of  $150\text{ rad/s}$  and after the speed gets stable it is decelerated in steps to  $75\text{ rad/s}$  at  $t = 2.0\text{sec}$  and finally to  $10\text{ rad/s}$  at  $t = 3.5\text{sec}$ .

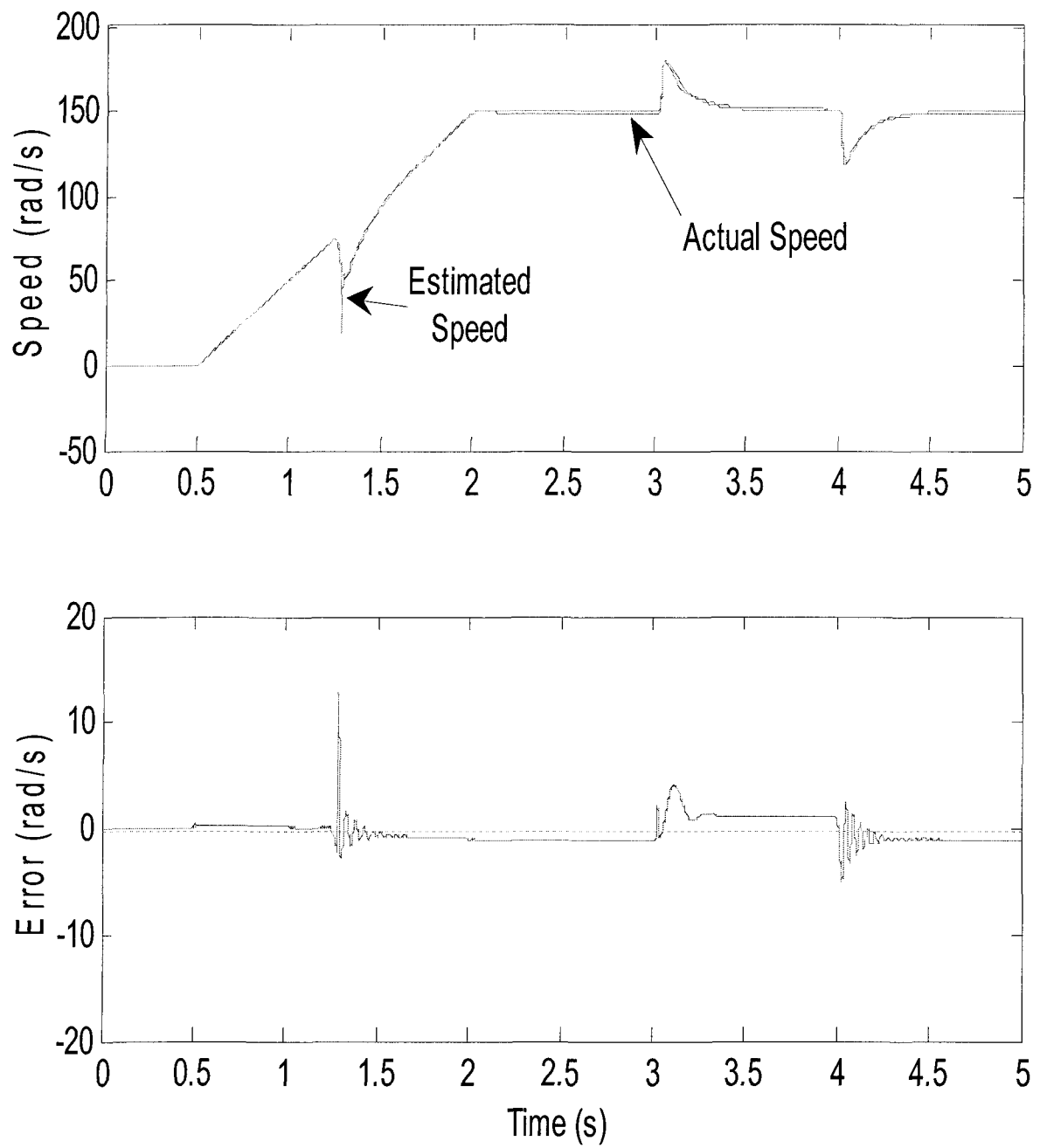
The speed, estimated speed and the speed estimation error are shown in figure 4-16. The accuracy of estimation was found to be very good with a small error of less than 1% under steady state conditions and no instability is experienced during the changes.



**Figure 4-16.** Full-loaded Machine Speed Estimation



As a final test result, the performance of the estimator while the motor is loaded and unloaded during acceleration is studied. The motor is accelerated with a constant slope of  $2/3$  from 0 to 150 rad/s starting at  $t = 0.5\text{sec}$ . In the middle of its way in the accelerating period, at  $t = 1.25\text{sec}$ , the motor fully loaded. The speed would reach the maximum value of 150 rad/s at  $t = 2\text{sec}$ . Then after getting stable, at the moment of  $t = 3\text{sec}$  it's completely unloaded, and then at  $t = 4\text{sec}$  full load is applied again, this time at maximum speed. The estimated speed is compared with the reference speed for this operation in 4-17. The accuracy of estimation was again found to be very good with a small error of less than 1% under steady state conditions and no instability is experienced during the changes.



**Figure 4-17.** Speed Estimation in Combined Operation

Some more tests are done in different terms, which no one has shown results weaker than the presented ones. So, it is found that the NN speed estimator has good accuracies under both transient and steady state conditions, at high and low speeds and also at no load and loaded conditions of the speed sensorless vector control induction motor drive system.

## **CHAPTER 5**

## **CONCLUSION**

## 5.1 Contribution

Induction machines have a very important role in industry nowadays. The weakness of open-loop controllers and all the economic and implementation problems of using the available closed-loop speed control systems have encouraged a lot of researchers to work on finding a robust speed estimation method to work in a sensorless vector control system.

The problems of using every of flux based MRAS and other studied methods leaded us to a new neural network speed estimator with a back *emf* based MRAS structure where the reference model is independent of speed, and the adaptive model needs speed information which it obtains from the NN estimator. Thanks to the non-functional nature of the neural networks it has been possible to use them for the induction machines which have a non-linear behavior with some non-measurable variables.

It's believed that this structure is new, and the results show the system robust enough in a wide range of speed and machine load for both transient and steady state. No pre-computation is needed and the adaptation of the speed estimator is done online, as the network is designed to work in real-time. Therefore, the system has independence to the knowledge of some machine parameters or their variation during working which meets the circumstantial necessities for such a system. So, this achievement is considered to be an important contribution to sensorless operation of induction machine drives.

## 5.2 Future Topics

As the real-time digital simulation results show the method is capable of accurate speed estimation under different conditions, it's believed the system can be implemented as hardware-in-the-loop to prove the results in experiment. The HIL implementation could be possible thanks to the technology of DSPs or the PC-based platforms according to the step time calculation and considering the most important expressions in the field of real-time data processing. More progress on the system then may be considered as it's implemented as a real-time task.

**APPENDIX A**  
**A CONFERENCE ARTICLE**

**Proceedings of the IASTED International Conference on  
Artificial Intelligence and Applications  
AIA'2005, Innsbruck, Austria**

# Neural Network Based Real-Time Speed Estimation in an Induction Motor Sensorless Drive

Farhad Haghighoeian

Mohand A. Ouhrouche

Jogendra S. Thongam

mouhrouc@uqac.ca

Department of Applied Science

University of Quebec at Chicoutimi

555, Blvd. de l'Université, Chicoutimi, QC, CANADA G7H 2B1

**Abstract** -- This paper presents a sensorless speed estimation method of an induction motor using recurrent neural networks in an indirect field-oriented control system. The measured voltages and currents of the motor are used as the inputs of the neural network, while the feedback of the estimated speed realizes the structure of a Jordan network. The proposed network is trained online by using backpropagation algorithm and the error between electromotive field voltage obtained from this estimated speed (adaptive model) and the one calculated directly from voltages and currents (reference model) is used for training. This method achieves a wide bandwidth of speed control and is verified using real-time simulation using 'Matlab-Simulink' on PC platform.

**Keywords:** Induction Motor, Neural Networks, Vector Control, Sensorless Speed Estimation.

## I. Introduction

Vector Control (VC) is the most widely used method for controlling induction motors in high performance drive applications [1]. The rotor speed information which is necessary to realize high-performance and high-precision speed control of induction motors in a closed loop system is usually measured by using speed sensors. These sensors are usually expensive and bulky. Therefore, the cost and size of the drive systems are increased [2]. A sensorless drive is another possibility in which the control technique is based on a mathematical representation of the motor [3], estimating the feedback speed by feeding the model with measurements of stator voltage and current. This can essentially reduce the cost and complexity of the drive system. Many researches have been done on modeling the induction motor, based on using "Model Reference Adaptive System",

"Extended Kalman Filter" or "Artificial Neural Networks".

Using ANNs in model has advantages of extremely fast parallel computing, immunity from input harmonic ripples, and fault tolerance characteristics [4]. Some researchers have used NN with offline training [5][6][7], but in the online solution, the neural network seems to be more robust towards load and parameter changes [8].

In this paper, a speed estimation method of the induction motor is proposed in which a multilayer Jordan neural network with 4 inputs, one hidden layer consisting of 8 neurons and a feedback of the output speed is used. The weights of the neurons are continuously updated by using backpropagation method. We've used the MRAS speed identification method based on calculating electromotive field voltage [9] in  $\alpha\beta$  frame, and the error between the emf voltage obtained from the reference model with the adaptive (NN) model is used to modify the neurons weights. This achievement is believed to be an important contribution to sensorless operating of electric drives.

## II. MRAS Speed Estimation using NN

It has been proved that an MRAS scheme is very effective in identifying motor speed [10]. The MRAS scheme for speed identification without integrators can be express in the stationary  $\alpha\beta$  frame as below [9]:

$$\vec{v}_s = R_s \vec{i}_s + \sigma L_s \frac{d\vec{i}_s}{dt} + \vec{e}_m \quad (1)$$

$$\frac{d\vec{i}_m}{dt} = \vec{\omega}_r \otimes \vec{i}_m - \frac{1}{T_r} \vec{i}_m + \frac{1}{T_r} \vec{i}_s \quad (2)$$



where  $\vec{v}_s$  and  $\vec{i}_s$  are the stator voltage and current vectors respectively,  $\vec{i}_r$  and  $\vec{i}_m$  are rotor and magnetized current vectors respectively,  $\vec{e}_m$  is counter EMF vector,  $L_s$ ,  $L_r$  and  $L_m$  are stator, rotor and mutual inductances respectively,  $\sigma$  is leakage coefficient,  $R_s$  is stator resistance,  $T_r$  is rotor circuit time constant and  $\vec{\omega}_r$  is a vector whose magnitude  $\omega_r$  is rotor electrical angular velocity.

From (1) and (2),  $\vec{e}_m$  can be delivered as followed:

$$\vec{e}_m = \vec{v}_s - \left( R_s \vec{i}_s + \sigma L_s \frac{d\vec{i}_s}{dt} \right) \quad (3)$$

$$\vec{e}_m = \frac{L_m^2}{L_r} \left( \vec{\omega}_r \otimes \vec{i}_m - \frac{1}{T_r} \vec{i}_m + \frac{1}{T_r} \vec{i}_s \right) \quad (4)$$

Defining  $\alpha$ ,  $\beta$  as stator fixed reference, and  $e_m = e_{m\alpha} + je_{m\beta}$ , using equation (3) derives the emf for the reference model below:

$$\begin{bmatrix} e_{m\alpha} \\ e_{m\beta} \end{bmatrix} = \begin{bmatrix} v_{s\alpha} \\ v_{s\beta} \end{bmatrix} - (R_s + \sigma L_s \cdot p) \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} \quad (5)$$

where  $p$  stands for  $d/dt$

By using (2) and (4) the emf for the adaptive model can be derived as below:

$$\begin{bmatrix} e_{m\alpha} \\ e_{m\beta} \end{bmatrix} = \begin{bmatrix} -\frac{R_r}{L_r} & -\omega_r \\ \omega_r & -\frac{R_r}{L_r} \end{bmatrix} \begin{bmatrix} Ie_{m\alpha} \\ Ie_{m\beta} \end{bmatrix} + \frac{L_m^2}{L_r^2} R_r \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} \quad (6)$$

Fig. 1 illustrates the structure of the proposed speed estimator. In the Reference Model block we are using equation (5) to find out the emf voltage independent of the motor speed ( $\omega_r$ ) as in Adaptive Model block the current and the estimated speed (obtained from our ANN) are used for that. The error between these two blocks then is used to adjust the weights of the neurons in the speed estimator. The bigger error between the two emf model outputs, the more correction in the weights is exerted.

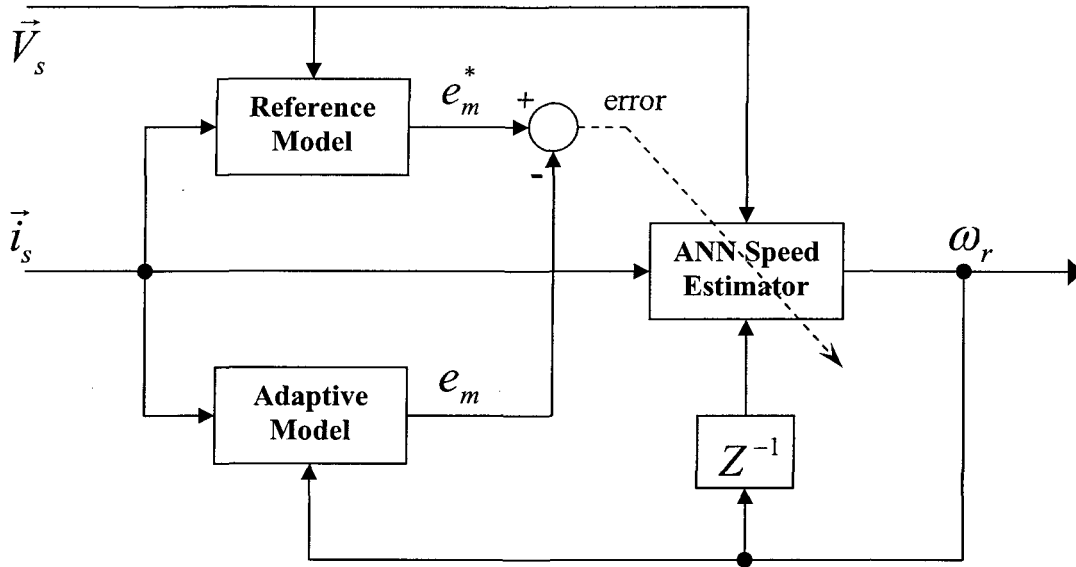


Fig. 1 Structure of speed estimator using NN

The structure of the ANN Speed Estimator is the well known structure of Jordan recurrent network [11] which is shown in Fig. 2. The inputs are the voltage and current of the motor in the stationary  $\alpha\beta$  frame (4 inputs:  $V_{s\alpha}$ ,  $V_{s\beta}$ ,  $i_{s\alpha}$ ,  $i_{s\beta}$ ) plus the output (estimated speed) of the previous step which is used as the extra input.

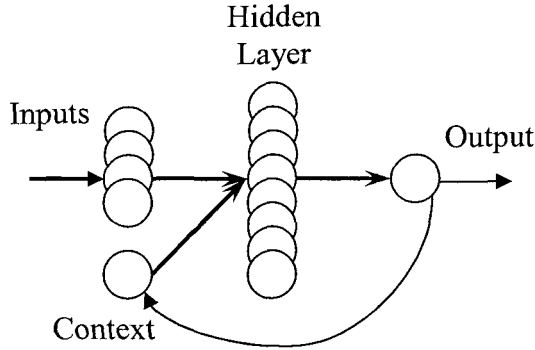


Fig. 2 Structure of NN

The adjustment of the weights in hidden layer is achieved by using the error of emf. Note that though emf is a vector, but as using the amplitude of it conduces to acceptable result, it's used in order to simplify the calculation. Backpropagation method is used in this way, so the weight correction for the  $i \rightarrow j$  connection is calculated from this equation [12]:

$$\Delta w_{ji}(k) = \alpha \Delta w_{ji}(k-1) + \eta_{ji} \sum_{b=1}^B \delta_j^{(b)}(k) y_i^{(b)}(k) \quad (7)$$

Or by naming the output for neuron  $i$  as  $o_i$  it could be written as bellow [13]:

$$\Delta w_{ji}(k) = \alpha \Delta w_{ji}(k-1) + \eta \delta_j o_i \quad (8)$$

To start the training the weights are initially randomize from -0.5 to 0.5 as  $\alpha$  is set equal to 0.02 and  $\eta$  equal to 0.01 (these two recent parameters are obtained by try and error in several tests to have the best result. Theoretically they can take some amount between 0 and 1 with no exact formula [13]). After that in each step the estimated speed is used in adaptive model and the error between two models would replace

to repeat the calculation. So, the training is completely online (real-time) with no necessary pre-calculation.

### III. Simulation Results

All the system is simulated by using the Matlab/Simulink software package. A 3-phase induction motor with Indirect Vector Speed Controller is used in the model and the result is compared with the actual speed, simulated in Matlab/Simulink. The parameters of the induction motor are:

Related Power	Pr	500 W
Line-Line Voltage	Vr	220 V
Related Torque	T	3.41 N m
Number of Poles	P	4
Stator Resistance	Rs	4.495 $\Omega$
Rotor Resistance	Rr	5.365 $\Omega$
Stator Inductance	Ls	165 mH
Rotor Inductance	Lr	162 mH
Magnetizing Inductance	Lm	149 mH
Rotor Moment of Inertia	J	.00095 Kg m <sup>2</sup>

The simulation is run in different conditions to read the behavior of the model. Fig. 3 shows the speed performance and the error between the estimated speed and real speed in no load conditions.

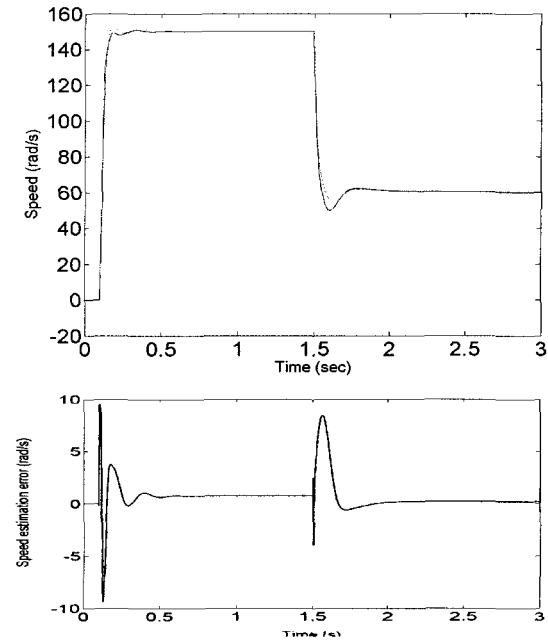


Fig. 3 No Load estimated speed

Reference speed of motor has step changes. At  $T=0.1s$  the reference is set from 0 to 150 rad/s (near the maximum speed of motor) and at  $T=1.5s$  is decreased to 60rad/s. There is an error of less than 0.5% in the estimated speed, but trying to decrease it (by changing the training parameters) causes some instability in the output.

In Fig. 4 the estimated speed while loading is studied. The motor is started soft, then a load of 1.2 is put on, during working on the speed of 150rad/s at  $T=1.2s$  and then the reference speed is suddenly decreased to 25rad/s after returning back to the stable speed. The error seems acceptable again.

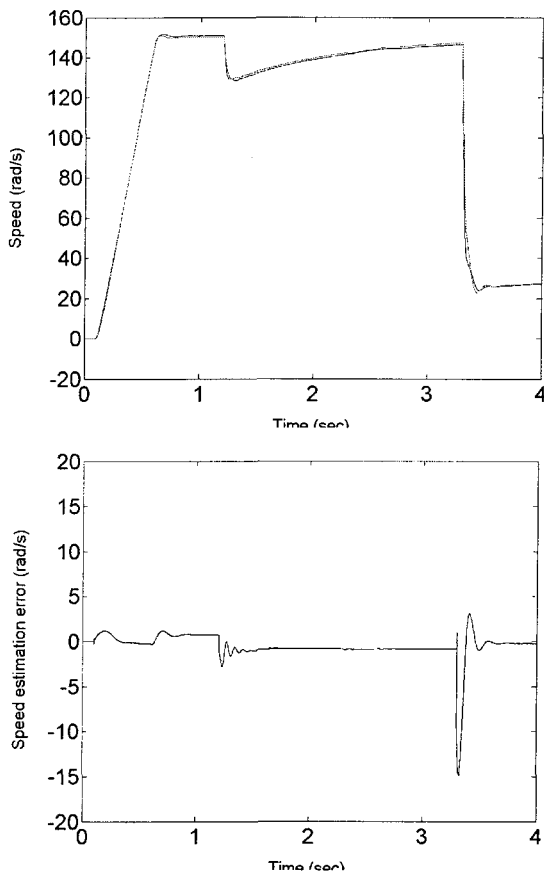


Fig. 4 Estimated speed while loading motor

Fig. 5 shows the behavior of the estimator for changes of speed while running under load. Increasing the speed when the motor is under load does not affect the quality and the output error never reaches 0.5% again. Decreasing the load and again changing the reference speed are shown in this figure in the following steps.

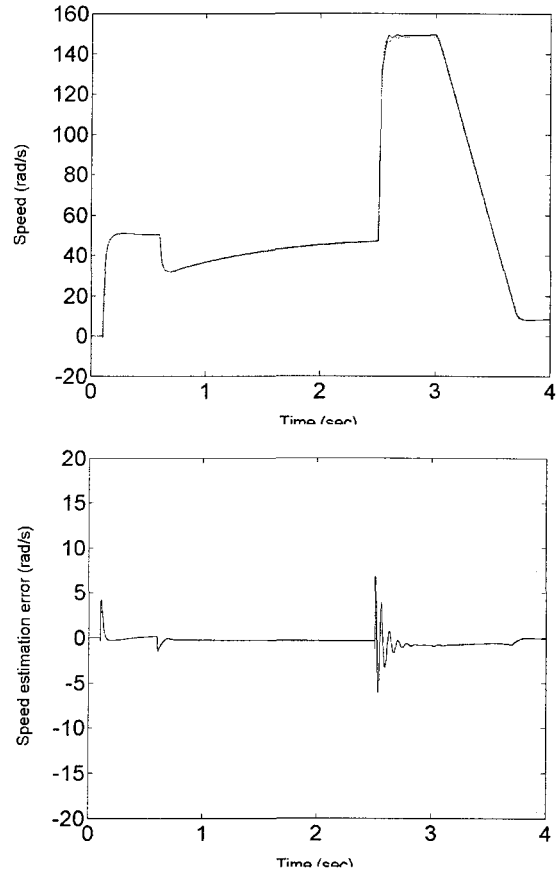


Fig. 5 Estimated speed changes under Load

#### IV. Conclusion

In this paper, a neural network based sensorless speed estimation method for induction motor is proposed. The proposed method uses online training of NN in a vector control algorithm and has good results in different ranges of speed from low to maximum. Also, it has robust speed estimation in no load or under load conditions and different ways of speed changes. So, it's believed to be an important contribution to sensorless operating of induction motor drives.

#### V. References

- [1] F. Blaschke, The Principal of Field-Orientation as Applied to the New Transvector closed-Loop Control System for Rotating-Field Machines, *Siemens Review XXXIX*, (5), 1972, 217-219.
- [2] H. Tajima and Y. Hori, Speed sensorless field-oriented control of the induction machine,

*IEEE Trans. Ind. Applicat.*, 29(1), Jan./Feb. 1993, 175-180.

[3] H. Kubota, K. Matsuse and T. Nakano, New adaptive flux observer of induction motor for wide speed range motor drives, *Proc. IEEE IECON Conference*, 1990, 921-926.

[4] Seong-Hwan Kim, Tae-Sik Park, Ji-Yoon Yoo and Gwi-Tae Park, Speed-Sensorless Vector Control of an Induction Motor Using Neural Network Speed Estimation, *IEEE Trans. Ind. Applicat.*, 48(3), June 2001.

[5] Seung-Ill Moon, Ali Keyhani and Srinivas Pillutla, Nonlinear Neural-Network Modeling of an Induction Machine, *IEEE Trans. Ctrl. Sys. Tech.*, 7(2), Mar. 1999.

[6] D. Fodor, J. P. Six and D. Diana, Neural networks applied for induction motor speed sensorless estimation, *Proc. ISIE '95*, 1995, 181-186.

[7] K. L. Shi, T. F. Chan, Y. K. Wong and S. L. Ho, Direct Self Control of Induction Motor Based on Neural Network, *IEEE Trans. Ind. Appl.*, 37(5), Sep./Oct. 2001.

[8] Bartlomiej Beliczynski and Lech Grzesiak, Induction motor speed estimation: neural versus phenomenological model approach, *Elsevier Science B. V., Neurocomputing* 43, 2002, 17-36.

[9] Fang-Zheng Peng and Tadashi Fuako, Robust Speed Identification for Speed-Sensorless Vector Control of Induction Motors, *IEEE Trans. Ind. Applicat.*, 30(5), Sep./Oct. 1994.

[10] Colin Schauder, Adaptive speed Identification for Vector Control of Induction Motors without Rotational Transducers, *IEEE Trans. Ind. Appl.*, 28(5), 1992, 1054-1061.

[11] Jose C. Principe, Neil R. Euliano and W. Curt Lefebvre, Neural and Adaptive Systems, *John Wiley & Sons Inc.*, 2000, 529-544.

[12] Simon Haykin, Neural Networks, AComprehension Foundation, *Macmillan College Pub.*, 1994, 197.

[13] K.S. Narendra and K. Parthasarathy, Identification and control of dynamical systems using neural networks, *IEEE Trans. Neural Networks*, 1(1), Mar. 1990, 4-27.

[14] N. Kandil, K. Khorasani, R. V. Petal and V. K. Sood, Optimum Learning Rate for Back-propagation Neural Networks, *IEEE Selected Conference Papers, Neural Networks Theory, Technology and Applications*, 1993, 249-252.

[15] Ned Mohan, Advanced Electric Drives, *MNPERE Pub.*, 2001, 4-1

[16] J. Campbell and M. Summer, Practical sensorless induction motor drive employing an

artificial neural network for online parameter adaption, *IEE Proc.-Electr. Power Appl.* 149(4), July 2002.

## **APPENDIX B**

### **GLOSSARY OF SYMBOLS**

## A.1 Symbols

$e, E$	voltage	V
$e, exp$	2.71828	
$f$	stator voltage frequency	Hz
$g$	per unit slip	
$i, I$	current	A
$j$	$\sqrt{-1}$	
$K$	non-dimensional factor	
$L$	self inductance	H
$M$	mutual inductance	H
$N$	turns number of the coil	
$n, N_r$	mechanical speed of rotor	min <sup>-1</sup>
$N_s$	mechanical speed of rotor	Min <sup>-1</sup>
$P$	power	W
$p$	poles number	
$p$	d/dt	
$R$	resistance	$\Omega$
$S$	speed slip	Rad/s
$T$	torque	Nm
$t$	time	sec
$V$	voltage	V
$X$	reactance	$\Omega^{-1}$
$\pi$	3.14159	
$\sigma$	leakage factor	
$\varphi$	flux vector angle	o
$\psi, \Psi$	flux linkage (flux)	Vs
$\omega$	angular velocity of rotor	Rad/sec

## A.2 Subscripts

<b><i>a, b, c</i></b>	three-phase parameters
<b><i>d</i></b>	direct component
<b><i>e</i></b>	electric parameter
<b><i>k</i></b>	discrete time step
<b><i>L</i></b>	load parameter
<b><i>M</i></b>	motor parameter
<b><i>m</i></b>	magnetizing parameter
<b><i>p</i></b>	phasor parameter
<b><i>q</i></b>	quadrature component
<b><i>r</i></b>	rotor parameter
<b><i>s</i></b>	stator parameter

## **APPENDIX C**

## **REFERENCES**



- [1] VAS, P., "Sensorless Vector and Direct Torque Control," *Oxford University Press, Inc., New York*, 1998.
- [2] LIU, J.-J. , KUNG, I.-C. & CHAO, H.C., "Speed Estimation of Induction Motor Using a Non-linear Identification Technique," *Proc. Natl. Sci. ROC(A)*, vol. 25, no. 2, 2001, pp. 107-114.
- [3] VAS, P., "Vector Control of AC Machine," *Clarendon Press, Oxford*, 1990.
- [4] BLASCHKE, F., "The Principal of Field-Orientation as Applied to the New Transvector Closed-Loop Control System for Rotating-Field Machines," *Siemens Review XXXIX*, vol. 5, 1972, pp. 217-219.
- [5] TAKAHASHI, I. & NOGUCHI, T., "A New Quick Response and High Efficiency Control Strategy of an Induction Motor," *IEEE IAS Annual Meeting*, 1985, pp. 496-502.
- [6] DEPENBROCK, M., "Direkte Selbstregelung (DSR) für hochdynamische Drefeldantriebe mit Stromrichterschaltung," *ETZ A* 7, pp. 211-218.
- [7] TIITINEN, P. , POHJALAINEN, P. & LALU, J., "The Next Generation Control Method: Direct Torque Control (DTC)," *EPE Journal*, vol. 5, no. 1, Mar. 1995, pp. 14-18.
- [8] PETERSON, B., "Induction Machine Speed Estimation," *Industrial Electrical Engineering and Automation, Lund Institute of Technology, Sweden*, 1996.
- [9] HOLTZ, J., "State of the art of controlled AC drives without speed sensors," *International Journal of Electronics*, vol. 80, no. 2, pp. 249-263, Feb. 1996.
- [10] KIM, Y.-R. , SUL, S.-K. & PARK, M.-H., "Speed Sensorless Vector Control of Induction Motor Using Extended Kalman Filter," *IEEE Trans. on Ind. Appl.*, vol. 30, no. 5, Sep./Oct. 1994, pp. 1225-1233.
- [11] SHI, K.L. , CHAN, T.F. , WONG, Y.K. & HO, S.L., "Speed estimation of an Induction Motor Drive Using an Optimized Extended Kalman Filter," *IEEE Trans. on Ind. Electronics*, vol. 49, no. 1, pp.124-133, Feb. 2002.
- [12] KULKARNI, A.S. & EL-SHARKAWI, M.A., "Speed Estimator for Induction Motor Drives Using an Artificial Neural Network," *IEEE International Electric Machines and Drives Conf., Milwaukee, WA*, May 1997, pp. MD2/2.1 - MD2/2.3.
- [13] THEOCHARIS, J. & PETRIDIS, V., "Neural Network Observer for Induction Motor Control," *IEEE Control Systems Magazine*, vol. 14, issu. 2, Apr. 1994, pp.26-37.
- [14] BEN-BRAHIM, L., "Motor Speed Identification Via Neural Networks," *IEEE Industry Application Magazine*, vol. 1, issu. 1, Jan./Feb. 1995, pp. 28-32.
- [15] KOSOKO, B., "Neural Networks and Fuzzy Logic," *Prentice-Hall, Inc., Englewood Cliffs*, 1992.
- [16] THALER, G.J. & MILTON, L.W., "Electric Machines: Dynamic and Steady State," *John Wiley & Sons Inc., p165, New York*, 1966.

- [17] KRAUS, P.C. , WASYNCZUK, O. & SUDHOFF, S.D., "Analysis of Electric Machinery," *IEEE Press, Piscataway*, 1994.
- [18] BOSE, B.K., "Modern Power Electronics and AC Drives," *Prentice-Hall, Inc., Upper Saddle River*, 2002.
- [19] SEN, P.C., "Principles of Electric Machines and Power Electronics," *John Weily & Sons, Inc., Toronto*, 1996.
- [20] NOVOTNY, D.W. & LIPO, T.A., "Vector Control and Dynamics of AC Drives," *Oxford University Press, Inc., New York*, 1996.
- [21] FITZGERALD, A.E. & KINGSLEY, C., "Electrical Machinery," *McGraw-Hill Books, New York*, 2<sup>nd</sup> ed., 1961.
- [22] BOSE, B.K., "Power Electronics and AC Drives," *Prentice-Hall, Inc.*, 1986.
- [23] DORF, R.C. & BISHOP, R.H. "Modern Control Systems", *Addison-Wesley Publishing Company*, 1995.
- [24] SAY, M.G., "Alternating Current Machines," *Pittman Publishing*, 1976.
- [25] LAITHWAITE, E.R., "Induction Machines for Special Purposes," *George Newnes*, 1966.
- [26] BOLDEA, I. & Nasar, S.A., "Electric Drives," *CRC Press LLC, New york*, 1999.
- [27] GREEN, T.C., "Scalar Controlled Induction Motor Drives," *Ph.D Thesis, Heriot-Watt University*, Aug. 1990.
- [28] BLASCHKE, F., "The Principle of Field Orientation – the Basic for the Transvector Control of Three-Phase Machines," *Siemens Zeitschrift*, vol. 45, no. 10, 1971, pp. 757-760.
- [29] HASSE, K., "On the Dynamics of Speed Control of a Static AC Drive with a Squirrel-Cage Induction Machine," *PhD Dissertation, Tech. Hochsch, Darmstadt*, 1969.
- [30] XU, X. , DE DONCKER, R. & NOVOTNY, D.W., "A Stator Flux Oriented Induction Machine Drive," *IEEE PESC, Kyoto*, Apr. 1998, pp. 870-876.
- [31] DE DONCKER, R. & NOVOTNY, D.W., "The Universal Field Oriented Controller," *IEEE-IAS Annual Meeting*, Oct. 1998, pp. 450-456.
- [32] LEONHARD, W., "Field-Orientation for Controlling ac-machine, Principle and Application," *IEE Conf. Pub.*, no. 291, pp. 277-282.
- [33] TELFORD, D. , DUNNIGAN, M.W. & WILLIAMS, B.W., "Online Identification of Induction Machine Electrical Parameters for Vector Control Loop Tuning," *IEEE Trans. Ind. Electronics.*, vol. 50, no.2, Apr. 2003, pp. 253-261.

- [34] DE DONCKER, R. & NOVOTNY, D.W., "The Universal Field Oriented Controller," *IEEE Trans. Ind. Appl.*, vol. 30, no.1, Jan. 1994, pp. 92-100.
- [35] ABBONDANTI, A. & BRENNEN, M.B., "Variable Speed Induction Motor Drives Use Electronic Slip Calculator Based on Motor Voltages and Currents," *IEEE Trans. On Ind. Appl.*, vol. 1A-11, no. 5, Sept./Oct. 1975, pp. 483-488.
- [36] SCHAUDER, C., "Adaptive Speed Identification for Vector Control of Induction Motor without Rotational Transducers," *IEEE Trans. Ind. Appl.*, vol. 28, no. 5, Sept./Oct. 1992, pp. 1054-1061.
- [37] TA, C-M. , UCHIDA, T. & HORI, Y., "MRAS-based Speed Sensorless Control for Induction Motor Drives Using Instantaneous Reactive Power," *27<sup>th</sup> Annual Conference of the IEEE Industrial Electronics Society, IECON '01*, vol. 2, 2001, pp. 1417-1422.
- [38] PENG, F.Z. & FUKAO, T., "Robust Speed Identification for Speed Sensorless Vector Control of Induction Motors," *IEEE Trans. Ind. Appl.*, vol. 30, no. 5, 1994, pp. 1234-1240.
- [39] OHTANI, T. , TAKADA, N. & TANAKA, K., "Vector Control of Induction Motor without Shaft Encoder," *IEEE Trans. On Ind. Appl.*, vol. 28, no. 1, Jan./Feb.1992, pp. 157-164.
- [40] HOLTZ, J., "Methods for Speed Sensorless Control of AC Drives," *Sensorless Control of AC Motors*, IEEE Press Books, 1996.
- [41] KOBUTA, H., MATSUSE, K. & Nakano, T., "DSP-Based Adaptive Flux Observer of Induction Motor," *IEEE Trans. on Ind. Appl.*, vol. 29, no. 2, Mar./Apr. 1993, pp. 344-348.
- [42] HARNEFORS, L. & NEE, H.P., "Adaptive Sensorless Control of Induction Motors for Improved Low-Speed Performance," *IEEE IAS Annual Meeting*, vol. 1, 1996, pp. 278-285.
- [43] GRIVA, G. , ILAS, C. , EASTHAM, J.F. , PROFUMO, F. & VRANKA, P., "High Performance Sensorless Control of Induction Motor Drives for Industry Applications," *IEEE PCC, Nagaoka*, vol. 2, 1997, pp. 535-539.
- [44] LOVATI, V. & MARCHESONI, M. "A Microcontroller-Based Sensorless Stator Flux-Oriented Asynchronous Motor Drive for Traction Applications," *IEEE Trans. Power Electronics*, vol. 46, no. 5, Jul.1996, pp. 777-784.
- [45] BITTANTI, S. & AL., "The Riccati Equation," *Springer, New York*, 1991, pp. 263-291.
- [46] KALMAN, R.E. & BUCY, R.S. "New Results in Linear Filtering and Prediction Theory," *Journal of Basic Engineering*, 1961, pp. 95-108.
- [47] VERGHESE, G.C. & SANDERS, S.R., "Observers for Flux Estimation in Induction Machines," *IEEE Trans. Ind. Appl.*, vol. 35, no. 1, Feb. 1988, pp. 85-94.
- [48] ATKINSON, D.J. , ARCANELEY, P.P. & FINCH, E.J., "Observers for Induction Motor State and Parameter estimation," *IEEE Trans. On Ind. Appl.*, vol. 27, no. 6, Nov./Dec. 1991, pp. 1119-1127.

- [49] OUHROUCHE, M.A., "Estimation of Speed, Rotor Flux and Rotor Resistance in Cage Induction Motor Sensorless Drive Using the EKF Algorithm," *International Journal of Power and energy systems*, vol. 22, no. 2, 2002, pp. 157-164.
- [50] LUENBERGER, D.G., "An Introduction to Observers," *IEEE Trans. Automatic Control*, vol. AC-16, no. 6, Dec.1971, pp. 596-602.
- [51] DU, T. , VAS, P. & STRONACH, F., "Design and application of Extended Observers for Joint State and Parameter Estimation in High-Performance AC Drives," *IEE Proc. Electr. Power Appl.*, vol. 142, no. 2, Mar. 1995, pp. 71-78.
- [52] EL-MOUCCARY, C. , GARCIA-SOTO, G. & MENDES, E., "Robust rotor Flux, Rotor Resistance and speed estimation of an Induction Machine Using the Extended Kalman Filter," *IEEE ISIE, Bled, Slovenia*, 1999, pp. 742-746.
- [53] PARBERRY, I., "Circuit Complexity and Neural Networks," *The MIT Press, Massachusetts*, 1994, p.1.
- [54] HAYKIN, S., "Neural Networks: a comprehensive foundation," *Macmillan College Publishing Company, New York*, 1994.
- [55] VAS, P., "Artificial-Intelligence-Based Electrical Machines and drives," *Oxford University Press, Inc., New York*, 1999.
- [56] WISHART, M.T. & HARLEY, R.G., "Identification and Control of Induction Machines Using Artificial Neural Networks," *IEEE Trans. Ind. Appl.*, vol. 31, May/June 1995, pp. 612-619.
- [57] SIMOES, M.G. & BOSE, B.K., "Neural Network Based Estimation of Feedback Signals for a Vector Controlled Induction Motor Drive," *IEEE Trans. Ind. Appl.*, vol. 31, May./Jun. 1995, pp. 620-629.
- [58] CUIBUS, M. , BOSTAN, V. , AMBROSII, S. , ILAS, C. & MAGUREANU, R., "Luenberger, Kalman and Neural Network Observers for Sensorless Induction Motor Control," *3<sup>rd</sup> International Power Electronics and Motion Control Conference, Beijing*, vol. 3, Aug. 2000, pp. 1256-1261.
- [59] KIM, Y-H. & KOOK, Y-S. "Neural Network based Speed Sensorless Induction Motor Drives with Kalman Filter Approach," *Proc. 24th Annual Conference of the IEEE, Aachen*, vol.2, Aug./Sep 1998, pp.997-1001.
- [60] KIM, S-H. , PARK, T-S. & YOO, J-Y., "Speed Sensorless Vector Control of an Induction Motor Using Neural Network Speed Estimation," *IEEE Trans. on Ind. Appl.*, vol. 48, no. 3, Jun. 2001, pp.609-614.
- [61] KOWALASKA, T.O. & KOWALASKI, C.T., "Neural Network application for Flux and Speed Estimation in the sensorless Induction Motor Drive," *IEEE ISIE, Guimaraes, Portugal*, 1997, pp. 1253-1258.

- [62] TA, C-M. , UCHIDA, T. & HORI, Y., "MRas-based speed Sensorless Control for Induction Motor Drives Using Instantaneous Reactive Power," *IECON'01*, 2001, pp. 1417-1422.
- [63] CHEN, T-C. & SHEU, T-T., "Model Reference Neural Network Controller for Induction Motor Speed Control," *IEEE Trans. Energy. Conversion.*, vol. 17, no. 2, June. 2002, pp. 157-163.
- [64] PINTO, J.O.P. , BOSE, B.K. & SILVA, L.E.B., "A Stator Flux Oriented Vector-Controlled Induction Motor Drive with Space VECTOR PWM AND FLUX VECTOR SYNTHESIS NEURAL NETWORKS," *Industry Applications Conference*, vol. 3, 2000, pp.1605-1612.
- [65] HEBB, D.O., "The Organization of Behavior: A Neuropsychological Theory," *Wiley, New York*, 1949.
- [66] HOPFIELD, J., "Neural Networks and Physical Systems with Emergent Collective Computational Properties," *Proc. National Academy of Science of the USA*, vol.79, 1982, pp. 2554-2588.
- [67] HOPFIELD, J.J. & TANK, D.W., "Neural Computation of Decisions in Optimization Problems," *Biological Cybernetics*, vol. 52, 1985, pp. 141-152.
- [68] CAUDILL, M. & BUTLER, C., "Understanding Neural Networks," *The MIT Press, Massachusetts*, 1994.
- [69] LISBOA, P.G.J., "Neural Networks: Current Applications," *Chapman & Hall, London*, 1992.
- [70] SCHALKOFF, R.J., "Artificial Neural Networks," *The McGraw-Hill Companies Inc., New York*, 1997.
- [71] JORDAN, M., "Attractor Dynamics and Parallelism in a Connectionist Sequential Machine," *Proc. 8th Annual Conf. of Cognitive Science Society, Amherst, MA*, 1986.
- [72] ELMAN, J.L., "Finding Structure in Time," *Cognitive Science*, vol. 14, pp. 179-211, 1990.
- [73] PRINCIPLE, J.C. , EULIANO, N.R. & LEFEBVRE, W.C., "Neural and Adaptive Systems," *John Wiley & Sons Inc.*, 2000, pp. 529-544.
- [74] MENDEL, J.M. & MCLAREN, R.W., "Reinforcement-learning control and pattern recognition systems," In *Adaptive, Learning, and Pattern Recognition Systems; Theory and Applications*, Academic Press, New york, 1970, pp. 287-318.
- [75] BEALE, R. & JACKSON, T., "Neural Computing: An Introduction," *Adam Hilger Publishing, Philadelphia*, 1990.
- [76] RUMELHART, D. , HINTON, G. & WILLIAMS, R., "Learning Internal Representation by Error Propagation," *Parallel distributed Processing I, MIT Press, Cambridge*, 1986, pp. 318-362.

- [77] TUINENGA, P.W., "Spice: A Guide to Circuit Simulation and Analysis Using PSpice," 3<sup>rd</sup> ed., *Prentice-Hall, Inc.*, 1995.
- [78] DUMITRESCU, A. , FODOR, D. , JOKINEN, T. , ROSU, M. & BUCURENCIO, S., "Modeling and Simulation of Electric Drive Systems Using Matlab/Simulink Environment," *International Conference on Electric Machines and Drives*, 1999, pp. 451-453.
- [79] WADE, S. , DUNNIGAN, M.W. & WILLIAMS, B.W., "Modeling and Simulation of Induction Machine Vector Control with Rotor Resistance Identification," *IEEE Trans. Power Electronics*, vol. 12, no. 3, May 1997, pp. 495-506.
- [80] DE AGUIAR, M.L. & CAD, M.M., "The Concept of Complex Transfer Functions Applied to the Modeling of Induction Motor," *Power Engineering Society Winter Meeting*, 2000, pp. 387-391.
- [81] TANG, L. & RAHMAN, M.F., "A New Direct Torque Control Strategy for Flux and Torque Ripple Reduction for Induction Motors Drive – A Matlab/Simulink Model," *IEEE International Electric Machines and Drives Conference*, 2001, pp. 884-890.
- [82] FODOR, D. , SIX, J.P. & DIANA, D., "Neural Networks Applied for Induction Motors Speed Sensorless Estimation," *Proc. ISIE'95*, 1995, pp. 181-186.
- [83] BEN-BRAHIM, L. & KOUDOR, T., "Implementation of an Induction Motor Speed Estimator Using Neural Networks," *Proc. IPEC 1995*, 1995, pp. 52-57.
- [84] MOHAN, N. , UNDELAND, T.M. & ROBBINS, W.P. "Power Electronics," *John Wiley & Sons Inc., Toronto*, 1995.
- [85] ELLIS, M.G., "Electronic Filter Analysis and Synthesis," *Artech House Inc., Norwood*, 1994.
- [86] NARENDRA, K.S. & PARTHASARATHY, K., "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Trans. Neural Networks*, vol. 1, no. 1., Mar. 1990, pp. 4-27.
- [87] OTTERBACH, R. & LEINFELLNER, R., "Real-Time Simulation," *Translation from "Virtuelles Ausprobieren," Elektronik 8*, 1999, pp. 1-4.
- [88] BACKER, T.P., "Stack-Based Scheduling of Real-time Process," *The Journal of Real-Time Systems*, no. 3, 1991, pp. 67-99.
- [89] LAPLANTE, P., "Real-Time Systems: Design and analysis rientation for Controlling ac-machine, Principle and Application," *IEEE Press, Piscataway*, 1993.
- [90] STALLINGS, W., "Operating Systems," *Englewood Cliffs, Prentice Hall*, 1995.
- [91] Proakis, J.G. & MANOLAKIS, D.G., "Digital Signal Processing: Principles, Algorithms, and Applications," 2<sup>nd</sup> ed., *MacMillan, New York*, 1992.

- [92] ZHEN, L. , KYTE, M. & JOHNSON, B., "Hardware-in-the-Loop Real-Time Simulation Interface Software Design," *Proc. 7<sup>th</sup> International IEEE Conference*, 2004, pp. 1012-1017.
- [93] WELLS, R.B. , FISHER, J. , YING, Z. , JOHNSON, B.K. & KYTE, M., "Hardware and Software Considerations for Implementing Hardware-in-the-Loop Traffic Simulation," *Proc. 27<sup>th</sup> Annual Conference of the IEEE, IECON '01*, vol. 3, 2001, pp. 1915-1919.
- [94] OUHROUCHE, M.A. , LECHEVIN, N. & ABOURIDA, S., "RT-Lab Based Real-Time Simulation of a Direct Field-Oriented Controller for an Induction Motor," *Proc. 7<sup>th</sup> International Conference on Modeling and Simulation of electric Machines, Converters and Systems, Montreal*, 2002, pp. 1-6.
- [95] GREGA, W., "Hardware-in-the-Loop Simulation and Its Application in Control Education," *Proc. 29<sup>th</sup> ASEE/IEEE Frontiers in Education Conference*, Nov. 1999, pp. 12b6\_7-12b6\_11.