
Electrical Machines Parameter Identification Using Genetic Algorithms

Konstantinos T. Kampsios,

BEng (Hons), MSc

GEORGE GREEN LIBRARY OF
SCIENCE AND ENGINEERING T



**The University of
Nottingham**

Department of Electrical & Electronic Engineering

Submitted to the University of Nottingham for the degree of Doctor of
Philosophy, September 2009

Abstract

In Indirect Field Orientation (IFO) of induction motors, the interest for parameters identification has increased rapidly due to the great demand for high performance drives and more sophisticated control systems that have been made possible by the development of very powerful processors, such as floating point DSPs. Accurate knowledge of the machine electrical parameters is also required to ensure correct alignment of the stator current vector relative to the rotor flux vector, to decouple the flux – and torque – producing currents and to tune the current control loops. The accuracy and general robustness of the machine is dependant on this model.

Artificial intelligent technologies have been tested in the field of electro mechanics like neural networks, fuzzy logic, simulated annealing and genetic algorithms. These methods are increasingly being utilised in solving electric machine problems.

This thesis addresses a novel non – intrusive approach for identifying induction motor equivalent circuit parameters based on experimental transient measurements from a vector controlled Induction Motor (I.M.) drive and using an off line Genetic Algorithm (GA) routine with a linear machine model. The evaluation of the electrical motor parameters at rated flux operation is achieved by minimising the error between experimental responses (speed or current) measured on a motor drive and the respective ones obtained by a simulation model based on the same control structure as the experimental rig. An accurate and fast estimation of the electrical motor parameters is so achieved. Results are verified through a comparison of speed, torque and line current responses between the experimental IM drive and a Matlab – Simulink model.

The second part of the research work introduces a new approach based on heuristic optimisation for identifying induction motor electrical parameters under different operating conditions such as different load and flux levels. Results show via interpolation test the effect of the most important electrical parameters, the

magnetising inductance L_m and rotor resistance R_r , at each different operating condition.

Konstantinos T. Kampisios

Κωνσταντίνος Θ. Καμπίσιος

Acknowledgements

I would like to thank my supervisors, Dr Pericle Zanchetta and Dr Chris Gerada for the consistently invaluable help and support throughout the period of my research work. I would also like to express my gratitude to Dr Mark Sumner for his advice and valuable information he provided me when I had any difficulties and enquiries, and also for acting as my internal examiner. Many thanks to Dr Stefano Bifaretti for serving as my external examiner.

I would like to thank all my colleagues of the Power Electronics, Machines and Control (PEMC) group at the University of Nottingham. In particular, I would like to thank Dr Omar Jasim for his useful advice and healthy discussions during my research project. My special thanks to Dr Kostas Papastergiou for making the endless hours in the office enjoyable. Without these coffee breaks together with Omar and Kostas, my PhD would be very boring. Thanks a lot guys.

A big “Thank you” to all my friends and relatives that they were close to me and helped me in many different ways to finish my thesis.

I would also like to thank my partner Lina for the priceless moments she offered me so far. She was always next to me during the good and bad times I had especially during the writing up of my thesis.

Lastly, I would like to thank my beloved parents Theodoro and Evangelia and my brothers, Giorgio and Mario for their continued love, care and support. Without these people, I would never have finished my PhD.

To my parents Theodoro and Evangelia.

Στους γονείς μου Θεόδωρο και Ευαγγελία.

Table of Contents

Abstract	ii
.	iv
Acknowledgements	v
.	vi
Table of contents	vii
List of figures	xii
List of tables	xvi
1. Introduction	1
1.1 Background	1
1.2 Literature Review	4
1.3 Project Objectives	14
1.4 Thesis Overview	15
2. Vector Controlled IM Drives	17
2.1 Introduction	17
2.2 DC Drive analogy to an Induction Motor Drive	18
2.3 The flux Angle λ	22
2.4 Field – Oriented Control	23
2.4.1 Indirect Vector Control	24
2.5 Controller Design and System Bandwidth	29
2.5.1 dq current control loop	29
2.5.2 Speed control loop	32
2.5.3 Voltage compensation	35
2.6 Parameter Sensitivity and Adaptation	36
2.6.1 Detuned operation	38
2.6.2 Variation Effect for Steady – state	40
2.6.2.1 Constant Flux and Torque Commands	40
2.6.2.2 Saturation Effects	42

2.6.3	Adaptation Schemes	42
2.6.3.1	Spectral analysis techniques	42
2.6.3.2	Observer based techniques	43
2.6.3.3	Model reference adaptive system based techniques	43
2.6.3.4	Heuristic methods	44
2.7	Chapter Summary	44
3.	IM Parameter Identification	46
3.1	Introduction to Optimization – Heuristic Strategies	46
3.1.1	What is an optimum?	48
3.1.2	Optimisation Algorithms – Overview	49
3.1.3	Premature Convergence	54
3.2	Why Evolutionary Algorithms?	55
3.3	The Genetic Algorithm	58
3.3.1	Genetic Algorithm Procedure	59
3.3.2	Fundamental Elements of the Genetic Algorithm	60
3.3.2.1	Population Representation and Initialisation	62
3.3.2.2	The Objective and Fitness Function	64
3.3.2.3	Selection	66
3.3.2.3.1	Roulette Wheel Selection Methods	67
3.3.2.3.2	Tournament Selection	69
3.3.2.3.3	Stochastic Universal Sampling	71
3.3.2.4	Genetic Operators	71
3.3.2.4.1	Crossover (Recombination)	72
3.3.2.4.2	Mutation	74
3.3.2.5	Reinsertion	75
3.3.2.6	Termination of the GA	77
3.3.3	Applying GAs in Systems Identification	78
3.4	Improving Hybrid Genetic Algorithms for System Identification – HGASI	79
3.4.1	Search Space Reduction Method (SSRM)	81
3.4.1.1	Number of runs for evaluation of search limits	82
3.4.1.2	Total runs	83

3.4.1.3	Width of Window	84
3.4.2	Modified GAs based on migration and artificial selection (MGAMAS)	85
3.4.2.1	Floating – point representation	86
3.4.2.2	Multiple species	87
3.4.2.3	Fitness scaling	89
3.4.2.4	Mutation Operators	89
3.4.2.5	Crossover Operators	90
3.4.2.6	Fitness Evaluation	91
3.4.2.7	Tagging	92
3.4.3	Cataclysmic Mutation	93
3.4.4	Sigma Truncation Scaling Method	94
3.5	Chapter summary	95
4.	Experimental Implementation	99
4.1	Introduction	99
4.2	Experimental System	100
4.3	Motor Drive	101
4.4	The dSPACE DS1104 Controller Board	102
4.5	DSP to Motor Drive Interface System	104
4.5.1	Inverter Interface – Incorporating Dead – time Protection	105
4.5.2	Voltage and Current Measurement	106
4.5.3	Analogue Filtering	106
4.6	Implementing the Drive Structures	106
4.7	Implementing the Control Systems	108
4.8	Accuracy/Resolution of the experimental data	109
4.9	Chapter summary	110
5.	Applying GAs to Vector Controlled Drives At Rated Operation	
Parameter Identification		111
5.1	Introduction	111
5.2	Description of the system	113
5.2.1	Experimental Induction Motor Drive	113

5.2.2	Simulated Induction Motor Drive	115
5.3	GAs Experimental Parameters Identification	120
5.3.1	Application of the SSRM	123
5.4	Results	126
5.4.1	Comparison of Speed Response	127
5.4.2	Comparison of I_{sq} Current Response	128
5.4.3	Discuss and Analysis of the Results	130
5.4.3.1	Speed and I_{sq} current error	131
5.5	Sensitivity of the Parameters Estimation	133
5.5.1	Variation of Rotor Resistance (R_r)	134
5.5.2	Variation of Magnetising Inductance (L_m)	136
5.5.3	Variation of Stator Leakage Inductance (L_{ls})	137
5.5.4	Variation of Rotor Leakage Inductance (L_{lr})	138
5.5.5	Variation of Stator Resistance (R_s)	139
5.5.6	Discussion and Analysis of the Parameters' Sensitivity	141
5.6	Accurate knowledge of moment of Inertia J	142
5.7	Parameter Identification under Poorly Tuned Vector Control System	143
5.8	Average L_m and R_r for Varying Flux Levels	143
5.9	Chapter Summary	145

6. Electrical Parameter Identification for Varying Load and

Flux Levels	147
6.1 Introduction	147
6.2 Description of the Updated System	149
6.2.1 Fitness Function	150
6.2.2 Identified Parameters	153
6.2.3 Simulation Model Characteristics	153
6.2.4 GAs Characteristics	153
6.3 Results – Comparison	154
6.3.1 Variation of Magnetising Inductance (L_m) for different operating conditions	159
6.3.2 Improvement of the vector controlled drive	160

6.4 Chapter Summary	163
7. Conclusions and Future Work	164
7.1 Conclusion	164
7.2 Future Work	167
Appendix A Induction Motor Equivalent Circuit	169
Appendix B Simulink Implementation of Induction Machine Model ..	173
Appendix C Traditional Identification Methods	180
Appendix D Dynamic Equations of the Induction Motor	192
Appendix E Reference frames	194
Appendix F Estimation of mechanical parameters J and B	200
References	202
Publications	218

List of Figures

2.1	(a) Separately excited dc motor, (b) Vector – Controlled I.M.	19
2.2	Vectors inside an induction motor	20
2.3	Resolving I_s into two components in a rotating frame	21
2.4	The rotor flux angle λ	22
2.5	Stator current space vector and its component in (α, β) and in the dq rotating reference frame	23
2.6	Indirect vector control phasor diagram	25
2.7	Indirect Rotor Flux Orientation	26
2.8	Indirect Vector (Speed) Control (without field weakening)	28
2.9	Block diagram of the current control loop	29
2.10	Simplified block diagram of the current control loop	30
2.11	PI dq current controller	31
2.12	Block diagram of the speed control loop	32
2.13	Equivalent circuit	33
2.14	Simplified block diagram of the speed control loop	33
2.15	PI speed controller	34
2.16	Voltage compensation	35
2.17	Vector (phasor) diagram for IFOC with correct value of slip calculator time constant	39
2.18	Vector (phasor) diagram for IFOC with slip	39
3.1	Diagram of a function or process that is to be optimised. Optimisation varies the input to achieve a desired output	47
3.2	Local and global maxima and minima	49
3.3	Classification of Deterministic Optimisation Algorithms	52
3.4	Classification of Probabilistic Optimisation Algorithms	52

3.5	Premature convergence in the objective space	54
3.6	The basic cycle of genetic algorithms	60
3.7	A Simple Genetic Algorithm	61
3.8	Example of individual representation	63
3.9	Examples of initialisation methods	64
3.10	Gradual fitness improvement during the run (minimisation)	66
3.11	Roulette Wheel Selection	68
3.12	Roulette Wheel Selection	69
3.13	Stochastic universal sampling	71
3.14	Producing offspring using crossover operator	73
3.15	Mutation operator	75
3.16	The principle scheme for parameters estimation	78
3.17	Search Space Reduction Method	81
3.18	Diagram of empirical rule within three standard deviations of the mean	84
3.19	MGAMAS architecture	86
3.20	Multipoint crossover operator	91
4.1	Structure of the experimental drive rig	101
4.2	Drive rig	102
4.3	Block diagram of the DS1104 controller board	103
4.4	The breakout panel of the dSPACE DS1104 R&D Controller Board	104
4.5	Interface card of the FKI industrial drive inverter	105
4.6	Structure of an implemented drive system	107
5.1	Block diagram representing the experiment	112
5.2	Speed and I_{sq} current response of the experimental I.M. drive without load	113

5.3	Speed and I_{sq} current response of the experimental I.M. Drive with load	114
5.4	Simplified equivalent circuit of an I.M.	115
5.5	Measured and modelled Speed (a) and I_{sq} current (b) responses: no load and standard parameters	116
5.6	Measured and modelled Speed (a) and I_{sq} current (b) responses: with load (15.5Nm) and standard parameters	117
5.7	Zoom of Speed and current response of in Figure 5.6 at the time in which load (15.5Nm) is applied	118
5.8	Speed (a) and current (b) error between measured and modelled (standard parameters) speed and current responses	119
5.9	Block diagram representing the GAs routine	121
5.10	Measured and GAs (modelled) Speed responses with no load	127
5.11	Measured and GAs (modelled) Speed responses with a load torque of 15.5Nm	128
5.12	Zoom of speed response of in figure 5.11 at the time that the load is applied	128
5.13	Measured and GAs (modelled) I_{sq} current responses with no load ...	129
5.14	Measured and GAs (modelled) I_{sq} current responses with a load torque of 15.5Nm	129
5.15	Zoom of I_{sq} current response of in figure 5.14 at the time that the load is applied	130
5.16	Speed error between measured and GAs speed response (error from figure 5.10)	131
5.17	I_{sq} current error between measured and GAs I_{sq} current response (error from figure 5.13)	131
5.18	Difference between speed errors from figure 5.8(a) and figure 5.16 .	132
5.19	Difference between I_{sq} current errors from figure 5.8(b) and figure 5.17	132
5.20	Difference between speed responses with 10% increase of R_r	134
5.21	Difference between I_{sq} current responses with 10% increase of R_r ...	135
5.22	Difference between speed responses with 10% increase of L_m	136

5.23	Difference between I_{sq} current responses with 10% increase of L_m . . .	136
5.24	Difference between speed responses with 10% increase of L_{ls}	137
5.25	Difference between I_{sq} current responses with 10% increase of L_{ls} . . .	138
5.26	Difference between speed responses with 10% increase of L_{lr}	139
5.27	Difference between I_{sq} current responses with 10% increase of L_{lr} . .	139
5.28	Difference between speed responses with 10% increase of R_s	140
5.29	Difference between I_{sq} current responses with 10% increase of R_s . . .	140
5.30	Values of L_m in function of the I_{sd}^* (different flux levels) found by GA optimisation	144
5.31	Values of R_r in function of the I_{sd}^* (different flux levels) found by GA optimisation	145
6.1	Block diagram representing the optimisation set – up	148
6.2	Vector Control Motor Drive without speed control	151
6.3	Tuned Vector Control Motor Drive	152
6.4	Detuned Vector Control Motor Drive	152
6.5	Experimental, new GAs and GAs (5 th Chapter) I_{sq} responses for $I_{sd}^* = 4.9A$ and $I_{sq}^* = 9A$	155
6.6	Zoom of I_{sq} responses of figure 6.5	156
6.7	Experimental, new GAs and GAs (5 th Chapter) speed responses for $I_{sd}^* = 4.9A$ and $I_{sq}^* = 9A$	156
6.8	Zoom of speed responses of figure 6.5	157
6.9	Experimental, new GAs and GAs (5 th Chapter) V_s voltage responses for $I_{sd}^* = 4.9A$ and $I_{sq}^* = 9A$	157
6.10	Magnetising Inductance vs I_{sd} step responses with variable I_{sq} current responses	159
6.11	Vector Control block diagram with look – up table.	160
6.12	Magnitude of the fundamental current.	161
6.13	Rotor time constant	162
6.14	Current I_{sqr}/I_{sdr} vs time	162

List of Tables

3.1	Selection probability and fitness value	68
4.1	Specification of the Induction Machine	102
5.1	Experimental vector controlled I.M. parameters	114
5.2	Electrical Parameters from different 4kW – 4 pole motor manufacturers.	122
5.3	Genetic Algorithm Estimated Motor Parameters in the Initial Runs. .	125
5.4	Genetic Algorithm Characteristics	126
5.5	Final Genetic Algorithm Estimated Motor Parameters	126
5.6	Comparison between the new GA estimated motor parameters and the traditional ones	133
5.7	Sensitivity of the estimated GA parameters	141
6.1	GA Estimated Motor Parameters for $I_{sd}^* = 4.9A$ and $I_{sq}^* = 9A$	154
6.2	Final GA Estimated Motor Parameters for $I_{sd}^* = 4.9A$ and $I_{sq}^* = 9A$.	155
6.3	Final GA Estimated Motor Parameters for all operating conditions . .	158

Chapter 1

Introduction

1.1 Background

In 1888, Nikola Tesla developed the first induction machine and a couple of years later he succeeded to advance the operation of an electrical machine as it didn't require brushes. The result of this was a radical change in electrical engineering, and so the widespread use of polyphase generation and distribution systems become a fact.

The most common industrial motors that are often referred as the workhorse of the industry are the AC induction motor drives with cage – type machines for variable – speed applications, covering in a wide power range. These applications include industrial and domestic environments. As more than 65% of all the electrical energy generated in the world is used by cage induction motors, the energy – saving aspect of variable – frequency drives is getting a lot of attention. Generally, induction machines have been mostly used at fixed speed for many years, while DC machines have been used for variable speed applications using

the Ward – Leonard controller. However, this kind of controller uses 3 machines, an AC induction motor driven at a fixed speed from the main supply, driving a DC generator which in turn powers a shunt wound DC motor. Therefore, it is clear that this method is expensive and requires careful maintenance.

The rapid developments in the field of power electronics gave a new inspiration and challenge to variable speed applications of both DC and AC machines. So, DC machines use thyristor controlled rectifiers to provide high performance torque, speed and flux control while variable speed induction motor drives use mainly PWM techniques to generate a polyphase supply of a given frequency. The majority of the induction motor drives are performed better under a constant flux in the machine and this can be achieved by keeping a constant voltage/frequency (V/f) ratio. Even if the control of V/f drives is comparably simple, the flux and torque dynamic performance is extremely poor. As a result, many applications that need precise torque and speed control still use DC motors.

The problem of poor dynamic performance can be solved by vector or field – oriented control. The invention of field – oriented control by Blaschke in 1972 was probably the biggest progress in the development of induction motor drives as the principle behind it is that the machine flux and torque are controlled independently like a separately excited DC motor. Instantaneous stator currents are transformed to a rotating reference frame aligned with the rotor, stator or air – gap flux vectors, to produce a d axis component of current (flux producing) and a q axis component of current (torque producing) [Gimenez, 1995].

Two possible approaches of field orientation control for induction motor drives were identified. Direct Field Orientation (DFO) control and Indirect Field Orientation (IFO) control as mentioned by Bose, 2002. The former one is achieved by direct calculation of the motor flux using measured voltages and currents (Hall sensors mounted in the air gap) with some known parameters of the machine. However, the drawback of this approach is its poor performance due to machine parameter dependency and the use of pure integration for the flux calculation [Karanayil, 2005], as well as the fragility of the Hall sensors devalue the inherent robustness of an induction machine.

On the other hand, in IFO induction motor drives, field orientation is achieved by a feedforward slip control derived from the rotor dynamic equations. In addition, the operation of IFO needs precise alignment of the dq reference frame with the rotor flux vector. The achievement of this, plus the accuracy of the slip calculation used in Rotor Flux Oriented Control (RFOC), depends mainly on the accurate knowledge on the rotor time constant T_r of the machine, which change due to temperature and the load of the machine.

Even though the poor dynamic performance can be solved by IFO Control, however this will not lead to the achievement of perfect control of these drives. Therefore, control engineers need to have more information available regarding the motor characteristic. More specific, engineers should be able to identify accurate motor parameters and so to properly find accurate controller parameters of the vector control model. The consequence of having incorrect parameter values used in vector control and thus used in the controllers will create an error in both flux and torque, resulting in a change in dynamics. To overcome these problems, various methods (such as optimisation techniques) in control systems theory have been applied to improve the robustness of a motor.

Initially, conventional optimisation techniques (like linear programming models) had been introduced to solve the above problem. However, their applications were not successful as a result of the problem of convergence to a local minimum instead of global one. Another drawback of these optimisation techniques is that the optimum identified parameter values depends substantially on the initial guess of the parameter. This means that if the initial parameter will change slightly then the algorithm will converge to a totally different solution. All these disadvantages gave the inspiration to researchers to investigate alternative optimisation techniques.

Hence, scientists proposed the use of soft computational methods such as Neural Networks (ANNs) and Fuzzy Logic and also the use of Genetic Algorithms (GAs) in order to be able to identify and control nonlinear dynamic systems including induction machines. The main reason of using the above methods is to make the

resulting system of controller and plant intelligent, which is to make a self – adjusting system to changes in environment and system parameters. In this thesis, the selected optimisation method is based on a GA. In the future, there is the possibility to have a fully automatic control system using an intelligent technique like GAs, which will be possible to identify the required parameters in different operating conditions, decide on the control strategy and self – commissioning the drive.

1.2 Literature Review

The interest for parameter identification in the field orientation control (FOC) drives has increased in the past few years. Moreover, it is well known that the method of vector control in an induction motor drive allows high – performance control of torque and speed and it can be achieved only if both the electrical and mechanical parameters of the machine and load drive are accurately known in all operating conditions [Leonard, 1997]. It is clear that the precise knowledge of all IM parameters is very important for indirect rotor field oriented control (IRFO). However, this is hard to achieve due to the variation of parameters at different machine operating points such as the temperature, flux level, torque level and the nonlinearities caused by skin effect and saturation.

There are many different ways to identify I.M. parameters. Traditional ways were used to identify the resistance and inductance of both rotor and stator by performing the locked – rotor and no – load tests [Fitzgerald, 2003], [Hughes, 2006], [Slemon, 1981]. Nevertheless, one main disadvantage of this method is that the motor has to be locked mechanically and the temperatures of the stator winding and rotor cage have to be measured [Despalatovic, 2005]. These tests also seem to be inaccurate and not appropriate for the synthesis of high dynamic performance systems. The main reason for that comes from the hypothesis that the parameters remain the same regardless the operating conditions.

Many researchers have discussed the estimation of the machine parameters while the motor is at standstill. Willis *et al.* (1989) proposed a frequency domain test to determine the machine parameters. Seok *et al.* (1997) proposed a parameter identification procedure to determine the rotor resistance R_r , the rotor time constant T_r , and the stator resistance R_s , by using several tests implemented with a pulse width modulated (PWM) inverter controlled by the indirect rotor flux oriented control technique. In the beginning, an initial estimate of some parameters is calculated from the nameplate data and then by using a nonlinear technique the identification of the rotor time constant is determined. The advantage of this test is that it can be performed regardless the mechanical load of the machine and it is also achievable to apply an inverter to any induction machine for rotor flux oriented control even though both electrical and mechanical parameters are unknown [Seok, 1997]. It is also possible to integrate these types of tests within standard commissioning tests where these would be automatically performed rather than needing a trained engineer. Ribeiro *et al.* (1997) suggested a linear least – squares approach to estimate R_s , R_r and T_r . An attractive way is to develop the estimation method from the transfer function of the I.M. model at standstill. The continuous transfer function or the discrete function of an I.M. at standstill has been utilized. Peixoto *et al.* (2000) developed a recursive least (RLS) estimating method from the continuous transfer function, while Michalik *et al.* (1998) and Barrero *et al.* (1999) derived a RLS method from the discrete transfer function. The continuous transfer function was also used by Couto (1998) to develop a step – response model fitting method. Fang *et al.* (2005) presented an on – line estimator to determine stator resistor, rotor resistor, stator inductance and mutual inductance which also requires the transfer function of the motor at standstill. Most of these strategies have been shown to work effectively under steady state conditions (constant torque and constant flux). However, as mentioned in the previous paragraph, the major drawback associated with all these techniques becomes from the fact that the motor has to be locked.

Moons (1995) suggested a parameter identification method to identify all electric parameters simultaneously. This is a time – varying linear model and Moons' idea is based on filtering the current, voltage and speed signals of the motor so that the set of obtained signals are related by simple linear equations. In this case, two

estimation methods are applied. The first one is the General Total Least Squares (TLS) method which proved to be sensitive to noise. The second method uses TLS but with extra constraints and its performance was better under high – noise conditions. However, the drawback of this method is the occurrence of pole – zero cancellations and non – persistency of excitation.

Cirrincione (2003) presented a new experimental approach to the parameter estimation of induction motors with least – squares techniques. In particular, it exploits the robustness of total least – squares (TLS) techniques in noisy environments by using a new neuron, the TLS EXIN (a new ANN technique), which is easily implemented online. The TLS EXIN neuron is applied numerically and experimentally for retrieving the parameters of an induction motor by means of a test bench. Moreover, for the case of very noisy data, a refinement of the TLS estimation has been obtained by the application of a nonlinear constrained optimisation algorithm. This method proved to be well performed under high – noise conditions. However, some interesting phenomena may occur like pole – zero cancellations and non – persistency of excitation.

Generally, the parameters of I.M. have an intensely change during the start – up of the motor (transient response) rather than at steady state conditions. This is due to the fact that large currents incur during the start – up of the motor resulting in an increase of the temperature and this may cause large changes in the motor parameters. Hence, the majority of the research work in this field is focusing on this interval since the free – acceleration response data gives more information on machine nonlinearity and so it is more appropriate for parameter identification. The work of this research work is mainly based on this time duration and it is especially applied for the parameters' estimation in different operating conditions.

Shaw (1997) describes three methods for estimating the model parameters of an induction motor using startup transient data. The first method proposed applies simple models with limited temporal domains of validity and obtains parameter estimates by extrapolating the model error bias to zero. However, this method is only applied as a means of finding a good initial guess and does not minimize any specific error criterion for a conventional iterative maximum – likelihood or least –

squares estimator. The second method proposed minimizes equation errors in the induction motor model in the least – square sense using a Levenburg – Marquardt iteration. Finally, the third method which is a continuation of the Levenburg – Marquardt method, minimizes errors in the observations in the least – squared sense and is therefore a maximum – likelihood estimator under appropriate conditions of normality. The drawback of the mentioned techniques is that the determination of the parameter can be bad if the initial guesses are far away from the best values.

Conventional gradient techniques are generally fast and efficient methods for finding the minima of functions. However, these techniques face a major drawback with functions that include noisy experimental data such as the ones found in induction motors. This problem linked to the limited value when several local minima exist. Therefore, researchers thought of finding new different algorithm techniques for solving this problem. Genetic Algorithms (GAs) and Artificial Neural Networks (ANNs) have been gained recognition as an optimisation technique because of their inherent robustness. They can also be used for curve fitting with experimental data as they can find the global minimum even if many local minima exist. In the next paragraphs a wide attention of induction motor parameter estimation is covered by applying these techniques, presented from various perspectives.

Moon (1999) suggested a new approach to identify the nonlinear model of an induction machine. Krause (2002) used the same nonlinear model for the simulation study. Applying the three – phase AC power to a 5 hp induction motor while it is in standstill condition without mechanical load, the measurements of the stator voltages, stator currents and rotor angular velocity can be obtained. As input to the simulation model the measured stator voltages and rotor angular speed are used and the stator currents are simulated as output. Feed – Forward Neural Networks (FNNs) models are utilized to represent the nonlinear parameters as functions of operating conditions. The purpose of the FNN is to estimate the nonlinear functional relationship between input and output patterns. In this technique, the whole free acceleration test data set is divided into subsets (time intervals of 0.05s) and it is supposed that in each subset the parameter values are

constant. As the operating condition changes, the input pattern changes as well and the FNN model determines the parameter values, which correspond to the operating condition. The maximum likelihood (ML) algorithm is used to estimate the parameter values from each subset of data. These estimated parameter sets represent the nonlinear model parameter values for different operating conditions.

Wishart (1995) presented a technique to identify and control induction machine using Artificial Neural Networks (ANNs). In this case, two systems are presented. The first one is a system to adaptively control the stator currents through identification of the electrical dynamics and the second one is a system to adaptively control the rotor speed through identification of the mechanical and current – fed system dynamics. ANN is introduced in both systems and observable forms of the models are described. Finally, the performances of these controlling schemes are compared with the standard vector control scheme.

In practice ANNs do not always work well because they experience problems associated with the accuracy of prediction [Krose, 1996]. The problem of premature convergence (will be discussed in detail in sub – section 3.1.3) is also critical for ANNs.

Karanayil (2005) proposed the application of ANNs and fuzzy logic for the on – line rotor time constant / rotor resistance adaptation. Fuzzy logic is simple to implement and estimators using fuzzy logic have no convergence issues. On the other hand, the ANN estimator has the advantages of faster execution speed and fault tolerant characteristics compared to the estimators implemented in a DSP based system. They have the attributes of estimating parameters of a non – linear system with good accuracy and fast response. The above benefits of using an ANN for parameter identification have been utilized to adapt the rotor resistance in a rotor flux oriented induction motor drive. However, this method is only applied for the identification of the rotor resistance and not for the rest of the electrical parameters of induction motors such as the magnetizing inductance which will be proved in this thesis that is a crucial parameter for a vector controlled drive.

Ursem and Vadstrup (2004) compared eight stochastic optimisation algorithms with respect to parameter identification of two induction motors. The eight algorithms represent four main groups of stochastic optimisation algorithms used today (local search (LS), evolution strategies (ESs), generational EAs, and particle swarm optimizers (PSOs)). From each group they included a simple and an advanced algorithm. Comparison of the algorithms showed that the two LS techniques had the worst performance of all eight algorithms. The simple population – based approaches had rather good performance, while the advanced algorithms had the best performance. Of the advanced techniques, the diversity – guided EA illustrated the best average performance for both motors and it seems to be a promising direction for future studies. However more precise parameter values can be achieved by improving the performance of the advanced techniques.

Bishop and Richards (1990) were one of the pioneers in the applicability of genetic algorithms (GAs) to the problem of the motor parameter determination. They used a simplified steady state model of an induction machine for parameter identification from load test data. The objective is to determine the values of the induction motor equivalent circuit except the stator resistance and rotor reactance which are found from other tests or handbook data and may be considered known. The motor load tests consist of measurement of motor terminal complex impedance, Z_m , at rated voltage for at least two different values of slip s_1 and s_2 . A measurement of electrical torque, T_m , is also made at each of the test speeds. The GA is now used to find a set of parameters such as to minimise the error function. The error is a function of Z_m and T_m at each different slip. If the test is to be repeated at several slips, the error will be the sum of all errors at each corresponding slip. This method has proved that GA can be used to parameter identification problems and experience indicates that it is robust and may find results under condition of noisy and inconsistent test data.

Nangsue *et al.* (1999) suggested motor parameter identification by using steady state models of approximate equivalent circuits, exact equivalent circuits and a deep bar circuits. In this case the fitness function is defined as the error between the input motor torques and the calculated ones using the circuit equations. The input torques are the full load, the locked rotor and the breakdown torques obtained

from real manufacturer's data. The objective of these models is the minimization of the fitness function errors. GA is then applied to determine suitably accurate parameters. In this determination problem, motor parameters are assumed to be constant, mostly because the intention is to use the parameters for system level studies where extreme precision is unnecessary. Higher accuracy may be obtained by letting the leakage reactances vary as a function of the current. This approach shows that the performance of GAs can be affected by numerical values of constants needed in implementation such as mutation or crossover rate. The outcome of the results indicates that the use of the approximate equivalent circuit prevents the GAs from determining the motor parameters accurately. As example given is the case of a 5 HP machine, since the error between the full load torque and the calculated one was 20% while for the deep bar model it reduces to 3%. Results showed that the deep bar model is a more accurate model representation in the motor parameter determination as gave better results, and particularly covering a wide speed range. This became a very important issue in modeling and controlling the induction machine and much investigation in selecting the accurate model of induction machine has been done. The drawback in this case is the calculations of the full load, locked rotor and breakdown torques from real manufacturer's data.

Alonge *et al.* (1999) and (2001) considered a nonlinear dynamic model of an induction machine for solving the problem of parameter identification. In this model it is assumed $L_{lr} = L_{ls}$ and so there are four electrical parameters to be found and also the mechanical system parameters such as inertia and friction of an inverter – fed IM. An experimental test is carried out consisting of a transient from standstill to certain speed and a mathematical model with the desired structure is implemented with the aim of simulation of the experimental test. In this method the GA's cost function is computed as a weighted sum of either square or absolute differences of the output stator currents acquired experimentally and those computed by simulation at the same instant. Analytical Least Squared Technique is applied to identify the parameters and a comparison is made with GAs with the aim to select the identification method, which gives the best results.

Huang *et al.* (1999) proposed a parameter identification of induction motor via GAs. The technique was undertaken using the starting performance with four different levels of measurement noise. For variable speed applications, the motor's general mathematical model based upon Kron's voltage equations is employed to estimate the parameters, and the motor's start – up performance is used as the measurement during the identification process. For comparison, the results of a Simple Random Search (SRS) method under the same condition are also given. It is concluded that GAs gives much better results than that of the SRS technique.

Chung *et al.* (2000) presented a parameter identification method for induction machines based on enhanced genetic algorithm that operates on real – valued parameter sets. The induction motor model used is similar to the nonlinear one proposed by Krause (2002). This model is discretized and the stator voltages and the speed of the rotor are used as the inputs to the model. Fourth order Runge Kutta method is employed to determine the states of induction motor model as for example rotor flux linkages and stator currents. Chung recommended, since simulating the machine in the whole transient response is computationally expensive, to study the effects of smaller time intervals for the given machine excitation so that computational cost of calculating fitness is reduced while still maintaining acceptable parameter estimates. The GA is run several times for each interval and the best parameter sets obtained are averaged, thus assuming the same parameters during transient period of machine. Concluding, some time intervals provided better estimation results (while comparing estimated stator currents and currents measured by exciting the induction motor under no load) suggesting that the conditions right after machine start – up are favorable for parameter estimation since at steady state the errors are always relatively high as the current values are small and thus there is insufficient information to estimate the electrical parameters. Even if the computational time applying GA at several times for each interval has been reduced enough, it was still high for this GA application real – valued parameter sets.

A universal drive should be applicable to different types of motors for various applications with unknown motor parameters and load characteristics. In order to meet the high performance dynamic requirements often demanded from electrical

drives, it is essential to develop controllers that overcome the influence of varying motor parameters, the influence of load variation and keeps the performance of the overall system unchanged. Therefore, self – commissioning control of electric drives was originally proposed by Jotten [Jotten, 1987] and Schierling [Schierling, 1988a], [Schierling, 1988b] and then has been developed for the control of induction drives [Khambadkone, 1991] and [Sumner, 1993]. Khambadkone and Sumner used the standard techniques for self – commissioning which consist of specific sequences of tests to measure electrical and mechanical parameters of the motor. In the electric drives market, there are two basic requirements for a universal drive: self – commissioning and auto – tuning. The first one concerns the initialization and starting of the electrical drive when it is connected to a new servo motor with some unknown parameters while the second one concerns the tuning of the control parameters when the electric drive is connected to a new mechanical load with unknown dynamic load dynamics [Tzou, 1997]. Some self – commissioning tests have been developed with regards to the magnetizing inductance which is usually identified at the rated flux [Boussak, 1988], [Bertoluzzo, 1997]. When the flux differs from the rated level, the magnetizing inductance changes due to the nonlinear behaviour of the iron core.

Cascella et al [2005] proposed an on – line auto – tuning based on a hybrid genetic algorithm for a vector – controlled PMSM drive. This method can be simply embedded as a fully – automated tool for industrial drives, or based on a PC which can access control data through an industrial communication protocol and it provides better performance with higher reliability than that obtained with accurate hand – calibrations. One of the drawbacks of this technique is that the on – line evolution of the control system may need many tests, and the controlled process can be critically stressed by poorly performing solutions. Lastly, a model – based design can be adopted in order to achieve a stable control system in a short time. Although only a sub – optimal tuning is achieved because the issues concerning the multiple inputs, system nonlinearities and uncertainties of the model – based design remain unaddressed.

Drive manufacturers (such as ABB and Control Techniques), are providing three types of ID autotune – runs for the machine parameters identification in closed

loop vector mode. These are a stationary/standstill test, a rotating/normal/full test and an inertia measurement test. The first will give moderate performance, whereas a rotating autotune will give improved performance as it measures the actual values of the motor parameters required by the drive. Lastly, an inertia measurement test should be performed separately to a standstill or rotating autotune.

- i. *Stationary/Standstill autotune test:* This test can be used when the motor is loaded and it is not possible to remove the load from the motor shaft. In practise, this mode is selected if a) motor cannot be decoupled from the driven equipment or load; b) if flux reduction is not allowed while the motor is running (i.e. in case of a motor with an integrated brake supplied from the motor terminals). The stationary test measures the stator resistance and transient inductance of the motor. These parameters are mainly used to calculate the current loop gains.
- ii. *Rotating/Normal/Full autotune test:* This test should only be used if the motor is unloaded. Firstly, it performs a stationary autotune before rotating the motor at 2/3 of motor rated frequency in the direction selected for approximately 30s. During the rotating autotune the stator inductance and the motor saturation breakpoints are modified by the drive.
- iii. *Inertia measurement test:* This test can measure the total inertia of the load and the motor. This is used to set the speed loop gains and to provide torque feed – forwards when required during acceleration.
The drive accelerates the motor in the direction selected up to 3/4 x rated load rpm and then back to standstill. In this case, the drive uses rated torque/16, but if the motor cannot be accelerated to the required speed then the drive increases the torque progressively to $\times 1/8$, $\times 1/4$, $\times 1/2$, $\times 1$ rated torque. Therefore the acceleration and deceleration times are used to calculate the motor and load inertia.

Naturally, there are certain drive limit settings need to be applied before the ID run, Upon the completion of ID run, the limit values are then set back to as required by the specific application.

The idea for the selection of the performance index for GAs implementation in this work comes from Trentin *et al.* (2006). Trentin proposed a new heuristic approach based on GAs to estimate mechanical and electrical parameters of an induction motor (IM) in all operative conditions using only speed transient measurements. The basic proposal is to use a GA routine off – line to optimize the motor parameters in a simulation model by recursively running the simulation and trying to minimise the error between the real measured speed response and the simulated one under the same experimental conditions. The validity of the proposed approach has been proven as the match between simulation and experimental test is very good.

The additional aspect in this thesis regarding Trentin's work is that it is emphasize the variation of the magnetizing inductance not only as a function of the field producing current but as well of the load current and without the need of any loading mechanism. It will be also shown that this variation is giving us valuable information at low field current levels.

1.3 Project Objectives

The primary objective in this thesis is the identification of induction motor electrical parameters in all operating conditions. The strategy of this work is developed using genetic algorithms (GAs), a heuristic optimisation technique based on Darwin's theory of natural selection and survival of the fittest. More specific, this thesis addresses the below distinct objectives:

- Design, develop and Simulink Implementation of an Induction Machine Model.
- Achieve evaluation of Electrical Parameter of Induction Motors.
- Analysing Parameter Sensitivity and Adaptation.
- Improve Control Performance of Induction Motors by Using a Genetic Algorithms Optimization Technique.

- Modify GA strategy based on migration and artificial selection (MGAMAS) in order to improve the accuracy and computational time.
- Improve Genetic Algorithm for Structural Identification via the Search Space Reduction Method (SSRM) in order to reduce the search space for those parameters that converge quickly.
- Applying GAs to Vector Controlled Drives for Average Parameter Identification.
- Develop a novel approach for identifying induction motor electrical parameters under different operating conditions (different flux – and torque – producing reference currents, I_{sd}^* and I_{sq}^* respectively).
- Analysis and Sensitivity of the Estimated Parameters.
- Validation of parameter estimation.

1.4 Thesis Overview

The thesis has been structured in the following manner:

Chapter 2 introduces the vector control theory of IM drives. The Field – Oriented control is discussed with focused on the Indirect Vector Control technique as this control scheme is being used in this thesis. Fully description of the system is given with a block diagram illustrating the indirect vector control scheme without field weakening. The PI controller design of both dq current and speed control is given. Finally, the last section of this chapter will be focused on the effects of parameter sensitivity and adaptation.

The strategy developed in this thesis is based on genetic algorithms. Hence, Chapter 3 is dedicated to describing the key features of a classical genetic algorithm and how GA theory is able to account for the ability of algorithms to converge to good solutions. It also highlights the reason why GAs is the preferred choice. In this chapter the proposed identification strategy is also presented. The strategy consists of a Search Space Reduction Method (SSRM) which uses a

Modified GA based on Migration and Artificial Selection (MGAMAS) as the main search engine. The method is designed to provide accurate and reliable identification results for dynamic problems. The strategy includes some new operators and procedures and the motivation behind these is explained.

Chapter 4 presents the practical hardware and software requirements and implementations. The description of the experimental setup will be presented which was used to develop and test the vector control algorithms. As a first step the squirrel-cage 4kW induction machine is introduced. Finally, it will be discussed the dSPACE DS1104 controller card, and the Digital Signal Processor to Motor Drive Interface System.

Chapter 5 presents an accurate non – intrusive and fast method for evaluation of the electrical induction motor parameters. The motor used for this research is a 4 kW, 4 – pole I.M. A Matlab model was utilised within the heuristic GA based identification routine. The evaluation of the average electrical motor parameters can be achieved by minimising, using a GAs approach the error between the experimental response (speed or current) and the respective one obtained by a Matlab – Simulink model. Discuss and analysis of the results is verified through a comparison of speed and current response. Finally the sensitivity of the GA estimated parameters were effectively used so as to find the influence of each parameter to the optimisation's fitness function.

In Chapter 6, a non – intrusive approach for identifying induction motor equivalent circuit parameters in function of varying load and flux levels is represented. More specifically, the variation of the magnetising inductance (L_m) as a function of flux and load levels will be demonstrated and results will be analytically explained.

Chapter 7 presents the main conclusions of this work. This thesis concludes with a series of suggestions for further investigation.

Chapter 2

Vector Controlled IM Drives

2.1 Introduction

It was mentioned in the first chapter that the main disadvantage of an induction motor was that it could not be controlled easily as a DC machine. So, a sophisticated technique called Vector Control de – couples the vectors of field current and armature flux so that they may be controlled independently to provide fast transient response. In this chapter, we will present the fundamentals of vector control of an induction motor.

Firstly, a DC drive analogy to an Induction Motor Drive will be analysed. In that way, it will be shown how the DC machine – like performance can be extended to an induction motor and this can be achieved only if the machine control is considered in a synchronously rotating reference frame, where the sinusoidal variables appear as dc quantities in steady state.

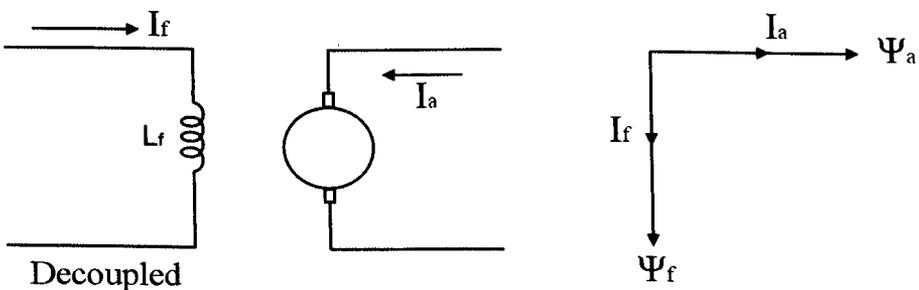
Then, a brief explanation of the flux angle λ will be introduced as the angle between one of the stator phases and the rotor flux.

In a later section, the idea of Field – Oriented Control is widely discussed with focus on the Indirect Vector Control as this control scheme is being used in this thesis. Fully description of the system is given, within a block diagram illustrating the indirect vector control scheme without field weakening. The estimation of both dq current and speed control (PI controllers) is presented. The complete simplified block diagrams of both control loops will be also displayed.

Finally, the last section of this chapter will be focused on the effects of parameter sensitivity and adaptation. In high – performance drives, a priori knowledge of the machine parameters is required, which makes the indirect vector control scheme machine parameter dependent. The importance of parameter adaptation is discussed and categorized based on the extent of use of the induction motor parameters.

2.2 DC Drive Analogy to an Induction Motor Drive

As mentioned above, a vector controlled motor drive operates like a separately excited dc motor drive. Figure 2.1 shows the analogy between them.



(a)

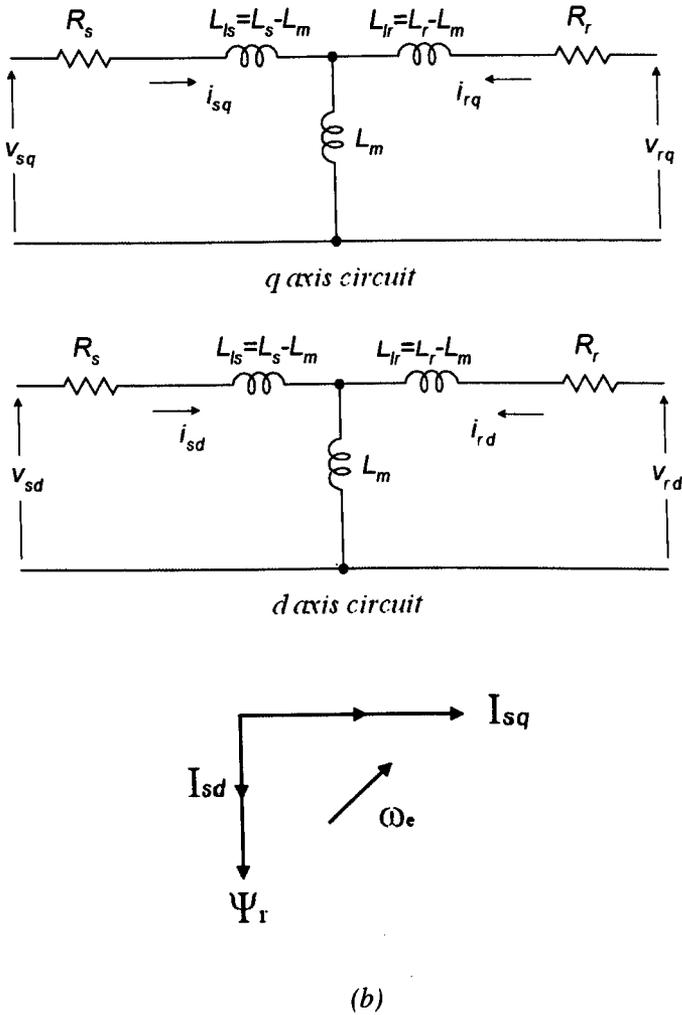


Figure 2.1: (a) Separately excited dc motor, (b) Vector – Controlled I.M.

In a dc machine, the developed torque is given by:

$$T_e = K_t \psi_f \psi_a = K_t' I_a I_f \tag{2.1}$$

Where ψ_f is the field flux, ψ_a is the armature flux, I_a is the armature current, I_f is the field current and K_t , K_t' are constant values. It can be seen from the figure 2.1(a) that the field flux ψ_f produced by the current I_f is perpendicular to the armature flux ψ_a which is produced by the armature current I_a . These space

vectors which are stationary in space are decoupled or orthogonal in nature [Bose, 2002]. As a consequence of this, the field flux ψ_f is not influenced when torque is controlled (by controlling the current I_α) and therefore fast transient torque and current response is achieved. Due to decoupling, the field flux ψ_f is affected when the field current I_f is controlled, while the armature flux ψ_α remains the same. In most cases an induction motor cannot give such fast response as in DC motors because of the inherent coupling problem.

The same methodology can be applied to an induction motor drive if the machine control is considered in a synchronously rotating reference frame (dq). In figure 2.1 (b), the control current inputs I_{sd}^* and I_{sq}^* (asterisk is used in symbols to show command or reference quantities) are representing the direct and quadrature axis component of the stator current, respectively, in a synchronously rotating reference frame. Therefore, the rotor flux vector ψ_r of IM is analogous to the field flux ψ_f of dc machine, I_{sd} is analogous to field current I_f and I_{sq} is analogous to armature current I_α of a dc machine.

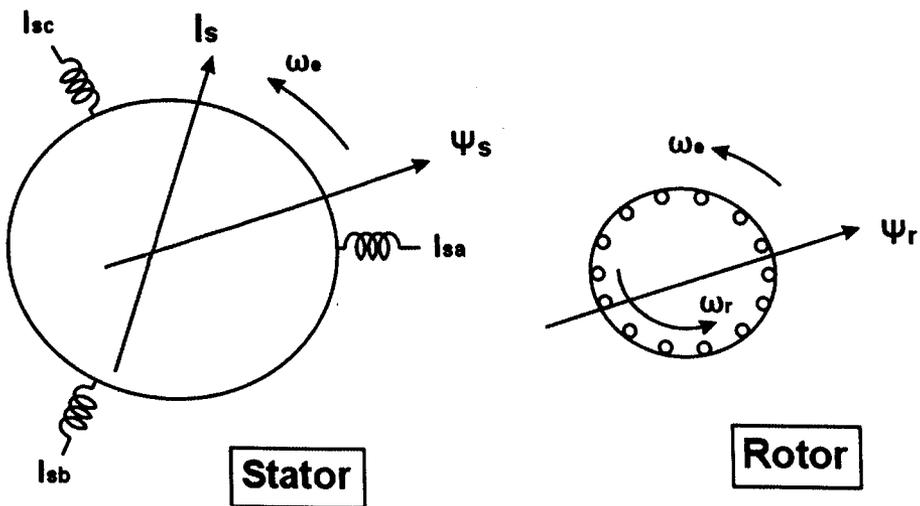


Figure 2.2: Vectors inside an induction motor

In order the analogy of the currents to be more understandable, consider that a three – phase supply will produce a current vector I_s that rotates at the supply

frequency ω_e . If this current is supplied to the spatially distributed winding of an induction motor, then a rotating magnetic field ψ_r that rotates at the supply frequency will be produced. Figure 2.2 shows these two vectors that rotate at the same speed, but with a phase difference of θ (see Figure 2.3).

In vector control theory, I_s is resolved into two components I_{sd} and I_{sq} in a rotating co – ordinate axis. The rotor flux vector ψ_r is aligned with d – axis so that there is no relative rotation between the vectors ψ_r and I_{sd} as shown in figure 2.3.

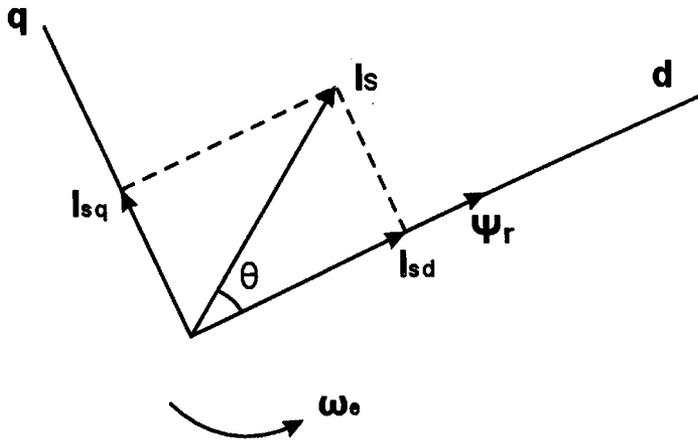


Figure 2.3: Resolving I_s into two components in a rotating frame

As a result, I_{sd} and I_{sq} are called the field and torque producing currents respectively if the coordinate axis (the dq – axis) lies on ψ_r .

The torque can be expressed as:

$$T_e = K_t'' \cdot \hat{\psi}_r \cdot I_{sq} \quad (2.2)$$

or

$$T_e = K_t''' \cdot I_{sd} \cdot I_{sq} \quad (2.3)$$

where $\hat{\psi}_r = \text{absolute } \bar{\psi}_r$ is the peak value of the sinusoidal space vector and K_t'' , K_t''' are constant values. This dc – machine like performance is only possible if

the system field orientated (on the rotor flux) and this can be achieved when the d – axis of the co – ordinate reference frame lies in the direction of rotor flux ψ_r and I_{sq} is established perpendicular to it. Consequently, when I_{sq}^* is controlled, it affects the actual I_{sq} current only but not the rotor flux ψ_r . In the same way, when I_{sd}^* is controlled, it can change the flux (field producing current) but not the IM torque current.

2.3 The flux Angle λ

The angle λ is defined as the angle between one of the stator phases (e.g. the α axis) and the rotor flux vector ψ_r , as indicates figure 2.4.

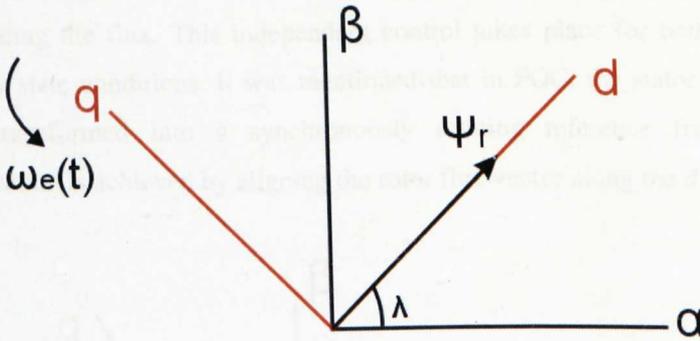


Figure 2.4: The rotor flux angle λ

The dq frame which is aligned along the rotor flux vector rotates at instantaneous frequency ω_e .

$$\omega_e(t) = \frac{d}{dt} \lambda(t) \quad (2.4)$$

$$\lambda(t) = \int \omega_e(t) dt \quad (2.5)$$

It is important that the flux angle $\lambda(t)$ is known at every instant in time, both during steady – state and motor transient operation (i.e. ω_e changing). This angle can be obtained by both Direct and Indirect Vector Control.

2.4 Field – Oriented Control

So far, we have discussed that the discovery of vector control in the beginning of 1970's by Blaschke brought a reawakening in the high performance control of AC drives. Vector control is also known as decoupling, transvector or orthogonal control because of the dc machine – like performance. Blaschke examined how field orientation happens naturally in a separately excited dc motor in which the field flux ψ_f produced by the field current I_f is perpendicular to the armature flux ψ_a which is produced by the armature current I_a . In an induction machine a related situation can be formed with appropriate control of stator currents in the synchronously rotating frame of reference.

Field – oriented control (FOC) provides independent (decoupling) control for the two components of stator current, one producing the torque and the other producing the flux. This independent control takes place for both transient and steady state conditions. It was mentioned that in FOC, the stator phase currents are transformed into a synchronously rotating reference frame and field orientation is achieved by aligning the rotor flux vector along the d – axis.

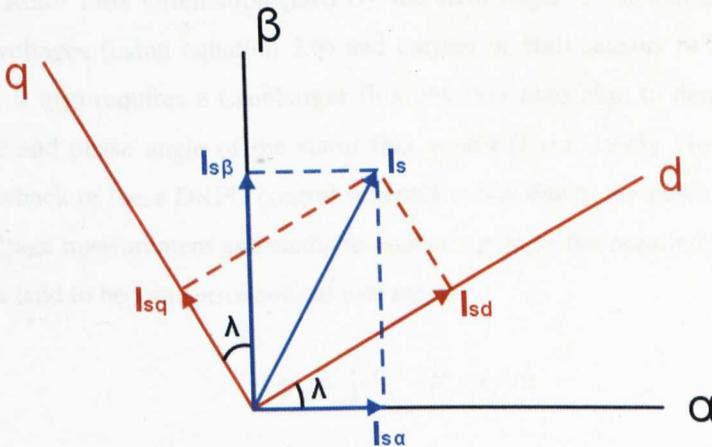


Figure 2.5: Stator current space vector and its component in (α, β) and in the dq rotating reference frame

Figure 2.5 shows the stator current space vector and its component in (α, β) and in the dq rotating reference frame. The dq – axis model of the induction motor with the reference axes rotating at synchronous speed ω_e is given by Bose [Bose, 2002].

The idea of field orientation indicates that the current components supplied to the machine should be oriented in phase (flux component) and in quadrature (torque component) to the rotor flux vector $\vec{\psi}_r$, such that the rotor flux is aligned to the d – axis. This results to the mathematical constraint $\psi_{rq} = 0$.

This control scheme is usually referred to as Rotor Flux Oriented Control (RFOC) as the control is performed on quantities obtained in the synchronous reference frame and when the rotor flux vector is chosen for decoupling. One of the basic schemes of field orientation is the Indirect Field Oriented control (IFO) that will be examined analytically in the next sub – section.

2.4.1 Indirect Vector Control

In Direct Rotor Flux Orientation (DRFO), the field angle is calculated by using terminal voltages (using equation 2.6) and current or Hall sensors or flux sense windings. It also requires a Luenberger flux observer algorithm to determine the amplitude and phase angle of the stator flux vector [Lyra, 1995]. However, the main drawback of these DRFO control schemes is that due to the inevitable errors in the voltage measurement and stator resistance estimate the required integration of signals tend to become erroneous at low speed.

$$\psi_s = \int (V_s - R_s \cdot I_s) dt \quad (2.6)$$

As a simplification of this, Indirect Vector Control scheme is based in voltage vector orientation, as it is illustrated by its phasor diagram shown in Figure 2.6. The $\alpha\beta$ axis is fixed on the stator and is stationary, while the $(\alpha)(\beta)$ axis which is fixed on the rotor is moving at speed ω_r .

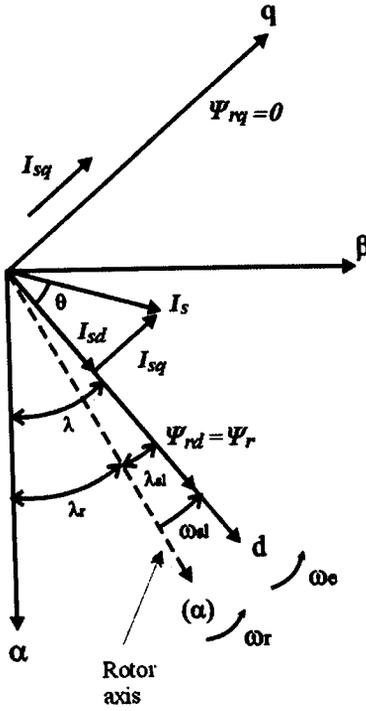


Figure 2.6: Indirect vector control phasor diagram

Synchronously rotating axis dq is rotating ahead of the $(\alpha)(\beta)$ axis by the positive slip angle λ_{sl} corresponding to slip frequency ω_{sl} . Since the rotor pole is directed on the d axis and $\omega_e = \omega_r + \omega_{sl}$, then:

$$\lambda = \int \omega_e dt = \int (\omega_r + \omega_{sl}) dt = \lambda_r + \lambda_{sl} \quad (2.7)$$

In Appendix D it is shown that the two “vector control equations” are:

$$\frac{L_r}{R_r} \frac{d}{dt} I_{mrd} + I_{mrd} = I_{sd} \quad (2.8)$$

which is used to derive the magnetising current and in the control of flux and:

$$\omega_{sl} = \frac{1}{\tau_r I_{mrd}} I_{sq} \quad (2.9)$$

where, the rotor time constant τ_r can be defined as:

$$\tau_r = \frac{L_r}{R_r} \quad (2.10)$$

Equation (2.7) is used to derive the flux angle λ . Placing equation (2.9) to the equation (2.7), the flux angle λ can be rewritten as:

$$\lambda = \int \left(\omega_r + \frac{I_{sq}}{\tau_r \cdot I_{mrd}} \right) dt \quad (2.11)$$

Figure 2.7 illustrates the vector control scheme block diagram of the Indirect Rotor Flux Orientation.

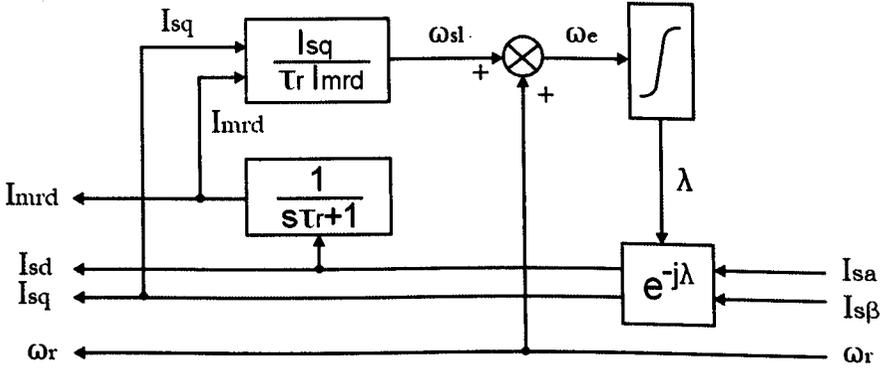


Figure 2.7: Indirect Rotor Flux Orientation

However, in this thesis the IRFO control is operating without field weakening and then figure 2.7 can be simplified, replacing I_{mrd} by I_{sd} .

$$I_{mrd} = I_{sd} \quad (2.12)$$

Substituting the above equation (2.12) to equations (2.9) and (2.11), we get:

$$\omega_{sl} = \frac{1}{\tau_r I_{sd}} I_{sq} \quad (2.13)$$

$$\lambda = \int \left(\omega_r + \frac{I_{sq}}{\tau_r \cdot I_{sd}} \right) dt \quad (2.14)$$

The torque developed by the induction motor is given by:

$$T_e = \frac{2}{3} \frac{p}{2} (\psi_{rd} I_{rq} - \psi_{rq} I_{rd}) \quad (2.15)$$

Because of decoupled control (letting $\psi_{rq} = 0$), equation (2.15) is written as follows:

$$T_e = \frac{2}{3} \frac{p}{2} \psi_{rd} I_{rq} \quad (2.16)$$

From Appendix B, equation B9 can be written as:

$$I_{rq} = \frac{1}{L_r} \psi_{rq} - \frac{L_m}{L_r} I_{sq} \quad (2.17)$$

Using equation 2.17 to replace I_{rq} in 2.16 and letting $\psi_{rq} = 0$, it follows:

$$T_e = \frac{2}{3} \frac{p}{2} \frac{L_m}{L_r} \psi_{rd} I_{sq} \quad (2.18)$$

Figure 2.8 shows how the rotor flux position can be obtained by integrating the sum of the rotor speed and the command slip frequency calculated using equation 2.13.

Concluding, it can be said that Field Orientation occurs according to the following argument:

- With fast current loop, assume $I_{sq} = I_{sq}^*$
- ω_e is being forced on machine via PWM generator
- ω_r is measured (i.e. real)
- ω_{sl} in controller = real slip in the machine
- if τ_r is correct, slip gain is right
- the below equation is true only if machine is field orientated

$$\omega_{sl} = \frac{I_{sq}}{\tau_r I_{sd}} = \kappa \cdot I_{sq}$$

- $I_{sq} (= I_{sq}^*)$ must be the real q axis current in the machine
- therefore the machine is field orientated and λ must be correct

2.5 Controller Design and System Bandwidths

For estimating the PI controllers of the Indirect Vector Control Scheme it is necessary to know the electrical and mechanical parameters of the induction motor as well as few other parameters depending on the needs of the model. The motor used in this thesis is a 4kW, 4 – pole induction motor which is coupled to a 10kW DC machine. The nameplate data of the examined motor are: $U=415$ V, Δ connection, $I=8.4$ A and $n= 1420$ rpm. Using the traditional methods (see Appendix C) based on “no-load” and “locked rotor” tests the induction motor equivalent circuit parameters can be identified. The values of them are: Stator resistance $R_s = 5.25\Omega$, Rotor resistance $R_r = 3.76\Omega$, Magnetizing inductance $L_m = 0.5343H$, Stator leakage inductance $L_{ls} = 0.04H$ and Rotor leakage inductance $L_{lr} = 0.033H$. In addition, the mechanical parameters should be known for evaluating the PI speed controller. Therefore, the moment of inertia J and the friction B of the whole system (induction motor and DC machine) is $J=0.152Kgm^2$ and $B=0.0147Nms$ correspondingly (found by deceleration test).

2.5.1 dq current control loop

Consider the design of the I_{sq} controller within the I_{sq} loop shown in figure 2.9. The I_{sd} loop will be identical. All the dynamics or delays between the controller output V_{sq}^* and the feedback signal I_{sq} need to be determined so that the PI controller terms can be calculated.

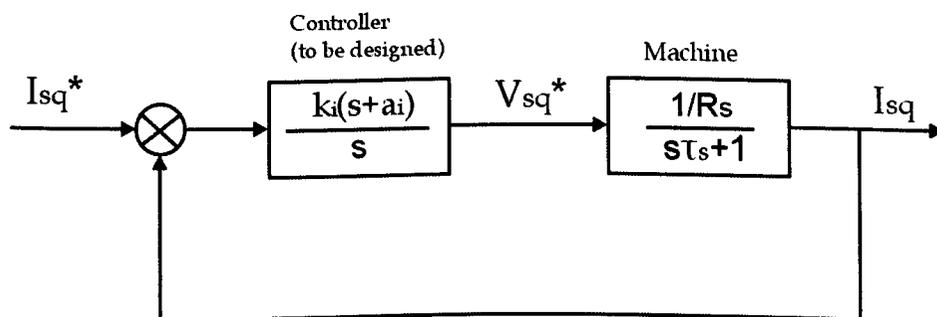


Figure 2.9: Block diagram of the current control loop

Figure 2.9 demonstrates the block diagram of the dq current control loop. From the machine block diagram, τ_s is called the effective armature time constant where:

$$\tau_s = \frac{\sigma L_s}{R_s} \quad (2.19)$$

and

$$\sigma L_s = \frac{L_r L_s - L_m^2}{L_r} \quad (2.20)$$

where σ is the leakage coefficient.

After calculations from equation 2.20, it gives:

$$\sigma L_s = 0.07108$$

Substituting the values of σL_s and R_s to equation 2.19, it gives:

$$\tau_s = \frac{0.07108}{5.25} \Rightarrow \tau_s = 0.0135 \text{ sec}$$

Thus, the machine transfer function will be:

$$\frac{1/R_s}{s\tau_s + 1} = \frac{1/5.25}{0.0135s + 1} = \frac{14.11}{s + 74.07}$$

Then,

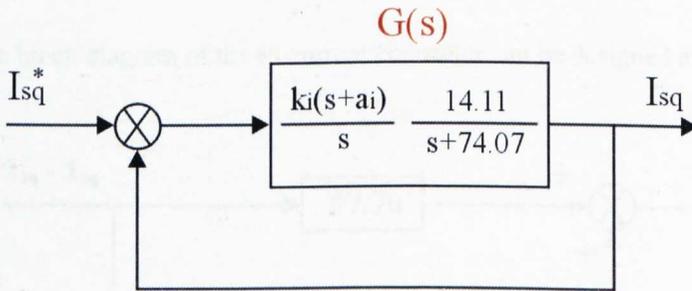


Figure 2.10: Simplified block diagram of the current control loop

The closed loop transfer function will be:

$$\begin{aligned} \frac{G(s)}{1+G(s)} &= \frac{\frac{14.11k_i(s+a_i)}{s(s+74.07)}}{\frac{s(s+74.07)+14.11k_i(s+a_i)}{s(s+74.07)}} = \frac{\text{nominator}}{s^2 + 74.07s + 14.11k_i s + 14.11k_i a_i} = \\ &= \frac{\text{nominator}}{s^2 + (74.07 + 14.11k_i)s + 14.11k_i a_i} \end{aligned}$$

The desired closed loop denominator have roots where chosen to be with $\omega_n = 100\text{Hz} = 628.32\text{rad/sec}$ and $\zeta=0.707$.

The second order form is: $s^2 + 2\zeta\omega_n s + \omega_n^2$ or

$$s^2 + 888.44s + 394786.0224 \quad (2.21)$$

Hence, the denominator from the closed loop transfer function is:

$$s^2 + (74.07 + 14.11k_i)s + 14.11k_i a_i \quad (2.22)$$

From the equations 2.21 and 2.22:

$$\left\{ \begin{array}{l} 74.07 + 14.11k_i = 888.44 \\ 14.11k_i a_i = 394786.0224 \end{array} \right\} \Leftrightarrow \text{after calculations we get:}$$

$$\left\{ \begin{array}{l} k_i = 57.71 \\ a_i = 484.774 \end{array} \right\}$$

Hence, the block diagram of the PI dq current controller can be designed as:

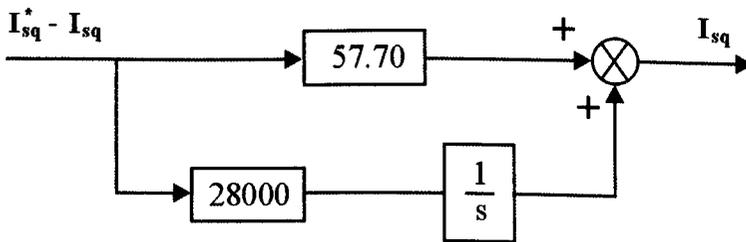


Figure 2.11: PI dq current controller

2.5.2 Speed Control Loop

The speed control loop takes the speed error ($\omega_r^* - \omega_r$) and outputs torque demand which is proportional to an I_{sq}^* demand.

Delays are:

- Delay of I_{sq}^* loop (same as I_{sq} loop)
- The mechanical dynamics: assume first order delay τ_m
- Possible filter for speed measurements

As there is no field weakening, the torque is proportional only to I_{sq} . The control loop is shown in figure 2.12.

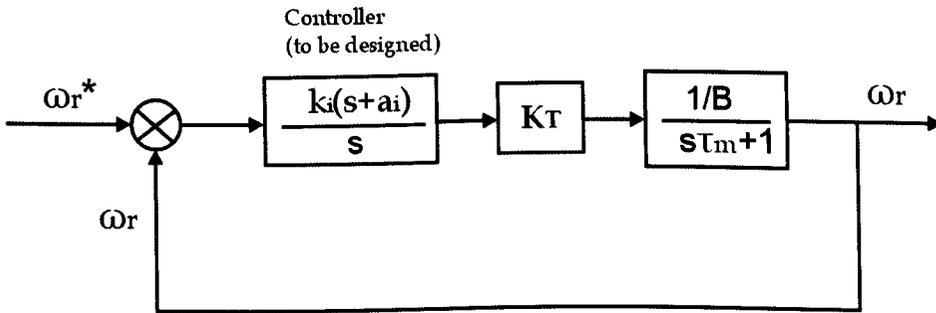


Figure 2.12: Block diagram of the speed control loop

where:

$$\tau_m = \frac{J}{B} \quad (2.23)$$

and

$$K_T = \frac{2}{3} \frac{p}{2} \frac{L_m^2}{L_r} I_{sd}^* \quad (2.24)$$

After calculations, equation (2.23) will give: $\tau_m = 10.63$

From equation (2.24) the only unknown parameter is the current I_{sd}^*

From no – load test, I_{sd}^* can be calculated using the equivalent circuit of figure 2.13.

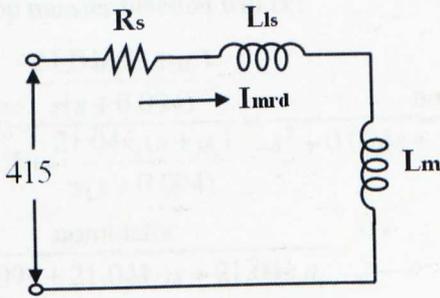


Figure 2.13: Equivalent circuit

The current I_{mrd} is equal to:
$$I_{mrd} = \frac{415}{\sqrt{(R_s)^2 + X_L^2}}$$

where: $X_L = 2\pi \cdot (L_{ls} + L_m) \cdot f \Rightarrow$ after calculations

$$\Rightarrow X_L = 178.132\Omega$$

Then,
$$I_{mrd} = \frac{415}{\sqrt{5.25^2 + 178.132^2}} \Rightarrow I_{mrd} = 2.32A$$

At no-load condition, $I_{sq} \approx 0$ and because $|I_{mrd}| = \sqrt{I_{sq}^2 + I_{sd}^2}$ then

$$I_{sd}^* = \frac{3}{2} I_{mrd} \sqrt{2} = \frac{3}{2} 2.32 \sqrt{2} \Rightarrow I_{sd}^* = 4.9A,$$

Where $\frac{3}{2}\sqrt{2}$ is the called “one and a half – times Peak Convention”.

Therefore, substituting the above value to equation (2.24) we will get:

$$K_T \approx 3.288$$

Then the simplified control loop can be represented as shown in figure 2.14.

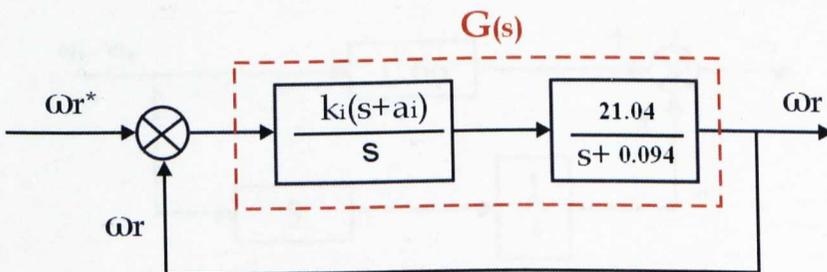


Figure 2.14: Simplified block diagram of the speed control loop

So, the closed loop transfer function will be:

$$\begin{aligned} \frac{G(s)}{1+G(s)} &= \frac{\frac{21.04k_i(s+a_i)}{s(s+0.094)}}{1+\frac{21.04k_i(s+a_i)}{s(s+0.094)}} = \frac{\text{nominator}}{s^2+0.094s+21.04k_i s+21.04k_i a_i} = \\ &= \frac{\text{nominator}}{s^2+(0.094+21.04k_i)s+21.04k_i a_i} \end{aligned}$$

The desired closed loop denominator have roots with $\omega_n = 10$ rad/sec and $\zeta=0.707$.

The second order form is: $s^2 + 2\zeta\omega_n s + \omega_n^2$ or

$$s^2 + 14.14s + 100 \tag{2.25}$$

Hence, the denominator from the closed loop transfer function is:

$$s^2 + (0.094 + 21.04k_i)s + 21.04k_i a_i \tag{2.26}$$

From the equations (2.25) and (2.26) above:

$$\left\{ \begin{array}{l} 0.094 + 21.04k_i = 14.14 \\ 21.04k_i a_i = 100 \end{array} \right\} \Leftrightarrow \text{after calculations we get:}$$

$$\left\{ \begin{array}{l} k_i = 0.66 \\ a_i = 7.12 \end{array} \right\}$$

Therefore, the block diagram of the PI speed controller can be designed as the below figure 2.15.

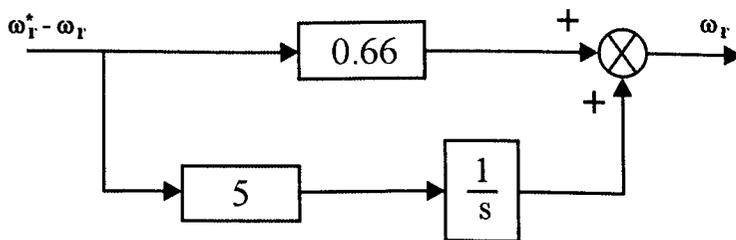


Figure 2.15: PI speed controller

2.5.3 Voltage Compensation

Voltage compensation is used in our model to improve the performance of the dq control loops. Consider the two stator equations D1 and D2 (Appendix D) in field – orientated co – ordinates ($\psi_{rq} = 0$).

$$u_{sd} = R_s I_{sd} + \sigma L_s \frac{d}{dt} I_{sd} - \omega_e \sigma L_s I_{sq} + \frac{L_m}{L_r} \frac{d}{dt} \psi_{rd} \quad (2.27)$$

$$u_{sq} = R_s I_{sq} + \sigma L_s \frac{d}{dt} I_{sq} + \omega_e \sigma L_s I_{sd} + \omega_e \frac{L_m}{L_r} \psi_{rd} \quad (2.28)$$

If we had:

$$u_{sd} = R_s I_{sd} + \sigma L_s \frac{d}{dt} I_{sd} \quad (2.29)$$

$$u_{sq} = R_s I_{sq} + \sigma L_s \frac{d}{dt} I_{sq} \quad (2.30)$$

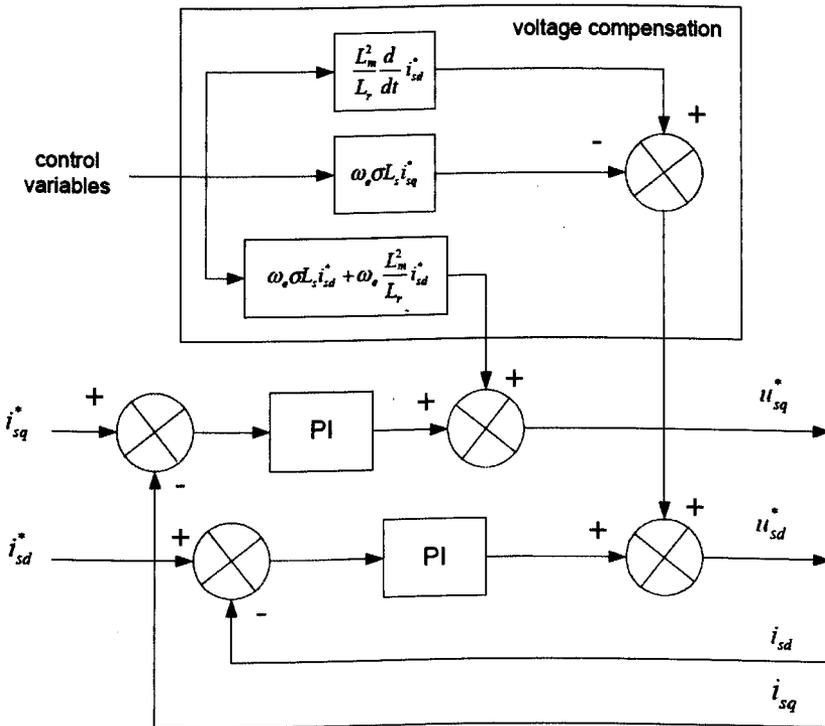


Figure 2.16: Voltage compensation

Then we have a linear dynamic equation between the dq voltages and the dq currents. This makes for an easy and linear design of the dq current loops.

This is achieved by adding the extra terms in equations (2.27) and (2.28) as feed – forward or compensations terms. Figure 2.16 shows the block diagram of the voltage compensation and the additional blocks. As it can be seen, the command values of I_{sq} and I_{sd} are used in order to have smoother inputs without too much noisy responses.

2.6 Parameter Sensitivity and Adaptation

It is needless to mention that the machine parameters in the equivalent circuit (Appendix A) changes during different operating conditions. In high – performance drives with induction motors, the accurate information of electrical parameters is of high importance. Variations in the machine’s parameters detune both steady – state and dynamic operation of induction motor drives. However, the precise knowledge of these parameters is extremely difficult to find as they are not constant and change with temperature, magnetic saturation and slip frequencies (due to skin effect). Both rotor and stator resistances change linearly (increase) due to temperature and depend mainly on the temperature coefficient of the resistance of the material. In addition, rotor resistance can change significantly with rotor frequency due to skew/proximity effecting machines with double – cage and deep – bar rotors. The skin effect also causes an increase of resistance but a decrease of leakage inductance. The magnetising inductance is subjected to saturation with higher magnetising current. Saturation at higher currents can be caused in both the stator and rotor leakage inductances [Bose, 2002].

In motors with indirect field orientation control, it is very important to have an accurate representation of the rotor time constant (τ_r) of the machine as control errors resulting from inaccurate knowledge of (τ_r) spoil the orientation, producing steady – state errors and transient oscillations of the flux and torque [Trzynadlowski, 2001]. To be more precise, the slip relation given in equation

(2.13) depends mainly on the value of the rotor time constant and is employed to obtain the correct subdivision of the stator current into the flux and torque components, I_{sd} and I_{sq} correspondingly. It is clear then that if the rotor time constant is not correctly known, this subdivision will not be correctly achieved and as a result the controller will be detuned due to the loss of correct field orientation.

Besides, there is a load torque disturbance effect. It is reported by Sumner & Asher [1991] and [1993] that there is indication for change of direct axis inductance and rotor time constant (τ_r) with load. There is no doubt that τ_r changes with load current due to heating. However, Sumner and Asher evidence an instantaneous change in τ_r with load current which is not related to the thermal effect as this can take long time to indicate. It was identified thus that τ_r changes with load current as a result of cross – coupling due to rotor skewing [Gerada, 2003]. In addition to the above, it is worthy to mention that the identifying rotor time constant schemes are also based on the constant parameter induction machine model. These hypotheses can lead to highly inaccurate results using these adaptive methods rather than a fixed τ_r [Sokola, 2000]. Hence, for the correct determination of the resistance variation, the vector control system should take into account other machine non – linearities and compensate for them [Sokola, 2000], [Levi, 1996a], [Levi, 1996b] and [Levi, 1994]. For example, Levi has looked at the effect of double – cage and deep – bar structures [1996a] and core losses [1996b] on IRFO. However, these modifications are rarely used in commercial drives.

Researchers found various methods to minimise the consequences of parameter sensitivity in indirect vector control schemes. These parameter adaptation schemes are classified as direct by direct monitoring the alignment of the flux and torque [Gabriel, 1982], and as indirect by measurement and estimation of rotor flux [Ohtani, 1983], [Ishihara, 1983], [Nagase, 1983], deviation of the field angle [Schumacher, 1983], or a combination of the rotor flux and torque – producing component of the stator current [Akamatsu, 1981], [Krishnan, 1987]. Both of these direct and indirect schemes for parameter adaptation algorithm present

different kind of disadvantages depending on their use of additional transducers or on their dependency on the motor parameters [Krishnan, 1991].

In the next sub – sections will be discussed the variation effects of machine parameters on the steady state and dynamic performance of an indirect vector controlled induction motor drive. However, before that it is desirable to give an explanation about the detuned operation through the vector (phasor) diagram.

2.6.1 Detuned Operation

From figure 2.8, it can be illustrated that the two current components I_{sq}^* and I_{sd}^* that represent torque and flux demands respectively, are the input values. For steady state, the rotor loop equation from the conventional equivalent circuit can be written as:

$$j\omega_e L_m I_s + \left[\frac{R_r}{s} + j\omega_e (L_m + L_{lr}) \right] I_r = 0 \quad (2.31)$$

Solving equation 2.31 for I_r :

$$I_r = \frac{-j\omega_s^* L_m I_s}{R_r + j\omega_s^* L_r} \quad (2.32)$$

where ω_s^* is the calculated value of slip frequency using equation 2.13 as the slip calculator. For steady state condition, equation 2.32 can be rewritten in terms of d – q variables, as the currents I_r and I_s can be replaced by the vector d – q currents I_{rdq} and I_{sdq} respectively. Hence:

$$I_{rdq} = \frac{-j\omega_s^* L_m I_{sdq}^*}{R_r + j\omega_s^* L_r} \quad (2.33)$$

Figure 2.17 illustrates the vector (phasor) diagram for IFOC with correct value of slip calculator time constant [Nordin, 1985]. From that figure, the phase angle of the rotor self – impedance θ locates the position of the rotor current I_{rdq} . The

phase angle depends on the calculated value of slip frequency and the rotor time constant:

$$\theta = \tan^{-1}(\omega_s^* \cdot \tau_r) \quad (2.34)$$

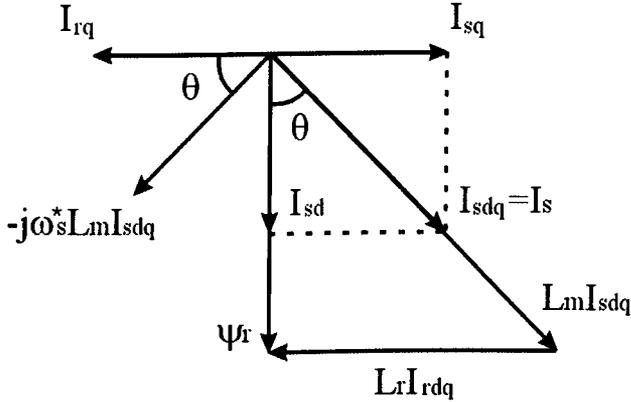


Figure 2.17: Vector (phasor) diagram for IFOC with correct value of slip calculator time constant

Correct field orientation is achieved when the rotor current will be exactly in opposition to the torque command current I_{sq}^* and this is applied if the reference time constant τ_r^* is equal to the actual rotor time constant τ_r . In this case, the rotor flux ψ_r , which is orthogonal to the rotor current I_{rdq} in the steady state, will be aligned with I_{sd} in the d – axis [Novotny, 1996].

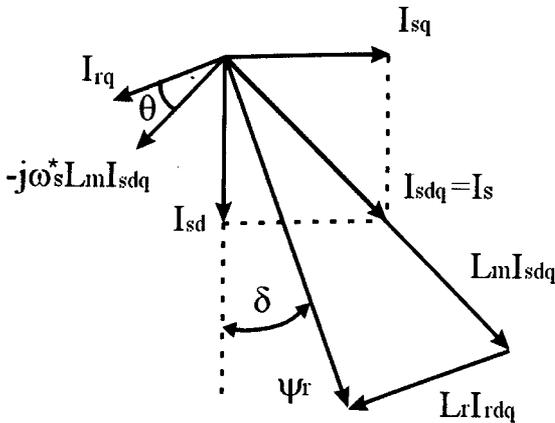


Figure 2.18: Vector (phasor) diagram for IFOC with slip calculator time constant too large (ω_s^* too small)

Detuned operation takes place with an incorrect slip ω_s^* , because of an incorrect slip calculator time constant τ_r^* as shown in figure 2.18. Assuming the slip is too small, then comparison between figure 2.17 and 2.18 shows that the phasor $-j\omega_s^* L_m I_{sdq}$ is shorter but has the same position as before. The angle θ is smaller as the slip is smaller and as a result, the rotor current I_{rq} doesn't lie in the q – axis. This means that the rotor flux ψ_r in no longer lie in the d – axis and is shifted by an angle δ so as to maintain orthogonality with the rotor current I_{rq} . Clearly, it can be noted that the magnitude of the rotor flux is larger than the value corresponding to the correct slip, meaning that the torque will also be incorrect and will not correspond properly to the torque command I_{sq} .

The effect of this mismatch of the rotor flux and torque under steady state and dynamic conditions are going to be described next.

2.6.2 Variation Effect for Steady – state

Several researchers examined the effect of errors in the slip calculator for steady state performance and several conditions of operation. Nordin (1985) investigated and plotted the steady state performance curves in terms of rotor flux and output torque against the ratio of actual to predicted rotor time constant τ_r / τ_r^* for different cases. In the next sub – chapters, a brief explanation of the steady – state effects will be given and the importance of the accurate knowledge of the motor parameter will be shown.

2.6.2.1 Constant Flux and Torque Commands

In the first case, it is assumed that both torque (I_{qs}^*) and flux (I_{ds}^*) commands are constant values and also the saturation effect is neglected. The flux curve [Nordin, 1985] shows that the flux ψ_r increases from its correct value as the predicted time constant in the slip calculator gets larger compared to the actual one

($\tau_r / \tau_r^* < 1.0$). However, the torque output decreases. In contrast, while the ratio τ_r / τ_r^* is bigger than 1 ($\tau_r / \tau_r^* > 1.0$), both the rotor flux and the torque output decrease with increase in error. Hence:

- When $\tau_r \downarrow$ then $\psi_r \uparrow$ and $T \downarrow$
- When $\tau_r \uparrow$ then $\psi_r \downarrow$ and $T \downarrow$

Or else, from the equation 2.13, the predicted slip frequency ω_s^* gets smaller than its correct value through an increase in rotor flux and vice versa.

It can be noticed that during the steady state operation the output torque characteristic always decreases whenever the predicted time constant or slip changes from its correct value. This has also been proved by the torque mathematical equation in Nordin's paper (1985) as:

$$T = \frac{2}{3} \left[\frac{p}{2} \right] \left[\frac{L_m}{L_r} \right] \left| \psi_r \right| \left| I_{qds} \right| \sin \theta \quad (2.35)$$

This is a general behavior of both rotor flux and torque for steady – state operation in any induction motor. Differences can happen in the quantities of the changes. This means that for a larger machine, the magnetising inductance will be larger and so the flux will experience a larger change than a small machine. However, for large machines (e.g. 100 – hp), the torque curve can be qualitatively different. This can happen when the ratio of actual to predicted time constant is between 0.4 and 1.0 ($0.4 < \tau_r / \tau_r^* < 1.0$) in which the torque will increase rather than decrease as for a small machine. Actually, this result is more typical of induction machines and displays that the slip for maximum torque with current excitation (maximum torque per ampere) is usually much smaller than the rated slip [Nordin, 1985]. This increase in torque is basically proportional to the flux increase. Nevertheless, at large errors of τ_r / τ_r^* , with values smaller than 0.4, the rotor impedance angle becomes sufficiently small and so the $\sin \theta$ term in equation (2.35) overcomes the flux increase. Hence, the torque starts to decrease again.

2.6.2.2 Saturation effects

The large increases in rotor flux shown in [Nordin, 1985], will lead to saturation of the main flux path in the machine. Saturation of the machine can be also viewed as reducing the value of the magnetising inductance and thus increasing the required magnetising current [Lorenz, 1990]. This means that the ratio I_{sq} / I_{sd} (corresponds to field orientation) will decrease and hence saturation will always reduce the sensitivity to detuning. In general, saturation is analogous to the temperature. Therefore, an increase of the temperature will cause an increase to the rotor resistance R_r , and this will lead to subsequent changes to (decrease) to the rotor time constant τ_r . Having said this, the saturation depends mainly on the physical characteristics of the machine used and its operating conditions.

2.6.3 Adaptation Effects

Methods of overcoming the problem of detuning effect, mainly due to the mismatch of the rotor time constant τ_r in the motor and IFOC have been a major research goal for a number of years. Numerous parameter adaptation schemes have been reported in the literature.

The on – line methods of rotor resistance identification can be categorized as:

- ❖ Spectral analysis techniques
- ❖ Observer based techniques
- ❖ Model reference adaptive systems based techniques (MRAS)
- ❖ Heuristic methods

2.6.3.1 Spectral analysis techniques

This technique is based on the measurement of parameters by injecting an external signal. In the case of spectral analysis, a disturbance signal is used because under

no load conditions of the induction motor, the motor parameters cannot be estimated as the rotor induced currents and voltages become zero and so slip frequency becomes also zero [Karanayil, 2005]. There are systems that the disturbance is provided by injecting negative sequence components [Matsuo, 1985], [Toliyat, 1993]. Another method proposed is the detection of misalignment between the actual motor flux and the rotor flux given by the model [Gabriel, 1982]. In the scheme proposed in [Sugimoto, 1987], a sinusoidal perturbation is injected into the flux producing stator current component. In general, this spectral analysis technique can estimate the value of the rotor resistance under any load and speed condition, however due to the installation of two search coils, the cost is high and will not be applicable to any off – the shelf induction motor.

2.6.3.2 Observer based techniques

The scheme proposed in [Lai, 1992] uses Extended Kalman Filter (EKF) to detect the inverse rotor time constant, by considering it as an extra state variable along with the stator and rotor currents. In this method, as perturbation is used the wide band harmonics contained in the PWM inverter output voltage. The disadvantage is the assumption that all other parameters are known and the magnetising inductance can cause large errors into the rotor time constant estimation.

2.6.3.3 Model reference adaptive system based techniques

This method of on – line rotor resistance identification is based on principles of model reference adaptive control. This approach is well – known by many researchers due to its relatively simple implementation requirements. The main idea is to calculate certain states from two different directions [Karanayil, 2005]. The first one should be calculated using states of the controller and the other is to estimate the same states using measured signals. One of these estimates should be independent of rotor resistance and then the error between them will provide the correction to the rotor resistance. This can be done using an adaptive mechanism such as a proportional – integral (PI) controller. The accuracy of these methods

depends mainly on the accuracy of the machine model used. The most frequently applied approach is the method which uses reactive power and is dependent on stator resistance [Rowan, 1991].

2.6.3.4 Heuristic methods

A new method for rotor resistance identification using a new coordinate axes selection has been proposed by Chan & Wand (1990). They set a new reference frame which was coincident with the stator current vector. They measured the steady – state stator voltage, current and motor speed and obtained the stationary reference frame components by using a three phase to two phase transformation. Finally, they derived an equation in which the rotor resistance was then calculated algebraically. This is a method that is valid only for steady – state operation of the motor.

Artificial intelligence such as artificial neural networks (ANN) and fuzzy logic (FL) were recently applied to the on – line rotor time constant / rotor resistance adaptation and there was a great interest of many researchers [Bose, 2002], [Karanayil, 2005]. Even though the results and the methods employed were elegant, the drawback of these methods is that none of them supported their modeling work by experimental data, and few of them had limited data file in the modeling.

2.7 Chapter summary

This chapter extensively introduced the Indirect Field Oriented control technique for cage – type induction motor drives. This class of drives is widely used in various industrial applications and their technology is increasingly developed.

The invention of vector control in the beginning of 1970s was a revolution in the high – performance control of ac drives as it could be controlled like a separately

excited dc motor. Because of that, vector control is also known as decoupling control.

The controller design of the dq current and speed control loop was implemented. For this, it was necessary to know the induction motor parameters and the estimation of them was initially based on the traditional methods.

Finally, the last section of this chapter was the study of parameter sensitivity in high – performance IFOC induction motor drives. This was achieved by analyzing the flux and torque deviations from the command value in a steady – state operating condition, both due to variations or mismatch in parameter values. In order to overcome the undesirable effects of parameter sensitivity, parameter adaptation is essential and methods of doing that was analytically explained.

In general, small low – efficiency machines are much less affected by detuning than large high – efficiency machines. It was proved that it is possible to operate low power drives without the need of parameter adaptation as small machines have low sensitivity to parameter detuning.

In the next chapter will be introduced Genetic Algorithms as one innovative parameter adaptation technique to overcome the effects of detuning in an IFOC system.

Chapter 3

IM Parameter Identification

3.1 Introduction to Optimisation – Heuristic Strategies

Nowadays computers are used to solve extraordinarily complex problems and are usually the perfect tool for optimisation as long as an engineer's or scientist's idea can be input in electronic format. So, the computer can be fed some data and get back the solution. Even though it seems that the idea will be realised, there is still a tough question to be answered. Is this the only solution? If not, is it the best one? Optimisation is the math tool that we rely on to get these answers [Haupt, 2004].

The definition “best” solution that it was referred above, means that in a complex problem there is more than one solution and these ones are not of equal value. Therefore, the terminology of “best” is relative and it depends of the problem at hand and its method of solution. Accordingly, the optimal solution depends on the person formulating the problem [Haupt, 2004].

As long as humankind exists, we endeavour for perfection and as a consequence for optimisation in many areas. As simple examples we can use the following questions. What is the best route to work? Which project do we confront first? We always want to reach a maximum degree of happiness with the least amount of effort. Hence, when designing something, the main goal of that is to minimise the cost or maximise the appeal of a product. Therefore, “*optimisation is the process of adjusting the inputs or characteristics of a device, mathematical process or experiment to find the minimum or maximum output or result*” (Figure 3.1). For this reason many researchers believe that optimisation is one of the oldest sciences which even extends into daily life [Neumaier, 2006], [Haupt, 2004]. From figure 3.1, it can be seen that as an input consists of variables, the function is known as the objective function or fitness function and the output is the cost or fitness. Since in this thesis the optimisation target is to minimise an error, optimisation becomes minimisation.

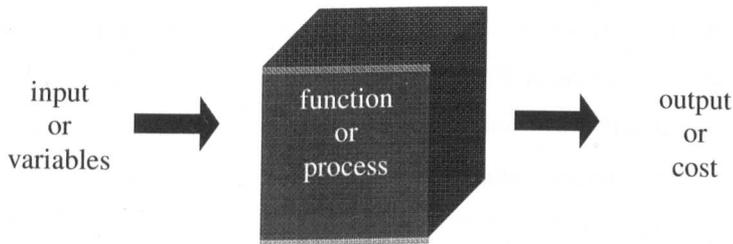


Figure 3.1: *Diagram of a function or process that is to be optimised. Optimisation varies the input to achieve a desired output.*

In general it can be said that if something is important and has to be optimised, there is always a mathematical discipline dealing with it. So, Global optimisation is the branch of applied mathematics and numerical analysis that focuses on to the optimisation of a function to a set of criteria [Weise, 2009]. As we referred above these criteria are expressed in a fitness function. Hence, the main aim of the global optimisation is to find such a set of inputs for which these fitness functions will return the optimal values [Passimo, 2006], [Okaeme, 2008]. The algorithms used to achieve these optimal values are termed Optimisation Algorithms.

Optimisation algorithms can be classified according to different optimisation techniques into two groups: deterministic and probabilistic algorithms.

A deterministic algorithm is an algorithm which behaves predictably and it is usually used if there is an obvious and not too complicated relation between a solution candidate and its “fitness”. Otherwise the problem is really hard to be solved deterministically and the same applies if the dimensionality of the search space is too high. Then, probabilistic algorithms come to employ a degree of randomness as part of its logic. In other words probabilistic algorithms violate the constraint of determinism. Deterministic algorithms guarantee that the results obtained using them are correct for a shorter runtime but they may not be the global optima. Finally, as far as these two optimisation algorithms are concerned, it is preferable to have a slightly inferior solution to the optimal one than one which takes many years to be found [Weise, 2009].

Heuristics used in global optimisation are functions that help decide which one of a set of possible solution candidate should be tested next or how the next individual can be produced. In other words heuristic is an algorithm that is able to produce an acceptable solution to a problem in many practical scenarios without any guarantee of its correctness. If there is no known method to find an optimal solution to a given problem, under the given constraints of time or space, then the best algorithm to be used is a heuristic algorithm [Weise, 2009], [Michalewicz, 2004], [Pearl, 1984].

3.1.1 What is an optimum?

It has been mentioned in the previous section that global optimisation is used to find the best possible solutions to a given problem. So, what makes a possible solution to be *optimal*? In the case of optimising an objective function f , an optimum is either its minimum or maximum, depending on what we are searching for. In mathematics, maxima and minima are the largest values (maximum) or smallest value (minimum) that a function takes in a point either within a given

neighbourhood (local extremum) or on the function domain in its entirety (global extremum). Figure 3.2 shows the local and global maxima and minima for a random function $f(x) = \cos(3\pi x)/x$ for $0.1 \leq x \leq 1.1$.

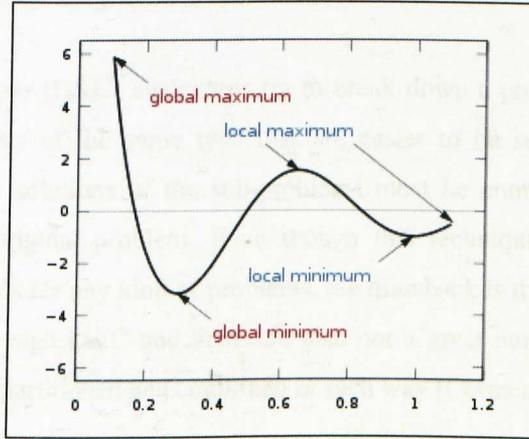


Figure 3.2: Local and global maxima and minima

Additionally, the exact meaning of optimal is problem dependent. In single – objective optimisation it either means minimum or maximum. In multi – objective optimisation minimum or maximum concepts are rather applied to sets F consisting of $n = |F|$ objective functions f_i , each representing one criterion to be optimised [Kalyanmoy, 2001].

$$F = \{ f_i : X \rightarrow Y_i : 0 < i \leq n, Y_i \subseteq \mathbb{R} \} \quad (3.1)$$

3.1.2 Optimisation Algorithms – Overview

To begin with, it has to be mentioned the main principles of traditional search algorithms. The simplest one is called *exhaustive search* as it tries all possible solutions from a predetermined set and then it chooses the best one.

Another well known traditional search algorithm is the *local search*. Local search is a version of exhaustive search but with limited search space. The technique is very simple. First an initial solution is randomly generated, and then the solutions in the neighbour of the current one are examined. When a better solution is found,

the current solution is immediately replaced with that one. If no solution that improves the current one is found then the current solution can be considered as a local optimal solution. Popular *hill – climbing* techniques belong to this class [Kokash, 2005], [Goldberg, 1989].

Divide and conquer (D&C) algorithms try to break down a problem into two or more sub-problems of the same type that are easier to be solved recursively. Followed by, the solutions of the sub-problems must be combinable to give a solution to the original problem. Even though this technique is the basis of efficient algorithms for any kind of problems, the drawback is that it takes time to understand and design D&C and there are also not a great number of problems that can be easily partitioned and combined in such way [Cormen, 2000].

Branch – and – bound is another technique that is implemented for finding the optimal solutions of various optimisation problems. It consists of a critical enumeration of all candidate solutions by using partition, sampling, and subsequent lower and upper bounding procedures [Kokash, 2005], [Goldberg, 1989].

Dynamic programming is an exhaustive search method for efficiently solving a wide range of search and optimisation problems which display the characteristics of overlapping sub problems (e.g., broken down into sub-problems which are reused multiple times) and optimal substructure (e.g., globally optimal solution can be constructed from locally optimal solutions to sub-problems). This is closely related to recursion [Bertsekas, 2000], [Goldberg, 1989].

Last but not least, another traditional algorithm that is perhaps the most straightforward and powerful one from those mentioned is called greedy algorithm. The idea is that always takes the best (local) solution while finding the global optimum for some optimisation problems, hoping that this choice will lead to a global optimum solution [Cormen, 2000], [Goldberg, 1989].

So, the most important differences between Evolutionary Algorithms (EAs) and traditional search and optimisation methods are:

- EAs, search a population of candidate solutions and not a single point.
- EAs don't require derivative information or other auxiliary knowledge (domain knowledge); only an objective function and the corresponding fitness levels influence the directions of search.
- EAs are stochastic methods, i.e., use probabilistic transition rules in comparison with the traditional methods that use deterministic ones.
- EAs work on an encoding of the parameter set rather than the parameter set itself (except in where real – valued individuals are used).
- Applies to a variety of problems and doesn't work in a restricted domain.
- Multiple solutions can be obtained without extra effort.
- EAs can be implemented on parallel machines.
- EAs are quite successful in locating the regions containing optimal solution(s), if not the optimum solution itself.

Evolutionary algorithms provides a number of potential solutions to a given problem and every time the user has to choose according to the desirable result that he is expecting, the final solution of the problem. On the assumption that a particular problem does not have one only solution, such as in multiobjective optimization and scheduling problems, then the EA is potentially useful for identifying these alternative solutions simultaneously [Chipperfield, 1994a], [Chipperfield, 1994b], [Mitchell, 1996].

Figures 3.3 and 3.4 shows the classification of optimisation algorithms discussed in section 3.1 enumerating a wide variety of optimisation algorithms. The most important and best known of them are the Evolutionary Algorithms (EAs) which are population – based metaheuristic optimisation algorithms that use biology – inspired mechanisms like mutation, crossover, natural selection and survival of the fittest in order to refine a set of solution candidates iteratively. [Back, 1996]. They belong to a class of evolutionary computation (EC) methods [Weise, 2009], [Hussain, 1997] which in their turn they are also an essential class of probabilistic

algorithms. The following describes the more popular evolutionary computation techniques: genetic algorithms (GA), evolutionary programming (EP), evolution strategies (ES), and genetic programming (GP).

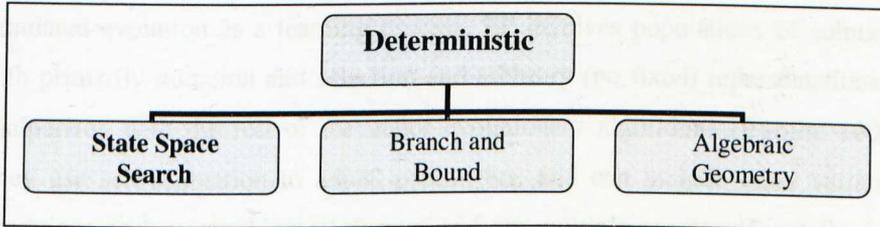


Figure 3.3: Classification of Deterministic Optimisation Algorithms.

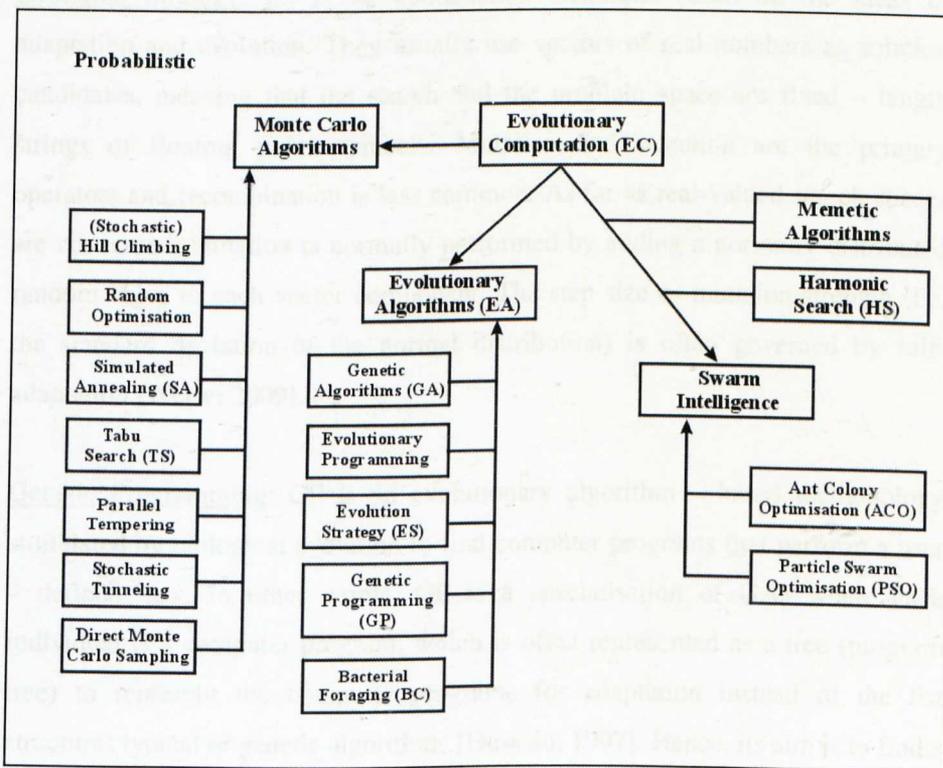


Figure 3.4: Classification of Probabilistic Optimisation Algorithms

Genetic Algorithms: GAs are a particular class of evolutionary algorithms which are implemented in a computer simulation in which a population of abstract representations (called chromosomes) of candidate solutions (called individuals)

to an optimisation problem evolves toward better solutions. More details in this evolutionary approach is provided in section 3.3.

Evolutionary Programming: EP was first used by Fogel in 1960 in order to use simulated evolution as a learning process. EP involves populations of solutions with primarily mutation and selection and arbitrary (no fixed) representations in comparison with the rest of the major evolutionary algorithms [Burgin, 2007]. They use self-adaptation to adjust parameters, and can include other variation operations such as combining information from multiple parents. Generally, it is hard to distinguish EP from GAs, ES, and GP.

Evolution Strategy: ES is an optimisation technique based on the ideas of adaptation and evolution. They usually use vectors of real numbers as solution candidates, meaning that the search and the problem space are fixed – length strings of floating point numbers. Mutation and selection are the primary operators and recombination is less common. As far as real-valued search spaces are concerned, mutation is normally performed by adding a normally distributed random value to each vector component. The step size or mutation strength (i.e. the standard deviation of the normal distribution) is often governed by self-adaptation [Weise, 2009].

Genetic Programming: GP is an evolutionary algorithm – based methodology stimulated by biological evolution to find computer programs that perform a user – defined task. In other words, GP is a specialisation of GAs where each individual is a computer program, which is often represented as a tree (program tree) to represent the computer programs for adaptation instead of the list structures typical of genetic algorithms [Hussain, 1997]. Hence, its aim is to find a global optimal solution by breeding populations of computer programs in terms evolution and principle of natural selection [Walker, 2001],[Okaeme, 2008].

So why is this variety (of optimisation algorithms) needed? One reasonable answer would be that there are so many different kinds of optimisation tasks presenting diverse characteristics and so there is a need for different approaches.

3.1.3 Premature Convergence

One of the most common problems that may be encountered during optimisation is *premature convergence*. An optimisation process has *prematurely converged* to a local optimum if it is no longer able to explore other parts of the search space than the area currently being examined and there exists another region that contains a superior solution [Ursem, 2003], [Schaffer, 1990]. Figure 3.5 shows examples for premature convergence. One cause of that is loss of diversity. This means that there is a state (convergence) where all the solution candidates are similar to each other under investigation. Maintenance of diversity is associated with sustaining a good balance between exploration and exploitation [Weise, 2009]. Exploration is used to investigate new and unknown areas in the search space, and exploitation to make use of knowledge found at points previously visited to help find better points [Holland, 1975], [Whitely, 1993]. More details on how we can avoid the problem of premature convergence will be discussed in a later section of this chapter.

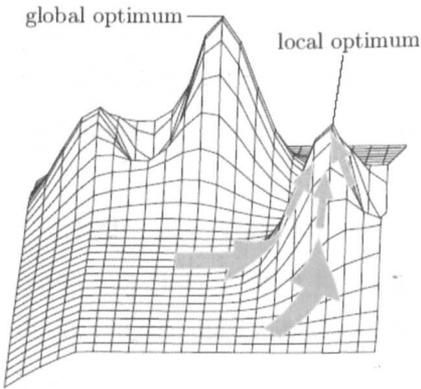


Figure 3.5.a: Maximisation

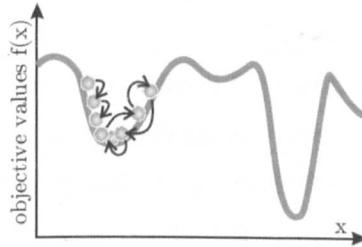


Figure 3.5.b: Minimisation

Figure 3.5: Premature convergence in the objective space

3.2 Why Evolutionary Algorithms?

Evolutionary computation is a recognised field from many researchers and it is very attractive because of the many successful application and also the huge number of publications in this field so far.

Evolution can be seen as an optimisation process as for any particular problem and it has the ability to adapt well to its environment due to the most recent feedback from it, hence [Back, 1997]: *“it is quite natural, therefore, to seek to describe evolution in terms of an algorithm that can be used to solve difficult engineering optimisation problems. The classic techniques of gradient descent, deterministic hill – climbing, and purely random search (with no heredity) have been generally unsatisfactory when applied to nonlinear optimisation problems, especially those with stochastic, temporal, or chaotic components. But these are the problems that nature has seemingly solved so very well. Evolution provides inspiration for computing the solutions to problems that have previously appeared intractable.”* In other words, the above quote place emphasis to the advantages of evolutionary computation methods concerning with problems within the field of science and engineering. However it can be difficult if not impossible to offer hard facts or evidence about why evolutionary algorithms are better than classical optimisation methods. In the next paragraphs will be analytically discussed why evolutionary algorithms are the most used optimisation techniques.

Standard – deterministic or classical methods are characterised by a gradient – based (e.g., non – linear least squared method), or approximation – based (e.g., direct and statistical methods). These optimisation techniques are more efficient in solving linear, quadratic, and other special problems [Back, 1997]. Hence, all classical methods find only one individual solution in a single run.

However most of the real – problems, have unconventional response surfaces such as discontinuous, non differentiable, noisy nature of the problem, uncertainty within the problem domain, etc. In this case, a logical approach would be an exhaustive (point – to – point) search of the solutions domain [Okaeme, 2008]. This can be efficiently done by applying Evolutionary algorithms (EA). As EA

are stochastic optimisation methods population – based stimulated by natural selection, they are capable to find various solutions in a single run. Hence, EA are suited to such extreme environments, highlighting one of the advantages they do have to standard – deterministic methods for real – world problems.

Another advantage showing that EAs are more efficient than standard – deterministic methods is the size of the search space. So, for most real – world problems the size of the search space of solution domain is usually too large to consider exhaustive search. In order this to be more understandable an example will be given, representing the following problem by Michalewicz [1997]:

$$\text{optimise } f(x_1, x_2, \dots, x_{100})$$

where f is very complex and x_i is 0 or 1. The size of the search space is $2^{100} \approx 10^{30}$. The importance of the size can be understood, considering the case when 1000 potential solutions have to be examined within one second. In the case of the previous search space, this will require more than 10^{19} years of calculations. Therefore, for such large search spaces, modern heuristic search methods (above all evolutionary computation) are of great importance.

Without any doubt, in order to solve a problem, it is quite important to understand and characterise the behaviour of it. For example, in control design optimisation, the first step before searching the controller parameters for a system, would be to model the system to be controlled. As a result, if the model is known, then the final controller for the system can be achieved accurately [Okaeme, 2008]. Hence, there are two approaches to derive a model for a real – world non - linear system:

- linearization (or simplification).
- a more complex (faithful) model.

Through the first procedure there are some relatively simple methods based on simple optimisation techniques and often not even meet up optimisation with which the global optimum can be obtained. When using a more complex and detailed model, optimal solution can be found by adopting EA strategy while standard – deterministic techniques often fails. However, most of the experimental

evidence [Schwefel, 1997] indicate that a globally optimal solution for a simplified model may be of a largely inferior quality to that of an approximate optimal solution for a real problem. This is a strong argument about the usefulness of evolutionary algorithms, which are an ideal tool in the scientist's hand as they are perfect in finding approximated near – optimum solutions.

During the initialisation for the EA, it is required to set a number of parameters such as the maximum number of generation, the population size, the values of mutation and crossover rates (they will be defined and discussed in a later section) etc. However, many researchers from time to time criticise the EA techniques, and their argument is that EA will fail to find the optimal solution(s) of a problem if the above mentioned number of parameters are wrong. But, EA are known about their robustness, meaning that even wrongly defined parameters could still give reasonably good results [Michalewicz, 1997]. Besides that, one of the most important areas of research towards EAs is the possibility of designing a self – adaptation algorithm of some internal parameters as it will tune the algorithm to the problem while solving the problem.

Lastly, another application that places EAs in the first choice of an optimisation problem is the temporal optimisation which is applied in many real – world problems. As the real world is always changing, these kinds of problems are more challenging and then the role of temporal optimisation is to seek the best behavioural strategy considering the most updated feedback regarding the performance of the current strategy. EA are efficiently adapted to changes in the problem domain and this is the reason that they are fitted so well in such applications. On the other hand, the disadvantage of the standard – deterministic approaches in such cases (difficulty in adapting to the changing instantaneous conditions) requires from the system to restart the procedure [Michalewicz, 1997], [Schwefel, 1997].

In this research work, Genetic Algorithm was used among EAs, as according to Davis [Davis, 1989]: *“It has seemed true to me for some time that we cannot handle most real – world problems with binary representations and an operator set consisting only of binary crossover and binary mutation. One reason for this is*

that nearly every real – world domain has associated domain knowledge that is of use when one is considering a transformation of a solution in the domain. I believe that genetic algorithms are the appropriate algorithms to use in a great many real – world applications. I also believe that one should incorporate real – world knowledge in one’s algorithm by adding it to one’s decoder or by expanding one’s operator set”. These hybrid/non – standard systems are very popular in the evolutionary computation community. In the next sections, there is a fully description of Genetic Algorithms and their usefulness towards other traditional search algorithms.

3.3 The Genetic Algorithm

Genetic algorithms (GAs) are stochastic global search techniques used in computer science to find approximate solutions to optimisation and search problems. A genetic or evolutionary algorithm *applies the principles of evolution found in nature* to the problem of finding an optimal solution to a Solver problem. In a genetic algorithm, the problem is encoded in a series of “genes” (e.g., bit strings) that are manipulated by the algorithm. These bit strings are coded representations of input variables. The basic concept of GAs is designed to simulate processes in natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin of survival of the fittest [Chipperfield, 1994a], [Chipperfield, 1994b], [Mitchell, 1996], [Goldberg, 1989].

GAs were invented by John Holland in the 1960s and were developed by Holland and his students and colleagues at the University of Michigan in the 1960s and the 1970s. GAs have also been widely studied, experimented and applied in many fields in the engineering world. In the next sections it will be given a detailed introduction to Genetic Algorithms and their applications in system identification problems will be presented.

3.3.1 Genetic Algorithms Procedure

As it was referred earlier, generally, Genetic Algorithms operate on a population of potential solutions applying the principle of survival of the fittest to produce better and better approximations to a solution. At each generation, a new set of approximations is created by the process of selecting individuals (each possible solution is called an individual) according to their level of fitness in the problem domain and breeding them together using operators borrowed from natural genetics. This process leads to the evolution of populations of individuals that are better suited to their environment than the individuals that they were created from, just as in natural adaptation.

A typical genetic algorithm requires two main things to be defined:

- A genetic representation of solutions
- A fitness function to evaluate them

The standard representation is an array of bits. For the reason that these genetic representations are easily changed due to their fixed size, it makes them to be very convenient and also the crossover operation is very simple. The only case that the crossover implementation is more complex is that when variable length representations were used.

On the other hand, the fitness function is defined over the genetic representation and measures the quality of the represented solution. The fitness function is always problem dependent. An ideal fitness function is connecting closely with the algorithm's goal and yet may be evaluated quickly because as a usual genetic algorithm must be repeated many times in order to achieve the desirable result and for a problem the speed of execution is very important. In some cases, it is hard to define the fitness function, so it is used an interactive genetic algorithm that uses human evaluation.

If both of the genetic representation and the fitness function are defined, GA proceeds to initialize a population of solutions randomly, and then improve it

through repetitive application of mutation, crossover and selection operators [Goldberg,1989], [Michalewicz, 1999], [Mitchell, 1996].

Figure 3.6 shows the main loop of genetic algorithms. The first step is to generate the initial population, and then to evaluate the fitness value using fitness or objective function. The next step is to perform competitive selection and apply the genetic operators in order to generate new solutions. Finally, evaluate the solutions into the population and start again the procedure from the performance of competitive selection and repeat until some convergence criteria are satisfied.

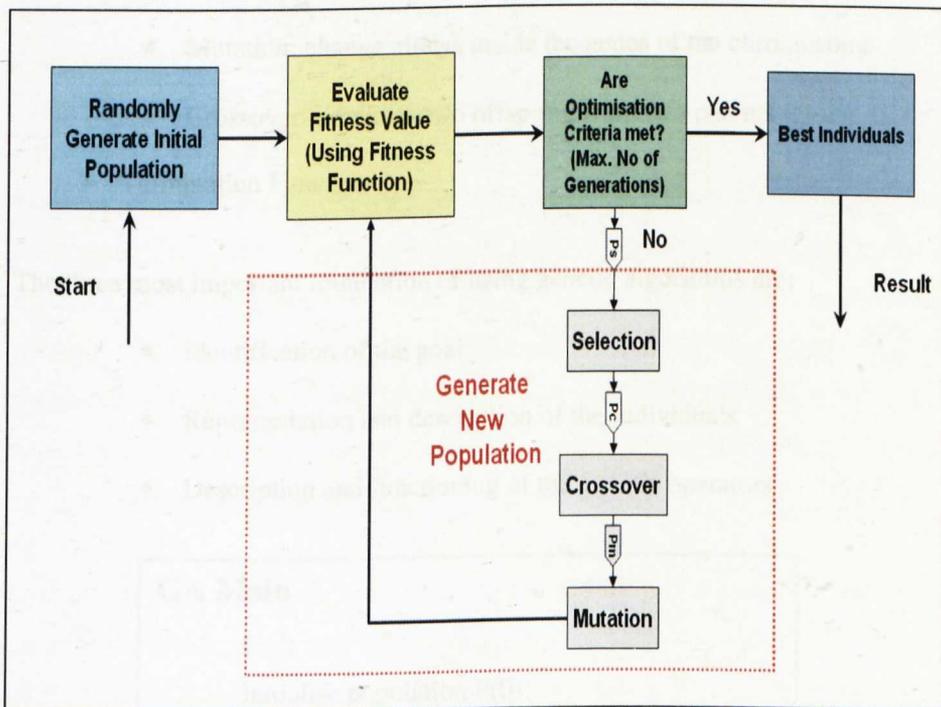


Figure 3.6: *The basic cycle of genetic algorithms*

3.3.2 Fundamental Elements of the Genetic Algorithm

Goldberg (1989) described the simple genetic algorithm (SGA) that involves nothing more complex than copying strings and swapping partial strings. When using the SGA there are some major issues to take into consideration. These major elements are:

- Representation of the individuals
- Initialisation function
- Selection methods:
 - Roulette wheel selection
 - Tournament selection
 - Rank selection
- Fitness / Evaluation functions
- Genetic operators
 - Mutation: change alleles inside the genes of the chromosome
 - Crossover: produces two offspring from two parents
- Termination Function

The three most important foundation of using genetic algorithms are:

- Identification of the goal
- Representation and description of the individuals
- Description and functioning of the genetic operators

```
GA Main  
t = 0  
initialise population P(0)  
evaluate population P(0)  
while(!(termination condition)) {  
    t = t+1  
    select population P'(t) from P(t - 1)  
    create population P(t) from P'(t)  
    evaluate population P(t)  
}
```

Figure 3.7: A Simple Genetic Algorithm

Concerning the implementation of GAs there is a variety in population structures and genetic operators. However GAs have an initialisation phase followed by a repetitious phase. Figure 3.7 illustrates the pseudo code of a simple Genetic Algorithm.

3.3.2.1 Population Representation and Initialisation

As mentioned previously genetic algorithms operate on a number of potential solutions, called individuals, consisting of some encoding of the parameter set simultaneously. The individuals are pumped into a population. Usually, a population is composed of between 30 and 100 individuals, but it can be large in size according to the problem [Chipperfield, 1995].

The presentation scheme determines how the problem is structured in the GA and also determines the genetic operators used. The most frequently used representation of chromosomes in the GA is the ones of a single – level binary string that linked together to form an individual. The use of Gray coding (which is a binary numeral system where two successive values differ in only one bit) has been recommended as a method to defeat the hidden representational bias in conventional binary representation.

Despite the fact that binary – coded GAs are most usually used, there is an countable amount of people that are interested in dissimilar encoding strategies such as integer and real – valued representations, above all to solve engineering design problems.

According to Wright (1991) the use of real – valued genes in GAs provides several advantages in numerical function optimisation over binary encodings. Considering that it is not necessary to convert chromosomes to phenotypes (the decision variables, or phenotypes, in the GA are obtained by applying some mapping from the chromosome representation into the decision variable space) before each function evaluation, genetic algorithms becomes more efficient because less memory is required, there is no loss in precision and there is greater

freedom to use different genetic operators. Figure 3.8 demonstrates an example of individual representation.

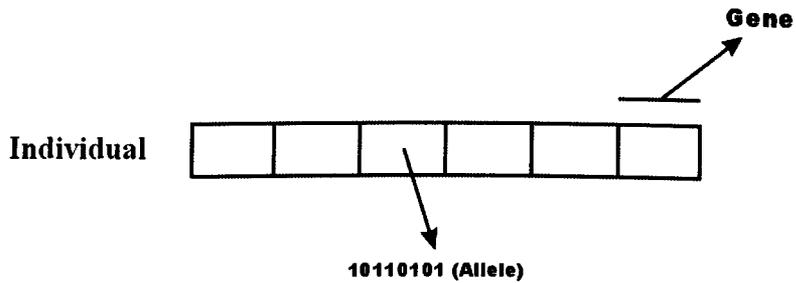


Figure 3.8: Example of individual representation

The starting point of the search is to create a group of individuals to form a population. This is called the initial population and can be created in a number of ways according to figure 3.9. The most common setup begins by generating the required number of individuals using a random number generator that uniformly distributes numbers in the desired range. As an example, if there is a binary population of $Nind$ individuals whose chromosomes are $Lind$ bits long, then $Nind \times Lind$ random numbers uniformly distributed from the set $\{0, 1\}$ would be produced. The main aim is to create a population with a good coverage of the search space and as a consequence having a gene pool with good potential for breeding better solutions. Otherwise, genomes can be spread out to the whole search space according to a regular grid-layout. Nevertheless, an entirely new selection of starting points can be the advantage to a random setup while runs are repeated. Another approach is that experts typically can estimate what a reasonable solution should be to a specific problem. By inserting this solution as one of the starting individuals, then the remaining individuals could be randomly arranged in a grid close to the best known solution. So, a problem with such an initialisation can achieve to have a search area near the special (best) solution.

In conclusion, the choice of initialisation methods depends on the problem itself and the approach it is going to be used. However, for real – world applications, experts' knowledge plays an important role in initialisation as in some circumstances there is the possibility to specify the initial search space positions

based on specific knowledge about the objective function [Chipperfield, 1995], [Ursem, 2003].

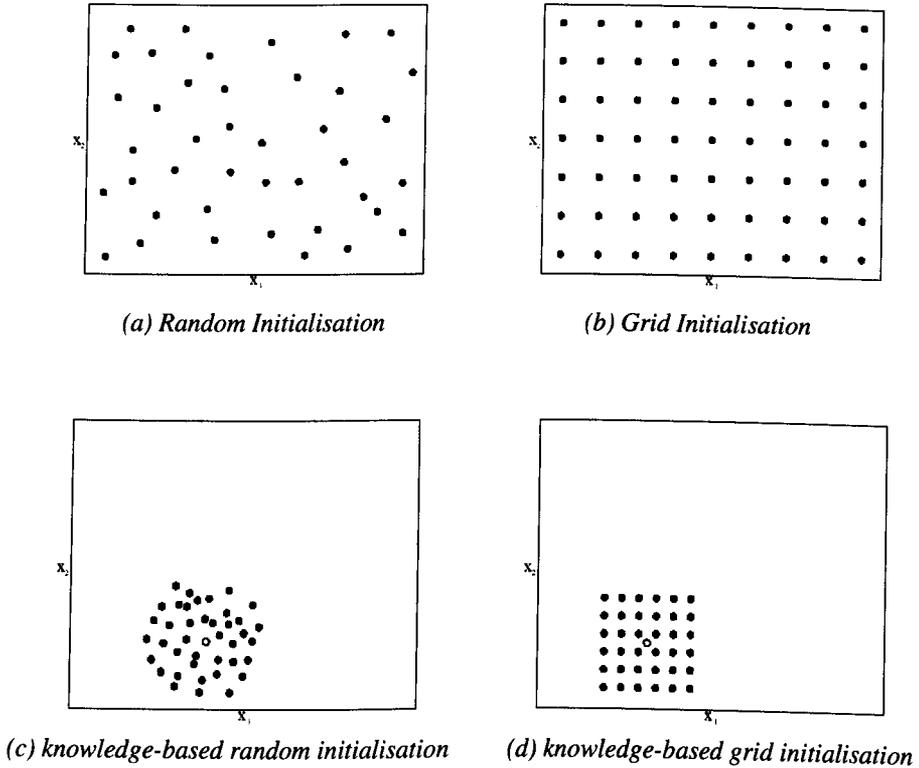


Figure 3.9: Examples of initialisation methods.

3.3.2.2 The Objective and Fitness Function

Once a population of solutions is created, each of the chromosomes in the population must be evaluated to see how well they solve the problem at hand. This is achieved with the objective function which decodes the chromosome, evaluates it and returns the performance to the genetic algorithm.

In case that the optimisation method has to be used as a minimisation problem, the fittest individuals will have the lowest numerical value of the associated objective function. This numerical value is used to figure out the relative performance of individuals in a GA. In order to transform the objective function value into a

measure of relative fitness, another function, called *fitness function*, is used [De Jong, 1975], so that:

$$F(x) = g(f(x)) \quad (3.2)$$

where f is the objective function, g transforms the value of the objective function to a non negative number and F is the resulting relative fitness. There are a lot of cases where the fitness function value is set in a way that corresponds to the number of offspring that an individual can expect to produce in the next generation. In this case, another transformation is usually used (see below) in which the individual fitness, $F(x_i)$, of each individual is calculated as the individual's raw performance, $f(x_i)$, relative to the whole population:

$$F(x_i) = \frac{f(x_i)}{\sum_{i=1}^{Nind} f(x_i)} \quad (3.3)$$

where $Nind$ is the population size and x_i is the phenotypic value of individual i . However one of the drawbacks of this fitness function is that it fails to account for negative objective function values because it make sure that each individual has a probability of reproducing according to its relative fitness.

Then, a linear transformation which offsets the objective function is often used to ensure that the resulting fitness values are non negative. The function used is illustrated below:

$$F(x) = a \cdot f(x) + b \quad (3.4)$$

where a is a positive factor if the optimisation has to be maximised and negative if it has to be minimised. The offset b is the one that compensates to the non negative results.

Figure 3.10 shows how the fitness of the best individual improves over time and the fitness value approaches gradually the zero point towards the end of the run.

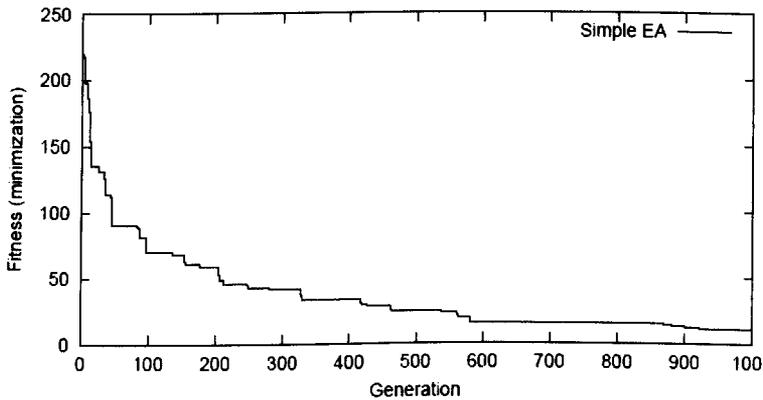


Figure 3.10: Gradual fitness improvement during the run (minimisation)

3.3.2.3 Selection

When the evaluation procedure of all of the possible solutions has been completed, two or more must be selected to be parents and to create offspring for the next generation. The selection process is usually a random process of determining the number of times, or trials, a particular individual is chosen for reproduction and consequently the number of offspring that an individual will produce. The main goal is to create a new generation of individuals that are potentially better solutions than their parents. In the selection, a string with a high fitness value has more chances to be selected as one of the parents than a string with a low fitness value. The selection of individuals can be analysed as two separate processes:

- Determination of the number of times an individual can expect to be chosen, and
- Conversion of the expected number of times an individual expects to be chosen into a discrete number of offspring.

The selection process is related with the fitness assignment of each individual. In other words, each individual is examined and evaluated using the fitness function and the output value will quantify the fitness of each individual. There are a number of ways to implement the selection. Some of the most popular and well – studied of them are:

- Roulette Wheel Selection Method
- Tournament Selection
- Stochastic Universal Sampling

3.3.2.3.1 Roulette Wheel Selection Method

Roulette wheel selection method is a genetic operator used in genetic algorithms for selecting potentially useful solutions for recombination based on some measure of their performance. In this method the fitness level is used to associate a probability of selection with each individual chromosome [Hussain, 1997].

The analogy to a roulette wheel can be predicted by imagining a roulette wheel in which each candidate solution represents a part on the wheel. The size of the parts represents an individual's fitness relative to the other individuals and is proportionate to the probability of selection of the solution. For example, selecting N chromosomes from the population is equivalent to playing N games on the roulette wheel, as each candidate is drawn independently [Abraham, 2005].

It is obviously that larger areas of the wheel will have higher fitness value and a better chance to be selected. Figure 3.11 shows the perimeter of the roulette wheel that is the sum of all six individual's fitness values.

From the below figure 3.11, it can be seen that the individual 5 is the fittest individual and having the largest part, it is obvious that it will have a better chance to be selected more than once, while individuals 4 and 6 are the least fit and have correspondingly a smaller part within the roulette wheel. This method is repeated

as many times as required until the desired numbers of individuals have been selected (called mating population) [Chipperfield, 1994a].

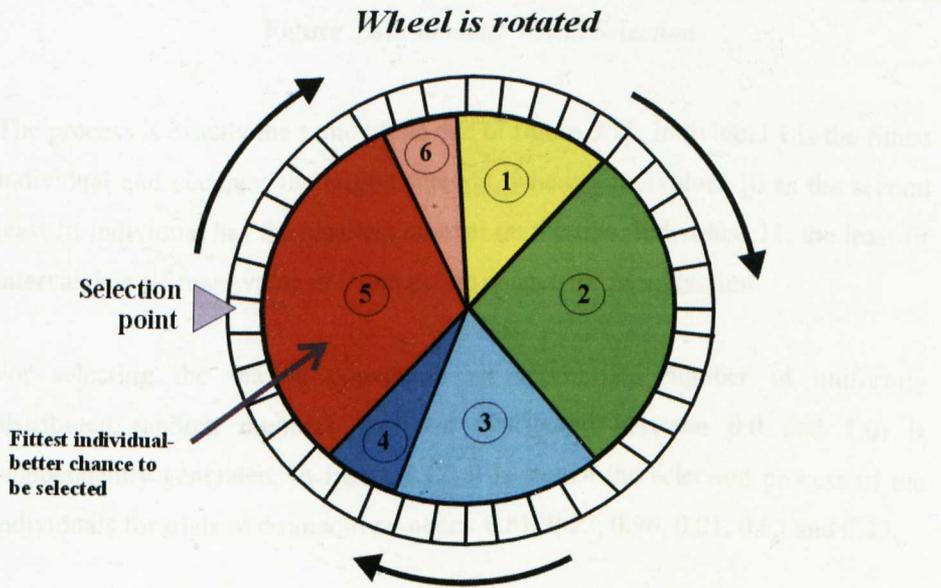


Figure 3.11: Roulette Wheel Selection

Another graphic example of the roulette wheel selection process is illustrated in figure 3.12 with the individuals of Table 1, which shows the selection probability for 11 individuals, linear ranking and selective pressure together with the fitness value.

Number of individuals	1	2	3	4	5	6	7	8	9	10	11
Fitness value	2.0	1.8	1.6	1.4	1.2	1.0	0.8	0.6	0.4	0.2	0.0
Selection probability	0.18	0.16	0.15	0.13	0.11	0.09	0.07	0.06	0.03	0.02	0.0

Table 3.1: Selection probability and fitness value

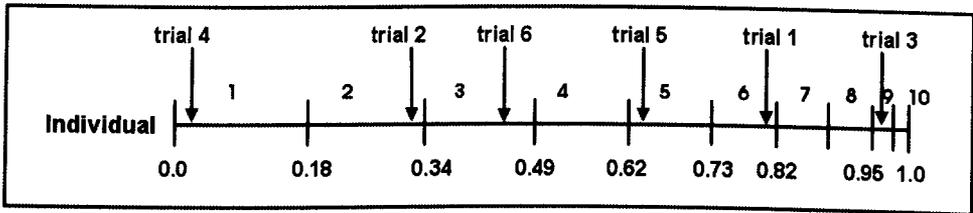


Figure 3.12: Roulette Wheel Selection

The process is exactly the same of the one of figure 3.11. Individual 1 is the fittest individual and occupies the largest interval, whereas individual 10 as the second least fit individual has the smallest interval on the line. Individual 11, the least fit interval, has a fitness value of 0 and get no chance for reproduction.

For selecting the mating population an appropriate number of uniformly distributed random numbers (uniform distributed between 0.0 and 1.0) is independently generated. In figure 3.12, it is shown the selection process of the individuals for trials of 6 random numbers: 0.81, 0.32, 0.96, 0.01, 0.65 and 0.42.

The roulette – wheel selection algorithm provides a zero bias (bias is the absolute difference between an individual's actual and expected selection probability) but does not guarantee minimum spread (spread is the range in the possible number of trials that an individual may achieve).

3.3.2.3.2 Tournament Selection

Tournament selection is one of the most popular and effective selection mechanisms commonly used by genetic algorithms. Tournament selection selects a group of k_{ts} individuals randomly from the population, where $k_{ts} < k_s$ (k_s is the total number of individuals in the population). The performance of the selected k_{ts} individuals is compared and the best individual from this group is selected and returned by the operator. For crossover with two parents, tournament selection is done twice, once for the selection of each parent [Engelbrecht, 2007]. If $k_{ts} = k_s$, the best individuals will always be selected, resulting in a very high selective

pressure (see below paragraph). On the other hand, if $k_s = 1$, random selection is obtained.

Tournament selection is increasingly being used as it satisfies the criteria of being an ideal selection method. These ideal criteria would be simple to code and efficient for both non – parallel and parallel architectures (parallel architecture is a form of computation in which many calculations are carried out simultaneously, operating on the principle that large problems can often be divided into smaller ones, which are then solved “in parallel”). In addition, tournament selection can also adjust the selection pressure to adapt to different domain and so for example the tournament size can be increased by increasing the tournament selection pressure. The higher tournament size, the more the better individuals are favoured to be selected and vice versa [Miller, 1995].

Not only does the selection pressure improve the population fitness of the GAs, but it also plays an important role regarding its convergence rate. More detailed, higher selection pressure results in higher convergence rates. Genetic Algorithms are able to identify optimal solutions under a wide range of selection pressure [Goldberg, 1993]. However, if the selection pressure is too low, the convergence rate will be very slow, and GAs will need more time than normal to find the optimal solution. On the other hand, if the selection pressure is too high, the convergence rate will be incredibly fast and then there is more chances to have a prematurely convergence that will lead to an inaccurate solution.

To conclude, the winner of the tournament will be the individual with the highest fitness of the entire participants and it will be inserted into the mating pool. The mating pool, being full of tournament winners has a higher average fitness than the average population fitness. This fitness difference provides the selection pressure, which forces the GA to improve the fitness of each succeeding generation [Miller, 1995].

3.3.2.3 Stochastic Universal Sampling

Stochastic universal sampling (SUS) is a development of roulette – wheel selection with minimum spread and zero bias. The individuals are mapped to adjacent part of a line, such that each individual's part is equal in size to its fitness exactly as in roulette – wheel selection. In this case, depending on the number of individuals to be selected, the same number of equally spaced pointers is placed over the line. Consider N -Pointer the number of individuals to be selected, then the distance between the pointers is $1/N$ -Pointer and the position of the first pointer is given by a randomly generated number in the range $[0, 1/N$ Pointer].

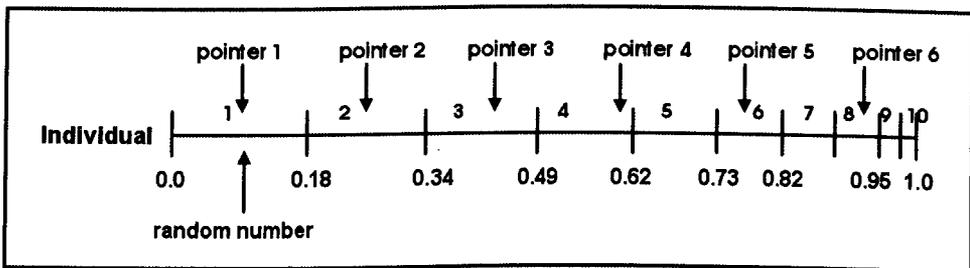


Figure 3.13: Stochastic universal sampling

Figure 3.13 shows the selection of the choice of 6 individuals where the distance between the pointers is $1/6 = 0.167$. Sample of 1 random number in the range $[0, 0.167]$: 0.1.

After selection, the mating population consists of the individuals: 1, 2, 3, 4, 6 and 8. Stochastic universal sampling ensures a selection of offspring which is closer to what is deserved than roulette wheel selection [Chipperfield, 1994a].

3.3.2.4 Genetic Operators

The genetic operators are separated in two categories:

- ❖ Crossover (or recombination)
- ❖ Mutation

Mutation is usually used in GAs to generate diversity while crossover to combine existing solutions into others. The main difference between them is that mutation operates on one chromosome while crossover on two different chromosomes.

3.3.2.4.1 Crossover (Recombination)

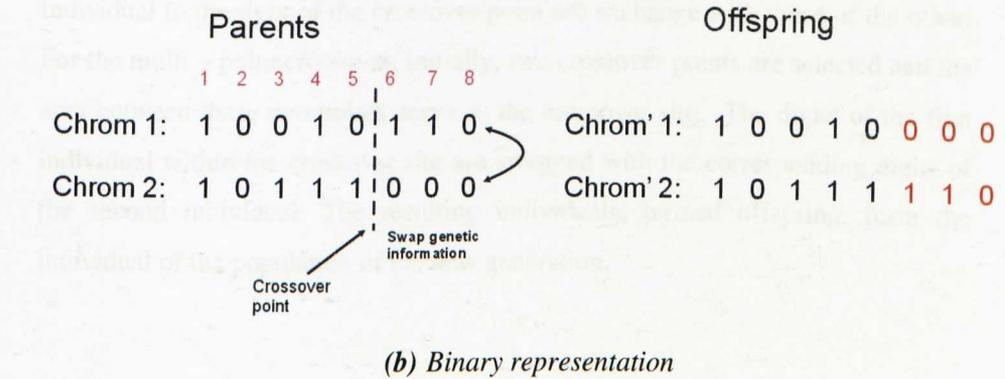
Crossover is one of the basic operators for reproducing new chromosomes in the genetic algorithm. Crossover produces new individuals that have some characteristics of parent strings. The recombination operator is used to exchange genetic information between pairs of individuals and produce offspring for the next generation. Two strings are selected randomly from the mating pool. The crossover probability p_c will determine if crossover should take place and how often, within any generation, the crossover function is carried out on pairs of individuals. This value is usually chosen to be in the range 0.5 – 1.0 [Srinivas, 1994]. The simplest recombination operator is the single – point crossover, although there are some other variations of crossover such as multi – point crossover, uniform crossover, intermediate recombination and line recombination. The difference between them is the generated crossover point. In this section the simple – point crossover will be described.

Consider the two parent binary strings:

$$P_1 = 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0, \text{ and}$$

$$P_2 = 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0.$$

As referred above, if crossover does take place, then two new offspring strings are produced. An integer point, i , is selected randomly between 1 and the string length, l , minus one $[1, l-1]$, hence, the genetic information is exchanged between the individuals about this point, i . The two offspring from the below figure 3.14a are produced when the crossover point $i=5$ is selected:



individual to the right of the crossover point are exchange with those of the other. For the multi – point crossover, initially, two crossover points are selected and the area between these two points serve as the crossover site. The digits of the first individual within the crossover site are swapped with the corresponding digits of the second individual. The resulting individuals, termed offspring, form the individual of the population of the new generation.

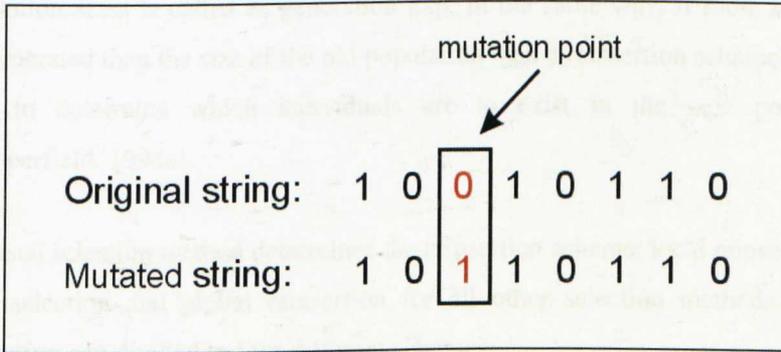
3.3.2.4.2 Mutation

A further genetic operator, called mutation is a random process where one allele of a gene is replaced by another to produce a new genetic structure and is applied to the new chromosomes with a set probability, p_m . The mutation operation does not occur as frequently as the crossover function and it is applied generally by using a low probability, usually in the range 0.005 – 0.05 [Srinivas, 1994]. As a result of this, the role of mutation is often seen as providing a guarantee that the probability of searching any given string will never be zero and acting as a safety net to recover good genetic material that may be lost through the action of selection and crossover [Goldberg, 1989]. Mutation causes the individual genetic representation to be changed according to some probabilistic rule. So, if l is the length of the chromosome then a number between 1 and l is selected randomly as the mutation point. In the binary string representation, mutation will cause a single bit to change its state, $0 \Rightarrow 1$ or $1 \Rightarrow 0$. The binary mutation process is demonstrated in Figure 3.15a.

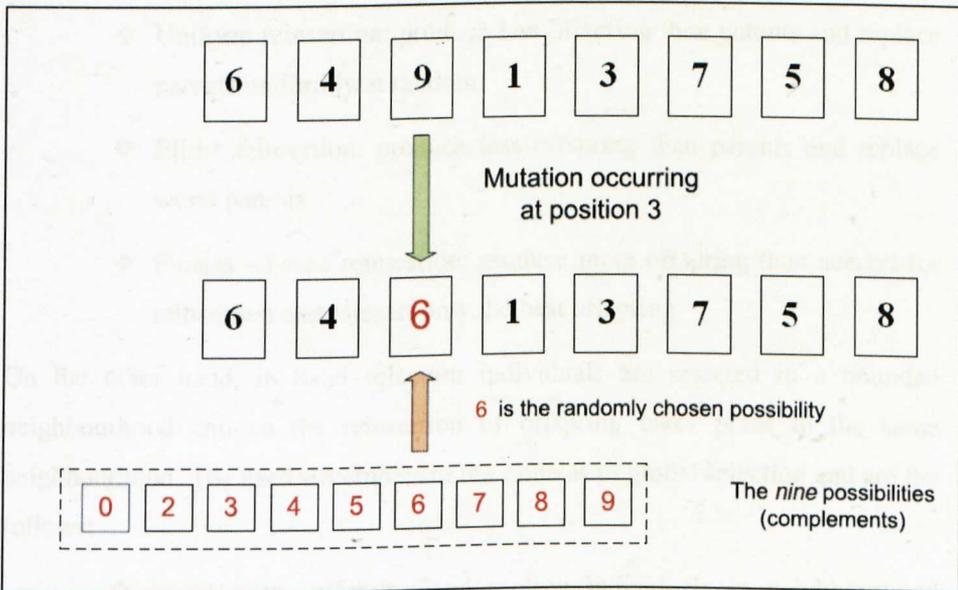
As it can be seen the mutation point is on bit three of the individual and so the binary mutation flips the value of the bit. Given that mutation is generally applied uniformly to an entire population of strings, it is possible that a given binary string may be mutated at more than one point.

Figure 3.15b illustrates the mutation for real valued representation of individuals. The application of mutation in this case includes randomly choosing a position to perform the operation and then changing the figure in that position to any of its

complementary values. For the decimal population, any figure (in any position) would have nine complementary values.



(a) Binary representation



(b) Real-valued representation

Figure 3.15: Mutation operator

3.3.2.5 Reinsertion

Once the offspring have been produced by selection, recombination and mutation of individuals from the old population, the fitness of the offspring may be

determined. In case that fewer offspring are produced than the size of the original population then the offspring have to be reinserted into the old population in order to keep the size of the original one. The difference between the new and old population sizes is called as generation gap. In the same way, if more offspring are generated than the size of the old population then a reinsertion scheme must be used to determine which individuals are to exist in the new population [Chipperfield, 1994a].

The used selection method determines the reinsertion scheme: local reinsertion for local selection and global reinsertion for all other selection methods. Global reinsertion can be divided in four different schemes:

- ❖ Pure reinsertion: produce as many offspring as parents and replace all parents by the offspring
- ❖ Uniform reinsertion: produce less offspring than parents and replace parents uniformly at random
- ❖ Elitist reinsertion: produce less offspring than parents and replace worst parents
- ❖ Fitness – based reinsertion: produce more offspring than needed for reinsertion and reinsert only the best offspring

On the other hand, in local selection individuals are selected in a bounded neighbourhood and so the reinsertion of offspring takes place in the same neighbourhood. The used structures are the same as in global selection and are the follows:

- ❖ Insert every offspring and replace individuals in neighbourhood uniformly at random,
- ❖ Insert every offspring and replace weakest individuals in neighbourhood,
- ❖ Insert offspring fitter than weakest individual in neighbourhood and replace weakest individuals in neighbourhood,
- ❖ Insert offspring fitter than weakest individual in neighbourhood and replace parent,

- ❖ Insert offspring fitter than weakest individual in neighbourhood and replace individuals in neighbourhood uniformly at random,
- ❖ Insert offspring fitter than parent and replace parent.

3.3.2.6 Termination of the GA

Because the GA is a stochastic search method, it is difficult to specify convergence criteria. The application of conventional termination criteria becomes problematic as the fitness of a population may remain static for a number of generations before a better individual is found. A normal procedure is to terminate the GA after a pre - specified number of generations and afterwards the best members of the population can be tested (usually test the quality) against the problem definition. At the end, the GA may be restarted if no satisfactory solutions are found.

In conclusion, it can be understandable that the most common terminating conditions are:

- A solution is found that satisfies minimum criteria
- An upper limit on the number of generations is reached
- Allocated computation time reached
- The highest ranking solution's fitness is reaching or has reached a plateau such that successive iterations no longer produce better results
- The absolute global optimum value
- Manual inspection

When these criteria are met, the elite chromosome is returned as the best solution found so far.

3.3.3 Applying GAs in Systems Identification

The area of system identification has received extreme interest over the last three decades as many problems in control engineering, signal processing and machine learning can be viewed as a system identification problem. System identification can be defined as “the process of developing or improving a mathematical model of a physical system using experimental data to describe the input, output or response, and noise relationship” [Juang, 1994]. Generally, a system identification problem can be formulated into two optimisation tasks. The first one is structural identification of the equations and the second one is an estimation of the model’s parameters.

GAs can be applied in systems identification if each individual in the population represent a model of the plant and the objective becomes a quality measure of the model, by evaluating its capacity of predicting the evolution of the measured outputs [Vladu, 2003], [Abonyi, 2004]. The measured output predictions are compared with the real plant’s measurements.

An example of the estimation of a plant model parameters by using GAs is shown in Figure 3.16.

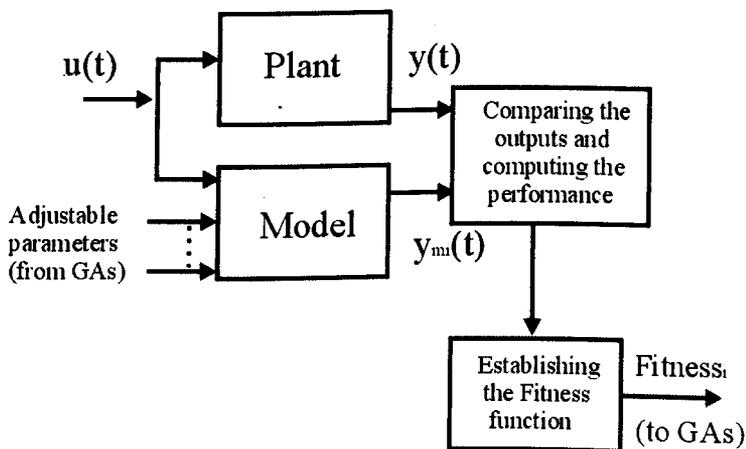


Figure 3.16: The principle scheme for parameters estimation

From the above figure, it can be seen that the principle scheme uses the *Plant* (Physical system), measured outputs $y(t)$ (experimental) as a reference (starting) point. Then, these $y(t)$ outputs are compared with the model's outputs $y_{mi}(t)$ (adjustable parameters) and a measure of the performance is obtained. Depending on the quality of the performance, the individual i has assigned the *Fitness_i* function.

Even if research of linear system identification have been tried and tested successfully for more than three decades, in practice most real – world systems are kind of, non – linear. Non – linear characteristics such as saturation, dead-zone are embedded in the very structure of many real systems. Techniques for the selection of structure and for non – linear in the parameters estimation are still the subject of ongoing research and development [Fleming, 2002], [Chipperfield, 1994b].

3.4 Improving Hybrid Genetic Algorithms for System Identification – HGASI

The need for a hybrid Genetic Algorithm (GA) approach starts from the insufficiency of linear programming and rule based (heuristic) systems in solving complex scheduling problems [Murata, 1996]. This is a strong argument if we take into consideration that we need a search algorithm which is independent of the nature of the solution domain. The main reason of using hybridisation is based on the balance of the global and the local exploitation. The first one can be found among population performed by GA and the latter one can be found around a candidate solution conducted by heuristic methods [Chainate, 2007]. Hybrid GA has been widely applied to combinatorial optimisation problems [Kido, 1993], [Murata, 1996], [Roach, 1996].

It has been generally proved that GA is a robust and efficient search technique for many optimisation problems. However, due to complexity of some problems and

as the number of variables involved increases, conventional GAs face some difficulties such as long computational time in finding the optimal solution, and premature convergence that can be subdued through variations of operations such as crossover and mutation. In the next sub – sections a novel method is introduced to overcome these difficulties. A Hybrid Genetic Algorithm including some dedicated features to adopt it to the problem of System Identification (HGASI) has been developed for this work. At the fundamental level, a Modified GA based on Migration and Artificial Selection (MGAMAS) is used to identify the system based on a given set of search space limits. Afterwards, the results of the MGAMAS are used from a Search Space Reduction Method (SSRM) to reduce the search space, delivering the new limits back to the MGAMAS for use in the next identification cycle. The main idea behind this technique is that for GAs, the accuracy and reliability of identification plus the convergence rate depends mainly on the size of the search space. As a consequence, the more the limits of the search space are reduced, the more accurate and efficient identification is possible.

Another variation included in the HGASI and presented in sub – section 3.4.3 is the *cataclysmic mutation*. It is usually activated if population prematurely converges and starts producing the same strings. As a result of this, all strings are significantly mutated unless they are the best ones. Cataclysmic mutation is a part of the well known and studied CHC algorithm developed by Larry Eshelman (1991). The technique is based on collecting the best strings found so far and stands for Cross generational elitist selection, Heterogeneous recombination and Cataclysmic mutation.

Finally, the sub – section 3.4.4 discusses the *sigma truncation scaling method*, another feature used in the HGASI. This method is one of the current scaling procedures presented in Forrest [1985] and is used to improve linear scaling both to deal with negative values and to incorporate the problem – dependent information into the mapping.

3.4.1 Search Space Reduction Method (SSRM)

In this section the technique used for reducing the search space for the HGASI is presented. The aim of this SSRM is to increase the accuracy and reliability of identification by reducing the search space during the algorithm operation. The SSRM is shown in figure 3.17 while the MGAMAS will be extendedly explained in section 3.4.2.

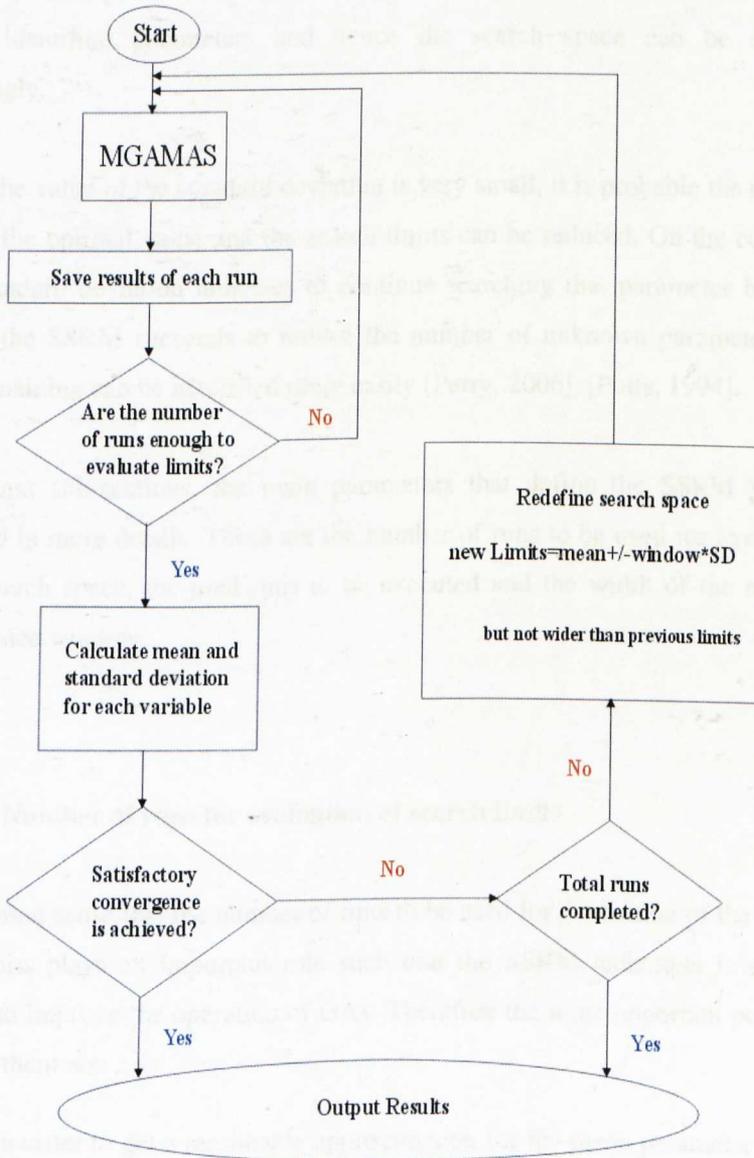


Figure 3.17: Search Space Reduction Method

This approach can efficiently eliminate search – space regions with low probability of containing a global optimum. The main idea of SSRM is simple; *Let the search space reduce for those parameters that converge quickly in order to reduce computational effort spent looking far outside the area where the optimal solution lies* [Perry, 2006]. This is achieved by accomplishing several initial runs of the MGAMAS and using a traditional genetic algorithm with elitist selection. From the identified parameters, the mean and standard deviation are then computed. The standard deviation gives us an indication of the uncertainty of the identified parameters and hence the search space can be reduced accordingly.

In case the value of the standard deviation is very small, it is probable the mean is close to the optimal value and the search limits can be reduced. On the contrary, large standard deviation indicates to continue searching that parameter broadly. Finally, the SSRM succeeds to reduce the number of unknown parameters and those remaining can be identified more easily [Perry, 2006], [Potts, 1994].

In the next sub-sections, the main parameters that define the SSRM will be discussed in more details. These are the number of runs to be used for evaluation of the search space, the total runs to be executed and the width of the reduced search space window.

3.4.1.1 Number of runs for evaluation of search limits

It is common sense that the number of runs to be used for evaluation of the search space limits plays an important role such that the SSRM technique is enough efficient to improve the operation of GAs. Therefore the most important points in choosing them are:

- In order to get a reasonable approximation for the mean parameter value, there must be a satisfactory number of runs.

- A large number of runs will consume a significant amount of time until GAs will terminate, and so it delays the time until the new limits of the identified parameters will be found. Additionally, large number of runs includes very old results that may slow the convergence down. On the other hand if the number of results is very small then it is likely to have premature convergence to local optima.

To conclude, the number of runs must be chosen carefully as large number of runs will make the system more robust, but convergence will be slow and therefore the computational time will be increased [Perry, 2006].

3.4.1.2 Total runs

Both systems' accuracy required and computational time allowed are the main factors to decide the total number of runs to be used. In theory, the results become more precise as the search space is reduced after each additional run. In reality, however, accuracy will be limited due to noise and after a time no further improvement in accuracy is possible.

Some other factors should also be taken into account. Many studies have been previously carried out on achieving the right balance of GA parameters using various combinations of them. It is interesting to note that large population sizes are preferred to classical GA (e.g. SGA), whereas SSRM is likely to work better with a small population. The mutation rate is also different, so for the SSRM, large mutation rates are preferred compared to small rates for SGA. A typical example of this, using a total number of 20 runs and 50 generations per run will result in the same computational time as would 10 runs and 100 generations per run. Therefore, the total runs should also consider some of the GA parameters like the population size and the number of generations.

3.4.1.3 Width of window

The width of window is an important parameter used in equations 3.5 and defines how quickly the search space is reduced for each variable to optimise.

$$\text{Search space} = \text{Mean} \pm \text{window} \times \text{standard deviation} \quad (3.5a)$$

or else

$$\text{Search space} = \mu \pm \text{window} \times \sigma \quad (3.5b)$$

This parameter can be explained by the three – sigma or empirical rule. This states that for a normal distribution, almost all values lie within 3 to 4 standard deviations of the mean, see figure 3.18. The empirical rule also tells how tight or loose a process is. Therefore:

- Approximately 68% of the values fall within 1 standard deviation of the mean: $\mu \pm \sigma$.
- Approximately 95% of the values fall within 2 standard deviation of the mean: $\mu \pm 2\sigma$.
- Approximately 99.7% of the values fall within 3 standard deviation of the mean: $\mu \pm 3\sigma$.

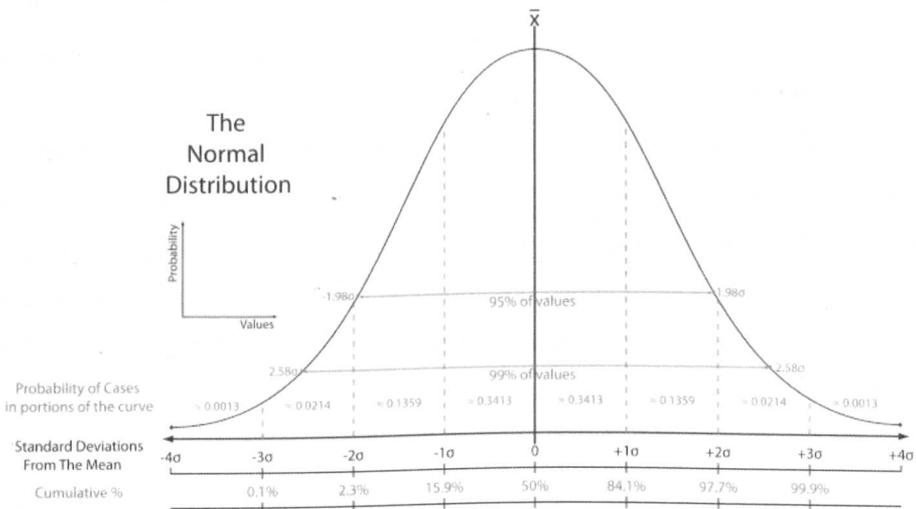


Figure 3.18: Diagram of empirical rule within three standard deviations of the mean

Furthermore, choosing the value of the window is a critical decision as it is wise to select a small value that will encourage convergence but big enough so the global solution has big chances to lie within the new, reduced, search space. In addition to this, a small value will be used for a simple problem where the results are expected to be consistent and a large one for a more complicated problem with uncertain results.

Like the total runs, the width of window will also depend on other GA parameters as well as the nature of the problem. In this thesis a value of window width of about 4 has been proven to give good performance.

3.4.2 Modified GAs based on migration and artificial selection (MGAMAS)

MGAMAS is an improved genetic algorithm whose architecture is particularly designed to alleviate the problem of premature convergence. This is the heart of SSRM and it is based on the fundamental GAMAS by Potts et al (1994) with the difference that uses a floating – point representation and contains new operators and techniques so to improve both the accuracy and the speed of identification. Figure 3.19 illustrates the MAGAMAS architecture and the key components behind it.

The basic distinguishing idea behind GAMAS is its expansion of a simple genetic algorithm (SGA). Therefore, GAMAS's concept is to include fitness scaling of each chromosome within the species, the creation and evolution of multiple parallel populations or species, migration of chromosomes between species, artificial selection of highly fit chromosomes from the species and their reintroduction back into the evolution process and lastly the species recycling.

The structures of GAMAS and MGAMAS algorithms is similar, with the difference focused on two main issues. In the first one, while GAMAS is a binary coded GA using standard genetic operators, MGAMAS suggested real encoding of variables using non – uniform mutation operators. This means the search will

vary not only across species but also over time. The second issue targets on a new tagging procedure and a reduced data length procedure which is designed for dynamic problems.

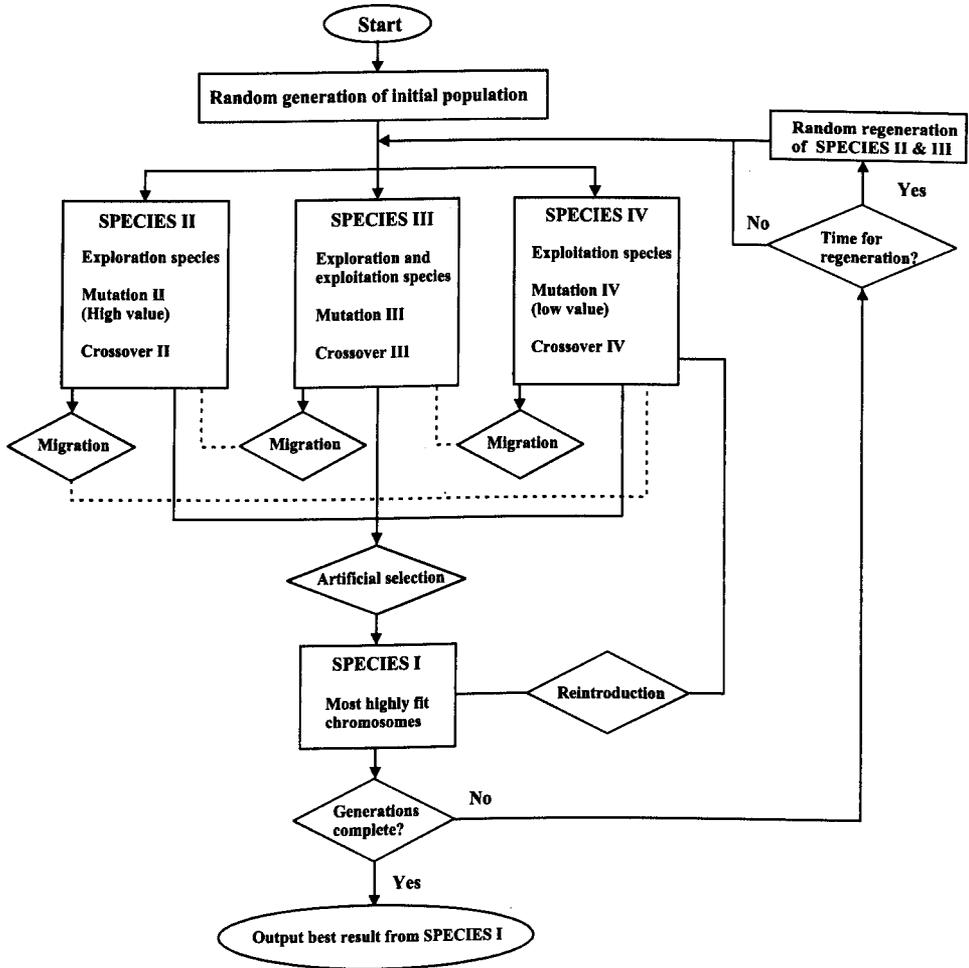


Figure 3.19: MGAMAS architecture

3.4.2.1 Floating – point representation

This representation method is applied to a population of individuals using floating – point numbers in a vector form. This means that each parameter is represented by a single value and the vector of all parameters makes up an individual. It is

argued that real – valued individuals offer a number of advantages in numerical function optimisation over binary individuals. If a real – valued individual is used, it is easier to see how the algorithm is converging. In addition, the direct representation of the floating – point numbers allows the design of mutation and crossover operators that are based on arithmetic operators and stochastic distributions [Ursen, 2003] with some of them being more difficult or impossible to implement in a binary system such as the non – uniform mutation operator introduced by Michalewicz (1999).

3.4.2.2 Multiple Species

One of the main advantages in using GAMAS and MGAMAS is the creation of multiple population or species. In the evolution process of the genetic search there are two important issues: exploration and exploitation. One of the main drawbacks of GAs has always been the trade off between exploration and exploitation. Exploration is the creation of population diversity by exploring the search space; exploitation is the reduction of the diversity by focusing on the individuals of higher fitness, or exploiting the fitness information represented within the population [Hansheng, 1999]. These factors are strongly connected as an increase in the exploitation decreases the diversity of the population or the exploration and vice versa. To be more specific, strong exploitation encourages premature convergence but has excellent ability to tune solutions when they are close to the optimum. On the other hand; weak exploitation can make the genetic search unsuccessful due to the limitation of exploration to improve the solution quality. Therefore, by splitting the population into multiple species the problem of finding a good balance between exploration and exploitation is greatly reduced.

Instead of generating a single population as in SGA, MGAMAS uses more populations (species), in the following example: SPECIES I, SPECIES II, SPECIES III, and SPECIES IV. Each of them is unique in its purpose. MGAMAS initially creates SPECIES II, SPECIES III, and SPECIES IV and then it generates SPECIES I after the artificial selection process to store the best results. So, in

order to eliminate the trade – off problem, one of the SPECIES's population is applied a high mutation rate (for exploration), one is applied a low mutation rate (for exploitation) and the last one is applied a mutation rate which lies between the other two [Potts, 1994].

SPECIES II is used as the *exploration* species. As a result of this, a high mutation rate is applied to the population and so more of the search space will be evaluated providing a better chance of deriving the global optimal.

SPECIES IV is a subpopulation used for *exploitation*. In this case, a low mutation rate is assigned so it attempts to achieve high exploitation by using a low mutation probability.

SPECIES III is an exploration and exploitation subpopulation. This way, it improves the performance of MGAMAS by providing an additional area of search and allowing for a third rate of mutation.

Finally, MGAMAS artificially selects the best individuals produced from species II, III and IV and presents them into SPECIES I whenever those are better than the elements in this subpopulation. The purpose of SPECIES I is then to preserve the best chromosomes appearing in the other species. Artificial selection carries on with each predetermined generations by replacing chromosomes in SPECIES I with chromosomes produced in the other three species if they are found to be highly fit. Additionally, the chromosomes in SPECIES I are reintroduced into SPECIES IV by replacing all of the current elements in the exploitation species and the artificial selection process continue until the number of maximum generations.

To further improve the evolution process, MGAMAS incorporate the idea of migration of randomly selected chromosomes between SPECIES II, III and IV. This allows MGAMAS to generate as many various chromosomes as possible during early and slows subsequent generations. So, all the highly fit chromosomes will be captured by artificial selection and this will further improve the solution exploration.

3.4.2.3 Fitness scaling

As it was mentioned in the beginning of section 3.4, one advantage GAMAS has in comparison to SGA is the fitness scaling of each chromosome within the multiple species. There are many cases in which chromosomes contained in the same generation, display very close numerical fitness values. These cause difficulties during the selection of chromosomes for reproduction. Occasionally, more highly fit chromosomes are left to die out while chromosomes which are weaker in fitness are selected. For that reason MGAMAS scales the fitness values of each chromosome linearly, providing a wider spread between these fitness values and ensuring that the most highly fit chromosomes are selected for mating. This scaling applies to the fitness values before the selection for reproduction begins.

3.4.2.4 Mutation Operators

One of the advantages of using multiple species and floating – point representation is the possibility to have different mutation operators. In our case, three different mutation operators are used for the species II, III and IV, as species I is held in isolation and so the chromosomes are not allowed to reproduce or mutate. The mutation operators are formed in such a way to give each species a different weight point so they can be beneficial to the enhancement of GA performance and on balancing the trade off between exploration and exploitation. Additionally, the mutation rate determines the probability of a chromosome to be mutated [Perry, 2006]. As the mutation rate is increased, mutation becomes more disruptive until the exploitative effects of selection are completely overwhelmed. Hence, high value of mutation rate allows the algorithm to explore different hyperplanes, while low mutation rate allows only the exploitation of a particular hyperplane.

One mutation operator is used for real – valued individuals and helps to increase the accuracy and convergence rate in optimisation problems it is named cyclic non – uniform mutation (Michalewicz, 1999). The main concept of this mutation is: let

the average size of the mutations to be decreased gradually within each regeneration cycle as the solution develop and let it increase again after the regeneration so it is desirable to search broadly again for new possibilities (Perry, 2006).

3.4.2.5 Crossover Operators

In the MGAMAS identification strategy, two crossover operators are used. The first one is the simple crossover and the second one is the multipoint crossover operator.

The simple crossover is similar to the one explained in the first part of this chapter with the only difference that the unknown parameters are represented by real numbers. As a result of this, the crossover operators do not change the values of individual parameters but recombine parameters from different individuals using the recombination method. The probability (p_c) of an individual in the crossover is given by the crossover rate. The value of p_c can vary widely from GA to GA and problem to problem.

The advantage of multipoint crossover instead of simple crossover is that it appears to encourage the exploration of the search space, rather than favouring the convergence to highly fit individuals early in search, so making the search more robust. The number of individuals involved in crossover for a given generation is again controlled by the crossover rate and pairs of individuals are randomly selected for crossover. The multipoint crossover also uses multi – switching points, and so recombination of parameters can be achieved from any position in the individuals.

A random number in the range [0 1] is generated and crossover of the parameter is performed when a value greater than 0.5 is restored. Figure 3.20 displays an example of this recombination method where the random numbers generated are: 0.43, 0.87, 0.69, 0.32, 0.11, 0.06, 0.72, 0.48, 0.25, and 0.97. It is clear that the multipoint crossover is happening at the 2nd, 3rd, 7th and 10th parameters.

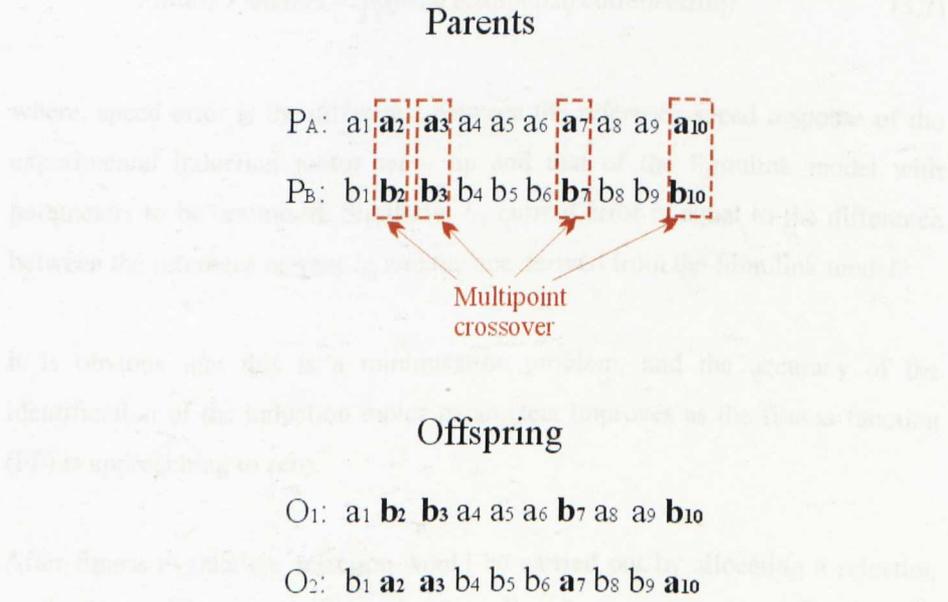


Figure 3.20: Multipoint crossover operator

In the MGAMAS both forms of crossover are used and are applied one after the other and so the total crossover rate should be considered as follows:

$$P_{ct} = 1 - (1 - P_{cs})(1 - P_{cm}) \quad (3.6)$$

where, P_{ct} is the total crossover, P_{cs} is the simple crossover and P_{cm} is the multipoint crossover.

3.4.2.6 Fitness Evaluation

For the identification problem studied in this thesis, the MGAMAS fitness is evaluated as the sum of the integral of the absolute value on both the speed and I_{sq} current error as shown in the below equation 3.7.

$$\text{Fitness Function} = \int (| \text{speed error} | + | I_{sq} \text{ current error} |) \quad (3.7)$$

where, speed error is the difference between the reference speed response of the experimental induction motor set – up and that of the Simulink model with parameters to be optimised. Similarly, I_{sq} current error is equal to the difference between the reference current I_{sq} and the one derived from the Simulink model.

It is obvious that this is a minimisation problem, and the accuracy of the identification of the induction motor parameters improves as the fitness function (FF) is approaching to zero.

After fitness evaluation, selection would be carried out by allocating a selection probability to each individual based on its fitness. However, the selection procedure becomes almost random, since during the identification process, many individuals may have very similar fitness values. A solution to this problem is to determine the selection probabilities by using a ranking procedure. So, the worst individuals are assigned a rank of 1 and the best a rank equal to the population size. Then, through the roulette – wheel selection method discussed later, reproduction is carried out.

Finally, the survival of the best results over all species and generations is guaranteed by the artificial selection procedure. For this to be perfectly achieved, it is important that the original fitness values must be also used for the artificial selection to ensure valid comparisons of individuals across different species.

3.4.2.7 Tagging

In the subsection 3.4.2.2, we have seen that artificial selection is crucial to the functioning of MGAMAS as stores in SPECIES I the fittest individuals generated so far. However, if the same individuals are selected many times then it is likely the SPECIES I to be saturated. In order to avoid this problem and guarantee the

diversity is maintained, a new idea by blocking multiple selections of the solutions is proposed. The tagging procedure is achieved as follows:

- All individuals are initially assigned a tag of 0
- All individual selected to species I, change its tag to 1
- If an individual is changed due to mutation or crossover its tag will be swapped to 0 and so it will be ready again for selection to species I.
- The tag follows the individual through the whole genetic algorithm procedure: migration, selection, reintroduction etc.

3.4.3 Cataclysmic Mutation

Cataclysmic mutation is the final part of the non-traditional genetic algorithm CHC created by Eshelman (1991). CHC is a combination of both the genetic algorithm and the $(\mu+\lambda)$ evolution strategy (contemporary derivatives of evolution strategy often use a population of μ parents and also recombination as an additional operator, called $(\mu+\lambda)$ -ES). This is believed to make them less prone to get stuck in local optima) using the reproduction operations, from μ parent individuals $\lambda \geq \mu$ offspring are created. After recombination, only the N fittest individuals from both the parents and offspring population are kept to create the next generation. Duplicates are also removed from the population. This kind of selection is known as truncation selection and so parents can be paired randomly for recombination, but only those string pairs which differ from each other by some number of bits (i.e., a mating threshold) are allowed to reproduce [Whitely, 1993]. CHC is typically run using small population sizes (e.g. 60) and at this point it uses cataclysmic mutation. Cataclysmic mutation is introduced only when the population has converged or there have been several generations without any new offspring accepted into the parent population. To be more precise, the population is reinitialised by using the best individuals as a template. This is done when the reproduction – recombination cycle reaches its termination criterion. However, the new population includes only one part of the template string as the remainder of the population is generated by mutating some percentage of bits (e.g. 35%) in the

template string. Therefore, one part of the best individual is added unchanged to the new population and in this way it is guaranteed that the next search cannot converge to a worse individual. This reinitialisation cycle is repeated until some termination circumstances are met like a fixed number of reinitialisations.

3.4.4 Sigma Truncation Scaling Method

At the start of GA runs, during the selection process, it is normal to have few superindividuals to take over a significant proportion of the finite population in a single generation. This is unwanted because it will lead the population to a premature convergence and therefore the termination of genetic algorithm. As the run matures, and the population is mainly converged, competition among population members is less strong and the procedure is likely to move about at random in the searching area. In both cases, fitness scaling can help. For example in the first case, objective function values (OFV) must be scaled back to prevent takeover of the population by the superindividuals and in the latter one OFV must be scaled up to point up differences between population members to continue to reward the best performers. In order to achieve these, a fundamental scaling procedure is used, which called linear scaling method that translates objective functions to fitness values as follows:

$$f' = \alpha \cdot f + b \quad (3.8)$$

where f is the raw fitness, f' is the scaled fitness and the coefficients a , b are chosen in such a way as the average scaled fitness f'_{avg} to be equal to the average raw fitness f_{avg} . This happens because subsequent use of the selection procedure will secure that each average population member contributes one expected offspring to the next generation [Goldberg, 1989]. It is possible to control the number of offspring given to the population member with maximum raw fitness, with equation 3.9:

$$f'_{\max} = C_{multi} \cdot f_{avg} \quad (3.9)$$

where C_{mult} is the number of expected copies desired for the best population member and its value is usually between 1.2 and 2 for small populations of 50 – 100 individuals. However, linear scaling becomes more complicated from the existence of negative objective scores. This is a problem that appears later in a run when most population members are highly fit but a few lethal have a very low value. To overcome this dilemma, sigma truncation scaling method is applied, in which a constant value is subtracted from raw fitness values as follows:

$$f' = f - (f_{avg} - c \times \sigma) \quad (3.10)$$

where f_{avg} and σ are the average and standard deviation of the objective function for the population. The constant c is chosen as a reasonable multiple of the population standard deviation and for extremely high-quality individuals varies between 1 and 3. In case of poor individuals with a value of c below the average and negative fitness values, these individuals are assigned a fitness of zero.

3.5 Chapter summary

This chapter mainly focuses on two parts. The first part introduced the GAs and the second one tried to explain various solutions that improved the IM parameter identification through GAs in our work.

Firstly, the current chapter has introduced the foundation for understanding genetic algorithms, their power, their mechanics and their weaknesses. As a starting point, it focused on the terminology of “Optimisation”, that is the process of making something better. It also analysed the classification of optimisation algorithms according to different optimisation techniques, highlighting their importance. Next, problems of optimisations were mentioned and a schematic example of them was presented. Then, Genetic Algorithms have been introduced and analysed as a stochastic global search technique. The basic concept of GAs is designed to simulate processes in natural system necessary for evolution,

specifically those that follow the principles of survival of the fittest. GA differ from traditional search algorithms in that they (1) work on an encoding of the parameter set rather than the parameter set itself, (2) search from a population of points, not a single point, (3) use an objective function rather than derivative information or other auxiliary knowledge and (4) work based on probabilistic rather than deterministic rules.

The chapter has also presented the detailed mechanics of a simple, two – operator genetic algorithm. Genetic algorithms operate on a number of strings, with the string coded to represent some fundamental parameter set. The most frequently used representation of chromosomes in the GA is the ones of a single – level binary string that linked together to form a chromosome.

Next, the starting point of the search is to create a group of individuals to form an initial population. Once a population of solutions is created, each of the chromosomes in the population must be evaluated to see how well they solve the problem at hand. This is achieved with the objective function which decodes the chromosome, evaluates it and returns the performance to the genetic algorithm.

Therefore, when the evaluation procedure of all of the solutions has been completed, two or more must be selected to be parents and to create offspring for the next generation. In the selection process, a string with a high fitness value has more chances to be selected as one of the parents than a string with a low fitness value. One of the most popular ways to implement the selection is (1) the Roulette Wheel Selection Method, (2) Tournament Selection and (3) Stochastic Universal Sampling.

Furthermore, genetic operators are usually used in GAs to generate diversity (mutation) and to combine existing solutions into others (crossover). Both crossover and mutation are applied to create new string populations. Lastly, when all the above mentioned criteria are met, the elite chromosome is returned as the best solution found so far.

The last section of GA's description presented an approach of GAs for systems identification. Generally, a system identification problem can be formulated as an optimisation task where the objective is to find a model and a set of parameters that minimise the prediction error between the plan outputs (measured data) and the model output.

The second largest part of this chapter has initially presented a new GA identification strategy applied to our model. The strategy uses a two – tier approach techniques whereby the first one is a search space reduction method (SSRM) and the second one is a modified GA based on migration and artificial selection (MGAMAS). The concept is that the SSRM uses the results of the MGAMAS to reduce the search space and return new search limits to the MGAMAS for further identification. However, the MGAMAS is the heart of the method as provides a robust search that simultaneously explores the search space and tries to preserve and improve the fittest individuals.

The applied MGAMAS includes a multiple species population, with fitness scaling and appropriate mutation and crossover operators as well as other features that are capable of improving the computational performance, in terms of identification accuracy and computational speed by controlling the search direction.

Another technique that is described in this chapter is called cataclysmic mutation. This algorithm introduces a new diversity into the population via a form of restart the search when the population starts to converge. Cataclysmic mutation uses the best individual in the population as a template to re-initialise the population.

Finally, the sigma truncation scaling method has been discussed. This method was designed as an improvement of linear scaling to deal with the negative evaluation values that appear in a run when most population members are highly fit but few of them have a very low value.

To conclude, we can finally assert that, Genetic algorithms are a very powerful tool for systems identification in searching for model parameters that best suit a real system characteristics.

Chapter 4

Experimental Implementation

4.1 Introduction

This chapter will briefly describe the experimental setup that was used to develop and test the vector control algorithms and parameter identification schemes. In this research work, the experimental responses speed and I_{sq} current will be used as a comparison with the modelled ones to obtain the fitness function error, presented in the following chapters. The rig was originally used in projects by Sumner [1990] and Blasco [1996] and developed by Leonardo Cascella. However, more recently, the same rig used by Jasim [2009], who set up all the different experimental systems in this work.

The test rig is based on a squirrel-cage 4kW induction machine and a 7.5kW IGBT inverter. The induction machine is coupled to a DC machine, with associated converter of higher power rating, and these act as loading devices. The DC load machine is rated at 10kW and fed by a Eurotherm 4-quadrant converter.

The IGBT inverter is modified to allow access to the gate drive circuitry of the power devices. The gate drive circuitry is interfaced to current mirror card and this circuit is itself interfaced to a PC via a DS1104 card using the PWM signals and running motor control code. The processing hardware can be connected to the motor drive by signal – acquisition, signal – conditioning and digital input – output circuits.

Processing capability is provided by using the DS1104 controller card, which is placed into the PCI slot of the personal computer. The DS1104 card contains all necessary peripherals (ADC, DAC, counters, timers, PWM etc.) and computing power (offered by the MPC8240 PowerPC 603e master-processor and TMS320F240 slave-DSP) for implementation of complex drives structures. This card is working with MatLab/Simulink environment. The software associated to the card provides the control for the implementation process from simulation up to real time experiment.

In the next sub – sections the experimental system and the motor drive will be described and the specifications of the Induction Motor used in this project will be presented. Finally, it will be discussed in more detail the dSPACE DS1104 controller card, and the Digital Signal Processor to Motor Drive Interface System.

4.2 Experimental System

Figure 4.1 represents the structure of the experimental rig. The external circuits include an inverter interface board, and a transducer board that were built by Turl [2002], while a filter/amplification board, and a current mirror board were used in other development work within the PEMC group at Nottingham University.

The developed system is capable of running what would be considered high performance AC motor control and the DSP and host PC set-up is such as to provide a highly flexible experimental research platform.

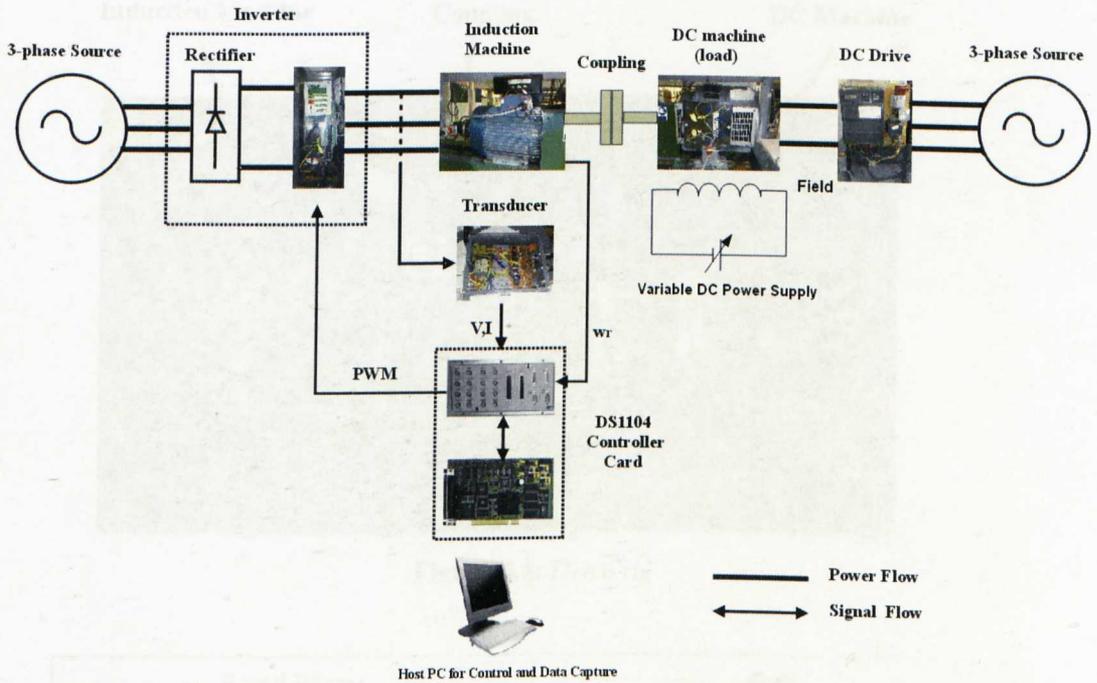


Figure 4.1: Structure of the experimental drive rig

4.3 Motor Drive

Figure 4.2 shows the motor test bed. The motor drive rig consists of a delta (Δ) connected, 4-pole, 4kW, closed and skewed slot, squirrel cage induction motor, manufactured by Asea. This machine is fed by a 7.5kW FKI Industrial Drives IGBT inverter and fitted with the 2,500-line encoder. The inverter has an integral dynamic braking resistor fitted suitable to dissipate energy due to deceleration.

Information about the motor, such as the number of rotor slots is known from the works of Blasco [1996] and Sumner [1990]. The specifications of the motor are presented in Table 4.1.

The DC drive is configured to operate under its own control and is set-up to provide variable torque demands. An external control box varies the torque

demand to the drive by varying a fed back reference voltage.

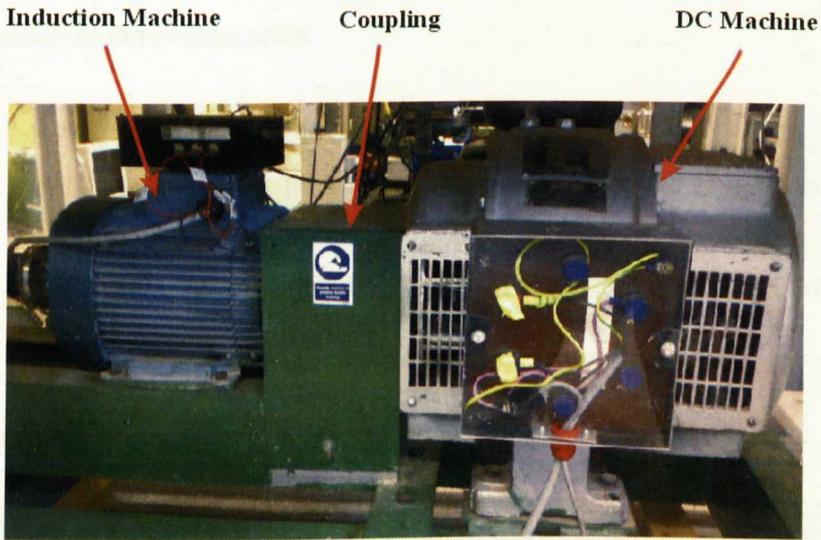


Figure 4.2: Drive rig

Rated Power	4kW
Rated Speed	1420rpm
Rated Torque	26.9Nm
Rated Voltage	415V
Rated Current	8.4A
No. Poles	4
Rated Phase I_{sq}	9A
Rated Phase I_{sd}	4.9A
No. Rotor Slots	28

Table 4.1: Specification of the Induction Machine

4.4 The dSPACE DS1104 Controller Board

The DS1104 controller card controls all the subsystems of the experimental equipment via the PC. The computing power is provided by the MPC8240 PowerPC 603e processor and TMS320F240 DSP on the card and it is necessary for real time control tasks. The communication between the processors and

peripherals takes place on the internal 24 bit wide bus. An efficient interrupt system is provided for this card to improve its performance. The dSPACE system consists of three components:

- ❖ The DS1104 controller board.
- ❖ A breakout panel for connecting signal lines to the DS1104 controller board.
- ❖ Software tools for operating the DS1104 board through the SIMULINK block diagram environment.

Figure 4.3 shows a block diagram of the DS1104 controller board.

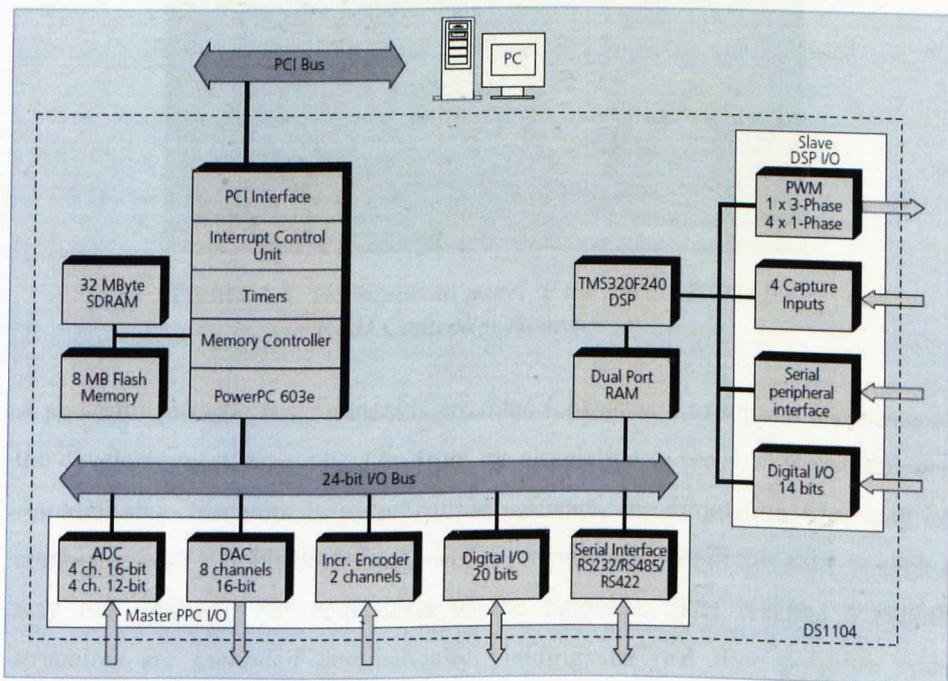


Figure 4.3: Block diagram of the DS1104 controller board

From the above Figure 4.3, it can be seen that the DS1104 controller card has eight, 16 – bit D/A converters each with an output range of $\pm 10V$. It also contains eight A/D channels. The first four channels share a single 16-bit A/D converter through an analogue multiplexer. Each of the remaining four channels has a dedicated 12-bit A/D converter, allowing the simultaneous sampling of four analogue input signals. The input range for all eight channels is $\pm 10 V$. The

DS1104 card also accommodate a 20-bit parallel digital I/O, two PWM ports, and two incremental encoder channels to picks up the encoder signal of the motor to measure the rotor speed.

The external devices include all necessary interfaces and measurement hardware required to control the motor drives and they are connected to the DS1104 R&D Controller Board via the breakout panel. Figure 4.4 displays the breakout panel for the dSPACE DS1104 R&D Controller Board. For further details about the DS1104 controller card, please refer to [dSPACE, 2004].

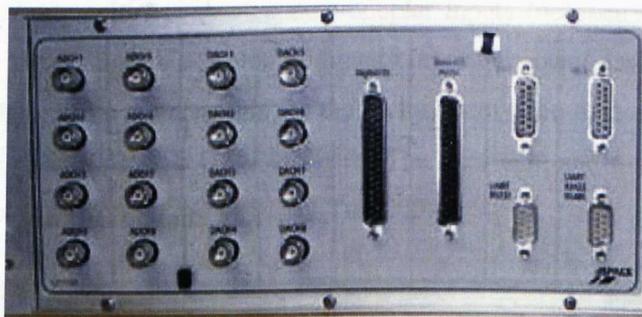


Figure 4.4: *The breakout panel of the dSPACE DS1104 R&D Controller Board*

A powerful software environment is provided for this card to support all phases of the development of an application from the simulation to physical measurements. As MatLab – Simulink is a compatible software, the simulation files may be completed with RTI block elements which are provided within this card, in such a way that realising the connection to the hardware. The resulted Simulink structures are compiled automatically yielding the real time program code executable on the controller card's processors [Jasim, 2009].

4.5 DSP to Motor Drive Interface System

This part focuses on the connection of the motor drive to the DS1104 controller card through the external devices. Specifications for the development of equipment interfaced to the DS1104 R&D controller card are contained in the

dSPACE Control Desk Experiment Guide [dSPACE, 2004]. The component parts of this hardware interface are outlined in the following sub – sections.

4.5.1 Inverter Interface – Incorporating Dead – time Protection

The output from the PWM port on the controller board is fed to the inverter interface as 3-channel corresponding to the three motor phases, current level gate drive signals (by using current mirror board). This has higher noise immunity, in what is quite an electrically noisy environment. Sending voltage level signals, the signals are highly suited to driving opto-coupler inputs (necessary isolation for safety). The signals are not suitable to directly drive the power devices and must first be separated into six channels for the inclusion of dead-time delays. In this work the dead-time delay was set at 5 μ sec. The minimum and maximum pulse time were set to be 10 μ sec and 115 μ sec [Turl 2002].

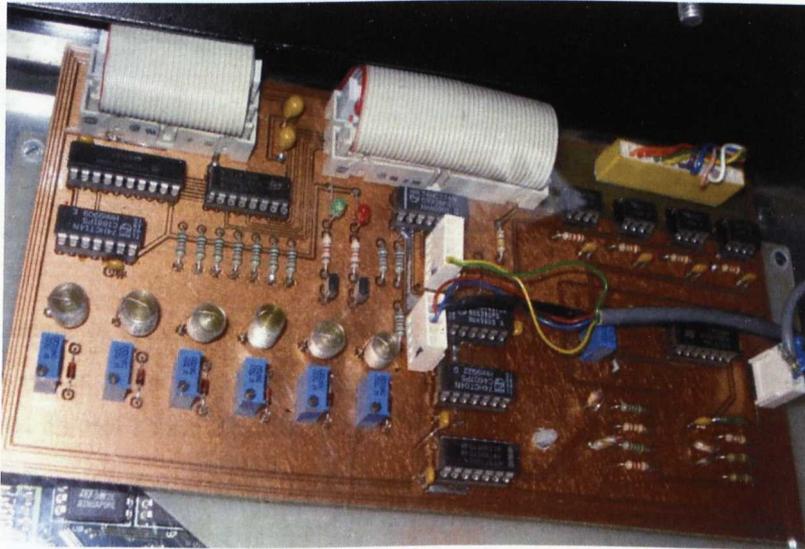


Figure 4.5: Interface card of the FK1 industrial drive inverter

Protection functionality is necessary and is included by the interface circuit. Figure 4.5 shows the board fitted to the FK1 inverter. A manual trip button is added to prevent the operation of the inverter and it is useful when the control of the system is lost. [Jasim, 2009], [Turl, 2002].

4.5.2 Voltage and Current Measurement

Line-to-line voltages and line currents are measured, in the motor rig, by suitable transducers placed between the inverter output and the motor input terminals.

Current measurement is required for the current controllers in vector control and for drive protection in commercial products. The transducers used are LEM hall-effect devices.

Voltage measurement is not always used in commercial projects. As the inverter can be considered a voltage source, sometimes reference value of the voltages will be used if it is needed. In this work, similarly to the currents, all three voltages are measured, as these will be used to improve the estimation of the electrical parameters in the last chapter.

The outputs of the transducers are passed through necessary signal conditioning circuit (amplifiers with gain and offset adjustment). Connection is made to the breakout panel of the DS1104 controller card using screened cable and BNC type connections.

4.5.3 Analogue Filtering

The outputs of the current transducers are passed, via the screened cable, through filters and on to the connection of the A/D converters in the breakout panel of the DS1104 control board. The filters are 2nd order low-pass butterworth type, with a cut-off frequency of around 1.3kHz, and are used to prevent anti-alias effects associated with a digitally sampled system [Balmer, 1991].

4.6 Implementing the Drive Structures

There are four steps in order to perform the implementation of a drive structure on DS1104 based experimental equipment. These steps are listed as:

- Creation the working simulation model of the desired drive structure through Simulink / MatLab, its extension with Real-Time Interface (RTI) blocks;
- Defining the sampling unit;
- Implementation of the software measuring, protection and command subsystem;
- Implementation of the virtual control panel;

Real-Time Interface (RTI) blocks is the link between dSPACE's real-time hardware and the MATLAB/Simulink development software from The MathWorks. It extends the C code generator Real-Time Workshop so the Simulink models can be implemented very easily on dSPACE real-time hardware.

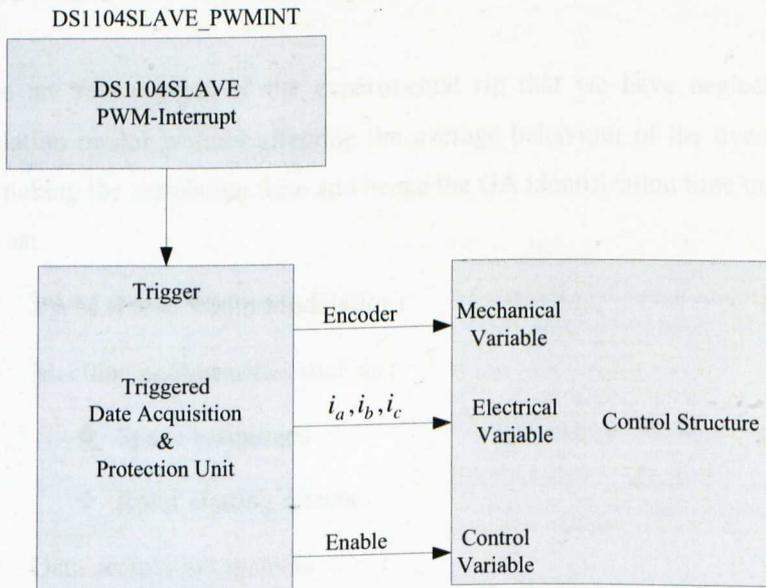


Figure 4.6: Structure of an implemented drive system

Figure 4.6 indicates the general structure of an implemented drive system. For a different drive system, the implementation of the measurement, protection and command block and the control structure block will be changed according to the requested task. The block *DS1104SLAVE_PWMINT* represents the RTI block which connects the simulation structure to the real PWM interrupt of the DS1104 card. More details about the implementation of the software measuring, protection

and command subsystem as well as the implementation of the virtual control panel can be found at Incze [Incze, 2005].

4.7 Implementing the Control Systems

The control system in the experimental rig is implemented as in the simulation model. The speed and current PI controllers in the experimental rig are in z – domain while in the simulation model are in s – domain. Hence, a transformation from s – domain to z – domain using bilinear method was necessary in order to find the PI controller parameters for the experimental rig. When designing the speed loop, dynamics of the current loop has been neglected, being much faster than the speed one. The sampling frequency of the current and speed control loops as well as the PWM frequency is set at 4kHz.

There are few aspects of the experimental rig that we have neglected in the simulation model without affecting the average behaviour of the overall system and making the simulation time and hence the GA identification time much faster, such as:

- PWM (Pulse Width Modulation)
- Machine nonlinearities such as :
 - ❖ Space harmonics
 - ❖ Rotor slotting effects
- Data acquisition systems
- Transducer delays

The presence of the PWM, slotting and space harmonics affects in the simulation model would result in high frequencies which in fact would not affect the average behavior of the overall system. Saturation is being accounted for the reason that we vary the field and torque producing currents for the identification of the motor parameters in different working conditions.

It is worthy to mention that a simulation model using z – domain PI controllers has been tested for identification. The results obtained, both as regards the vector control's behaviour and the estimation of the parameters are coincident with s -domain simulation.

4.8 Accuracy/Resolution of the experimental data

In the paragraphs below, a discussion concerning the expected accuracy and resolution of the identification will be addressed.

The resolution, which represents the smallest difference between two output levels of the converter is $20V/2^{16} = 0.305mV$. That is, the output voltage of the D/A converter can be varied from -10V to +10V in 0.305mV steps. The resolution of the 16-bit channels is 0.305mV; while the resolution for the 12-bit A/D converters is $20V/2^{12} = 4.9mV$.

Two incremental encoder channels (used as speed transducers) are used to pick up the encoder signal of the motor to measure the rotor speed. The actual rotor speed is calculated using 3000 pulses per revolution encoder which results in high measurement accuracies better than 0.01° mechanical (1.0 arc minutes) from one cycle to any other cycle.

Line to line voltages and line currents are measured by suitable transducers placed between the inverter output and the motor input terminals. These are called LEM Hall – effect devices. In fact the errors between nominal value and measured values generally result from the associate temperature changes of Hall – effect sensors. The reasonable work range of Hall sensors is up to 20 kHz. Once the switching frequency is higher than 20 kHz, the error is significant and cannot be omitted. However, in the experimental setup used for this thesis, the switching frequency is 4 kHz and therefore the effect caused by temperature can be neglected and the measurement accuracy is high [Yeong, 2004]. The current transducers used were made by TELCON and the overall accuracy is 0.5% of its nominal primary current.

The effect on the resolution and accuracy of the measured quantities will not significantly influence the estimated I.M. parameters.

4.9 Chapter Summary

In this chapter a briefly explanation of the experimental setup has been introduced. The built DS1104 controller card based experimental equipment is a very useful tool in research and development of high performance AC electrical drives.

The overall structure of the experimental system has been shown and description of all the hardware equipment was given. The system can be operated over the full rated speed and flux range of the induction machine and both the speed and current demand can be instantaneously varied. Fitting of speed encoders and current and voltage transducers, allows these measurements to be readily developed, tested, compared and used to the Genetic Algorithm code.

The implementation of a drive structure on DS1104 based experimental equipment is accomplished according to the principle 'from simple to complex'. This offers the most reliable way regarding the fail – safe operation of the hardware.

Lastly, the implementation of the control system is represented and aspects of the experimental rig that have been neglected in the simulation model are introduced.

Chapter 5

Applying GAs to Vector Controlled Drives at Rated Operation Parameter Identification

5.1 Introduction

This chapter describes a novel, accurate and non – intrusive approach for identifying induction motor equivalent circuit parameters based on experimental transient measurements and using an off line Genetic Algorithm (GA) routine to tune a linear machine model.

Transient measurements of current and speed can be taken in a non – intrusive way directly on the drive in which the motor is installed for its normal operation. Therefore precise knowledge of the electrical parameters will determine enhanced control of the drive.

The motor used for this research is a 4 kW, 4 – pole I.M., in a vector controlled drive while a Matlab Simulink model was utilised within the heuristic GA based identification routine. As it was mentioned in the 2nd chapter, vector control in an induction motor drive allows high – performance control of torque and speed only if both the electrical and mechanical parameters of the machine are accurately known in all operating conditions.

The basic idea of this research work is that the evaluation of the electrical motor parameters can be achieved by minimising, using a GAs approach, the error between the experimental response (speed or current) measured on the experimental motor drive and the respective one obtained by a Matlab – Simulink model implementing the same structure and control of the experimental rig, but with varying electrical parameters [Trentin, 2006].

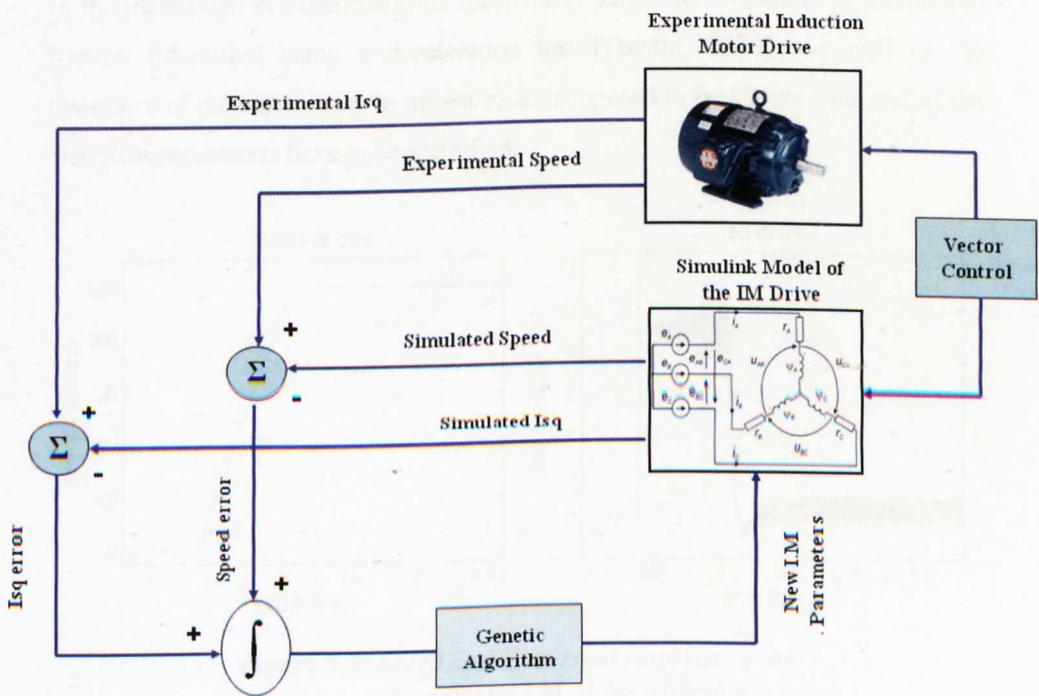


Figure 5.1: Block diagram representing the experiment.

5.2 Description of the system

Figure 5.1 represents the optimisation experimental set – up. Measurements of experimental transient responses of speed and current from the vector controlled electrical drive test rig are stored and used as reference signals for the optimisation. A fitness function based on the integral of the absolute error between these reference signals and the ones produced by the simulation model reproducing the experimental rig, are evaluated by the GAs intelligent search technique which will find the correct system electrical parameters that minimise it. [Kampisios, 2008b].

5.2.1 Experimental Induction Motor Drive

In the identification experiment the mechanical parameters, moment of inertia and friction (identified using a deceleration test [Trentin, 2006]), as well as the resistance of the stator (easy to measure) are supposed to be known - the rest of the electrical parameters have to be identified.

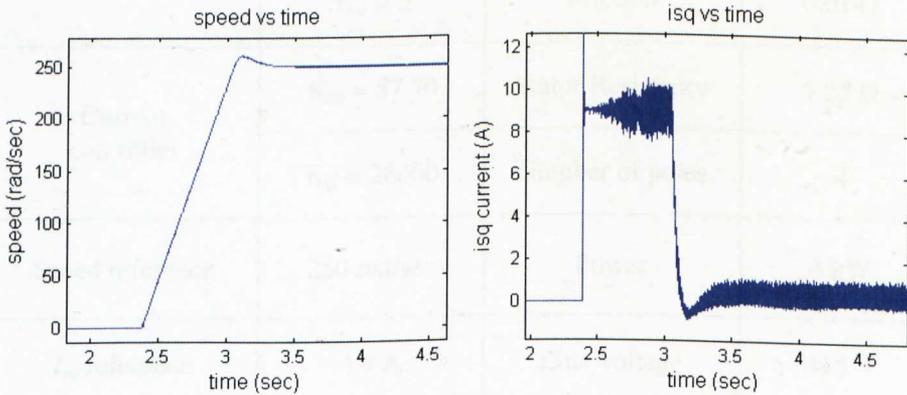


Figure 5.2: *Speed and I_{sq} current response of the experimental I.M. drive without load.*

Table 5.1 presents some of the most important characteristics from the Experimental vector controlled Induction Motor which are also used in the simulation model. The speed and current (I_{sq}) transient responses used as reference signals in the GAs optimisation are shown in figures 5.2 (without load). In figure

5.3 the same transient responses are represented but with an applied load torque (15.5Nm) at 6.5sec. They represent the system response to a speed (rated) demand of 250 rad/sec and I_{sq} current (rated) demand of 9A and within a constant (rated) flux of 4.9A.

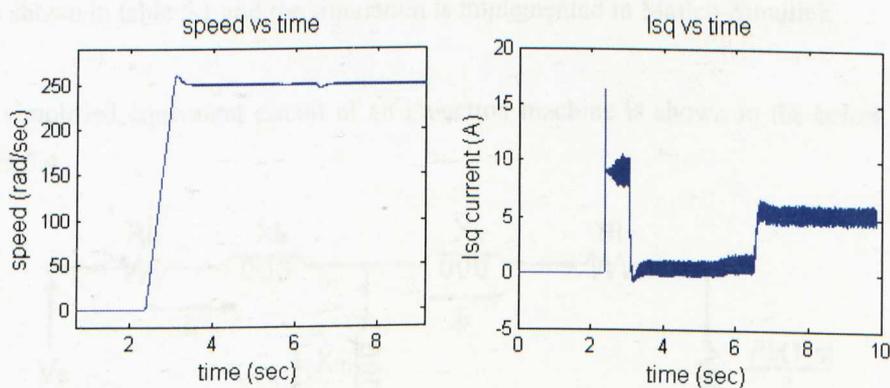


Figure 5.3: Speed and I_{sq} current response of the experimental I.M. Drive with load.

Vector Control Parameters		Induction Motor Parameters	
Speed controller	$K_{sp} = 0.66$	Inertia J	0.152
	$K_{si} = 5$	Friction	0.0147
Current controller	$K_{cp} = 57.70$	Stator Resistance	5.25 Ω
	$K_{ci} = 28000$	Number of poles	4
Speed reference	250 rad/sec	Power	4 kW
I_{sd} reference	4.9 A	Line voltage	415 V
Rotor time constant	0.1508	Frequency	50 Hz
Current limits	9A		
Voltage limits	1000V		

Table 5.1: Experimental vector controlled I.M. parameters

5.2.2 Simulated Induction Motor Drive

The induction machine is modeled using a traditional dynamic model in *abc* reference frame where the vector control parameters are the same experimental ones shown in table 5.1 and the simulation is implemented in Matlab-Simulink.

The simplified equivalent circuit of an induction machine is shown in the below figure 5.4.

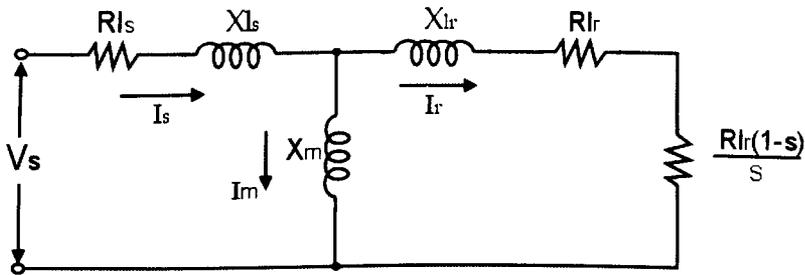


Figure 5.4: Simplified equivalent circuit of an I.M.

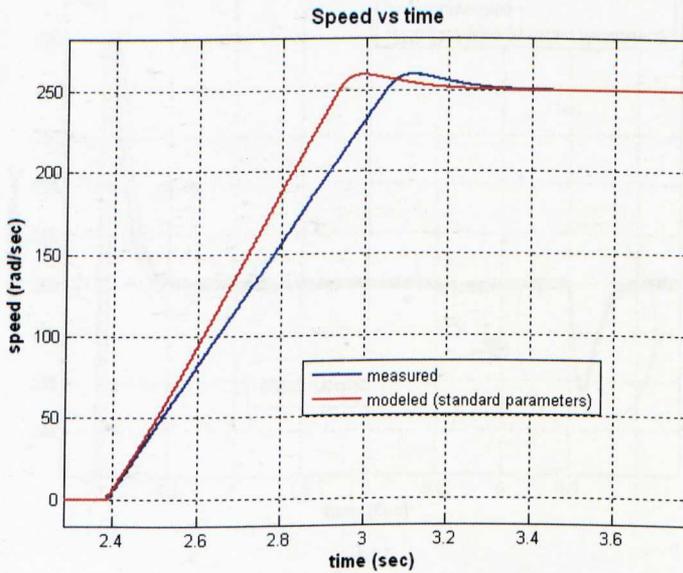
The equivalent steady state circuit of a three-phase Induction Motor can be represented by a combination of resistance and inductance from both the stator and the rotor. So, from the equivalent circuit, the rotor current and the torque equations can be described in the equations below:

$$I_r = \frac{V_s}{\sqrt{\left(R_{ls} + \frac{R_{lr}}{s}\right)^2 + (X_{ls} + X_{lr})^2}} \quad (5.1)$$

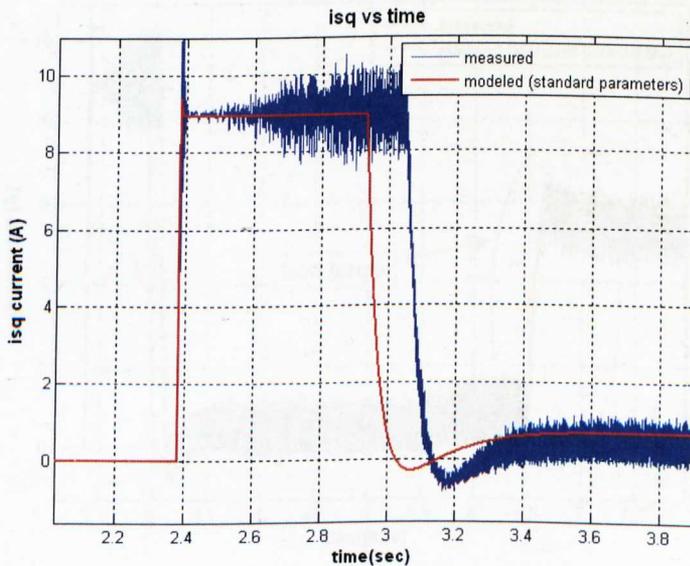
$$T = \frac{3R_r p}{s\omega_e} \frac{V_s^2}{2 \left[\left(R_{ls} + \frac{R_{lr}}{s}\right)^2 + (X_{ls} + X_{lr})^2 \right]} \quad (5.2)$$

The nameplate data of the examined motor are: $P = 4kW$, $U = 415V$, Δ connection, $I = 8.4$ A, $n = 1420$ rpm. Using the traditional Induction Machine parameters identification based on “no – load” and “locked – rotor” tests the following electrical parameters can be obtained: stator resistance $R_s = 5.25\Omega$, rotor resistance $R_r = 3.76\Omega$, magnetising inductance $L_m = 0.5343H$, stator

leakage inductance $L_{ls} = 0.04H$ and rotor leakage inductance $L_{lr} = 0.033H$. In the following, these values will be referred as “standard parameters”. A comparison between the modelled and measured time responses is shown in the below in figures 5.5, 5.6 and 5.7. In these figures a constant (rated) flux is applied at 0sec while a rated speed response is applied at approximately 2.4sec. In Figures 5.6 and 5.7 a load torque (15.5Nm) is also applied at 6.5sec.



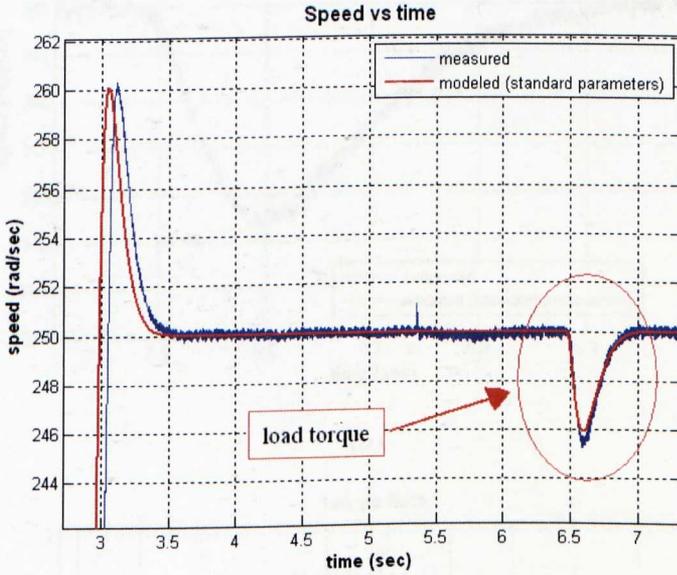
(a)



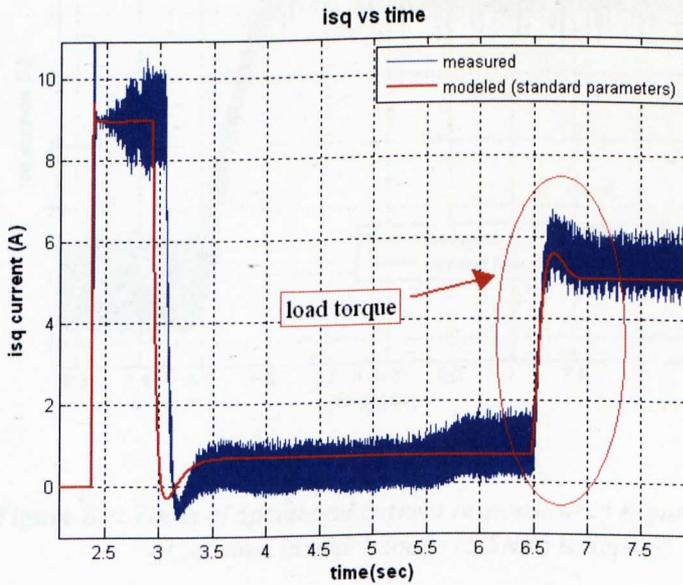
(b)

Figure 5.5: Measured and modelled Speed (a) and I_{sq} current (b) responses: no load and standard parameters.

As it can be noted there is an evident mismatch between the modelled and experimental responses indicating a difference between the experimental and estimated parameters. The results illustrate that this mismatch takes place only during transient response behaviour while at steady state the error between the responses is zero.



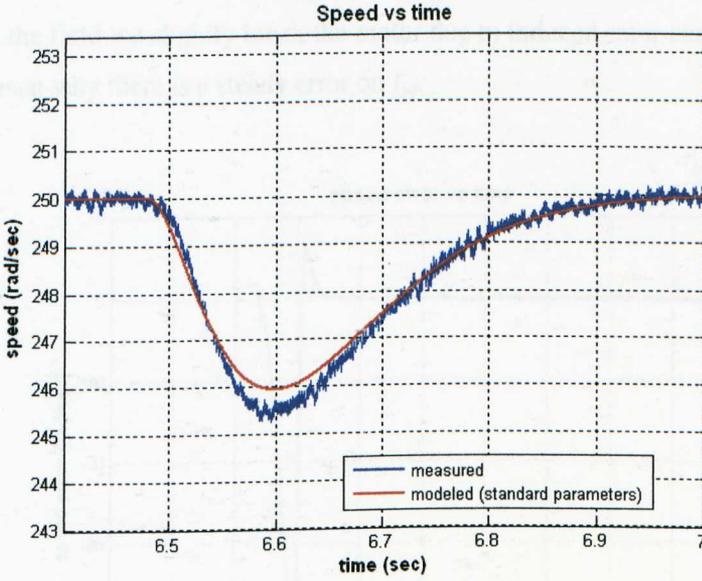
(a)



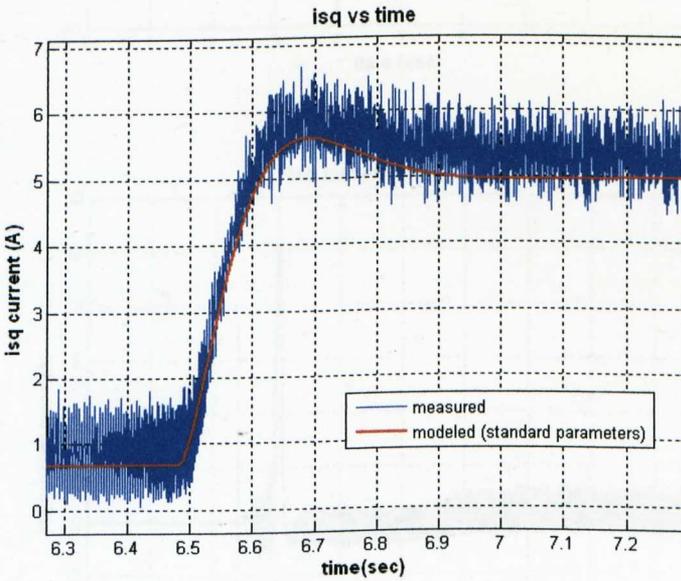
(b)

Figure 5.6: Measured and modelled Speed (a) and I_{sq} current (b) responses: with load (15.5Nm) and standard parameters.

Figures 5.6 and 5.7 shows a small mismatch while a load torque is applied but this mismatch occurs only during the peak time of the responses.



(a)

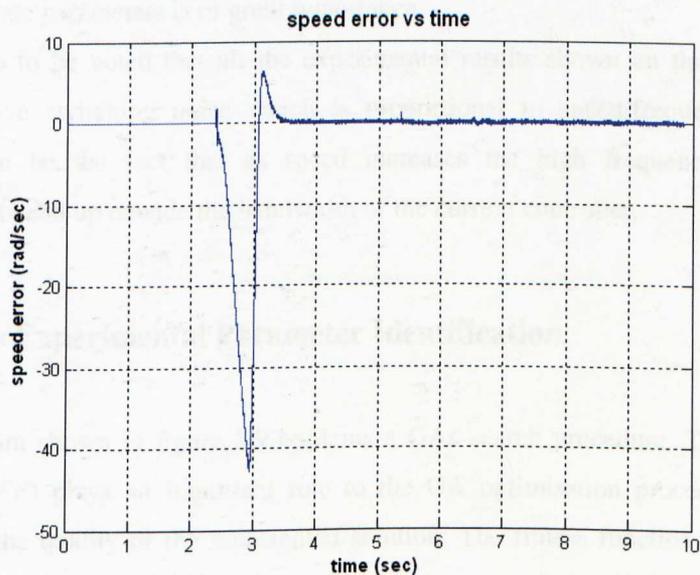


(b)

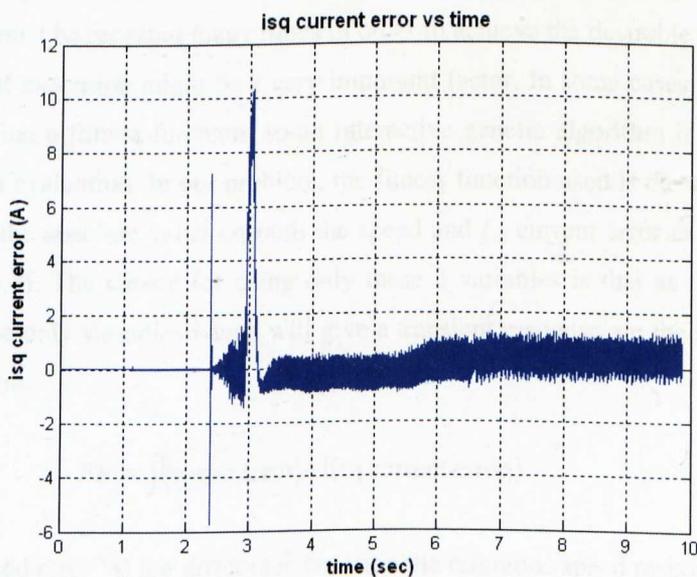
Figure 5.7: Zoom of Speed and current response of in Figure 5.6 at the time in which load (15.5Nm) is applied.

In fact, from figures 5.6b and 5.7b, there appears to be a steady state error on I_{sq} . This is because the drive used for the DC machine is only exciting the

armature current while the field current is being supplied by external voltage supply with a rectifier circuit to provide the field of the DC machine. When we switch the field we slightly break the motor due to induced rotor currents and this is a reason why there is a steady error on I_{sq} .



(a)



(b)

Figure 5.8: Speed (a) and current (b) error between measured and modelled (standard parameters) speed and current responses

Figure 5.8 represents the speed (a) and I_{sq} current (b) error between the experimental and modelled response using standard parameters (error between the two responses in figure 5.6(a) and figure 5.6(b)). It can be seen that in both graphs an error appears during the transient response period while in steady state and loaded condition the error is zero. Hence, during transient response the need of very accurate parameters is of great importance.

It has also to be noted that all the experimental results shown on the previous figures have switching noise which is proportional to speed/frequency. One reason can be the fact that as speed increases the high frequency current components end up outside the bandwidth of the current controller.

5.3 GAs Experimental Parameter Identification

The diagram shown in figure 5.9 explains a GAs search procedure. The fitness function (FF) plays an important role to the GA optimisation procedure as it measures the quality of the represented solution. The fitness function is always problem dependent. An ideal fitness function connects closely with the algorithm's goal and yet may be evaluated quickly because usually a genetic algorithm must be repeated many times in order to achieve the desirable result and the speed of execution might be a very important factor. In some cases, it is even hard to define a fitness function, so an interactive genetic algorithm is used that uses human evaluation. In our problem, the fitness function used is the sum of the integral of the absolute value on both the speed and I_{sq} current error as shown in equation (5.3). The choice for using only these 2 variables is that as the flux is constant, the only variables which will give a transient response are the speed and the I_{sq} current.

$$FF = \int (|\text{speed error}| + |I_{sq} \text{ current error}|) \quad (5.3)$$

where, "speed error" is the difference between the reference speed response of the experimental induction motor drive and that of the Simulink model. Similarly, " I_{sq} current error" is equal to the difference between the reference current I_{sq} and the one derived from the Simulink model.

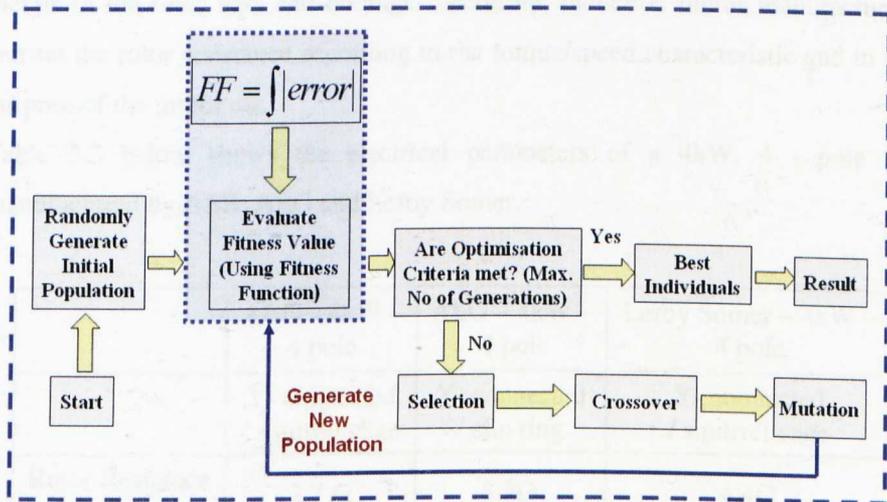


Figure 5.9: Block diagram representing the GAs routine

It is clear that this is a minimisation problem, whereby the accuracy of the identification of the electrical motor parameters improves as the fitness function (FF) approaches zero.

As previously mentioned the unknown electrical parameters of the motor are the rotor resistance (R_r), the magnetizing inductance (L_m), the leakage inductance of the stator (L_{ls}) and of the rotor (L_{lr}). Given a wide range of industrial I.M. the bounds for the GA search were considered in a way to create intervals which encompasses most practical parameter values (5.4):

$$\begin{aligned}
 1 \Omega &\leq R_r \leq 10 \Omega \\
 0.1 \text{ H} &\leq L_m \leq 1 \text{ H} \\
 0.005 \text{ H} &\leq L_{ls} \leq 0.1 \text{ H} \\
 0.005 \text{ H} &\leq L_{lr} \leq 0.1 \text{ H}
 \end{aligned}
 \tag{5.4}$$

The range of parameter values for different manufactured 4kW induction motors will mainly depend on the pole numbers. In addition, the rotor resistance can be varied for the same motor power but with different torque/speed characteristics [Wildi, 2006]. As an example, for designing a squirrel – cage motor, the rotor

resistance can be set over a wide range by using copper, aluminum, or other metals in the rotor bars and endings. Therefore, induction motor manufacturers can set the rotor resistance according to the torque/speed characteristic and to the purpose of the motor use.

Table 5.2 below shows the electrical parameters of a 4kW, 4 - pole IM manufactured by ABB, AEG and Leroy Somer.

	ABB – 4kW – 4 pole	AEG – 4kW – 4 pole	Leroy Somer – 4kW – 4 pole
	Y- connected / squirrel cage	Y- connected / slip ring	Y- connected / squirrel cage
Rotor Resiatnce (R_r)	2.7 Ω	3.2 Ω	4.4 Ω
Stator Resistance (R_s)	3.9 Ω	5.1 Ω	6.2 Ω
Magnetizing Inductance (L_m)	0.85H	0.48H	0.35H
Rotor Inductance (L_r)	0.799H	0.452H	0.331H
Stator Inductance (L_s)	0.818H	0.452H	0.331H

Table 5.2: Electrical Parameters from different 4kW – 4 pole motor manufacturers.

It is evident from the Table above that the expected range of parameter values for a 4kW / 4 pole I.M. will be as follows:

$$2\Omega \leq R_r \leq 8\Omega$$

$$0.2H \leq L_m \leq 0.9H$$

$$0.15H \leq L_m \leq 0.85H$$

$$0.15H \leq L_m \leq 0.85H$$

It is also well known that hand tuning of the vector control parameter can be a difficult task to do. On the other hand the optimal GA parameter settings (such as mutation, crossover rate, initial population, maximum generation) have been the subject of numerous studies in the GA literature. Most people use what has worked well in previously reported cases as there is no conclusive agreement on

what is best; For the identification of any induction motor parameters through GAs, it can be said that a given range of GA parameters which will give accurate estimation are as follows:

$$0.009 \leq \text{Mutation} \leq 0.15$$

$$0.7 \leq \text{Crossover} \leq 0.99$$

$$25 \leq \text{Initial population} \leq 50$$

$$60 \leq \text{Maximum generation} \leq 100$$

5.3.1 Application of the SSRM

It has been found by experience that, above all in identification problems, it would be desirable that the GA routine could restrict the parameters search ranges during its optimisation. In order to minimize the bounds of the electrical parameters, an identification strategy involving a search space reduction method (SSRM) is applied. The aim of this method has been analytically explained in chapter 3 and it is used to increase the accuracy and reliability of identification by reducing the search space during the algorithm operation. The idea is to reduce the search space for those parameters that converge quickly and thus reduce convergence time [Perry, 2006a].

In order to achieve this, a specified number of shorter initial runs is performed by the algorithm and the different results stored by the program; mean “ m_i ” and standard deviation “ σ_i ” are then estimated for each parameter to identify. Generally, the number of runs should be chosen as a compromise between the estimation accuracy of mean and standard deviation and the total optimisation time. At the end of these initial runs the search space is redefined for each parameter according to (5.5):

$$\text{Bounds parameter } i = m_i \pm W\sigma_i \quad (5.5)$$

where W , width of the window, defines how much the search space is reduced, but it has to ensure that the new bounds are not wider than the original limits. Practically it is found that a value of window width of about 4 gives good performance for all parameters. Five initial runs are performed in this case using a traditional genetic algorithm with elitist selection. Each GA run contains of 30 individuals and maximum 30 generations. The number of generations has been chosen empirically as it was noted that in this application and with this specific fitness function, over 30 generations the value of the error stabilizes and only seldom keeps decreasing further. In this case, it is assumed that genetic algorithms found the best (sub – optimal) solutions to the system as the error stays the same, and therefore, it will be probably wasting time if we set a bigger number of maximum generations.

From figure 5.8, it has been noticed that the biggest error in both the speed and the I_{sq} current response, appears during transients. During steady state condition, the error in both graphs is zero and so it would be more consistent to try to estimate the motor parameters during the transient period as less simulation time will be lead to less computational time. However, as a starting point, we will focus on the rated estimation of the electrical parameters and then the next step will be the improvement of these parameters or else a faster computational time.

As a first step, the inputs for the simulation model are the I_{sd}^* reference current and the reference speed ω_r^* (see Figure 5.2). These values will be the rated ones and so the motor will run with rated (full) flux ($I_{sd}^* = 4.9A$) and with $\omega_r^* = 250rad/sec$. The flux will be applied at simulation time of 0 sec while the speed reference at 2.3838sec. The PI speed and current controllers are exactly the same in both the experimental and simulation system and the rotor time constant is also the same and stable $T_r=0.15087$. The simulation (identification) time will start few milliseconds before the speed reference is applied and it will be from 2.2496 to 4.8628sec with a maximum step size of 0.2msec that covers both transient and steady state response.

The obtained results are shown in the below Table 5.3.

Electrical Parameters	1 st run	2 nd run	3 rd run	4 th run	5 th run
Rotor Resistance (R_r)	4.1111 Ω	4.1111 Ω	4.1375 Ω	4.1397 Ω	4.1488 Ω
Magnetizing Inductance (L_m)	0.5476 H	0.5462 H	0.5453 H	0.5453 H	0.5448 H
Leakage inductance of stator (L_{ls})	0.0568 H	0.0250 H	0.0429 H	0.0297 H	0.0297 H
Leakage inductance of rotor (L_{lr})	0.0557 H	0.0557 H	0.0558 H	0.0558 H	0.0558 H
Best Objective Value	0.9032	0.8794	0.8524	0.8511	0.8499

Table 5.3: Genetic Algorithm Estimated Motor Parameters in the Initial Runs

From the results it can be seen that all runs produce similar parameters estimations except for the stator inductance. The optimisation routine applies then the search space reduction method (SSRM) in order to estimate the new bounds for the parameters. From equation (5.5) it can be found that the new limits for the electrical parameters are given as in (5.6). The GAs optimisation settings selected are shown in Table 5.4.

$$\begin{aligned}
 4.0672 \Omega &\leq R_r \leq 4.1920 \Omega \\
 0.54188 \text{ H} &\leq L_m \leq 0.54979 \text{ H} \\
 0.01354 \text{ H} &\leq L_{ls} \leq 0.06010 \text{ H} \\
 0.05556 \text{ H} &\leq L_{lr} \leq 0.05595 \text{ H}
 \end{aligned}
 \tag{5.6}$$

GAs parameters				
	Species I	Species II	Species III	Species IV
Crossover rate	0.85	0.9	0.95	0.99
Mutation rate	0.15	0.1	0.05	0.009
Generation Gap	0.9			
Initial runs	Maximum Generations	30		
	Number of Individuals	30		
Final run	Maximum Generations	50		
	Number of Individuals	40		

Table 5.4: Genetic Algorithm Characteristics

5.4 Results

Finally, applying the last wider genetic algorithms optimisation including the new parameters bounds we obtain the final estimation of our I.M. electrical parameters shown in Table 5.5.

Electrical Parameters	
Rotor Resistance (R_r)	4.1636 Ω
Magnetizing Inductance (L_m)	0.5435 H
Leakage inductance of stator (L_{ls})	0.0291 H
Leakage inductance of rotor (L_{lr})	0.0556 H
Best Objective Value	0.84901

Table 5.5: Final Genetic Algorithm Estimated Motor Parameters

From Table 5.3 and 5.5 can be noted that the new SSRM method gives very good reliable identification results in a reasonable time. It can also be seen that the best objective value in Table 5.5 is smaller compared to the ones in Table 5.3 and so the identification accuracy has also been improved. I could also consider reducing the number of decimal points and using more realistic numbers as the 3rd or 4th decimal point is not influencing the accuracy of the results.

In the next sub – sections a comparison of the speed and I_{sq} current responses between the experimental induction motor drive performance and the Simulink model using the new parameters will give us a clear view of the accuracy of the estimated parameters.

5.4.1 Comparison of Speed Response

Figure 5.10 represents the speed response between the experimental and the Simulink modeled response with the new GA evaluated electrical parameters.

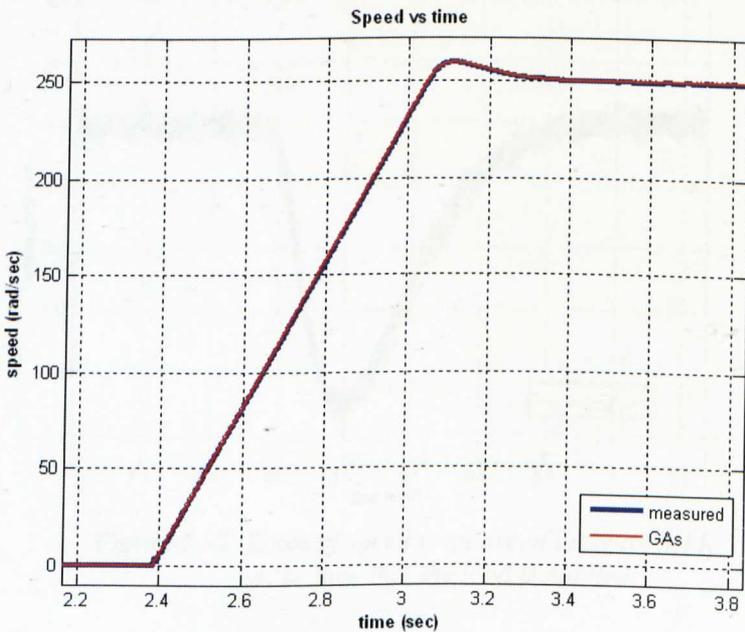


Figure 5.10: Measured and GAs (modelled) Speed responses with no load.

The same results, but in the case in which a 15.5 Nm load torque is applied at time $t = 6.5$ sec are shown in the below figures 5.11 and 5.12.

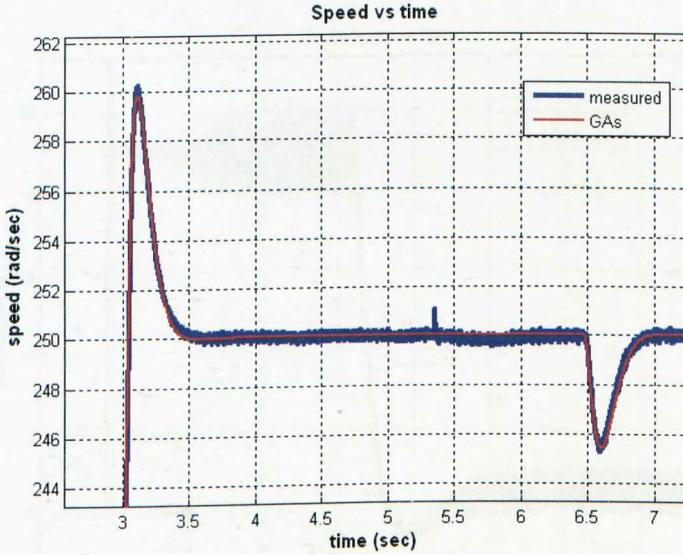


Figure 5.11: Measured and GAs (modelled) Speed responses with a load torque of 15.5Nm.

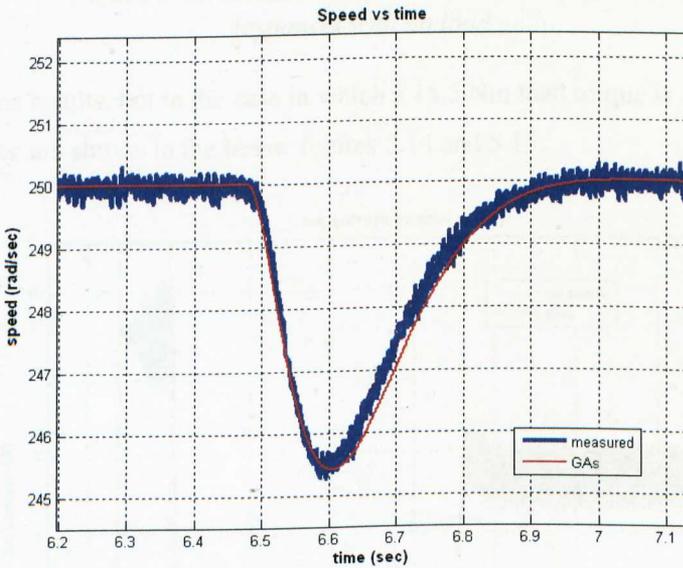


Figure 5.12: Zoom of speed response of in figure 5.11 at the time that the load is applied.

5.4.2 Comparison of I_{sq} Current Response

Figure 5.13 represents the I_{sq} current response between the experimental and the Simulink modeled response behavior with the new GA evaluated electrical parameters.

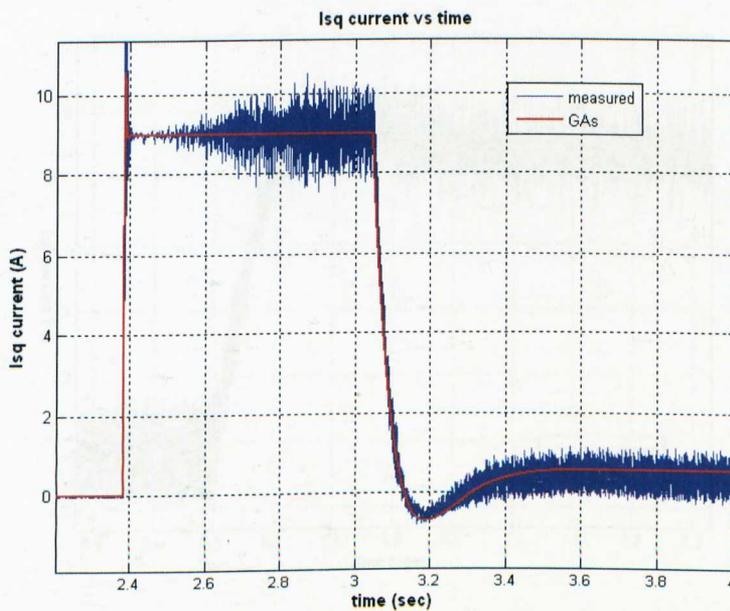


Figure 5.13: Measured and GAs (modelled) I_{sq} current responses with no load.

The same results, but in the case in which a 15.5 Nm load torque is applied at time $t = 6.5$ sec are shown in the below figures 5.14 and 5.15.

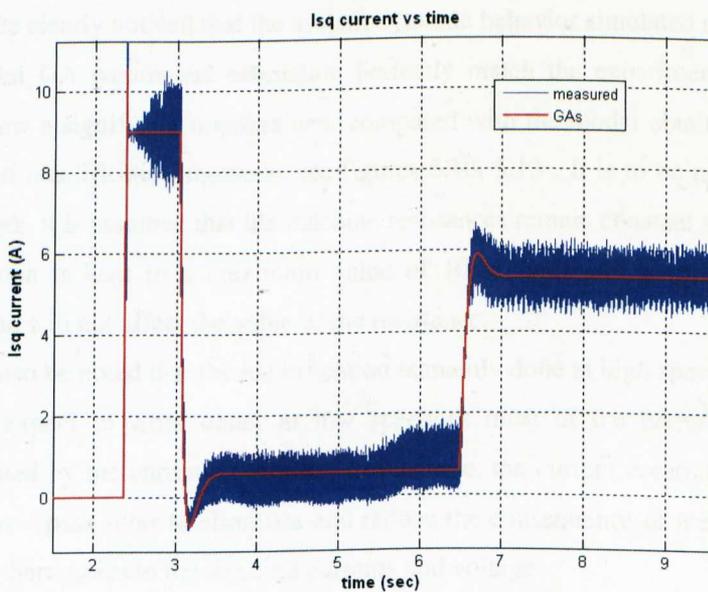


Figure 5.14: Measured and GAs (modelled) I_{sq} current responses with a load torque of 15.5Nm.

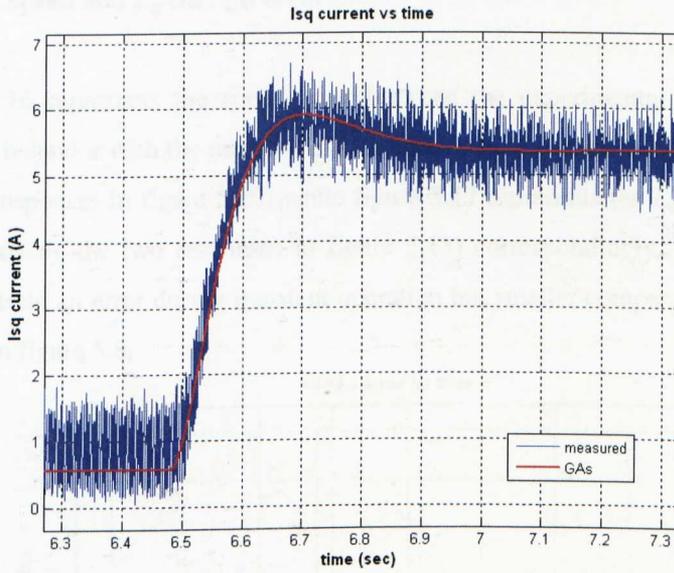


Figure 5.15: Zoom of I_{sq} current response of in figure 5.14 at the time that the load is applied.

5.4.3 Discuss and Analysis of the Results

It can be clearly noticed that the system dynamic behavior simulated using the new proposed GA parameters estimation perfectly match the experimental measures and show a significant improvement compared with the model obtained using the standard identification methods, see figures 5.10, 5.13 . It is to be noticed that in this work, it is assumed that the machine resistances remain constant since the tests simulation is kept to a maximum value of 10 seconds and so the temperature variation will not affect the value of the resistance.

It can also be noted that the identification is mainly done at high speed. However I would expect to work better at low speed as most of the harmonics will be eliminated by the current controller. In this case, the current controller will work as a low – pass filter to eliminate and reduce the consequence of the existence of the low harmonics in the winding currents and voltages.

5.4.3.1 Speed and I_{sq} current error

Figure 5.16 represents the speed error between the experimental and modeled response behavior with the new GA evaluated electrical parameters (error between the two responses in figure 5.10) while figure 5.17 represents the I_{sq} current error (error between the two responses in figure 5.13) correspondingly. These figures also illustrate an error during transient operation but smaller compare with the one showed in figure 5.8.

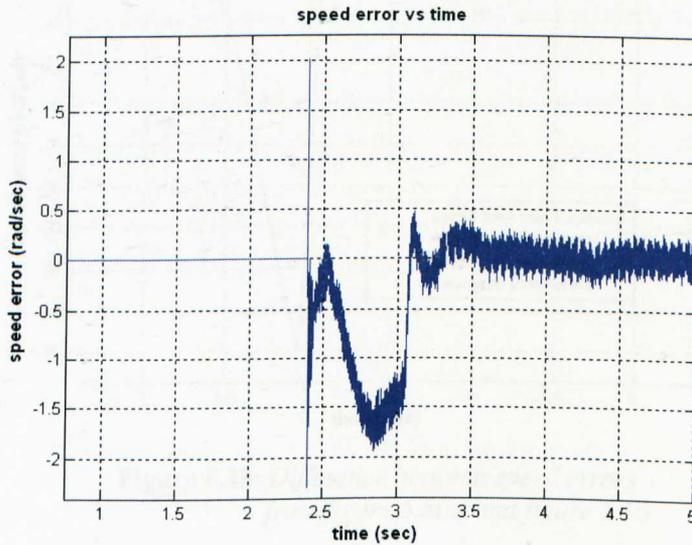


Figure 5.16: Speed error between measured and GAs speed response (error from figure 5.10)

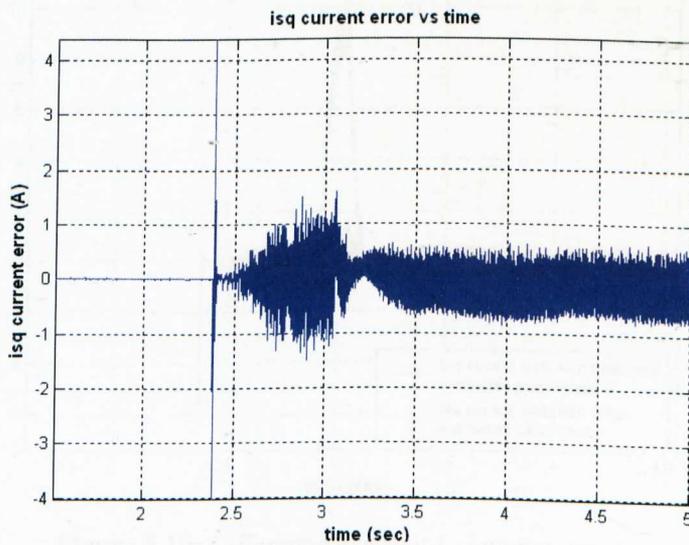


Figure 5.17: I_{sq} current error between measured and GAs I_{sq} current response (error from figure 5.13)

Figure 5.18 shows the difference between the speed errors from both figure 5.8(a) and figure 5.16. Figure 5.19 illustrates the difference between the I_{sq} current errors from both figure 5.8(b) and figure 5.17. It can be obviously seen that GAs give us reduced speed and I_{sq} current error compared with standard methods, which means more accurate and reliable identification of electrical parameters.

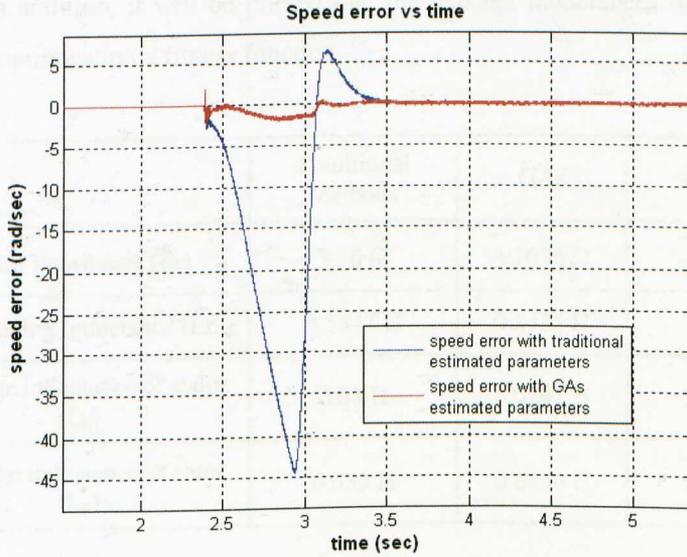


Figure 5.18: Difference between speed errors from figure 5.8(a) and figure 5.16

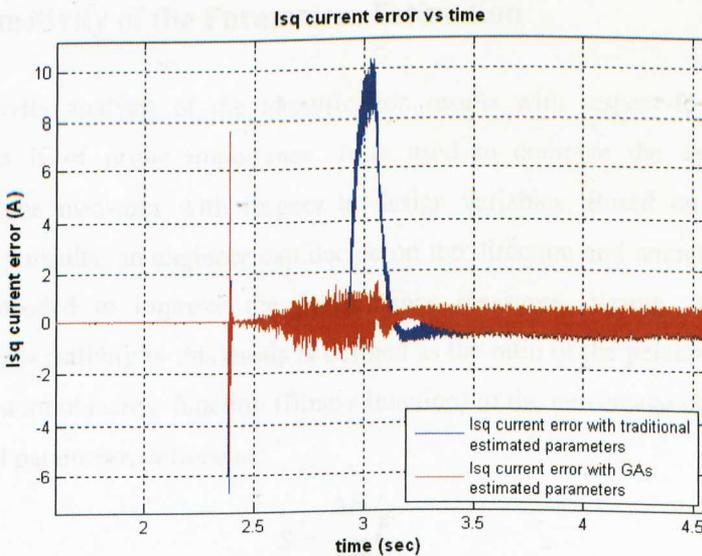


Figure 5.19: Difference between I_{sq} current errors from figure 5.8(b) and figure 5.17

Table 5.6 also shows the error introduced by standard estimation methods. From this table, it is noticeable that the smallest error appeared in the magnetising inductance (L_m) while the biggest one was introduced in the leakage rotor inductance (L_{lr}). However, the importance of the parameters' error and their influence on the speed and I_{sq} current response will be shown in a later sub – section. In addition, it will be proved that the leakage inductances do not affect much the optimisation's fitness function.

	Traditional Methods	GAs	%error
Rotor Resistance (Rr)	3.76 Ω	4.1636 Ω	10.7
Magnetizing Inductance (Lm)	0.5343 H	0.5435 H	1.72
Leakage inductance of stator (L_{ls})	0.04 H	0.0291 H	27.25
Leakage inductance of rotor (L_{lr})	0.033 H	0.0556 H	68.48

Table 5.6: Comparison between the new GA estimated motor parameters and the traditional ones

5.5 Sensitivity of the Parameters Estimation

A sensitivity analysis of the identification results with respect to parameters variations is of prime importance. It is used to compute the sensitivity of performance measures with respect to design variables. Based on the design sensitivity results, an engineer can decide on the direction and amount of design change needed to improve the performance measures [Kyung, 2004]. Each parameter sensitivity in this thesis is defined as the ratio of the percentage change in the system objective function (fitness function) to the percentage change of the estimated parameter, defined as:

$$S = \frac{\Delta F / F}{\Delta P / P} \quad (5.7)$$

where P is the estimated parameter, and F the fitness function of the system.

In the following sections the system sensitivity will be calculated for a small variation of the estimated parameters, keeping the motor parameters used in the controller unaltered. Then the influence of this alteration will be shown in the speed and I_{sq} current response.

5.5.1 Variation of Rotor Resistance

It is known that the rotor resistance variation is affected by change in rotor temperature. In case that the rotor resistance is increased by 10% of the GA estimated one ($R_r=4.1636\Omega$) then its new value will be $R_r'=4.57996\Omega=1.1\times R_r$. This theoretically small variation can change one of the most important vector control parameter and this is the rotor time constant T_r . The calculated slip frequency is also incorrect and the flux angle is no longer appropriate for field orientation. As a result of this, instantaneous errors emerge in both flux and torque.

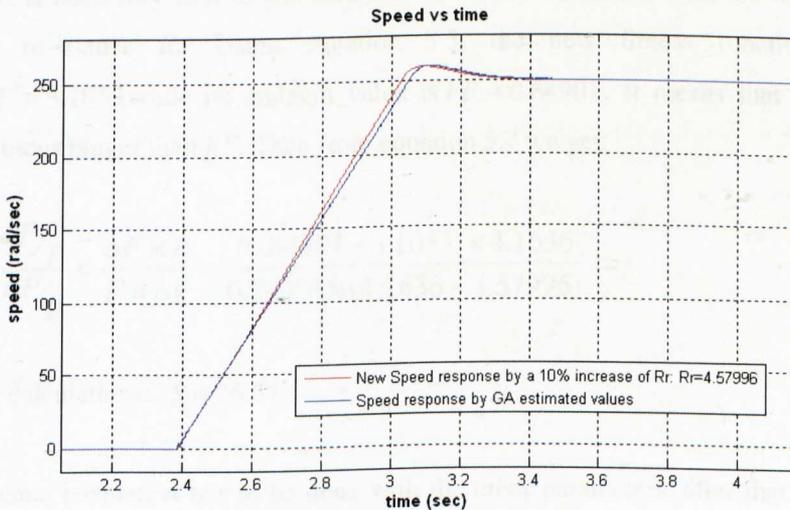


Figure 5.20: Difference between speed responses with 10% increase of R_r .

Figures 5.20 and 5.21 shows the effect of the 10% increase of R_r in the speed and I_{sq} current response correspondingly. From these figures, it can be seen that there

is a significant error during the transient response in both speed and current. During steady state condition, both the I_{sq} current and speed responses have not been affected from the change in the rotor temperature.

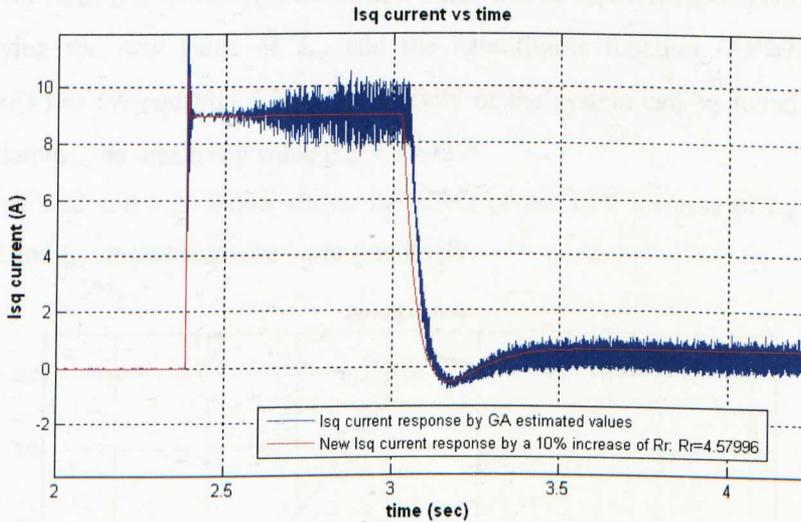


Figure 5.21: Difference between I_{sq} current responses with 10% increase of R_r

The sensitivity of this case can be calculated from equation 5.7. It is obvious then that it is necessary first to calculate the new fitness function with the increased rotor resistance R_r . Using equation 5.3, the new fitness function will be $FF' = 3.1053$ while its ambient value is $FF = 0.84901$. It means that FF' is 3.65 times bigger than FF . Then from equation 5.7 we get:

$$S = \frac{\Delta F / F}{\Delta P / P} = \frac{\Delta F \times P}{F \times \Delta P} = \frac{(0.84901 - 3.1053) \times 4.1636}{0.84901 \times (4.1636 - 4.57996)} \Rightarrow$$

After calculations: $S = 26.57$

The same evaluation has to be done with the other parameters, after that we can analyze which of them are more sensitive and which ones does not affect too much the vector control of the motor.

5.5.2 Variation of Magnetising Inductance (L_m)

On the assumption that the magnetising inductance L_m increases by 10% of its ambient value ($L_m=0.5435\text{H}$) then its new value will be $L_m'=1.1\times L_m=0.59785\text{H}$.

Applying the new value of L_m' and the new fitness function ($FF\hat{=}7.1674=8.44\times FF$) to the equation 5.7, the sensitivity of the system can be found. After calculations, the sensitivity value is $S = 74.42$.

Figures 5.22 and 5.23 below shows the effect of the 10% increase of L_m in the speed and I_{sq} current response correspondingly.

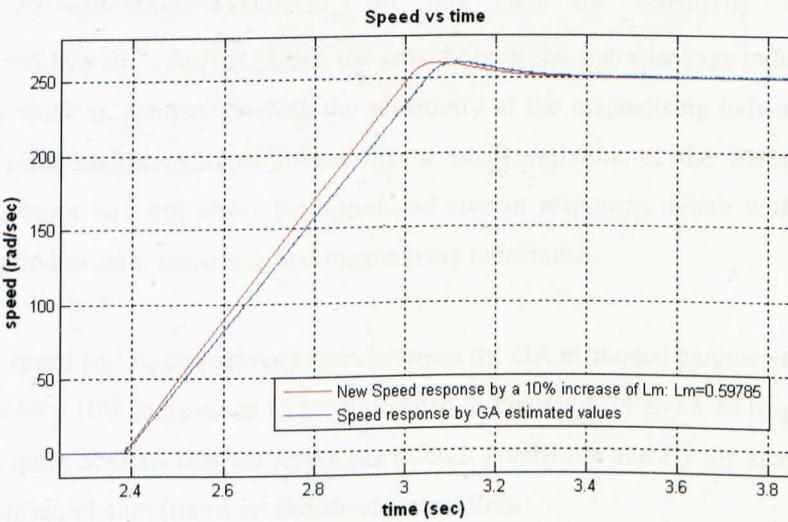


Figure 5.22: Difference between speed responses with 10% increase of L_m

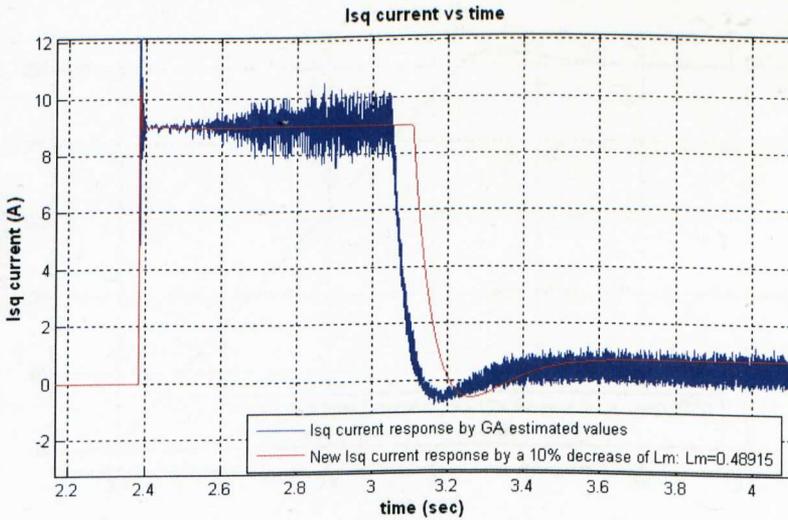


Figure 5.23: Difference between I_{sq} current responses with 10% increase of L_m

From figures 5.22 and 5.23, it can be noticed that there is a considerable error during the transient response in both speed and I_{sq} current while during the steady state condition, both the I_{sq} current and speed responses have a zero error.

5.5.3 Variation of Stator Leakage Inductance (L_{ls})

If the value of the stator leakage inductance is increased by 10%, then its new value will be $L'_{ls}=0.03201\text{H}$ and the new fitness function will be slightly increased by $FF \hat{=} 0.84906 = 1.000058 \times FF$. In this case the sensitivity is equal to $S = 5.89 \times 10^{-4}$. At first glance the sensitivity of the stator leakage inductance is very small in comparison with the sensitivity of the magnetising inductance and the rotor resistance. This means that a small variation of the stator leakage inductance will not affect the speed and current responses unlike with a small variation of rotor resistance and magnetising inductance.

The speed and I_{sq} current responses between the GA estimated parameters and the ones by a 10% increase of L_{ls} are illustrated in Figures 5.24 and 5.25 respectively. It is quite obvious that the responses in both graphs are exactly the same for the whole simulation (transient and steady state) time.

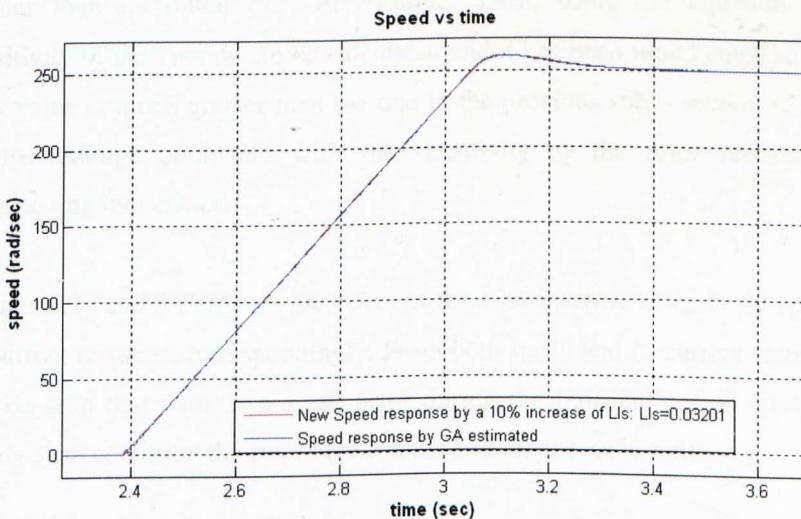


Figure 5.24: Difference between speed responses with 10% increase of L_{ls}

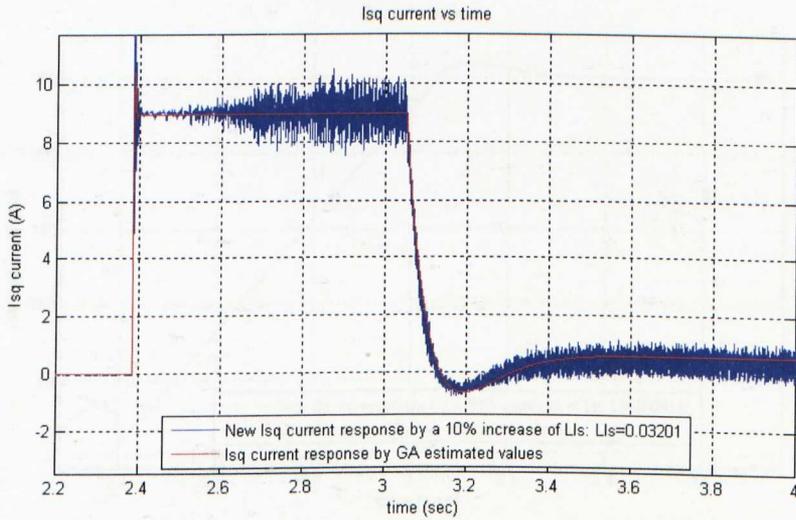


Figure 5.25: Difference between I_{sq} current responses with 10% increase of L_{ls}

5.5.4 Variation of Rotor Leakage Inductance (L_{lr})

While the stator leakage inductance presented an insignificant sensitivity to our system, it is quite interesting to see how the system behaves within the same variation of a 10% increase of the rotor leakage inductance. In this case the new value of L_{lr} will be 0.06116H and the new fitness function will be 1.96 times greater than its initial one, $FF \approx 1.6662$. Then, using the equation 5.7 the sensitivity of the system can be calculated and it has been found equal to $S=9.62$. This value is much greater than the one in the previous sub – section (5.5.3) but smaller enough compared with the sensitivity of the rotor resistance and magnetising inductance.

Figures 5.26 and 5.27 shows the effect of the 10% increase of L_{lr} in the speed and I_{sq} current response correspondingly. From both speed and I_{sq} current responses, it can be seen that there is a small error during the transient period while in the steady state condition the error between the two responses is zero.

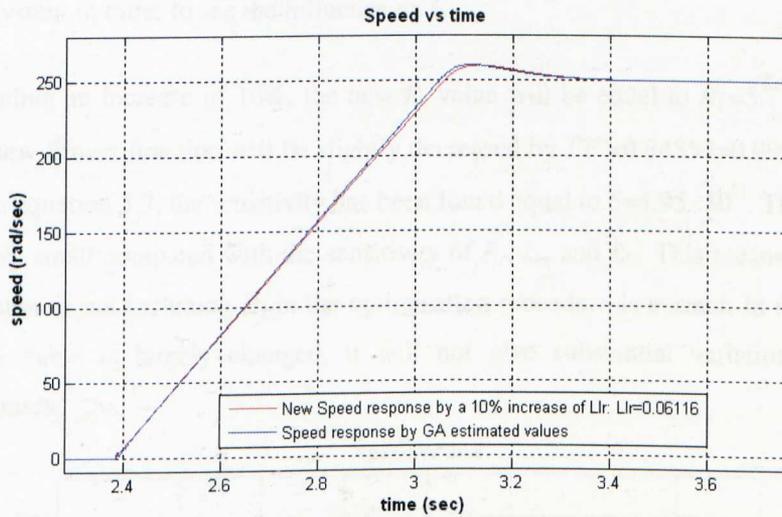


Figure 5.26: Difference between speed responses with 10% increase of L_r

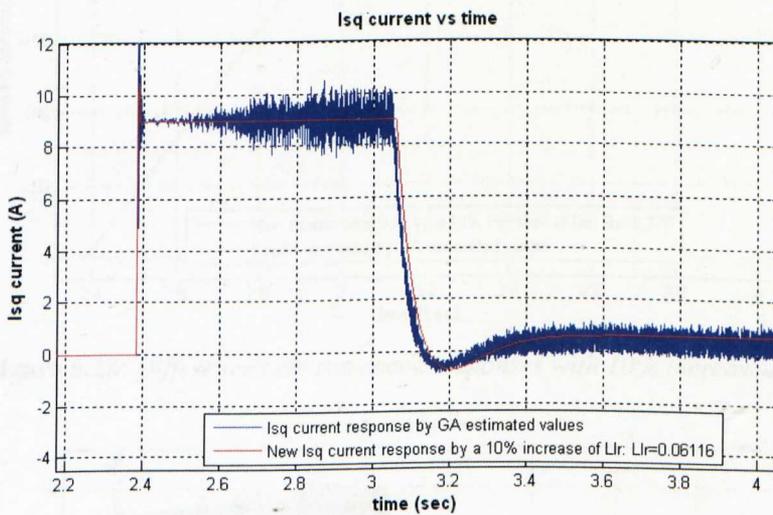


Figure 5.27: Difference between I_{sq} current responses with 10% increase of L_r

5.5.5 Variation of Stator Resistance (R_s)

The stator resistance of an induction motor is easy to measure (see Appendix C), and so in this thesis it will not be included in the GA electrical estimated parameters. This will reduce the electrical parameters to four (4) and so the GA will have less computational burden and it will give more accurate results.

However it is necessary the knowledge of the sensitivity of the stator resistance to the system in order to see the influence on it.

Including an increase of 10%, the new R_s value will be equal to $R_s' = 5.775\Omega$ and the new fitness function will be slightly decreased by $FF' = 0.84859 = 0.99950 \times FF$. From equation 5.7, the sensitivity has been found equal to $S = 4.95 \times 10^{-3}$. This value is very small compared with the sensitivity of R_r , L_m and L_{lr} . This means that the decision of not including R_s in the optimisation procedure is correct. In fact even if its value is largely changed, it will not give substantial variation in the responses.

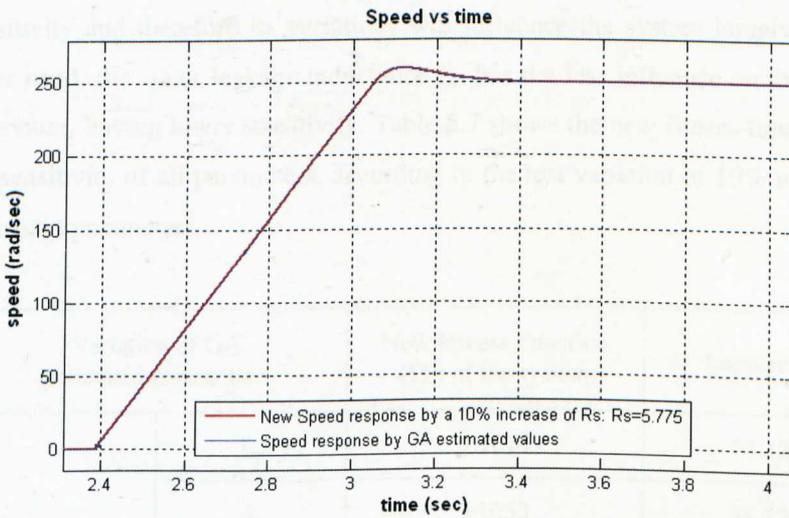


Figure 5.28: Difference between speed responses with 10% increase of R_s

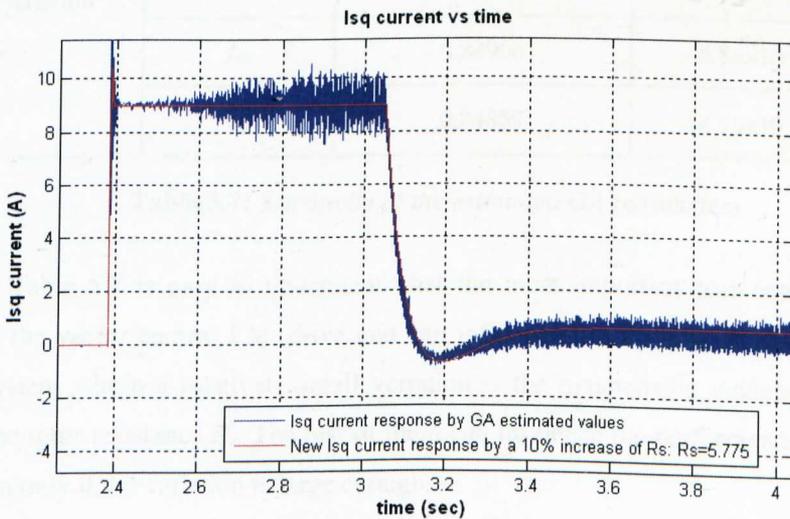


Figure 5.29: Difference between I_{sq} current responses with 10% increase of R_s

The effect of the 10% increase of R_s in the speed and I_{sq} current response are illustrated in Figures 5.28 and 5.29 respectively. It is very clear that in both figures, the error between the responses is zero for the whole simulation (transient and steady state period) time. This supports the decision of not including R_s in the optimisation procedure.

5.5.6 Discussion and Analysis of the Parameters' Sensitivity

Of all the parameters, the magnetising inductance L_m is found to have the highest sensitivity and therefore its variations will influence the system largely. On the other hand, the stator leakage inductance L_{ls} has the less influence on the system behaviour, having lower sensitivity. Table 5.7 shows the new fitness function and the sensitivity of all parameters, according to the test variation of 10% in the GA estimated parameters.

Variation of GA estimated parameters		New Fitness Function (FF) of the system	Sensitivity
10% variation	L_m	7.1674	74.42
	R_r	3.1053	26.57
	L_{lr}	1.6662	9.62
	L_{ls}	0.84906	5.89×10^{-4}
	R_s	0.84859	4.95×10^{-3}

Table 5.7: Sensitivity of the estimated GA parameters

From Table 5.7 is easy to understand that the most important parameters that affect the vector control I.M. drive and can influence the dynamic behaviour of the system within a relatively small variation is the magnetising inductance L_m and the rotor resistance R_r . The rest of them can influence the performance of the system only if the variation is large enough.

The sensitivity analysis also demonstrates that it is worth running the GA optimisation only for R_r and L_m . In this particular motor L_m is much larger than L_{lr} and then it will not affect relevantly the rotor time constant. This is classified by the following test. The whole idea is to find for the same fitness variation of function, the percentage of the variation of each of the electrical parameters. For example, from Table 5.7, let us use the first value of the fitness function which corresponds to an alteration of 10% increase of L_m . It is possible to calculate that to obtain the same value of FF , R_r has to be increased by 35% and L_{lr} by 65% respectively. For the other two parameters, L_{ls} and R_s , we need a tremendous change of the rated estimated GA value to achieve such a FF variation.

5.6 Accurate knowledge of moment of Inertia J

It was noted that one of the limitations of this work is that the use of GAs for the identification of electrical parameters, requires accurate knowledge of the mechanical parameter moment of inertia J . The first reason of this conclusion is that GAs gives very different results from the estimated ones when trying to identify both electrical and mechanical parameters. It was also noted that the estimated parameters changed significantly when the value of inertia was increased or decreased by 10% from its true value. So, an increase of the inertia J by 10% caused an increase of the magnetizing inductance L_m by 10.05% and an increase of J by 20% caused an increase of L_m by 18.6%. This proves that the mechanical parameters are very influential for the accurate estimation of the electrical parameters through GAs.

A potential solution to this limitation of the Genetic Algorithms is probably the further modification of the Fitness Function (FF). In other words, a more analytical investigation of the fitness function including other parameters (which will be affected by the mechanical parameters) could solve this problem.

There are a couple of methods to accurately estimate the mechanical parameter. One method which is used to calculate the inertia without knowing any of the electrical parameters is presented in Appendix F.

5.7 Parameter Identification under Poorly Tuned Vector Control System

In this PhD thesis, the identification of the IM parameters was successfully achieved while all the experimental data (current, speed and voltage) were taken under a well tuned vector controller. However, it would be interesting to determine whether the GA can still accurately identify the parameters when the experimental data come from a drive with poorly designed control parameters. One way to do that is to change the rotor time constant in the vector controller.

Therefore, increasing the rotor time constant by 20% of the value which has been found using the estimated parameters of Table 5.6 ($T_r=0.1438$), the increased value will be equal to $T_r=0.1726$. The new estimated parameters under rated operating conditions will then be:

$$R_r=4.1264\Omega$$

$$L_m=0.5257H$$

A comparison of the above values with the ones from Table 5.6 for rated operating conditions has to be made. Results show that both the resistance of the rotor and the magnetizing inductance are almost the same. The rotor time constant based on the new estimated values is equal to $T_r=0.141$. This value is almost the same with the one for the well tuned vector controller. This means that GAs is robust and gives us accurate parameters in the case that the experimental data are taken under a detuned vector controller.

5.8 Average L_m and R_r for Varying Flux Levels

Given the nature of the optimisation and of the general transient measurements used, the parameters estimation performed has identified a sort of average parameters set. It would be interesting to be able to identify machine parameters variations in function of some relevant quantities. A good exercise to verify this

possibility is to apply the same methodology for example to estimate the parameters under different flux levels. The results are a set of electrical parameters for each specific operating condition, from which it is possible to extrapolate their behavior in function of the flux level.

Figure 5.30 shows the behavior of the magnetising inductance L_m in function of the I_{sd}^* (flux level); as it can be seen the value of L_m decreases as the flux increases and in low flux level (below 50%) the value of L_m is almost constant as it is expected. This fact has also been previously experienced by Gerada et al. (2003) and Sumner et al. (1993) for the same motor. This confirms the validity of the proposed identification approach.

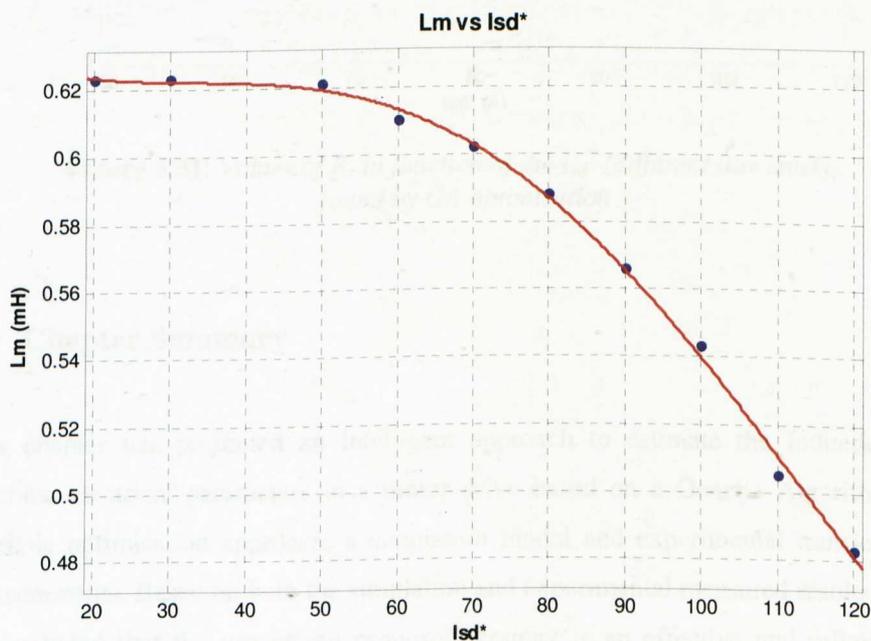


Figure 5.30: Values of L_m in function of the I_{sd}^* (different flux levels) found by GA optimisation

Figure 5.31 shows the behaviour of rotor resistance R_r in function of the motor d -axes current I_{sd}^* (flux level); it can be noticed that the value of R_r changes slightly with different flux levels and this is due to either numerical errors or variation of rotor temperature while the pilot experimental measures were taken.

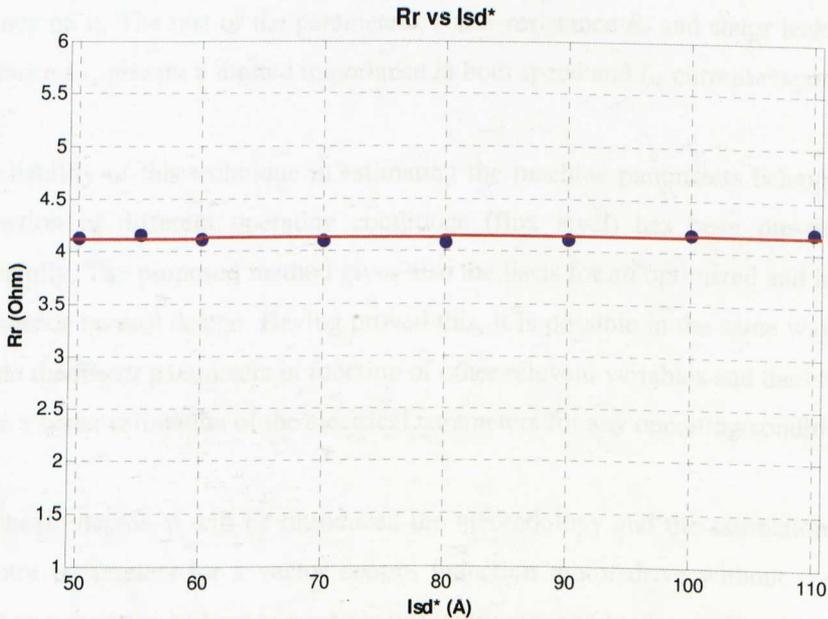


Figure 5.31: Values of R_r in function of the I_{sd}^* (different flux levels) found by GA optimisation

5.9 Chapter Summary

This chapter has presented an intelligent approach to estimate the Induction Machine electrical parameters in a motor drive based on a Genetic Algorithm heuristic optimisation approach, a simulation model and experimental transient measurements. Based on both the simulation and experimental measured results it is concluded that the use of the proposed strategy is an effective and reliable method for induction motor parameter identification and for accurately modelling the behaviour of the drive system. The total processing time for the PC was used (conventional PC) was around 2 ½ hours.

The sensitivity of the GA estimated parameters were effectively used so as to find the influence of each parameter in the optimisation's fitness function. Results were also verified through comparison of speed and I_{sq} current response within a variation of 10% variation in each parameter. It was found that the magnetising

inductance L_m and the rotor resistance R_r appeared to have a large sensitivity to our GA optimisation while the rotor leakage inductance L_{lr} has a considerable influence on it. The rest of the parameters, stator resistance R_s and stator leakage inductance L_{ls} , present a limited importance in both speed and I_{sq} current response.

The reliability of this technique in estimating the machine parameters behaviour in function of different operating conditions (flux level) has been presented successfully. The proposed method gives also the basis for an optimized and high performance control design. Having proved this, it is possible in the same way to estimate the motor parameters in function of other relevant variables and therefore to have a better estimation of the electrical parameters for any operating condition.

In the next chapter, it will be introduced the methodology and the estimation of the motor parameters for a vector control induction motor drive without speed control as a function of load current for different magnetic loadings . Results will give us important information about the values of the electrical parameters for different working condition.

Chapter 6

Electrical Parameter Identification for Varying Load and Flux Levels

6.1 Introduction

This chapter introduces a new non – intrusive approach for identifying induction motor equivalent circuit parameters in function of varying load and flux levels and based on heuristic optimisation. This strategy for parameter identification, just need transient measurements of speed, current and voltage and can be applied to motors already in service, like for example in a drive. In our case we use a vector controlled Induction Motor (I.M.) drive and we also use an off line Genetic Algorithm (GA) routine to tune a linear machine model.

The electrical parameters assumed to be known are the stator resistor, and the stator and rotor leakage inductances. As already mentioned in the previous chapter, the first one is easy to be measured, while the leakage inductances found

6.2 Description of the Updated System

Figure 6.1 displays the optimisation set – up of the genetic algorithm system used in this chapter. A comparison between the figures 5.1 and 6.1 shows the different fitness functions of the two examined cases. This last case includes an additional measurement of experimental transient response of the reference amplitude V_s voltage (see section 6.2.1 for more details) from the controlled electrical drive test rig which is also stored and used as reference signal for the optimisation. The V_s was not actually calibrated against the actual measured voltage but the voltage transducers have been calibrated before taking the experimental results. Equation 6.1 presents the new fitness function (FF) which becomes equal to:

$$FF = \int (W_v \cdot |V_s \text{ Voltage error}| + W_s \cdot |\text{speed error}| + W_I \cdot |I_{sq} \text{ current error}|) \quad (6.1)$$

where W_v , W_s , W_I , are the weights of voltage, speed and current errors correspondingly. The fitness function is displayed as the sum of 3 weighted vectors (errors) which is the same as using normalized error. The purpose of using these weights is to give to some elements in the sum more “weight” or influence on the resulting fitness function. The values of these weights will be found based on rated I_{sq}^* and I_{sd}^* reference current responses and they will be fixed in each different operating condition. In order to design the values of these weights an initial comparison of the voltage, speed and current error is necessary. In this way, the largest error will always give a weight of ‘1’ for simplification, while the other two weights will be equal to the ratio between the largest error over its corresponding one. For example, in case that the voltage error is the largest one then:

$$\begin{aligned} \text{voltage weight : } W_v &= 1 \\ \text{speed weight : } W_s &= \frac{\text{voltage error}}{\text{speed error}} \\ \text{current weight : } W_I &= \frac{\text{voltage error}}{\text{current error}} \end{aligned} \quad (6.2)$$

To set the values of these weights before starting the optimisation, it is essential to find at least an approximate estimation of these element errors. However, how this can be done as the electrical parameters of the simulation model are unknown?

The idea is to use the electrical parameters found in the 5th chapter in the simulation model. Then each error can be calculated as the sum of the absolute value between experimental and simulation model response as it is shown in equation 6.3.

$$error = \text{sum}(\text{abs}(\text{experimental response} - \text{simulation model response})) \quad (6.3)$$

As a result of this, applying equation 6.2, it is easy to estimate the weights for rated flux and torque producing currents (I_{sd}^* , I_{sq}^*). For this operating condition, it was found that the voltage weight W_V is '1' as the voltage error is the largest one in the vector controlled induction motor drive. The rest of the weights are: $W_S = 15$ and $W_I = 200$. This also means that the current error of the absolute value between experimental and simulation model response is the smallest one. Applying these values to equation 6.1, the final fitness function is displayed as:

$$FF = \int (|V_s \text{ Voltage error}| + 15 \cdot |\text{speed error}| + 200 \cdot |I_{sq} \text{ current error}|) \quad (6.4)$$

Therefore, a comparable contribution of all elements to the fitness function will be achieved.

6.2.1 Fitness Function

The new fitness function has been introduced in the above equation 6.4. It contains one more element from the one presented in the 5th Chapter (equation 5.3). This new element is the error of the reference voltage amplitude V_S calculated between experimental and simulation model responses. Generally, the reference voltage amplitude V_S for the PWM modulation is equal to:

$$V_S = \sqrt{V_{sq}^2 + V_{sd}^2} \quad (6.5)$$

where V_{sq} and V_{sd} are the output voltages of the PI current controllers from the vector control diagram shown in the following figure 6.2.

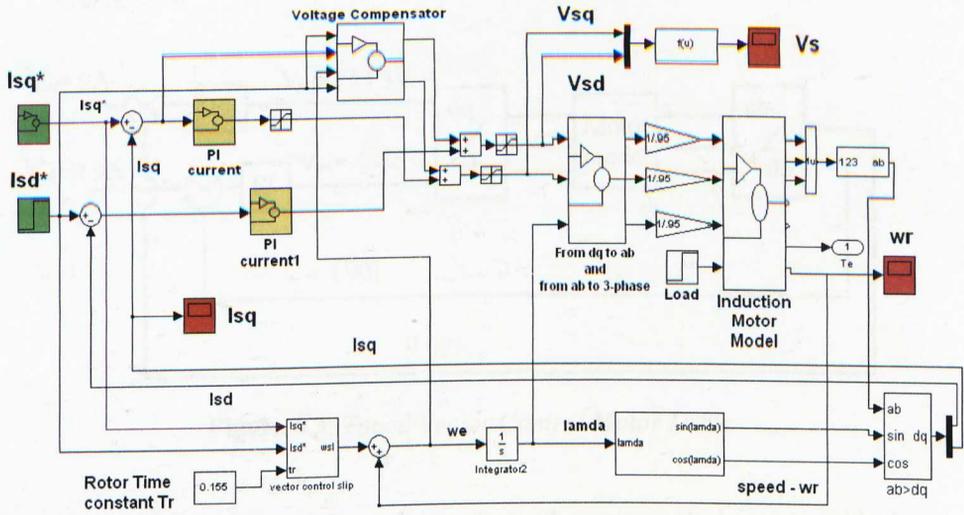


Figure 6.2: Vector Control Motor Drive without speed control.

However, in the experimental rig, it is not feasible to measure the voltages V_{sq} and V_{sd} . The only voltages that can be measured through voltage transducers are the V_a , V_b , and V_c , input voltages for the induction motor. Through the 3-phase to dq transformation, both V_{sq} and V_{sd} can then be calculated, stored and used as reference signal for the genetic algorithm's fitness function. These signals are the output of the PWM inverter and therefore have high frequency harmonics. On the other hand, the simulation model does not include the PWM as the simulation time will be too long and the GA optimisation will never end. However, the average behaviour will be the same in both experimental and simulation model.

The main reason for using this extra element in the fitness function (voltage error V_s) can be explained through an example which is shown in figures 6.3 and 6.4. Figure 6.3 represents a tuned vector control load drive system in which the inputs are the I_{sq}^* and I_{sd}^* reference currents. Typically PI controllers are used to control these currents to their reference values. The outputs of the PI controllers are the $d-q$ components of the voltage reference vector for the stator. These are the V_{sd} and V_{sq} voltages which are used in equation 6.3. The values of these voltages are calculated based on the rated I_{sq}^* and I_{sd}^* reference currents. Since the system is tuned the orientation considered of being ideal.

Tuned system

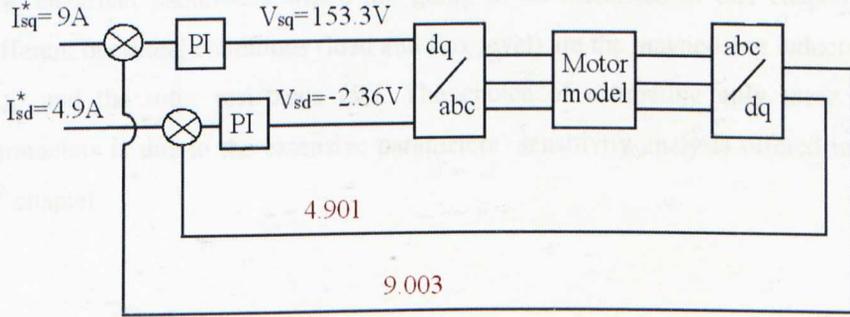


Figure 6.3: Tuned Vector Control Motor Drive.

On the other hand, figure 6.4 presents a detuned vector control system with the same inputs as in figure 6.3. In order to detune the system the rotor time constant was increased by 30%. The three – phase output currents from the motor model are determined as the measured ones and they are compared (after transformation from 3 – phase to dq) with the input reference currents (desired ones). Analysing both figures 6.3 and 6.4, it is noticeable that even if the measured and input reference currents are the same, the V_{sq} and V_{sd} voltages are different. This means that these voltages can give us significant information about the variation of the parameters while the Field Oriented Control is detuned. So, this is a good reason of including these voltages as an extra element in the fitness function.

Detuned system

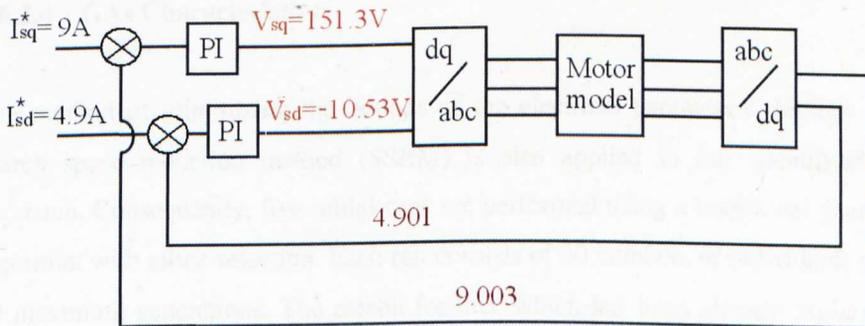


Figure 6.4: Detuned Vector Control Motor Drive.

6.2.2 Identified Parameters

The electrical parameters which are going to be identified in this chapter for different operating conditions (load and flux level) are the magnetising inductance (L_m) and the rotor resistance (R_r). The choice of estimating only these two parameters is due to the extensive parameters' sensitivity analysis offered in the 5th chapter.

6.2.3 Simulation Model Characteristics

It was previously mentioned that the inputs for the simulation model are the I_{sd}^* and I_{sq}^* reference currents. The flux (I_{sd}^*) will always be applied at simulation time of 0 sec while the I_{sq}^* reference step current will be imposed when the flux is settled down. The PI speed and current controllers are the same in both experimental and simulation system and their values are the ones given in the 2nd chapter. The rotor time constant inside the vector control algorithm will be the same at all times in all operating conditions and its value is the one calculated from the standard parameters ($T_r=0.15087$). Finally, the simulation (identification) time will start a few milliseconds before the I_{sq}^* reference current is applied to the system and will cover only the transient response.

6.2.4 GAs Characteristics

The method of minimizing the bounds of the electrical parameters through the search space reduction method (SSRM) is also applied in this identification approach. Consequently, five initial runs are performed using a traditional genetic algorithm with elitist selection. Each run consists of 30 numbers of individuals and 30 maximum generations. The reason for this, which has been already explained previously, is that beyond 30 maximum generations, the value of the fitness function's error decreases with a very slow rate, showing an affect of "convergence".

In this case the GAs optimisation settings selected are exactly the same of the ones presented in the previous Chapter (see Table 5.3).

6.3 Results – Comparison

Before presenting the results, it is worth to mention the different sets of operating conditions which will be used in the GAs optimisation set up. The I_{sd}^* current response (which represents the flux level) will consist of 4 different sets (5.88A, 4.9A = rated value, 3.92A and 2.94A), while the I_{sq}^* current response will consist of 5 different sets (10.5A, 9A = rated value, 7.5A, 6A and 4.5A). The identification of the parameters for these 20 different operating conditions will give us essential information about the exact motor parameters in a wide range and therefore an accurate design of the PI controllers and the rotor time constant T_r will be possible.

As a starting point, these values will be set to the rated ones and so the motor will run with a rated flux level ($I_{sd}^* = 4.9A$) and with $I_{sq}^* = 9A$. Therefore:

$$I_{sd}^* = 4.9A \text{ and } I_{sq}^* = 9A$$

The obtained results are shown in the below Table 6.1.

Identified Motor Parameters through GAs					
Electrical Parameters	1 st run	2 nd run	3 rd run	4 th run	5 th run
Rotor Resistance (R_r)	4.2013	4.2051	4.2054	4.2052	4.2062 Ω
Magnetizing Inductance (L_m)	0.5358	0.5329	0.5321	0.5319	0.5319 H
Best Objective Value	2.5869	2.5753	2.5716	2.5713	2.5713

Table 6.1: GA Estimated Motor Parameters for $I_{sd}^* = 4.9A$ and $I_{sq}^* = 9A$

Table 6.2 shows the final results after the Search Space Reduction Method (SSRM) is applied and including the new parameter bounds.

<i>Final Parameters</i>	
Rotor Resistance (R_r)	4.1984 Ω
Magnetizing Inductance (L_m)	0.5313 H
Best Objective Value	2.5711

Table 6.2: Final GA Estimated Motor Parameters for $I_{sd}^* = 4.9A$ and $I_{sq}^* = 9A$

Figure 6.5 represents the I_{sq} experimental current response together with the one simulated the new GAs evaluated electrical parameters and the one obtained with the GAs electrical parameters from the 5th chapter.

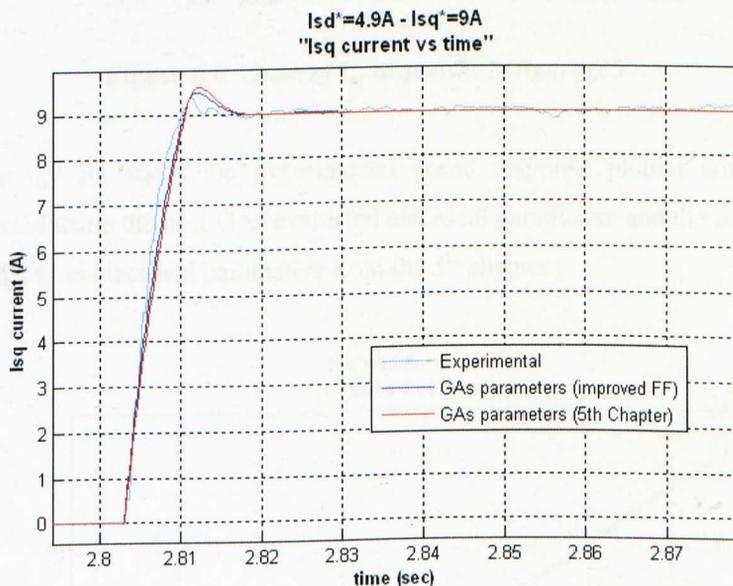


Figure 6.5: Experimental, new GAs and GAs (5th Chapter) I_{sq} responses for $I_{sd}^* = 4.9A$ and $I_{sq}^* = 9A$

From figures 6.5 and 6.6, it can be seen the I_{sq} current response behaviour with the new GA evaluated electrical parameters present a slightly better match of the experimental curve. This indicates that the GAs with the new fitness function, give accurate and reliable identified parameters which are a closer representation of the real ones for this particular operating condition.

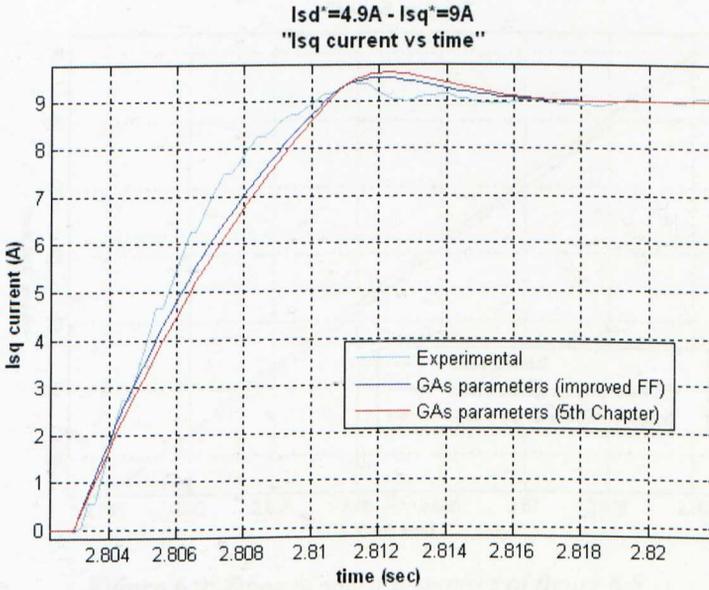


Figure 6.6: Zoom of I_{sq} responses of figure 6.5

Figure 6.7 illustrates the experimental speed response plotted with the one simulated using the new GAs evaluated electrical parameters and the one obtained with the GAs electrical parameters from the 5th chapter

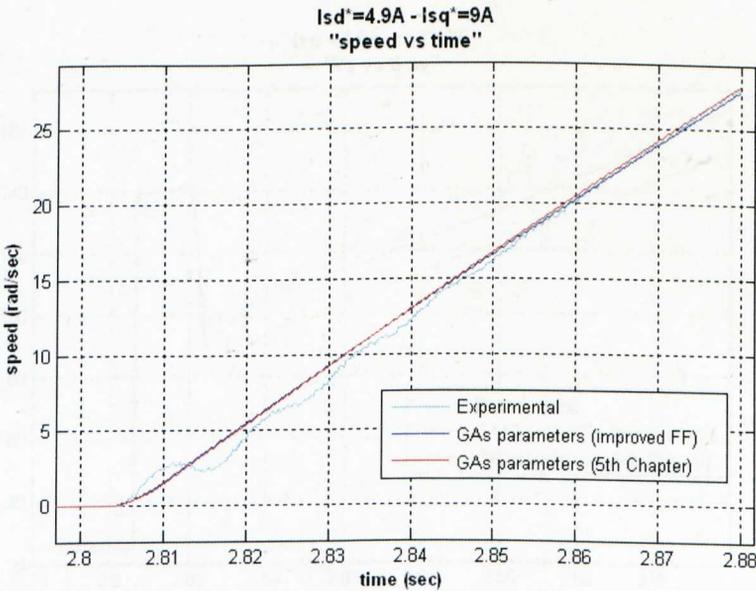


Figure 6.7: Experimental, new GAs and GAs (5th Chapter) speed responses for $I_{sd}^* = 4.9A$ and $I_{sq}^* = 9A$

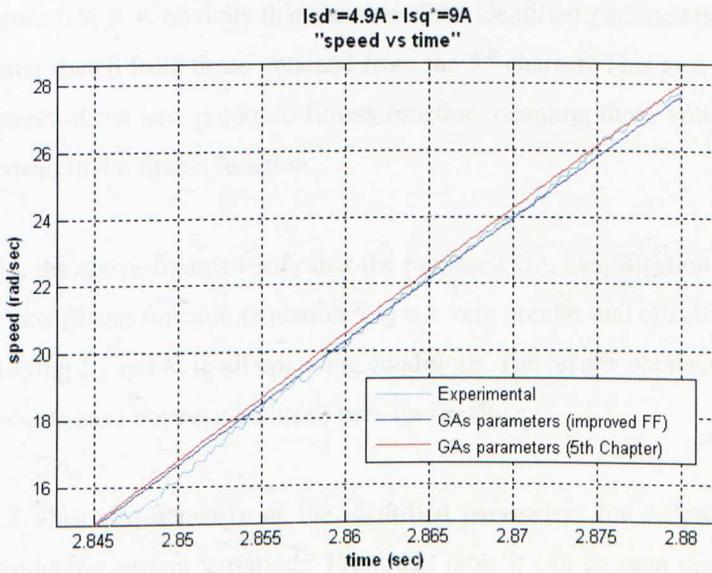


Figure 6.8: Zoom of speed responses of figure 6.5

Figures 6.7 and 6.8, shows that the simulated speed responses using the GA identified parameters (5th Chapter) and the ones with the improved FF, present both a really good match of the experimental curve. This shows the validity of the two applied GA identification techniques.

Finally, figure 6.9 shows the V_s voltage responses in the same cases.

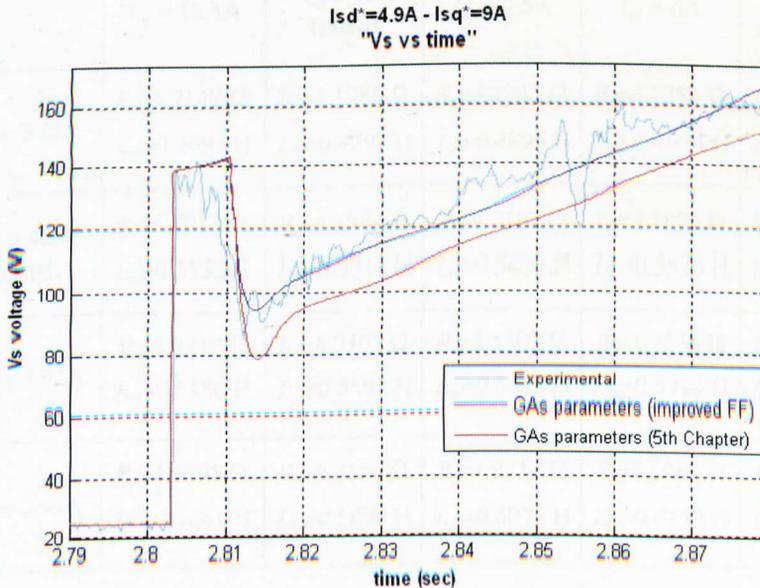


Figure 6.9: Experimental, new GAs and GAs (5th Chapter) V_s voltage responses for $I_{sd}^* = 4.9A$ and $I_{sq}^* = 9A$

From figure 6.9, it is obvious that the new GAs identified parameters present a much better match from those obtained from the 5th chapter. This also shows the effectiveness of the new proposed fitness function, counting these voltages as an extra element in the fitness function.

Generally, the above figures verify that the proposed GA identification technique with the new fitness function (equation 6.1) is a very precise and effective method for identifying L_m and R_r in all operating conditions. The results obtained in all the other 19 cases have not been included here for brevity.

Table 6.3 illustrates a matrix of the identified parameters for different field – torque producing current variations. From this table it can be seen that the rotor resistance is almost constant for all different conditions as it is expected. It can be also observed that this value is approximately the same to the one identified in the previous chapter and this difference can be due to numerical errors or variation of rotor temperature or due to the improved fitness function. Analytical explanation about the behaviour of the magnetising inductance will be displayed in the next sub – section.

	$I_{sq} = 10.5A$	$I_{sq} = 9A$ (rated)	$I_{sq} = 7.5A$	$I_{sq} = 6A$	$I_{sq} = 4.5A$
$I_{sd} = 5.88A$	$R_r=4.2147 \Omega$ $L_m=0.4694 H$	$R_r=4.1986 \Omega$ $L_m=0.4797 H$	$R_r=4.1938 \Omega$ $L_m=0.4895 H$	$R_r=4.2254 \Omega$ $L_m=0.4954 H$	$R_r=4.2078 \Omega$ $L_m=0.5070 H$
$I_{sd} = 4.9A$ (rated)	$R_r=4.2012 \Omega$ $L_m=0.5152 H$	$R_r=4.1984 \Omega$ $L_m=0.5313 H$	$R_r=4.2107 \Omega$ $L_m=0.5400 H$	$R_r=4.1896 \Omega$ $L_m=0.5526 H$	$R_r=4.1832 \Omega$ $L_m=0.5726 H$
$I_{sd} = 3.92A$	$R_r=4.2215 \Omega$ $L_m=0.5486 H$	$R_r=4.2201 \Omega$ $L_m=0.5693 H$	$R_r=4.2302 \Omega$ $L_m=0.5818 H$	$R_r=4.2156 \Omega$ $L_m=0.5996 H$	$R_r=4.2237 \Omega$ $L_m=0.6186 H$
$I_{sd} = 2.94A$	$R_r=4.1998 \Omega$ $L_m=0.5681 H$	$R_r=4.2156 \Omega$ $L_m=0.5890 H$	$R_r=4.2214 \Omega$ $L_m=0.6071 H$	$R_r=4.2064 \Omega$ $L_m=0.6272 H$	$R_r=4.1925 \Omega$ $L_m=0.6412 H$

Table 6.3: Final GA Estimated Motor Parameters for all operating conditions

6.3.1 Variation of Magnetising Inductance (L_m) for different operating conditions.

Figure 6.10 presents the magnetising inductance (L_m) variation as a function of I_{sd} and I_{sq} found through the current step response method detailed in this chapter. As it can be observed, the value of L_m decreases as the flux producing current increases irrespective of the load current level. This is an expected behaviour due to saturation of the main magnetising flux path, the same behaviour observed in figure 5.30. It is also worth noting that the value of L_m decreases as the I_{sq} current increases. This shows cross coupling between the d- and q-axis current components possibly as a result of cross – saturation. This phenomenon has been previously experienced by Gerada et al. (2003). They demonstrated through simulation and experimental measurements a variation of the main flux as a function of load current caused by cross saturation effects in IRFO induction machines arising from skew leakage flux. The results are similar however the method proposed in this thesis has the distinct advantage of not needing a load machines.

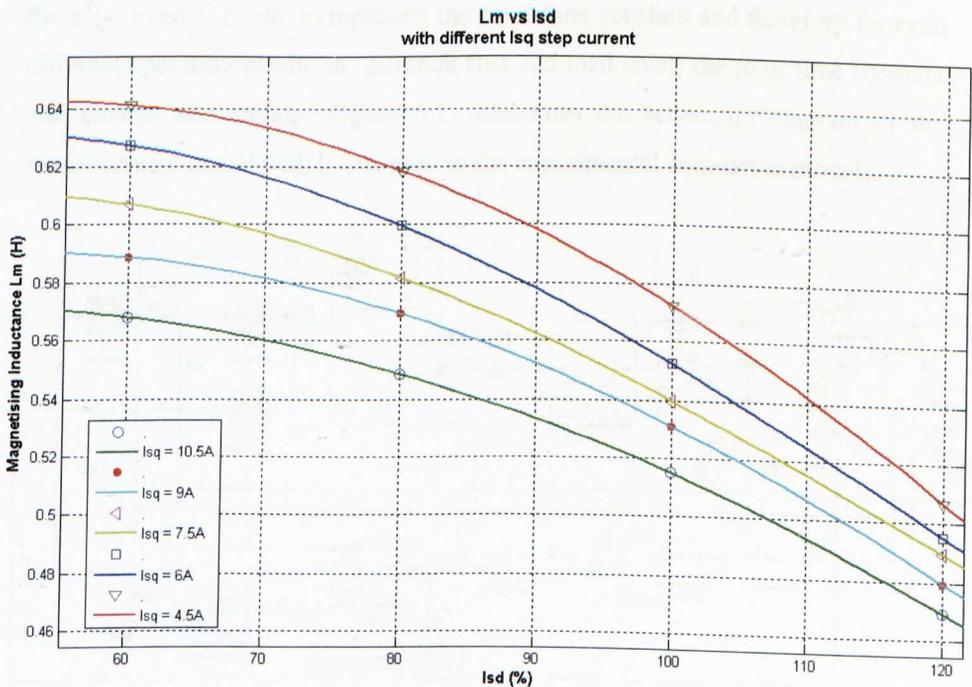


Figure 6.10: Magnetising Inductance vs I_{sd} step responses with variable I_{sq} current responses

Another important thing to note is the amount of inductance variation with load current for different magnetic loadings. The results illustrate that when the machine is operated with rated field current, and hence with the magnetic circuit operating at the onset of saturation, the amount of variation of L_m with the load current is considerably less than if operating at less than rated values. Machine operation in this region is often desirable to either extend the speed range or to improve drive efficiency. The results presented indicate that the magnetising inductance value is relatively sensitive to the load applied and appropriate compensation within the control system would be needed if correct orientation is to be maintained.

6.3.2 Improvement of the vector controlled drive

The improvement that can be achieved on a vector controlled drive by using the results of the GA tuning (Table 6.3) will be demonstrated in the figures below. In this case, a look – up table including all the results from table 6.3 has been used in the experimental model to represent the rotor time constant and therefore for each different operating condition (different flux and load level) the rotor time constant will change accordingly. Figure 6.11 illustrates the schematic diagram of the vector control model which was used in the experimental simulation model.

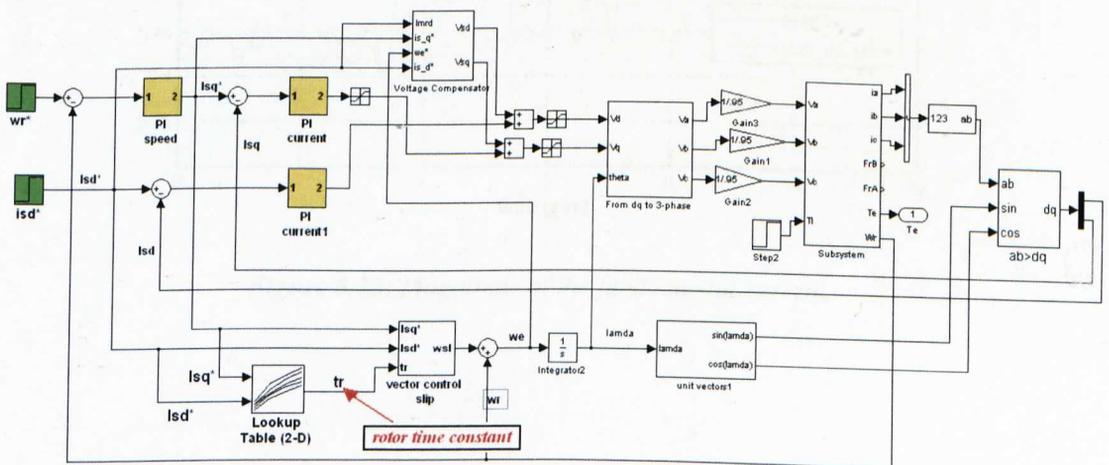


Figure 6.11: Vector Control block diagram with look – up table.

Figure 6.12 represents the magnitude of the fundamental current for both a fixed and an auto tuning (through the look – up table) rotor time constant (T_r) while the flux is changing (decreasing) and the load is constant. As it can be seen the current with the auto tuning T_r is smaller than the one with the fixed T_r , meaning that the performance of the vector controlled drive has been improved. As in a vector controlled drive, the best tuned system will give maximum Torque/Ampere, this proves the effectiveness and improvement of this method. Therefore GAs can be a suitable method for achieving a high performance of electrical servo drives. Finally, Figure 6.13 shows the value for both the fixed and the auto tuning rotor time constant (T_r), while figure 6.14 illustrates the change in the I_{sdr} and I_{sqr} currents.

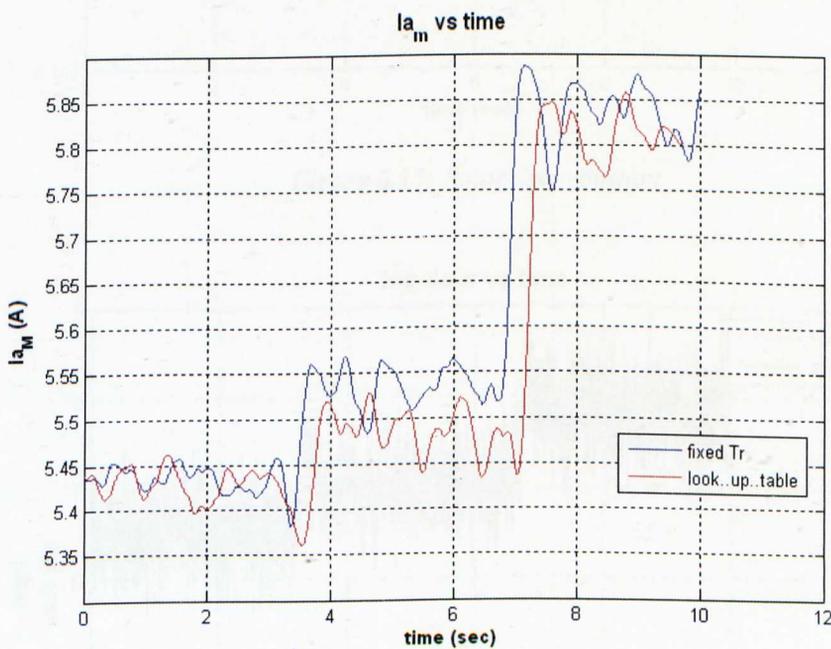


Figure 6.12: Magnitude of the fundamental current.

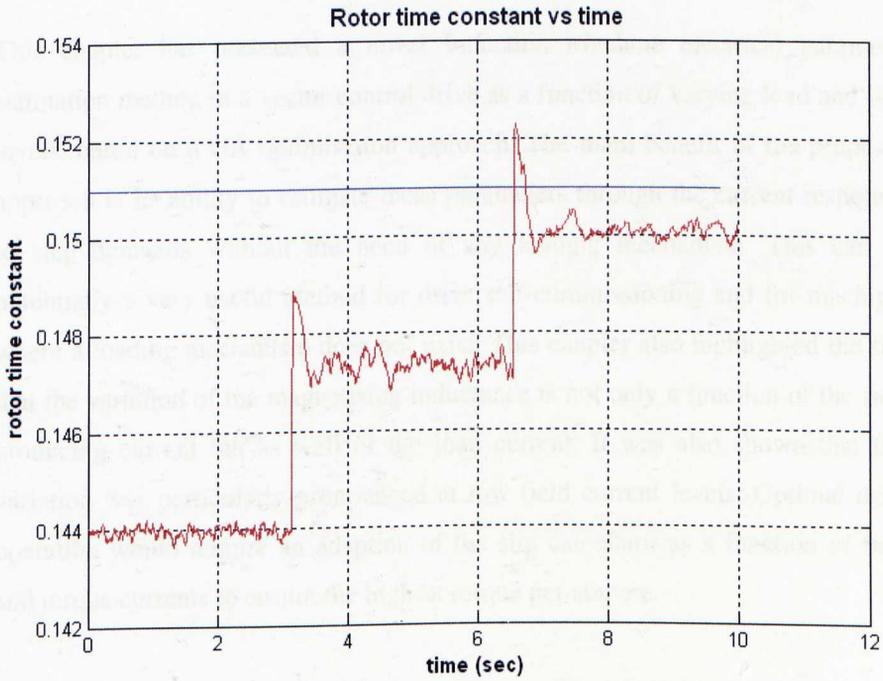


Figure 6.13: Rotor time constant

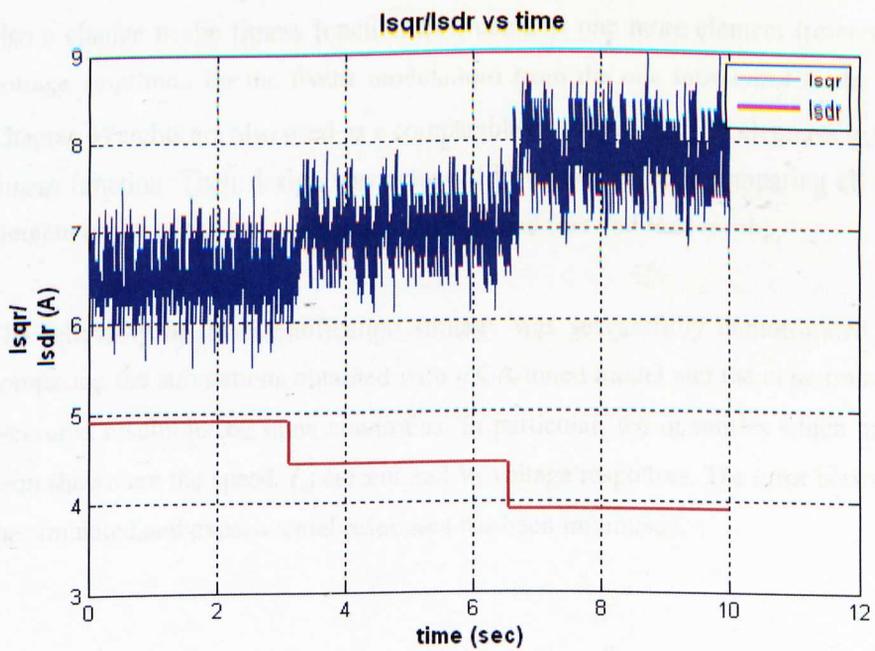


Figure 6.14: Current I_{sq}/I_{sdr} vs time

6.4 Chapter Summary

This chapter has presented a novel Induction Machine electrical parameter estimation method in a vector control drive as a function of varying load and flux levels, based on a GA optimisation approach. The main benefit of the proposed approach is its ability to estimate these parameters through the current responses to step demands without the need of any loading mechanism. This can be potentially a very useful method for drive self-commissioning and for machines where a loading mechanism does not exist. This chapter also highlighted the fact that the variation of the magnetising inductance is not only a function of the field producing current but as well of the load current. It was also shown that this variation was particularly pronounced at low field current levels. Optimal drive operation would require an adaption of the slip calculator as a function of field and torque currents to ensure the highest torque per ampere.

The implementation of the vector control scheme has been altered from the work done so far as there is no use of speed control. This is due to the fact that I_{sq} step current response used as input in the vector control scheme. In this work, there is also a change in the fitness function as it contains one more element (reference voltage amplitude for the PWM modulation) from the one introduced in the 5th Chapter. Weights are also used as a comparable contribution of all elements to the fitness function. Their design was achieved by evaluating and comparing all the element errors of the fitness function during rated flux and load level.

The reliability of this identification strategy was successfully demonstrated by comparing the simulations obtained with a GA tuned model and the experimental measured results in the same conditions. In particular, the quantities which have been shown are the speed, I_{sq} current and V_s voltage responses. The error between the simulated and experimental responses has been minimised.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

This thesis reports the results of a detailed investigation about the application of Genetic Algorithms to identify the electrical parameters of an induction motor under different operating conditions non – intrusively, when it is already inserted for example in a rotor flux oriented motor drive. Genetic Algorithms (GA) has been chosen as system identification method, it works on the basis of natural selection and uses only forward analysis. Hence, GA can readily be applied to a wide range of problems without the need for reformulation of equations or auxiliary information such as gradients required by some classical techniques. In addition to this, GA is less prone to premature convergence to local optima as searches in a global sense.

A literature review on different identification methods of I.M. parameters was included in Chapter 1. The application of intelligent methods was also introduced

and the importance of using more advanced algorithms on real – world problems was underlined.

An extensive analysis of the Indirect Field Oriented control technique for cage – type induction motor drives was presented in chapter 2. The design of the dq current and speed loop for the drive vector control was also explained. This chapter also included a study of parameter sensitivity in high – performance IFOC induction motor drives and parameter adaptation methods were analytically explained in order to overcome the effects of parameter sensitivity.

The foundation for understanding genetic algorithms, their power, their mechanics and their weaknesses was introduced in Chapter 3. A classification of optimisation algorithms according to the different approaches and techniques, highlighting their importance was also presented. In this chapter, Genetic Algorithms have been introduced and analysed as a stochastic global search technique. The basic concept of GAs is to simulate processes in evaluation of natural systems, specifically those that follow the principles of survival of the fittest. An analytical explanation of the differences between GA and other traditional search algorithms was also given.

The development of a robust and efficient identification strategy based on GA was also presented. The two – tier strategy proposed in this thesis has been designed primarily for structural identification problems but is robust enough to be applied to a range of problems. At the first tier a modified GA based on migration and artificial selection (MGAMAS) is used to identify the motor parameters within some given parameter search limits. An upper tier, the search space reduction method (SSRM) uses the results from the MGAMAS to assess the extent to which the parameters have converged and reduces the search space accordingly. The MGAMAS then searches within the reduced limits resulting in a more efficient and accurate estimation of the parameters. The MGAMAS provides robustness to the search strategy, simultaneously allowing for broad search while preserving and improving the most promising individuals. In this case, the population is split into multiple species, with real value encoding of variables and appropriate mutation operators, controlling the search direction in each species. Rank based

selection is used to maintain a constant selective pressure, while a tagging procedure guarantees diversity in the pool of best solutions.

Another technique that was presented in this thesis for improving GA is called cataclysmic mutation. This algorithm introduces a new diversity into the population via a form of restart the search when the population starts to converge. Cataclysmic mutation uses the best individual in the population as a template to re-initialise the population.

Sigma truncation scaling method has been also described. This method was designed as an improvement of linear scaling to deal with the negative evaluation values that appear in a run when most population members are highly fit but few of them have a very low value.

The experimental setup that was used to develop and test this new machine parameters identification method was briefly introduced in chapter 4. The overall structure of the experimental system has been shown and a detailed description of the hardware equipment is given. The test rig utilise a squirrel-cage 4kW induction machine and a 7.5kW IGBT inverter. The induction machine is coupled to a DC machine (act as loading devices), which is rated at 10kW and fed by a 4-quadrant converter.

Chapters 5 and 6 presented the core of this work, where the non – intrusive novel Induction Machine electrical parameters estimation method has been described and tested. The strategy based on a Genetic Algorithm heuristic optimisation approach, a simulation model and experimental transient measurements is able to identify the machine parameters directly in a motor drive. Based on both simulation and experimental investigations it is concluded that the use of the proposed strategy is an effective and reliable method for induction motor parameter identification. The reliability of this technique in estimating the machine parameters both in an average set and in function of different operating conditions (flux levels) has been presented successfully. This method gives also the basis for an optimised and high performance control design.

Finally, the identified parameters' behaviour in function of flux (I_{sd}^*) and torque (I_{sq}^*) level has been re – constructed using the new proposed fitness function. Results from figure 6.10 illustrate that the value of L_m decreases as the flux producing current increases irrespective of the load current level. On the other hand, the value of R_r remains almost the same for all operating conditions.

7.2 Future Work

The proposed GA strategy has proved to be an accurate off line algorithm and it is a novel approach for identifying the electrical I.M. parameters if appropriate measurement and objective functions are available. It can be further improved by:

- Incorporating more advanced GA techniques to increase the convergence speed and accuracy of the proposed method.
- Studying different performance indices in order to select the best objective function to this type of problem.

Generally, application to real systems depends heavily on how well the mathematical model is able to reproduce the response of the actual system. For GA for example, when using more complex models, it may unnecessarily increase the computational time and make convergence difficult. Hence, an improvement is to use models which have a good physical connection to the actual system but are simple enough that simulations can be carried out in a reasonable time. This research would involve for example looking into different types of finite element models or various time – history integration schemes to determine how the system response can be properly modelled in a simple way.

The next step would be to have an on – line system identification using genetic algorithms being applied to the drive experimental rig. In this case a very robust system identification scheme would be in place. If anything changed in the system thereafter, such as heating on the motor resulting in a increase of the rotor resistance, this scheme of system identification should still be able to produce

accurate models. However, the main argument of this idea is how to find a way to reduce the computational time as the system will be too slow.

Genetic algorithms could also be used in designing the PI controllers and hence producing a truly self – tuning system. If a change in the system occurred, the identification procedure would automatically produce a different model. The controller would then adapt to this new model without the need of engineer intervention.

- T_e : developed torque
- P_{out} : output power

From the above equivalent circuit it can be assumed that:

$$z_1 = R_s + j \cdot \omega_e \cdot L_{ls}$$

$$z_2 = j \cdot \omega_e \cdot L_m$$

$$z_3 = R_r + j \cdot \omega_e \cdot L_{lr}$$

$$z_4 = R_r \frac{\omega_r}{\omega_e - \omega_r} \quad \text{and}$$

$$z_5 = \frac{z_2(z_3 + z_4)}{z_2 + z_3 + z_4}$$

The stator phase current is given by:

$$I_s = \frac{V}{z_1 + z_5}$$

and the rotor phase current:

$$I_r = I_s \frac{z_2}{z_2 + z_3 + z_4} \quad \text{so, the magnetising current is:}$$

$$I_m = I_s - I_r$$

The electrical torque is given by:

$$T_e = 3 \cdot \frac{p}{2} \cdot \frac{I_r^2}{s} \cdot \frac{R_r}{\omega_e} \quad \text{and because the slip is equal to } s = \frac{\omega_e - \omega_r}{\omega_e} \text{ then:}$$

$$T_e = 3 \cdot \frac{p}{2} \cdot I_r^2 \cdot R_r \cdot \frac{1}{\omega_e - \omega_r}$$

- For the starting conditions the slip is equal to 1, $s = 1$
- For pull – out conditions we have to find the slip at the maximum torque, so it is :

$$\frac{dT}{ds} = 0, \quad s_{pull_out} = \frac{R_r}{\sqrt{R_{th}^2 + (X_r + X_{th})^2}}$$

A.1 Thevenin theorem of the equivalent circuit of Induction Motor

Thevenin's theorem states that it is possible to simplify any linear circuit, no matter how complex, to an equivalent circuit with just a single voltage source and series resistance connected to a load.

Thevenin's theorem is especially useful in analyzing power systems and other circuits where one particular resistor in the circuit (called the "load" resistor) is subject to change, and re-calculation of the circuit is necessary with each trial value of load resistance, to determine voltage across it and current through it.

Figure A.2 represents the Thevenin's theorem equivalent circuit.

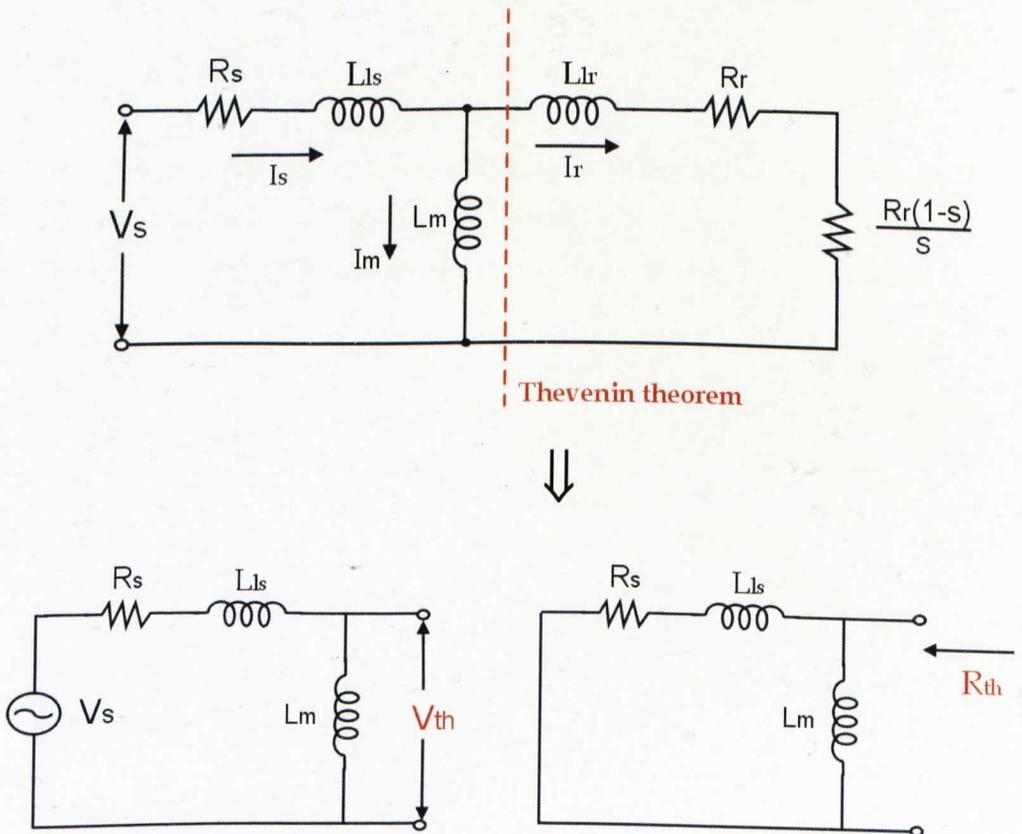


Figure A.2: Thevenin's Theorem Equivalent Circuit

From the above circuits it can be found that:

$$V_{th} = V_s \frac{j\omega_e L_m}{j\omega_e L_m + j\omega_e L_{ls} + R_s} \quad \text{and}$$

$$R_{th} = \frac{\omega_e^2 \cdot L_m^2 \cdot R_s}{R_s^2 + (\omega_e L_{ls} + \omega_e L_m)^2}, \quad X_{th} = \frac{\omega_e (L_m R_s^2 + \omega_e^2 L_{ls}^2 L_m + \omega_e^2 L_{ls} L_m^2)}{R_s^2 + (\omega_e L_{ls} + \omega_e L_m)^2}$$

Appendix B

B. Simulink Implementation of Induction Machine Model

The advantage of Simulink over circuit simulators is the ease in modelling the transients of electrical machines and drives and to include drive controls in the simulation. As the equations are known any drive or control algorithm can be modelled in Simulink.

B.1 Induction Motor Model

One of the most popular induction motor models derived from the below dynamic equivalent circuit (Figure B1) is Krause's model.

According to his model, the modelling equations in flux linkage form are as follows:

$$\frac{dF_{qs}}{dt} = \omega_b \left[v_{qs} - \frac{\omega_e}{\omega_b} F_{ds} + \frac{R_s}{X_{ls}} (F_{mq} - F_{qs}) \right] \quad (\text{B1})$$

$$\frac{dF_{ds}}{dt} = \omega_b \left[v_{ds} + \frac{\omega_e}{\omega_b} F_{qs} + \frac{R_s}{X_{ls}} (F_{md} - F_{ds}) \right] \quad (\text{B2})$$

$$\frac{dF_{qr}}{dt} = \omega_b \left[v_{qr} - \frac{(\omega_e - \omega_r)}{\omega_b} F_{dr} + \frac{R_r}{X_{lr}} (F_{mq} - F_{qr}) \right] \quad (\text{B3})$$

$$\frac{dF_{dr}}{dt} = \omega_b \left[v_{dr} + \frac{(\omega_e - \omega_r)}{\omega_b} F_{qr} + \frac{R_r}{X_{lr}} (F_{md} - F_{dr}) \right] \quad (\text{B4})$$

Appendix B: Simulink Implementation of Induction Machine Model

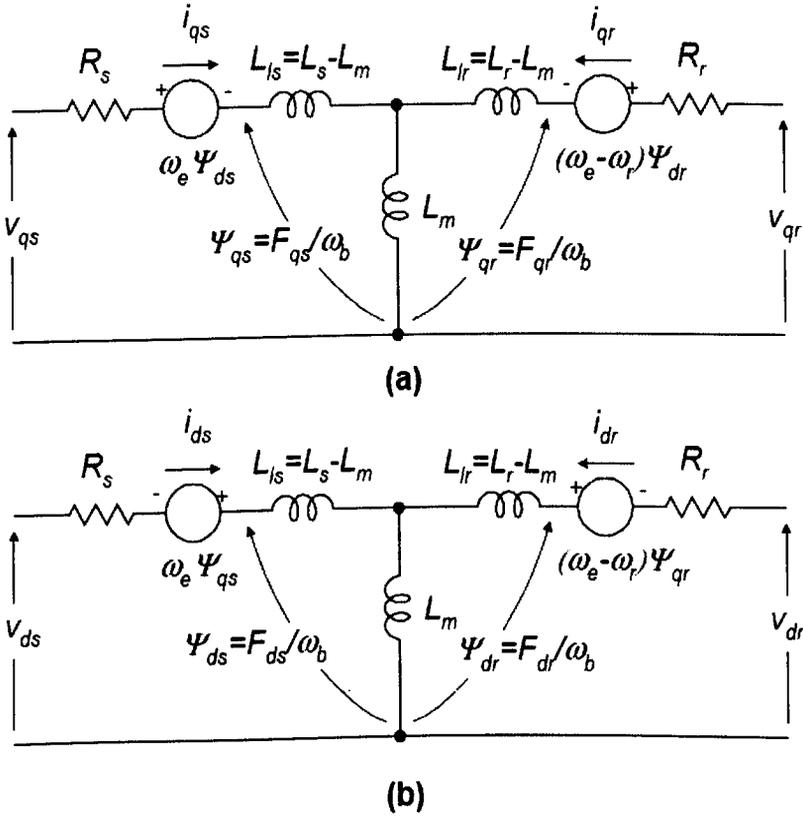


Figure B1: Dynamic or $d-q$ equivalent circuit of an induction machine

$$F_{mq} = x_{ml}^* \left[\frac{F_{qs}}{x_{ls}} + \frac{F_{qr}}{x_{lr}} \right] \quad (B5)$$

$$F_{md} = x_{ml}^* \left[\frac{F_{ds}}{x_{ls}} + \frac{F_{dr}}{x_{lr}} \right] \quad (B6)$$

$$i_{qs} = \frac{1}{x_{ls}} (F_{qs} - F_{mq}) \quad (B7)$$

$$i_{ds} = \frac{1}{x_{ls}} (F_{ds} - F_{md}) \quad (B8)$$

$$i_{qr} = \frac{1}{x_{lr}} (F_{qr} - F_{mq}) \quad (B9)$$

$$i_{dr} = \frac{1}{x_{lr}} (F_{dr} - F_{md}) \quad (B10)$$

$$T_e = \frac{3}{2} \left(\frac{p}{2} \right) \frac{1}{\omega_b} (F_{ds} \cdot i_{qs} - F_{qs} \cdot i_{ds}) \quad (\text{B11})$$

$$T_e - T_L = J \left(\frac{2}{p} \right) \frac{d\omega_r}{dt} \quad (\text{B12})$$

where

- d: direct axis,
- q: quadrature axis,
- s: stator variable,
- r: rotor variable,
- F_{ij} is the flux linkage ($i=q$ or d and $j=s$ or r),
- v_{qs}, v_{ds} : q and d – axis stator voltages,
- v_{qr}, v_{dr} : q and d – axis rotor voltages,
- F_{mq}, F_{md} : q and d axis magnetizing flux linkages,
- R_r : rotor resistance,
- R_s : stator resistance,
- X_{ls} : stator leakage reactance ($\omega_e L_{ls}$),
- X_{lr} : rotor leakage reactance ($\omega_e L_{lr}$),
- X_{ml}^* : $\frac{1}{\left(\frac{1}{X_m} + \frac{1}{X_{ls}} + \frac{1}{X_{lr}} \right)}$,
- i_{qs}, i_{ds} : q and d – axis stator currents,
- i_{qr}, i_{dr} : q and d – axis rotor currents,
- p: number of poles,
- J: moment of inertia,
- T_e : electrical output torque,
- T_L (or T_l): load torque,
- ω_e : stator angular electrical frequency,
- ω_b : stator angular electrical base frequency, and
- ω_r : rotor angular electrical speed.

In our case, we are using a squirrel cage induction machine so v_{qr} and v_{dr} in (B3) and (B4) are set to zero.

The five differential equations that represents an induction machine model can be solved by rearranging them in state – space form, $\dot{x} = Ax + b$ where $x=[F_{qs} F_{ds} F_{qr} F_{dr} \omega_r]^T$ is the state vector. Note that $F_{ij} = \psi_{ij} \omega_b$, where ψ_{ij} is the flux.

As a consequently, state – space form can be achieved by inserting (B5) and (B6) in (B1 – B4) and collecting the similar terms together so that each state derivative is a function of only other state variables and model inputs. Then, the modelling equations (B1 – B4 and B12) of a squirrel cage induction motor in state – space become:

$$\frac{dF_{qs}}{dt} = \omega_b \left[v_{qs} - \frac{\omega_e}{\omega_b} F_{ds} + \frac{R_s}{X_{ls}} \left(\frac{X_{ml}^*}{X_{lr}} F_{qr} + \left(\frac{X_{ml}^*}{X_{ls}} - 1 \right) F_{qs} \right) \right] \quad (B13)$$

$$\frac{dF_{ds}}{dt} = \omega_b \left[v_{ds} + \frac{\omega_e}{\omega_b} F_{qs} + \frac{R_s}{X_{ls}} \left(\frac{X_{ml}^*}{X_{lr}} F_{dr} + \left(\frac{X_{ml}^*}{X_{ls}} - 1 \right) F_{ds} \right) \right] \quad (B14)$$

$$\frac{dF_{qr}}{dt} = \omega_b \left[-\frac{(\omega_e - \omega_r)}{\omega_b} F_{dr} + \frac{R_r}{X_{lr}} \left(\frac{X_{ml}^*}{X_{ls}} F_{qs} + \left(\frac{X_{ml}^*}{X_{lr}} - 1 \right) F_{qr} \right) \right] \quad (B15)$$

$$\frac{dF_{dr}}{dt} = \omega_b \left[\frac{(\omega_e - \omega_r)}{\omega_b} F_{qr} + \frac{R_r}{X_{lr}} \left(\frac{X_{ml}^*}{X_{ls}} F_{ds} + \left(\frac{X_{ml}^*}{X_{lr}} - 1 \right) F_{dr} \right) \right] \quad (B16)$$

$$\frac{d\omega_r}{dt} = \left(\frac{p}{2J} \right) (T_e - T_L) \quad (B17)$$

B.2 Simulink Implementation

The inputs of a squirrel cage induction machine are the three – phase voltages, their fundamental frequency and the load torque. On the other hand, the outputs are the three – phase currents, the electrical torque and the rotor speed.

The d – q model requires that all the three – phase variables have to be transformed to the two – phase synchronously rotating frame. So, the induction

machine model will have blocks transforming the three – phase voltages to the d – q frame and the d – q currents back to the three – phase.

The induction machine model implemented in this paper is shown in figure B2 below. As it can be seen, it consists of four major blocks: abc – syn conversion, syn – abc conversion, unit vector calculation and the induction machine d – q model blocks. There is actually another block, the o – n conversion block that is required for an isolated neutral system, otherwise it can be bypassed.

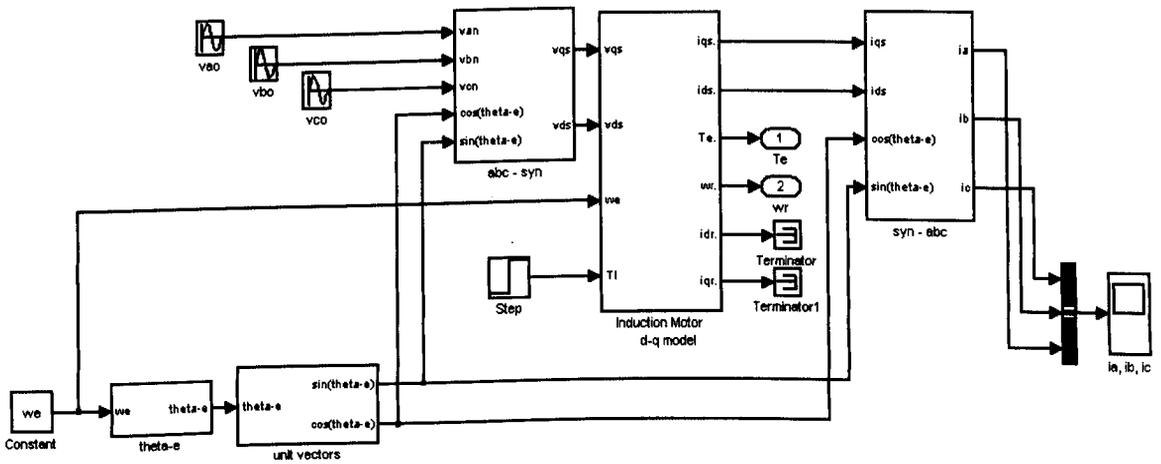


Figure B2: The complete induction machine Simulink model

B.2.1 Unit vector calculation block

Unit vectors $\cos\theta_e$ and $\sin\theta_e$ are used in vector rotation blocks, “abc – syn conversion block” and “syn – abc conversion block”. The angle, θ_e is calculated directly by integrating the frequency of the input three – phase voltages, ω_e .

$$\theta_e = \int \omega_e dt \quad (B18)$$

The unit vectors are obtained simply by taking the sine and cosine of θ_e . This block can be shown at the below figure B3:

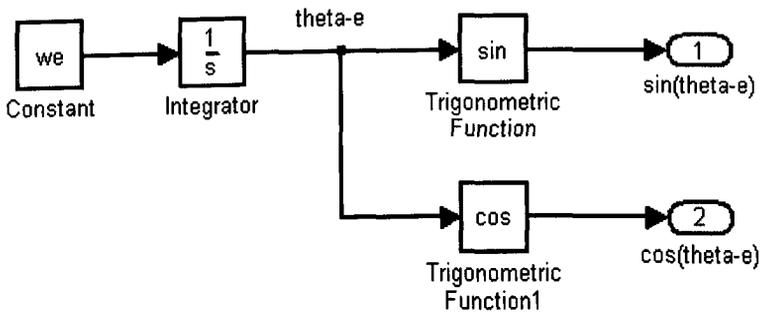


Figure B3: Unit vector calculation block

B.2.2 Induction Machine d – q model block

The inside of this block can be shown from the below figure B4 where each equation from the induction machine model is implemented in a different block.

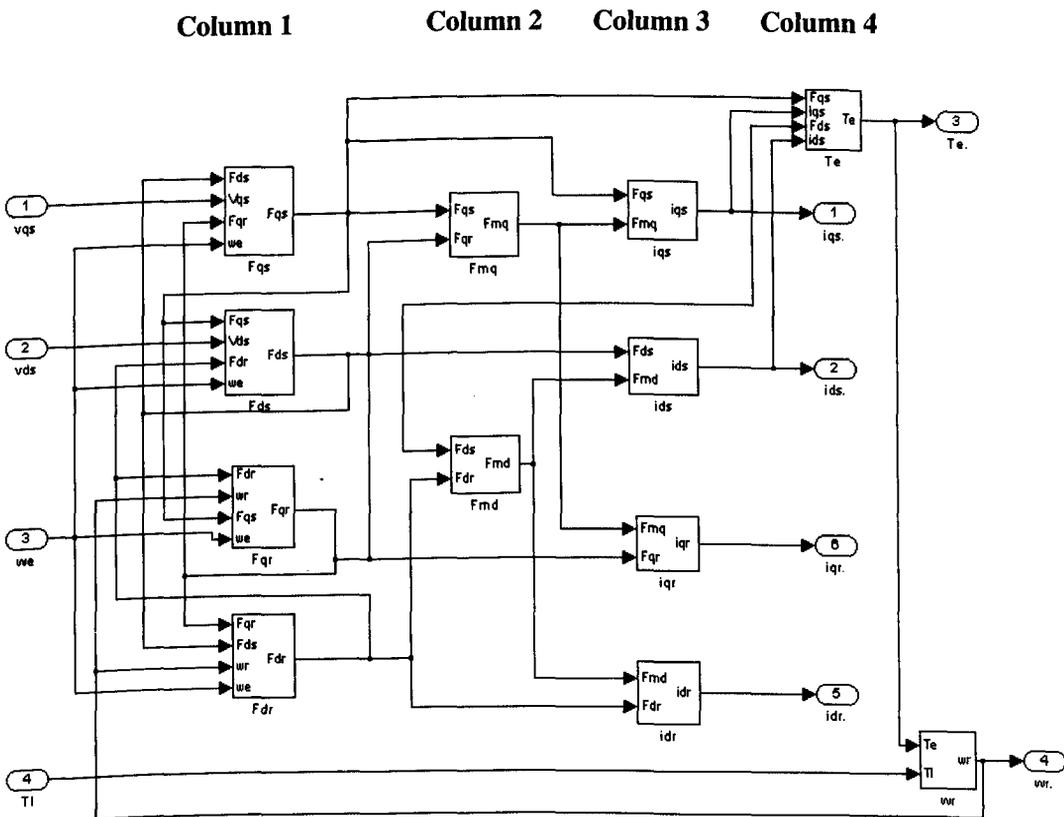


Figure B4: Induction machine dynamic model implementation in Simulink

The first column consists of four blocks that each of those represents the flux linkage state equations where these equations could be implemented using Simulink “State – space” block. Figure B5 shows what is inside of the block solving the equation (B1). All the other blocks in column are similar to this block. The blocks in column 2 solve the equations (B5) and (B6). Equations (B7-B10) use the flux linkages to solve for the stator and rotor d and q currents. The fourth and the last column include the electrical torque calculation from (B11) and the rotor speed calculation using the last state equation (B12). The rotor speed information is required for the calculation of the rotor flux linkages in column 1.

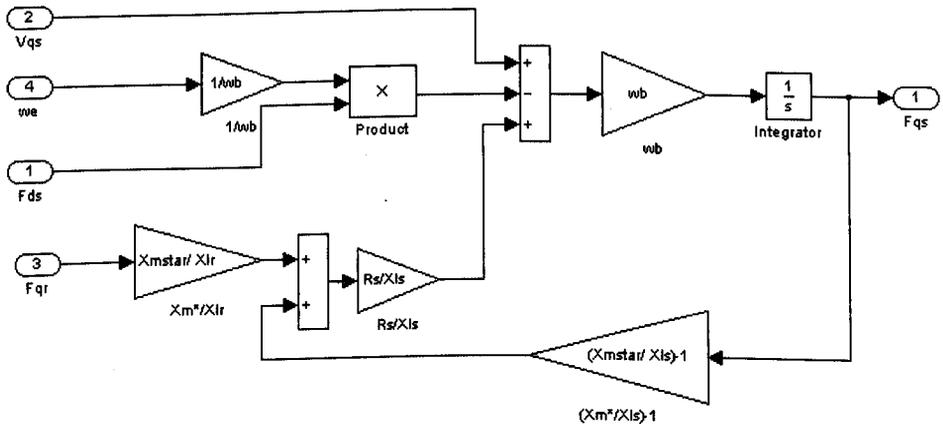


Figure B5: Implementation of (B1) in Simulink

The resulting model is modular and easy to follow. Any variable can be easily traced using the Simulink “Scope” blocks. The two blocks on column 4 calculate the torque and the speed of the induction machine, which can be used in torque control or speed control loops. These two variables can also be used to calculate the output power of the machine.

Appendix C

C. No – Load Test

To simulate the open – circuit test we need to run the motor with a slip of zero, so in this case the referred resistance of the rotor (R_2 / s) will be infinite and then the rotor current will be zero. In order to be able to achieve zero slip we have to drive the motor from another source such as a synchronous motor. We have also to run the test at rated frequency and with balanced 3 - phase voltages applied to the stator terminal. The measurements that are available from the no – load test if we suppose that the motor is operating at rated electrical frequency f_r are:

$$V_{NL} = \text{line – to – neutral stator voltage [V]}$$

$$I_{NL} = \text{line current [A]}$$

$$P_{NL} = \text{3 – phase electrical input power [W]}$$

In order to estimate the line – to – neutral voltage we have to measure the line – to – line voltage and then to divide by $\sqrt{3}$ (for a 3 – phase motor).

At no – load, we need a small value current (rotor) such as to overcome the friction and the windage losses that are linked to the rotation by producing an adequate torque. As a consequence, the value of rotor loss I^2R is insignificant small. Even though the tests are used to obtain the equivalent circuit parameters of an induction motor are almost the same as for the transformer, in an induction motor the magnetizing path includes an air gap which plays an important role to the equivalent circuit because it increases the exciting current. Moreover, we have to think that the value of the no – load stator I^2R loss will be considerable because of the significant increase of the exciting current.

When the motor is running it is possible to find the rotational loss P_{rot} by subtracting the stator I^2R losses from the no – load input power (the rotor I^2R losses are neglecting).

$$P_{rot} = P_{nl} - 3 I_{s, nl}^2 R_s \quad (C1)$$

For the reason that the value of the stator resistance R_s depends on the stator – winding temperature then we have to use the correct value to the equation (C1) of the no – load test.

It can be seen from the previous paragraphs that the core-loss as well the core – loss resistance are neglected and the entire no – load losses to friction and windage losses are assigned. There are many ways to separate the friction and windage losses from the core losses. As an example is that when the motor is not connected to the supply energy then an external motor can be used to drive the rotor to the no – load speed and the rotational loss will be the same to the output power.

In a different way, if the motor is running at no load and rated speed and if it is not connected to the power supply, the decrease in motor speed will be measured by the rotational loss as:

$$J \frac{d\omega_m}{dt} = -T_{rot} = -\frac{P_{rot}}{\omega_m} \quad (C2)$$

Where: J is the rotor inertia, P_{rot} is the rotational loss

The rotational loss at any speed ω_m is:

$$P_{rot}(\omega_m) = -\omega_m J \frac{d\omega_m}{dt} \quad (C3)$$

In addition, the rotational losses at rated speed can be calculated by equation (C3) as the motor stop working when it is operating at rated speed.

Then the core losses can be determined as:

$$P_{core} = P_{nl} - P_{rot} - 3I_{s, nl}^2 R_s \quad (C4)$$

The voltage drop across the stator resistance and leakage reactance can be ignored under no – load conditions because the stator current has very small value. So, in

this case the voltage across the core – loss resistance will be the same to the no – load line – to – neutral voltage and the core – loss resistance can be determined as:

$$R_c = \frac{3V_{S,nl}^2}{P_{core}} \quad (C5)$$

In order to make the equivalent circuit simpler than before and for the cause that the motor is working under the rated speed and the rated voltage, then the core – loss resistance will be unimportant in the equivalent circuit so, it is obviously that the core – loss resistance can be ignored and to simply include the core losses with the rotational losses.

At no – load, the motor slip s_{nl} is very small and as a consequently the rotor resistance R_2/s_{nl} is very large. The parallel combination of rotor and magnetizing branches then becomes jX_m shunted by the rotor leakage reactance X_2 in series with a very high resistance, and the reactance of this parallel combination is equal to X_m . As a result the reactance X_{nl} measured at the stator terminals at no load is equal to X_1+X_m , which is the self reactance X_{11} of the stator.

$$X_{nl} = X_{11} = X_1 + X_m \quad (C6)$$

So, the self – reactance of the stator can be determined from the no – load measurements. The reactive power at no load Q_{nl} can be determined as:

$$Q_{nl} = \sqrt{S_{nl}^2 - P_{nl}^2} \quad (C7)$$

where

$$S_{nl} = 3 \cdot V_{1,nl} \cdot I_{1,nl} \quad (C8)$$

is obviously the total power input at no load.

The no – load reactance X_{nl} can be calculated as:

$$X_{nl} = \frac{Q_{nl}}{3 \cdot I_{1,nl}^2} \quad (C9)$$

At no – load the power factor is small because the reactive power is much larger than the total input power ($Q_{nl} \gg P_{nl}$) so the no – load reactance is almost equal to the no – load impedance.

$$X_{nl} \approx \frac{V_{1,nl}}{I_{1,nl}} \quad (C10)$$

Blocked – Rotor Test

The blocked – rotor test will take place when the rotor shaft is clamped so the rotor remains stationary (hence the slip is equal to unity) with the rotor terminals short – circuited for a wound – rotor machine. Then we apply a reduced electromotive force (emf) to the stator terminals such as the full – load current flows in the stator windings.

With the rotor locked the input power, voltage and current are measured at a fraction of rated voltage to limit the input current and prevent overheating.

The measurements that are available from the blocked – rotor test are:

- $V_{1,bl}$ = The line – to – neutral voltage [V]
- $I_{1,bl}$ = The line current [A]
- P_{bl} = The total 3 – phase electrical input power [W]
- f_{bl} = The frequency of the blocked – rotor test [Hz]

The equivalent circuit for blocked – rotor conditions are very similar to that of a short – circuited transformer. The blocked rotor leakage impedance of an induction motor may be affected by magnetic saturation of the leakage – flux paths, by rotor frequency and by rotor position but the affection of the last one is small with squirrel - cage rotors.

The performance conditions of the blocked – rotor test such as the current and rotor frequency are approximately the same as those when the motor is running under operating conditions. As an example if we take into consideration the motor characteristics during starting condition then the slip is very close to unity and the blocked – rotor test should be taken at normal frequency and with currents approximately the same as came across in starting. On the other hand, in normal running conditions of the motor, the blocked – rotor test should be taken at a reduced voltage and frequency, since the values of rotor effective resistance and leakage inductance at the low rotor frequencies that means the values of slip will

be small as well, may be considerably different from their values at normal frequency, especially with double – cage or deep – bar rotors.

According to the blocked – rotor measurements, the blocked – rotor reactance can be found from the blocked rotor reactive power:

$$Q_{bl} = \sqrt{S_{bl}^2 - P_{bl}^2} \quad (C11)$$

where

$$S_{bl} = 3 \cdot V_{1,bl} \cdot I_{1,bl} \quad (C12)$$

is the total blocked – rotor apparent power. The blocked – rotor reactance, at rated frequency is:

$$X_{bl} = \left(\frac{f_r}{f_{bl}} \right) \cdot \left(\frac{Q_{bl}}{3 \cdot I_{1,bl}^2} \right) \quad (C13)$$

The blocked – rotor resistance can be calculated from the blocked – rotor input power as:

$$R_{bl} = \frac{P_{bl}}{3 \cdot I_{1,bl}^2} \quad (C14)$$

Once these parameters have been determined, the equivalent circuit parameters can be determined. From the below figure C1 and under blocked – rotor conditions can be determined the stator input impedance with slip equal to unity ($s=1$) as:

$$\begin{aligned} Z_{bl} &= R_1 + jX_1 + (R_2 + jX_2) \text{ in parallel with } jX_m \\ &= R_1 + R_2 \cdot \left(\frac{X_m^2}{R_2^2 + (X_m + X_2)^2} \right) \\ &\quad + j \left(X_1 + \frac{X_m (R_2^2 + X_2 (X_m + X_2))}{R_2^2 + (X_m + X_2)^2} \right) \end{aligned} \quad (C15)$$

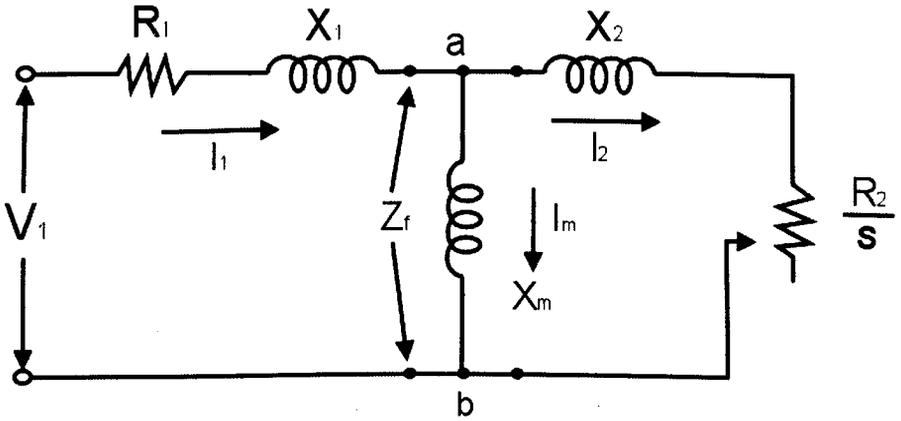


Figure C1: Equivalent circuit with the core – loss resistance R_c neglected

If we assume that $R_2 \ll X_m$ then the equation C15 is converted as:

$$Z_{bl} = R_1 + R_2 \left(\frac{X_m}{X_2 + X_m} \right)^2 + j \left(X_1 + X_2 \left(\frac{X_m}{X_2 + X_m} \right) \right) \quad (C16)$$

From the above equation C16 the apparent resistance and reactance under blocked – rotor conditions are given by:

$$R_{bl} = R_1 + R_2 \left(\frac{X_m}{X_2 + X_m} \right)^2 \quad (C17)$$

and

$$X_{bl} = X_1 + X_2 \left(\frac{X_m}{X_2 + X_m} \right) \quad (C18)$$

From equations C17 and C18, the rotor leakage reactance X_2 and resistance R_2 can be found as:

$$X_2 = (X_{bl} - X_1) \cdot \left(\frac{X_m}{X_m + X_1 - X_{bl}} \right) \quad (C19)$$

and

$$R_2 = (R_{bl} - R_1) \cdot \left(\frac{X_2 + X_m}{X_m} \right)^2 \quad (C20)$$

As referred on the no – load test, the stator resistance R_1 varies with stator – winding temperature so, when applying equation C20, care should be taken to use the value corresponding to the temperature of the blocked – rotor test.

Substituting for X_m from equation C6 into equation C19 gives the rotor leakage reactance X_2 in terms of the measured quantities X_{nl} and X_{bl} and the unknown stator leakage reactance X_1 .

$$X_2 = (X_{bl} - X_1) \cdot \left(\frac{X_{nl} - X_1}{X_{nl} - X_{bl}} \right) \quad (C21)$$

As it can be seen from the equation C21 it is mathematically impossible to find an equation in which we can determine the quantities X_1 and X_2 separately so, if the motor class is unknown, it is very usual to assume that X_1 and X_2 are equal.

In this case, from equation C21 the quantities X_1 and X_2 can be determined and then from equation C6 can be determined the magnetizing reactance X_m as:

$$X_m = X_{nl} - X_1 \quad (C22)$$

Eventually, knowing the values of X_m and X_2 as well as the value of the known stator resistance, from equation C20 can be determined the rotor resistance R_2 .

Stator Winding Resistance

As the windings are hot after the blocked – rotor test, the resistance between each pair of stator terminals can be measured by means of a bridge. Half of the average of the three resistance measurements gives the per – phase resistance of the stator windings, in case of that they are wye – connected.

Squirrel – Cage Motor Calculations

The most important equivalent – circuit parameters may be calculated from the results of both no – load and locked – rotor test as follows:

On no – load the slip s is equal to zero so that in figure C2,

$$\frac{R_R}{s} \gg |R_s + jX_L| \quad \Omega \quad (C23)$$

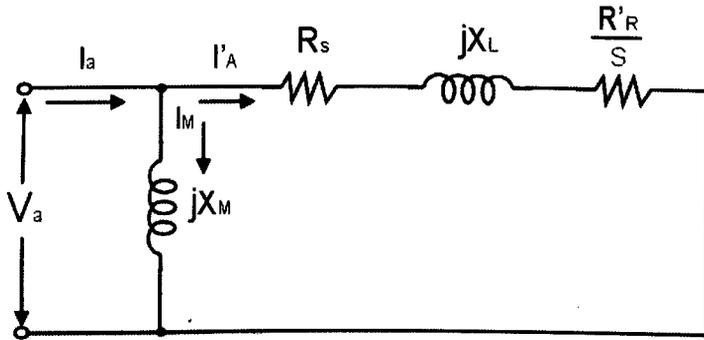


Figure C2: Equivalent circuit employed for analysis

As a result, the equivalent circuit under these conditions may be shown in figure C3(a).

In regarding the circuit of figure C2 we have neglected the core losses in the machine. However, core losses may be taken into consideration by a resistance connected across the stator terminals and it must be combined with the parallel resistance R'_R/s to give a resistance R_{rot} , which accounts for the rotational losses of the motor due to friction, windage and core loss. The resulting circuit is shown in figure C3(b).

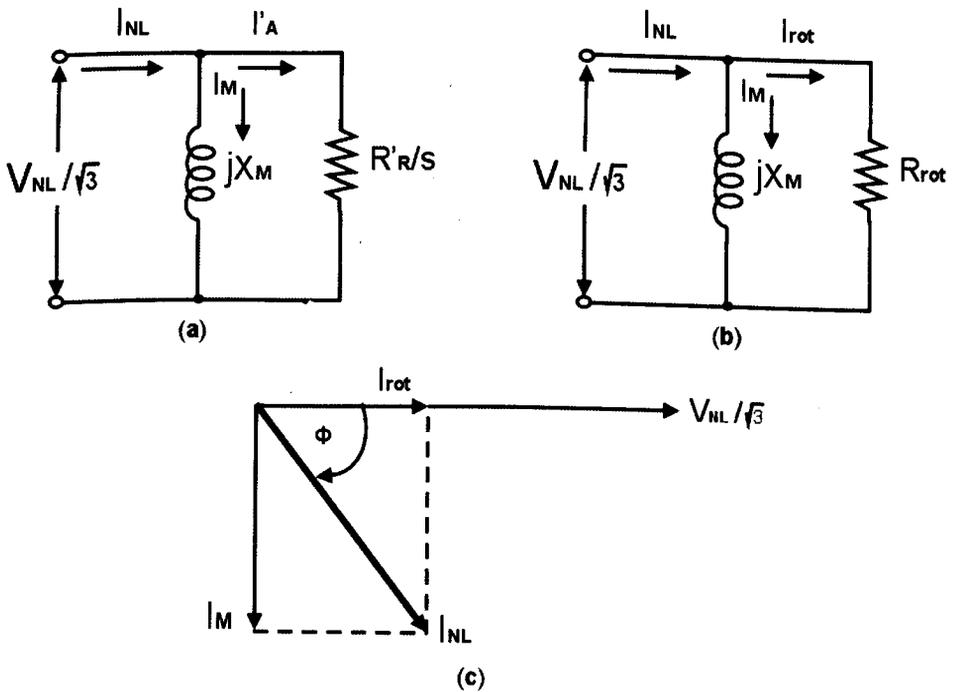


Figure C3: Induction motor no – load test.

The combined resistance R_{rot} is greater than the magnetizing reactance of the motor, which operates on no – load at a low power factor. The phase diagram for this condition of operation is shown in figure C3(c), where

$$\cos \phi = \frac{P_{NL}}{\sqrt{3} \cdot V_{NL} \cdot I_{NL}} \quad (C24)$$

$$I_{rot} = I_{NL} \cos \phi (A) \quad (C25)$$

$$I_M = I_{NL} \sin \phi (A) \quad (C26)$$

Then

$$X_M = \frac{V_{NL}}{\sqrt{3} I_M} (\Omega) \quad (C27)$$

In most machines, $I_M \gg I_{rot}$, so that the magnetizing reactance is very little greater than the no – load impedance as seen from the stator terminals. That is,

$$X_M \cong Z_{NL} = \frac{V_{NL}}{\sqrt{3} \cdot I_{NL}} (\Omega) \quad (C28)$$

Under locked – rotor conditions the slip is equal to unity ($s=1$). For most induction machines,

$$X_M \gg |(R_S + R'_R) + jX_L| (\Omega) \quad (C29)$$

Thus the branch X_M may be considered an open circuit and the effective equivalent circuit is that shown in the below figure C4

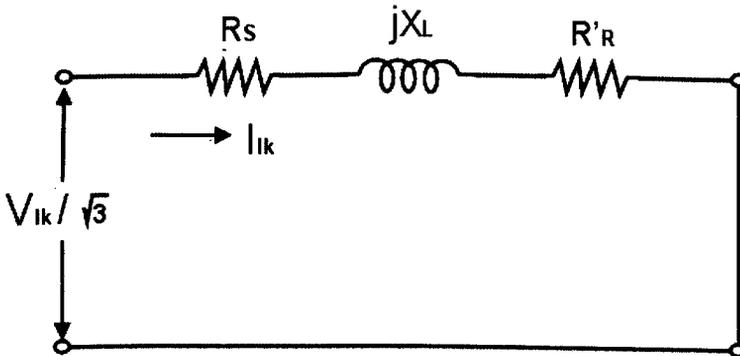


Figure C4: Induction motor locked – rotor test

Locked – rotor resistance is:

$$R_{lk} = \frac{P_{lk}}{3 \cdot I_{lk}^2} \quad (\Omega) \quad (C30)$$

Locked – rotor impedance is:

$$Z_{lk} = \frac{V_{lk}}{\sqrt{3} \cdot I_{lk}} \quad (\Omega) \quad (C31)$$

Locked – rotor reactance is:

$$X_{lk} = \sqrt{Z_{lk}^2 - R_{lk}^2} \quad (\Omega) \quad (C32)$$

and to a close approximation,

$$R_{lk} + jX_{lk} \cong R_S + R_R + jX_L \quad (\Omega) \quad (C33)$$

All the above parameters were calculated according to the squirrel – cage motors. If a wound – rotor motor is to be employed, it will try to set up an external rotor circuit resistance for the purpose of starting and/or speed control. For this reason it is very important to know the effective turns ratio, the factor k and the actual rotor – circuit resistance.

Ratios of Terminal Potential Difference to Induced Electromotive Force

Clamp the rotor. Open circuit the rotor terminals and apply rated electromotive force (emf) to the stator terminals. Measure the electromotive force appearing at the rotor terminals. As may be seen from the below figure C5, this will give the ratio $V_a/E_{m\Lambda}$.

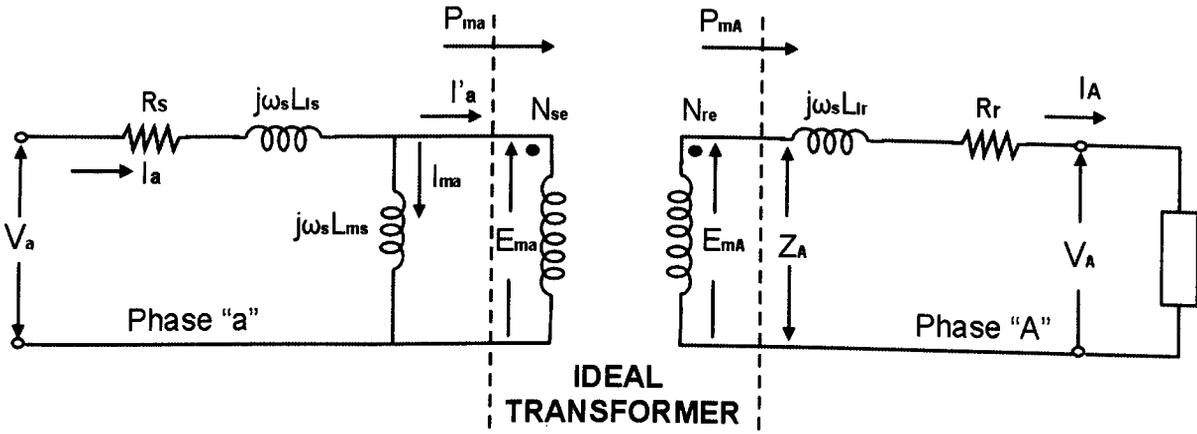


Figure C5: Per – phase equivalent circuit of the stationary motor

Open circuit the stator terminals and apply rated rotor potential difference (emf) to the rotor terminals. From figure C5, this will give the ratio E_{ma}/V_A .

Normally,

$$X_{ls} = \omega_s L_{ls} \gg R_s \quad (\Omega) \quad (C34)$$

and

$$X_{lr} = \omega_s L_{lr} \gg R_r \quad (\Omega) \quad (C35)$$

Since

$$X_{ms} = \omega_s L_{ms} \quad (\Omega) \quad (C36)$$

then

$$\frac{V_a}{E_{mA}} = \frac{N_{se}}{N_{re}} \frac{V_a}{E_{ma}} \approx \frac{N_{se}}{N_{re}} \left(\frac{X_{ls} + X_{ms}}{X_{ms}} \right) \quad (C37)$$

Similarly

$$\frac{E_{ma}}{V_A} = \frac{N_{se}}{N_{re}} \frac{E_{mA}}{V_A} \approx \frac{N_{se}}{N_{re}} \left(\frac{X_{ms}}{X_{ms} + X_{lr}} \right) \quad (C38)$$

For a wound – rotor motor, on the basis of symmetry of the leakage flux paths, it may be supposed that

$$X_{ls} = X_{lr}' \quad (\Omega) \quad (C39)$$

Then, from Eqs. (C37) and (C38), the turns ratio is given by

$$\frac{N_{se}}{N_{re}} = \left[\frac{V_a}{E_{mA}} \frac{E_{ma}}{V_A} \right]^{1/2} \quad (C40)$$

The factor k may also be obtained from Eqs. (C37) and (C38) as:

$$\left[\frac{X_{ms}}{X_{ms} + X_{ls}} \right]^2 = \frac{E_{ma}}{V_A} \frac{E_{mA}}{V_a} \quad (C41)$$

so

$$k = \frac{X_{ms}}{X_{ms} + X_{ls}} = \left[\frac{E_{ma}}{V_A} \frac{E_{mA}}{V_a} \right]^{1/2} \quad (C42)$$

Rotor Winding Resistance

The actual winding resistance R_r is measured in the same way as the stator winding resistance, but it can be under consideration that the measurements must be made between the slip rings (not between rotor terminals, since this would cause an additional resistance into the measurements according to the brush contact resistance). Then

$$R'_R = \frac{R'_r}{k^2} \quad (\Omega) \quad (C43)$$

This value should correspond closely with that obtained from Eq. (C33).

Appendix D

D. Dynamic Equations of the Induction Motor

The dynamic equations of an induction machine are always written for the equivalent 2 – coil system. They can be written in:

- stationary or stator $\alpha\beta$ frame
- a frame rotating at ω_e – the dq frame
- (a frame rotating at ω_r – fixed to the rotor)

The four equations governing the behaviour of the induction motor in the dq (arbitrary) frame are:

a. stator equations:

$$u_{sd} = R_s i_{sd} + \sigma L_s \frac{d}{dt} i_{sd} - \omega_e \sigma L_s i_{sq} + \frac{L_m}{L_r} \frac{d}{dt} \psi_{rd} - \omega_e \frac{L_m}{L_r} \psi_{rq} \quad (D1)$$

$$u_{sq} = R_s i_{sq} + \sigma L_s \frac{d}{dt} i_{sq} + \omega_e \sigma L_s i_{sd} + \frac{L_m}{L_r} \frac{d}{dt} \psi_{rq} + \omega_e \frac{L_m}{L_r} \psi_{rd} \quad (D2)$$

b. rotor equations:

$$0 = \frac{R_r}{L_r} \psi_{rd} + \frac{d}{dt} \psi_{rd} - \frac{L_m}{L_r} R_r i_{sd} - \omega_{sl} \psi_{rq} \quad (D3)$$

$$0 = \frac{R_r}{L_r} \psi_{rq} + \frac{d}{dt} \psi_{rq} - \frac{L_m}{L_r} R_r i_{sq} + \omega_{sl} \psi_{rd} \quad (D4)$$

It can be noticed from equations D1 – D4 that there are 4 state variables: $i_{sd}, i_{sq}, \psi_{rd}, \psi_{rq}$. These equations can also be written with different stages e.g. $i_{sd}, i_{sq}, i_{rd}, i_{rq}$ but there will always be four equations, two for the 2 – phase stator coils and two for the 2 – phase coils.

The field orientation equations can be written by taking the equation D1 – D4 and placing $\psi_{rq} = 0$.

$$u_{sd} = R_s i_{sd} + \sigma L_s \frac{d}{dt} i_{sd} - \omega_e \sigma L_s i_{sq} + \frac{L_m}{L_r} \frac{d}{dt} \psi_{rd} \quad (D5)$$

$$u_{sq} = R_s i_{sq} + \sigma L_s \frac{d}{dt} i_{sq} + \omega_e \sigma L_s i_{sd} + \omega_e \frac{L_m}{L_r} \psi_{rd} \quad (D6)$$

$$0 = \frac{R_r}{L_r} \psi_{rd} + \frac{d}{dt} \psi_{rd} - \frac{L_m}{L_r} R_r i_{sd} \quad (D7)$$

$$0 = -\frac{L_m}{L_r} R_r i_{sq} + \omega_{sl} \psi_{rd} \quad (D8)$$

where σ is the leakage coefficient:

$$\sigma = \frac{L_s L_r - L_m^2}{L_s L_r} \quad (D9)$$

From equations D5 to D8, the only interesting ones are the rotor equations (D7, D8). The stator equations are not very useful and so they can be omitted. Therefore, letting $\psi_{rd} = L_m i_{mrd}$ (where i_{mrd} is called the “equivalent magnetising current”) then equations D7 and D8 becomes:

$$\frac{L_r}{R_r} \frac{d}{dt} I_{mrd} + I_{mrd} = I_{sd} \quad (D10)$$

$$\omega_{sl} = \frac{R_r}{L_r I_{mrd}} I_{sq} \quad (D11)$$

Equations D10 and D11 are called vector field equation and vector constraint equation correspondingly and often termed as “vector control equations”.

Appendix E

E. Reference Frames

Vector control involves controlling the Induction Motor in field oriented coordinates (using V_{sd} , V_{sq} , I_{sd} , I_{sq} , ψ_{rd}).

However, in vector control there are three different reference frames as it can be seen in figure E.1. These frames are:

- The stator frame $\alpha\beta$ which is stationary
- The rotor frame ($\alpha\beta$) which rotates at the rotor frequency ω_r , with respect to the stator frame
- The synchronous frame dq which rotates at the excitation frequency ω_e with respect to the stator frame.

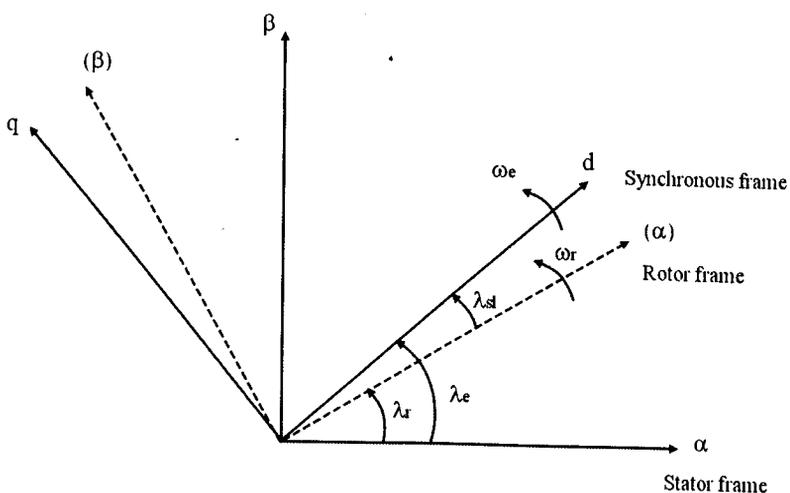


Figure E.1: Vector Control of IM reference frames

From figure E.1, λ represents the angles between different reference frames and ω represents the angular frequencies. The relationship between them is shown in equation E.1.

$$\omega_r = \frac{d\lambda_r}{dt}, \omega_e = \frac{d\lambda_e}{dt}, \omega_{sl} = \frac{d\lambda_{sl}}{dt} = \omega_e - \omega_r \quad (\text{E.1})$$

Where ω_{sl} is called the slip frequency.

E.1 Transform to Field Orientation

We must transform the 3 – phase stator quantities in 2 – axis dq quantities. This can be achieved within the transformation from “3 – phase to $\alpha\beta$ ” and then from “ $\alpha\beta$ to dq ”. It is known that:

$$I = I_\alpha + jI_\beta \quad (\text{E.2})$$

$$I = I_r e^{j0} + I_b e^{j\frac{2\pi}{3}} + I_y e^{j\frac{4\pi}{3}} \quad (\text{E.3})$$

E.1.1 Transformation to $\alpha\beta$ (from rby to $\alpha\beta$)

Figure E.2 shows the block diagram from the 3 – phase stator currents to 2 – axis dq currents.

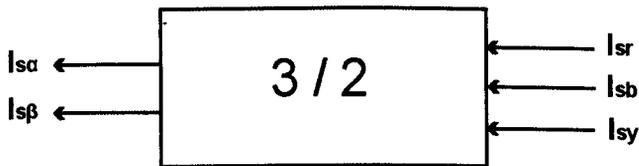


Figure E.2: Block diagram of a 3/2 transformation

Figure E.3 represents the geometrical transformation from 3 – phase to 2 – phase stator currents:

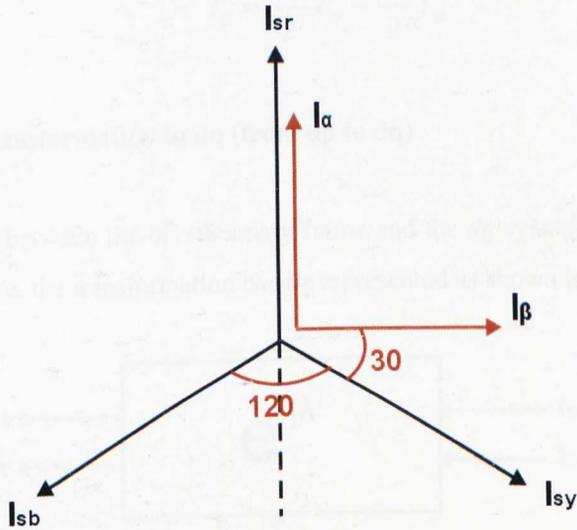


Figure E.3: Geometry of a 3/2 transformation

So, from figure E.3, it can be easily calculated that:

$$\begin{cases} I_\alpha = I_{sr} - I_{sy} \cos 60^\circ - I_{sb} \cos 60^\circ \\ I_\beta = I_{sy} \cos 30^\circ - I_{sb} \cos 30^\circ \end{cases} \Leftrightarrow$$

$$\begin{cases} I_\alpha = I_{sr} - \frac{1}{2} I_{sy} - \frac{1}{2} I_{sb} \\ I_\beta = \frac{\sqrt{3}}{2} I_{sy} - \frac{\sqrt{3}}{2} I_{sb} \end{cases} \Leftrightarrow \quad (E.4)$$

Moreover, from the balance condition:

$$I_{sr} + I_{sb} + I_{sy} = 0 \quad (E.5)$$

From the equations E.4 and E.5, it can be found the following equations E.6 and E.7 that are showing the conversion from 3-phase currents to I_α and I_β components.

$$I_\alpha = \frac{3}{2} I_{sr} \quad (E.6)$$

$$I_{\beta} = \frac{\sqrt{3}}{2} I_{sy} - \frac{\sqrt{3}}{2} I_{sb} \quad (\text{E.7})$$

E.1.2 Transformation to dq (from $\alpha\beta$ to dq)

If the angle λ between the $\alpha\beta$ stationary frame and the dq synchronously rotating frame is known, the transformation can be represented as shown in figure E.4.

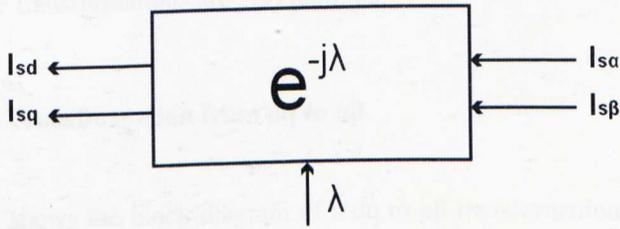


Figure E.4: Block diagram of a $\alpha\beta$ /dq transformation

From figure E.5, the flux and torque components of the current vector are determined by the following equations:

$$\begin{cases} I_{sd}(t) = I_{s\alpha}(t) \cos \lambda + I_{s\beta}(t) \sin \lambda \\ I_{sq}(t) = -I_{s\alpha}(t) \sin \lambda + I_{s\beta}(t) \cos \lambda \end{cases}$$

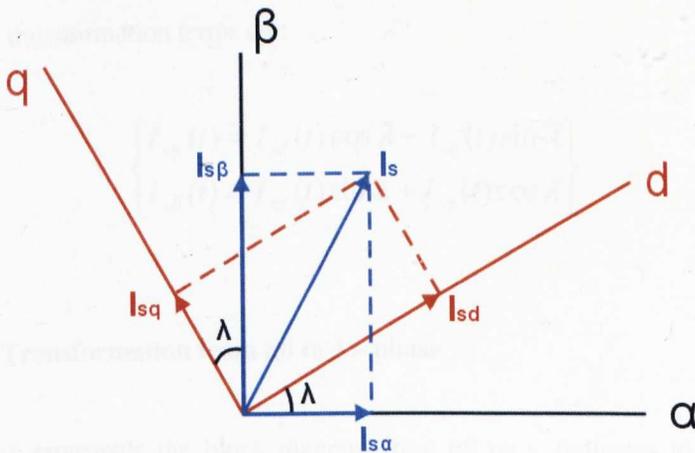


Figure E.5: Stator current space vector and its component in (α, β) and in the dq rotating reference frame

The symbol $e^{-j\lambda}$ is used since:

$I_{sdq} = I_{sd} + jI_{sq} = I_{s\alpha\beta} e^{-j\lambda}$ is the vector form of relating the dq currents to the $\alpha\beta$ current.

E.2 Transform FROM Field Orientated

The inverse transformations are also required.

E.2.1 Transformation from dq to $\alpha\beta$

Figure E.6 shows the block diagram of a dq to $\alpha\beta$ transformation. It is worthy to mention again that it is necessary to know the angle λ between these two frames.

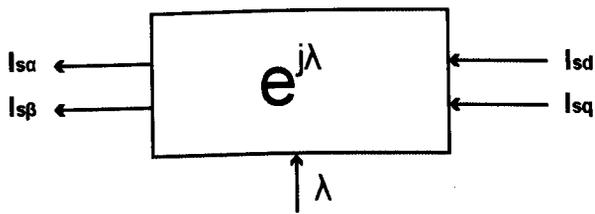


Figure E.6: Block diagram of a $dq/\alpha\beta$ transformation

The transformation terms are:

$$\begin{cases} I_{s\alpha}(t) = I_{sd}(t) \cos \lambda - I_{sq}(t) \sin \lambda \\ I_{s\beta}(t) = I_{sd}(t) \sin \lambda + I_{sq}(t) \cos \lambda \end{cases}$$

E.2.2 Transformation from $\alpha\beta$ to 3 – phase

Figure E.7 represents the block diagram from $\alpha\beta$ co – ordinates to 3 – phase transformation.

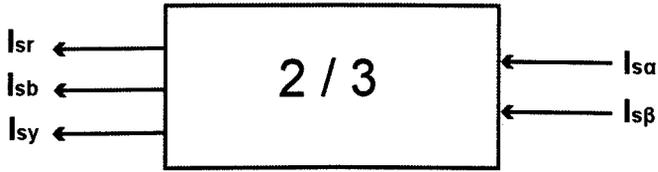


Figure E.7: Block diagram of a 2/3 transformation

The transformations terms are given by:

$$\left\{ \begin{array}{l} I_{sr}(t) = \frac{2}{3} I_{s\alpha}(t) \\ I_{sb}(t) = -\frac{1}{3} I_{s\alpha}(t) + \frac{1}{\sqrt{3}} I_{s\beta}(t) \\ I_{sy}(t) = -\frac{1}{3} I_{s\alpha}(t) - \frac{1}{\sqrt{3}} I_{s\beta}(t) \end{array} \right\}$$

The same transforms hold for the voltages and fluxes.

Appendix F

F. Estimation of mechanical parameters J and B

From equation B12 is known that: $T_e - T_L = J \left(\frac{2}{p} \right) \frac{d\omega_r}{dt} + B \cdot \omega_r$, and if the

load torque is equal to zero then

$$T_e = J \left(\frac{2}{p} \right) \frac{d\omega_r}{dt} + B \cdot \omega_r \Rightarrow \frac{P_{NL}}{\omega_r} = J \left(\frac{2}{p} \right) \frac{d\omega_r}{dt} + B \cdot \omega_r \quad (F1)$$

During no – load test and running the motor at different speed rates (etc. $\omega_5, \omega_4, \omega_3, \omega_2$), the power losses can be measured [Bose, 2002] for each of these speed rates (see Figure F1). In steady state, $\frac{d\omega_{r2}}{dt}, \frac{d\omega_{r3}}{dt}, \frac{d\omega_{r4}}{dt}$ and $\frac{d\omega_{r5}}{dt}$ are equal to

zero. Then from equation F1: $\frac{P_{NL}}{\omega_r} = B \cdot \omega_r$ and solving as for B (viscous friction

coefficient), B can be calculated for any speed rates ($\omega_5, \omega_4, \omega_3, \omega_2$).

To perform the free deceleration test, the induction motor is disconnected of the three phase ac power and the free deceleration speed characteristic is obtained (Figure F2). From figure F2, all the components $d\omega_r/dt$ for each speed rate ($\omega_5, \omega_4, \omega_3, \omega_2$), $d\omega_5/dt, d\omega_4/dt, d\omega_3/dt, d\omega_2/dt$, can be found. Then, from equation F1, solving as J:

$$J = \frac{P_{NL} p}{\omega_r} \frac{1}{2 d\omega_r/dt} - B \cdot \omega_r$$

The value of J is calculated for all different speed rates. Hence the moment of inertia can be estimated as the average value of each J:

Appendix F: Estimation of mechanical parameters J and B

$$J = \frac{J_2 + J_3 + J_4 + J_5}{4}$$

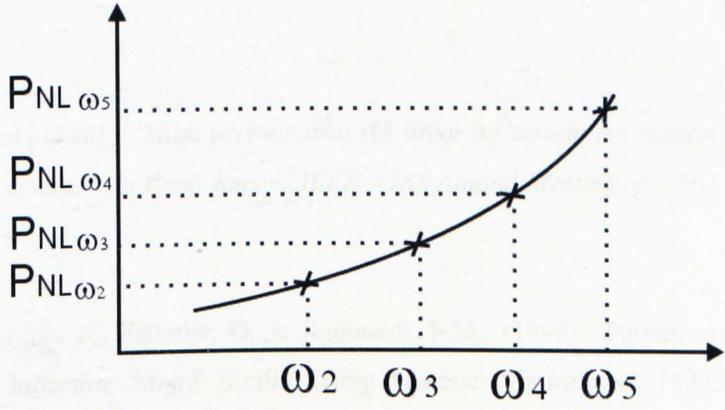


Figure F1: No – load Power losses vs speed

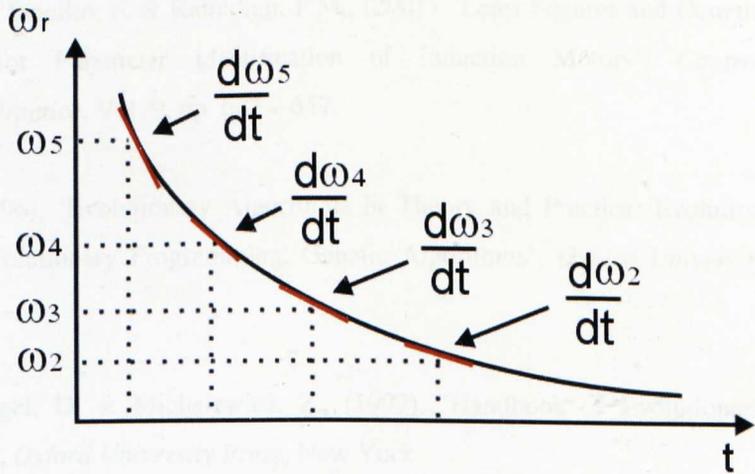


Figure F2: Deceleration speed

References

Abraham, A., (2005). 'Evolutionary computation', *Tech. Rep.*, Oklahoma State University.

Akamatsu, M. et al., (1981). 'High performance IM drive by coordinate control using a controller inverter', in *Conf. Record IEEE – IAS Annual Meeting*, pp. 562 – 571.

Alonge, F., D'Ippolito, F., Ferrante, G. & Raimondi, F.M., (1999). 'Parameter Identification of Induction Motor Model Using Genetic Algorithms'. *IEEE Proceedings Control Theory Appl.*, Vol. 145, No. 6, pp. 587 – 593.

Alonge, F., D'Ippolito, F. & Raimondi, F.M., (2001). 'Least Squares and Genetic Algorithms for Parameter Identification of Induction Motors'. *Control Engineering Practice*, Vol. 9, pp. 647 – 657.

Back, T., (1996). 'Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms', *Oxford University Press US*.

Back, T., Fogel, D. & Michalewicz, Z., (1997). 'Handbook of Evolutionary Computation', *Oxford University Press*, New York

Balmer, L., (1991). 'Signals and Systems – an Introduction', *Prentice Hall*

Barrero, F., Perez, J., Millan, R. & Franquelo, L.G., (1999). 'Self – commissioning for voltage – referenced voltage – fed vector controlled induction motor drives'. In *Proceedings of 1999 IEEE – IECON'99 annual meeting* (pp. 1033 – 1038).

Bertoluzzo, M., Buja, G. & Menis, R., (1997). 'Identification techniques of the induction motor parameters for self – commissioning field – oriented drives'. *Automatika*, vol. 4, pp. 103-115.

Bertsekas, D. P., (2000). 'Dynamic Programming and Optimal Control', *Vols. 1 & 2, 2nd ed. Athena Scientific*. ISBN 1-886529-09-4.

Bishop, R.R. & Richards, G.G., (1990). 'Identifying Induction Machine Parameters Using a Genetic Optimisation Algorithm'. *IEEE Proceedings – Southeastcon*.

Blasco – Gimenez, R., (1996). 'High Performance Sensorless Vector Control of Induction Motor Drives', *PhD Thesis, The University of Nottingham, UK*

Blaschke, F. (1972). 'The principle of field orientation as applied to the new transvector closed loop system for rotating field machines', *Siemens review*, vol.34, pp. 217 – 220.

Blickie, T. & Thiele, L. (1995). 'A mathematical analysis of tournament selection' in Eshelman, L.J., *Proceedings of the 6th International Conference on Genetic Algorithms*, pp. 506 – 511.

Bose, B., K. (2002). 'Modern Power Electronics and AC drives', *Prentice Hall*, New Jersey.

Boussak, M. & Capolino, G.A., (1988). 'Modern control tools for identification of three phase induction motors', in *Conf. Rec. Int. Conf. Electrical Machines (ICEM)*, pp. 215-220.

Bramlette, M.F., (1991). 'Initialization, mutation and selection methods in genetic algorithms for function optimization', in Belew, R.K. & Booker, L.B. (eds.) *Proceedings of the 4th International Conference on Genetic Algorithms*, Morgan Kaufmann.

Burgin, M. & Calude, C.S., (2007). 'Complexity of algorithms and computations', *Theoretical Computer Science*, Vol. 383, No. 2-3, pp. 111 – 114.

Cascella, G.L., Salvatore, N., Sumner, M. & Salvatore, L., (2005). 'On – line Simplex – Genetic Algorithm for Self – Commissioning of Electric Drives'. *Power Electronics and Applications, 2005 European Conference*.

Chainate, W., Thapatsuwan, P. & Pongcharoen, P., (2007). 'A New Heuristic for Improving the Performance of Genetic Algorithms', *World Academy of Science*, Volume 21, ISSN 1307-6884.

Chan, C.C. & Wang, H., (1990). 'An effective method of rotor resistance identification for high performance induction motor vector control', *IEEE Transactions on Industrial Electronics*, Vol. 37, pp. 477-482.

Chipperfield, A.J., Fleming, P.J., Pohlheim, H., (1994a). 'A Genetic Algorithm Toolbox For MATLAB', *Proceedings of International Conference on Systems Engineering*, UK Coventry, pp. 200-207.

Chipperfield, A.J, Fleming, P.J, Fonseca, C.M., (1994b). 'Genetic Algorithm Tools for Control Systems Engineering', *Proceedings of Adaptive Computing in Engineering Design and Control*, Plymouth Engineering Design Centre, pp.128-133.

Chipperfield, A.J, Fleming, P.J, (1995). 'The MATLAB Genetic Algorithm Toolbox', *IEE Colloquium Applied Control Techniques Using MATLAB*, pp. 10/11 – 10/14.

Chung, P.Y., Dolen, M. & Lorenz, R.D., (2000). 'Parameter Identification for Induction Machines by Continuous Genetic Algorithms'. *ANNIE 2000 Conference*, p.341-346.

Cirrincone, M., Pucci, M., Cirrincone, G. & Capolino, G.A., (2003). 'A New Experimental Application of Least – Squares Techniques for the Estimation of the

Induction Motor Parameters'. *IEEE Transactions on Industry Applications*, Vol. 39, No 5.

Coley, D.A., (1999). An introduction to Genetic Algorithms for Scientists and Engineers, *World Scientific Publishing Co. Pte. Ltd.*, Singapore.

Cormen, T., H., Leiserson, C., E., & Rivest, R., L., (2000). 'Introduction to Algorithms', MIT Press.

Darwin, C., (1859). 'The Origin of Species reprinted Gramercy Books', New York, 1995.

Davis, L., (1989). 'Adapting Operator Probabilities in Genetic Algorithms', *Proceedings of the Third International Conference on Genetic Algorithms*, pp. 61-69.

De Jong, K.A., (1975). 'An analysis of the behavior of a class of genetic adaptive systems', Ph.D. thesis, University of Michigan, Ann Arbor.

Despalatovic, M., Jadric, M. & Terzic, B., (2005). 'Identification of Induction Motor Parameters from Free Acceleration and Deceleration Tests', ISSN 0005-1144, *ATKAAF* 46(3-4), 123-128.

dSPACE, (2004). 'ControlDesk Experiment Guide'.

Engelbrecht, A. P., (2007): 'Computational Intelligence : An Introduction'. 2nd Edition, John Wiley and Sons Ltd.

Eshelman, L.J., Caruana, R.A. & Schaffer, J.D., (1989). 'Biases in the crossover landscape' in Schaffer, J.D. (ed.), *Proceedings of the Third International Conference on Genetic Algorithms*, Morgan Kaufmann.

Eshelman, L.J., (1991). 'The CHC Adaptive Search Algorithm', *Foundations of Genetic Algorithms*, G. Rawlins, ed. Morgan-Kaufmann, pp. 256 – 283.

Fang, C.-M., Lin, S.-K. & Wang, S.-J., (2005). 'On – line parameter estimator of an induction motor at standstill'. *Control Engineering Practice* 13, pp. 535-540.

Fang, C.-M. & Lin, J.-J., (1999). 'Using a genetic algorithm to generate alternative sketch maps for urban planning'. *Computers Environment and Urban Systems*, 23 (2), 91-108.

Fitzgerald, A.E., Kingsley, C.Jr., Umans, S.D., (2003). 'Electric Machinery'. *Sixth Edition*.

Fleming, P.J., Purshouse, R.C., (2002). 'Evolutionary algorithms in control systems engineering: a survey'. *Control Engineering Practice* 10, 1223-1241

Forrest, S., (1985). 'Documentation for Prisoners Dilemma and Norms programs that use the genetic algorithm'. *Unpublished manuscript, University of Michigan, Ann Arbor*.

Futuyama, D.J., (1990). *Evolutionsbiologie*, Birkhauser Verlag, Basel.

Gabriel, R. and Leonhard, W., (1982). 'Microprocessor control of induction motor', *Record International Semiconductor Power Converter Conference*, Orlando, FL, pp. 385 – 396.

Gerada, C., Bradley, K., Sumner, M. & Sewell, P., (2003). 'Analysis of and Compensation for Cross Saturation due to Skew Leakage in Indirect Field Orientated Induction Motor Drives'. *In Proc. EPE Conf.*, Toulouse, France.

Gimenez, R., B. (1995). 'High Performance Sensorless Vector Control of Induction Motor Drives', PhD Thesis, University of Nottingham

Goldberg, D.E., (1989). 'Genetic Algorithms in Search, Optimization, and Machine Learning', *Addison – Wesley Publishing Company*.

Goldberg, D.E., (1990). 'A note on Boltzmann tournament selection for genetic algorithms and population – orientated simulated annealing' *Complex System*, 4, pp. 445-460.

Goldberg, D.E. & Deb, K., (1991). 'A comparative analysis of selection schemes used in genetic algorithms', Rawlins, G. (ed.), *Foundations of Genetic Algorithms*, Morgan Kaufmann, pp. 69-93.

Goldberg, D.E., Deb, K., & Thierens, D., (1993). 'Toward a better understanding of mixing in genetic algorithms', *Journal of the Society of Instrument and Control Engineers*, 32(1), 10 – 16.

Grefenstette, J.J., (1986). 'Optimization of control parameters for genetic algorithms', *IEEE Transactions on Systems, Man, and Cybernetics* 16(1), PP. 122-128.

Hansheng, Lin & Lishan, Kang, (1999). 'Balance between Exploration and Exploitation in Genetic Search', *Journal of Natural Sciences*, Vol. 4, No 1

Haupt, Randy L. & Haupt, Sue Ellen, (2004). 'Practical Genetic Algorithms', Second Edition, John Wiley & Sons, Inc., Publication.

Hinterding, R., Gielewski, H. & Peachey, T.C., (1995). 'The nature of mutation in genetic algorithms' in Eshelman, L.J., *Proceedings of the 6th International Conference on Genetic Algorithms*, pp. 65-72.

Holland, J., H., (1975). 'Adaptation in the Natural and Artificial Systems'. University of Michigan Press. 2nd ed: MIT Press, 1992.

Holland, J.H., (1992). 'Genetic Algorithms – Computer Programs that “evolve” in ways that resemble natural selection can solve complex problems even their creators do not fully understand'. *Scientific American*, July 1992, pp. 44-50.

Huang, K.S., Kent, W., Wu, Q.H. & Turner, D.R., (1999). 'Parameter Identification of an Induction Machine Using a Genetic Algorithms'. *Proceedings of the 1999 IEEE*.

Hughes, Austin, (2006). 'Electric Motor and Drives: Fundamentals, Types and Applications'. *Third Edition*.

Hung, Y-F, Shih, C-C & Chen, C-P, (1999). 'Evolutionary algorithms for production planning problems with setup decisions'. *Journal of the Operational Research Society*, 50, pp. 857-866.

Hussain, T., S., (1997). 'An introduction to evolutionary computation', Tech. Rep., Department of Computing and Information science, Queen's University, Kingston.

Incze, J.J., Szabo, C., Imecs, M., Matis, I. & Szoke, E., (2005). 'Computer Controlled Up-to-Date Experimental Equipment for AC Drive Development', 6th *International Symposium of Hungarian Researchers on Computational Intelligence*, November 18 – 19, Budapest.

Ishihara, K., et al., (1983). 'AC drive system for tension reel control', *Record IEEE – IAS Annual Meeting, Mexico*, pp. 61 – 66.

Jasim, O., (2009). 'An Extended Induction Motor Model for Investigation of Faulted Machines and Fault Tolerant Variable', *PhD Thesis*, University of Nottingham, UK.

Jones, T., (1995). 'Crossover, macromutation and population – based search', in Eshelman, L.J. (ed.), *Proceedings of the Sixth International Conference on Genetic Algorithms*, Morgan Kaufman.

Jotten, R. & Schierling, H., (1987). 'Adaptive and self – commissioning control for a drive with induction motor and voltage source inverter'. 2nd *European Conference on Power Electronics and Applications*, Grenoble, France.

Juang, J., N. & Phan, M., (1994). 'Linear system identification via backward – time observer models', *J. Guid. Control Dyn.*, 17(3), 505 – 512.

Kalyanmoy, D., (2001). 'Multi – Objective Optimisation Using Evolutionary Algorithms', *Wiley Interscience Series in Systems and Optimisation. John Wiley & Sons, Inc.*, New York, NY, USA, ISBN : 978-0-47187-339-6.

Kampisios, K., Zanchetta, P., Gerada, C., Trentin, A. & Jasim, O., (2008a). 'Induction Motor Parameters Identification using Genetic Algorithms for Varying Flux Levels', 13th *International Power Electronics and Motion Control Conference, EPE-PEMC 2008*, 1-3 Sept., Page(s): 887 – 892.

Kampisios, K., Zanchetta, P., Gerada, C. & Trentin, A., (2008b). 'Identification of Induction Machine Electrical Parameters Using Genetic Algorithms Optimization', *Industry Applications Society Annual Meeting, IAS '08. IEEE*, 5 – 9 Oct., Page(s): 1 – 7.

Karanayil, B., (2005). 'Parameter Identification for Vector Controlled Induction Motor Drives Using Artificial Neural Networks and Fuzzy Principles', *PhD Thesis*, The University of New South Wales.

Khambadkone, A.M. & Holtz, J., (1991). 'Vector – controlled induction motor with a self – commissioning scheme'. *IEEE Trans. On Ind. Electron.*, vol. 38, pp. 322 – 327.

Kido, T., Kitano, H. & Nakanishi, M., (1993). 'A Hybrid Search for Genetic Algorithms: Combining Genetic Algorithms, TABU Search, and Simulated Annealing', in *Proceedings of the 5th International Conference on Genetic Algorithms*, San Mateo, CA, p.641.

Kokash, Natallia, (2005). 'An Introduction to Heuristic Algorithms', Research Methodology course, Department of Informatics and Telecommunications, University of Trento, Italy.

Krause, P.C., Wasynczuk, O. & Sudhoff, S.D., (2002). 'Analysis of Electric Machinery and Drive Systems'. *IEEE Computer Society Press*.

Krishnan, R. & Doran, F.C., (1987). 'Study of Parameter Sensitivity in High – Performance Inverter – Fed Induction Motor Drive Systems', *IEEE Transactions on Industry Applications*, Vol. IA-23, No. 4.

Krishnan, R. & Bharadwaj, A.S., (1991). 'A Review of Parameter Sensitivity and Adaptation in Indirect Vector Controlled Induction Motor Drive Systems', *IEEE Transactions on Power Electronics*, Vol. 6, No. 4.

Krose, B. & Smagt, P., (1996). 'An introduction to Neural Networks'. University of Amsterdam, Nov. 1996.

Kyung, K. Choi & Nam H. Kim, (2004). 'Structural Sensitivity Analysis and Optimization: Linear Systems', Springer.

Lai, L.C., Demarco, C.L. & Lipo, T.A., (1992). 'An extended kalman filter approach to rotor time constant measurement in PWM induction motor drives', *IEEE Transactions on Industry Applications*, Vol. 28, pp. 96 – 104.

Leonard, W., (1997). 'Control of Electrical Drives', Springer.

Levi, E., (1994). 'Magnetic Saturation in Rotor – Flux – Oriented Induction Machine Drives: Operating Regimes, Consequences and Open Loop Compensation'. *European Transactions on Electrical Power ETEP*, Vol. 4, No.4.

Levi, E., (1996a). 'Main flux saturation modelling in double – cage and deep – bar induction machines', *IEEE Transactions on Energy Conversion*, Vol. 11, Issue 2.

Levi, E., Sokola, M., Boglietti, A. & Pastorelli, M., (1996b). 'Iron loss in rotor flux oriented induction machines: identification, assessment of detuning and compensation'. *IEEE Transactions on Power Electronics*, Vol. 11, Issue 5.

Li, X. & Yeh, A.G., (2005). 'Integration of genetic algorithms and GIS for optimal location search', *International Journal of Geographical Information Science*, 19(5), May 2005, pp. 581 – 601.

Lorenz, R.D. & Novotny, D.W., (1990). 'Saturation Effects in Field – Oriented Induction Machines', *IEEE Transactions on Industry Applications*, Vol. 26, No. 2.

Lyra, R.O.C., Silva, S.R. & Cortizo, P.C., (1995). 'Direct and Indirect Flux Control Of An Isolated Induction Generator', *IEEE Catalogue No. 95TH8025*

Matsuo, T. & Lipo T.A., (1985). 'A rotor parameter identification scheme for vector controlled induction motor drives', *IEEE Transaction on Industry Applications*, Vol. 21, pp. 624 – 632.

Mawdesley, M.J., Al – Jibouri, S.H. & Yang, H., (2002). 'Genetic Algorithms for Construction Site Layout in Project Planning' *Journal of Construction Engineering and Management* Sept/Oct 2002, pp. 418 – 426.

Michalewicz, Z. & Michalewicz, M., (1997). 'Evolutionary Computation Techniques and Their Applications', *IEEE International Conference on Intelligent Processing Systems*, October 28 – 31, Beijing, China.

Michalewicz, Zbigniew, (1999). 'Genetic Algorithms + Data Structures = Evolution Programs', Springer-Verlag.

Michalewicz, Z. & Fogel, B., David, (2004). 'How to Solve It. Modern Heuristics', Springer, second, revised and extended edition, ISBN: 978-3-54022-494-5.

Michalik, W. & Devices, W., (1998). 'Standstill estimation of electrical parameters in motors with optimal input signals'. In *Proceedings of 1998 IEEE circuits and systems* (pp. 407 – 413).

Miller, B., L. & Goldberg, D., E., (1995). 'Genetic Algorithms, Tournament Selection, and the Effects of Noise', *Department of General Engineering, University of Illinois at Urbana-Champaign*.

Mitchell, M., (1996). *An introduction to Genetic Algorithms*. MIT Press, 1996.

Moons, C. & De Moor, B., (1995). 'Parameter Identification of Induction Motor Drives'. *Automatica*.

Moon, S.I., Keyhani, A. & Pillutla, S., (1999). 'Nonlinear Neural Networks Modeling of an Induction Machine'. *IEEE Transactions on Control Systems Technology*, Vol.7, No 2, pp. 203 – 211.

Muhlenbein, H., (1992). 'How genetic algorithms really work: 1. Mutation and hillclimbing' in Manner R. & Manderick, B. (eds.), *Parallel Problem Solving from Nature 2*. North – Holland.

Murata, T., Ishibuchi, H., Tanaka, H., (1996). "Genetic algorithms for slowshop scheduling problems", *Computer & Industrial Engineering*, vol. 30, no. 4, pp. 1061-1071.

Nagase, H., et al., (1983). 'High Performance Induction Motor drive system using a PWM inverter', *Record IEEE – IAS Annual Meeting, Mexico*, pp. 596 – 603.

Nangsue, P., Pillay, P. & Conry, S.E., (1999). 'Evolutionary Algorithms for Induction Motor Parameter Determination'. *IEEE Transactions on Energy Conversion*, Vol.14, No. 3.

Neumaier, A., Bomze, I., Emiris, I., Wolsey, L., (2006). 'Global optimization and constraint satisfaction'. *Proceedings of GICOLAG workshop (of the research project Global Optimization, Integrating Convexity, Optimization, Logic Programming and Computational Algebraic Geometry*, online available at <http://www.mat.univie.ac.at/~neum/glopt.html> [accessed 2009 - 04 – 02].

Nordin, K.B., Novotny, D.W. & Zinger, D.S., (1985). 'The Influence of Motor Parameter Deviations in Feedforward Field Orientation Drive Systems', *IEEE Transactions on Industry Applications*, Vol. IA – 21, No. 4.

Novotny, D.W. & Lipo T.A., (1996). 'Vector Control and Dynamics of AC Drives', Oxford University Press, New York.

Ohtani, T., (1983). 'Torque control using the flux derived from magnetic energy in induction motors driven by static converter', *Record IPEC*, Tokyo, pp. 696 – 707.

Okaeme, N., (2008). 'Automated Robust Control System Design for Variable Speed Drives', *PhD Thesis*, University of Nottingham, UK

Passino, K., M., (2006). 'Biomimicry for optimisation, control and automisation'. Springer – Verlag, London.

Pearl, J., (1984). 'Heuristics: Intelligent Search Strategies for Computer Problem Solving'. The Addison-Wesley series in artificial intelligence. Addison-Wesley Pub (Sd), ISBN: 978-0-20105-594-8.

Peixoto, Z.M.A. & Seixas, P.F., (2000). 'Electrical parameter estimation considering the saturation effects in induction machines'. *In Proceedings of 2000 IEEE – PESC annual meeting* (pp. 1563 – 1568).

Perry, M.J., (2006). 'A Modified Genetic Algorithm Approach to System Identification with Structural and Offshore Applications', PhD Thesis, National University of Singapore.

Perry, M.J., Koh, C.G. & Choo, Y.S., (2006a). 'Modified genetic algorithm strategy for structural identification', *Elsevier Journal "Computers and Structures"* 84 (2006) 529 – 540.

- Potts, J.C., Giddens, T.J. & Yadav, S.B., (1994). 'The development and evaluation of an improved genetic algorithm based on migration and artificial selection', *IEEE Trans Syst Man Cyber*; 24(1); 73-86.
- Ribeiro, L.A.de S., Jacobina, C.B., Lima, A.M.N., (1997). 'The influence of the slip and the speed in the parameter estimation of induction machines', in *Proc. PESC Conf. Rec.*, pp. 1068 – 1074.
- Roach, A. & Nagi, R., (1996). 'A hybrid GA-SA algorithm for just-in-time scheduling of multi – level assemblies', *Computer & Industrial Engineering*, vol. 30, no. 4, pp. 1047-1060.
- Rowan, T., Kerkman, R. & Leggate, D., (1991). 'A simple on – adaptation for indirect field orientation of an induction machine', *IEEE Transactions on Industry Applications*, Vol. 37, pp. 720 – 727.
- Schaffer, J.D., Caruana, R.A., Eshelman, L.J. & Das, R., (1989). 'A study of control parameters affecting online performance of genetic algorithms for function optimization', in Schaffer, J.D., (ed.), *Proceedings of the 3rd International Conference on Genetic Algorithms*, Morgan Kaufmann.
- Schaffer, J.D., Eshelman, L., J. & Offutt, D., (1990). 'Spurious correlations and premature convergence in genetic algorithms', In *Proceedings of the First Workshop on Foundations of Genetic Algorithms (FOGA)*, pages 102-112.
- Schumacher, W. & Leonhard, W., (1983), 'Transistor – fed ac – servo drive with microprocessor control', in *Conf. Record IPEC*, Tokyo, pp. 1465 – 1476.
- Schwefel, H., P., (1997). 'Advantages (and disadvantages) of evolutionary computation over other approaches, in [Back, 1997], pp. A1.3:1 – 2.
- Seok, J.K., Moon, S.I. & Sul, S.K., (1997). 'Induction parameter identification using pwm inverter at standstill', *IEEE Trans. Energy Conv.* Vol. 12, pp. 127 – 132, June 1997.

Shaw, S.R., (1997). 'Numerical Methods for Identification of Induction Motor Parameters'. *Master's Thesis, MIT*.

Sherling, H., (1988a). 'Self – commissioning – a novel feature of modern inverter – fed induction motor drives'. *IEE Conf. on Power Electronics and Variable Speed Drive*, pp. 287 – 290.

Sherling, H., (1988b). 'Fast and reliable commissioning of ac variable speed drives by self – commissioning'. *IEEE IAS Annual Meeting Conf. Rec.*, pp. 489 – 492.

Slemon, G.R., Straughen, A., (1981). 'Electric Machines', *Addison-Wesley Publ., Edition 3*.

Sokola, M. & Levi, E., (2000). 'A novel induction machine model and its application in development of an advanced vector control scheme'. *International Journal of Electrical Engineering Education*, Vol. 27, Issue 3.

Spears, W.M. & De Jong, K.A., (1991). 'On the virtues of parameterized uniform crossover' in Belew, R.K. & Booker, L.B. (eds.), *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann.

Srinivas, M. & Patnaik, L.M., (1991). 'Learning neural network weights using genetic algorithms – improving performance by search – space reduction'. *IEEE International Joint Conference on Neural Networks*, pp. 2331-2336.

Srinivas, M. & Patnaik, L.M., (1994). 'Adaptive probabilities of crossover and mutation in genetic algorithms', *IEEE Transactions on Systems, Man and Cybernetic*, Vol. 24, No. 4, pp. 656 – 667.

Sugimoto, H. & Tamai, S., (1987). 'Secondary resistance identification of an induction motor applied model reference adaptive systems and its characteristics', *IEEE Transactions on Industry Applications*, Vol. IA 23, pp. 296 – 303.

Sumner, M., (1990). 'Vector Controlled Induction Motor Drive Using Transputer Parallel Processors', *PhD Thesis, The University of Nottingham, UK*

Sumner, M. & Asher, G.M., (1991). 'The Experimental Investigation of Multiparameter Identification Methods for Cage Induction Motors', *EPE Florence*, Vol.3, pp. 389 – 394.

Sumner, M. & Asher, G.M., (1993). 'Auto commissioning for Voltage – referenced Voltage – fed Vector – controlled Induction Motor Drives', *IEE proceedings on Electric Power Applications*, Vol. 140, No. 3.

Toliyat, H.A. & Hosseiny, A.A.GH., (1993). 'Parameter estimation algorithm using spectral analysis for vector controlled induction motor drives', in *Proc. IEEE International Symposium on Industrial Electronics*, pp. 90 – 95.

Trentin, A., Zanchetta, P., Wheeler, P. & Clare, J., (2006). 'A New Method for IMs Parameter Estimation Using GAs and Transient Speed Measurements', *IAS 2006 conference*, Tampa Florida.

Trzynadlowski, A., (2001). 'Control of Induction Motors', *Academic Press*, London.

Turl, G., (2002). 'A Synchronised Multi – Motor Control System Using Hybrid Sensorless Induction Motor Drives', *PhD Thesis, The University of Nottingham, UK*.

Tzou, Y.Y. & Lin, Y.F., (1997). 'Auto – Tuning Control of Self – Commissioning Electric Drives'. *Industrial Electronics, Control and Instrumentation*, IECON, vol. 2, pp. 483-487.

Ursem, R.K., (2003). 'Models for Evolutionary Algorithms and Their Applications in System Identification and Control Optimisation', *PhD Thesis, University of Aarhus, Denmark*.

Ursem, R.K. & Vadstrup, P., (2004). 'Parameter Identification of Induction Motors Using Stochastic Optimisation Algorithms'. *Applied Soft Computing* 4, pp. 49 – 64.

Vladu, E., E., (2003). 'Contributions in using genetic algorithms in engineering', *PhD Thesis, "Politehnica" University of Timisoara.*

Walker, M., (2001). 'Introduction to genetic programming', Tech. Rep., Massey University.

Weise, T., (2009). 'Global Optimisation Algorithms – Theory and Application', 2nd edition, online e-book.

Whitely, D., (1993). 'A Genetic Algorithm Tutorial', Technical Report CS-93-103, Colorado State University.

Wildi, T. (2006). 'Electrical Machines, Drives and Power Systems'. *Upper Saddle River, N.J. : Pearson Prentice Hall, 6th Edition, 2006.*

Willis, J.R., Brock, G.J. & Edmonds, J.S., (1989). 'Derivation of induction motor models from standstill frequency response tests', *IEEE Trans. Energy Conv.* Vol. 4, pp. 608 – 613.

Wishart, M.T. & Harley, R.G., (1995). 'Identification and Control of Induction Machines Using Artificial Neural Networks'. *IEEE Transactions on Industry Applications.*

Wright, A., H., (1991). 'Genetic Algorithms for Real Parameter Optimisation', *In Foundations of Genetic Algorithms*, J. E. Rawlins (Ed.), Morgan Kaufmann, pp. 205-218.

Yeong – Hwa, C., Hung – Wei, L. & Gong, C., (2004). 'Compensation of Hall effect sensor of PWM switching control', *Journal of Magnetism and Magnetic Materials* 282, 307-310.

Publications

Kampisios, K., Zanchetta, P., Gerada, C., Trentin, A. & Jasim, O., (2008a). 'Induction Motor Parameters Identification using Genetic Algorithms for Varying Flux Levels', *13th International Power Electronics and Motion Control Conference*, EPE-PEMC 2008, 1-3 Sept., Page(s): 887 – 892.

Kampisios, K., Zanchetta, P., Gerada, C. & Trentin, A., (2008b). 'Identification of Induction Machine Electrical Parameters Using Genetic Algorithms Optimization', *Industry Applications Society Annual Meeting, IAS '08*. IEEE, 5 – 9 Oct., Page(s): 1 – 7.