UNIVERSITY of LIVERPOOL

MACHINE LEARNING FOR PARAMETER IDENTIFICATION OF ELECTRIC INDUCTION MACHINES

Thesis submitted in accordance with the
requirements of the University of Liverpool
for the degree of Doctor of Philosophy

in

Electrical Engineering and Electronics

by

W. F. Kent
BSc(Eng), MSc in Aerospace Propulsion,
MSc in Radio Frequency Communications , PGCE, AKC, MIEE

January 2003

PAGINATED
BLANK PAGES
ARE SCANNED AS
FOUND IN
ORIGINAL
THESIS

NO
INFORMATION
MISSING

Dedicated to my late mother, Barbara L. Kent. Tempora mutantur, et nos mutamur in illis.

Acknowledgements

I would like to take this opportunity to thank my supervisor, Professor Q. H. Wu, for the time, patience and guidance throughout the duration of this project. I would also like to thank Mr. David Turner whose knowledge, interest and experience in electrical machines was much appreciated throughout the laboratory work. A special thanks must also extend to Prof. K. S. Huang, of Guangdong University of Technology, China. Appreciation is also due to Mr. Pu Sun and the splendid work done by the technicians in the Department of Electrical Engineering and Electronics. Finally, I would like to thank the EPSRC for supporting this work at the University of Liverpool.

Abstract

MACHINE LEARNING FOR PARAMETER IDENTIFICATION OF ELECTRIC INDUCTION MACHINES

by

W. F. Kent

This thesis is concerned with the application of simulated evolution (SE) to the steady-state parameter identification problem of a simulated and real 3-phase induction machine, over the no-load direct-on-line start period.

In the case of the simulated 3-phase induction machine, the Kron's two-axis dynamic mathematical model was used to generate the real and simulated system responses where the induction machine parameters remain constant over the entire range of slip. The model was used in the actual value as well as the per-unit system, and the parameters were estimated using both the genetic algorithm (GA) and the evolutionary programming (EP) from the machine's dynamic response to a direct-on-line start. Two measurement vectors represented the dynamic responses and all the parameter identification processes were subject to five different levels of measurement noise.

For the case of the real 3-phase induction machine, the real system responses were generated by the real 3-phase induction machine whilst the simulated system responses were generated by the Kron's model. However, the real induction machine's parameters are not constant over the range of slip, because of the nonlinearities caused by the skin effect and saturation. Therefore, the parameter identification of a real 3-phase induction machine, using EP from the machine's dynamic response to a direct-on-line start, was not possible by applying the same methodology used for estimating the parameters of the simulated, constant parameters, 3-phase induction machine.

Contents

List of Figures xi

1 Introduction 1
- 1.1 History of the Problem . . . 1
- 1.2 The 3-Phase Induction Machine, Vector Control and Parameter Identification . . . 3
- 1.3 The Mathematical Model of a 3-Phase Induction Machine . . . 4
- 1.4 The Optimization Problems for Parameter Identification of a 3-Phase Induction Machine . . . 5
- 1.5 Simulated Evolution . . . 6
- 1.6 Applying Simulated Evolution to the Parameter Identification Problem in a Simulated 3-Phase Induction Machine . . . 6
- 1.7 Applying Simulated Evolution to the Parameter Identification Problem of a Real 3-Phase Induction Machine . . . 7
- 1.8 Scope and Purpose of this Research . . . 9
- 1.9 Overview of Thesis . . . 9

2 The Dynamic Model of a Balanced, Symmetrical 3-Phase Induction Machine 11
- 2.1 Introduction . . . 11
- 2.2 The 3-Phase Induction Machine . . . 11
- 2.3 The 3-Phase Induction Machine Voltage Equations . . . 13
- 2.4 Voltage Equations of Machine Variables . . . 16
- 2.5 The Background to Reference-Frame Transformation Analysis . 20
- 2.6 Equations of Transformation for Stator Circuits . . . 22
- 2.7 Stator Circuit Variables Transformed to the Arbitrary Reference Frame . . . 24
  - 2.7.1 Stator Resistive Elements . . . 24
  - 2.7.2 Stator Inductive Elements . . . 25
- 2.8 Equations of Transformation for Rotor Circuits . . . 27
- 2.9 Rotor Circuit Variables Transformed to the Arbitrary Reference Frame . . . 28

2.9.1 Rotor Resistive Elements 28
2.9.2 Rotor Inductive Elements 29
2.10 Voltage Equations in Arbitrary Reference-Frame Variables . . . 31
2.11 Conclusion 33

3 The Dynamic Mathematical Model of a Symmetrical Induction Machine for Simulation and Analysis 35
3.1 Formation of Currents as State Variables 35
3.2 The Voltage Transformations 37
3.3 The Current Transformations 38
3.4 The Instantaneous Electrical Torque Equation 38
3.5 The Equation of Motion 39
3.6 The Induction Machine's Differential Equations Suitable for Simulation 39

4 The Simulated Evolution Techniques Applied to the Optimization Problem of Parameters Identification 41
4.1 Introduction 41
4.2 The Processes Involved in Optimization 42
4.2.1 The Global Optimization Problem 43
4.3 Traditional Optimization Methods and their Limitations 44
4.4 Natural Evolution 45
4.4.1 The Neo-Darwinian Paradigm 46
4.4.2 The Genotype and the Phenotype: The Optimization of Behaviour 46
4.4.3 Sexual Reproduction 50
4.4.4 Sexual Selection 52
4.5 Simulated Evolution 53
4.6 Genetic Algorithms 55
4.6.1 A Brief Historical Development 55
4.7 Algorithms Description 56
4.7.1 Selection 57
4.7.2 Crossover 58
4.7.3 Mutation 58
4.8 Convergence Properties of Genetic Algorithms 59
4.8.1 Genetic Algorithms without Bit Mutation 60
4.8.2 Genetic Algorithms with Bit Mutation 63
4.9 Evolutionary Programming 66
4.9.1 A Brief Historical Development 66
4.10 Evolutionary Programming 66
4.11 Algorithms Description 68
4.12 Convergence Properties of Evolutionary Programming 69
4.13 Conclusion 73

5 The Induction Machine Simulations **74**
5.1 Introduction 74
5.2 The Induction Machine's Differential Equations Used for Simulation 75
5.3 Simulation of a 3-Phase Induction Machine 75
5.4 The Dynamic Model Notation Used for the Induction Machine . 76
5.5 Identification of an Induction Machine's Dynamic Model Parameters applying Genetic Algorithms 78
5.6 A Multiparameter, Mapped, Fixed-Point Coding 80
5.7 Identification of an Induction Machine's Dynamic Model Parameters Applying Evolutionary Programming 81
5.8 The Induction Machine Model in the Per-Unit System 83
5.9 The Induction Machine's Equations in the Per-Unit System . . . 85
5.10 Implementation of the Simulated Evolution and its Simulation Results 85
5.10.1 Tabulation of Simulated Evolution Simulation Results . . 88
5.11 The Induction Machine Actual Value Model Responses applying the Measurement Vector $Y1$ 93
5.11.1 Simulation and Response Results 93
5.12 The Induction Machine Actual Value Model Responses applying the Measurement Vector $Y2$ 100
5.12.1 Simulation and Response Results 100
5.13 The Induction Machine Per-Unit System Model Responses applying the Measurement Vector $Y1$ 107
5.13.1 Simulation and Response Results 107
5.14 Conclusion 114

6 Parameter Identification of a Real Induction Machine **116**
6.1 Introduction 116
6.2 The Transient Start-Up Data Acquisition Method for a Balanced 3-Phase Symmetrial Induction Machine 118
6.3 The Rotor Shaft Speed Calculations 119
6.4 The Real Induction Machine's Specifications 121
6.5 Identification of a Real 3-Phase Induction Machine's Parameters Applying Evolutionary Programming 122
6.6 Implementation of EP for the Parameter Identification of a Real Induction Machine 125
6.6.1 The Initial Stator Current and Voltage Phase Angle Determination 125
6.6.2 Simulation Results 128
6.7 Limitations of the Balanced, Symmetrical 3-Phase Induction Machine Mathematical Model 133
6.7.1 The Kron's Two-Axis Dynamic Mathematical Model . . 133

6.7.2 The Nonlinear Magnetic Circuit . . . 135
6.7.3 Skin Effect . . . 137
6.7.4 Temperature Effect . . . 137
6.7.5 Harmonic Content of the M.M.F. Wave . . . 137
6.7.6 Friction and Windage Power Losses . . . 137
6.8 Discussion . . . 138

7 Discussion and Conclusions **141**
7.1 Introduction . . . 141
7.2 Research Synopsis . . . 141
7.3 Effectiveness of Simulated Evolution . . . 143
7.3.1 Parameter Identification of a Simulated 3-Phase Induction Machine . . . 143
7.3.2 Parameter Identification of a Real 3-Phase Induction Machine . . . 144
7.4 Conclusions . . . 145

A The Fundamental Processes of Simple GA **146**

B Glossary **149**

C Lists of Symbolic Notation **152**
C.1 The 3-Phase Induction Machine and Mathematical Model Symbols 152
C.1.1 Greek Letters . . . 155
C.2 Optimization, Simulated Evolution and Stochastic Operator Symbols . . . 156
C.2.1 Greek Letters . . . 156

Bibliography **159**

List of Figures

2.1 Winding Arrangement of a Two-Pole, 3-Phase, Wye-Connected Symmetrical Induction Machine 12
2.2 Transformation for Stator Circuits Portrayed by Trigonometric Relationships 17
2.3 Transformation for Rotor Circuits Portrayed by Trigonometric Relationships 28

5.1 Induction machine output with noise $\sigma^2 = 0.2$ 92
5.2 Stator current start-up response with noise of case 1 93
5.3 Electric torque start-up response with noise of case 1 94
5.4 Rotor speed start-up response with noise of case 1 95
5.5 Stator current start-up response with noise of case 3 95
5.6 Electric torque start-up response with noise of case 3 96
5.7 Rotor speed start-up response with noise of case 3 96
5.8 Stator current start-up response with noise of case 1 97
5.9 Electric torque start-up response with noise of case 1 97
5.10 Rotor speed start-up response with noise of case 1 98
5.11 Stator current start-up response with noise of case 3 98
5.12 Electric torque start-up response with noise of case 3 99
5.13 Rotor speed start-up response with noise of case 3 99
5.14 Stator current start-up response with measurement noise of case 1 100
5.15 Electric torque start-up response with measurement noise of case 1 101
5.16 Rotor speed start-up response with measurement noise of case 1 102
5.17 Stator current start-up response with measurement noise of case 3 102
5.18 Electric torque start-up response with measurement noise of case 3 103
5.19 Rotor speed start-up response with measurement noise of case 3 103
5.20 Stator current start-up response with measurement noise of case 1 104
5.21 Electric torque start-up response with measurement noise of case 1 104
5.22 Rotor speed start-up response with Measurement noise of case 1 105
5.23 Stator current start-up response with measurement noise of case 3 105
5.24 Electric torque start-up response with measurement noise of case 3 106
5.25 Rotor speed start-up response with measurement noise of case 3 106

5.26 Stator current start-up response with measurement noise of case 1 107
5.27 Electric torque start-up response with measurement noise of case 1 108
5.28 Rotor speed start-up response with measurement noise of case 1 109
5.29 Stator current start-up response with measurement noise of case 3 109
5.30 Electric torque start-up response with measurement noise of case 3 110
5.31 Rotor speed start-up response with measurement noise of case 3 110
5.32 Stator current start-up response with measurement noise of case 1 111
5.33 Electric torque start-up response with measurement noise of case 1 111
5.34 Rotor speed start-up response with measurement noise of case 1 112
5.35 Stator current start-up response with measurement noise of case 3 112
5.36 Electric torque start-up response with measurement noise of case 3 113
5.37 Rotor speed start-up response with measurement noise of case 3 113

6.1 Data Acquisition fron a 3-Phase Induction Machine in the Laboratory 117
6.2 Flowchart for the Rotor Shaft Speed Algorithm 120
6.3 Stator phase voltage response from the real induction machine . 126
6.4 Stator current response from the real induction machine 127
6.5 Stator current start-up response 130
6.6 Rotor speed start-up response 130
6.7 Stator current start-up response 131
6.8 Rotor speed start-up response 131

MACHINE LEARNING FOR PARAMETER IDENTIFICATION OF ELECTRIC INDUCTION MACHINES

by

W. F. Kent

Chapter 1

Introduction

1.1 History of the Problem

Induction machines have many advantages and applications. Their advantages are due to their size, weight, cost, reliability and efficiency [1]. Their applications have been used in fixed speed and variable-speed servo and drivers. The method of vector control allows high-performance control of torque, speed, or position to be achieved from an induction machine [2, 3]. It is important to know the machine parameters for both evaluation and application of these machines.

The performance estimation of a fixed speed 3-phase induction machine can be calculated from its equivalent circuit and its resulting equivalent circuit's mathematical equations [4]. The resistances and reactances of the equivalent circuit are recognised as the parameters of the induction machine. The induction machine parameters are normally obtained from the manufacturer who determines them from the standard locked rotor and no-load tests [1]. These parameter values are only valid for the steady-state operation of the 3-phase induction machine and under conditions governed by the machine's specifications. Therefore, at different machine operating points, the parameter values

vary because of temperature changes and the nonlinearities caused by the skin effect and saturation. This will effect the performance of vector control, which will be discussed in more detail in the next section, as the controller parameters will not match the real induction machine's parameters. Hence, parameter identification for the induction machine's dynamic model is necessary for high performance vector control.

Parameter or/and states identification algorithms have been used with great success for modelling and control of a.c. machines. The oberved stimulus-response data is often used to identify the parameters and a criterion of the response data is used as an objective function to be minimized, which is typically a function of the squared predictive errors. The Recursive Least-Square Method (RLS) has been used for this purpose [4, 5, 6, 7]. Among them, Lima et al, [4], estimated the nonlinear parameters of a steady-state induction machine. The Maximum Likelihood Parameter Estimation (ML), [8, 9], and Extended Kalman Filter (EKF), [10, 11, 13], are also techniques applied for parameter or/and states identification for a.c. machines. Among them, Tsai et al, [8], have applied ML for the estimation of synchronous machine parameters for small disturbance operating data and Atkinson et al, [10], have defined the rotor resistance as a state variable in EKF. Ma and Wu, [14], have applied Evolutionary Programming (EP) to identify the parameters of a synchronous generator connected in a two-machine system. Genetic Algorithms (GA) is a method used most recently [15, 16, 17, 18]. For example, Nangsue et al, [15], have determined the parameters of an induction machine using both GA and Genetic Programming (GP); Dong-Hyeok et al, [16], have applied niching GA for the design of an induction machine for use in an electric vehicle; Charette et al, [17], have used GA for in-situ efficiency measurements of an induction machine; Alonge et at, [18], have estimated the electrical and mechanical parameters of the induction machine's model.

1.2 The 3-Phase Induction Machine, Vector Control and Parameter Identification

As mentioned in the previous section, vector control allows high-performance control of torque, speed or position. Vector control provides decoupled control of the rotor flux magnitude and the torque producing rotor current, with a fast, near-step response change in torque achievable. The fast torque response obtained using vector control is achieved by measuring or estimating the magnitude and position of the rotor flux, then the control sequence alters the magnitude, frequency and phase of the three stator currents (almost) instantaneously in such a way that the frequency, magnitude and phase of the rotor current wave jumps suddenly from one steady-state to another. This is achieved by providing each phase with a fast-acting closed-loop current controller and without altering the amplitude or position of the resultant flux linkage relative to the rotor, i.e. without altering the stored energy significantly. Thus, the flux density wave remains the same while the rotor current and space-lag angle between the peak radial flux density and peak rotor currents change instantaneously to their new steady-state values, corresponding to the new steady-state slip and torque.

There are two methods of vector control, direct and indirect. The direct method measures the flux by search coils or Hall effect devices that are inserted into the air gap of the induction machine. The indirect method relies entirely on the mathematical model of the induction machine. This method uses a slip speed calculation to estimate the speed of rotation of the flux relative to the rotor and integrates this to find the slip position. Adding this to the rotor position measurement gives the rotor position.

Vector control is thus a control process. Both the direct and indirect methods use the multi-variable, non-linear, mathematical model of the induction

machine, and implementation of complex control algorithms requiring a large number of fast computations to be continually carried out so that the correct instantaneous voltages are applied to each stator winding. Thus, both methods use current feedback as an integral part of each stator current controller and, hence, digital signal processor (DSP) circuitry is used in the drive circuit.

Since the vector control indirect method uses only the induction machine's mathematical model in slip speed calculations for estimating the magnitude and position of the rotor flux in its complex control algorithms, it is important to know the induction machine's parameters, particularly the rotor resistance, at the required torque and speed instances. It is not possible to obtain directly on-line the induction machine's parameters for use in the indirect method's control algorithm. The identification of the 3-phase induction machine's parameters by Simulated Evolution (SE), which is the essence of the research work contained in this Thesis, is a computational method for identifying the parameters. SE can be used to identify the machine's parameters at particular torque and speed instances, but the computational time in so doing is much greater than the instantaneous computations required by the vector control indirect method. However, it is possible for SE to identify a range of induction machine parameters off-line, over the required torque and speed instances, and store these parameters on a data base for use with the indirect method's control algorithm on-line.

1.3 The Mathematical Model of a 3-Phase Induction Machine

For the simulation of the 3-phase inducion machine we shall use the Kron's two-axis dq dynamic mathematical model. This model will generate the state variables, namely the 3-phase stator currents and the rotor speed calculations,

during the simulation runs required by the objective function. The model is based on a 3-phase wye-connected symmetrical induction machine. It has identical stator windings with equivalent turns and resistances and also has rotor windings with equivalent turns and resistances. The air gap is uniform. The model transforms the 3-phase induction machine's axes into two perpendicular axes, the dq axes, thus eliminating the time-varying mutual inductances between the stator and rotor circuits from the induction machine's voltage equations [21, 23]. Furthermore, the model operates under the assumptions that the 3-phase induction machine has linear magnetic circuits and its resistances do not change due to temperature and frequency changes.

1.4 The Optimization Problems for Parameter Identification of a 3-Phase Induction Machine

The process of optimization is to improve performance towards some optimal point or points. The observed stimulus-response data, obtained from a simulated and a real 3-phase induction machine, shall be used to identify the machine's parameters and a criterion of the response data will be used as an objective function to be minimized. This will be a function of the squared predictive errors between the real system and simulated system response data. Traditional optimization techniques have been used extensively for parameter identification problems, but each of these methods has been hindered by its own peculiar set of drawbacks. Calculus-based methods are local in scope and, because of their dependence on functional derivatives, cannot deal with noisy data pollution. Enumerative schemes rely on search spaces that maybe so vast, that a slow rate of convergence results, or a search space too large to search. Random search algorithms lack efficiency due to their stochastic and random

approach to orientation within the search space for the optimal point or points. The following section will discuss why Simulated Evolution is applied in this research in order to by-pass the problems incurred in traditional optimization methods.

1.5 Simulated Evolution

Simulated Evolution (SE) is a stochastic search optimization technique. It differs from conventional optimization techniques in that it involves direct manipulation of a coding, it is a population-wide search rather than a point search, and the search is through sampling and using stochastic operators as opposed to deterministic rules. The evolutionary technique is based upon generating successive populations of feasible solutions based on the Darwinian evolutionary theory. SE generally takes the following form: a population of trial solutions to a particular problem is generated. Each of the initial solutions is scored with respect to a particular objective function. Through a variety of possible code operations, parents generate offspring which are scored in a similar fashion, the best subset of the solution are retained to serve as parents for successive generations. The population iteratively adapts its behaviour in view of the present objective. Two paradigms of SE are applied in this research: Genetic Algorithms and Evolutionary Programming.

1.6 Applying Simulated Evolution to the Parameter Identification Problem in a Simulated 3-Phase Induction Machine

Software has been developed, incorporating the Kron's two-axis dynamic mathematical model, which will generate the state variables for the simulated

3-phase induction machine. The 'plant' responses are generated by the Kron's two-axis dynamic mathematical model, without measurement noise, and using the machine's known parameters, and are termed the simulated system responses. The electrical parameters and mechanical coefficients, used in the mathematical model, were taken from the specifications of a real 3-phase induction machine [12]. The 'real' system responses are formed by the Kron's two-axis dynamic mathematical model simulation, using the same 3-phase induction machine specifications as previously mentioned, but with the addition of measurement noise added in varying levels to different simulation runs.

The 'real' and simulated system responses shall form part of the objective function in the optimization process. The optimization process will be used by the SE for the electrical parameter identification of a 3-phase induction machine. Moreover, the electrical parameters identified by the SE will be used in the Kron's two-axis dynamic mathematical model, using the same 3-phase induction machine specifications which will form the responses generated by the identified parameters. These responses shall be called 'EP' (if the parameters were identified by evolutionary programming) or 'GA' (if the parameters were identified by genetic algorithms). This will enable the research to compare the responses generated by the 'plant' with those responses generated by the identified parameters, either EP or GA, for different levels of measurement noise.

1.7 Applying Simulated Evolution to the Parameter Identification Problem of a Real 3-Phase Induction Machine

The state variables for the real 3-phase induction machine shall be generated by the machine from a direct-on-line start and collected by the data acquisition

system, over the data acquisition period, and stored on a data file. These responses shall be called the 'plant' and will form the real system responses for use in the objective function used in the optimization process.

The simulated system responses are formed by the Kron's two-axis dynamic mathematical model simulation. The electrical and mechanical specifications needed by the model were taken from the real 3-phase induction machine's nameplate. Neither the electrical parameters nor the mechanical coefficients were known. Therefore, various ranges of electrical parameters and mechanical coefficients shall be investigated to allow the EP to have ample scope for the parameter identification process. However, the electrical parameters of a 3-phase induction machine can be determined via the locked rotor and no-load tests. The reasons why this latter option was not followed and the parameter identification process was chosen shall now be explained.

The responses generated by both the real and simulated 3-phase induction machines will be over the transient start-up period. The real 3-phase induction machine's electrical parameters are normally determined via the locked rotor and no-load tests. In Chapter 6, Section 8, we will discuss why these parameters are no longer valid during transient periods of the machine's operations in the Kron's two-axis dynamic mathematical model. Therefore, the parameter identification of the real 3-phase induction machine shall be considered a black box problem. Moreover, the choices of the electrical parameters and mechanical coefficient ranges for use in the EP for the parameter identification process were chosen by logical deductions under the available criteria.

Once again, the real and simulated system responses shall form part of the objective function in the optimization process. The optimization process shall be used by the EP for the parameter identification of the real 3-phase induction machine. The parameters identified by the EP shall be used in the Kron's two-axis dynamic mathematical model to produce the responses generated by the identified parameters. These responses shall be called 'EP'.

This will again enable us to compare the responses generated by the 'plant' with those generated by the identified parameters 'EP'.

1.8 Scope and Purpose of this Research

This work focuses on the parameter identification of a 3-phase induction machine by the application of SE in the optimization process. The following objectives have been set for this work.

1. To identify the parameters of a simulated 3-phase induction machine where both the 'real' and simulated system responses are generated by the Kron's two-axis dynamic mathematical model over the transient start-up period, and where the 'real' system responses contain measurement noise added in varying levels to different simulation runs.

2. To identify the parameters of a simulated 3-phase induction machine, identical to the previous objective, but in this case using the induction machine mathematical model in the per-unit system.

3. To identify the parameters of a real 3-phase induction machine, where the real system responses are generated by the real 3-phase induction machine and the simulated system responses by the Kron's two-axis dynamic mathematical model over the transient start-up period.

1.9 Overview of Thesis

In Chapter 2 the dynamic model of a balanced, symmetrical 3-phase induction machine is developed and presented. How this model shall be used for simulation and analysis is presented in Chapter 3. Chapter 4 describes the limitations of traditional optimization methods and further describes how the processes involved in natural evolution has led to the development of simulated

evolution that bypasses the problems incurred with traditional optimization methods in search, optimization and machine learning. Chapter 5 presents the EP parameter identification process for the induction machine simulations, whilst Chapter 6 describes the attempts and processes involved in the EP parameter identification of a real induction machine. Chapter 7 presents the conclusions from this research and discusses suggestions for further work in this area.

Chapter 2

The Dynamic Model of a Balanced, Symmetrical 3-Phase Induction Machine

2.1 Introduction

In this chapter we will discuss the formulation of the voltage equations of a balanced, symmetrical 3-phase induction machine. We will look at how these voltage equations can be made tractable and solvable by applying the reference-frame transformations to them.

2.2 The 3-Phase Induction Machine

The winding arrangement of a 2-pole, 3-phase, wye-connected symmetrical induction machine is shown in Figure 2.1. The stator windings are identical with equivalent turns N_s and resistance R_s. The machine has a squirrel cage rotor but can be approximated by assuming it has 3 identical windings with equivalent turns N_r and resistance R_r. The air gap of the induction machine is

uniform and it is assumed that the stator and rotor windings may be approximated as sinusoidally distributed windings. The factors neglected by the model are the nonlinear magnetic circuit, changes in resistances due to temperature and frequency changes and harmonic content of the M.M.F. wave.

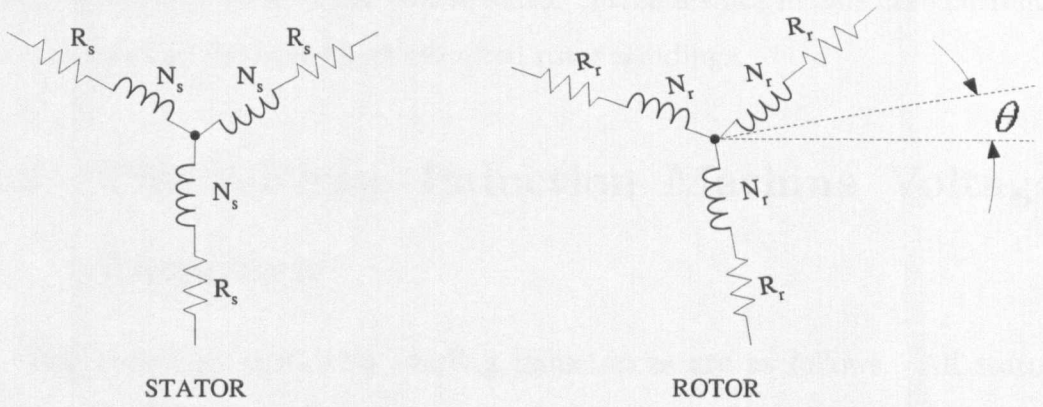

Figure 2.1: Winding Arrangement of a Two-Pole, 3-Phase, Wye-Connected Symmetrical Induction Machine

The induction machine under investigation is operated as a motor with the stator windings connected to a balanced 3-phase source and the rotor windings short-circuited. The principle of operation in this mode is as follows. With balanced 3-phase current flowing in the stator windings, a rotating air-gap Magnetomotive Force, MMF, is established which rotates about the air-gap at a speed determined by the frequency of the stator currents and the number of poles. If the rotor speed is different from the speed of this rotating M.M.F., balanced 3-phase currents will be induced in the short-circuited rotor windings. The frequency of the rotor currents corresponds to the difference in the speed of the rotating M.M.F. due to the stator currents and the speed of the rotor. The induced rotor currents will in turn produce an air-gap M.M.F. which rotates relative to the rotor at a speed corresponding to the frequency of the rotor currents. The speed of the rotor air-gap M.M.F. superimposed upon the rotor speed is the same speed as that of the air-gap M.M.F. established by the

currents flowing in the stator windings. These two air-gap M.M.F.s rotating in unison may be thought of as two synchronously rotating sets of magnetic poles. Torque is produced due to an interaction of these magnetic poles. It is clear, however, that torque is not produced when the rotor is running in synchronism with the air-gap M.M.F. due to the stator currents since in this case currents are not induced into the short-circuited rotor windings.

2.3 The 3-Phase Induction Machine Voltage Equations

The induction machine's winding inductances are as follows. All stator self-inductances are equal, that is $L_{rsrs} = L_{ysys} = L_{bsbs}$ with

$$L_{rsrs} = L_{ls} + L_{ms} \tag{2.3.1}$$

$$L_{ysys} = L_{ls} + L_{ms} \tag{2.3.2}$$

$$L_{bsbs} = L_{ls} + L_{ms} \tag{2.3.3}$$

where L_{ms} is the stator magnetizing inductance and L_{ls} is the stator leakage inductance.

All stator-to-stator mutual inductances are the same

$$L_{rsys} = -\frac{1}{2}L_{ms} \tag{2.3.4}$$

$$L_{rsbs} = -\frac{1}{2}L_{ms} \tag{2.3.5}$$

$$L_{ysbs} = -\frac{1}{2}L_{ms} \tag{2.3.6}$$

$$L_{ysrs} = -\frac{1}{2}L_{ms} \tag{2.3.7}$$

$$L_{bsrs} = -\frac{1}{2}L_{ms} \tag{2.3.8}$$

$$L_{bsys} = -\frac{1}{2}L_{ms} \tag{2.3.9}$$

It follows that the rotor self-inductances are equal with

$$L_{rrrr} = L_{lr} + L_{mr} \tag{2.3.10}$$

$$L_{yryr} = L_{lr} + L_{mr} \tag{2.3.11}$$

$$L_{brbr} = L_{lr} + L_{mr} \tag{2.3.12}$$

where L_{mr} is the rotor magnetizing inductance and L_{lr} is the rotor leakage inductance.

Likewise, all the rotor-to-rotor mutual inductances are the same

$$L_{rryr} = -\frac{1}{2}L_{mr} \tag{2.3.13}$$

$$L_{rrbr} = -\frac{1}{2}L_{mr} \tag{2.3.14}$$

$$L_{yrbr} = -\frac{1}{2}L_{mr} \tag{2.3.15}$$

$$L_{yrrr} = -\frac{1}{2}L_{mr} \tag{2.3.16}$$

$$L_{brrr} = -\frac{1}{2}L_{mr} \tag{2.3.17}$$

$$L_{bryr} = -\frac{1}{2}L_{mr} \tag{2.3.18}$$

Expressions for mutual inductances between stator and rotor windings are as follows

$$L_{rsrr} = L_{sr}\cos\theta_r \tag{2.3.19}$$

$$L_{rsyr} = L_{sr}\cos(\theta_r + \frac{2\pi}{3}) \tag{2.3.20}$$

$$L_{rsbr} = L_{sr}\cos(\theta_r - \frac{2\pi}{3}) \tag{2.3.21}$$

$$L_{ysyr} = L_{sr}\cos\theta_r \tag{2.3.22}$$

$$L_{ysbr} = L_{sr}\cos(\theta_r + \frac{2\pi}{3}) \tag{2.3.23}$$

$$L_{ysrr} = L_{sr}\cos(\theta_r - \frac{2\pi}{3}) \tag{2.3.24}$$

$$L_{bsbr} = L_{sr}\cos\theta_r \tag{2.3.25}$$

$$L_{bsrr} = L_{sr}\cos(\theta_r + \frac{2\pi}{3}) \tag{2.3.26}$$

$$L_{bsyr} = L_{sr}\cos(\theta_r - \frac{2\pi}{3}) \tag{2.3.27}$$

The voltage equations for the induction machine are as follows

$$v_{rs} = R_s i_{rs} + \frac{d\lambda_{rs}}{dt} \tag{2.3.28}$$

$$v_{ys} = R_s i_{ys} + \frac{d\lambda_{ys}}{dt} \tag{2.3.29}$$

$$v_{bs} = R_s i_{bs} + \frac{d\lambda_{bs}}{dt} \tag{2.3.30}$$

$$v_{rr} = R_r i_{rr} + \frac{d\lambda_{rr}}{dt} \tag{2.3.31}$$

$$v_{yr} = R_r i_{yr} + \frac{d\lambda_{yr}}{dt} \tag{2.3.32}$$

$$v_{br} = R_r i_{br} + \frac{d\lambda_{br}}{dt} \tag{2.3.33}$$

where R_s is the resistance of each stator phase winding and R_r the resistance of each rotor phase winding. The flux linkages may be written as follows

$$\lambda_{rs} = L_{rsrs}i_{rs} + L_{rsys}i_{ys} + L_{rsbs}i_{bs} + L_{rsrr}i_{rr} + L_{rsyr}i_{yr} + L_{rsbr}i_{br} \tag{2.3.34}$$

$$\lambda_{ys} = L_{ysrs}i_{rs} + L_{ysys}i_{ys} + L_{ysbs}i_{bs} + L_{ysrr}i_{rr} + L_{sysyr}i_{yr} + L_{ysbr}i_{br} \tag{2.3.35}$$

$$\lambda_{bs} = L_{bsrs}i_{rs} + L_{bsys}i_{ys} + L_{bsbs}i_{bs} + L_{bsrr}i_{rr} + L_{bsyr}i_{yr} + L_{bsbr}i_{br} \tag{2.3.36}$$

$$\lambda_{rr} = L_{rrrs}i_{rs} + L_{rrys}i_{ys} + L_{rrbs}i_{bs} + L_{rrrr}i_{rr} + L_{rryr}i_{yr} + L_{rrbr}i_{br} \tag{2.3.37}$$

$$\lambda_{yr} = L_{yrrs}i_{rs} + L_{yrys}i_{ys} + L_{yrbs}i_{bs} + L_{yrrr}i_{rr} + L_{yryr}i_{yr} + L_{yrbr}i_{br} \tag{2.3.38}$$

$$\lambda_{br} = L_{brrs}i_{rs} + L_{brys}i_{ys} + L_{brbs}i_{bs} + L_{brrr}i_{rr} + L_{bsyr}i_{yr} + L_{brbr}i_{br} \tag{2.3.39}$$

These equations are solved as matrices and as such may be expressed as follows

$$\begin{bmatrix} \lambda_{rs} \\ \lambda_{ys} \\ \lambda_{bs} \\ \lambda_{rr} \\ \lambda_{yr} \\ \lambda_{br} \end{bmatrix} = \begin{bmatrix} L_{rsrs} & L_{rsys} & L_{rsbs} & L_{rsrr} & L_{rsyr} & L_{rsbr} \\ L_{ysrs} & L_{ysys} & L_{ysbs} & L_{ysrr} & L_{ysyr} & L_{ysbr} \\ L_{bsrs} & L_{bsys} & L_{bsbs} & L_{bsrr} & L_{bsyr} & L_{bsbr} \\ L_{rrrs} & L_{rrys} & L_{rrbs} & L_{rrrr} & L_{rryr} & L_{rrbr} \\ L_{yrrs} & L_{yrys} & L_{yrbs} & L_{yrrr} & L_{yryr} & L_{yrbr} \\ L_{brrs} & L_{brys} & L_{brbs} & L_{brrr} & L_{bryr} & L_{brbr} \end{bmatrix} \begin{bmatrix} i_{rs} \\ i_{ys} \\ i_{bs} \\ i_{rr} \\ i_{yr} \\ i_{br} \end{bmatrix} \tag{2.3.40}$$

which may be expressed in matrix shorthand notation as

$$\lambda = \mathbf{Li} \tag{2.3.41}$$

2.4 Voltage Equations of Machine Variables

The winding arrangement for a 2-pole, 3-phase, wye-connected, symmetrical induction machine is shown in Figure 2.1. The stator windings are identical, sinusoidally distributed windings, displaced 120 degrees, with N_s equivalent turns and resistance R_s. For the purposes at hand, the squirrel-cage rotor will be considered to consist of three identical sinusoidally distributed windings, displaced 120 degrees, with N_r equivalent turns and resistance R_r. The positive direction of the magnetic axis of each winding is anti-clockwise. The positive direction of the magnetic axes of the stator windings coincides with the direction of f_{rs}, f_{ys} and f_{bs} as specified by the equations of transformation and shown in Figure 2.2. The voltage equations of machine variables may be expressed

$$\mathbf{v}_{rybs} = \mathbf{R}_s\mathbf{i}_{rybs} + D\lambda_{rybs} \tag{2.4.1}$$

$$\mathbf{v}_{rybr} = \mathbf{R}_r\mathbf{i}_{rybr} + D\lambda_{rybr} \tag{2.4.2}$$

where

$$\mathbf{f}_{rybs}^T = [f_{rs} \quad f_{ys} \quad f_{bs}] \tag{2.4.3}$$

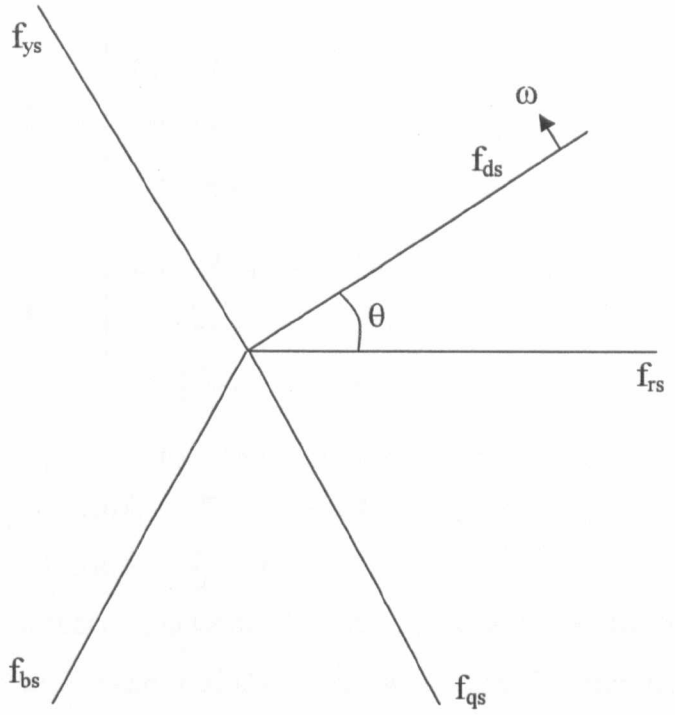

Figure 2.2: Transformation for Stator Circuits Portrayed by Trigonometric Relationships

$$\mathbf{f}_{rybr}^{T} = [f_{rr} \quad f_{yr} \quad f_{br}] \tag{2.4.4}$$

In the above equations the *ryb* subscripts denote the 3-phases, the latter s subscript denotes variables and parameters associated with the stator circuits, and the latter r subscript denotes variables and parameters associated with the rotor circuits. D is the differential operator d/dt. Both $\mathbf{R}_s$ and $\mathbf{R}_r$ are diagonal matrices each with equal nonzero elements. For a magnetically linear system, the flux linkages may be expressed

$$\begin{bmatrix} \lambda_{rybs} \\ \lambda_{rybr} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_s & \mathbf{L}_{sr} \\ \mathbf{L}_{sr}^{T} & \mathbf{L}_r \end{bmatrix} \begin{bmatrix} \mathbf{i}_{rybs} \\ \mathbf{i}_{rybr} \end{bmatrix} \tag{2.4.5}$$

The winding inductances in matrix format may be expressed

$$\mathbf{L}_s = \begin{bmatrix} L_{ls}+L_{ms} & -\frac{1}{2}L_{ms} & -\frac{1}{2}L_{ms} \\ -\frac{1}{2}L_{ms} & L_{ls}+L_{ms} & -\frac{1}{2}L_{ms} \\ -\frac{1}{2}L_{ms} & -\frac{1}{2}L_{ms} & L_{ls}+L_{ms} \end{bmatrix} \tag{2.4.6}$$

$$\mathbf{L}_r = \begin{bmatrix} L_{lr}+L_{mr} & -\frac{1}{2}L_{mr} & -\frac{1}{2}L_{mr} \\ -\frac{1}{2}L_{mr} & L_{ls}+L_{mr} & -\frac{1}{2}L_{mr} \\ -\frac{1}{2}L_{mr} & -\frac{1}{2}L_{mr} & L_{ls}+L_{mr} \end{bmatrix} \tag{2.4.7}$$

$$\mathbf{L}_{sr} = L_{sr} \begin{bmatrix} \cos\theta_r & \cos(\theta_r+\frac{2\pi}{3}) & \cos(\theta_r-\frac{2\pi}{3}) \\ \cos(\theta_r-\frac{2\pi}{3}) & \cos\theta_r & \cos(\theta_r+\frac{2\pi}{3}) \\ \cos(\theta_r+\frac{2\pi}{3}) & \cos(\theta_r-\frac{2\pi}{3}) & \cos\theta_r \end{bmatrix} \tag{2.4.8}$$

In the above inductance equations, L_{ls} and L_{ms} are, respectively, the leakage and magnetizing inductances of the stator windings; L_{lr} and L_{mr} are for the rotor windings. The inductance L_{sr} is the amplitude of the mutual inductances between stator and rotor windings.

We shall refer all the rotor variables to the stator windings by appropriate turns ratios as follows.

$$\mathbf{i}'_{rybr} = \frac{N_r}{N_s}\mathbf{i}_{rybr} \tag{2.4.9}$$

$$\mathbf{v}'_{rybr} = \frac{N_s}{N_r}\mathbf{v}_{rybr} \tag{2.4.10}$$

$$\lambda'_{rybr} = \frac{N_s}{N_r}\lambda_{rybr} \tag{2.4.11}$$

The magnetizing, L_{ms} and L_{mr}, and mutual inductances, L_{sr}, are associated with the same magnetic flux path. In particular

$$\mathbf{L}_{ms} = \frac{N_s}{N_r}\mathbf{L}_{sr} \tag{2.4.12}$$

We will define

$$\mathbf{L}'_{sr} = \frac{N_s}{N_r}\mathbf{L}_{sr} \tag{2.4.13}$$

$$= L_{ms}\begin{bmatrix} \cos\theta_r & \cos(\theta_r + \frac{2\pi}{3}) & \cos(\theta_r - \frac{2\pi}{3}) \\ \cos(\theta_r - \frac{2\pi}{3}) & \cos\theta_r & \cos(\theta_r + \frac{2\pi}{3}) \\ \cos(\theta_r + \frac{2\pi}{3}) & \cos(\theta_r - \frac{2\pi}{3}) & \cos\theta_r \end{bmatrix} \quad (2.4.14)$$

L_{ms} may be expressed as

$$L_{mr} = \left(\frac{N_r}{N_s}\right)^2 L_{ms} \quad (2.4.15)$$

and also

$$\mathbf{L}_r^{'} = \left(\frac{N_s}{N_r}\right)^2 \mathbf{L}_r \quad (2.4.16)$$

and equation 2.4.7 can now be expressed as

$$\mathbf{L}_r^{'} = \begin{bmatrix} L_{lr}^{'} + L_{ms} & -\frac{1}{2}L_{ms} & -\frac{1}{2}L_{ms} \\ -\frac{1}{2}L_{ms} & L_{lr}^{'} + L_{ms} & -\frac{1}{2}L_{ms} \\ -\frac{1}{2}L_{ms} & -\frac{1}{2}L_{ms} & L_{lr}^{'} + L_{ms} \end{bmatrix} \quad (2.4.17)$$

where

$$L_{lr}^{'} = \left(\frac{N_s}{N_r}\right)^2 L_{lr} \quad (2.4.18)$$

The flux linkages may now be expressed

$$\begin{bmatrix} \lambda_{rybs} \\ \lambda_{rybr}^{'} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_s & \mathbf{L}_{sr}^{'} \\ \mathbf{L}_{sr}^{'T} & \mathbf{L}_r^{'} \end{bmatrix} \begin{bmatrix} \mathbf{i}_{rybs} \\ \mathbf{i}_{rybr}^{'} \end{bmatrix} \quad (2.4.19)$$

The voltage equations expressed in terms of machine variables referred to the stator windings may now be written

$$\begin{bmatrix} \mathbf{v}_{rybs} \\ \mathbf{v}_{rybr}^{'} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_s + D\mathbf{L}_s & D\mathbf{L}_{sr}^{'} \\ D\mathbf{L}_{sr}^{'T} & \mathbf{r}_r^{'} + D\mathbf{L}_r^{'} \end{bmatrix} \begin{bmatrix} \mathbf{i}_{rybs} \\ \mathbf{i}_{rybr}^{'} \end{bmatrix} \quad (2.4.20)$$

where

$$\mathbf{r}_r^{'} = \left(\frac{N_s}{N_r}\right)^2 \mathbf{r}_r \quad (2.4.21)$$

The voltage equations are only approximate in that they do not take account of the power losses that inevitably accrue. These electric power losses are a result of the following. The input power to an induction machine is in the form

of three-phase electric voltage and currents. The first electric power losses encountered in the machine are stator copper losses in the stator windings. Then some amount of power is lost in the stator core as hysteresis, saturation and eddy currents. The power remaining at this point is transferred to the rotor of the machine across the air gap between the stator and rotor. This power is called air-gap power of the machine. After the power is transferred to the rotor, some of it is lost as rotor copper and core losses. However, we see the complexity of the voltage equations due to the time-varying mutual inductances between stator and rotor circuits since the circuits are in relative motion. The following sections will explain how a change of variables eliminates the time-varying inductances resulting in voltage equations which are still nonlinear but much more manageable.

2.5 The Background to Reference-Frame Transformation Analysis

The voltage equations which describe the performance of an induction machine was established in Section 2.3. It was found that some of the machine inductances are functions of the rotor speed, whereupon the coefficients of the differential equations (voltage equations) which describe the behaviour of the machines are time-varying except when the rotor is stalled. A change of variables shall be used to reduce the complexity of these differential equations. There are several changes of variables which are used and it was originally thought that each change of variable was different and therefore they were treated separately [19, 20, 21, 22]. It was later learned that all changes of variables used to transform real variables are contained in one [23, 24]. This general transformation refers machine variables to a frame of reference which rotates at an arbitrary angular velocity. All known real transformations are

obtained from this general transformation by assigning the speed of rotation of the reference frame.

In the late 1920s, R. H. Park [19] introduced a new approach to electric machine analysis. He formulated a change of variables which, in effect, replaced the variables (voltages, currents and flux linkages) associated with the stator windings of a synchronous machine with variables associated with fictitious windings rotating with the rotor. In other words, he transformed, or referred, the stator variables to a frame of reference fixed in the rotor. Park's transformation has the unique property of eliminating all time-varying inductances from the voltage equations of the synchronous machine which occur due to (1) electric circuits in relative motion and (2) electric circuits with varying magnetic reluctance.

In the late 1930s, H. C. Stanley [20] employed a change of variables in the analysis of induction machines. He showed that the time-varying inductances in the voltage equations of an induction machine due to electric circuits in relative motion could be eliminated by transforming the variables associated with the rotor windings (rotor variables) to variables associated with fictitious stationary windings. In this case the rotor variables are transformed to a frame of reference fixed in the stator.

G. Kron [21] introduced a change of variables which eliminated the time-varying inductances of a symmetrical induction machine by transforming both the stator variables and the rotor variables to a reference frame rotating in synchronism with the rotating magnetic field. The reference frame is commonly referred to as the synchronously rotating reference frame.

D. S. Brereton et al. [22] employed a change of variables which also eliminated the time-varying inductances of a symmetrical induction machine by transforming the stator variables to a reference frame fixed in the rotor. This is essentially Park's transformation applied to induction machines.

Park, Stanley, Kron and Brereton et al. developed changes of variables each

of which appeared to be uniquely suited for a particular application. Consequently, each transformation was derived and treated separately in literature until it was noted by P. C. Krause [23] that all known real transformations used in induction machine analysis are contained in one general transformation which eliminates all time-varying inductances by referring the stator and rotor variables to a frame of reference which may rotate at any angular velocity or remain stationary. All known real transformations may then be obtained by simply assigning the appropriate speed of rotation to this *arbitrary reference frame.*

2.6 Equations of Transformation for Stator Circuits

A change of variables which formulates a transformation of the 3-phase variables of stator circuit elements to the arbitrary reference frame may be expressed

$$\mathbf{f}_{dq0s} = \mathbf{K}_s \mathbf{f}_{rybs} \tag{2.6.1}$$

where

$$\mathbf{f}_{dq0s}^T = [f_{ds} \quad f_{qs} \quad f_{0s}] \tag{2.6.2}$$

$$\mathbf{f}_{rybs}^T = [f_{rs} \quad f_{ys} \quad f_{bs}] \tag{2.6.3}$$

$$\mathbf{K}_s = \frac{2}{3}\begin{bmatrix} \cos\theta & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) \\ \sin\theta & \sin(\theta - \frac{2\pi}{3}) & \sin(\theta + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \tag{2.6.4}$$

$$\theta = \int_0^t \omega(\xi)d\xi + \theta(0) \tag{2.6.5}$$

where ξ is a dummy variable of integration. The inverse transformation of $\mathbf{K}_s$ is

$$\mathbf{K}_s^{-1} = \begin{bmatrix} \cos\theta & \sin\theta & 1 \\ \cos(\theta - \frac{2\pi}{3}) & \sin(\theta - \frac{2\pi}{3}) & 1 \\ \cos(\theta + \frac{2\pi}{3}) & \sin(\theta + \frac{2\pi}{3}) & 1 \end{bmatrix} \tag{2.6.6}$$

In the above equations, f can represent either voltage, current or flux linkage. The superscript T denotes the transpose of a matrix. The subscript s indicates the variables, parameters and transformation associated with the stator circuits. The frame of reference may rotate at any constant or varying angular velocity or it may remain stationary. The connotation of arbitrary stems from the fact that the angular velocity of the transformation is unspecified and can be selected arbitrarily to expedite the solution of the system equations or to satisfy the system constraints.

The transformation to the arbitrary reference frame is a change of variables and needs no physical connotation, it is convenient to visualize the transformation equations as trigonometric relationships between variables as shown in Figure 2.2. In particular, the equations of transformation may be thought of as if the f_{ds} and f_{qs} variables are 'directed' along paths orthogonal to each other and rotating at an angular velocity of ω, whereupon f_{rs}, f_{ys} and f_{bs} may be considered as variables directed along stationary paths each displaced by 120 degrees. It is important to note that the 0s variables are not associated with the arbitrary reference frame. Instead, the zero variables are related arithmetically to the ryb variables, independent of θ. Portraying the transformation as shown in Figure 2.2 is particularly convenient when applying it to induction machines where the direction of f_{rs}, f_{ys} and f_{bs} may be thought of as the direction of the magnetic axes of the stator windings. It is found that the direction of f_{ds} and f_{qs} can be considered as the direction of the magnetic axes of the 'new' windings created by the change of variables.

The total instantaneous power may be expressed in the *ryb* variables as

$$P_{rybs} = v_{rs}i_{rs} + v_{ys}i_{ys} + v_{bs}i_{bs} \tag{2.6.7}$$

The total power expressed in the *dq*0 variables must equal the total power expressed in the *ryb* variables, hence substituting (2.6.1) into (2.6.7) yields

$$\begin{aligned} P_{dq0s} &= P_{ryb} \\ &= \frac{3}{2}(v_{ds}i_{ds} + v_{qs}i_{qs} + 2v_{0s}i_{0s}) \end{aligned} \tag{2.6.8}$$

The 3/2 factor comes about due to the choise of the constant used in the transformation. Although the waveforms of the *ds* and *qs* voltages, currents and flux linkages are dependent upon the angular velocity of the frame of reference, the waveform of the total power is independent of the frame of reference. In other words, the waveform of the total power is the same regardless of the reference frame in which it is evaluated.

2.7 Stator Circuit Variables Transformed to the Arbitrary Reference Frame

Let us treat resistive and inductive circuit elements separately.

2.7.1 Stator Resistive Elements

For a 3-phase resistive circuit

$$\mathbf{v_{rybs}} = \mathbf{R_s i_{rybs}} \tag{2.7.1}$$

From Section 2.6

$$\mathbf{v_{dq0s}} = \mathbf{K_s R_s K_s}^{-1}\mathbf{i_{dq0s}} \tag{2.7.2}$$

Since we are analysing a symmetrical induction machine, all the resistances are the same.

$$\mathbf{K_s R_s K_s}^{-1} = \mathbf{R_s} \tag{2.7.3}$$

2.7.2 Stator Inductive Elements

For a 3-phase inductive circuit

$$\mathbf{v_{rybs}} = D\lambda_{\mathbf{rybs}} \tag{2.7.4}$$

where D is the operator d/dt. Thus, in terms of the substitute variables, (2.7.4) becomes

$$\mathbf{v_{dq0s}} = \mathbf{K_s}D[\mathbf{K_s}^{-1}\lambda_{dq0s}] \tag{2.7.5}$$

which can be written

$$\mathbf{v_{dq0s}} = \mathbf{K_s}D[\mathbf{K_s}^{-1}]\lambda_{\mathbf{dq0s}} + \mathbf{K_s}\mathbf{K_s}^{-1}D\lambda_{\mathbf{dq0s}} \tag{2.7.6}$$

Since

$$\mathbf{K}_s = \frac{2}{3}\begin{bmatrix} \cos\theta & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) \\ \sin\theta & \sin(\theta - \frac{2\pi}{3}) & \sin(\theta + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \tag{2.7.7}$$

and

$$\mathbf{K}_s^{-1} = \begin{bmatrix} \cos\theta & \sin\theta & 1 \\ \cos(\theta - \frac{2\pi}{3}) & \sin(\theta - \frac{2\pi}{3}) & 1 \\ \cos(\theta + \frac{2\pi}{3}) & \sin(\theta + \frac{2\pi}{3}) & 1 \end{bmatrix} \tag{2.7.8}$$

It can be shown that

$$D[\mathbf{K_s}^{-1}] = \omega\begin{bmatrix} -\sin\theta & \cos\theta & 0 \\ -\sin(\theta - \frac{2\pi}{3}) & \cos(\theta - \frac{2\pi}{3}) & 0 \\ -\sin(\theta + \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) & 0 \end{bmatrix} \tag{2.7.9}$$

Therefore,

$$\mathbf{K_s}D[\mathbf{K_s}^{-1}] = \omega\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.7.10}$$

Equation (2.7.6) may now be expressed

$$\mathbf{v_{dq0s}} = \omega\lambda_{\mathbf{qds}} + D\lambda_{\mathbf{dq0s}} \tag{2.7.11}$$

where

$$\lambda_{\mathbf{qds}}{}^{T} = [\lambda_{qs} \quad -\lambda_{ds} \quad 0] \tag{2.7.12}$$

Equation (2.7.11) written in the expanded form becomes

$$v_{ds} = \omega\lambda_{qs} + D\lambda_{ds} \tag{2.7.13}$$

$$v_{qs} = -\omega\lambda_{ds} + D\lambda_{qs} \tag{2.7.14}$$

$$v_{0s} = D\lambda_{0s} \tag{2.7.15}$$

It should be noted that the first term on the right side of (2.7.13) or (2.7.14) is referred to as a 'speed voltage' with the speed being the angular velocity of the arbitrary reference frame.

For a magnetically linear system, the flux linkages may be expressed

$$\lambda_{\mathbf{rybs}} = \mathbf{L_s i_{rybs}} \tag{2.7.16}$$

Whereupon, the flux linkages in the arbitrary reference frame may be written

$$\lambda_{\mathbf{dq0s}} = \mathbf{K_s L_s K_s}^{-1}\mathbf{i_{dq0s}} \tag{2.7.17}$$

The stator inductance matrix for the symmetrical induction machine is of the form

$$\mathbf{L_s} = \begin{bmatrix} L_{ls} + L_{ms} & -\frac{1}{2}L_{ms} & -\frac{1}{2}L_{ms} \\ -\frac{1}{2}L_{ms} & L_{ls} + L_{ms} & -\frac{1}{2}L_{ms} \\ -\frac{1}{2}L_{ms} & -\frac{1}{2}L_{ms} & L_{ls} + L_{ms} \end{bmatrix} \tag{2.7.18}$$

where L_{ls} is a stator leakage inductance and L_{ms} is a stator magnetizing inductance. In the arbitrary reference frame it may be written as

$$\mathbf{K_s L_s K_s}^{-1} = \begin{bmatrix} L_{ls} + \frac{3}{2}L_{ms} & 0 & 0 \\ 0 & L_{ls} + \frac{3}{2}L_{ms} & 0 \\ 0 & 0 & L_{ls} \end{bmatrix} \tag{2.7.19}$$

2.8 Equations of Transformation for Rotor Circuits

In the analysis of induction machines it is desirable to transform the variables associated with the symmetrical rotor windings to the arbitrary reference frame. A change of variables which formulates a transformation of the 3-phase variables of the rotor circuits to the arbitrary reference frame is

$$\mathbf{f}'_{dq0r} = \mathbf{K}_r \mathbf{f}'_{rybr} \tag{2.8.1}$$

where

$$\mathbf{f}'^{T}_{dq0r} = [f'_{dr} \quad f'_{qr} \quad f'_{0r}] \tag{2.8.2}$$

$$\mathbf{f}'^{T}_{rybr} = [f'_{rr} \quad f'_{yr} \quad f'_{br}] \tag{2.8.3}$$

$$\mathbf{K}_r = \frac{2}{3}\begin{bmatrix} \cos\beta & \cos(\beta - \frac{2\pi}{3}) & \cos(\beta + \frac{2\pi}{3}) \\ \sin\beta & \sin(\beta - \frac{2\pi}{3}) & \sin(\beta + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \tag{2.8.4}$$

where $\beta = \theta - \theta_r$. The angular displacement θ and θ_r are defined as follows

$$\theta = \int_0^t \omega(\xi)d\xi + \theta(0) \tag{2.8.5}$$

$$\theta_r = \int_0^t \omega_r(\xi)d\xi + \theta_r(0) \tag{2.8.6}$$

where ξ is a dummy variable of integration. The inverse is

$$\mathbf{K}_r^{-1} = \begin{bmatrix} \cos\beta & \sin\beta & 1 \\ \cos(\beta - \frac{2\pi}{3}) & \sin(\beta - \frac{2\pi}{3}) & 1 \\ \cos(\beta + \frac{2\pi}{3}) & \sin(\beta + \frac{2\pi}{3}) & 1 \end{bmatrix} \tag{2.8.7}$$

The r subscript indicates the variables, parameters and transformations associated with the rotor circuit. Once again, one can visualize the transformation equations as trigonometric relationships between vector quantities as shown in Figure 2.3.

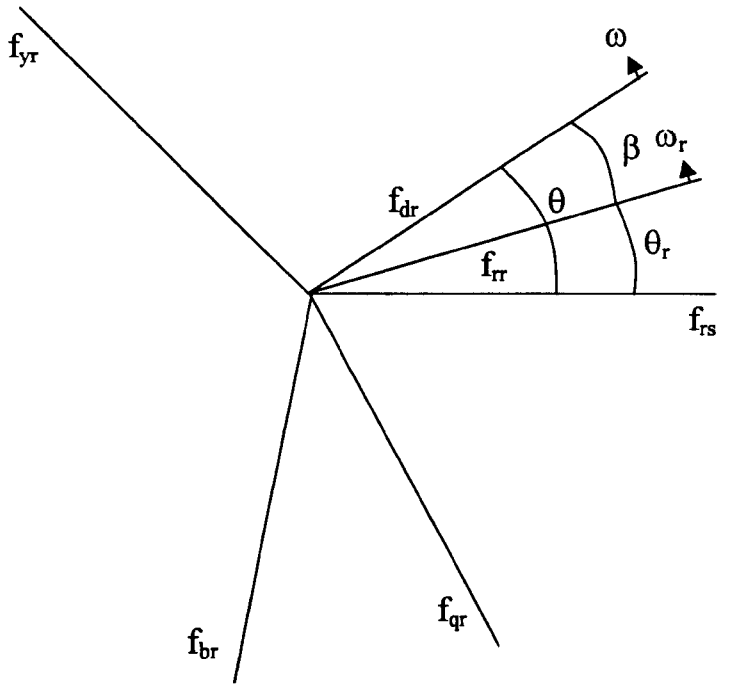

Figure 2.3: Transformation for Rotor Circuits Portrayed by Trigonometric Relationships

2.9 Rotor Circuit Variables Transformed to the Arbitrary Reference Frame

As before, we will treat resistive and inductive elements separately.

2.9.1 Rotor Resistive Elements

For a 3-phase resistive circuit

$$\mathbf{v}'_{rybr} = \mathbf{R}'_r \mathbf{i}'_{rybr} \tag{2.9.1}$$

and by transforming the parameters and variables we have

$$\mathbf{v}'_{dq0r} = \mathbf{K}_r \mathbf{R}'_r \mathbf{K}_r^{-1} \mathbf{i}'_{dq0r} \tag{2.9.2}$$

Since we are analysing a symmetrical induction machine, all the resistances are the same.

$$\mathbf{K}_r\mathbf{R}_r'\mathbf{K}_r^{-1} = \mathbf{R}_r' \tag{2.9.3}$$

2.9.2 Rotor Inductive Elements

For a 3-phase circuit

$$\mathbf{v}_{rybr}' = D\lambda_{rybr}' \tag{2.9.4}$$

Thus, in terms of the substitute variables, equation (2.9.4) becomes

$$\mathbf{v}_{dq0r}' = \mathbf{K}_r D[\mathbf{K}_r^{-1}\lambda_{dq0r}'] \tag{2.9.5}$$

which can be written

$$\mathbf{v}_{dq0r}' = \mathbf{K}_r D[\mathbf{K}_r^{-1}]\lambda_{dq0r}' + \mathbf{K}_r\mathbf{K}_r^{-1}D\lambda_{dq0r}' \tag{2.9.6}$$

Since

$$\mathbf{K}_r = \frac{2}{3}\begin{bmatrix} \cos\beta & \cos(\beta-\frac{2\pi}{3}) & \cos(\beta+\frac{2\pi}{3}) \\ \sin\beta & \sin(\beta-\frac{2\pi}{3}) & \sin(\beta+\frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \tag{2.9.7}$$

where $\beta = \theta - \theta_r$ and

$$\mathbf{K_r^{-1}} = \begin{bmatrix} \cos\beta & \sin\beta & 1 \\ \cos(\beta-\frac{2\pi}{3}) & \sin(\beta-\frac{2\pi}{3}) & 1 \\ \cos(\beta+\frac{2\pi}{3}) & \sin(\beta+\frac{2\pi}{3}) & 1 \end{bmatrix} \tag{2.9.8}$$

it can be shown that

$$D[\mathbf{K_r}^{-1}] = (\omega-\omega_r)\begin{bmatrix} -\sin\beta & \cos\beta & 0 \\ -\sin(\beta-\frac{2\pi}{3}) & \cos(\beta-\frac{2\pi}{3}) & 0 \\ -\sin(\beta+\frac{2\pi}{3}) & \cos(\beta+\frac{2\pi}{3}) & 0 \end{bmatrix} \tag{2.9.9}$$

Therefore,

$$\mathbf{K_r}D[\mathbf{K_r}^{-1}] = (\omega - \omega_r)\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.9.10}$$

Equation (2.9.6) may now be expressed

$$\mathbf{v}'_{dq0r} = (\omega - \omega_r)\lambda'_{qdr} + D\lambda'_{dq0r} \tag{2.9.11}$$

where

$$(\lambda'_{qdr})^T = [\lambda'_{qr} \quad -\lambda'_{dr} \quad 0] \tag{2.9.12}$$

Equation (2.9.11) written in the expanded form becomes

$$v'_{dr} = (\omega - \omega_r)\lambda'_{qr} + D\lambda'_{dr} \tag{2.9.13}$$

$$v'_{qr} = -(\omega - \omega_r)\lambda'_{dr} + D\lambda'_{qr} \tag{2.9.14}$$

$$v'_{0s} = D\lambda'_{0s} \tag{2.9.15}$$

For a magnetically linear system, the rotor flux linkages may be expressed

$$\lambda'_{rybr} = \mathbf{L}'_r\mathbf{i}'_{rybr} \tag{2.9.16}$$

so that the flux linkages in the arbitrary reference frame is written as

$$\lambda'_{dq0r} = \mathbf{K}_r\mathbf{L}'_r\mathbf{K}_r^{-1}\mathbf{i}'_{dq0r} \tag{2.9.17}$$

The rotor inductance matrix for the symmetrical induction machine is of the form

$$\mathbf{L}'_r = \begin{bmatrix} L'_{lr} + L_{ms} & -\frac{1}{2}L_{ms} & -\frac{1}{2}L_{ms} \\ -\frac{1}{2}L_{ms} & L'_{lr} + L_{ms} & -\frac{1}{2}L_{ms} \\ -\frac{1}{2}L_{ms} & -\frac{1}{2}L_{ms} & L'_{lr} + L_{ms} \end{bmatrix} \tag{2.9.18}$$

where L'_{lr} is a rotor leakage inductance and L_{ms} is a rotor magnetizing inductance. In the arbitrary reference frame it may be written as

$$\mathbf{K}_r\mathbf{L}'_r\mathbf{K}_r^{-1} = \begin{bmatrix} L'_{lr} + \frac{3}{2}L_{ms} & 0 & 0 \\ 0 & L'_{lr} + \frac{3}{2}L_{ms} & 0 \\ 0 & 0 & L'_{lr} \end{bmatrix} \tag{2.9.19}$$

2.10 Voltage Equations in Arbitrary Reference-Frame Variables

The voltage equations in the arbitrary reference frame may be expressed as

$$\mathbf{v}_{dq0s} = \mathbf{R}_s\mathbf{i}_{dq0s} + \omega\lambda_{qds} + D\lambda_{dq0s} \tag{2.10.1}$$

$$\mathbf{v}'_{dq0r} = \mathbf{R}'_r\mathbf{i}'_{dq0r} + (\omega - \omega_r)\lambda'_{qdr} + D\lambda'_{dq0r} \tag{2.10.2}$$

where

$$\lambda^T_{qds} = [\lambda_{qs} \quad -\lambda_{ds} \quad 0] \tag{2.10.3}$$

$$\lambda'^T_{qdr} = [\lambda'_{qr} \quad -\lambda'_{dr} \quad 0] \tag{2.10.4}$$

The flux linkage equations for a magnetically linear system can be expressed as

$$\begin{bmatrix} \lambda_{dq0s} \\ \lambda'_{dq0r} \end{bmatrix} = \begin{bmatrix} \mathbf{K}_s\mathbf{L}_s\mathbf{K}_s^{-1} & \mathbf{K}_s\mathbf{L}'_{sr}\mathbf{K}_r^{-1} \\ \mathbf{K}_r\mathbf{L}'^T_{sr}\mathbf{K}_s^{-1} & \mathbf{K}_r\mathbf{L}'_r\mathbf{K}_r^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{i}_{dq0s} \\ \mathbf{i}'_{dq0r} \end{bmatrix} \tag{2.10.5}$$

From Section (2.7) we can express $\mathbf{L}_s$ as follows

$$\mathbf{K}_s\mathbf{L}_s\mathbf{K}_s^{-1} = \begin{bmatrix} L_{ls} + L_m & 0 & 0 \\ 0 & L_{ls} + L_m & 0 \\ 0 & 0 & L_{ls} \end{bmatrix} \tag{2.10.6}$$

where

$$L_m = \frac{3}{2}L_{ms} \tag{2.10.7}$$

Likewise, for $\mathbf{L}'_r$, we have from Section (2.9)

$$\mathbf{K}_r\mathbf{L}'_r\mathbf{K}_r^{-1} = \begin{bmatrix} L'_{lr} + L_m & 0 & 0 \\ 0 & L'_{lr} + L_m & 0 \\ 0 & 0 & L'_{lr} \end{bmatrix} \tag{2.10.8}$$

It can be shown that

$$\mathbf{K}_s\mathbf{L}'_{sr}\mathbf{K}_r^{-1} = \mathbf{K}_r\mathbf{L}'^T_{sr}\mathbf{K}_s^{-1} = \begin{bmatrix} L_m & 0 & 0 \\ 0 & L_m & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.10.9}$$

The voltage equations are often written in expanded form as follows

$$v_{ds} = R_s i_{ds} + \omega\lambda_{qs} + D\lambda_{ds} \quad (2.10.10)$$

$$v_{qs} = R_s i_{qs} - \omega\lambda_{ds} + D\lambda_{qs} \quad (2.10.11)$$

$$v_{0s} = R_s i_{0s} + D\lambda_{0s} \quad (2.10.12)$$

$$v'_{dr} = R'_r i'_{dr} + (\omega - \omega_r)\lambda'_{qr} + D\lambda'_{dr} \quad (2.10.13)$$

$$v'_{qr} = R'_r i'_{qr} - (\omega - \omega_r)\lambda'_{dr} + D\lambda'_{qr} \quad (2.10.14)$$

$$v'_{0r} = R'_r i'_{0r} + D\lambda'_{0r} \quad (2.10.15)$$

Substituting equations (2.10.6), (2.10.8) and (2.10.9) into equations (2.10.5) yields the expressions for the flux linkages which in expanded form are

$$\lambda_{ds} = L_{ls} i_{ds} + L_m(i_{ds} + i'_{dr}) \quad (2.10.16)$$

$$\lambda_{qs} = L_{ls} i_{qs} + L_m(i_{qs} + i'_{qr}) \quad (2.10.17)$$

$$\lambda_{0s} = L_{ls} i_{0s} \quad (2.10.18)$$

$$\lambda_{dr} = L_{lr} i_{dr} + L_m(i_{ds} + i'_{dr}) \quad (2.10.19)$$

$$\lambda_{qr} = L_{lr} i_{qr} + L_m(i_{qs} + i'_{qr}) \quad (2.10.20)$$

$$\lambda_{0r} = L_{lr} i_{0r} \quad (2.10.21)$$

We shall select the currents as the independent variables whereby the voltage equations become

$$\mathbf{v} = \mathbf{Z}\mathbf{i} \quad (2.10.22)$$

where

$$\mathbf{v} = \begin{bmatrix} v_{ds} & v_{qs} & v_{0s} & v'_{dr} & v'_{qr} & v'_{0r} \end{bmatrix}^T \quad (2.10.23)$$

$$\mathbf{Z} = \begin{bmatrix} R_s + L_s D & -\omega L_s & 0 & L_m D & -\omega L_m & 0 \\ \omega L_s & R_s + L_s D & 0 & \omega L_m & L_m D & 0 \\ 0 & 0 & R_s + L_{ls} D & 0 & 0 & 0 \\ L_m D & -(\omega - \omega_r) L_m & 0 & R_r' + L_r' D & -(\omega - \omega_r) L_r' & 0 \\ (\omega - \omega_r) L_m & L_m D & 0 & (\omega - \omega_r) L_r' & R_r' + L_r' D & 0 \\ 0 & 0 & 0 & 0 & 0 & R_r' + L_{lr}' D \end{bmatrix} \tag{2.10.24}$$

where

$$L_s = L_{ls} + L_{ms} \tag{2.10.25}$$

$$L_r' = L_{lr}' + L_{ms} \tag{2.10.26}$$

and

$$\mathbf{i} = \begin{bmatrix} i_{ds} & i_{qs} & i_{0s} & i_{dr}' & i_{qr}' & i_{0r}' \end{bmatrix} \tag{2.10.27}$$

The above voltage equation contains the zero-sequence quantities. We will assume that the rotor does not have an external (slip-ring) connection to the star-point. We also assume that zero-sequence currents cannot flow in the stator. The above voltage equations therefore reduce to

$$\begin{bmatrix} v_{ds} \\ v_{qs} \\ v_{dr}' \\ v_{qr}' \end{bmatrix} = \begin{bmatrix} R_s + L_s D & -\omega L_s & L_m D & -\omega L_m \\ \omega L_s & R_s + L_s D & \omega L_m & L_m D \\ L_m D & -(\omega - \omega_r) L_m & R_r' + L_r' D & -(\omega - \omega_r) L_r' \\ (\omega - \omega_r) L_m & L_m D & (\omega - \omega_r) L_r' & R_r' + L_r' D \end{bmatrix} \begin{bmatrix} i_{ds} \\ i_{qs} \\ i_{dr}' \\ i_{qr}' \end{bmatrix} \tag{2.10.28}$$

2.11 Conclusion

The limitations of the differential equations (voltage equations) due to some of the machine inductances being functions of rotor speed, thus causing time-varying coefficients, has been abated by the application of the reference-frame

transformations. The following chapter will further develop these voltage equations in order to make them more amenable to simulation and analysis for this research.

Chapter 3

The Dynamic Mathematical Model of a Symmetrical Induction Machine for Simulation and Analysis

3.1 Formation of Currents as State Variables

Has mentioned in Chapter 2, the derivation of the voltage equations of an induction machine in the arbitrary frame of reference are of the form

$$\begin{bmatrix} v_{ds} \\ v_{qs} \\ v'_{dr} \\ v'_{qr} \end{bmatrix} = \begin{bmatrix} R_s + L_s D & -\omega L_s & L_m D & -\omega L_m \\ \omega L_s & R_s + L_s D & \omega L_m & L_m D \\ L_m D & -(\omega - \omega_r) L_m & R'_r + L'_r D & -(\omega - \omega_r) L'_r \\ (\omega - \omega_r) L_m & L_m D & (\omega - \omega_r) L'_r & R'_r + L'_r D \end{bmatrix} \begin{bmatrix} i_{ds} \\ i_{qs} \\ i'_{dr} \\ i'_{qr} \end{bmatrix} \tag{3.1.1}$$

We shall use the stationary reference frame first employed by H. C. Stanley [20]. This is achieved by placing the arbitrary electrical angular speed, ω, to zero in equation (3.1.1). Since the stator and rotor variables have been transformed

to an arbitrary frame of reference, we will place the angular velocity of the arbitrary reference frame to zero, so that the d-axis coincides with the stator phase axis as, hence, the matrix equation (3.1.1) becomes

$$\begin{bmatrix} v_{ds} \\ v_{qs} \\ v'_{dr} \\ v'_{qr} \end{bmatrix} = \begin{bmatrix} R_s + L_s D & 0 & L_m D & 0 \\ 0 & R_s + L_s D & 0 & L_m D \\ L_m D & L_m\omega_r & R'_r + L'_r D & L'_r\omega_r \\ -L_m\omega_r & L_m D & -L'_r\omega_r & R'_r + L'_r D \end{bmatrix} \begin{bmatrix} i_{ds} \\ i_{qs} \\ i'_{dr} \\ i'_{qr} \end{bmatrix} \tag{3.1.2}$$

Thus, equation (3.1.2) can now be manipulated so that the currents become the state variables for the numerical computation and analysis. Also, the rotor voltages are zero which leads to the following

$$X = \begin{bmatrix} i_{ds} & i_{qs} & i'_{dr} & i'_{qr} \end{bmatrix}^T \tag{3.1.3}$$

$$V = \begin{bmatrix} v_{ds} & v_{qs} \end{bmatrix}^T \tag{3.1.4}$$

$$A = \frac{1}{\sigma L_s L'_r} \begin{bmatrix} -R_s L'_r & L_m^2\omega_r & L_m R'_r & L_m L'_r\omega_r \\ -L_m^2\omega_r & -R_s L'_r & -L_m L'_r\omega_r & L_m R'_r \\ L_m R_s & -L_m L_s\omega_r & -L_s R'_r & -L_s L'_r\omega_r \\ L_m L_s\omega_r & L_m R_s & L_s L'_r\omega_r & -L_s R'_r \end{bmatrix} \tag{3.1.5}$$

$$B = \frac{1}{\sigma L_s L'_r} \begin{bmatrix} L'_r & 0 \\ 0 & L'_r \\ -L_m & 0 \\ 0 & -L_m \end{bmatrix} \tag{3.1.6}$$

Therefore

$$\dot{X} = AX + BV \tag{3.1.7}$$

and

$$\sigma = 1 - \frac{L_m^2}{L_s L'_r} \tag{3.1.8}$$

3.2 The Voltage Transformations

Remembering that the reference frame is attached to the stator with the direct axis, d, chosen to coincide with the stator as axis so that the arbitrary angle, θ, now becomes zero leading to the following

$$\begin{bmatrix} v_{ds} \\ v_{qs} \\ v_{0s} \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos(0) & \cos(-\frac{2\pi}{3}) & \cos(-\frac{4\pi}{3}) \\ \sin(0) & \sin(-\frac{2\pi}{3}) & \sin(-\frac{4\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} v_{rs} \\ v_{ys} \\ v_{bs} \end{bmatrix} \tag{3.2.1}$$

and

$$\begin{bmatrix} v_{rs} \\ v_{ys} \\ v_{bs} \end{bmatrix} = \begin{bmatrix} \cos(0) & \sin(0) & 1 \\ \cos(-\frac{2\pi}{3}) & \sin(-\frac{2\pi}{3}) & 1 \\ \cos(-\frac{4\pi}{3}) & \sin(-\frac{4\pi}{3}) & 1 \end{bmatrix} \begin{bmatrix} v_{ds} \\ v_{qs} \\ v_{0s} \end{bmatrix} \tag{3.2.2}$$

Now, let us suppose that the 3-phase peak voltages are symmetric and described as follows

$$v_{rs} = \sqrt{2}V_1 \cos(\omega_1 t) \tag{3.2.3}$$

$$v_{ys} = \sqrt{2}V_1 \cos(\omega_1 t - \frac{2\pi}{3}) \tag{3.2.4}$$

$$v_{bs} = \sqrt{2}V_1 \cos(\omega_1 t - \frac{4\pi}{3}) \tag{3.2.5}$$

This leads to the following reference frame voltages

$$\begin{bmatrix} v_{ds} \\ v_{qs} \\ v_{0s} \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos(0) & \cos(-\frac{2\pi}{3}) & \cos(-\frac{4\pi}{3}) \\ \sin(0) & \sin(-\frac{2\pi}{3}) & \sin(-\frac{4\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} v_{rs} \\ v_{ys} \\ v_{bs} \end{bmatrix} \tag{3.2.6}$$

$$= \frac{2}{3}\sqrt{2}V_1 \begin{bmatrix} \frac{3}{2}\cos(\omega_1 t) \\ \frac{3}{2}\sin(-\omega_1 t) \\ 0 \end{bmatrix} \tag{3.2.7}$$

When written in the expanded form this becomes

$$v_{ds} = \sqrt{2}V_1 \cos(\omega_1 t) \tag{3.2.8}$$

$$v_{qs} = \sqrt{2}V_1 \cos(\omega_1 t + \frac{\pi}{2}) = -\sqrt{2}V_1 \sin(\omega_1 t) \tag{3.2.9}$$

$$v_{0s} = 0 \tag{3.2.10}$$

3.3 The Current Transformations

The reference frame transformations, and vice versa for the stator and rotor currents are as follows

$$\begin{bmatrix} i_{ds} \\ i_{qs} \\ i_{0s} \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos(0) & \cos(-\frac{2\pi}{3}) & \cos(\frac{2\pi}{3}) \\ \sin(0) & \sin(-\frac{2\pi}{3}) & \sin(\frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} i_{rs} \\ i_{ys} \\ i_{bs} \end{bmatrix} \tag{3.3.1}$$

$$\begin{bmatrix} i_{rs} \\ i_{ys} \\ i_{bs} \end{bmatrix} = \begin{bmatrix} \cos(0) & \sin(0) & 1 \\ \cos(-\frac{2\pi}{3}) & \sin(-\frac{2\pi}{3}) & 1 \\ \cos(\frac{2\pi}{3}) & \sin(\frac{2\pi}{3}) & 1 \end{bmatrix} \begin{bmatrix} i_{ds} \\ i_{qs} \\ i_{0s} \end{bmatrix} \tag{3.3.2}$$

$$\begin{bmatrix} i'_{dr} \\ i'_{qr} \\ i'_{0r} \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos\beta & \cos(\beta - \frac{2\pi}{3}) & \cos(\beta + \frac{2\pi}{3}) \\ \sin\beta & \sin(\beta - \frac{2\pi}{3}) & \sin(\beta + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} i'_{rr} \\ i'_{yr} \\ i'_{br} \end{bmatrix} \tag{3.3.3}$$

$$\begin{bmatrix} i'_{rr} \\ i'_{yr} \\ i'_{br} \end{bmatrix} = \begin{bmatrix} \cos\beta & \sin\beta & 1 \\ \cos(\beta - \frac{2\pi}{3}) & \sin(\beta - \frac{2\pi}{3}) & 1 \\ \cos(\beta + \frac{2\pi}{3}) & \sin(\beta + \frac{2\pi}{3}) & 1 \end{bmatrix} \begin{bmatrix} i'_{dr} \\ i'_{qr} \\ i'_{0r} \end{bmatrix} \tag{3.3.4}$$

where

$$\beta = \theta - \theta_r \tag{3.3.5}$$

and, as mentioned previously, θ_r is the rotor angular displacement, $\theta_r = \omega_r t$.

3.4 The Instantaneous Electrical Torque Equation

The instantaneous electrical torque generated by the stator and rotor MMFs may be expressed as follows:

$$T_e = L_m Q(i_{ds} i'_{qr} - i_{qs} i'_{dr}) \tag{3.4.1}$$

where T_e is the instantaneous electrical torque (Nm) and Q is the number of pole pairs.

3.5 The Equation of Motion

Application of Newton's second law yields the following equation of motion for the 3-phase induction machine under investigation:

$$\frac{J}{Q}\frac{d\omega_r}{dt} = -T_e + T_r - \frac{B\omega_r}{Q} \tag{3.5.1}$$

where T_r is the load torque (Nm), B denotes the damping coefficient and J represents the inertia moment.

The speed of the rotor, ω_r, is negative for induction machines operating in the motoring mode.

3.6 The Induction Machine's Differential Equations Suitable for Simulation

By meticulous rearrangement of equations (3.1.2), (3.4.1) and (3.5.1), the induction machine's state equations are as follows

$$Di_{ds} = \sigma(-R_s L_r^{'} i_{ds} + L_m^2 i_{qs}\omega_r + L_m R_r^{'} i_{dr}^{'} + L_m L_r^{'} i_{qr}^{'}\omega_r) + \sigma L_r^{'} v_{ds} \tag{3.6.1}$$

$$Di_{qs} = \sigma(-L_m^2 i_{ds}\omega_r - R_s L_r^{'} i_{qs} - L_m L_r^{'} i_{dr}^{'}\omega_r + L_m R_r^{'} i_{qr}^{'}) + \sigma L_r^{'} v_{qs} \tag{3.6.2}$$

$$Di_{dr}^{'} = \sigma(R_s L_m i_{ds} - L_s L_m i_{qs}\omega_r - L_s R_r^{'} i_{dr}^{'} + L_s L_r^{'} i_{qr}^{'}\omega_r) - \sigma L_m v_{ds} \tag{3.6.3}$$

$$Di_{qr}^{'} = \sigma(L_s L_m i_{ds}\omega_r + R_s L_m i_{qs} + L_s L_r^{'} i_{dr}^{'}\omega_r - L_s R_r^{'} i_{qr}^{'}) - \sigma L_m v_{qs} \tag{3.6.4}$$

$$D\omega_r = \frac{Q}{J}[QL_m(-i_{ds} i_{qr}^{'} + i_{qs} i_{dr}^{'}) + T_r - \frac{B\omega_r}{Q}] \tag{3.6.5}$$

where

$$\sigma = \frac{1}{L_s L_r - L_m^2} \tag{3.6.6}$$

The state equations shown above are of the nonlinear differential equation form for which we shall use the Runge-Kutta method to solve them for the induction machine's currents and rotor speed in time steps of 1 millisecond.

Chapter 4

The Simulated Evolution Techniques Applied to the Optimization Problem of Parameters Identification

4.1 Introduction

In this chapter the fundamental mechanisms of optimization are discussed. The limitations of traditional optimization methods are described as well as how these limitations do not apply to Simulated Evolution (SE). Two paradigms of SE are applied in this research, Genetic Algorithms (GAs) and Evolutionary Programming (EP), and are described in terms of how they differ from traditional optimization techniques, their historical development, an algorithms description, their respective convergence properties and also their methodoligies.

4.2 The Processes Involved in Optimization

Let us establish what we mean when we say we want to optimize a function or a process. The conventional view is presented in [62] as follows:

> Man's longing for perfection finds expression in the theory of optmization. It studies how to describe and attain what is Best, once one knows how to measure and alter what is Good or Bad ... *Optimization theory* encompasses the quantitative study of optima and methods for finding them.

Thus optimization seeks to improve performance towards some optimal point or points. Or a more mathematically description is to refer optimization to function minimization and the equivalent process of function maximization. The optimization strategy is discussed in [64, 65] and may be summarized as follows:

1. guess a solution;
2. analyse the guess solution;
3. a performance comparison is made between the guess solution and the actual requirements:
   - if satisfactory stop;
   - if not satisfactory, change the values of one or more system parameters;
4. repeat 2 or 3 until results are adequate.

The 'performance comparison' phase of the process may be thought of as the evaluation of a discrepancy or error measure, and the task is to reduce this error to its lowest possible value. The lowest value is zero which corresponds to zero error but many problems do not have an exact solution, and such cases

will have a nonzero final error. The problem can be formulated in essence as a function minimization problem.

For example, suppose a particular system response is specified as A_{st_k} over a simulation period of T_N, and likewise, the actual or real system as a response A_{t_k} over the same period. An overall error may then be assembled as follows. For example, suppose we have identified T_N time instances of interest, then an error measure namely an objective function Φ which may be given by the sum-of-squares function

$$\Phi = \sum_{t_k=0}^{T_N} (A_{st_k} - A_{t_k})^2 \tag{4.2.1}$$

which has the required property ($\Phi = 0$ if and only if $A_{st_k} = A_{t_k}$ for all t_k).

The optimization problem may be summarized as follows. Given a suitable objective function, Φ, of n system parameters, $x_1, x_2, \ldots, x_n$, of the real system response, A_{t_k}, locate the set of parameter values, $x_1^*, x_2^*, \ldots, x_n^*$, of the simulated mathematical model system response, A_{st_k}, over the same time instances, which minimizes the objective function, $\Phi(x_1, x_2, \ldots, x_n)$. Therefore, $x_1^*, x_2^*, \ldots, x_n^*$ are the identified parameters, since the objective function, Φ, has located its optimum value. Hence, the most important goal of optimization is improvement, and to achieve a good level of performance quickly [63].

4.2.1 The Global Optimization Problem

The global opimization problem can be defined as follows:

$$max_x f(x), \qquad x \in S \tag{4.2.2}$$

where S is a bounded set on R^n and $f : S \rightarrow R$ is an $n-$dimensional real-valued function. The problem is to find a point $x_{max} \in S$ such that $f(x_{max})$ is globally maximal on S. Here f does not need to be continuous but it must be bounded and have a finite number of maximum points over S.

4.3 Traditional Optimization Methods and their Limitations

Optimization problems may firstly be classified as either constrained or unconstrained. A constraint may be applied to the range of values within which the system parameters may fall within or to some function of the parameters , a full description of these methods may be found in [66, 67, 68, 69, 70, 71]. Unconstrained optimization places no restriction on the parameter range [76, 77].

Traditional optimization and search methods are divided into calculus-based, enumerative and random. Calculus-based methods are further subdivided into two main classes: indirect and direct. Indirect methods seek local extrema by solving the usually nonlinear set of differential equations resulting from setting the gradient of the objective function equal to zero [72, 73, 74, 75]. By contrast, direct methods seek local optima by moving in a direction related to a local gradient [78, 79, 80, 81, 82, 83]. For example, the notation of hill-climbing to find the local maximum is, in each step, climbing the function in the 'steepest gradient'. Both of these calculus-based methods, however, have certain limitations.

First, both indirect and direct methods are local in scope; the optima they seek are the best in a neighbourhood of the current point. Thus, if a low peak is found, further improvement must be sought through a random restart or other means in order to find the global higher peak. Second, calculus-based methods depend upon the existance of derivatives. However, many practical search spaces of parameters have little respect for the notion of a derivative and the smoothness this implies. The real world of search is fraught with discontinuities, vast multimodal and noisy search space. None of them satisfy the requirements of calculus-based methods. Therefore, methods depending

upon the restrictive requirements of continuity and derivative existence are unsuitable for practical requirements. For these reasons and because of their inherently local scope of search, they are insufficiently robust.

Enumerative schemes have been considered in many shapes and size [84]. The idea is fairly straightforward; within a finite search space, or a discretized infinite search space, the search algorithm starts looking at objective function values at every point in the space, one at a time. Although the simplicity of this type of algorithm is attractive, it lacks efficiency because many practical spaces are too large to search. Random search algorithms have achieved increasing popularity as researchers have recognized the shortcomings of calculus-based and enumerative schemes. Yet, the evidence suggests that random walks and random schemes, in the long run, can be expected to do no better than enumerative schemes. Huang and Wu et al compared the performance of a simple random search (SRS) with genetic algorithms (GAs), to be discussed shortly, and found that the GA performed with better accuracies and fitnesses in the parameter identification problem of an induction motor [85].

4.4 Natural Evolution

Because of the afore mentioned difficulties encountered by traditional optimization methods, a new approach was introduced some two to three decades ago, namely Simulated Evolution (SE), and has been researched and developed from thereon. Before discussing SE in greater detail, it will be found quite enlightening to firstly discuss the fundamental principles upon which SE algorithms rest upon, namely the process of natural evolution based on the collective arguments known as the neo-Darwinian paradigm.

4.4.1 The Neo-Darwinian Paradigm

A revolution in biological thought, and indeed philosophy, was begun when Charles Darwin and Alfred Russel Wallace each presented their evidence for the theory of evolution before the Linnean Society of London on July 1, 1858. Classic Darwinian evolutionary theory, combined with the selectionism of Weismann and the genetics of Mendel, has now become a rather universally accepted set of arguments known as the neo-Darwinian paradigm [25, 26, 27, 28, 29, 30, 31, 32, 33].

Neo-Darwinism asserts that the history of the vast majority of life is fully accounted for by only a very few statistical processes acting on and within populations and species [34]. These processes are reproduction, mutation, competition and selection. Reproduction is an obvious property of all life. But similarly as obvious, mutation is guaranteed in any system that continuously reproduces itself in a positively entropic universe. Competition and selection become the inescapable consequences of any expanding population constrained to a finite arena. Evolution is then the result of these fundamental interacting stochastic processes as they act on populations, generation after generation [35, 36].

4.4.2 The Genotype and the Phenotype: The Optimization of Behaviour

Living organisms can be viewed as a duality of their genotype (the underlying genetic code) and their phenotype (the manner of response contained in the behaviour, physiology and morphology of the organism). This may be thought of in the manner put forward by [37] by specifying two state spaces: a populational genotypic (informational) space **G** and a populational phenotypic (behavioural) space **P**. The function mappings were stated in [38] as follows:

$$f_1 : \mathbf{I} \times \mathbf{G} \rightarrow \mathbf{P}$$

$$f_2 : \mathbf{P} \rightarrow \mathbf{P}$$

$$f_3 : \mathbf{P} \rightarrow \mathbf{G}$$

$$f_4 : \mathbf{G} \rightarrow \mathbf{G}$$

The function f_1, epigenesis, maps the element $g_1 \in \mathbf{G}$ into the phenotypic space $\mathbf{P}$ as a particular collection of phenotypes p_1 whose development is modified by its environment, an indexed set of symbols $(i_1, \ldots, i_k) \in \mathbf{I}$, where $\mathbf{I}$ is the set of all such environmental sequences. The function f_2, selection, maps phenotypes p_1 into p_2. As natural selection only operates on the phenotypic expressions of the genotype [39, 40, 41], the underlying coding g_1 is not involved in function f_2. The function f_3, genotypic survival, describes the effects of selection and migration processes on $\mathbf{G}$. Function f_4, mutation, maps the representative codings $g_2 \in \mathbf{G}$ to a point $g_1^{'} \in \mathbf{G}$. This function represents the 'rules' of mutation and recombination, encompasses all genetic changes. With the creation of the new population of genetypes $g_1^{'}$, one generation is complete. Evolutionary adaptation occurs over successive iterations of these mapping functions.

The neo-Darwinian argument asserts that natural selection is the predominant mediating evolutionary force that prevails in shaping the phenotypic characters in the vast majority of situations encountered in nature [42, 34, 43]. It is strictly an a posteriori process that rewards current success [42] primarily through the statistical culling of inappropriate individuals. Selection acts in the face of phenotypic variation. Parents and their offspring typically demonstrate at least a general resemblance in their phenotypic traits [41]. Their behaviours may in fact be virtually identical, that is, reproduction may be very nearly replication [34] But phenotypic variation is always observed within populations and species and is largely the result of mutation and recombination (if

applicable), environmental constraints placed on an individual's development, and replicative errors of the genome. The interaction between the species and its environment (which includes other organisms) determines the relative success (differential reproduction) or failure (genetic death) of the species.

Selection is often viewed as leading to the maintenance or increase of populations' 'fitness', where fitness is defined as the ability to survive and reproduce in a specific environment [41]. It has been asserted that although fitness cannot be directly measured, 'its distribution in a population can be roughly estimated in any given environmental context on the basis of ecology and functional morphology of the organisms. Hence it is an empirically testable biological proposition' [34]. W. Atmar in [44] indicated that a singular measure of evolutionary fitness is the appropriateness of a species' behaviour in terms of its ability to anticipate its environment. The quantitative ability to perform suitable prediction and elicit appropriate response yields a measure of fitness.

Pleiotropy is the effect that a single gene may simultaneously affect several phenotypic traits. Polygeny is the effect that a single phenotypic characteristic of an individual may be determined by the simultaneous interaction of many genes. Assigning fitness to individual genes implicitly ignores these interactions, or describes them only on the average, and often assumes a one-gene/one-trait model of the relationship between the genotype and the phenotype. [45] suggested 'evolution can no longer be looked at solely as changes in gene frequencies within populations'.

The one-gene/one-trait model of evolutionary genetics is an over simplification. Naturally evolved systems are extensively pleiotropic and highly polygenic. Selection acts on collections of interactive phenotypic traits, not on singular traits in isolation. The appropriateness of an organism's holistic functional behaviour in light of the physics of its environment is the sole quality that is optimized through selection.

The fundamental characteristics of diverse environments are often pro-

foundly similar, and convergent evolution is the general outcome. The compound image-forming eye has been invented at least three times during the course of evolution: in mollusk, arthropods, and vertebrates. The most common ancestor of these three taxa could have occurred no later than 450 million years ago and more probably 550 to 600 million years ago. But in all three cases, a profoundly similar informational neurophysiology has been invented, although slightly differently realized. In vertebrates, the neural connections lay on top of the retinal surface. In mollusks, they occur behind the eye as they do in arthropods. But in all three, a network of collateralized synaptic connections occur in a series of amacrine, horizontal, and bipolar cells that generate edge enhancement, motion detection, and modulatable light sensitivities in strikingly similar manners [46]. The convergent functional behaviour of these independently evolved constructions leads to the conclusion that the invention of such a functional structure is exceedingly probable. Although there are obvious differences in the implementation of the eye arrangements, their functional behaviour is profoundly similar. The physical and historical constraints woven into the mechanisms and process of the construction of a cell appear to predestine the evolution of a particular image-forming eye [46].

Similarly, the invention of flight has occurred repeatedly. Gliding has been invented independently in fish, amphibians, reptiles and mammals. Self-sustained flight has evolved independently in pterosaurs, birds, mammals and insects [45]. Other forms of flight have been invented in insects, arachnids and plants. Young spiders, in which the young are transported over great distances by means of a parachute produced from silken thread, have been documented to drift over 400 miles [47]. The underlying genetic structures for these independently evolved phenotypic characteristics are diverse, yet the functionalities are notably similar.

Even ethologies are often profoundly convergent. During displays of aggression, animals as varied as Australian crayfish, baboon spiders, toads, wolves

and black-headed gulls all assume postures that make them appear larger to their adversaries. These behaviours include the raising of body hair, standing on hind legs in an upright manner, and spreading of the forelegs or claws [48].

Therefore, pleiotropy and polygeny preclude the possibility of singular genes for these complex effects. While it is possible to alter specific traits through single gene mutations, this does not indicate a gene 'for' that particular trait. There are no individual genes for specific behaviours, such as 'making oneself appear big', in natural evolved systems.

4.4.3 Sexual Reproduction

The term sex derives from the Latin secare, to cut or divide something that was originally whole. The exchange of genetic information is fundamental to the cutting and dividing of DNA that takes place during sexual reproduction. This exchange takes many forms: bacterial conjugation, transduction via viral transport, and the more familiar recombination that occurs in plants and animals. But sex requires finding a mate, either by happenstance, active search or attraction. Further, the behaviours that are associated with attracting mates often make individuals more vulnerable to predators. Yet the benefits of sexuality must outweigh these costs, for the great majority of known species have a sexual phase in their life cycle.

Diploid organisms have two copies of each chromosome in every somatic cell, with one copy being contributed from each parent. The total number of chromosomes in the cells of most eukaryotes (organisms with fully nucleated cells) varies between 10 and 50 but can exceed 1000 in some polyploid plants. During recombination, haploid (nonredundant) gametes are fused to form new diploid cells.

A reduction division is then required for this infusion to generate a stable number of chromosomes in the sexually reproducing lineage. This division is

termed meiosis. In this process, each chromosome from both parents is duplicated. A crossing-over between homologous chromosomes may occur in which segments of the chromosome are exchanged. The rate of crossing-over varies widely from species to species and between sexes. Male fruit flies exhibit no crossing-over at all, while in mammals the rate of crossing-over is about 30 percent higher in females [49]. During meiosis, a reduction division occurs in which one of the two homologous chromosomes is sorted into each cell. If crossing-over does take place, it occurs predominantly during this first meiotic division. The chromatids then undergo a second reduction division such that each gamete receives a potentially hybrid combination of each pair of homologous chromosomes from each parent.

In contrast, sexual reproduction operates in a different manner. Sexual reproduction offers a significant ability for a species to generate genetic diversity and, as a consequence, phenotypic diversity. The overt functional advantage of sexual recombination to an evolving species is the significantly increased rate of exploration of the genotypic/phenotypic state space, as compared with clonal parthenogenesis, especially in changing environments. Recombination will ultimately tend to expose a wide variety of individual genotypes to various environmental conditions. The number of such genetic combinations is, for all practical purposes, unlimited. It is presumed that each offspring receives each chromosome from each parent in a random manner. In humans, 2^{23} distinct zygotes can be produced from only two parents by simply considering the possible combinations.

Three potential advantages of recombination have been identified [50]: (1) greater efficiency for adjusting to a changing environment, (2) bringing together independently created beneficial mutations, and (3) removing deleterious mutations.

4.4.4 Sexual Selection

Sexual reproduction allows for sexual selection. This term refers to the process by which males and females of a species determine or 'select' their mates. These processes often involve prolonged demonstrations of vigour between members of the species, typically the males. Male-male competition has been viewed as an important source of sexual selection since Darwin (1859). He concluded that the result of these encounters was not generally death to the vanquished, but 'few or no offspring' [49]. A genetic death is equally as real as a physiological death in purging a defect from the germline.

Males in many species have evolved specific attributes that appear designed solely for male-male competition. Antlers, for example, are used primarily in intraspecies competition but otherwise appear poorly constructed for predator defense. The enlarged claw of male crabs (especially fiddler crabs, in which the pincer may constitute one-third of the crab's total body weight) is similarly observed only in mature males and it is not required for survival. Rather, the claw is used in wrestling matches between males and for attracting females.

Aggressive behaviour is typical of virtually all males and can be triggered by specific stimuli. In experiments with stickleback fish, Tinbergen [51] indicated that males are more aggressive toward decoys with a red colouration but no visual resemblance to a fish than towards anatomically correct, colourless decoys. Similar results have been descibed by Lack [52] in which male robins threatened a pile of red feathers more readily than a stuffed young male that lacked a red breast. Male sexual maturation is often accompanied by these sex-specific characters that incite aggressive behaviour in other males.

There are numerous examples of male-male competition in a great variety of diverse species [49]. A male stag beetle has overdeveloped horns and an enlarged mandible that appear specialized for intraspecies fighting. Combat between male stalk-eyed flies in Australia includes a ritual in which the males

line up eye-to-eye and apparently determine which individual has a wider-set pair of eyes. Male salmon returning from the sea to spawn develop a hooked jaw that aids in fights with other males. As the males die shortly after spawning, the problems in feeding caused by the hooked jaw are of no consequence. Male elephant seals, due to the very limited breeding ground, are under intense sexual selection, and high mortality rates are not uncommon.

Demonstrations of male vigour appear to be a principal component of the evolutionary process that exposes and expurgates significant genetic errors from germline DNA. Those males that prove themselves 'fragile' through prolonged competition do not mate. Males have often become a sacrificial component of a reproductive subpopulation. Sexual selection may be viewed as operating as a gene defect filter by continually testing the male germline for vigour [53], and such male-male competition serves to cull genetic defects from the breeding deme prior to reproduction.

4.5 Simulated Evolution

Let us now discuss how the principles invoked in the neo-Darwinian paradigm are deployed in Simulated Evolution. The SE provides a stochastic search optimization technique and has been applied to difficult combinatorial problems such as the travelling salesman problem, training and designing neural networks and system identification. There are three widely researched paradigms in SE: Genetic Algorithms (GAs), Evolution Strategies (ES) and Evolutionary Programming (EP).

These evolutionary techniques are population based: Successive populations of the feasible solutions are generated in a stochastic manner following laws similar to that of natural selection. This is based on the Darwinian evolutionary theory which describes a point of view of 'survival of the fittest' in the history of life [86]. This is in contrast to standard programming techniques

that normally follow just one trajectory (deterministic or stochastic), perhaps repeated many times until a satisfactory solution is reached. In the evolutionary approach, multiple stochastic solution trajectories proceed simultaneously, allowing various interactions among them towards one or more regions of the search space. These approaches can be justified by the fact that a population-based algorithm automatically stores in time a sampled replica of the profile of the function being optimized, providing important clues for the global structure of the function.

SE generally takes the following form: a population trial of solutions to a particular problem is generated. Each of the initial solutions is scored with respect to some objective function. Through a variety of possible code operations (crossover, mutation, etc.), parents generate offsprings which are scored in a similar manner, the best subset of the solutions are retained to serve as parents for successive generations. The population iteratively adapts its behaviour in light of the given goal. The criteria for halting the simulation are typically based on the adequacy of the generated solution or a limit on the available execution time.

GA and EP have some obvious similarities. Both the algorithms operate on a population of candidate solutions, subject these solutions to alterations, and employ a selection criterion to determine which solutions to maintain for future generations. As described in [87, 88], GA differs from EP in the following aspects:

1. GAs use the binary coding (ie. bit string) of the parameters to be evolved, not the parameters themselves; whilst EP uses float point representations.

2. In GAs, the number of offspring to be created from each parent is proportional to the parent's fitness relative to all other members of the current population; in EP, successful simulations need not create more than a single offspring of each parent.

3. In GAs, parents create offspring through the use of specified genetic operators, such as crossover and mutation; in EP, the offsprings are created through various mutation operations that follow naturally from the chosen problem representation. Selection is often made a probabilistic function of fitness.

SE has developed extremely quick over the past two decades. Most of the early research concerned the numerical experiments and applications in various system engineering areas and both the results show the effective power of the SE. We will focus on two of the paradigms in the following two sections, namely the EP and the GA.

4.6 Genetic Algorithms

4.6.1 A Brief Historical Development

GAs have been developed by John Holland, his colleagues, and his students at the University of Michigan. The goals of their research have been twofold:

1. to abstract and rigorously explain the adaptive processes of natural systems;

2. to design artificial system software that retains the important mechanisms of natural systems.

This approach has led to important discoveries in both natural and artificial systems. Early work on GAs was carried out by Holland [89, 90]. In these two papers two fundamental concepts which are of chief importance for GAs are focused: the ability of simple representations to encode complex structures and the power of simple operations to improve them. In 1975 Holland set the frame work for the approach in searching and adaptation. This work can be

considered the birth of GAs. In this volume Holland established the theoretical results that, if certain conditions in the problem domain are assumed, GAs tend to converge toward the global optimal solution. Until the early 1980s a lot of theoretical studies on GAs was performed: some relevant results were produced by Hollstein [91], who made an analysis on the effects that different selection and mating strategies have on the performance of GAs, and by De Jong who tried to grasp the features of the adaptive mechanism in GAs [92]. Starting from the early 1980s, many system areas have been covered by GAs: Goldberg, for example, studied steady-state and transient operation of a gas pipeline [93] while Englander and Grefenstette and Fitzpatrick introduced GAs in pattern recognition problems [95, 96], Ackley and Cohoon et al applied GAs to neural networks [97, 98].

4.7 Algorithms Description

A GA consists of an $n-$tuple of binary strings b_i of length l, where the bits of each string are considered to be the genes of an individual chromosome and where the $n-$tuple of individual chromosomes is said to be a population. Following the terminology of organic evolution, the operations performed on the population are called *mutation, crossover* and *selection.* Each individual, b_i represents a feasible solution of the problem and its objective function value $f(b_i)$ is said to be its fitness, which is said to be maximised.

Davis describes the standard GA [99]. The standard procedure is sketched as follows:

Choose an initial population
determine the fitness of each individual
perform the selection
repeat

perform crossover
perform mutation
determine the fitness of each individual
perform selection
until some stopping criterion applies.

As suggested by Holland, the coding for a solution, termed a *chromosome* in GA literature, is described as a string of symbols from $\{0, 1\}$. These components of the chromosome are then labeled as *genes*. The number of bits that must be used to describe the parameters is problem dependent. Let each solution in the population of m such solutions, b_i, $i = 1, 2, \ldots, m$, be a string of symbols $\{0, 1\}$ of length l. Typically, the initial population of m solutions is selected completely at random, with each bit of each solution having a 50 percent chance of taking the value 0.

The GA parameter identification algorithm applied in this research for the parameter identification of an induction machine is desribed in detail in chapter 5. We will now discuss in more detail the three operators which will be used in simple GA.

4.7.1 Selection

The selection operator applied in the simple GA used in this research is named the *proportional selection*. This is because the population of the next generation is determined by n independent random experiments, the probability that individual b_i is selected from the tuple $(b_1, b_2, \ldots, b_n)$ to be a member of the next generation at each experiment is given by

$$P\{b_i \text{ is selected}\} = \frac{f(b_i)}{\sum_{j=1}^{n} f(b_j)} > 0 \tag{4.7.1}$$

This simple selection operator may be described as follows. Proportional selection may be thought of as placing the string probabilities on a weighted

roulette wheel and spinning the wheel to select a string. The probabilities on the roulette wheel are determined by the string's fitness as a percentage of the total population fitness. The roulette wheel selection utilizes random numbers to simulate a spin of the wheel. Once a string is selected by the reproduction operator, the string is copied into a mating pool and waits to be selected for further genetic operator action. The roulette wheel scheme does not guarantee that the fittest strings will be selected, although their probability for selection is high.

4.7.2 Crossover

Crossover is an important random operator in GAs and the one used in this work is the one point crossover. Typically, the probability for crossover ranges from 0.6 to 0.95 [92, 110]. From the biological point of view, the fundamental consequence of crossover is that the population entropy always increases. This means crossover wants to catch the population with the highest amount of energy.

Crossover may best be viewed by taking crossover as a two-step process that involves mating and swapping of partial strings. Each time the crossover operator takes action, two randomly selected strings from the mating pool are mated. Then, for one point crossover, a position along one string is selected at random, and all binary digits following the position are swapped with the second string. The result is two entirely new strings that move on to the next generation.

4.7.3 Mutation

Mutation operates independently on each individual by probabilistically perturbing each bit string. The event that the jth bit of the ith individual is flipped is stochastically independent and occurs with probability $p_m \in (0, 1)$.

Typically, the probability for bit mutation ranges from 0.001 to 0.01 [92, 110]. The probability that string b_i resembles string b_i' after mutation can be described as:

$$P\{b_i \rightarrow b_i'\} = p_m^{H(b_i,b_i')}(1-p_m)^{l-H(b_i,b_i')} > 0 \tag{4.7.2}$$

where $H(b_i, b_i')$ denotes the *Hamming* distance between the strings b_i and b_i'. The mutation operation is a secondary genetic algorithm. It is used to maintain diversity in the population–that is, to keep the population from prematurely converging on one solution–and to create genetic material that may not be present in the current population. An example of simple GA is shown in Appendix A.

4.8 Convergence Properties of Genetic Algorithms

Markov chains and stochastic Theorems offer an appropriate model to analyse GAs. They have been used to study and prove the probabilistic convergence of the SE algorithms [100, 101, 102, 103, 104, 105, 106, 107, 108, 109]. The Markov process is a sequence of possibly dependent random variables $(x_1, x_2, x_3, \ldots)$–identified by increasing values of a parameter, which for parameter identification of an induction machine is time, and has the property that any prediction of the value of x_n may be based on x_{n-1} alone. That is, the future value of the variable depends only upon the present value and not on the sequence of past values. Since the research work is built around the Simulated Evolution, these essentially discrete-valued variables are called Markov chains. We will firstly apply this method to the GA, and then later apply it to EP.

4.8.1 Genetic Algorithms without Bit Mutation

To provide an initial examination of the convergence properties of a GA, consider a state space of possible solutions to be coded as strings of n bits $\{0,1\}$. Let there be a population of m such bit strings, and let each possible configuration have a fitness f_i, $i = 1, \ldots, 2^n$. Let f^* be the globally optimum value. Let a GA with only crossover and selection (differential reproduction) operate on the bit strings.

The search can be formulated as a finite-dimension Markov chain [54] characterised by a state vector π and a transition Matrix $\mathbf{P}$. Given π, a row vector describing the probability of being in each state, $\pi\mathbf{P}$ yields the probability of being in each state after one transition. Thus the probability of being in each state after k transitions is $\pi\mathbf{P}^k$.

For GA, the states of the chain can be defined by every possible configuration of an entire population of bit strings. There are 2^{mn} such states. For example, if $n = 2$ and $m = 2$, then the following possibilities exist:

$$(00,00),(00,01),(00,10),(00,11),$$
$$(01,00),(01,01),(01,10),(01,11),$$
$$(10,00),(10,01),(10,10),(10,11),$$
$$(11,00),(11,01),(11,10),(11,11),$$

Each of these 16 collections of pairs of bit strings represents a possible state in the chain. For practical problems, often $m = 100$ and $n > 50$; thus there may be more than 2^{5000} possible states in the chain. Yet the chain is of finite dimension and possesses the *time homogeneous* and *no memory* properties required [55].

A Markov chain is said to be *irreducible* if every state communicates with every other state. Two states, i and j, are said to *communicate* if there is at least one path from i to j and vice versa. Such a path may require multiple steps (i.e., there exists $P_{ij}^x > 0$, $P_{ji}^y > 0$, for some $0 \leq x < M$, $0 \leq y < M$,

where there are M states and where P_{ij} is the probability of transitioning from state i to state j in one step).

The chain defined by the GA with only crossover and selection is not irreducible because there are absorbing states (i.e., states that do not communicate with any other state). Let the only *ergodic* classes (i.e., sets of states from which every path leads to a state in the set) be single absorbing states; no oscillations between states will be considered.

A state in this chain will be absorbing if all the members of the population are identical. Under such conditions, crossing over bit strings in the population simply yields the original population. Otherwise, a state is transient. As time progresses, the behaviour of the chain will be described by either (1) a transition to an absorbing state, (2) a transition to a state from which there may be a transition to an absorbing state with some nonzero probability, or (3) a transition to a state from which there is no probability of transitioning to an absorbing state in a single step. Thus the state can be indexed such that the state transition matrix, $\mathbf{P}$, for the chain satisfies

$$\mathbf{P} = \begin{bmatrix} \mathbf{I_a} & \mathbf{0} \\ \mathbf{R} & \mathbf{Q} \end{bmatrix} \tag{4.8.1}$$

where $\mathbf{P}$ is the transition matrix, $\mathbf{I_a}$ is an $a \times a$ identity matrix describing its absorbing states, $\mathbf{R}$ is a $t \times a$ transition submatrix describing transitions to an absorbing state, $\mathbf{Q}$ is a $t \times t$ transition submatrix describing transitions to transient states and not to an absorbing state, and a and t are positive integers.

The behaviour of such a chain satisfies

$$\mathbf{P}^k = \begin{bmatrix} \mathbf{I_a} & \mathbf{0} \\ \mathbf{N_k R} & \mathbf{Q^k} \end{bmatrix} \tag{4.8.2}$$

where $\mathbf{P}^k$ is the $k-$step transition matrix, $\mathbf{N}_k = \mathbf{I_t} + \mathbf{Q} + \mathbf{Q}^2 + \mathbf{Q}^3 + \ldots + \mathbf{Q}^{k-1}$,

and $\mathbf{I_t}$ is the $t \times t$ identity matrix. As k tends to infinity,

$$\lim_{k\to\infty} \mathbf{P}^k = \begin{bmatrix} \mathbf{I}_a & \mathbf{0} \\ (\mathbf{I_t} - \mathbf{Q})^{-1}\mathbf{R} & \mathbf{0} \end{bmatrix} \tag{4.8.3}$$

[55]. The matrix $(\mathbf{I_t} - \mathbf{Q})^{-1}$ is guaranteed to exist [55]. Therefore, given infinite time, the chain will transition with probability one to an absorbing state. Note that there is a nonzero probability that the absorbing state may not be the global best state unless all the absorbing states are globally optimal.

The results may be summarized by the following theorem.

Theorem. Let $\Gamma \equiv \{0, 1\}$, m be the number of solutions in Γ^n maintained at each iteration, the loss function $\mathbf{L} : (\Gamma^n)^m \rightarrow \mathbf{R_d}$, where $\mathbf{R_d}$ describes the set of real numbers representable in a given digital machine, and let $\mathbf{L}(\gamma)$, $\gamma \in (\Gamma^n)^m$, be single valued. After k iterations, the GA without bit mutation arrives at a state γ:

$$\gamma \in (\Gamma^n)^m \ni Pr(\gamma \in \mathbf{A}) = \sum_{i=1}^{a} (\pi^* \mathbf{P}^k)_i \tag{4.8.4}$$

$$= \sum_{i=1}^{a} \left(\pi^* \begin{bmatrix} \mathbf{I}_a \\ \mathbf{N_k R} \end{bmatrix} \right)_i \tag{4.8.5}$$

where $(\pi^* \mathbf{P}^k)_i$ denotes the ith element of the row vector $(\pi^* \mathbf{P}^k)$, $\mathbf{A}$ is the set of all absorbing states, π^* is the row vector describing the initial probability of being in each state, $\mathbf{P}$ is the transition matrix defining the Markov chain of the form

$$\mathbf{P} = \begin{bmatrix} \mathbf{I}_a & \mathbf{0} \\ \mathbf{R} & \mathbf{Q} \end{bmatrix} \tag{4.8.6}$$

$\mathbf{N}_k = \mathbf{I_t} + \mathbf{Q} + \mathbf{Q}^2 + \mathbf{Q}^3 + \ldots + \mathbf{Q}^{k-1}$, and $\mathbf{I}_a$ is an $a \times a$ identity matrix. The limit of the probability of absorption is

$$\lim_{k\to\infty} \sum_{i=1}^{a} \left(\pi^* \begin{bmatrix} \mathbf{I}_a \\ \mathbf{N_k R} \end{bmatrix} \right)_i = \sum_{i=1}^{a} \left(\pi^* \begin{bmatrix} \mathbf{I}_a \\ (\mathbf{I} - \mathbf{Q})^{-1}\mathbf{R} \end{bmatrix} \right)_i = 1 \tag{4.8.7}$$

It is natural to examine the number of absorbing states in such a chain. Absorbing states are those in which each bit string in the population is identical. There are 2^n such absorbing states. The total number of states is 2^{mn}, and this leads to two observations: The density of absorbing states decreases exponentially with the length of the bit string (i.e., $2^{n(1-m)}$), but the actual number of absorbing states increases exponentially with the length of the bit strings (i.e., 2^n). For coding other than bit strings, the base of 2 will change but the relationships will not.

Because the chain must eventually transition to an absorbing state under the rules of the transition matrix described above in (4.8.3), if the transitions between states are of equal probability, the time required to find an absorbing state may increase exponentially with n. When such a state is found, the likelihood of it being a globally optimum state may decrease exponentially with n. Of course, for matrices in which the transition probabilities are not equal, as would be the case in practice, the specific entries in each matrix will determine the mean waiting time before absorption and the likelihood of discovering a global optimum. Further, should any particular bit be fixed across the population to a value that is not associated with a global optimum solution, a global optimum will never be discovered.

4.8.2 Genetic Algorithms with Bit Mutation

A bit mutation operation is incorporated in GA to avoid the problems of premature convergence associated with the repeated use of crossover. 'Mutation is a background operator, assuring that the crossover operator has a full range of alleles (i.e., bits) so that the adaptive plan is not trapped on local optima' [88]. The term *local optima* may be somewhat misleading in this context, however. Typically, a point is locally optimal if no improvement can be made by searching in a nonempty neighbourhood around that point. This

may not be the case for GA relying solely on crossover. The sequence of trials may stagnate anywhere, at any homogenous collection of points (all identical). Under such conditions, the point is locally optimal only because the search algorithm is incapable of proceeding further. Many researchers [57, 58] have recognized this problem and some have proposed that to fine tune the search, hill-climbing and other heuristic procedures be employed after execution of a GA.

If each bit is given a probability of mutation (i.e., flipping), $p_m \in (0, 1)$, then the absorbing states of the transition matrix defined by the GA operating only with crossover that are not globally optimal will become transient. All globally optimal states can be collected and described as a single state. The mean time before entering this state if the chain starts in state i can be calculated by

$$\sum_{j \in T} k_{ij} \tag{4.8.8}$$

where $\mathbf{N} = (\mathbf{I} - \mathbf{Q})^{-1} = [k_{ij}]$, and $\mathbf{T}$ is the set of all transient states [55]. Thus the specific number of steps taken until reaching a globally optimal state is highly dependent on the characteristics of $\mathbf{P}$.

Because the probability for bit mutation is generally very low ([59], p. 15 suggests a 0.008 probability per bit), this operation may indeed be viewed as a background operator, following [88]. It will be likely that the chain of states generated by crossover and bit mutation will generally follow the state generated by crossover alone, with mutation serving to prevent complete stagnation [88]. Thus the path is likely to transition to a *metastable state* (an absorbing state of the chain defined by the GA relying on only crossover), whereupon it will wait for mutation to affect the appropriate changes of bits of chromosomes in the population and then proceed to another such metastable state, and so forth.

Such transitions may require a very long waiting time, depending on the number of bits that must be flipped simultaneously. The probability of flipping

b specific bits in a single chromosome (i.e., a bit string) and not flipping any others is $p_m^b(1-p_m)^{n-b}$, where p_m is the probability of flipping a single bit. As the length of the codings, n, grows longer, the number of metastable states increases exponentially and very long waiting times become virtually certain.

Davis [59] commented that 'when the population has converged on a chromosome that would require mutation of a good many bits to cause any improvement, the run of the GA is for all intents and purposes completed. Rather than continuing to run it to no avail, a better usage of computer cycles might be to run the algorithm again using a different random number seed and take the best results, or to use a hill-climbing heuristic to search methodically for improvements.

Premature convergence may also be a somewhat misleading term in this context. Rudolph [60], using Markov chain analysis, showed that the canonical GA [88] is not convergent at all, regardless of the initialization, crossover operator, and objective function: It does not generate a sequence of solutions that converges to any point in the sample space, including any global optima. But the search can easily be made to globally converge by *elitist selection*, that is, by incorporating a heuristic to always maintain the best solution in the population into successive generations. That is to say, the *elitist selection* ensures that the best individual (with highest fitness) survives with probability one. Premature convergence merely indicates that the search becomes stagnant for a long and random amount of time.

The specific waiting times in the metastable states are problem-dependent. There may well be problems for which such waiting times are relatively short or for which the paths to metastable states are unlikely. Because it is infeasible to analyze the behaviour of individual components in transition matrices that may be larger than $2^{5000} \times 2^{5000}$, the degree to which immensely long waiting times is endemic to GA can only be reasonably assessed through experimentation.

4.9 Evolutionary Programming

4.9.1 A Brief Historical Development

Fogel L. conceived of using SE on a population of contending algorithms to develop artificial intelligence and explored this possibility of EP in a series study [111, 112, 113, 114, 115]. Intelligence behaviour was viewed as requiring the composite ability to:

1. predict one's environment, coupled with
2. a transition of the predictions into a suitable response in light of the given goal.

For the sake of generality, the environment was described as a sequence of symbols taken from a finite alphabet. The evolutionary problem was defined as evolving an algorithm that would operate on the sequence of symbols thus far observed in such a manner so as to produce an output symbol that is likely to maximise the algorithm's performance in light of both the next symbol to appear in the environment and a well-defined payoff function.

Since 1985 research into EP has dramatically increased. More recently, the technique has been applied to diverse combinatorial optimization problems. Representations have been chosen based on the problem at hand and mutation operations are constructed that maintain a strong behavioural linkage between each parent and progeny. The procedure has been applied, for example, to the path planning problem, training and designing of neural networks, automatic control, gaming, and general function optimization.

4.10 Evolutionary Programming

There are three main operations in EP : mutation, competition and reproduction. The numerical process is as follows.

The initial population is determined by selecting individuals, p_i, $i = 1, 2, \ldots, m$, from the set of $U(a,b)^n$, where m is the population size and $U(a,b)^n$ denotes a uniform distribution ranging over $[a,b]$ in n dimensions. Each p_i, $i = 1, 2, \ldots, m$, is assigned a fitness score h_i, $h_i = H(p_i)$, $H : p_i \rightarrow \mathbf{R}$. H can be as complex as required and is usually used as an objective function. Statistics is then used to get the maximum fitness, minimum fitness, average fitness and sum of fitnesses of the population. The mutation operation is carried out based on the statistics to double the population size from m to $2m$. Each p_i, $i = 1, 2, \ldots, m$, is mutated and assigned to p_{i+m} in the following way:

$$p_{i+m,j} := p_{i,j} + N\left(0, \beta_j \frac{h_i}{h_\Sigma}\right), \qquad \forall j; j = 1, \ldots, n \tag{4.10.1}$$

where $p_{i,j}$ denotes the jth element of the ith individual; $N(\mu, \sigma^2)$ represents a Gaussian random variable with mean μ and variance σ^2; h_Σ is the sum of fitnesses; β_j is a constant of proportionality to scale h_i/h_Σ and $0 < \beta_j \leq 1$. Each p_{i+m}, $i = 1, 2, \ldots, m$, is again assigned a fitness score h_{i+m}. Based on the mutated population with the size of $2m$, a competition is conducted to reproduce offsprings. For each p_i, $i = 1, 2, \ldots, 2m$, a value w_i is assigned to weight the individual according to the following equation :

$$w_i = \sum_{t=1}^{s} w_t \tag{4.10.2}$$

and

$$w_t = \begin{cases} 1, & \text{if } u_1 < h_r/(h_r + h_i), \\ 0, & \text{otherwise} \end{cases} \tag{4.10.3}$$

where s is the number of competitors, r =int$(2mu_2 + 1)$, int(x) denotes the greatest integer less than x, and u_1, $u_2 \sim U(0,1)$. The individuals p_i, $i = 1, 2, \ldots, 2m$, are ranked in descending order of their corresponding value w_i. The first m individuals are transcribed along with their corresponding fitnesses h_i to be the basis of the next generation. The process will be carried out repeatedly until the given conditions are satisfied.

4.11 Algorithms Description

Fogel D. outlined the standard EP procedure for real-valued optimization problems [87]. This is described here with only a minor alteration in the competition rules. Assume the problem is function minimization. EP is conducted as follows:

1. Select an initial population of m trial solutions, p_i, $i = 1, \ldots, m$, by sampling from a uniform distribution across the preselected range of each parameter.

2. Score all parent solutions p_i, $i = 1, \ldots, m$, with respect to the chosen function $H(p)$.

3. From each parent p_i, $i = 1, \ldots, m$, create an offspring, denoted as p_{i+m}, by adding a Gaussian random variable with zero mean and positive variance set proportional to $H(p_i)$ to each component of the parent, where $H(p_i)$ is the parent's error score.

4. Score all offspring p_i, $i = m + 1, \ldots, 2m$, with respect to the chosen function $H(p)$.

5. For each p_i, $i = 1, \ldots, 2m$, select c competitors at random from a population. Conduct pairwise comparisons between p_i and each of the competitors. If the error score of p_i is less than or equal to that of its opponents, assign it a win.

6. Select the m solutions that have the greatest number of wins to be parents for the next generation.

7. If the available computing time has expired then halt, else proceed to step 3.

Various modifications of this procedure are described in [116]. The EP parameter identification algorithm applied in this research for the parameter identification of an induction machine is desribed in detail in chapter 5.

4.12 Convergence Properties of Evolutionary Programming

As with GA, the EP can be viewed as a finite-dimensional Markov chain. The states of the Markov chain again comprise the entire collection of vectors in the population. The components of the vectors are no longer bits $\{0, 1\}$ but rather discretized real numbers. For instance, for the following distribution ranges, if $a_1 = 0$, $a_2 = 0$, $b_1 = 1$, $b_2 = 1$, and the number of parameters $n = 2$, and the number in the population $m = 2$, then two possible instances of states in the chain are

$$(0, 0; 1, 1), (0.1, 0.523; 0.7, 0.003)$$

By the competition and selection rules in steps 5 and 6 of the EP algorithm, the best vector in the population will always be retained during selection. This results in the formation of an equivalence class of all states that contain a global best vector. This class may be described as one containing a single state. For example, if the global optimum is uniquely $(0, 0)$, then all collections of vectors containing $(0, 0)$ are characterized as the same state. The Markov chain may then be written in the form

$$\mathbf{P}_{EA} = \begin{bmatrix} 1 & 0 \\ \mathbf{R} & \mathbf{Q} \end{bmatrix} \tag{4.12.1}$$

where $\mathbf{P}_{EA}$ is the transition matrix, 1 is a 1×1 identity matrix decribing the absorbing state, $\mathbf{R}$ is a strictly positive (all entries are greater than zero) $t \times 1$ transition submatrix, $\mathbf{Q}$ is a $t \times t$ transition submatrix, and t is a positive

integer. The state containing global optima is the only absorbing state, and all other states are transient.

Asymptotic global convergence of the evolutionary algorithm is then transparent because every absorbing chain will reach an absorbing state, and in this case there is just one such state. Further, the matrix $\mathbf{P}_{EA}$ is a special case of the matrix in (4.8.1):

$$\mathbf{P} = \begin{bmatrix} \mathbf{I}_a & \mathbf{0} \\ \mathbf{R} & \mathbf{Q} \end{bmatrix} \tag{4.12.2}$$

It has already been shown (4.8.2) that a Markov chain described by $\mathbf{P}$ will proceed as

$$\mathbf{P}^k = \begin{bmatrix} \mathbf{I}_a & \mathbf{0} \\ \mathbf{N}_k\mathbf{R} & \mathbf{Q}^k \end{bmatrix} \tag{4.12.3}$$

Thus as in (4.8.3)

$$\lim_{k\to\infty} \mathbf{P}_{EA}^k = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ (\mathbf{I}_t - \mathbf{Q})^{-1}\mathbf{R} & \mathbf{0} \end{bmatrix} \tag{4.12.4}$$

Therefore, the probability of absorption (that is, the probability of discovering a global optimum solution) for the chain with transition matrix $\mathbf{P}_{EA}$ increases over k steps as a geometric series $(\mathbf{N}_k\mathbf{R})$ converging to 1.0. This result can be summarized in the following theorem.

Theorem. Let $\mathbf{\Gamma} \equiv \mathbf{R}_d$ the set of elements of $\mathbf{R}$ representable in a given digital machine, m be the number of maintained solutions, $\mathbf{L} : (\mathbf{\Gamma}^n)^m \rightarrow \mathbf{\Gamma}$ be the loss function, and $\mathbf{L}(\gamma), \gamma \in (\mathbf{\Gamma}^n)^m$, be single valued. Then, after k iterations, the evolutionary algorithm arrives at a state:

$$\gamma \in (\mathbf{\Gamma}^n)^m \ni Pr(\gamma \in \mathbf{A}) = (\pi^*\mathbf{P}_{EA}^k)_1 = \pi^* \begin{bmatrix} \mathbf{1} \\ \mathbf{N}_k\mathbf{R} \end{bmatrix} \tag{4.12.5}$$

where the subscript 1 denotes the first entry in the row vector $(\pi^*\mathbf{P}_{EA}^k)$, $\mathbf{A}$ is the singleton set containing the absorbing state, π^* is the $1 \times (t+1)$ row vector describing the initial probability of being in each state, $\mathbf{P}_{EA}$ is the transition

matrix defining the Markov chain, of the form

$$\mathbf{P}_{EA} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{R} & \mathbf{Q} \end{bmatrix} \tag{4.12.6}$$

and $\mathbf{N}_k = \mathbf{I}_t + \mathbf{Q} + \mathbf{Q}^2 + \mathbf{Q}^3 + \ldots + \mathbf{Q}^{k-1}$.

Furthermore, the limit of the probability of absorption is:

$$\lim_{k\to\infty} \pi^* \begin{bmatrix} \mathbf{1} \\ \mathbf{N}_k\mathbf{R} \end{bmatrix} = \pi^* \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = \sum_{i=1}^{t+1} \pi_i^* = 1 \tag{4.12.7}$$

This theorem directly indicates a convergence in probability, but can be extended to imply convergence with probability one. Convergence in probability sufficiently fast implies convergence with probability one [61]. More specifically, given a sequence x_k, if

$$\sum_{k=1}^{\infty} Pr(|\ x_k - x\ | > \varepsilon) < \infty \text{ for every } \varepsilon > 0 \tag{4.12.8}$$

then x_k converges with probability one to $x (x_k \overset{wp1}{\rightarrow} x)$.

This result can be applied to obtain the following theorem.

Theorem. Given a $(t+1) \times (t+1)$ state transition matrix of the form $\mathbf{P}_{EA}$ (4.12.1), where there is a single absorbing state and t transient states, and an initial $1 \times (t+1)$ state probability row vector π^* describing a Markov chain, let π_t^* be the $1 \times t$ row vector describing the probability of starting in each of the transient states. Form the sequence

$$x_k(\omega) = \begin{cases} 1, & \text{if the chain has reached the absorbing} \\ & \text{state by the } k\text{th iteration;} \\ 0, & \text{otherwise} \end{cases}$$

The sequence $x_k(\omega) \overset{wp1}{\rightarrow} 1$.

Proof

Consider the two cases (1) $k = 0$ or $k = 1$ and (2) $k \geq 2$.

1. The probability of arriving at the absorbing state by the zeroth or first iteration is simply $\pi^* \begin{bmatrix} \mathbf{1} \\ \mathbf{R} \end{bmatrix} = p_a$.

2. To analyze the sequence for $k \geq 2$, it is useful to note the following. For $\mathbf{vQ}^m \neq 0$, where $\mathbf{Q}$ is a $t \times t$ substochastic matrix (4.12.1), $\mathbf{v}$ is a $1 \times t$ row vector, $\mathbf{v} \neq 0$, $\mathbf{v}_i \geq 0 \; \forall \; i = 1, \ldots, t$, $\forall \; m = 0, 1, 2, \ldots$ ($\mathbf{vQ}^m = 0$ indicates $x_k(\omega) \overset{wpl}{\rightarrow} 1$, trivially), since $q_{ij} < 1$,

$$\begin{aligned} \left\| \frac{\mathbf{v}}{\| \mathbf{v} \|_1} . \mathbf{Q} \right\|_2 &< \left\| \frac{\mathbf{v}}{\| \mathbf{v} \|_1} . \mathbf{Q} \right\|_1 \\ &= \| \pi \mathbf{Q} \|_1, \qquad \pi = \frac{\mathbf{v}}{\| v \|_1} \\ &= \sum_{i=1}^{t} \pi_i \sum_{j=1}^{t} q_{ij} \\ &\leq \sum_{i=1}^{t} \pi_i \max_i \sum_{j=1}^{t} q_{ij} \\ &= \max_i \sum_{j=1}^{t} q_{ij} \\ &= \gamma \\ &< 1 \end{aligned} \tag{4.12.9}$$

Hence, $\forall \; k \geq 2$, $\pi_t^* \neq 0$, $\pi_{t,i}^* \geq 0 \; \forall \; i = 1, \ldots, t$, and $\varepsilon > 0$,

$$\begin{aligned} P(\mid x_k(\omega) - 1 \mid > \varepsilon) &= P(x_k(\omega) \neq 1) \\ &= \pi_t^* \mathbf{Q}^k \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \end{aligned} \tag{4.12.10}$$

Therefore,

$$\sum_{k=2}^{\infty} P(x_k(\omega) \neq 1) = \sum_{k=2}^{\infty} \pi_t^* \mathbf{Q}^k \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

$$
\begin{aligned}
&= \sum_{k=2}^{\infty} \left\| \pi_t^* \mathbf{Q}^k \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \right\|_2 \\
&\leq \sum_{k=2}^{\infty} \| \pi_t^* \mathbf{Q}^k \|_2 \sqrt{t} \text{ (Cauchy-Schwarz)} \\
&= \sum_{k=2}^{\infty} \| \pi_t^* \mathbf{Q}^{k-1} \mathbf{Q} \|_2 \sqrt{t} \\
&= \sum_{k=2}^{\infty} \| \pi_t^* \mathbf{Q}^{k-1} \|_1 \left\| \frac{\pi_t^* \mathbf{Q}^{k-1}}{\|\pi_t^* \mathbf{Q}^{k-1}\|_1} . \mathbf{Q} \right\|_2 \sqrt{t} \\
&\leq \sum_{k=2}^{\infty} \| \pi_t^* \mathbf{Q}^{k-1} \|_1 . \gamma . \sqrt{t}, \quad 0 \leq \gamma < 1 \\
&= \sum_{k=2}^{\infty} \| \pi_t^* \mathbf{Q}^{k-2} \|_1 \left\| \frac{\pi_t^* \mathbf{Q}^{k-2}}{\|\pi_t^* \mathbf{Q}^{k-2}\|_1} . \mathbf{Q} \right\|_1 \gamma \sqrt{t} \\
&\leq \sum_{k=2}^{\infty} \| \pi_t^* \mathbf{Q}^{k-2} \|_1 . \gamma^2 . \sqrt{t} \\
&\leq \sum_{k=2}^{\infty} \| \pi_t^* \|_1 . \gamma^k . \sqrt{t} \\
&\leq \sum_{k=2}^{\infty} \mathbf{K} \gamma^k \\
&< \infty. \qquad (4.12.11)
\end{aligned}
$$

4.13 Conclusion

The limitations of traditional optimization techniques, namely the calculus-based methods, numerative schemes and random search algorithms, have been discussed which has justified the reasons for using SE. The following chapter will apply SE for the parameter identification problem of an induction machine simulation with differing variances of measurement noise.

Chapter 5

The Induction Machine Simulations

5.1 Introduction

In this chapter we will look at the induction machine's dynamic differential equations which are suitable for simulation and therefore provide an analytical tool of an actual induction machine. The dynamic model notation is discussed which is used to mathematically describe the simulated evolution processes that deploy the induction machine model during the parameter identification process. The different strategies deployed by the two chosen processes of simulated evolution used in the parameter identification process are explained and listed, namely the genetic algorithm (GA) and the evolutionary programming (EP). The induction machine's model in the per-unit system is described, which allows the simulated evolution to be more effective in the optimal parameter search by keeping the initial parameter ranges within a reasonable range in the search space. This is concluded by the simulation process, tabulated results and responses which are fully illustrated and discussed.

5.2 The Induction Machine's Differential Equations Used for Simulation

By meticulous rearrangement of equations (3.1.2), (3.4.1) and (3.5.1), the induction machine's state equations are as follows

$$Di_{ds} = \sigma(-R_s L_r' i_{ds} + L_m^2 i_{qs}\omega_r + L_m R_r' i_{dr}' + L_m L_r' i_{qr}'\omega_r) + \sigma L_r' v_{ds} \quad (5.2.1)$$

$$Di_{qs} = \sigma(-L_m^2 i_{ds}\omega_r - R_s L_r' i_{qs} - L_m L_r' i_{dr}'\omega_r + L_m R_r' i_{qr}') + \sigma L_r' v_{qs} \quad (5.2.2)$$

$$Di_{dr}' = \sigma(R_s L_m i_{ds} - L_s L_m i_{qs}\omega_r - L_s R_r' i_{dr}' + L_s L_r' i_{qr}'\omega_r) - \sigma L_m v_{ds} \quad (5.2.3)$$

$$Di_{qr}' = \sigma(L_s L_m i_{ds}\omega_r + R_s L_m i_{qs} + L_s L_r' i_{dr}'\omega_r - L_s R_r' i_{qr}') - \sigma L_m v_{qs} \quad (5.2.4)$$

$$D\omega_r = \frac{Q}{J}[QL_m(-i_{ds} i_{qr}' + i_{qs} i_{dr}') + T_r - \frac{B\omega_r}{Q}] \quad (5.2.5)$$

where

$$\sigma = \frac{1}{L_s L_r - L_m^2} \quad (5.2.6)$$

The state equations shown above are of the nonlinear differential equation form for which we shall use the Runge-Kutta method.

5.3 Simulation of a 3-Phase Induction Machine

A 3-phase induction machine with the specifications given in Table 1 shall be simulated applying the Kron's two-axis dq dynamic mathematical model.

Table 1 Specifications of the 3-Phase Induction Machine

Induction Machine type	TYPZK 90L-6/IEC 34-1
Output power P_{out}	1100 W
Voltage	220/380 V
Current	5.7/3.3 A
Frequency	50 Hz
Speed	910 rpm
Stator resistance R_s	5.85 Ω
Rotor resistance R_r	5.87 Ω
Stator inductance L_s	0.252 H
Rotor inductance L_r	0.252 H
Magnetizing inductance L_m	0.2346 H
Moment of inertia J	0.005
Damping coefficient B	0.0008
Rated torque T_r	11.45 Nm

5.4 The Dynamic Model Notation Used for the Induction Machine

Let the parameter identification problem be described as follows:

$$DX = f(\theta, X, V) \tag{5.4.1}$$

where

X = the state variable vector

V = the stimulus, i.e., the input vector from the supply

θ = the parameter variable vector to be identified

D = the differential operator $\frac{d}{dt}$

Let

$$p = [p_1\ p_2 \cdots p_l]^T \tag{5.4.2}$$

be the parameter variable matrix, and

$$p_i = [p_{i1}\ p_{i2} \cdots p_{im}] \qquad (i = 1, 2, \cdots, l) \tag{5.4.3}$$

the parameter vector, where l is the number of the individuals in a generation, and m is the number of parameters to be identified.

$$Y(t_k) = CX(t_k) + w(t_k) \tag{5.4.4}$$

is the measured performance vector which is contaminated by the measurement noise, $w(t_k)$, and k is the sequence number of the series measurement sample, t_k is the measuring time instant corresponding to the kth measurement. The whole measurement is suppose to start at $t_k = 0$ and end at $t_k = T_N$; T_N is the time span of the series measurement sample. w is the measurement noise vector, and C is a coefficient matrix depending on which measurements of performance are used.

The measurement vector, $Y(t_k)$, can be alternatively expressed as:

$$Y(t_k) = [y_1(t_k)\ y_2(t_k) \cdots y_M(t_k)]^T \tag{5.4.5}$$

where the measurement $y_i(t_k)$ $(i = 1, 2, \cdots, M)$ can be either a state variable or a non-state variable which includes the measurement noise. M is the number of variables to be measured. Let g stand for the sequence number of generations. The following two sections describe the processes involved in parameter identification of an induction machine, firstly applying genetic algorithms which is followed by a section on applying evolutionary programming. The similarities between the two processes are repeated for greater clarity as well as to show the different strategies of the two simulated evolution methods.

5.5 Identification of an Induction Machine's Dynamic Model Parameters applying Genetic Algorithms

1. An initial population of parameters, $\hat{p}^{(0)} = \{\hat{p}_i^{(0)} \mid i = 1, 2, \ldots, l\}$, is formed with randomly selected individuals. Each individual parameter vector is constrained with the following conditions:

$$p_{min,j} \leq \hat{p}_{i,j}^{(g)} \leq p_{max,j} \qquad j = 1, 2, \ldots, m \tag{5.5.1}$$

where $p_{min,j}$ and $p_{max,j}$ are the limits of the jth element of the parameter vector, $\hat{p}_i^{(g)}$, given with *a prior* knowledge. $\hat{p}_{i,j}^{(g)}$ denotes the estimation of the jth element of the ith individual in the gth generation of the parameter. The process starts at $g := 0$.

2. Each individual, $\hat{p}_i^{(g)}$, is used to calculate the state variable $\hat{X}(t_k)$ and the general performance $\hat{Y}(t_k)$ as follows:

$$DX(t_k) = f(\hat{p}_i^{(g)}, X(t_k), V(t_k)) \tag{5.5.2}$$

$$\hat{Y}(t_k) = CX(t_k) \tag{5.5.3}$$

and the error vector

$$E(\hat{p}_i^{(g)}, t_k) = Y(t_k) - \hat{Y}(t_k) \tag{5.5.4}$$

where $Y(t_k)$ stands for the measured performance vector, $\hat{Y}(t_k)$ the simulated one and E the expectation operator. We thus get an error function of $\hat{p}_i^{(g)}$:

$$h(\hat{p}_i^{(g)}) = \sum_{t_k=0}^{T_N} E(\hat{p}_i^{(g)}, t_k)^T \Lambda E(\hat{p}_i^{(g)}, t_k) \tag{5.5.5}$$

where Λ is a unit matrix, t_k is the discrete computing time which is the same as the measuring time instant. This leads to the fitness of $\hat{p}_i^{(g)}$:

$$Fitness = \frac{1}{1 + h(\hat{p}_i^{(g)})} \tag{5.5.6}$$

The aim of the genetic algorithm is to minimize the error function or to maximize the fitness.

3. A new generation, $\hat{p}^{(g+1)}$, with the same individual number of $\hat{p}^{(g)}$, is formed by means of reproduction, crossover, and mutation based on the previous $\hat{p}^{(g)}$.

4. The range of the fitnesses of the individuals in a generation is calculated as follows:

$$maxfitness - minfitness < \varepsilon \tag{5.5.7}$$

where

$$\varepsilon = 0.001 * minfitness \tag{5.5.8}$$

During the simulation, each time the statement above is true, the following integer variable, *convergence*, is incremented as follows:

$$convergence := convergence + 1 \tag{5.5.9}$$

Should the integer variable *convergence* become equal to an integer constant, *maxconvergence*, which has been initially set at a particular chosen integer value, then the simulation will stop as it has been judged that the simulation has converged sufficiently, thus identifying the required parameters $\theta = \hat{p}_i^{(g)}$ corresponding to a minimum error function, or alternately, a maximum *Fitness*. ε is a very small real. The process will also stop if $g = MG$, where MG is a big integer representing the maximum generation number, which has also been initially set. If the convergence criterion is not satisfied, $g := g + 1$, and the process is repeated from step 2.

5.6 A Multiparameter, Mapped, Fixed-Point Coding

As mentioned in Chapter 4, the population in simple GA consists of individuals, each representing a chromosome. Each individual, however, comprises of parameters representing the problem to be solved. How these parameters map onto a chromosome shall now be discussed. The parameters to be identified with respect to the induction machine are the stator resistance, R_s, the rotor resistance, R_r, the mutual inductance, L_m, the stator inductance L_s, and the rotor inductance, L_r. These parameters, for the purpose of this parameter identification research, are all assumed to lie betweem a range of $\pm 50\%$ of their real values. The R_s range is 2.925Ω to 8.775Ω, the R_r range is 2.935Ω to 8.805Ω, the L_m range is $0.1173H$ to $0.3519H$, and both L_s and L_r ranges are $0.126H$ to $0.378H$. Since the population's individuals in simple GA comprise of chromosomes, which are binary bit strings, we shall map the decoded unsigned integer linearly from $[0, 2^l]$ to the specified parameter range $[U_{min}, U_{max}]$. This procedure is done for a multiparameter problem by concatenating each parameter as a binary substring so that the chromosome binary string comprises of the required number of binary substrings, each one representing a particular coded parameter. The mechanism for multiparameter, mapped, fixed-point coding is as follows:

$$U_{min} + \left(\frac{U_{max} - U_{min}}{2^l - 1}\right) x$$

where

- U_{min}–minimum real parameter value,
- U_{max}–maximum real parameter value,
- l–particular binary substring bit length,
- x–real integer value between 0 and $2^l - 1$.

The step or precision of the binary substring is described by the following portion of the coding:

$$\left(\frac{U_{max} - U_{min}}{2^l - 1}\right)$$

For the purposes of this research a chromosome of binary string length of 70 bits was chosen for the precision in the search space this would offer. Both R_s and R_r have binary substring lengths of 14 bits and precisions of approximately 0.0003. L_m has a binary substring length of 16 bits and a precision of approximately 0.000003. And both L_s and L_r have binary substring lengths of 13 bits and precisions of approximately 0.00003. In view of the induction machine's parameter decimal fraction sizes, the numerical precision for the parameter identification process is certainly more than adequate.

5.7 Identification of an Induction Machine's Dynamic Model Parameters Applying Evolutionary Programming

1. An initial population of parameters, $\hat{p}^{(0)} = \{\hat{p}_i^{(0)} \mid i = 1, 2, \ldots, l\}$, is formed with randomly selected individuals. Each individual parameter vector is constrained with the following conditions:

$$p_{min,j} \leq \hat{p}_{i,j}^{(g)} \leq p_{max,j} \qquad j = 1, 2, \ldots, m \tag{5.7.1}$$

where $p_{min,j}$ and $p_{max,j}$ are the limits of the jth element of the parameter vector, $\hat{p}_i^{(g)}$, given with *a prior* knowledge. $\hat{p}_{i,j}^{(g)}$ denotes the estimation of the jth element of the ith individual in the gth generation of the parameter. The process starts at $g := 0$.

2. Each individual, $\hat{p}_i^{(g)}$, is used to calculate the state variable $\hat{X}(t_k)$ and

the general performance $\hat{Y}(t_k)$ as follows:

$$DX(t_k) = f(\hat{p}_i^{(g)}, X(t_k), V(t_k)) \tag{5.7.2}$$

$$\hat{Y}(t_k) = CX(t_k) \tag{5.7.3}$$

and the error vector

$$E(\hat{p}_i^{(g)}, t_k) = Y(t_k) - \hat{Y}(t_k) \tag{5.7.4}$$

where $Y(t_k)$ stands for the measured performance vector, $\hat{Y}(t_k)$ the simulated one and E the expectation operator. Then we get the error function of $\hat{p}_i^{(g)}$:

$$h(\hat{p}_i^{(g)}) = \sum_{t_k=0}^{T_N} E(\hat{p}_i^{(g)}, t_k)^T \Lambda E(\hat{p}_i^{(g)}, t_k) \tag{5.7.5}$$

where Λ is a unit matrix, t_k is the discrete computing time which is the same as the measurement sampling time. This leads to the fitness of $\hat{p}_i^{(g)}$:

$$Fitness = \frac{h(\hat{p}_i^{(g)})}{1 + h(\hat{p}_i^{(g)})} \tag{5.7.6}$$

The aim of the evolutionary programming, therefore, is to minimize the error function or to minimize the fitness.

3. A Gaussian random variable, with zero mean and its variance proportional to the fitness of $\hat{p}_i^{(m)}$ scaled by β_j, is added to each element of $\hat{p}_i^{(g)}$, $i = 1, 2, \ldots, l$, to produce new individuals $\hat{p}_{i+l}^{(g)}$. β is a factor in association with the jth identified parameter in the parameter vector. $\hat{p}_{i+l}^{(g)}$ has to satisfy the conditions laid down by expression (5.7.1).

4. The new individuals, $\hat{p}_{i+l}^{(g)}$, where $i+l = l+1, \ldots, 2l$, are used to calculate the general performance index as their predecessors to get their fitness, $h(\hat{p}_{i+l}^{(g)})$.

5. Each individual, $\hat{p}_i^{(g)}$, where $i \in \{1, \ldots, 2l\}$, competes with some others according to the competition rule. A rank of competition is set up.

6. A new generation, $\hat{p}^{(g+1)}$, with the same individual number of $\hat{p}^{(g)}$, is formed based on the first l ranked $\hat{p}_i^{(g)}$, $i \in \{1, 2, \ldots, 2l\}$.

7. The range of the fitnesses of the individuals in a generation is calculated as follows:

$$maxfitness - minfitness < \varepsilon \tag{5.7.7}$$

where

$$\varepsilon = 0.001 * minfitness \tag{5.7.8}$$

During the simulation, each time the statement above is true, the following integer variable, *convergence*, is incremented as follows:

$$convergence := convergence + 1 \tag{5.7.9}$$

Should the integer variable *convergence* become equal to an integer constant, *maxconvergence*, which has been initially set at a particular chosen integer value, then the simulation will stop as it has been judged that the simulation has converged sufficiently, thus identifying the required parameters $\theta = \hat{p}_i^{(g)}$ corresponding to a minimum error function, or alternately, a minimum *Fitness*. ε is a very small real. The process will also stop if $g = MG$, where MG is a big integer representing the maximum generation number, which has also been initially set. If the convergence criterion is not satisfied, $g := g + 1$, and the process is repeated from step 2.

5.8 The Induction Machine Model in the Per-Unit System

Chapter 3 described the dynamic mathematical model of a symmetrical induction machine requiring actual parameter values. The mathematical model

using the per-unit system, as well as the reasons for using such a system, will now be discussed.

The per-unit system has many advantages over the actual value notation. In the per-unit system, the values of the machines parameters fall into well defined ranges so that the initial ranges of the parameters required by (6.5.2) can be easily defined and limited. All responses of the machine model in the per-unit system are of the same order of magnitude. This will contribute to the accuracy of the parameter estimate.

Let V_n be the rated voltage applied to a stator phase winding of the induction machine, I_n the rated current, f_1 stands for the supply frequency, V_b, I_b, ω_b, R_b, L_b, P_b and T_b stand for the base values of the voltage, current, electrical angular speed, resistance, inductance, power and torque respectively.

$$V_b = V_n \qquad I_b = I_n \qquad \omega_b = \omega_1 = 2\pi f_1 \tag{5.8.1}$$

$$R_b = \frac{V_b}{I_b} \qquad L_b = \frac{V_b}{\omega_b I_b} \qquad P_b = m_{ph} V_b I_b \qquad T_b = \frac{QP_b}{\omega_b} \tag{5.8.2}$$

where ω_1 is the synchronous angular speed, Q denotes the number of pole pairs; m_{ph} denotes the number of stator winding phases. Hence, the symmetrical induction machine dynamic mathematical model, as utilized in the previous section, may be expressed in per-unit values as follows [119]:

$$\begin{bmatrix} v_{ds} \\ v_{qs} \\ v'_{dr} \\ v'_{qr} \end{bmatrix} = \begin{bmatrix} R_s + \frac{L_s D}{\omega_1} & 0 & \frac{L_m D}{\omega_1} & 0 \\ 0 & R_s + \frac{L_s D}{\omega_1} & 0 & \frac{L_m D}{\omega_1} \\ \frac{L_m D}{\omega_1} & L_m \omega_r & R'_r + \frac{L'_r D}{\omega_1} & L'_r \omega_r \\ -L_m \omega_r & \frac{L_m D}{\omega_1} & -L'_r \omega_r & R'_r + \frac{L'_r D}{\omega_1} \end{bmatrix} \begin{bmatrix} i_{ds} \\ i_{qs} \\ i'_{dr} \\ i'_{qr} \end{bmatrix} \tag{5.8.3}$$

$$T_e = \frac{1}{3} L_m Q (i_{ds} i_{qr} - i_{qs} i_{dr}) \tag{5.8.4}$$

$$\left(\frac{\omega_1}{Q}\right)^2 \left(\frac{J}{P_b}\right) \frac{d\omega_r}{dt} = -T_e + T_r - \left(\frac{\omega_1}{Q}\right) \left(\frac{B}{T_b}\right) \omega_r \tag{5.8.5}$$

As was the case of the actual valued model, the rotor speed, ω_r, is negative for an induction motor. The damping coefficient, B, and the moment of inertia, J, are actual values, all the others are per-unit values.

5.9 The Induction Machine's Equations in the Per-Unit System

Meticulous rearrangement of equations (5.8.3), (5.8.4) and (5.8.5), yields the induction machine's state equations as :

$$Di_{ds} = \sigma(-R_s L_r^{'} i_{ds} + L_m^2 i_{qs}\omega_r + L_m R_r^{'} i_{dr}^{'} + L_m L_r^{'} i_{qr}^{'}\omega_r) + \sigma L_r^{'} v_{ds} \quad (5.9.1)$$

$$Di_{qs} = \sigma(-L_m^2 i_{ds}\omega_r - R_s L_r^{'} i_{qs} - L_m L_r^{'} i_{dr}^{'}\omega_r + L_m R_r^{'} i_{qr}^{'}) + \sigma L_r^{'} v_{qs} \quad (5.9.2)$$

$$Di_{dr}^{'} = \sigma(R_s L_m i_{ds} - LsL_m i_{qs}\omega_r - LsR_r^{'} i_{dr}^{'} + L_s L_r^{'} i_{qr}^{'}\omega_r) - \sigma L_m v_{ds} \quad (5.9.3)$$

$$Di_{qr}^{'} = \sigma(L_s L_m i_{ds}\omega_r + R_s L_m i_{qs} + L_s L_r^{'} i_{dr}^{'}\omega_r - LsR_r^{'} i_{qr}^{'}) - \sigma L_m v_{qs} \quad (5.9.4)$$

$$D\omega_r = \left(\frac{P_b}{J}\right)\left(\frac{Q}{\omega_1}\right)^2 [\frac{1}{3}L_m(-i_{ds}i_{qr} + i_{qs}i_{dr}) + T_r] - \frac{B}{J}\omega_r \quad (5.9.5)$$

where

$$\sigma = \frac{\omega_1}{L_s L_r - L_m^2} \quad (5.9.6)$$

As with the actual valued state equations of the previous section, these are also nonlinear differential equations for which the forth-order Runge-Kutta method shall be deployed for thier approximate solutions.

5.10 Implementation of the Simulated Evolution and its Simulation Results

Both EP and GA have been employed to identify the parameters of the induction machine's dynamic mathematical model in the optimization process both in actual values and in the per-unit system, being based on a no-load, direct on-line start. Chapter 4, Section 7, describes the algorithmic process and the terminology of organic evolution for GA. In the case of GA, the probability of crossover, p_c, was chosen to be 0.9 in order to allow a greater probability of crossover in the chromosome population whilst still retaining a probable few

unchanged chromosomes. In simple GA, reproduction and crossover are the dominant search operators but may lose some useful genetic material (1's and 0's at particular chromosome positions) [94]. The mutation operator protects against such irrecoverable losses. Empirical GA studies on mutation yields good results on the order of about one mutation per thousand bit transfers [94]. Therefore, the probability of mutation, p_m, was chosen to be 0.005. [57] has reported that increasing p_m by more than a factor of 10 of this probability, has mentioned in Section 4.7.3., degrades the performance of GA.

The 3-phase induction machine parameter identification optimization process was done over the transient, no-load, direct on-line start. The phase angle between the machine's 3-phase currents is only established when steady state conditions have been achieved. To investigate the effects on the optimization process of the number of state variables used, four and two state variables, for the actual Kron's two-axis dq dynamic mathematical model simulations, have been chosen. The two measurement vectors which have achieved the afore mentioned objectives are respectively defined as:

$$Y_1(t_k) = [i_{rs}(t_k)\ i_{ys}(t_k)\ i_{bs}(t_k)\ \omega_r(t_k)]^T \tag{5.10.1}$$

and

$$Y_2(t_k) = [i_{rs}(t_k)\ \omega_r(t_k)]^T \tag{5.10.2}$$

the corresponding coefficient matrices C_1 and C_2 for equations (5.5.3) and (6.5.6) are respectively:

$$C_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 0 & 0 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.10.3}$$

and

$$C_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.10.4}$$

so that the calculated performance corresponding to the measurement $Y_1(t_k)$ and $Y_2(t_k)$ will be:

$$\hat{Y}_1(t_k) = C_1 X(t_k) \tag{5.10.5}$$

and

$$\hat{Y}_2(t_k) = C_2 X(t_k) \tag{5.10.6}$$

where

i_{rs} = the current in stator winding Red

i_{ys} = the current in stator winding Yellow

i_{bs} = the current in stator winding Blue

The parameter identification was undertaken for five cases of different levels of measurement noise $w(t)$. The measurement noise was chosen to be a Gaussian random variable, $N(\mu, \sigma^2)$, where μ is the mean (set to zero) and σ^2 is the variance of the noise. The noise variances for the five cases are listed in Table 2. The ranges of the identified parameters are assumed to be $\pm 50\%$ per cent of the real values that are determined from the traditional tests. For both the EP and GA, the maximum generation number is $MG = 2000$ for the model in actual values and $MG = 1000$ for the model in the per-unit system. The number of individuals in a generation, l, is set to 50 for both GA and EP.

Table 2 Measurement Noise Variances σ^2

	i_{rs}	i_{ys}	i_{bs}	ω_r
case 1	0.0005	0.0005	0.0005	0.0005
case 2	0.005	0.005	0.005	0.005
case 3	0.01	0.01	0.01	0.01
case 4	0.05	0.05	0.05	0.05
case 5	0.10	0.10	0.10	0.10

The period for the simulation and the test is set at is set to 0.3 seconds, that is, $T_N = 0.3$ seconds. The error, E and the maximum error, ME are respectively defined as:

$$Error_i(\%) = \frac{\theta_i - \hat{\theta}_i}{\theta_i} \times 100\% \qquad (i = 1, 2, \ldots, 5) \tag{5.10.7}$$

$$ME(\%) = Maximum\{Error_1, Error_2, \ldots, Error_5\} \tag{5.10.8}$$

where $\hat{\theta}_i$ stands for the identified parameters, and θ_i its real value. $y_j(t_k)$ stands for the performance measured, $\hat{y}_j(t_k)$ the corresponding performance simulated by the parameter $\hat{\theta}$.

5.10.1 Tabulation of Simulated Evolution Simulation Results

Tables 3 to 8 give the results of the parameter identification of the induction machine's mathematical model both in actual values and in the per-unit system, using the EP and the GA, based on the measurement vectors $Y1$ and $Y2$. The maximum generation (MG) used for the actual value parameter search is 2000 whilst 1000 for the per-unit system parameter search. The results include the identified parameters, their fitness (F), their maximum error (ME) and the number of generations that have occured in which EP or GA have found the identified parameters (FG). The convergence number used for this process in both EP and GA was 50. RV stands for the real values of the parameters. Figure 5.1 shows the start-up performance of stator current i_{rs}, electrical torque T_e and rotor speed ω_r which are contaminated with noise.

Table 3

The Identified Parameters Applying EP to the Model in Actual Values

(Using $l = 50$, $MG = 2000$ and $Y1$)

	$R_s(\Omega)$	$R_r(\Omega)$	$L_m(H)$	$L_s(H)$	$L_r(H)$	$ME(\%)$	FG	F
RV	5.85	5.87	0.2346	0.252	0.252			
Case 1	5.8874	5.9369	0.2438	0.2485	0.2508	7.61	109	0.2483
Case 2	6.0359	5.9425	0.2409	0.2463	0.2494	10.37	118	0.2491
Case 3	6.0621	5.6062	0.2389	0.2420	0.2517	14.04	193	0.4887
Caes 4	6.1425	5.9999	0.2393	0.2433	0.2549	13.85	144	0.4957
Case 5	6.0382	5.5767	0.2375	0.2405	0.2552	15.27	163	0.4924

Table 4

The Identified Parameters applying GA to the Model in Actual Values

(Using $l = 50$, $MG = 2000$ and $Y1$)

	$R_s(\Omega)$	$R_r(\Omega)$	$L_m(H)$	$L_s(H)$	$L_r(H)$	$ME(\%)$	FG	F
RV	5.85	5.87	0.2346	0.252	0.252			
Case 1	6.1052	5.7959	0.2326	0.2405	0.2533	11.56	MG	0.2530
Case 2	6.0591	5.8090	0.2274	0.2479	0.2444	12.33	MG	0.0813
Case 3	5.9272	5.7597	0.2369	0.2395	0.2611	12.75	MG	0.0100
Caes 4	6.0355	6.1070	0.2378	0.2411	0.2582	15.36	MG	0.0025
Case 5	6.1188	6.0933	0.2434	0.2489	0.2438	16.63	MG	0.0012

Table 5

The Identified Parameters applying EP to the Model in Actual Values

(Using $l = 50$, $MG = 2000$ and $Y2$)

	$R_s(\Omega)$	$R_r(\Omega)$	$L_m(H)$	$L_s(H)$	$L_r(H)$	$ME(\%)$	FG	F
RV	5.85	5.87	0.2346	0.252	0.252			
Case 1	5.7509	5.8198	0.2454	0.2483	0.2499	9.46	176	0.3334
Case 2	5.6342	5.7729	0.2450	0.2476	0.2589	14.99	195	0.3424
Case 3	6.0836	5.6871	0.2431	0.2471	0.2582	15.12	353	0.3528
Caes 4	6.1424	5.6056	0.2363	0.2416	0.2463	16.61	280	0.4753
Case 5	6.1425	5.5765	0.2340	0.2415	0.2606	17.84	321	0.4870

Table 6

The Identified Parameters applying GA to the Model in Actual Values

(Using $l = 50$, $MG = 2000$ and $Y2$)

	$R_s(\Omega)$	$R_r(\Omega)$	$L_m(H)$	$L_s(H)$	$L_r(H)$	$ME(\%)$	FG	F
RV	5.85	5.87	0.2346	0.252	0.252			
Case 1	5.7918	5.8061	0.2450	0.2495	0.2574	9.65	MG	0.6969
Case 2	5.6645	5.9886	0.2449	0.2477	0.1535	11.88	MG	0.5655
Case 3	6.1327	5.8267	0.2394	0.2458	0.2457	12.58	MG	0.5182
Caes 4	5.6840	5.7259	0.2367	0.2411	0.2645	15.47	MG	0.0568
Case 5	6.0355	6.1070	0.2378	0.2412	0.2457	15.36	MG	0.0003

Table 7

The Identified Parameters applying EP to the Model in the Per-Unit System

(Using $l = 50$, $MG = 1000$ and $Y1$)

	R_s	R_r	L_m	L_s	L_r	$ME(\%)$	FG	F
RV	0.08775	0.08805	1.10553	1.18522	1.18522			
Case 1	0.08838	0.08740	1.07715	1.17810	1.12818	9.44	143	0.0152
Case 2	0.08586	0.08446	1.09301	1.17900	1.16927	9.23	158	0.0153
Case 3	0.08789	0.08450	1.16083	1.15789	1.18430	11.58	229	0.0595
Caes 4	0.08335	0.09245	1.11907	1.19046	1.12818	16.48	145	0.1161
Case 5	0.08631	0.09244	1.07034	1.12811	1.12738	18.79	106	0.1566

Table 8

The Identified Parameters applying GA to the Model in the Per-Unit System

(Using $l = 50$, $MG = 1000$ and $Y1$)

	R_s	R_r	L_m	L_s	L_r	$ME(\%)$	FG	F
RV	0.08775	0.08805	1.10553	1.18522	1.18522			
Case 1	0.08752	0.08843	1.14278	1.16525	1.17131	6.92	855	0.8745
Case 2	0.08864	0.08957	1.15838	1.21749	1.18023	10.67	738	0.8277
Case 3	0.08964	0.08672	1.05218	1.17783	1.16519	10.79	MG	0.7908
Caes 4	0.08701	0.09138	1.14430	1.18804	1.14320	11.92	MG	0.6173
Case 5	0.09142	0.08582	1.13087	1.17882	1.23181	13.473	MG	0.6177

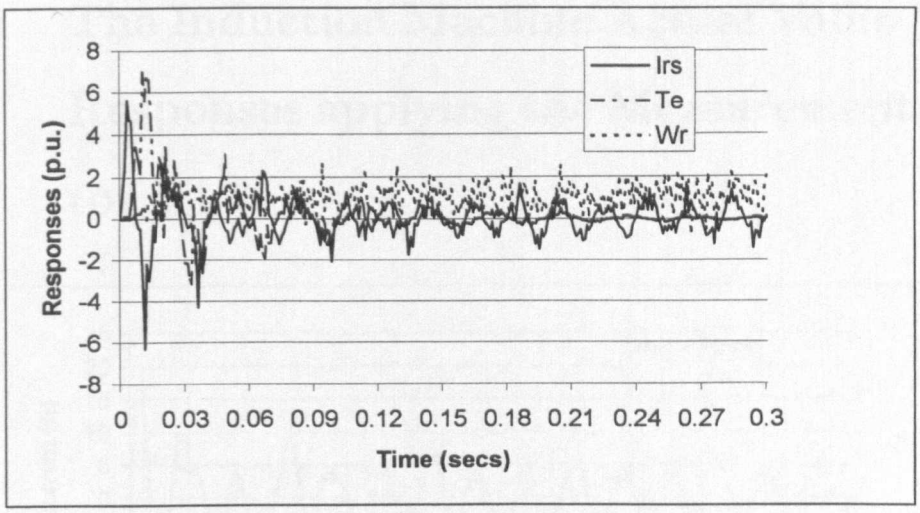

Figure 5.1: Induction machine output with noise $\sigma^2 = 0.2$

5.11 The Induction Machine Actual Value Model Responses applying the Measurement Vector $Y1$

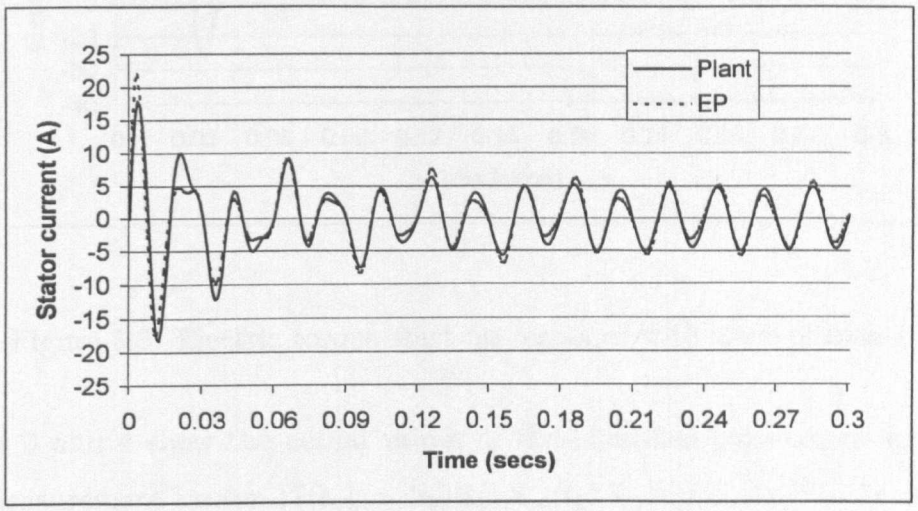

Figure 5.2: Stator current start-up response with noise of case 1

5.11.1 Simulation and Response Results

The start-up performance responses compare the plant with a simulation of an induction machine using parameters identified with either EP or GA under different variances of measurement noise in the Kron's two-axis dynamic mathematical model. The plant responses are generated by the Kron's two-axis dynamic mathematical model, without measurement noise, and using the machine's known parameters. It should be noted that the fitness, F, for EP is better the lower it is whilst for GA its fitness is better the larger it is.

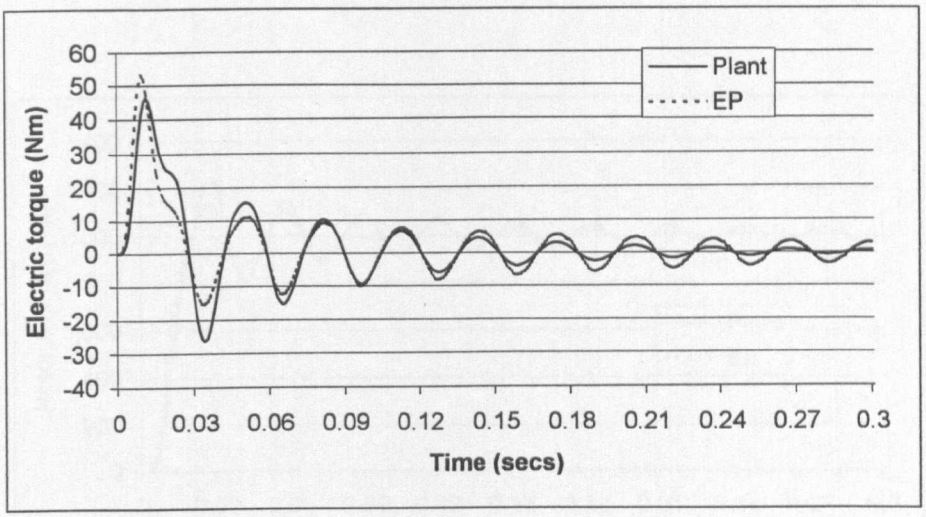

Figure 5.3: Electric torque start-up response with noise of case 1

Tables 3 and 4 show the actual values of the identified parameters applying the measurement vector $Y_1(t_k) = [i_{rs}(t_k)\ i_{ys}(t_k)\ i_{bs}(t_k)\ \omega_r(t_k)]^T$. As will be noticed from the Tables, both EP and GA follow similar numerical trends with respect to their maximum errors and fitnesses, although the convergence speed for EP is much quicker. The reponses produced for the actual valued case with the real parameters and the identified parameters obtained for case 1 and case 3 with the measurement vector $Y_1(t_k)$ are shown in Figures $5.2-5.7$ for EP and Figures $5.8-5.13$ for GA. What is made clear from these is that the responses produced by the induction machine with real parameters and the GA identified parameters show far fewer errors than those by the real parameters and the EP identified parameters.

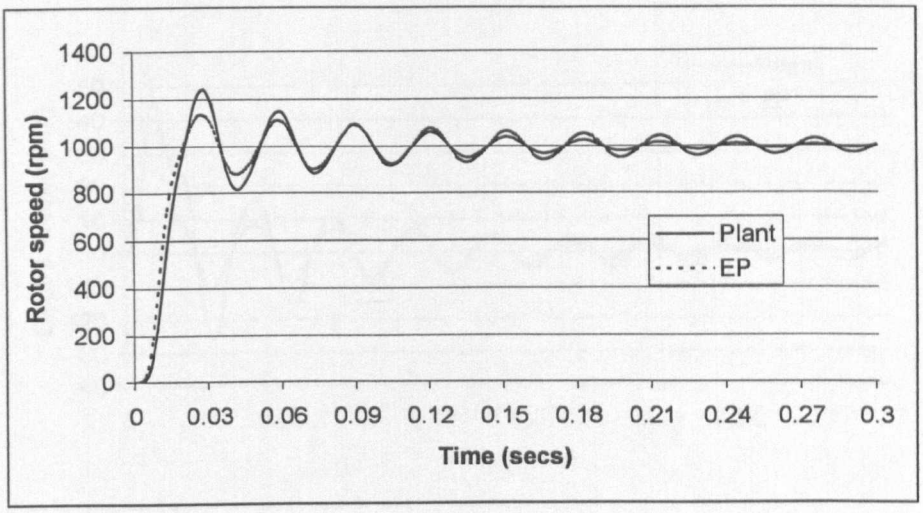

Figure 5.4: Rotor speed start-up response with noise of case 1

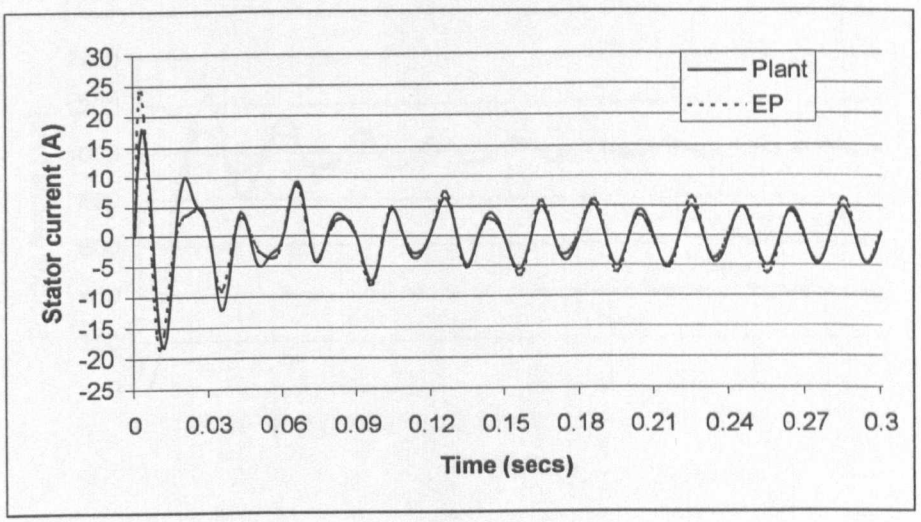

Figure 5.5: Stator current start-up response with noise of case 3

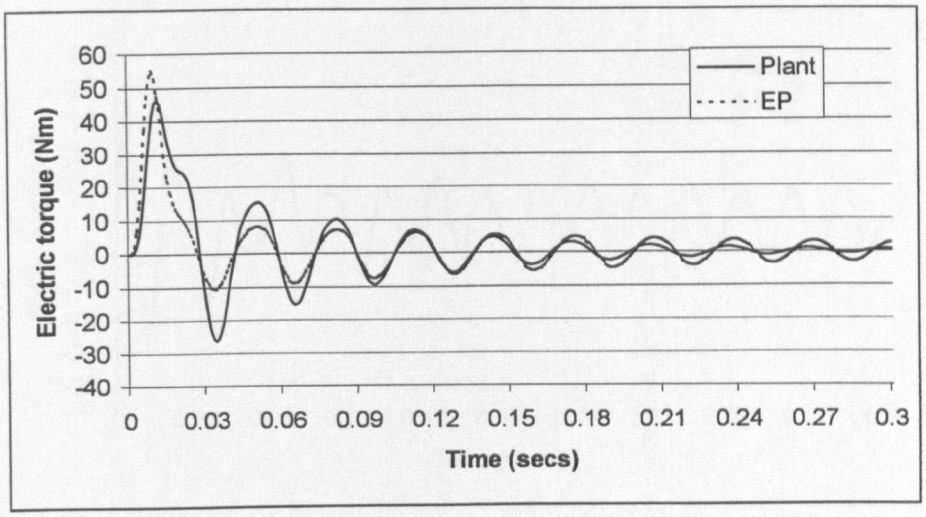

Figure 5.6: Electric torque start-up response with noise of case 3

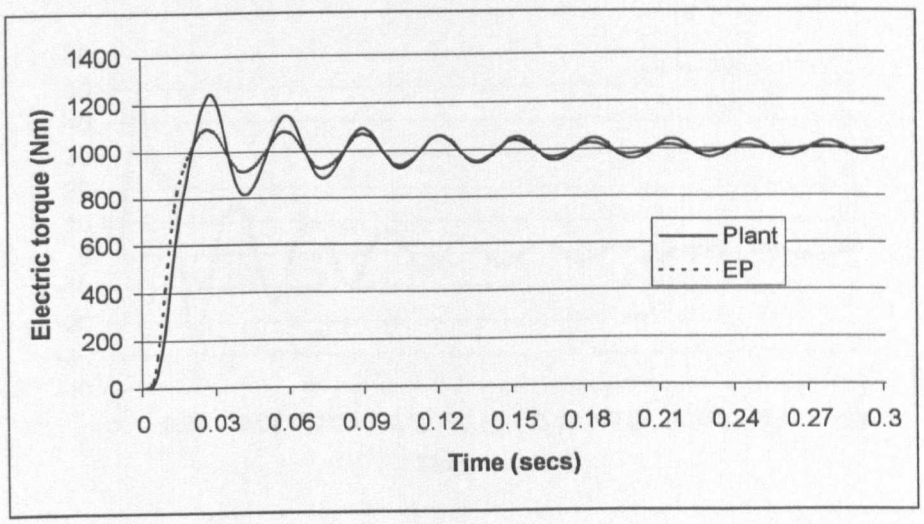

Figure 5.7: Rotor speed start-up response with noise of case 3

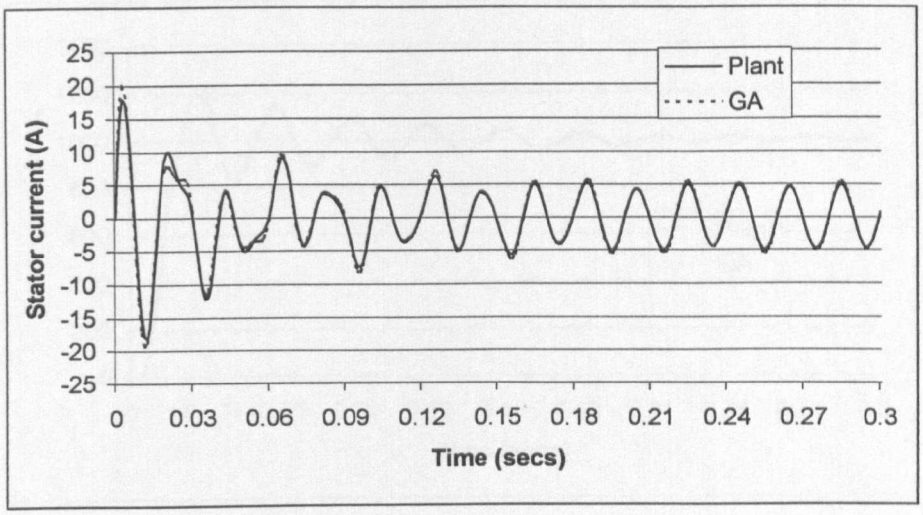

Figure 5.8: Stator current start-up response with noise of case 1

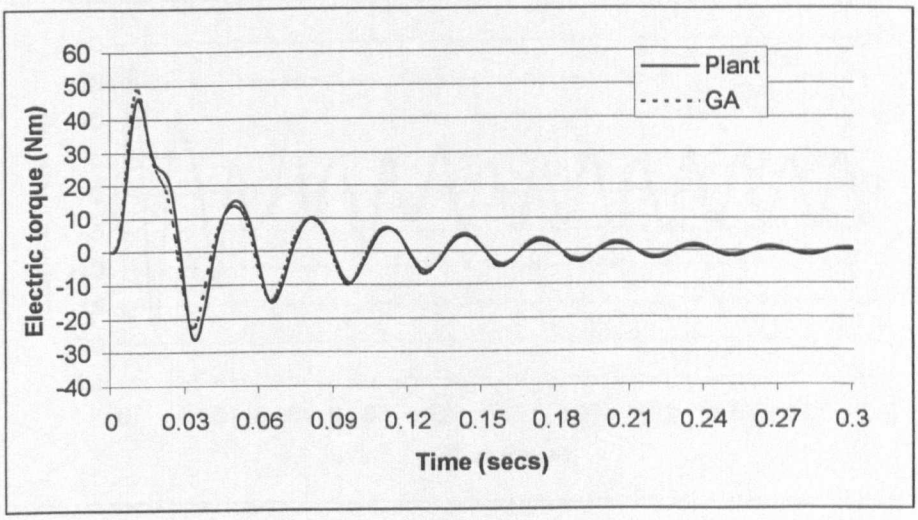

Figure 5.9: Electric torque start-up response with noise of case 1

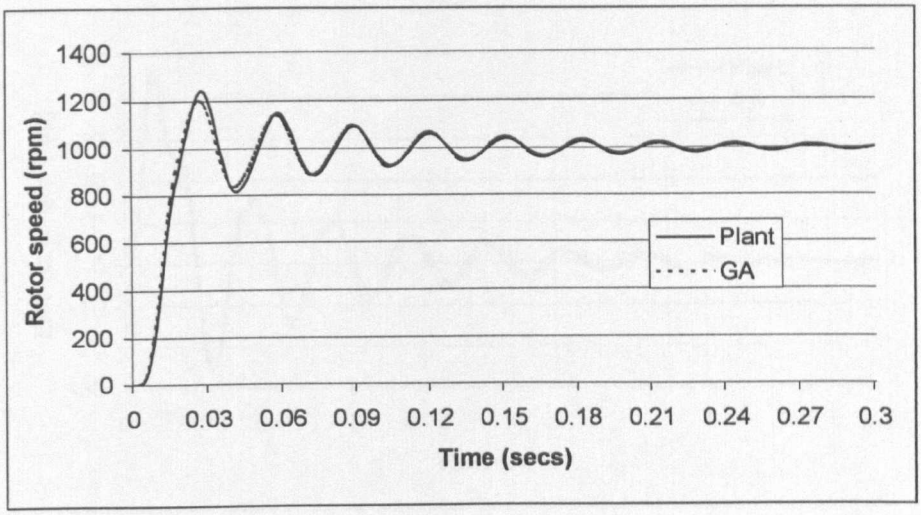

Figure 5.10: Rotor speed start-up response with noise of case 1

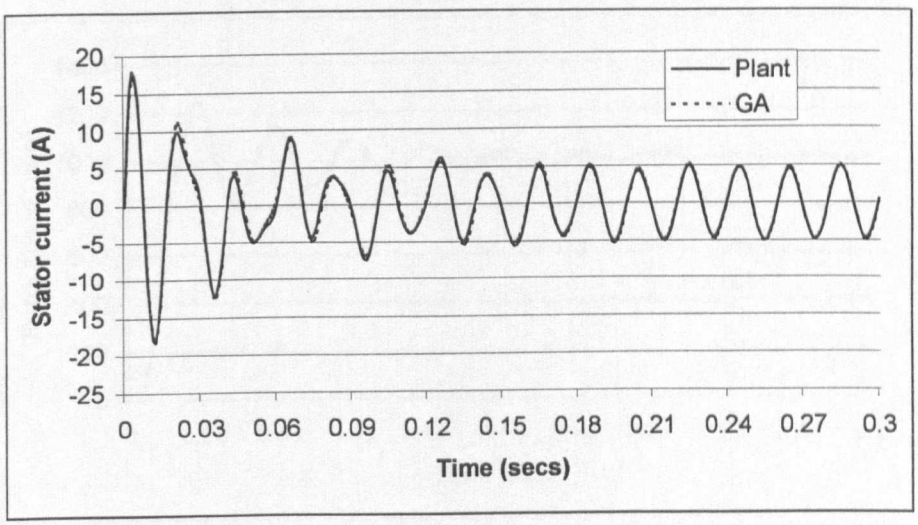

Figure 5.11: Stator current start-up response with noise of case 3

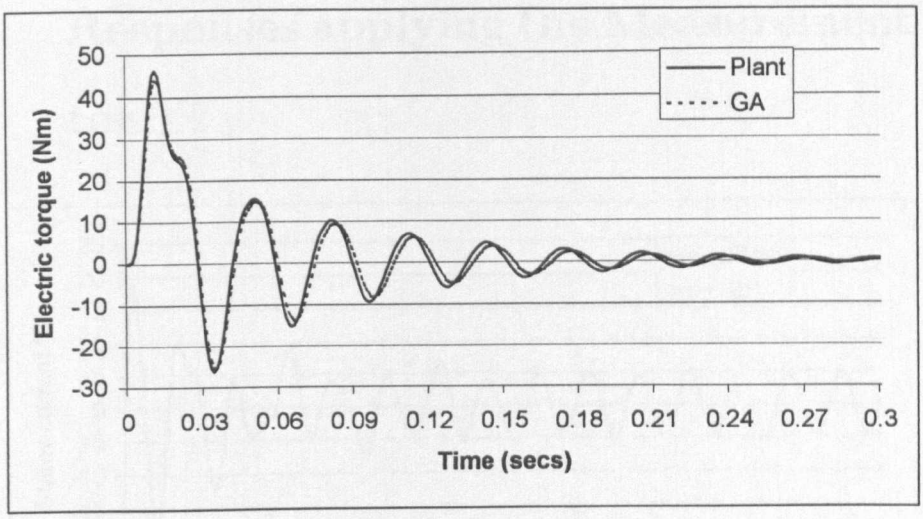

Figure 5.12: Electric torque start-up response with noise of case 3

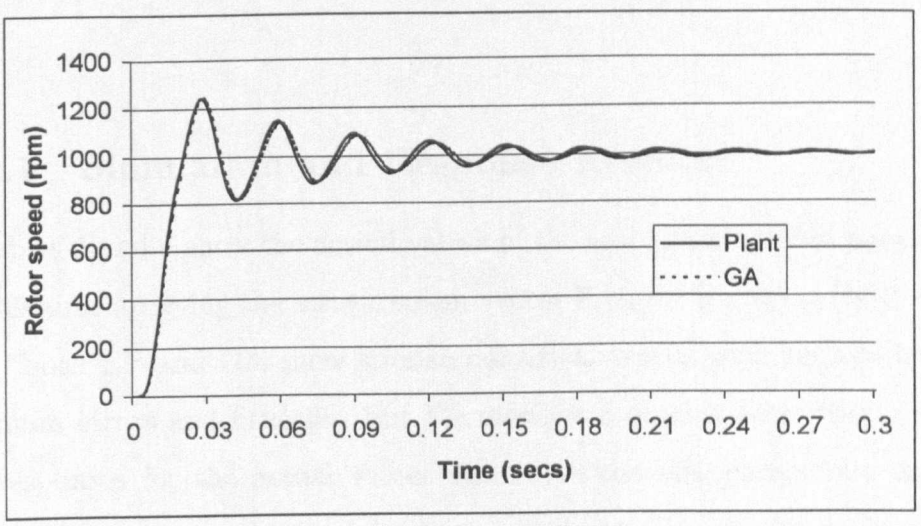

Figure 5.13: Rotor speed start-up response with noise of case 3

5.12 The Induction Machine Actual Value Model Responses applying the Measurement Vector $Y2$

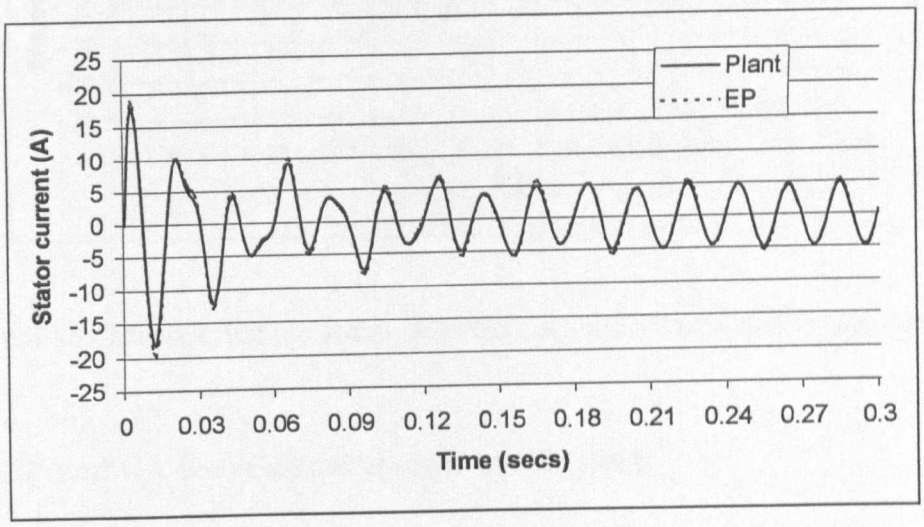

Figure 5.14: Stator current start-up response with measurement noise of case 1

5.12.1 Simulation and Response Results

Tables 5 and 6 show the actual values of the real and identified parameters but this time applying the measurement vector $Y_2(t_k) = [i_{rs}(t_k)\ \omega_r(t_k)]^T$. Once again, both EP and GA show similar numerical trends with respect to their maximum errors and fitnesses, but EP displays a quicker convergence speed. The responses for the actual valued case with the real parameters and the identified parameters obtained for case 1 and case 3 with the measurement vector $Y_2(t_k)$ are shown in Figures 5.14 – 5.19 for EP and Figures 5.20 – 5.25 for GA. In this comparison, the responses between the induction machine with the real parameters and the errors produced by the identified parameters of

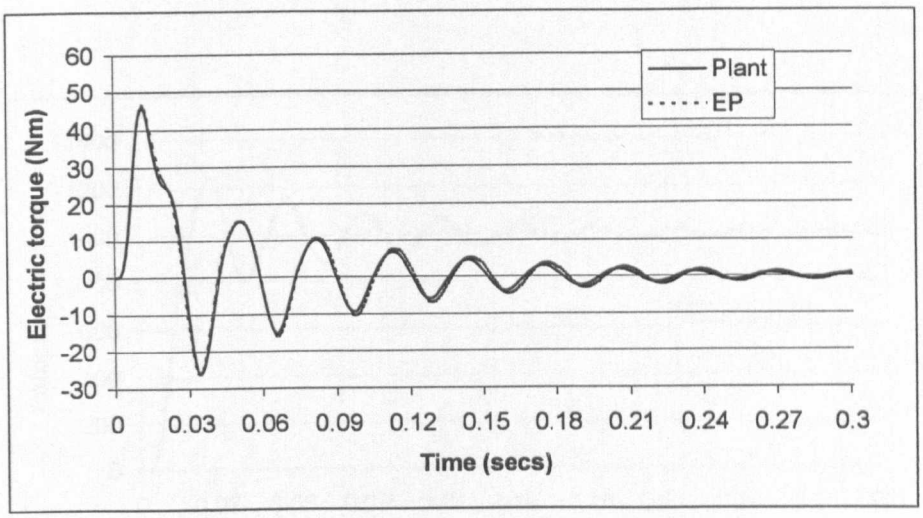

Figure 5.15: Electric torque start-up response with measurement noise of case 1

both EP and GA are of a similar order of magnitude.

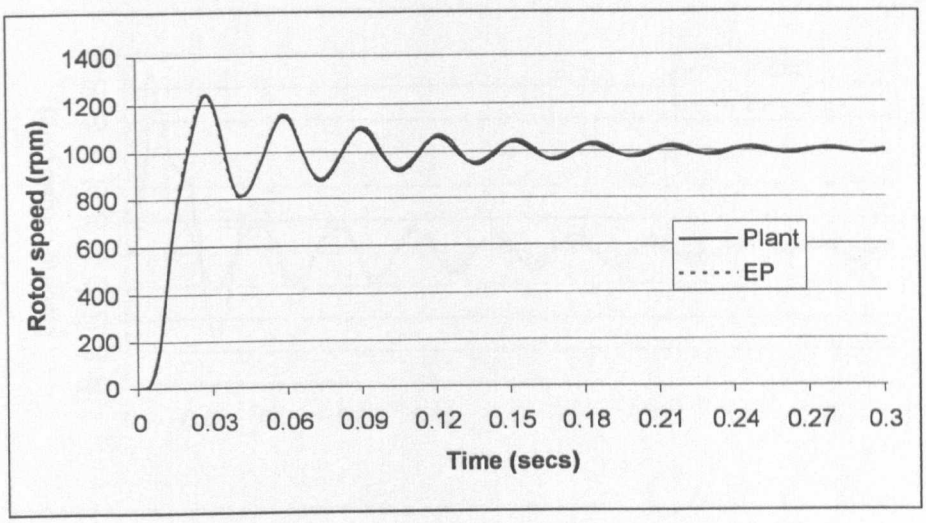

Figure 5.16: Rotor speed start-up response with measurement noise of case 1

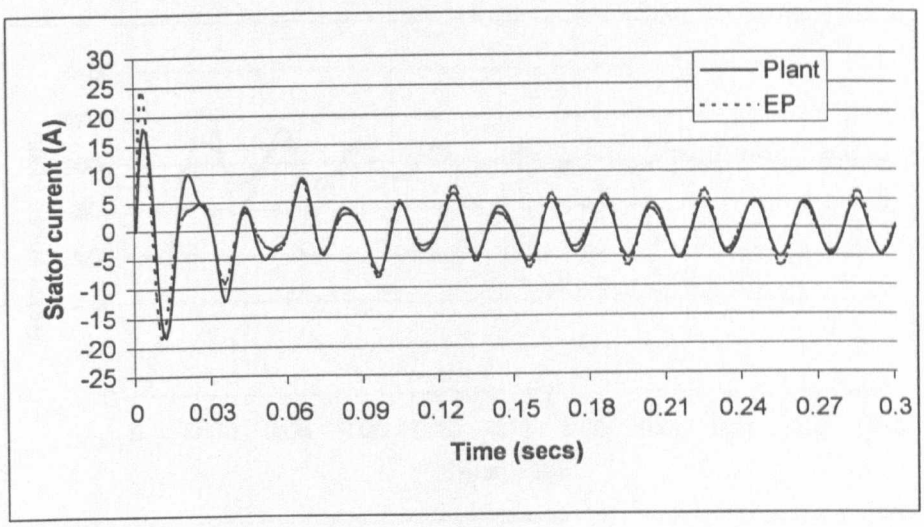

Figure 5.17: Stator current start-up response with measurement noise of case 3

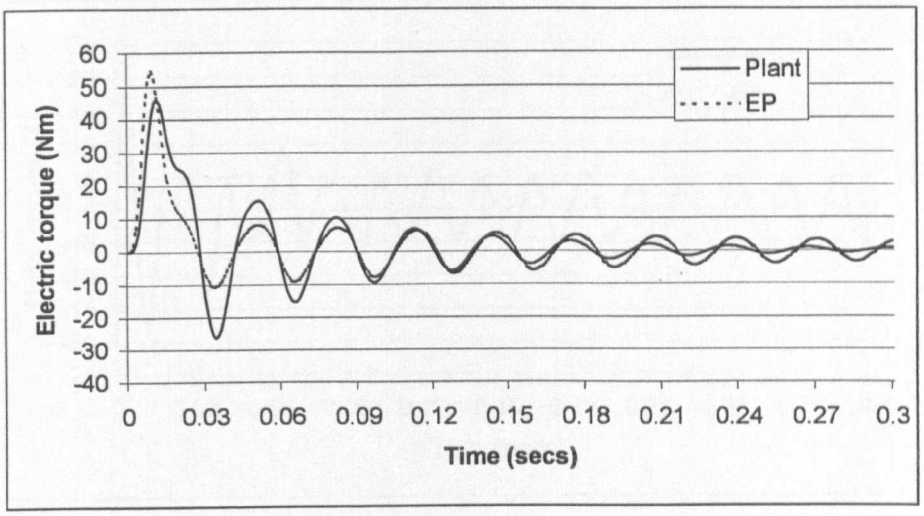

Figure 5.18: Electric torque start-up response with measurement noise of case 3

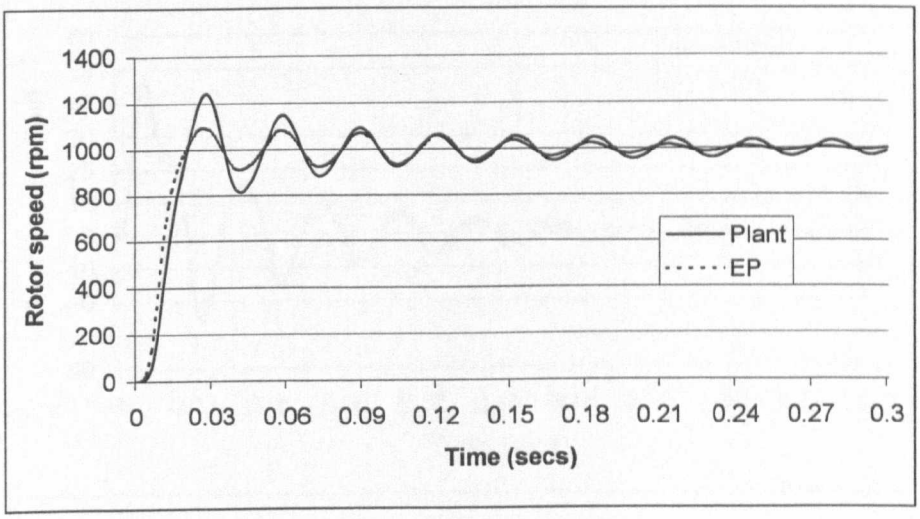

Figure 5.19: Rotor speed start-up response with measurement noise of case 3

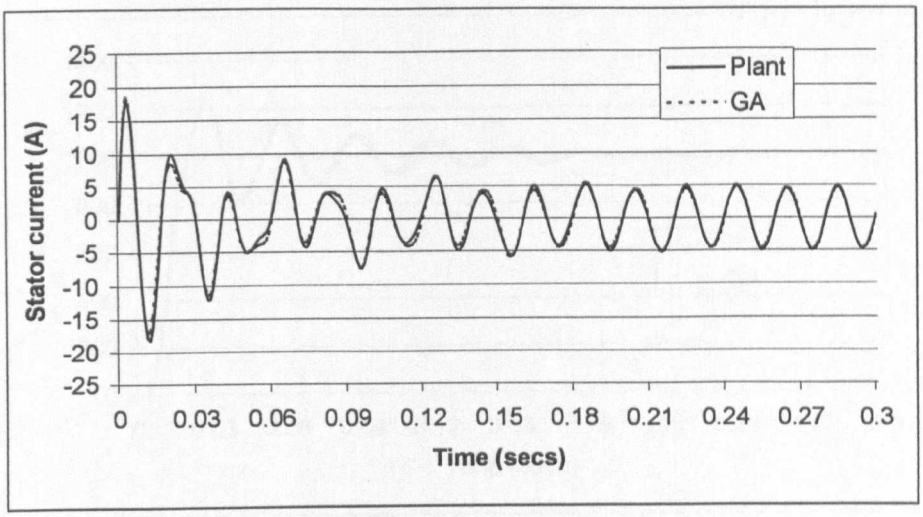

Figure 5.20: Stator current start-up response with measurement noise of case 1

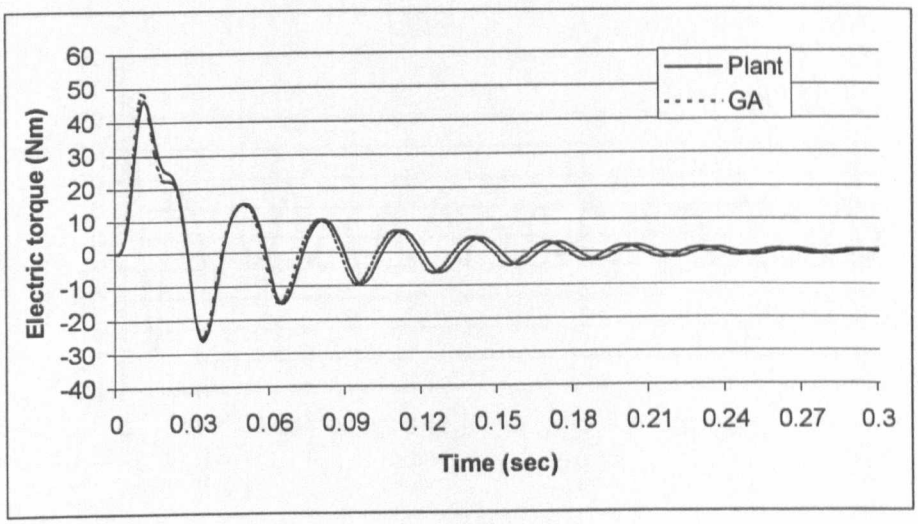

Figure 5.21: Electric torque start-up response with measurement noise of case 1

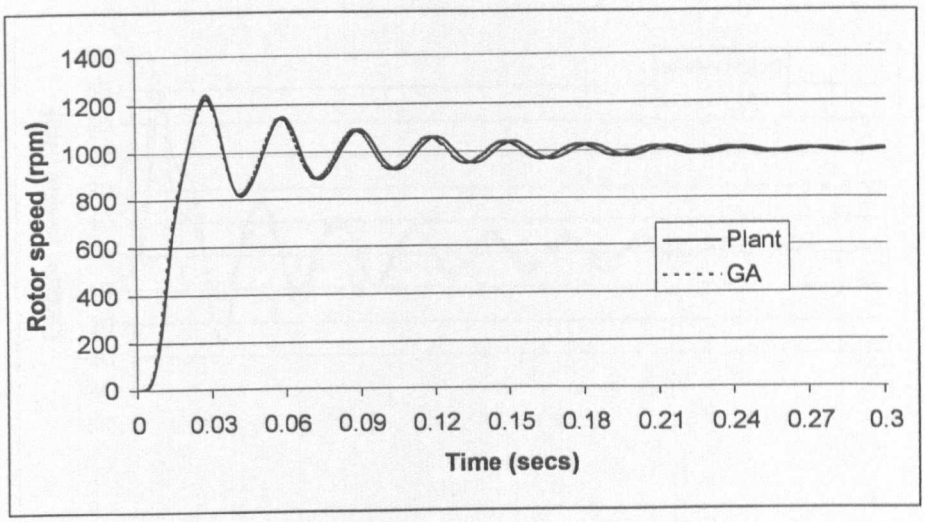

Figure 5.22: Rotor speed start-up response with Measurement noise of case 1

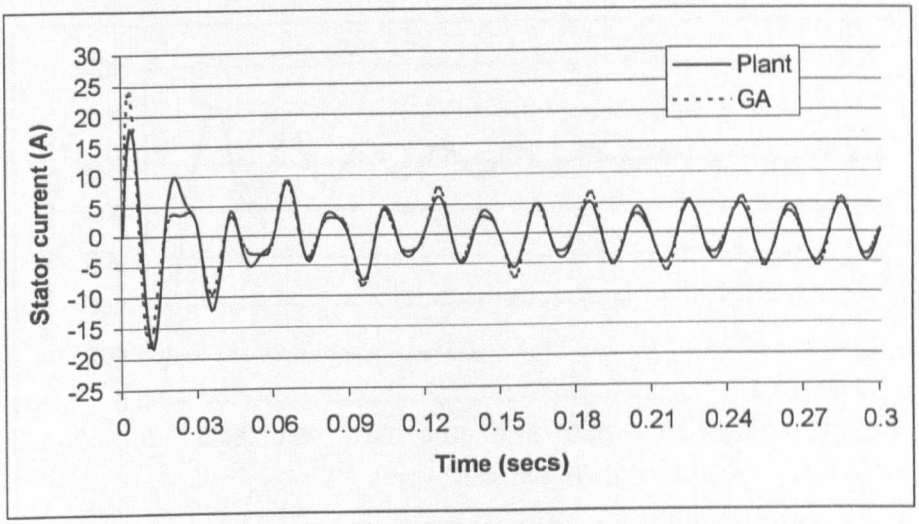

Figure 5.23: Stator current start-up response with measurement noise of case 3

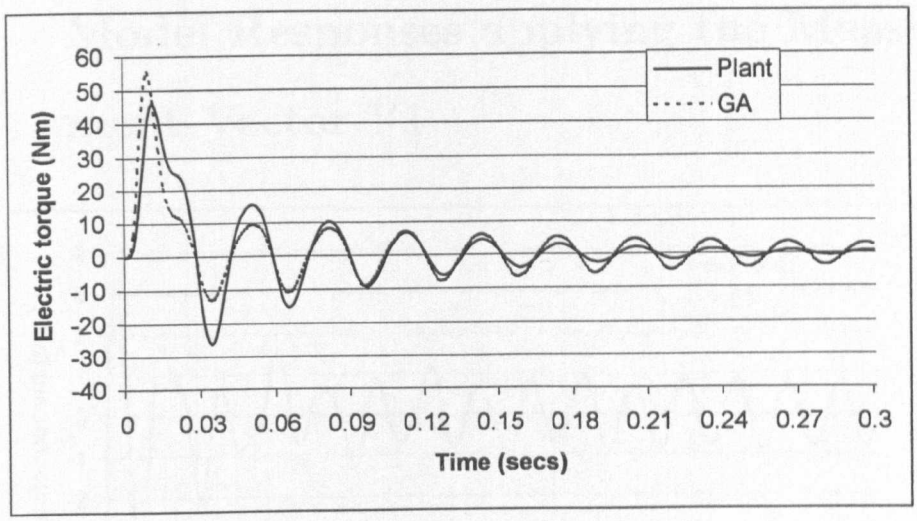

Figure 5.24: Electric torque start-up response with measurement noise of case 3

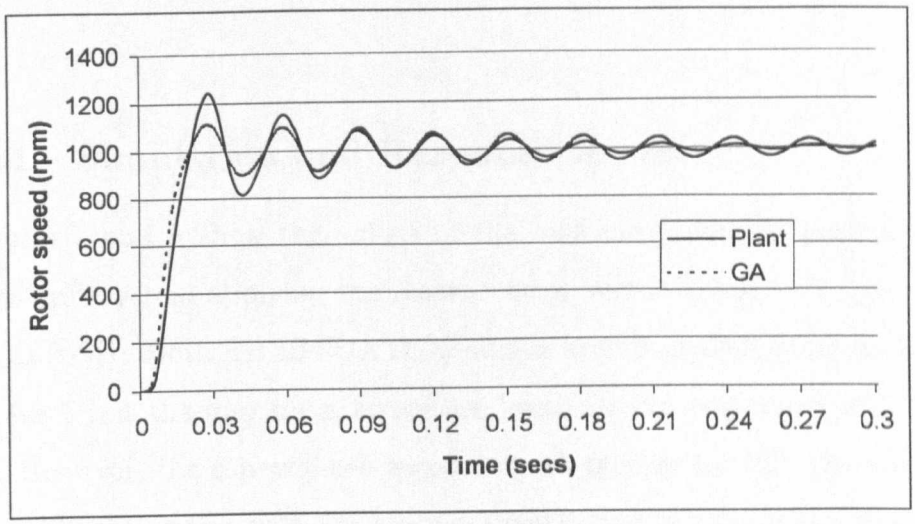

Figure 5.25: Rotor speed start-up response with measurement noise of case 3

5.13 The Induction Machine Per-Unit System Model Responses applying the Measurement Vector $Y1$

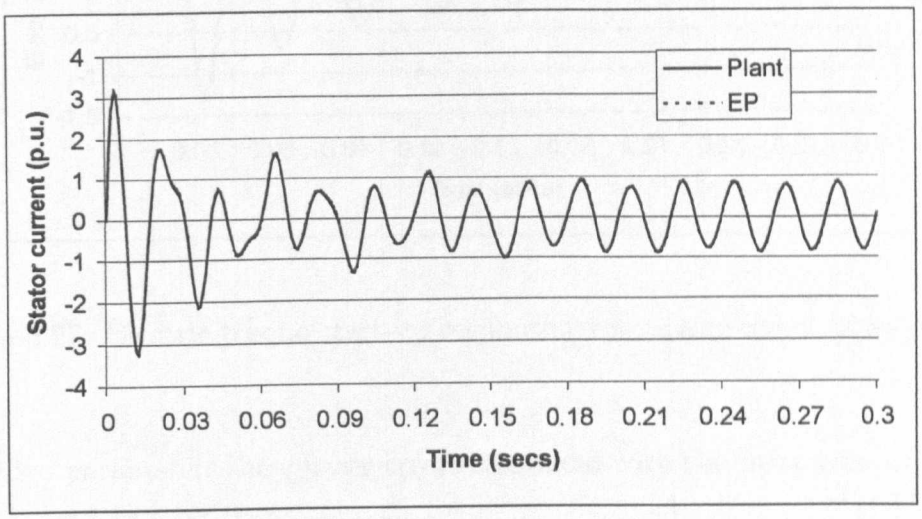

Figure 5.26: Stator current start-up response with measurement noise of case 1

5.13.1 Simulation and Response Results

Tables 7 and 8 show the values of the real and identified parameters in the per-unit system applying the measurement vector $Y_1(t_k) = [i_{rs}(t_k)\ i_{ys}(t_k)\ i_{bs}(t_k)\ \omega_r(t_k)]^T$. Both EP and GA show similar trends in their fitnesses for case 1 to case 5 but the maximum errors are lower for GA compared with thoses of EP. However, the convergence speed is much quicker for EP. The responses for the per-unit system with the real parameters and the identified parameters obtained for case 1 and case 3 with the measurement vector $Y_1(t_k)$ are shown in Figures 5.26 – 5.31 for EP and Figures 5.32 – 5.37 for GA exclusively. The responses produced by the induction machine with real parameters and the EP

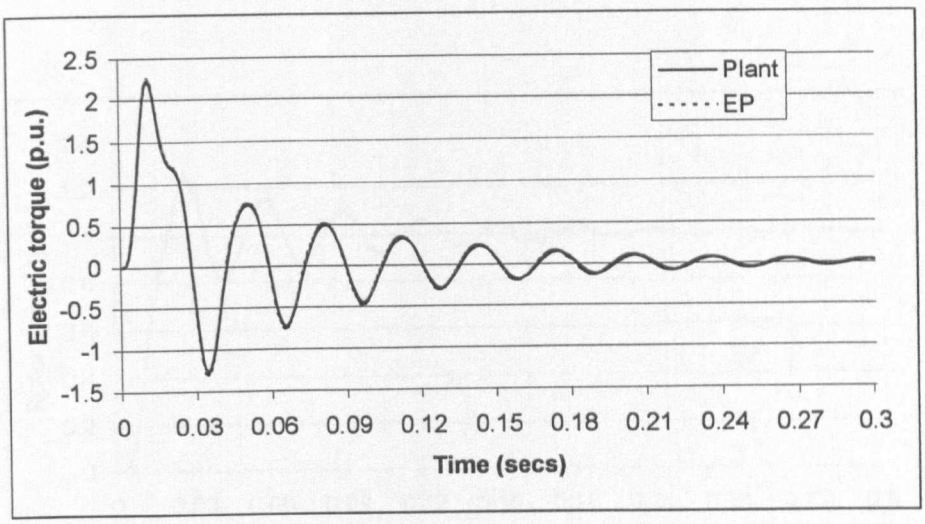

Figure 5.27: Electric torque start-up response with measurement noise of case 1

identified parameters show fewer errors compared with the responses produced with the GA identified parameters.

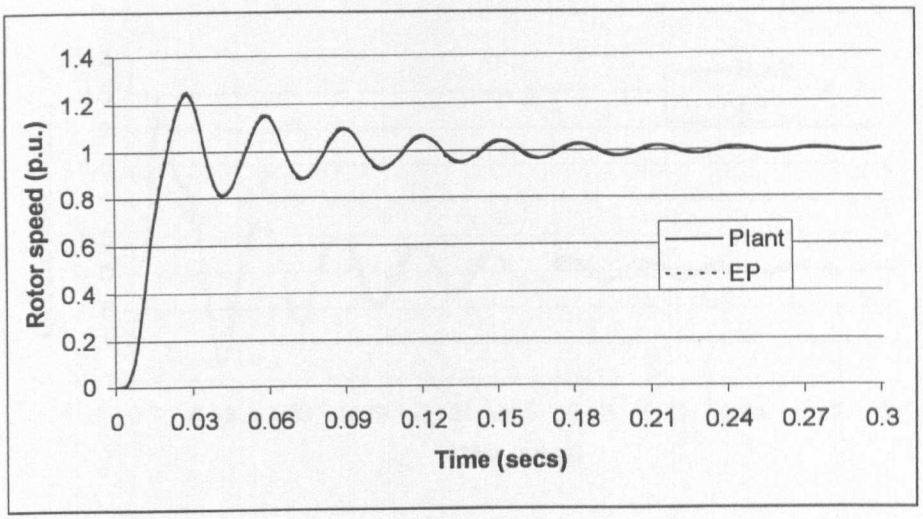

Figure 5.28: Rotor speed start-up response with measurement noise of case 1

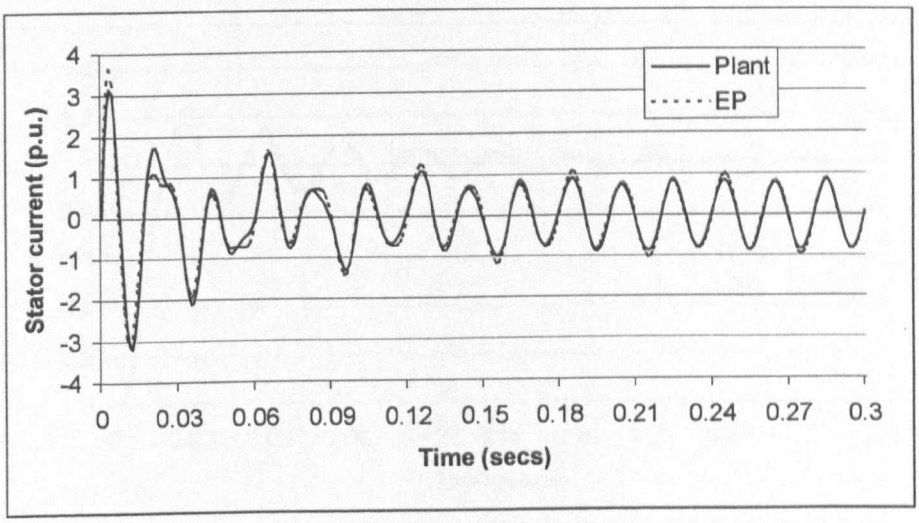

Figure 5.29: Stator current start-up response with measurement noise of case 3

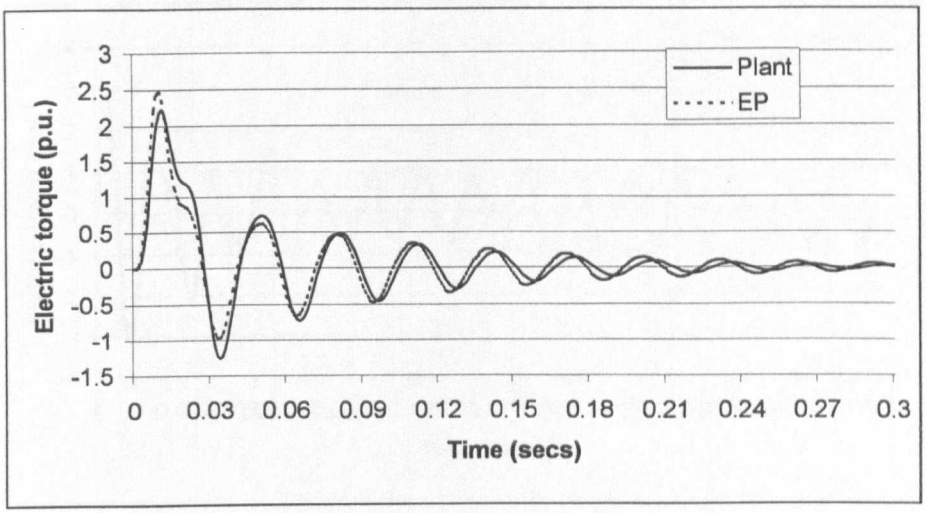

Figure 5.30: Electric torque start-up response with measurement noise of case 3

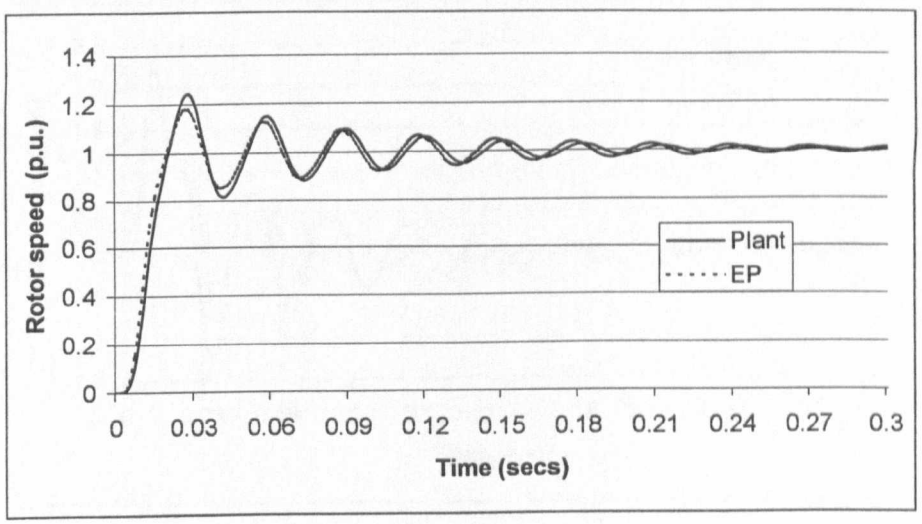

Figure 5.31: Rotor speed start-up response with measurement noise of case 3

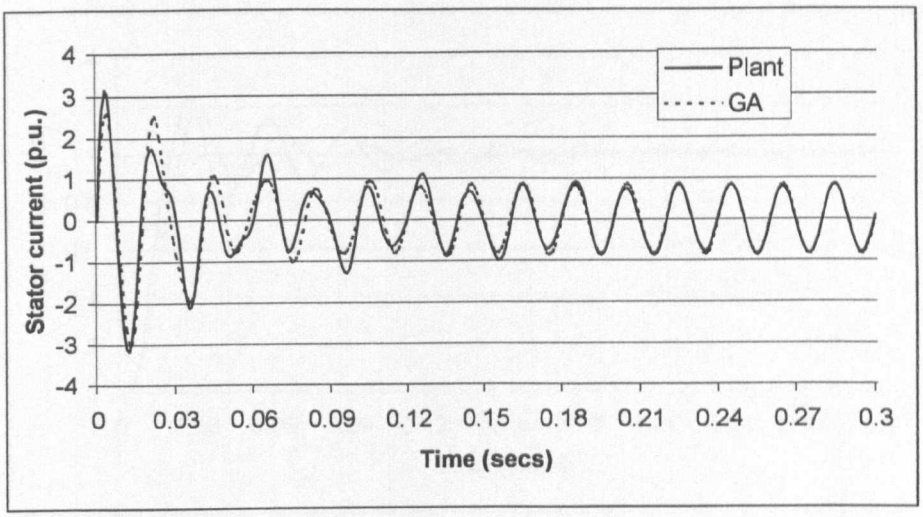

Figure 5.32: Stator current start-up response with measurement noise of case 1

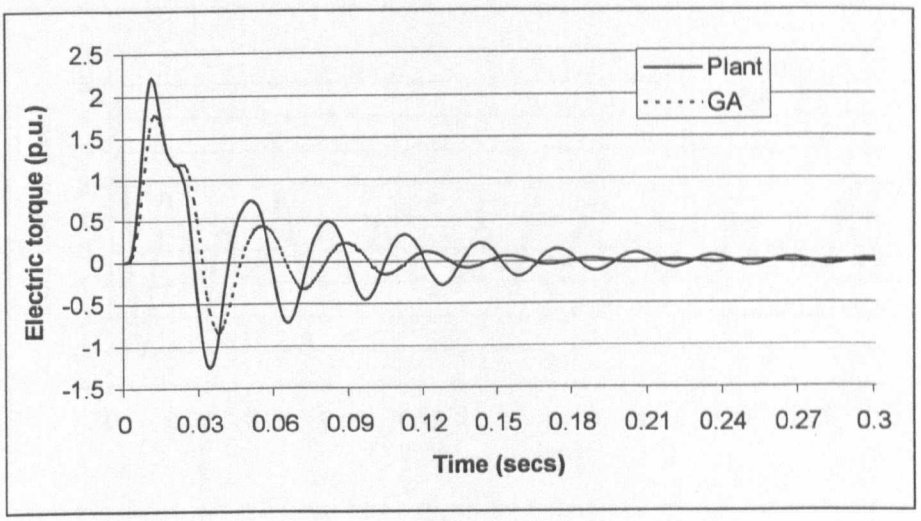

Figure 5.33: Electric torque start-up response with measurement noise of case 1

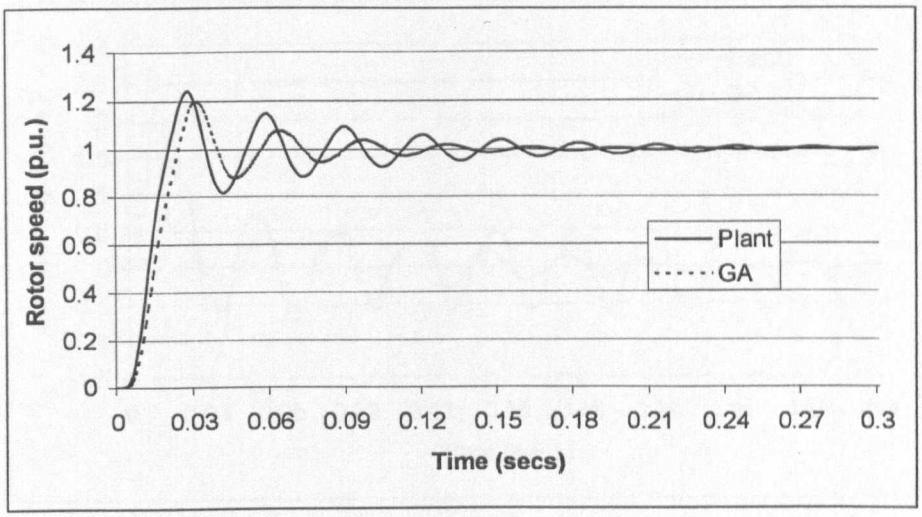

Figure 5.34: Rotor speed start-up response with measurement noise of case 1

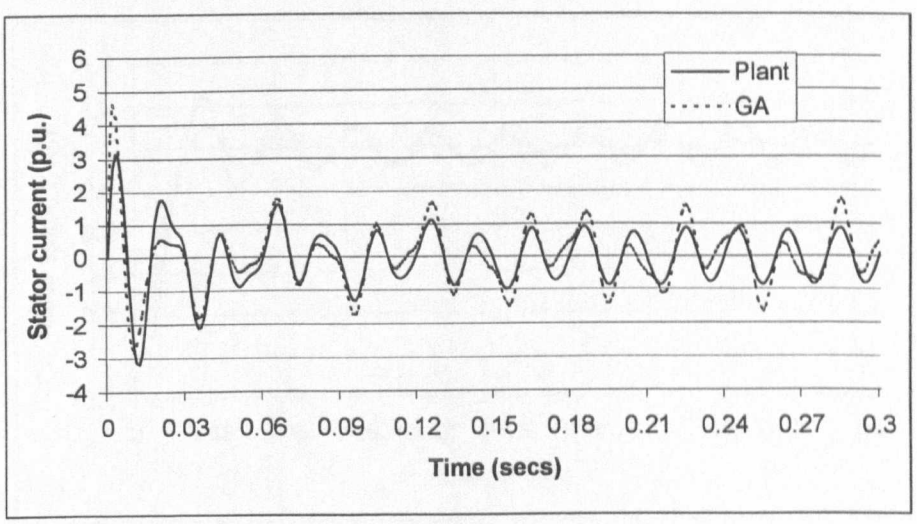

Figure 5.35: Stator current start-up response with measurement noise of case 3

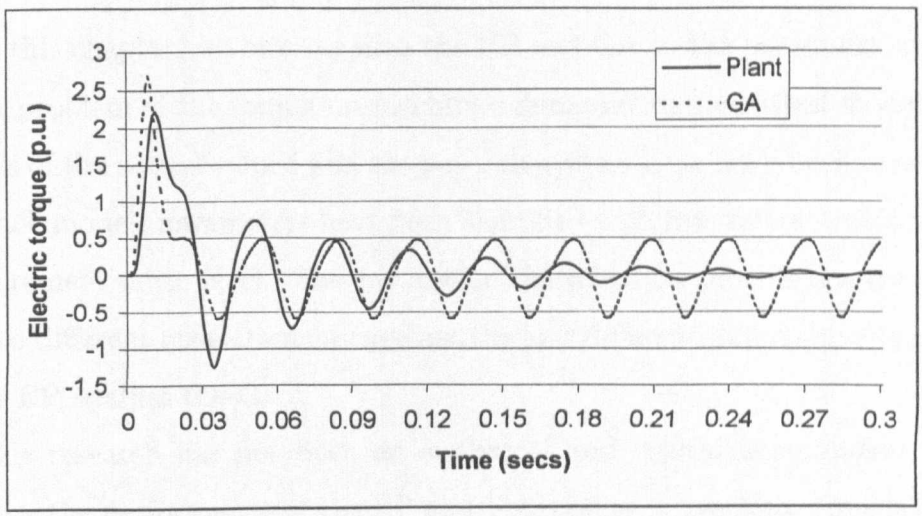

Figure 5.36: Electric torque start-up response with measurement noise of case 3

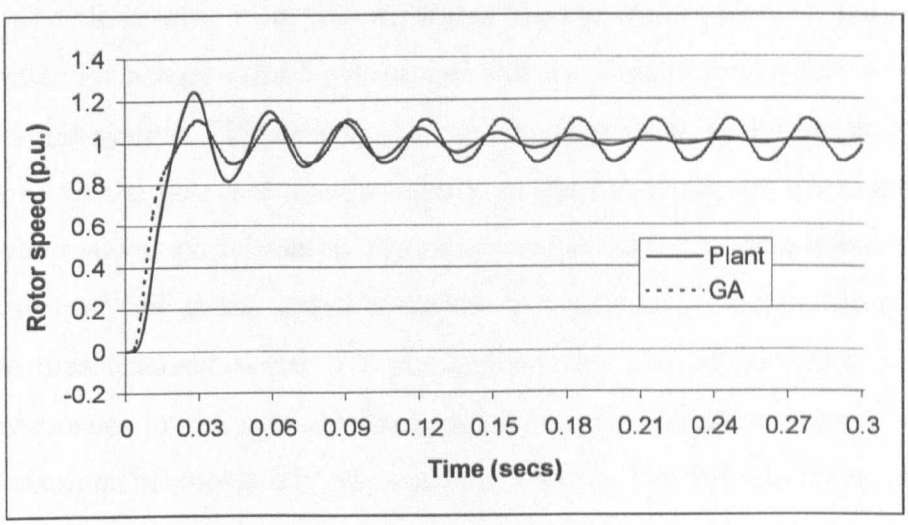

Figure 5.37: Rotor speed start-up response with measurement noise of case 3

5.14 Conclusion

In this chapter, we have applied the EP and GA to the parameter identification problem of the induction machine's dynamic mathematical model. The models of the actual valued and the per-unit system have both been presented, and each models parameters have been identified with five different variances of measurement noise level. The parameter identification process has compared the two different measurements against the four different measurements as well as the EP against the GA.

This research has not been an analytical and critical comparative study between the performance of the EP and GA, rather, a practical application of these SE stochastic processes so that their results can be judged with regard to the parameter identification process of a simulated 3-phase induction machine. Let us look at the results obtained over the different measurement noise variances. Clearly tabulated, both the EP and GA fitnesses follow numerical trends as expected, resulting from their individual searches in the search space, and their error measures regarding the optimum identified parameters are similar for actual valued parameters but are slightly better for the GA in the per-unit system. However, the rate of convergence is far quicker for the EP compared to the GA, and since a number of the GA responses are much more closer the real valued responses, one can ascertain that, for these cases, the EP has converged not at the global optimum, but rather at a local optimum.

The measurement vector $Y2$ was applied and the EP and GA provided good responses in the actual valued parameter identification process. When the measurement vector $Y1$ was applied, both of the SE processes resulted in optimum identified parameters providing good general responses. What is clearly visible from the simulation tabulated results is that increasing the level of measurement noise variance makes the location of optimum parameter values more difficult, a point that can be seen in the responses for the noisier

search spaces. This clearly implies that the more noisier the data used in the parameter identification process by the SE, a greater number of generations is needed for the optimum parameters to be located.

None of the SE methods presented has proved totally victorious over the other. Therefore, one is able to say that simulated evolution has provided good results in general, resulting in high efficiency and good practicality. The parameter identification of the induction machine's model in the actual valued or per-unit system using either the GA or EP is a powerful and robust technique which can be used in a real environment where there exists measurement noise and the number of measurements is limited. The following chapter will apply some of these principles to the parameter identification of a real induction machine where there will exist real measurement noise, and the number of measurements is limited, and, furthermore, the parameters of the real 3-phase induction machine are, unlike the simulation results just presented, unknown.

Chapter 6

Parameter Identification of a Real Induction Machine

6.1 Introduction

In this chapter we shall attempt to find the parameters of a real 3-phase induction machine. These parameters are the stator and rotor resistances, the stator and rotor inductances, the magnetising inductance, the viscous damping coefficient and the inertia moment. These parameters are unknown. The technical equipment used to measure the machine variables, namely the 3-phase stator currents, one stator phase voltage and the rotor shaft speed are described. The software used in the data acquisition is discussed as to are the results of the data acquisition and EP parameter identification process.

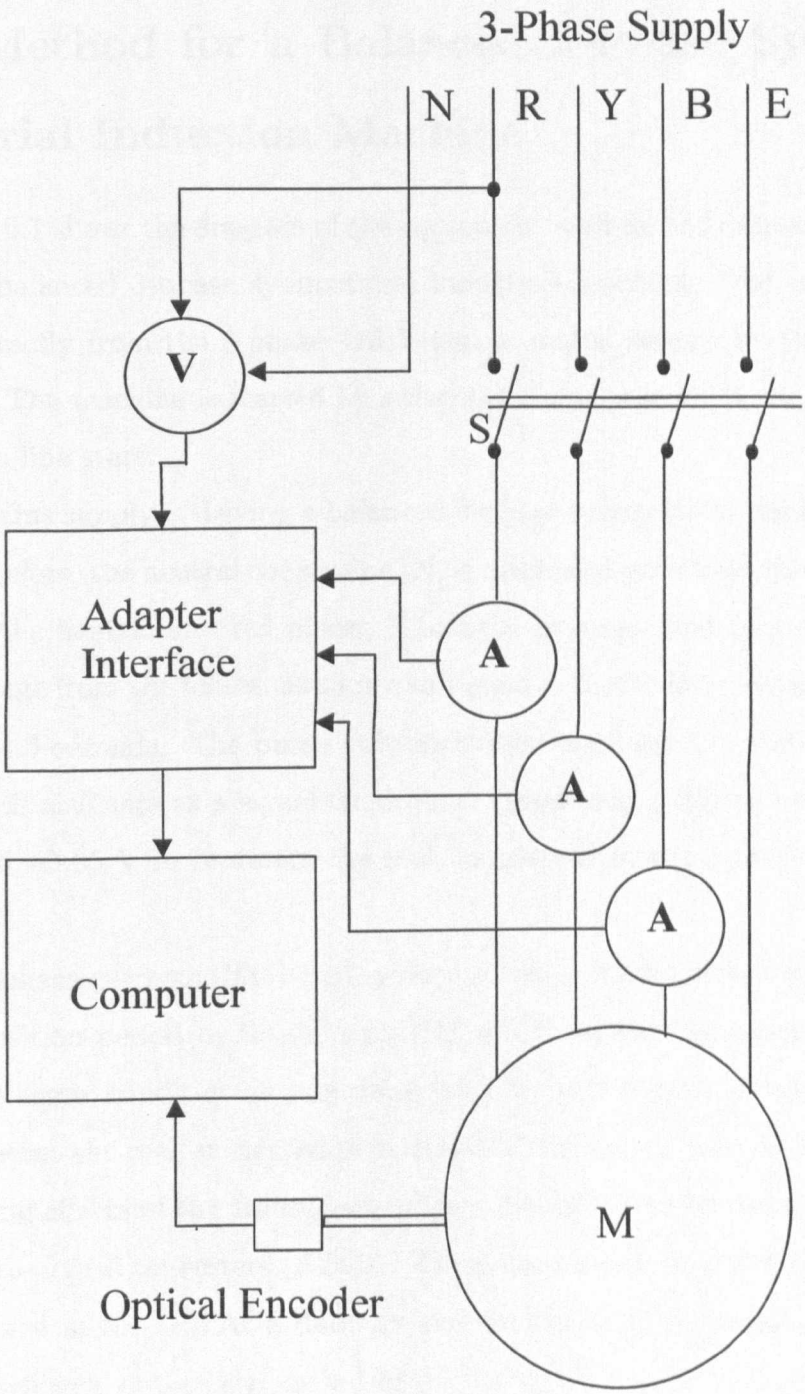

Figure 6.1: Data Acquisition fron a 3-Phase Induction Machine in the Laboratory

6.2 The Transient Start-Up Data Acquisition Method for a Balanced 3-Phase Symmetrial Induction Machine

Figure 6.1 shows the diagram of the equipment used for the data acquisition from the balanced 3-phase symmetrical induction machine. The machine is started directly from the 3-phase 415 V r.m.s. mains supply, by the contact switch, S. The machine is started by switching the contactor, S, manually for a direct-on-line start.

The mains supply is driving a balanced 3-phase symmetrical induction machine, therefore, the neutral connection, N, is dispensed with from the contacts. However, the neutral and red phase, R, connections are used to measure the phase voltage from the mains, as shown in Figure 6.1, over the data acquisition period of 4.5 seconds. The phase voltage is measured using a voltage transducer which also acts as a signal conditioner, producing peak-to-peak output voltages of ± 5.65 V to represent the real, measured peak-to-peak voltages of ± 339 V.

The 3-phase currents, RYB (red, yellow, blue), are also measured over the data acquisition period by 3 solid core Hall effect current transducers. These also act as signal conditioners producing peak-to-peak output voltages of ± 10 V to represent the real, measured peak-to-peak currents of ± 40 A. The signal conditioning effects of the transducers allows the signals to be suitable for the analogue-to-digital converters (ADCs). These have input ranges of ± 10 V and are contained in the DSPACE data acquisition board. The sampling rate for these transducers' output signals is 1 kHz. The ADCs have a 16-bit resolution.

The induction machine was ran under no-load conditions. Its rotor shaft speed was measured using an optical encoder, which generated 360 pulses per revolution. Figure 6.1 shows the leads from the current and voltage transducers

joined to the adapter interface, from which their corresponding leads feed into the input ports of the data acquisition card. The optical incremental encoder provides a digital signal for the data acquisition board and, therefore, uses a seperate channel for this purpose.

The software used with the data acquisition system was MATLAB. The software contained a voltage data coefficient of 60 and a current data coefficient of 4 which multiplied their respective conditioned signals back to their true values for the output data file. The resulting data file from the PC contained the required measurement vector, having the two induction machine state variables, as discussed in Chapter 5, Section 9:

$$Y_2(t_k) = [i_{rs}(t_k)\omega_r(t_k)]^T \tag{6.2.1}$$

where t_k is the measuring time instant (1 ms), corresponding to the k_{th} measurement of i_{rs}, the red-phase stator current and ω_r, the rotor shaft speed. This output data file, the measurement vector, was used together with the output data file of the two same state variables generated by the Kron's two-axis dynamic mathematical model, over the same time instances, to form the error function for the EP. This is part of the parameter identification process which will be discussed in greater detail later in this Chapter.

6.3 The Rotor Shaft Speed Calculations

The rotor shaft speed is measured and calculated by use of an optical incremental shaft encoder. The optical incremental shaft encoder produces 360 serial voltage pulses per shaft revolution. The serial pulse chain edge triggers an asynchronous counter in the data acquisition card. The count value is sampled from the asynchronous counter every millisecond and stored in successive memory locations. Because the asynchronous counter count value samples, stored in successive data memory locations, are count value summations, one

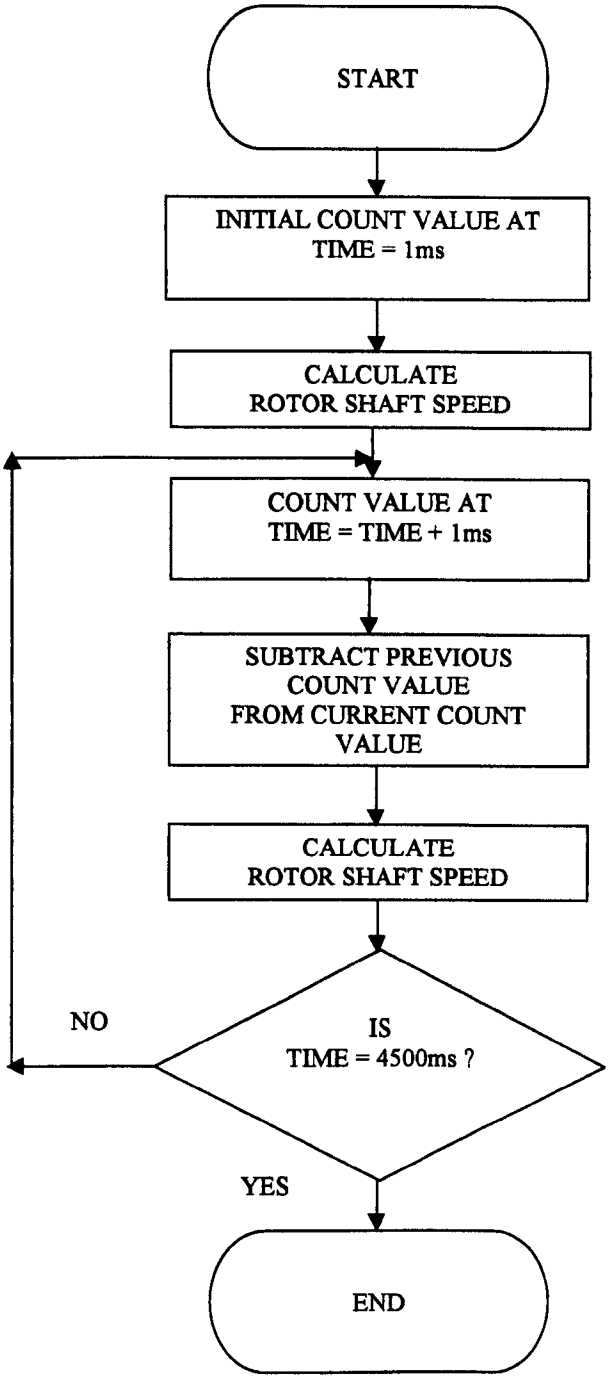

Figure 6.2: Flowchart for the Rotor Shaft Speed Algorithm

must subtract the previous count value, $count_{n-1}$, from the current count value, $count_n$, so that one is able to calculate the rotor shaft speed (rad/sec) over this 1 millisecond time interval. This process is repeated over the data acquisition period. The process used to achieve this is has follows. The initial rotor shaft speed, $speed_1$, at the first millisecond time interval is:

$$speed_1 = 2\pi(count_1)/360 \tag{6.3.1}$$

The speed, $speed_2$, at the second millisecond interval is:

$$speed_2 = 2\pi(count_2 - count_1)/360 \tag{6.3.2}$$

And the speed, $speed_n$, at the n_{th} millisecond interval is:

$$speed_n = 2\pi(count_n - count_{n-1})/360 \tag{6.3.3}$$

This process is contained in the software to produce a data file which calculates the rotor shaft speed over the data acquisition period in steps of 1 ms. The flowchart for the rotor shaft speed algorithm is shown in Figure 6.2.

6.4 The Real Induction Machine's Specifications

The real 3-phase induction machine that was used in the data acquisition investigation to identify its parameters has the following specifications presented in Table 9.

Table 9

Specifications of the Induction Machine

Manufacturer	MarelliMotori spa
Induction Machine Type	MA 100 LA 4
Rated Output Power	2200 W
Voltage	240/415 V
Current	8.8/5.1 A
Frequency	50 Hz
Rotor Shaft Speed	1420 rpm
Number of Poles	4
Power Factor $\cos\varphi$	0.8
Rated Torque	14.8 Nm

6.5 Identification of a Real 3-Phase Induction Machine's Parameters Applying Evolutionary Programming

The parameter identification process applying EP is repeated here, albeit somewhat abridged from Chapter 5, in order to clarify the main principles involved in the parameter identification during the simulation cases apply equally well to the parameter identification processes for the real induction machine.

The induction machine parameter identification may be written as follows:

$$DX = f(\theta, X, V) \tag{6.5.1}$$

where

X = the state variable vector

V = the stimulus, i.e., the input vector from the supply

θ = the parameter variable vector to be identified

D = the differential operator $\frac{d}{dt}$

The EP algorithm follows the same route as such:

1. An initial population of parameters, $\hat{p}^{(0)} = \{\hat{p}_i^{(0)} \mid i = 1, 2, \ldots, l\}$, is formed with randomly selected individuals. Each individual parameter vector is constrained with the following conditions:

$$p_{min,j} \leq \hat{p}_{i,j}^{(g)} \leq p_{max,j} \qquad j = 1, 2, \ldots, m \tag{6.5.2}$$

where $p_{min,j}$ and $p_{max,j}$ are the limits of the jth element of the parameter vector, $\hat{p}_i^{(g)}$, estimated without *a prior* knowledge. $\hat{p}_{i,j}^{(g)}$ denotes the estimation of the jth element of the ith individual in the gth generation of the parameter. The process starts at $g := 0$.

2. Each individual, $\hat{p}_i^{(g)}$, is used to calculate the state variable $\hat{X}(t_k)$ and the general performance $\hat{Y}_2(t_k)$ as follows:

$$DX(t_k) = f(\hat{p}_i^{(g)}, X(t_k), V(t_k)) \tag{6.5.3}$$

$$Y_2(t_k) = [i_{rs}(t_k)\ \omega_r(t_k)]^T \tag{6.5.4}$$

$$C_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.5.5}$$

$$\hat{Y}_2(t_k) = C_2 X(t_k) \tag{6.5.6}$$

and the error vector

$$E(\hat{p}_i^{(g)}, t_k) = Y_2(t_k) - \hat{Y}_2(t_k) \tag{6.5.7}$$

where $Y_2(t_k)$ stands for the measured performance vector taken from the real induction machine, $\hat{Y}_2(t_k)$ the simulated vector generated by the Kron's two-axis dynamic mathematical model and E the expectation operator. Then we get the error function of $\hat{p}_i^{(g)}$:

$$h(\hat{p}_i^{(g)}) = \sum_{t_k=0}^{T_N} E(\hat{p}_i^{(g)}, t_k)^T \Lambda E(\hat{p}_i^{(g)}, t_k) \tag{6.5.8}$$

where Λ is a unit matrix, t_k is the discrete computing time which is the same as the measurement sampling time. This leads to the fitness of $\hat{p}_i^{(g)}$, where the *Fitness* is a function of $h(\hat{p}_i^{(g)})$. The aim of the evolutionary programming, therefore, is to minimize the error function or to minimize the fitness.

3. A Gaussian random variable, with zero mean and its variance proportional to the fitness of $\hat{p}_i^{(m)}$ scaled by β_j, is added to each element of $\hat{p}_i^{(g)}$, $i = 1, 2, \ldots, l$, to produce new individuals $\hat{p}_{i+l}^{(g)}$. β is a factor in association with the jth identified parameter in the parameter vector. $\hat{p}_{i+l}^{(g)}$ has to satisfy the conditions laid down by expression (6.5.2).

4. The new individuals, $\hat{p}_{i+l}^{(g)}$, where $i+l = l+1, \ldots, 2l$, are used to calculate the general performance index as their predecessors to get their fitness, $h(\hat{p}_{i+l}^{(g)})$.

5. Each individual, $\hat{p}_i^{(g)}$, where $i \in \{1, \ldots, 2l\}$, competes with some others according to the competition rule. A rank of competition is set up.

6. A new generation, $\hat{p}^{(g+1)}$, with the same individual number of $\hat{p}^{(g)}$, is formed based on the first l ranked $\hat{p}_i^{(g)}$, $i \in \{1, 2, \ldots, 2l\}$.

7. The process will stop if $g = MG$, where MG is a big integer representing the maximum generation number, which has been initially set. If this criterion is not satisfied, $g := g+1$, and the process is repeated from step 2.

6.6 Implementation of EP for the Parameter Identification of a Real Induction Machine

6.6.1 The Initial Stator Current and Voltage Phase Angle Determination

As described in the previous Section, the EP is used for the parameter identification of a real induction machine, and we described the error vector as

$$E(\hat{p}_i^{(g)}, t_k) = Y_2(t_k) - \hat{Y}_2(t_k) \tag{6.6.1}$$

and, has stated, $Y_2(t_k)$ is the measured performance vector taken from the real induction machine, and $\hat{Y}_2(t_k)$ is the simulated vector generated by the Kron's two-axis dynamic mathematical model. The Kron's two-axis dynamic mathematical model, used in this research, operates on the assumption that on the instance the machine is switched on, $t_k = 0$, the red-phase stator voltage, a cosine function used in this research, is at its maximum value and the corresponding initial red-phase current will be zero. However, due to the direct-on-line start to the main's supply by the induction machine, the red-phase stator voltage was not at its maximum cosine value when the machine was switched on and the red-phase stator current was zero. The angle between the point where the machine was switched on, at $t_k = 0$, when the red-phase stator current was zero, to the point where the corresponding red-phase stator voltage, $v_{rs} = \sqrt{2}V_1 \cos(\omega_1 t)$, was at its cosine maximum value, $v_{rs} = \sqrt{2}V_1$, is defined as the point on wave switching angle. This angle was determined so that it could be accounted for in the induction machine's state equations of the Kron two-axis dynamic mathematical model has described in Chapter 5, Section 2 and are as follows

$$Di_{ds} = \sigma(-R_s L_r^{'} i_{ds} + L_m^2 i_{qs}\omega_r + L_m R_r^{'} i_{dr}^{'} + L_m L_r^{'} i_{qr}^{'}\omega_r) + \sigma L_r^{'} v_{ds} \quad (6.6.2)$$

$$Di_{qs} = \sigma(-L_m^2 i_{ds}\omega_r - R_s L_r^{'} i_{qs} - L_m L_r^{'} i_{dr}^{'}\omega_r + L_m R_r^{'} i_{qr}^{'}) + \sigma L_r^{'} v_{qs} \quad (6.6.3)$$

$$Di_{dr}^{'} = \sigma(R_s L_m i_{ds} - L_s L_m i_{qs}\omega_r - L_s R_r^{'} i_{dr}^{'} + L_s L_r^{'} i_{qr}^{'}\omega_r) - \sigma L_m v_{ds} \quad (6.6.4)$$

$$Di_{qr}^{'} = \sigma(L_s L_m i_{ds}\omega_r + R_s L_m i_{qs} + L_s L_r^{'} i_{dr}^{'}\omega_r - L_s R_r^{'} i_{qr}^{'}) - \sigma L_m v_{qs} \quad (6.6.5)$$

$$D\omega_r = \frac{Q}{J}[QL_m(-i_{ds} i_{qr}^{'} + i_{qs} i_{dr}^{'}) + T_r - \frac{B\omega_r}{Q}] \quad (6.6.6)$$

where

$$\sigma = \frac{1}{L_s L_r - L_m^2} \quad (6.6.7)$$

The point on wave switching angle was calculated from the data collected from the data acquisition system used with the induction machine, by direct-on-line starting the machine over the data acquisition period of 4.5 seconds.

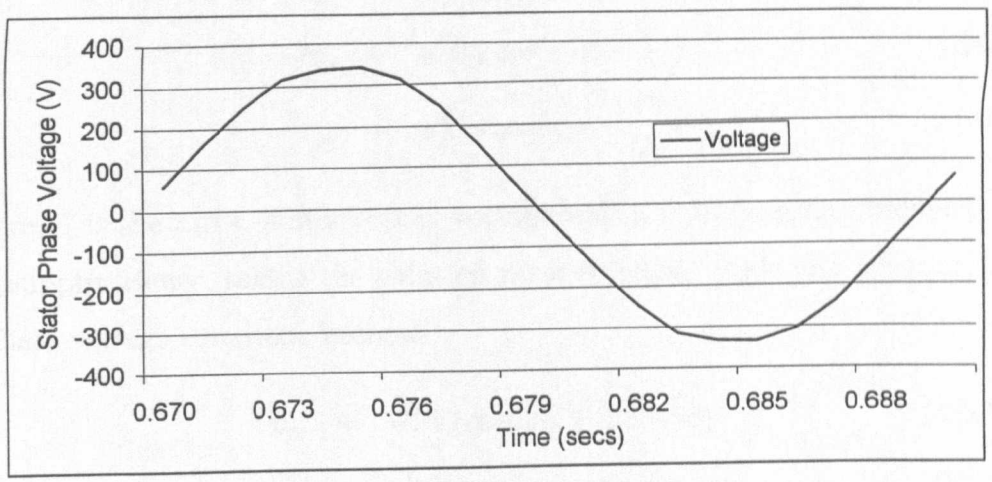

Figure 6.3: Stator phase voltage response from the real induction machine

The transient portions of the data for the red-phase stator voltage and current responses are shown in Figure 6.3 and Figure 6.4 respectively. These responses show the point on wave switching angle as 1.88496 radians. When the point on wave switching angle is zero, the 3-phase voltage equations for

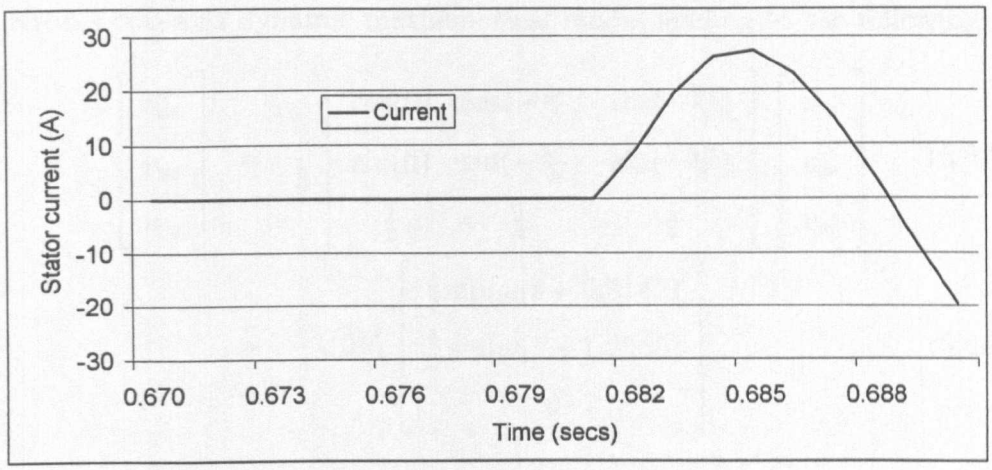

Figure 6.4: Stator current response from the real induction machine

use in the Kron's two-axis dynamic mathematical model are described as

$$v_{rs} = \sqrt{2}V_1 \cos(\omega_1 t) \tag{6.6.8}$$

$$v_{ys} = \sqrt{2}V_1 \cos(\omega_1 t - \frac{2\pi}{3}) \tag{6.6.9}$$

$$v_{bs} = \sqrt{2}V_1 \cos(\omega_1 t - \frac{4\pi}{3}) \tag{6.6.10}$$

where V_1 is the r.m.s. stator supply voltage and ω_1 is the angular frequency of the supply. Hence, taking the point on wave switching angle into account, the 3-phase voltage equations become

$$v_{rs} = \sqrt{2}V_1 \cos(\omega_1 t + 1.88496) \tag{6.6.11}$$

$$v_{ys} = \sqrt{2}V_1 \cos(\omega_1 t - 0.20944) \tag{6.6.12}$$

$$v_{bs} = \sqrt{2}V_1 \cos(\omega_1 t - 2.30383) \tag{6.6.13}$$

By applying the reference frame voltage transformation, as used in Chapter 3, Section 2, we transform the voltage equations into the dq model required by

the Kron's two-axis dynamic mathematical model leading to the following

$$\begin{bmatrix} v_{ds} \\ v_{qs} \\ v_{0s} \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos(0) & \cos(-\frac{2\pi}{3}) & \cos(-\frac{4\pi}{3}) \\ \sin(0) & \sin(-\frac{2\pi}{3}) & \sin(-\frac{4\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} v_{rs} \\ v_{ys} \\ v_{bs} \end{bmatrix} \tag{6.6.14}$$

$$= \frac{2}{3}\sqrt{2}V_1 \begin{bmatrix} \frac{3}{2}\sin(\omega_1 t + 0.3142) \\ \frac{3}{2}\sin(\omega_1 t - 1.2566) \\ 0 \end{bmatrix} \tag{6.6.15}$$

6.6.2 Simulation Results

Has described in Section 6.5, the focus of the identification of a real 3-phase induction machine's parameters applying EP is achieved by minimizing the error vector

$$E(\hat{p}_i^{(g)}, t_k) = Y_2(t_k) - \hat{Y}_2(t_k) \tag{6.6.16}$$

where $Y_2(t_k)$ stands for the measured performance vector taken from the real induction machine, $\hat{Y}_2(t_k)$ the simulated vector generated by the Kron's two-axis dynamic mathematical model and E the expectation operator. Tables 10, 11 and 12 show three of the many different parameter ranges researched for the induction machine parameter identification process, applying the EP over 1000 generations. The parameters to be identified are unknown but, as described in Chapters 2 and 3 are R_s, the stator resistance, R_r, the rotor resistance, L_s, the stator inductance, L_r, the rotor inductance, L_m, the magnetizing inductance, J, the moment of inertia and B, the damping coefficient.

Table 10 shows the parameter ranges coupled with their associated 'identified' parameters located from the EP after 1000 generations. The stator current response and rotor speed response compare the plant, namely the responses generated by the real induction machine, with the EP 'identified' parameter responses. These latter responses result from the 'identified' parameters, shown in Table 10, being substituted into the Kron's two-axis dynamic mathematical

model. The comparison of these responses are shown in Figure 6.5 and Figure 6.6. Clearly, the 'identified' parameters' responses are far from the true responses of the real induction machine.

Table 11 shows a different parameter range coupled with their associated corresponding 'identified' parameters. In particular, the 'identified' inductances, moment of inertia and damping coefficient differ from those shown 'identified' in Table 10. Figure 6.7 and Figure 6.8, once again compare the real stator current and rotor speed responses with those produced by the EP 'identified' parameters, shown in Table 11. Comparison of the true and 'identified' parameter responses clearly indicates that parameter identification has not occurred.

Table 12 shows a much wider ranges of parameters compared with thoses shown in the previous two tables. However, in this case, no parameters were 'identified' by the EP over the 1000 generations.

Tables 10 and 11 show the 'identified' parameters identified by the EP. However, their corresponding responses clearly show that these are not the truly identified parameters. The optimization tool, deployed by the EP, has located these parameters by way of it converging to a local optimal plateau.

The major contributing factor explaining why the EP was not able to identify the real induction machine parameters is coupled to the optimization of the test results, from the real induction machine, with the data generated from Kron's two-axis dynamic mathematical model. This will be discussed in greater depth in the following section.

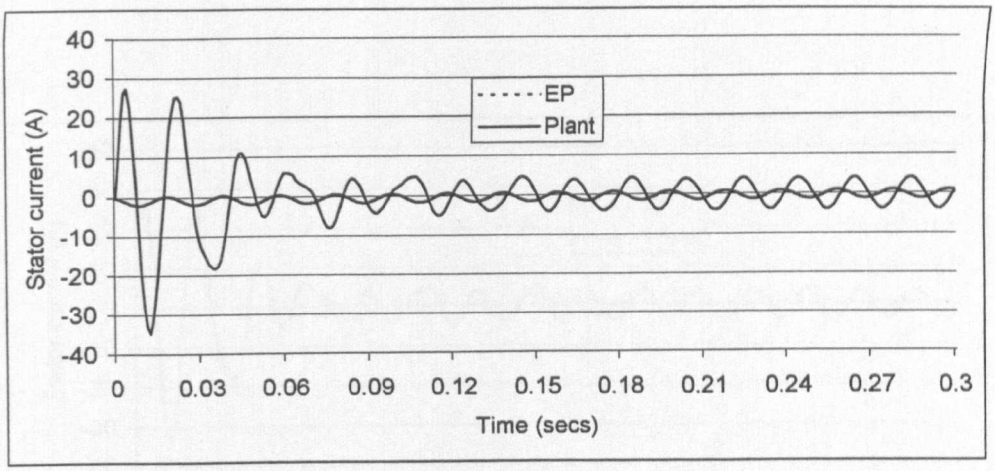

Figure 6.5: Stator current start-up response

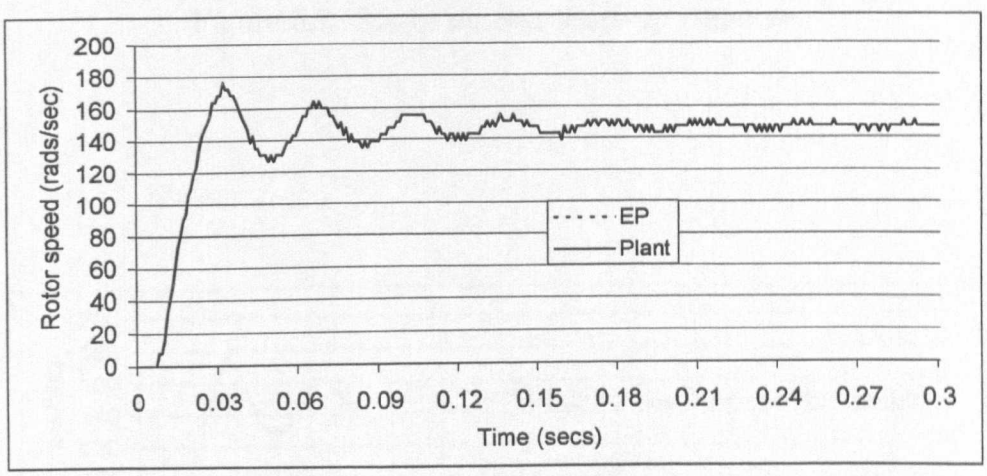

Figure 6.6: Rotor speed start-up response

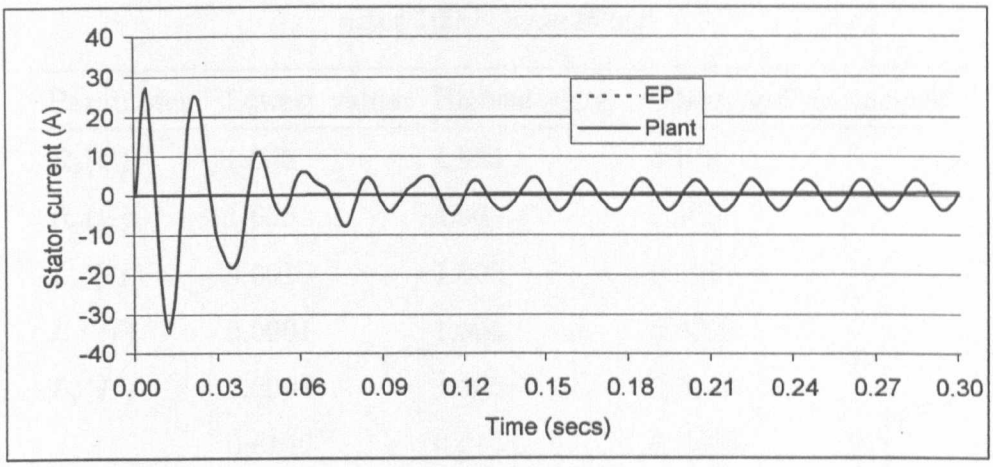

Figure 6.7: Stator current start-up response

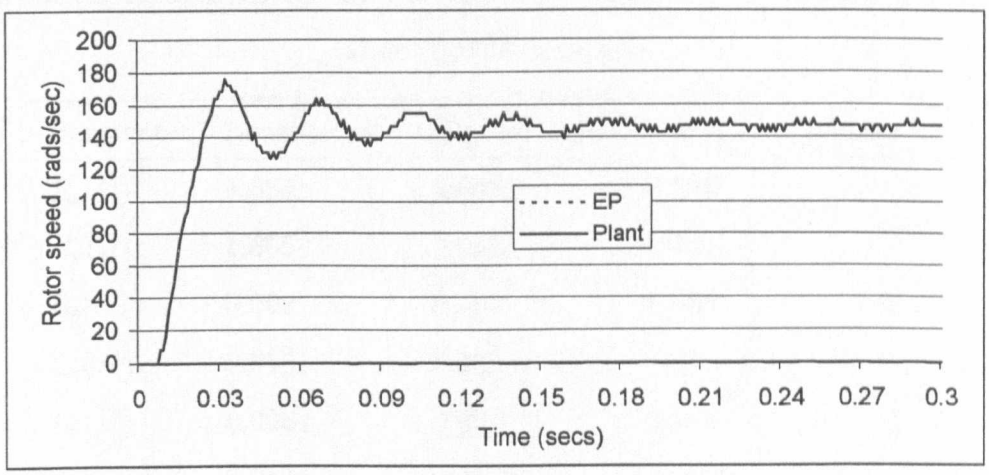

Figure 6.8: Rotor speed start-up response

Table 10

The Parameter Ranges for the EP Search Process showing the 'Identified' Parameters after 1000 Generations

Parameter	Lowest value	Highest value	'Identified' parameter
$R_s(\Omega)$	1.655	4.965	4.648
$R_r(\Omega)$	1.655	4.965	1.655
$L_m(H)$	0.001	1.000	0.001
$L_s(H)$	0.0001	1.000	0.9798
$L_r(H)$	0.0001	1.000	0.9342
J	0.0146	0.0438	0.0408
B	0.0004	0.0012	0.00064

Table 11

The Parameter Ranges for the EP Search Process showing the 'Identified' Parameters after 1000 Generations

Parameter	Lowest value	Highest value	'Identified' parameter
$R_s(\Omega)$	1.655	4.965	4.879
$R_r(\Omega)$	1.655	4.965	1.655
$L_m(H)$	0.001	0.900	0.900
$L_s(H)$	0.0001	0.090	0.009
$L_r(H)$	0.0001	0.090	0.007
J	0.0146	0.0438	0.0438
B	0.0004	0.0012	0.0012

Table 12

The Parameter Ranges for the EP Search Process showing no 'Identified' Parameters after 1000 Generations

Parameter	Lowest value	Highest value
$R_s(\Omega)$	0.010	5.000
$R_r(\Omega)$	0.010	5.000
$L_m(H)$	0.001	10.000
$L_s(H)$	0.001	10.000
$L_r(H)$	0.001	10.000
J	0.0146	0.0438
B	0.00001	0.01000

6.7 Limitations of the Balanced, Symmetrical 3-Phase Induction Machine Mathematical Model

6.7.1 The Kron's Two-Axis Dynamic Mathematical Model

The Kron's two-axis dynamic mathematical model, used throughout this research, models a balanced symmetrical 3-phase induction machine, as discussed in Chapter 2, with the following assumptions:

1. Uniform air gap,

2. linear magnetic circuit,

3. identical stator windings, distributed so as to produce a sinusoidal MMF wave in space with the phases and arranged so that only one rotating MMF wave is established by balanced stator currents,

4. rotor bars are arranged so that, for any fixed time, the rotor MMF wave can be considered to be a space sinusoidal wave having the same number of poles as the stator MMF wave.

The symmetrical machine is an idealised machine and some important factors which effect the performance, and therefore the generated responses, of the actual real induction machine but have been neglected in the symmetrical machine. These are:

1. Nonlinear magnetic circuit,

2. changes in resistance and inductance due to temperature and frequency changes,

3. harmonic content of the MMF wave.

These latter three neglected factors have consequential affects on the responses produced by the symmetrical induction machine, and since this response data together with the response data generated by the real induction machine, are used in the objective function, which we have defined as the error vector in Section 6.6, for the parameter identification problem of the real induction machine. These factors will now be discussed in more detail.

6.7.2 The Nonlinear Magnetic Circuit

Saturable Inductances

In the Kron's two-axis dynamic mathematical model, deploying the dq reference frame, L_s, L_r' and L_m are assumed to be constant. This can be assumed during the steady-state operating conditions of the machine. However, during transient operations in the induction machine, the large reactive currents in the stator windings produce high leakage fluxes and, hence, local saturation of the teeth in the magnetic circuit. This, in turn, leads to the saturation of the leakage and magnetizing inductances. During transient operation, the saturation levels within the induction machine are also changing with time. Saturation of the induction machine's magnetic circuit influences:

1. The magnetic permiability of the iron parts (reduction),
2. the magnitude of the starting currents,
3. the duration of the starting period,
4. the thermal capacity of the machine,
5. the winding insulation,
6. the pulsating torque,
7. the starting current on the power system.

Since the error vector, which is an intricate component of the error function, used by the EP for the parameter identification of a real induction machine, as described in Section 6.6, is

$$E(\hat{p}_i^{(g)}, t_k) = Y_2(t_k) - \hat{Y}_2(t_k) \tag{6.7.1}$$

where $Y_2(t_k)$ is the measured performance vector taken from the real induction machine during the machine start transient period where the real magnetic

circuit saturates, and $\hat{Y}_2(t_k)$ is the simulated vector generated by the Kron's two-axis dynamic mathematical model, which assumes the induction machine operates using a linear magnetic circuit, even during transient operating conditions. As a result, the responses generated by $Y_2(t_k)$ and $\hat{Y}_2(t_k)$ will not be the same.

The induction machine's inductances, L_s and L'_r, as descibed in Chapter 2, Section 10, are the sum of the leakage and magnetizing inductances:

$$L_s = L_{ls} + L_m \tag{6.7.2}$$

$$L'_r = L'_r + L_m \tag{6.7.3}$$

However, to take into account the inductance saturation effects, L_s and L'_r can be described as [12]:

$$L_s = (L_{lsa} + L_{lsi}) + L_m \tag{6.7.4}$$

$$L'_r = (L'_{lra} + L'_{lri}) + L_m \tag{6.7.5}$$

where L_{lsa} and L'_{lra} are the air-dependent end-winding leakage inductances, and the terms L_{lsi} and L'_{lri} correspond to the sum of the iron-dependent saturated leakage inductances [117]. For the saturated magnetic circuit case, inductances L_m, L_{lsa} and L'_{lra} are nonlinear and vary with excitation current.

Hysteresis and Eddy Current Effects

Further nonlinear magnetic circuit effects which are neglected in the Kron's two-axis dynamic mathematical model, although of relatively small effects, are hysteresis and eddy current effects. Hysteresis loss is due to the heating of the core as a result of the internal molecular structure reversals which occur as the magnetic flux alternates. Eddy current loss is heating of the core due to

the e.m.f.'s being induced in the core that generate circulating currents. We shall now discuss other effects not incorporated in the Kron's two-axis dynamic mathematical model.

6.7.3 Skin Effect

The current density induced in the rotor bars changes with rotor current frequency. Consequently, rotor bar resistance and reactance can change by as much as a factor of 4 or more with rotor speed [118].

6.7.4 Temperature Effect

The resistances of the current carrying conductors of the stator and rotor of the induction machine change with temperature [118].

6.7.5 Harmonic Content of the M.M.F. Wave

The complex form of the magnetomotive force (M.M.F.) wave is caused by the harmonic effects due to 'tooth ripple', which results from the effects of the slots in the stator which accommodate the windings, and the non-linearity of the hysteresis loop, namely the relationship between the flux density and magnetic field strength.

6.7.6 Friction and Windage Power Losses

In Chapter 2, Section 5, we stated how Newton's second law yields the equation of motion for the 3-phase induction machine as:

$$\frac{J}{Q}\frac{d\omega_r}{dt} = -T_e + T_r - \frac{B\omega_r}{Q} \tag{6.7.6}$$

However, this equation does not take into account the friction and windage losses which are the mechanical and aerodynamic losses associated with the rotation of the rotor.

6.8 Discussion

In Chapter 4, Section 2, we discussed the processes involved in optimization. Optimization has been applied to the parameter identification of a 3-phase induction machine throughout this research. The principle function used in the optimization process is the objective function. In Chapter 5, Section 3, Table 1 stated the induction machine's parameters that were used in the simulation over the transient start-up period. These parameters remain constant in the Kron's two-axis dynamic mathematical model throughout each time instance of the simulation run. Also, in Chapter 5, Sections 4 and 6, we defined the objective function as an error function which is itself a function of the following error vector:

$$E(\hat{p}_i^{(g)}, t_k) = Y_2(t_k) - \hat{Y}_2(t_k) \tag{6.8.1}$$

The state variables used for the parameter identification of a real induction machine are the direct axis stator current, i_{ds} (which coincides with the red-phase current, i_{rs}, as described in Chapter 3, Section 2), and the rotor speed, ω_r. The parameters of the real induction machine are not constant over the transient start-up period and are unknown. Chapter 5, Section 9, states that the measurement performance vector, using data from the real 3-phase induction machine is:

$$Y_2(t_k) = [i_{rs}(t_k), \omega_r(t_k)]^T \tag{6.8.2}$$

and the calculated performance vector, using simulated data generated by the Kron's two-axis dynamic mathematical model is:

$$\hat{Y}_2(t_k) = C_2 X(t_k) \tag{6.8.3}$$

However, we have observed in this Chapter, Section 8.1, that the Kron's two-axis dq dynamic mathematical model does not include the following factors:

1. Nonlinear magnetic circuit,

2. changes in resistance due to temperature and frequency changes,

3. harmonic content of the MMF wave.

Therefore, the data and responses generated by the state variables of the Kron's two-axis dynamic mathematical model, $\hat{Y}_2(t_k)$, reflect the omission of these factors. However, the data and responses generated by the state variables of the real 3-phase induction machine, $Y_2(t_k)$, include these factors. As a net result, the start-up transient responses, $Y_2(t_k)$ and $\hat{Y}_2(t_k)$, although appear visually similar, do in fact differ.

The parameter identification problem undertaken by the EP evolves as it searches for the electrical parameters and mechanical coefficients so that the state variables of $\hat{Y}_2(t_k)$ tend towards the numerical values of $Y_2(t_k)$. Hence, the error vector should get numerically smaller as the search progresses from generation to generation. This driving objective can be visualised in the error vector:

$$E(\hat{p}_i^{(g)}, t_k) = Y_2(t_k) - \hat{Y}_2(t_k) \tag{6.8.4}$$

However, the calculated performance vector, $\hat{Y}_2(t_k)$, generated by the Kron's two-axis dq dynamic mathematical model contains electrical parameters which remain constant over the 3-phase induction machine's transient start-up period, whereas the measurement performance vector, $Y_2(t_k)$, generated by the real 3-phase induction machine results from varying electrical parameters and, in particular, saturated inductances which are nonlinear, and vary with excitation current and time during the induction machine's transient start-up period.

Therefore, since the parameter identification process in this research takes place over the transient start-up period, the error vector, $E(\hat{p}_i^{(g)}, t_k)$, is not

a true or valid representation of the optimization process in this case. This is because the Kron's two-axis dq dynamic mathematical model is not a true numerical representation of the real 3-phase induction machine over the transient start-up period which is due to the mathematical model not including the factors mentioned above.

As a consequence of the absense of the afore mentioned factors from the Kron's two-axis dq dynamic mathematical model, identification of the 3-phase induction machine's parameters by the EP was not possible over the induction machine's transient start-up period. This, therefore, vindicates the results obtained in this research and presented in Section 7.2 of this Chapter.

Chapter 7

Discussion and Conclusions

7.1 Introduction

This thesis has detailed the development, implementation and testing of the application of SE to the parameter identification process of a 3-phase induction machine. This chapter presents the conclusions drawn from the research work and results.

7.2 Research Synopsis

In this present work, the voltage equations of a balanced, symmetrical 3-phase induction machine have been used to provide the essential mathematical model for this research. Account was taken of the induction machine's inductances and resistances. This brought to light the complexity of the voltage equations due to the time-varying mutual inductances between the stator and rotor circuits, since the circuits are in relative motion. A transformation of machine variables (voltage, current and flux linkages) was applied to the voltage equations to eliminate the time-varying inductances. This transformation has transformed the mathematical model to a two-axis model, the dq axes. To

use the transformed voltage equations for the simulation and analysis required by this work, the arbitrary electrical angular speed was equated to zero in the transformed voltage equations. The direct axis, d, is chosen to coincide with the stator red phase axis, rs. The transformed voltage equations were manipulated so the the dq currents became the state variables. The transformed current equations, together with the induction machine's equation of motion, formed the induction machine's differential equations that were the bedrock throughout this research work.

The principle of optimization is for it to improve performance towards some optimal point or points. The optimization problem may be formulated from an objective function, Φ, of n system parameters, $x_1, x_2, \ldots, x_n$, of the real system response, A_{t_k}, locate the set of parameter values, $x_1^*, x_2^*, \ldots, x_n^*$, of the simulated mathematical model system response, A_{st_k}, over the same time instances, which minimizes the objective function, $\Phi(x_1, x_2, \ldots, x_n)$. Therefore, $x_1^*, x_2^*, \ldots, x_n^*$ are the identified parameters, since the objective function, Φ, has located its optimum value. The traditional optimization methods are calculus-based, enumerative and random searches. Calculus-based methods are local in scope and naturally rely on function derivatives. Calculus-based methods have restrictions placed on them when the data may cover a large search space and may also be fraught with a multitude of noises. Enumerative schemes lack efficiency since real search spaces are generally very large and require a correspondingly large amount of time to search and cover in order to find the optimum solution. Random search algorithms lack efficiency due to their stochastic random approach to orientation in the search space during the parameter identification problem.

SE provides an approach to optimization methods that avoid the afore mentioned difficulties, and is based upon the process of natural evolution that is derived from the collective arguments known as the neo-Darwinian paradigm. It differs from conventional optimization techniques in that it involves direct

manipulation of a coding, it is a population-wide search rather than a point search, and the search is through sampling and using stochastic operators as opposed to deterministic rules. SE is based upon generating successive populations of the feasible solutions in a stochastic manner following laws similar to those of natural selection. Each of the initial solutions is scored with respect to an objective function. Parents generate offsprings whereby the best subset of the solutions are retained to serve as parents for successive generations. The population iteratively adapts its behaviour in light of the given goal. Two paradigms of SE have been applied in this research: GA and EP.

7.3 Effectiveness of Simulated Evolution

7.3.1 Parameter Identification of a Simulated 3-Phase Induction Machine

The electrical parameters of a simulated 3-phase induction machine have been identified by the SE for 5 different cases of measurement noise. The electrical parameters were identified over the series sampling measurements, namely the no-load transient start-up period by a direct-on-line start. Two paradigms of SE were used in the research work, namely EP and GA and both were subject to simulations with two different measurement vectors. The parameter identification process was repeated for simulations on the actual variable values and the per-unit system variable values. Both the 'real' system responses and the simulated system responses, both utilized in the objective function of the optimization process, were generated by the Kron's two-axis dynamic mathematical model. Therefore, both the 'real' and simulated system responses were produced under the assumptions that the symmetrical 3-phase induction machine had linear magnetic circuits, its resistances did not change due to temperature and frequency changes, and there were no harmonic content

in the M.M.F. wave. Indeed, the induction machine parameters were constant over all ranges of slip.

7.3.2 Parameter Identification of a Real 3-Phase Induction Machine

The electrical parameters and mechanical coefficients of a real induction machine have not been identified. The state variables were measured from the real 3-phase induction machine over the series sample measurements, namely the no-load transient start-up period by a direct-on-line start. These state variables formed the real system responses in the objective function of the optimization process. Furthermore, the real system responses, generated by the real 3-phase induction machine, were generated over the transient start-up period, resulting in a wide spread of slip and, hence, when the machine has a nonlinear magnetic circuit, and changes in resistance values due to temperature and frequency changes, as well as harmonic content in the M.M.F. wave. Thus, the electrical parameters' values, of the real 3-phase induction machine, change over the transient start-up period, where the range of slip is at its widest, and they will only settle into constant values during steady-state machine operation. The simulated system responses were formed by the Kron's two-axis dynamic mathematical model. The factors neglected by the Kron's model are the nonlinear magnetic circuit, changes in resistances due to temperature and frequency changes and harmonic content of the M.M.F. wave. Therefore, the electrical parameters of the simulated 3-phase induction machine remained constant over the transient start-up period.

The parameter identification of a real 3-phase induction machine utilized the objective function which was the squared difference between the real and simulated system responses. Because of the resulting differences between the real and simulated system responses, due to variable induction machine param-

eters from the real system responses and constant induction machine parameters from the simulated system responses, over such a range of slip, the EP was unable to identify the real 3-phase induction machine's parameters over the transient start-up period.

7.4 Conclusions

The steady-state electrical parameters of a symmetrical 3-phase induction machine have been identified by the SE over the no-load transient start-up period by a direct-on-line start. This was achieved by virtue of both the real and simulated system responses being generated by the Kron's two-axis dynamic mathematical model, where the nonlinear magnetic circuit, changes in resistances due to temperature and frequency changes, and harmonic content in the M.M.F. wave are neglected. Therefore, the induction machine parameters are constant over the entire range of slip.

The induction machine parameters and mechanical coefficients of a real 3-phase induction machine have not been identified by the EP over the no-load transient start-up period by a direct-on-line start. This was because the real system responses were generated by the real 3-phase induction machine where the electrical parameters vary over the entire range of slip. However, the simulated system responses were generated by the Kron's model, where the induction machine's parameters are constant over the whole range of slip. The process of the EP selecting constant induction machine parameter values, for the entire start-up period simulation, for the criterion of reducing the objective function to its lowest numerical value, when the electrical parameters for the real 3-phase induction machine change for different slip values, provided the optimization process with an insurmountable obstacle.

Appendix A

The Fundamental Processes of Simple GA

This Appendix shows the mechanisms involved in simple GA via a simulation by hand. Let us consider maximizing the function $f(x) = x^2$, where x varies from 0 to 31. We will firstly code the variable x as a binary unsigned integer of string length 5. This binary digit coding yields integer values between 00000(0) and 11111(31).

Selection of a population is obtained by tossing a fair coin 20 times. Where a Head represents a 1 and a tail a 0. Looking at this population, shown on the left-hand side of Table A1, we see that the decoded values of x are presented along with the fitness or objective function $f(x)$.

A generation of the simple GA begins with selection. We select the mating pool of the next generation by spinning the weighted roulette wheel four times. The number of slots in the circumference of the roulette wheel are allocated to each string as a proportion of their fitnesses. This has resulted in string 1 and string 4 receiving one copy in the mating pool, string 2 receiving 2 copies, and string 3 receiving none, as shown on the far right of Table A1. This has resulted in the best getting more copies, the average staying even and the worst

dying off.

Simple crossover proceeds in two steps: (1) strings are mated randomly, using coin tosses to pair off the strings, and (2) mated string couples cross over, using coin tosses to select the crossing sites. Referring to Table A2, one can see the second string has been selected to be mated with the first. With a crossing site of 4, the two strings 01101 and 11000 cross and yield two new strings 01100 and 11001. The same procedure also happens to string 3 and 4 as shown in Table A2.

The last operator, mutation, is performed on a bit-by-bit basis. We assume that the probability of mutation in this test is 0.001. With 20 transferred bit positions we should expect $20 \times 0.001 = 0.02$ bits to undergo mutation during a given generation. Simulation of this process indicates that no bits undergo mutation for this probability value. As a result, no bit positions are changed from 0 to 1 or vice versa during this generation.

Following selection, crossover and mutation, the new population is ready to be tested. The new strings created by the simple GA are decoded and the fitness values are calculated. The results of a single generation of the simulation are shown at the right of Table A2. We can see how the stochastic process of simple GA combine high performance notions to achieve better performance. In the table, note how both the maximal and average performance have improved in the new population. The population average fitness has improved from 293 to 439 in one generation. The maximum fitness has increased from 576 to 729 during that same period. The best string of the first generation (11000) receives two copies because of its high, above-average performance. When this combines at random with the next highest string (10011) and is crossed at location 2 (again at random), one of the resulting strings (11011) proves to be a very good string.

Table A1 A Simple GA by Hand

String No	Initial Population (randomly generated)	x Value (unsigned integer)	$f(x)$ x^2	$pselect_i$ $\frac{f_i}{\sum f}$	Expected *count* $\frac{f_i}{\bar{f}}$	Actual Count *(from* roulette wheel)
1	0 1 1 0 1	13	169	0.14	0.58	1
2	1 1 0 0 0	24	576	0.49	1.97	2
3	0 1 0 0 0	8	64	0.06	0.22	0
4	1 0 0 1 1	19	361	0.31	1.23	1
Sum			1170	1.00	4.00	4.0
Average			293	0.25	1.00	1.0
Max			576	0.49	1.97	2.0

Table A2 A Simple GA by Hand

Mating Pool after Reproduction (cross site shown)	Mate (randomly selected)	Crossover Site (randomly selected)	New Population	x Value	$f(x)$ x^2
0 1 1 0 | 1	2	4	0 1 1 0 0	12	144
1 1 0 0 | 0	1	4	1 1 0 0 1	25	625
1 1 | 0 0 0	4	2	1 1 0 1 1	27	729
1 0 | 0 1 1	3	2	1 0 0 0 0	16	256
Sum					1754
Average					439
Max					729

Appendix B

Glossary

Behaviour: The response of an organism to the present stimulus and its present state. It is the total sum of behaviours of an organism that define the fitness of the organism to its present environment; thus it is the operative function against which selection operates.

Chromatids: The two identical parts of a duplicated chromosome.

Chromosome: Rod-shaped bodies in the nucleus of eukaryotic cells, most visible particularly during cell division, which contains the hereditary units or genes.

Competition: To strive against others in order to win and survive.

Crossing-over: The exchange of corresponding segments of genetic material between chromatids of homologues found in gametes.

Diploid: Twice the number of chromosomes found in gametes.

DNA: Deoxyribonucleic acid.

Ethology: The study of pattern of animal behaviour.

Eukaryotic Cell: Cell with a membrane-enclosed nucleus and organelles found in animals, fungi, plants and protists.

Fitness: A summation of the quality of environmental prediction by an organism throughout its range of regimes. The probability of or propensity for survival of an individual of population.

Gamete: A reproductive cell that joins with another in fertilization to form a zygote, most often an egg or sperm.

Genome: The total genetic constitution of an organism.

Genotype: The sum of inherited characters maintained within the entire reproducing population. Often also used to refer to the genetic constitution underlying a single trait or set of traits.

Haploid: The number of chromosomes.

Homologues: Duplicated chromosomes that look alike and have genes affecting the same traits.

Meiosis: Type of cell division that occurs during the population of gametes, by means of which the daughter cells receive the haploid number of chromosomes.

Morphology: Form and its development.

Mutation: Change of an existing gene to a new gene by altered DNA coding.

Nucleus: A large organelle containing the chromosomes and acting as a control centre for the cell.

Phenotype: The behavioural expression of the genotype in a specific environment. The functional expression of a trait.

Physiology: Science which deals with functions and life processes of plants, animals and human beings.

Pleiotropy: The capacity of a gene to affect a number of different phenotypic characteristics.

Polygeny: The circumstance where a single phenotypic trait is affected by multiple genes.

Polyploidy: Multiple sets of the number of chromosomes in germ cells (23) beyond the normal (46) for tissues other than germ cells, e.g., triploid = 69.

Reproduction: Process of multiplication of living individuals or units whereby the species is perpetuated, either sexual or asexual.

Somatic: Cells of the body other than germ cells.

Zygote: Fertilized egg that is always diploid.

Appendix C

Lists of Symbolic Notation

C.1 The 3-Phase Induction Machine and Mathematical Model Symbols

B damping coefficient

D the differential operator $\frac{d}{dt}$

f supply frequency (Hz)

i_{rr} peak red-phase rotor current (A)

i_{yr} peak yellow-phase rotor current (A)

i_{br} peak blue-phase rotor current (A)

i_{rs} peak red-phase stator current (A)

i_{ys} peak yellow-phase stator current (A)

i_{bs} peak blue-phase stator current (A)

i'_{dr} peak d-axis rotor current referred to the stator (A)

i'_{qr} peak q-axis rotor current referred to the stator (A)

i_{ds} peak d-axis stator current (A)

i_{qs} peak q-axis stator current (A)

I_b per-unit current base value

J	moment of inertia
$\mathbf{K}_s$	stator transformation matrix to the arbitrary reference frame
$\mathbf{K}_s^{-1}$	inverse stator transformation matrix to the 3-phase variables
$\mathbf{K}_r$	rotor transformation matrix to the arbitrary reference frame
$\mathbf{K}_r^{-1}$	inverse rotor transformation matrix to the 3-phase variables
L_{rsrs}	red-phase stator self inductance (H)
L_{ysys}	yellow-phase stator self inductance (H)
L_{bsbs}	blue-phase stator self inductance (H)
L_{ms}	stator magnetizing inductance (H)
L_{ls}	stator leakage inductance (H)
L_{rsys}	stator-to-stator mutual inductance (H)
L_{rsbs}	stator-to-stator mutual inductance (H)
L_{ysrs}	stator-to-stator mutual inductance (H)
L_{ysbs}	stator-to-stator mutual inductance (H)
L_{bsrs}	stator-to-stator mutual inductance (H)
L_{bsys}	stator-to-stator mutual inductance (H)
L_{rrrr}	red-phase rotor self inductance (H)
L_{yryr}	yellow-phase rotor self inductance (H)
L_{brbr}	blue-phase rotor self inductance (H)
L_{mr}	rotor magnetizing inductance (H)
L_{lr}	rotor leakage inductance (H)
L'_{lr}	rotor leakage inductance referred to the stator (H)
L_{rryr}	rotor-to-rotor self inductance (H)
L_{rrbr}	rotor-to-rotor self inductance (H)
L_{yrbr}	rotor-to-rotor self inductance (H)
L_{yrrr}	rotor-to-rotor self inductance (H)
L_{brrr}	rotor-to-rotor self inductance (H)
L_{bryr}	rotor-to-rotor self inductance (H)

L_{sr} amplitude of the mutual inductances between the stator and rotor windings (H)

L_s stator inductance (H)

L_r' rotor inductance referred to the stator (H)

L_m magnetising inductance (H)

L_b per-unit base inductance

N_r rotor windings

N_s stator windings

P_{out} output power (W)

P_b per-unit base power

Q number of pole pairs

R_r rotor resistance per phase (Ω)

R_r' rotor resistance per phase referred to the stator (Ω)

R_s stator resistance per phase (Ω)

R_b per-unit base resistance

s fractional slip

T_e instantaneous electrical torque (Nm)

T_r rated torque (Nm)

T_b per-unit base torque

v_{rs} peak red-phase stator voltage (V)

v_{ys} peak yellow-phase stator voltage (V)

v_{bs} peak blue-phase stator voltage (V)

v_{rr} peak red-phase rotor voltage (V)

v_{yr} peak yellow-phase rotor voltage (V)

v_{br} peak blue-phase rotor voltage (V)

v_{ds} peak d-axis stator voltage (V)

v_{qs} peak q-axis stator voltage (V)

v_{dr}' peak d-axis rotor voltage referred to the stator (V)

v_{qr}' peak q-axis rotor voltage referred to the stator (V)

V_b per-unit base voltage

C.1.1 Greek Letters

ω_b per-unit base angular velocity

ω angular velocity of the arbitrary reference frame (rad/sec)

ω_1 supply voltage angular velocity (rad/sec)

ω_r rotor angular velocity (rad/sec)

ω_s stator angular velocity (rad/sec)

θ_r angular phase displacement between the stator and rotor red phase (rad)

θ angular displacement between the stator d-axis and red phase

λ_{rs} peak red-phase stator flux linkage

λ_{ys} peak yellow-phase stator flux linkage

λ_{bs} peak blue-phase stator flux linkage

λ_{rr} peak red-phase rotor flux linkage

λ_{yr} peak yellow-phase rotor flux linkage

λ_{br} peak blue-phase rotor flux linkage

λ'_{dr} peak d-axis rotor flux linkages referred to the stator

λ'_{qr} peak q-axis rotor flux linkages referred to the stator

λ_{ds} peak d-axis stator flux linkages

λ_{qs} peak q-axis stator flux linkages

C.2 Optimization, Simulated Evolution and Stochastic Operator Symbols

$\mathbf{A}$	a set of all absorbing states
$\mathbf{C}$	coefficient matrix
E	expectation operator
f	fitness of a string
f^*	globally optimum value
h	error function
$\mathbf{I}_a$	$a \times a$ identity matrix describing its absorbing state
$\mathbf{I}_t$	$t \times t$ identity matrix
k	number of transitions
m	population of bits
n	number of bits in a string
$\mathbf{P}$	transition matrix
P_{ij}	probability of transitioning from state i to state j in one step
p	parameter variable matrix
$\mathbf{Q}$	$t \times t$ transition submatrix describing transitions to transient states and not absorbing states
$\mathbf{R}$	$t \times a$ transition submatrix describing transitions to an absorbing state
$\mathbf{T}$	set of all transient states
V	input vector from the supply
X	state variable vector
Y	measured performance vector
$\hat{Y}$	simulated performance vector

C.2.1 Greek Letters

Φ	objective function
π	state vector

Bibliography

Bibliography

Bibliography

[1] Chapman, S. J., *Electric Machinery Fundamentals*, McGraw-Hill, (1999).

[2] Wade, S., Dunnigan, M. W., and Williams, B. W., 'Modelling and Simulation of Induction Machine Vector Control with Rotor Resistance Identification', *IEEE TRAN. Power Electronics*, Vol. 12, No. 3, May 1997.

[3] Vas, P., 'Vector Control of AC Machines', London, U. K.: Oxford Univ. Press, 1990.

[4] Lima, A. M. N., Jacobina, C. B., Filho, E. B. S., 'Nonlinear Parameter Estimation of Steady-State Induction Machine Model', *IEE Tran. Ind. Rlectrn.*, Vol. 44, No. 4, pp. 390 – 397, June 1997.

[5] Lin, F. J., 'Robust Speed Controlled Induction-Motor Drive using EKF and RLS Estimators', *IEE Proc. Electr. Power Appl.*, Vol. 143, No. 3, May 1996.

[6] Lin, F. J., and Su, H. M., 'A High-Performance Induction Motor Drive with On-Line Rotor Time-Constant Estimation', *IEEE Tran. Energy Conversion*, January 1997.

[7] Jarray, K., Gossa, M., Chaari, A. and Jemli, M., 'Induction Machine Parameters and Flux On-Line Estimation using the Recursive Least Squares Algorithm'.

[8] Tsai, H., Keyhani, A., Demcko, J., and Farmer, R. G., 'On-Line Synchronous Machine Parameter Estimation for Small Disturbance Operating Data', *IEEE Tran. Energy Conversion*, Vol. 10, No. 1, pp. 25 – 36, March 1955'

[9] Keyhani, A., Hao, S., and Schulz, R. P., 'Maximum Likelihood and Prediction Error Method', *Automatica*, Vol. 16, pp. 331 – 574, 1980.

[10] Atkinson, D. J., Finch, J. W., and Acarnley, P. P., 'Estimation of Rotor Resistance in Induction Motors', *IEE Proc. Electr. Power Appl.*, Vol. 143, No. 1, pp. 87 – 94, January 1996.

[11] Atkinson, D. J., Finch, J. W., and Acarnley, P. P., 'Observers for Induction Motor State and Parameter Estimation', *IEEE Tran. Industry Appl.*, Vol. 27, No. 6, December 1991.

[12] Faiz, J. and Seifi, A. R., 'Dynamic Analysis of Induction Motors and Saturable Inductances', *Electric Power Systems Research*, 34, pp. 205-210, 1995.

[13] Kim, Y. M., Sol, S. K., and Park, M. H., 'Speed Sensorless Vector Control of Induction Motors using EKF', *IEEE Tran. Industry Appl.*, Vol. 30, No. 5, October 1994.

[14] Ma, J. T., and Wu, Q. H., 'Generator Parameter Identification using Evolutionary Programming', *Electrical Power & Energy Systems*, 17, (6), pp. 417-423, 1995.

[15] Nangsue, P., Pillay, P., and Conroy, S. E., 'Evolutionary Algorithms for Induction Motor Parameter Determination', *IEEE Tran. Energy Conversion*, Vol. 14, No. 3, September 1999.

[16] Cho, D., and Jung, H., 'Induction Motor Design for Electric Vehicle Using a Niching Genetic Algorithm', *IEEE Tran. Industry Appl.*, Vol. 37, No. 4, July/August 2001.

[17] Charette, A., Xu, J., Ba-Razzouk, A., Pillay, P., and Rajagopalan, V., 'The Use of the Genetic Algorithm for in-situ Efficiency Measurement of an Induction Motor', 0-7803-5935-6/00/$10.00, 2000, IEEE.

[18] Alonge, F., D'Ippolito, F., Ferrante, G., and Raimondi, F. M., 'Parameter Identification of Induction Motor Model using Genetic Algorithms', *IEE Proc.-Control Theory Appl.*, Vol. 145, No. 6, November 1998.

[19] Park, R. H., ' Two-Reaction Theory of Synchronous Machines–General Method of Analysis–Part 1', *AIEE Trans.*, Vol. 48, july 1929, pp 716-727.

[20] Stanley, H. C., 'An Analysis of the Induction Motor', *AIEE Trans.*, Vol. 57(Supplement), 1938, pp. 751-755.

[21] Kron, G., *Equivalent Circuits of Electric Machinery,* John Wiley and Sons, Inc., New York, N.Y., 1951.

[22] Brereton, D. S., Lewis, D. G., and Young, C. G., 'Representation of Induction Motor Loads During Power System Stability Studies', *AIEE Trans.*, Vol. 76, August 1957, pp 451-461.

[23] Krause, P. C. and Thomas, C. H., 'Simulation of Symmetrical Induction Machinery', *IEEE Trans. Power Apparatus and Systems*, Vol. 84, November 1965, pp. 1038-1053.

[24] Krause, P. C., Nozari, F., Skvarenina, T. L. and Olive, D. W., 'The Theory of Neglecting Stator Transients', *IEEE Trans. Power Apparatus and Systems*, Vol. 98, Jan/Feb 1979, pp. 141-148.

[25] Fisher, R. A., *The Genetical Theory of Natural Selection*, Oxford: Clarendon Press, (1930).

[26] Wright, S., 'Evolution in Mendelian Populations', *Genetics*, Vol. 16, pp. 97 – 159, (1931).

[27] Haldane, J. B. S., *The Causes of Evolution*, London: Longman, (1932).

[28] Dobzhansky, T., *Genetics and the Origins of Species*, New York: Columbia University Press, (1937).

[29] Huxley, J. S., *Evolution: The Modern Synthesis*, London: Allen and Unwin, (1942).

[30] Mayr, E., *Systematics and the Origins of the Species*, New York: Columbia University Press, (1942).

[31] Simpson, G. G., *Tempo and Mode in Evolution*, New York: Columbia University Press, (1944).

[32] Rensch, B., *Neuere Probleme der Abstammungslehre*, Stuttgart: Fernínand Enke ,(1947).

[33] Stebbins, G. L., *Variation and Evolution in Plants*, New York: Columbia University Press, (1950).

[34] Hoffman, A., *Arguments on Evolution: A Paleontologist's Perspective*, New York: Oxford University Press, p. 39, (1989).

[35] Huxley, J., 'The Evolutionary Process', *Evolution as a Process*, New York: Collier Books, pp. 9 – 33, (1963).

[36] Wooldridge, D. E., *The Mechanical Man: The Physical Basis of Intelligent Life*, New York: McGraw-Hill, p. 25, (1968).

[37] Lewontin, R. C., *The Genetic Basis of Evolutionary Change*, New York: Columbia University Press, (1974).

[38] Atmar, W., 'On the Rules and Nature of Simulated Evolutionary Programming', *Proc. First Ann. Conf. on Evolutionary Programming*, CA: Evolutionary Programming Society, pp. 17 – 26, (1992).

[39] Mayr, E., 'The Evolution of Life', Panel in *Evolution after Darwin: Issues in Evolution*, Vol. 3, Chicago: University of Chicago Press, p. 24, (1960).

[40] Mayr, E., *Populations, Species and Evolution*, Cambridge, MA: Belknap Press, p. 131, (1970).

[41] Hartl, D. L., and Clark, A. G., *Principles of Population Genetics*, 2nd ed., Sutherland, MA: Sinauer, p. 431, (1989).

[42] Mayr, E., *Towards a New Philosophy of Biology: Observations of an Evolutionist*, Cambridge, MA: Belknap Press, (1988).

[43] Brunk, C. F., 'Darwin in an Age of Molecular Revolution@, *Contention*, Vol. 1:1, pp. 131 – 150, (1990).

[44] Atmar, W., 'Notes on the Simulation of Evolution', *IEEE Trans. on Neural Networks*, Vol. 5:1, pp. 130 – 148, (1994).

[45] Bonner, J. T., *The Evolution of Complexity by Means of Natural Selection*, Princeton, NJ: Princeton University Press, (1988).

[46] Atmar, J. W., 'The Inevitability of Evolutionary Invention', Unpublished manuscript, (1979).

[47] Sheppard, P. M., *Natural Selection and Heredity*, 4th ed. London, Hutchinson, (1975).

[48] Keeton, W. T., *Biological Science*, 3rd ed. New York: W. W. Norton, (1980).

[49] Gould, J. L., and Gould, C. G., *Sexual Selection*, New York: Scientific American Library, (1989).

[50] Crow, J. F., 'The Importance of Recombination', in *The Evolution of Sex: An Examination of Current Ideas*, edited by Michod, R. E. and Levin, B. R., Sunderland, MA: Sinauer, pp. 56 – 73, (1988).

[51] Tinbergen, N., *The Study of Instinct*, Oxford: Oxford University Press, (1951).

[52] Lack, D. L., *The Life of the Robin*, London: H. F. and G. Whiterby, (1946).

[53] Maynard Smith, J., *Did Darwin Get It Right?: Essays on Games, Sex, and Evolution*, New York: Chapman and Hall, (1988).

[54] Fogel, D. B., 'Evolving Artificial Intelligence', Ph.Dd diss., UCSD, (1992).

[55] Goodman, R., *Introduction to Stochastic Models*, Reading, MA: Benjamin/Cummings, (1988).

[56] Holland, J. H., *Adaptation in Natural and Artificial Systems*, Ann Arbor: University of Michigan Press, (1975).

[57] De Jong, K. A., 'Tha Analysis of the Behaviour of a Class of Genetic Adaptive Systems', Ph.D. diss. University of Michigan, Ann Arbbor, (1975).

[58] Baker, J. E., 'Adaptive Selection Methods for Genetic Algorithms', *Proc. of Intern. Conf. on Genetic Algorithms and Their Applications*, ed. by J. J. Grefenstette, San Mateo, CA: Morgan Kaufman, pp. 101 – 111, (1985).

[59] Davis, L., *Handbook of Genetic Algorithms*, New York: Van Nostrand Reinhold, (1991).

[60] Rudolph, G., 'Convergence Analysis of Camnonical Genetic Algorithms', *IEEE Trans. on Neural Networks*, Vol. 5:1, pp. 96 – 101, (1994).

[61] Serfling, R. J., *Approximation Theorems of Mathematical Statistics*, New York: John Wiley, (1980).

[62] Beighter, C. S., Phillips, D. T., and Wilde, D. J., *Foundations of Optimization*, Englewood Cliffs, NJ : Prentice-Hall, (1979).

[63] Simon, H. A., *The Sciences of the Artificial*, Cambridge, MA: MIT Press, (1969).

[64] Dixon, L. C. W. (Ed), *Optimization in Action*, Academic Press, (1976).

[65] Massara, R. E., *Optimization Methods in Electronic Circuit Design*, Longman Scientific and Technical, (1991).

[66] Fiacco, A. V., McCormick, G. P., *Nonlinear Programming*, John Wiley, (1968).

[67] Fletcher, R., *Practical Methods of Optimization* –Volume 2: *Constrained Optimization*, John Wiley, (1981).

[68] Gill, P. E., Murray, W. (Eds), *Numerical Methods for Constrained Optimization*, Academic Press, (1974).

[69] Lootsma, F. A., *Numerical Methods for Nonlinear Optimization*, Academic Press, (1972).

[70] Han, S. P., 'Superlinearly convergent variable metric algorithms for nonlinearly programming', *Math. Prog.* 11: 263–282, (1976).

[71] Han, S. P., 'A globally convergent method for nonlinear programming', *J. Opt. Theory and Applics.*, 22: 297–309, (1977).

[72] Fletcher, R., Powell, M. J. D., 'A rapidly convergent descent method for minimization', *Computer J.*, 6: 163–168, (1963).

[73] Fletcher, R., Reeves, C. M., 'Function minimization by conjugate gradients', *Computer J.*, 7: 149–154, (1964).

[74] Huang, C. A., 'Unified approach to quadratically convergent algorithms for function minimization', *J. Optimization Theory and Applications*, 5: 405–423, (1970).

[75] Marquardt, D. W., 'An algorithm for least squares estimation of nonlinear parameters', *J. SIAM*, 11: 431–441, (1963).

[76] Murray, W., (Ed), *Numerical Methods for Unconstrained Optimization*, Academic Press, (1972).

[77] Fletcher, R., *Practical Methods of Optimization -Volume 1: Unconstrained Optimization*, John Wiley, (1980).

[78] Box, M. J., Davies, D., Swann, W. H., *Nonlinear Optimization Techniques*, Oliver and Boyd, (1969).

[79] Hooke, R., Jeeves, T. A., 'Direct search solution of numerical and statistical problems', sl JACM, 8: 212–219, (1961).

[80] Palmer, J. R., 'An improved procedure for orthogonalising the search vectors in Rosenbrock's and Swann's direct search optimization methods', *Computer J.*, 12: 69–71, (1969).

[81] Powell, M. J. D., 'On the calculation of orthogonal vectors', *Computer J.*, 11: 302–304, (1968).

[82] Rosenbrock, H. H., 'An automatic method for finding the greatest or least value of a function', *Computer J.*, 3: 175–184, (1960).

[83] Swann, W. H., 'Report on the development of a new direct search method of optimization', *ICI Central Instrument Laboratory Research Note*, 64–3, (1964).

[84] Bellman, R., *Adaptive Control Processes: A Guided Tour*, Princeton, NJ: Princeton University Press, (1961).

[85] Huang, K. S., Wu, Q. H., Turner, D. R., 'Effective identification of FOC induction motor parameters based on fewer measurements', (1999).

[86] Fogel, D. B., *System Identification through Simulated Evolution: a Machine Learning Approach to Modelling*, Ginn Press, MA, (1991).

[87] Fogel, D. B., *Towards a New Philosophy of Machine Intelligence*, IEEE Press, (1995).

[88] Holland, J. H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, (1975).

[89] Holland, J. H., 'Processing and processor for schemata', F. L. Jacks (Ed), *Associative Information Processing*, American Elsevier, NY, pp. 127–146, (1971).

[90] Holland, J. H., 'Schemata and intrinsically parallel adaptation', *Proc. of NFS: Workshop on Learning System Theory and its Application*, pp. 43–46, (1973).

[91] Hollstein, P. B., 'Artificial genetic adaptation in computer control systems', *Dissertation Abstract Int.*, 32, (2), 1510B, (1971).

[92] De Jong, K. A., *The Analysis of the Behaviour of a Class of Genetic Adaptive Systems*, Doctoral Dissertation, University of Michigan, Ann Arbor, (1975).

[93] Goldberg, D. E., 'Computer-aided gas pipeline operation using genetic algorithms and rule learning', *Dissertation Abstract Int.*, 44, (10), 3174B, (1983).

[94] Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Publishing Company, (1989).

[95] Englander, A. C., 'Machine learning of visual recognition using genetic algorithms', *Proc. 1st Int. Conf. on Genetic Algorithms*, pp. 197–202, (1985).

[96] Grefenstette, J. J., Fitzpatrick, J. M., 'Incorporating problem specific knowledge in genetic algorithms', *Proc. of 1st Int. Conf. on Genetic Algorithms*, pp. 112–120, (1985).

[97] Ackley, D. H., 'A connection algorithm for genetic search', *Proc. 1st Int. Conf. on Genetic Algorithms*, pp. 121–135, (1985).

[98] Cohoon, J. P., et al, 'Punctuated equilibria: A parallel genetic algorithm', *Proc. 2nd Int. Conf. on Genetic Algorithms*, pp. 148–154, (1987).

[99] Davis, L., *Handbook of Genetic Algorithms*, New York: Van Nostrand Reinhold, (1991).

[100] Cerf, R., 'Asymptotic convergence of a genetic algorithm', *Comptes Rendus de l'Academie des Sciences*, Serie 1, 319, (3), pp. 271–276.

[101] Eiben, A. E., Aarts, E. H. and Van Hee, K. M., 'Global convergence of genetic algorithms: An infinite Markov chain analysis', *Parallel Problem Solving from Nature*, Schwefel, H. and Männer, Eds., Heidelberg, Spring-Verlag, pp. 4–12, (1991).

[102] Fogel, D. B., 'Asymptotic convergence properties of genetic algorithms and evolutionary programming: Analysis and experiments', *Cybernetics and Systems*, pp. 389–407, (1994).

[103] Nix, A. and Vose, M. D., 'Modelling genetic algorithms with Markov chains', *Annals of Mathematics and Artificial Intelligence*, 5, pp. 27–34, (1992).

[104] Qi, X. and Palmieri, F., 'Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space, Part 1: Basic properties of selection and mutation', *IEEE Trans. on Neural Networks*, 5, (1), pp. 102–119, (1994).

[105] Qi, X. and Palmieri, F., 'Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space, Part 2: Analysis of the diversification role of crossover', *IEEE Trans. on Neural Networks*, 5, (1), pp. 120–129, (1994).

[106] Rudolph, G., 'Convergence properties of canonical genetic algorithms', *IEEE Trans. on Neural Networks*, 5, (1), pp. 96–101, (1994).

[107] Rudolph, G., 'Convergence of non-elitist strategies', *IEEE Conference Proc. on Evolutionary Computation*, New York, pp. 63–66, (1994).

[108] Suzuki, J., 'A Markov chain analysis on simple genetic algorithms', *IEEE Trans. on System, Man and Cybernetics*, 25, (4), pp. 655–659, (1995).

[109] Vose, M. D., 'Modelling of genetic algorithms', *Foundations of Genetic Algorithms*, 2, Whitley, L. D., Ed., pp. 63–73, (1992).

[110] Schraudolph, N. N., Belew, R. K., 'Dynamic parameter encoding for genetic algorithms', *Machine Learning*, 9, pp. 9–21, (1992).

[111] Fogel, L. J., 'Autonomous automata', *Industrial Research*, 4, pp. 14–19, (1962).

[112] Fogel, L. J., 'Toward inductive inference automata', *Proc. of the Int. Federation for Information Processing Congress*, Munich, pp. 395–400, (1962).

[113] Fogel, L. J., *Biotechnology: Concepts and Applications*, Englewoods Cliffs, NJ: Prentice Hall,(1963).

[114] Fogel, L. J., Owens, A. J., and Walsh, M. J., *Artificial Intelligence through Simulated Evolution*, New York: John Wiley, (1966).

[115] Fogel, L. J., *On the Organization of Intellect*, PhD Diss., UCLA, (1964).

[116] Fogel, D., *Evolving Artificial Intelligence*, Doctoral Dissertation, UCSD, (1992).

[117] Lipo, T. A. and Consoli, 'Modelling and Simulation of Induction Motors with Saturable Leakage Reactances', *IEEE Trans. on Ind. Appl.*,IA–20, pp. 180–189, (1984).

[118] Ruppert Fillio, E. and Avolio, E., 'Squirrel-Cage Induction-Machine Dynamic Simulations using an Electrical and Thermal Mathematical Model based on the Manufacturer's Technical Bulletin Data and on Technical Standard Statement', *International Journal of Power and Energy Systems*, Vol. 14, No. 1, (1994).

[119] Huang, K. S., Wu, Q. H., and Turner. D. R., 'Effective Identification of Induction Motor Parameters Based on Fewer Measurements', *IEEE Trans. on Energy Conversion*, Vol. 17, No. 1, March 2002.