

# Estimation of Synchronous Generator Parameters using Time-domain Responses

ASG. Galbraith

Thesis presented in partial fulfilment of the requirements for the degree of

Master in Engineering

at the

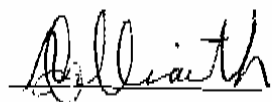
University of Stellenbosch



Supervisor: Prof. H.J. Vermeulen  
December 2005

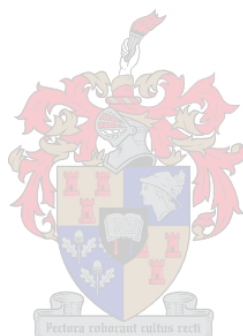
## Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work, and has not previously, in its entirety or in part, been submitted at any university for a degree.



ASG. Galbraith

30 September 2005



## Abstract

Power system stability simulations are of growing importance for studying the operational integrity of modern power systems, especially in developing economies where generating and transmission capacity lead the demand by relatively small margins. The relevant model topologies, i.e. for synchronous generators, automatic voltage regulators (AVR) and governor control systems, and the simulation software tools are well established.

The MATLAB<sup>®</sup> Power System Blockset provides engineers with a versatile power system stability simulation environment, particularly where the focus is on individual units or small systems. In comparison with dedicated power system simulation tools such as DIgSILENT<sup>®</sup>, the MATLAB<sup>®</sup> environment features a superior set of advanced data processing and data analysis features. This includes features such as optimisation and parameter estimation functions.

The main aim of this project is to make use of the MATLAB<sup>®</sup> package in a bid to test an alternative platform with which to estimate the synchronous machine parameters. Conditioning of field data can delay the process considerably, thus the secondary task of this thesis is to solve this issue by ensuring that only one platform is needed for the entire process starting in the field and ending in the modelling and parameter estimation environment within MATLAB<sup>®</sup>.

In closing, the following points summarise the essential aims of this project:

- An application using MATLAB<sup>®</sup> Script must be created that is responsible for importing and processing the data, so it is suitable for analysis purposes. The processing could include cropping, scaling and filtering of data.
- Once the data has been imported it must be used with appropriate models to estimate for machine parameters. This will require the use of the Power Systems Blockset. The actual estimation process also requires the creation of an effective cost function, thus a number of different scenarios will have to be investigated before a solution can be found.

## Opsomming

Kragstelsel-stabiliteitsimulasies is van toenemende belang vir die studie van operasionele intergniteit van moderne kragstelsels, veral in ontwikkelende lande waar die kapasiteit die aanvraag marginaal oorskry. Die relevante model topologieë vir sinchroongenerators, outomatiese spanningsreguleerders, spoedreëlaarstelsels en simulasië sageware gereedskap is goed gevestig.

Die MATLAB<sup>®</sup> *Power System Blockset* voorsien 'n veeldoelige kragstelsel-stabiliteit simulasië omgewing veral met die oog op individuele eenhede of klein stelsels. In vergelyking met doelgerigte kragstelselsimulasiëgereedskap soos DIGSILENT<sup>®</sup> voorsien die MATLAB<sup>®</sup> omgewing gevorderde data-prosesserings en analitiese eienskappe. Hierdie eienskappe sluit in optimerings en parameter estimasië funksies.

Die hoof doel van hierdie projek is om die Matlab omgewing te gebruik om 'n alternatiewe platform te evalueer vir die estimasië van sinchroonmasjien parameters. Hierdie proses word egter aansienlik vertraag deur die kondisionering van veld data. Hierdie probleem word aangespreek deur die daarstelling van 'n enkele platform wat benodig word vir die algehele proses. Hierdie proses neem 'n aanvang in die veld en eindig met die modellering en parameter estimasië binne MATLAB<sup>®</sup>.

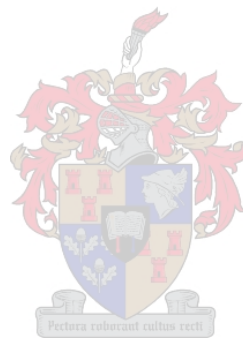
Die volgende punte is 'n opsomming van die belangrike doelwitte van hierdie projek.

- 'n Toepassing moet geskep word deur gebruik te maak van *Matlab Script* wat verantwoordelik is vir die inlees en verwerking van data sodat dit geskik is vir analitiese doeleindes. Hierdie proses mag data seleksie, skalering en filering insluit.
- Die ingeleesde data moet verwerk word met 'n geskikte model om die masjien parameters te estimateer deur gebruik te maak van die *Power Systems Blockset*. Die uiteindelijke estimasië proses benodig ook die skep van 'n effektiewe koste funksie. Hierdie funksie word bepaal deur verskillende scenario's te ondersoek alvorens 'n oplossing gevind kan word.

## Acknowledgements

I would like to thank the following individuals for their contributions and support:

1. My Family.
2. Prof. H.J. Vermeulen.
3. Daniela Kalinowski.
4. Francios, Steyn, Thinus, Prencorp, Munsie and all those other people who put up with my antics.



# Contents

<b>1. Introduction .....</b>	<b>1</b>
1.1. Background to Power System Stability Problem .....	1
1.2. Project Motivation .....	3
1.2.1. Field Measurements .....	4
1.2.2. Data Processing .....	5
1.2.3. Modelling and Analysis .....	6
1.2.4. The Parameter estimation process .....	6
1.3. Conclusion .....	7
<b>2. Literature Survey .....</b>	<b>8</b>
2.1. Power System Stability Studies .....	8
2.2. Overview of the Power System Stability Problem .....	8
2.2.1. The Basic Synchronous Generator .....	8
2.2.2. Power versus angle relationship .....	9
2.2.3. Small signal stability .....	12
2.2.4. Transient stability .....	13
2.3. The Synchronous Machine .....	14
2.3.1. The Rotor .....	16
2.3.2. Magnetic Saturation .....	17
2.3.3. Voltage and Torque Equations in Machine Variables .....	18
2.4. Estimation of Synchronous Generator Parameters .....	24
2.4.1. Introduction .....	24
2.4.2. Rotary Synchronous Machine Modelling .....	24
2.4.3. Traditional tests .....	24
2.4.4. Transient Response tests .....	26
2.5. Cost Function - Least Squares Method .....	30
<b>3. The Simulink simulation environment .....</b>	<b>31</b>
3.1. Introduction .....	31
3.1.1. Simulink Environment .....	32
3.2. Network Topology .....	33
3.3. Pre-simulation initialisation .....	34
3.4. Nonlinear Models .....	35
3.5. Simulink Solvers .....	36

3.5.1.	Solver type.....	36
3.5.2.	Error tolerances .....	37
<b>4.</b>	<b>Estimation of the Rotor Parameters.....</b>	<b>39</b>
4.1.	Introduction .....	39
4.2.	Measured Field Data used for Estimation of the Rotor Parameters.....	40
4.3.	Rotor Circuit Model .....	42
4.4.	Calculation of Field Impedances.....	43
4.4.1.	Field Base Values (Reciprocal per unit system) .....	43
4.4.2.	Change of Base (Reciprocal to Non-Reciprocal).....	44
4.4.3.	Calculation of Theoretical Field Parameter Values .....	48
4.5.	Method 1: RL Circuit Implementation in Simulink.....	48
4.5.1.	Cost Function: Measured and Simulated Field Current Comparison .....	48
4.5.2.	Test: 10 % PRBS Input Signal .....	49
4.6.	Method 2: RL Equation Implemented in Simulink.....	50
4.6.1.	Network Model Topology and Cost Function.....	50
4.6.2.	Test 1 – 1% PRBS Waveforms .....	51
4.6.3.	Test 2 – 10 % PRBS Waveforms .....	53
4.6.4.	Verification of Estimator.....	54
4.7.	Method 3: Detailed RL Equation Implemented in MATLAB® .....	55
4.7.1.	Derivation of Rotor Winding Transfer Function.....	56
4.7.2.	Cost Function .....	57
4.7.3.	Estimator Integrity Test.....	58
4.7.4.	Estimation of Field Parameters using PRBS Waveforms .....	60
4.8.	Results and Conclusions.....	65
<b>5.</b>	<b>Xd and Xq Parameters .....</b>	<b>67</b>
5.1.	Introduction .....	67
5.2.	System Setup .....	68
5.3.	Simulation Test Cases .....	71
5.3.1.	Test 1: 0.8 p.u. active power, 0.256 p.u. reactive power export .....	71
5.3.2.	Test 2: 0.8 p.u. active power, 0.2 p.u. reactive power import.....	73
5.3.3.	Test 3: 0.8 p.u. active power, 0 reactive power.....	74
5.3.4.	Test 4: 0.5 p.u. active power, 0.5 reactive power export .....	75
5.3.5.	Test 5: 0.5 p.u. active power, 0.5 reactive power import.....	76
5.3.6.	Test 6: Combined Cost Functions: Tests 1 and 2.....	77

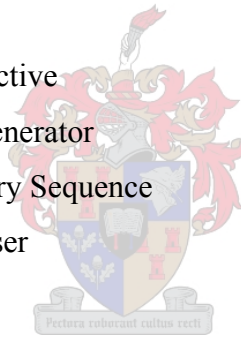
5.3.7.	Test 7: Combined Cost Functions: Tests 4 and 5.....	78
5.3.8.	Test 8: Combined Cost Function: Test Cases 1-3 .....	80
5.4.	Estimator performance results.....	81
5.4.1.	System with single machine exporting MVARs .....	81
5.4.2.	Two machines in parallel; Import and Export set points .....	82
5.4.3.	Three machines in parallel; Import, Export and Zero MVAR's .....	83
<b>6.</b>	<b>GPIS Data Analyser .....</b>	<b>83</b>
6.1.	Software Specifications.....	84
6.2.	Data Handling: Importing of measured data .....	85
6.2.1.	YOKOGAWA ASCII FILE FORMAT .....	85
6.2.2.	LabView file format .....	90
6.2.3.	Data structure in the MATLAB® Workspace .....	92
6.3.	Graphic User Interface (GUI) Layout .....	93
6.4.	GPIS Data Analyser - Toolboxes.....	94
6.4.1.	Scaling Toolbox .....	94
6.4.2.	Filter Toolbox.....	96
6.4.3.	The Frequency (Speed) Tracker Toolbox: Interpolation Method .....	99
6.4.4.	The Frequency (Speed) Tracker Toolbox: Newton-Raphson Method.....	105
6.4.5.	Root Mean Square (RMS) Toolbox .....	108
6.4.6.	Power Toolbox .....	112
<b>7.</b>	<b>Conclusions and Recommendations .....</b>	<b>114</b>
7.1.	Conclusion.....	114
7.2.	Recommendations .....	115
<b>8.</b>	<b>References .....</b>	<b>117</b>
	Appendix A - Magnetically Coupled Circuits.....	119
	Appendix B – Air Gap MMF .....	125
	Appendix C – Synchronous Generator Inductance Matrices .....	132
	Appendix D – MATLAB® Script Files and Simulink Models.....	143
	Appendix E - System Components .....	152



## List of Abbreviations and Symbols

### Basic Abbreviations

ATOL	Absolute Tolerance
AVR	Automatic Voltage Regulator
DC	Direct Current
GUI	Graphic User Interface
HTG	Hydro Turbine Governor
KVL	Kirchoff's Voltage Law
LSM	Linear Synchronous Machine
MMF	Magnetomotive Force
MVA	Mega Volt Amps
MVAR	Mega Volt Amps Reactive
PMG	Permanent Magnet Generator
PRBS	Pseudo-Random Binary Sequence
PSS	Power System Stabiliser
RMS	Root Mean Square
RSM	Rotary Synchronous Machine
RL	Resistance and Inductance
RTDS	Real Time Digital Simulator
RTOL	Relative Tolerance
OCC	Open Circuit Curves
WVF	Waveform Data File Format



### Symbols

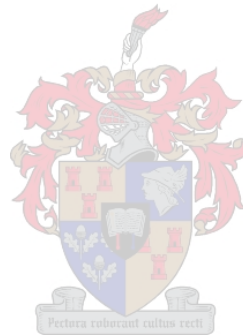
$E_g, E_m$	Voltage vector for generator and motor
$X_g, X_l, X_m$	Inductance for generator, line and motor
$Y$	Admittance

$Z$	Impedance
$\delta$	Angle between sending and receiving end voltage in degrees
$S_R$	Complex power at receiving end
$V_R, V_S$	Receiving and sending voltage
$I_R, I_S$	Receiving and sending current
$\delta_G, \delta_L, \delta_M$	Angle of generator, line and motor in degrees
$a, b, c, a', b', c'$	Phases
$n_s$	Speed of machine in rev/min
$f$	Frequency in hertz
$p_f$	Number of pole pairs
$i_{as}, i_{bs}, i_{cs}$	Line currents of stator windings as, bs and cs.
$fd$	Field winding of rotor.
$f_{as}, f_{bs}, f_{cs}$	Variables of the stator windings (flux linkages, currents or voltages).
$f_{qs}^r, f_{ds}^r, f_{0s}$	Variables of the q-axis, d-axis and zero variable windings of the stator transformed to the reference frame fixed with respect to the rotor.
$i_{fd}, i_{kd}$	Currents in the field (fd) and d-axis damper (kd) windings of the rotor circuit.
$i_{kq}, i_{kq1}, i_{kq2}$	Currents of the q-axis damper windings (kq, kq1, kq2).
$i_{fd}^r, i_{kd}^r$	Currents of the field (fd) and d-axis damper (kd) winding, referred to the stator and transformed to a fixed reference frame with respect to the rotor.
$i_{kq}^r, i_{kq1}^r, i_{kq2}^r$	Currents of the q-axis damper windings (kq, kq1, kq2), referred to the stator and transformed to the reference frame fixed with respect to the rotor.
$i_{qs}^r, i_{ds}^r, i_{0s}$	Currents of the q-axis, d-axis and zero variable windings of the stator transformed to the reference frame fixed with respect to the rotor.
$K_s^r$	Matrix for the transformation of the three phase variables of the stationary circuit elements of the arbitrary reference frame fixed with respect to the rotor.

$k_d, k_{d1}, k_{d2}$	Represent the d-axis damper windings $k_d, k_{d1}$ and $k_{d2}$ respectively, where the damper windings $k_d$ and $k_{d1}$ both represent the first damper winding and $k_{d2}$ denotes the second damper winding. When there is only one d-axis damper winding this is denoted by $k_d$ .
$k_q, k_{q1}, k_{q2}$	Represent the q-axis damper windings $k_q, k_{q1}$ and $k_{q2}$ respectively, where the damper windings $k_q$ and $k_{q1}$ both represent the first damper winding and $k_{q2}$ denotes the second damper winding. When there is only one d-axis damper winding this is denoted by $k_q$ .
$L_{ad}, L_{adu}$	Mutual inductance between the stator and field, unsaturated form.
$L_r$	Inductance matrix to obtain the flux linking the rotor windings due to current flowing through the rotor windings.
$L'_r$	Inductance matrix to obtain the flux linking the rotor windings rotor winding currents referred to the stator.
$L_s$	Inductance matrix to obtain the flux linking the stator windings due to current flowing through the stator windings.
$L_{sr}$	Inductance matrix to obtain the flux linking the stator windings due to current flowing through the rotor windings.
$L'_{sr}$	Inductance matrix to obtain the flux linking the rotor windings due to current flowing through the rotor winding referred to the stator.
$L_{kfd}$	Mutual leakage inductance.
$L_{kfd0}$	This inductance is part of the mutual impedance between the field and rotor body.
$L'_{lfd}, L'_{lkd}$	Leakage inductance of the field and damper windings referred to the stator.
$L'_{lkd}, L'_{lq1}, L'_{lkq2}$	Leakage inductances of the q-axis damper windings $k_q, k_{q1}$ and $k_{q2}$ referred to the stator.
$L_{ls}$	Stator leakage inductance.
$L_{md}, L_{mq}$	d and q-axis magnetising inductances.
$N_{fd}, N_{kd}$	Equivalent turns of the field winding $fd$ and the d-axis damper winding $kd$ .
$N_{kq}, N_{kq1}, N_{kq2}$	Equivalent turns of the q-axis damper winding $k_q, k_{q1}$ and $k_{q2}$ .
$N_s$	Equivalent turns of the stator windings.
$p$	Operator for the time derivative $d/dt$ .
$R_{fd}, R_{kd}$	Resistances of the field winding $fd$ and the d-axis damper winding $kd$ .

$R_{kq}, R_{kq1}, R_{kq2}$	Resistances of the q-axis damper windings kq, kq1 and kq2.
$R'_{fd}, R'_{kd}$	Resistances of the field winding fd and the d-axis damper winding kd, referred to the stator.
$R'_{kq}, R'_{kq1}, R'_{kq2}$	Resistances of the q-axis damper windings kq, kq1 and kq2, referred to the stator.
$R_{kfd}$	Resistance forming part of the mutual impedance between the field and rotor body.
$r_s$	Stator winding resistance.
$V_{as}, V_{bs}, V_{cs}$	Terminal voltages of stator windings as, bs and cs.
$V_{fd}, V_{kd}$	Terminal voltages of the field winding fd and d-axis damper winding kd.
$V_{kq}, V_{kq1}, V_{kq2}$	Terminal voltages of the q-axis damper windings kq, kq1 and kq2.
$V'^r_{fd}, V'^r_{kd}$	Terminal voltages of the field winding fd and d-axis damper winding kd, referred to the stator and transformed to the reference frame fixed with the rotor.
$V'^r_{kq}, V'^r_{kq1}, V'^r_{kq2}$	Terminal voltages of the q-axis damper windings kq, kq1 and kq2, referred to the stator and transformed to the reference frame fixed with the rotor.
$V^r_{qs}, V^r_{ds}, V_{0s}$	Terminal voltages of the q-axis, d-axis and zero variable windings of the stator transformed to the reference frame fixed with respect to the rotor.
$X_d, X_q$	d-axis and q-axis synchronous reactances.
$X'_q, X'_d$	d-axis and q-axis transient reactances.
$X_{ls}$	Stator leakage reactance.
$X'_{lfd}, X'_{lkq}$	Leakage reactances of the field winding fd and the q-axis damper winding kq referred to the stator.
$X_{md}, X_{mq}$	d-axis and q-axis magnetising reactances.
$Z_{fd}$	Field impedance.
$\lambda_{as}, \lambda_{bs}, \lambda_{cs}$	Flux linkages of the as, bs and cs windings.
$\lambda_{fd}, \lambda_{kd}$	Flux linkages of the field winding fd and d-axis damper winding kd.
$\lambda_{kq}, \lambda_{kq1}, \lambda_{kq2}$	Flux linkages of the q-axis damper windings kq, kq1 and kq2.
$\lambda'^r_{fd}, \lambda'^r_{kd}$	Flux linkages of the field winding fd and d-axis damper winding kd, referred to the stator and transformed to the reference frame fixed with the rotor.

$\lambda_{kq}^r, \lambda_{kq1}^r, \lambda_{kq2}^r$	Flux linkages of the q-axis damper windings kq, kq1 and kq2, referred to the stator and transformed to the reference frame fixed with the rotor.
$\lambda_{qs}^r, \lambda_{ds}^r, \lambda_{0s}$	Flux linkages of the q-axis, d-axis and zero variable windings of the stator transformed to the reference frame fixed with respect to the rotor.
$\theta_r$	Position of rotor in radians.
$T/\tau_{do}^r, T/\tau_{qo}^r$	d-axis and q-axis open circuit transient time constants.
$T/\tau_d^r, T/\tau_q^r$	d-axis and q-axis short-circuit transient time constants.
$\omega_b$	Base frequency.
$\omega_n$	Undamped natural frequency.
$\omega_r$	Rotation speed of rotor in rad/s.
$\xi$	Damping ratio.



# 1. Introduction

## 1.1. Background to Power System Stability Problem

Power system stability generally refers to that property of the power system that enables it to remain in a steady-state operating condition under normal circumstances, and following a transient disturbance, the ability to regain an acceptable state of equilibrium. In essence the stability problem is one of maintaining synchronous operation, and as a power system seldom consists of a single generating unit, the problem becomes one of keeping all the interconnected machines in synchronism [1]. This aspect of stability is thus closely related to generator rotor angles and power-angle relationships [1].

The complex problem of maintaining stability has been a challenge confronting engineers for almost 100 years and was first highlighted by C.P. Steinmetz [8] in 1920. Five years later the first field tests were conducted on a practical power system [2].

Before the advent of the computer, slide rules and mechanical calculators were the order of the day. Without the modern-day computing power, methods of analysis had to be kept simple. Initially these came in the form of the 'two-machine' systems. Steady state and transient stability were treated separately. As it became economically viable to interconnect formerly independent systems, the complexity of the required network analysis also increased. However, the development of the network analyzer in the 1930's provided a means with which to compute current, voltage and power anywhere within a given model of a power network containing numerous interconnected machines [1].

At this stage the focus was more on the network and the generator modelling aspect was overlooked. The generator was thus modelled as a constant voltage source behind a constant reactance. In many cases the load was also modelled as constant impedance. This was primarily due to the fact that computational tools at that stage were rather limited in their capability to solve differential equations.

The response of voltage regulators was gradually improved to minimize steady state monotonic instability (first swing instability). Although successful, this introduced the problem of oscillatory instability, due to the lack of damping of power swings. These trends demanded the development of enhanced analytical tools to handle the rapidly increasing complexity of the synchronous generator model [1].

The 1950's ushered in a new era in computational power, namely the digital computer. This offered a vastly superior analytical tool which enabled further improvements in the modelling of the system, both in terms of the size of the network and the detail thereof. During the 1960's power networks became increasingly interconnected due to economic considerations and improvements in stability became evident due to mutual assistance factor in terms load sharing by other interconnected stations in the event of a station going off line. However this also increased the complexity of the stability problem and in some cases magnified the consequences of unstable behaviour. This fact came to the fore in 1965 with the culmination of events leading to the northeast (USA) blackout, which brought the problem of power system stability into the open for all to see [1].

Since the 1960's the focus has shifted to transient stability. Present-day simulation packages provide powerful tools with which to carry out detailed network analysis. This has inspired increased research to improve model topologies and obtain the associated model parameters through testing and parameter estimation methodologies. This project addresses a number of the aspects involved in the determination of model parameters for synchronous machines for system stability applications. More specifically, the work focuses on the development of analytical tools for the processing of field measurement data and the estimation of specific parameters using measured machine responses.

## 1.2. Project Motivation

Figure 1.1 gives a high level outline of the various stages of the parameter estimation process based on measured responses of synchronous generators. The process can be split into three distinct stages. Collection of field data for appropriate test conditions and using suitable dedicated instrumentation represents the first stage, processing of such data then follows, which ultimately provides a link to the third and final stage entailing the analysis and modelling of the machine.

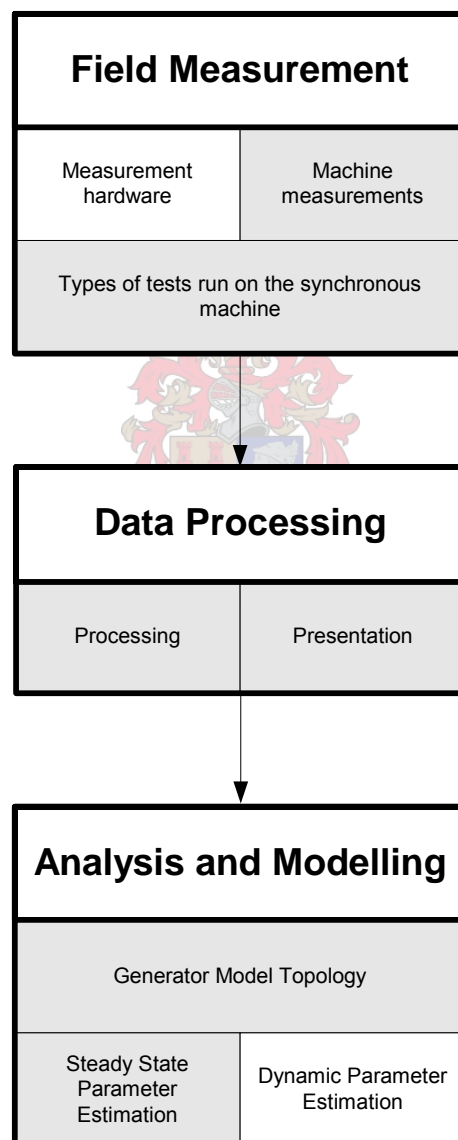


Figure 1.1 Parameter Estimation Flowchart



Each of the three stages consists of a number of segments, as shown in the flowchart of Figure 1.1. The blocks marked in grey represent topics dealt with in this thesis, while those unmarked are subjects for further research.

### 1.2.1. Field Measurements

Field measurements form the basis of all work involved in the determination of machine parameters. Electromagnetic interference with respect to measurements can be a problem due to the fact that large magnetic fields are present near the machine which could influence measurement accuracy. With this in mind the initial step would be to acquire noise-free data, and to this end an optically isolated measurement probe is needed. The development of such a device has already been undertaken and was used to great success in a parallel project run by Eskom, where data acquired from these tests was used in sections of this project. The hardware aspect solves the first issue; the second issue raises the question of what type of measurements one would need for modelling purposes. One would like to observe as many of the key indicators in the machine as possible such as terminal voltage and current, in order to fully understand the overall dynamics of the generator.

The machine measurements are thus crucial to the entire process and the most important of the group are listed below:

- Generator terminal voltages (phase to neutral voltages): The terminal voltages are important for two reasons; firstly it is used in the cost function for the estimator and secondly it is used together with the current to calculate and verify the machine power set points.
- Generator terminal currents (line currents): – adds the same value, measurement wise as does the terminal voltage.
- Field current, voltage: – the field current also performs the same function as the field voltage however together with the field voltage one can perform an estimation of the rotor parameters.
- Speed: The instantaneous speed of the machine indicates how well the governor control is working, it also allows one to dynamically track the frequency for any frequency dependent calculations
- Gate position: This is the primary measurement of the machine governor control response and gives an indication of mechanical power input.

The setpoints of the machine are also important, but are generally read off the machine's own measurement panels. Thus the voltage and current data serve to verify this reading.

Once the hardware is set up and the observation points chosen the final step in the measurement process involves the type of tests that the machine will be subjected to. For the time domain estimation, one is essentially observing machine response to fluctuations in load. In a parallel project run by Eskom the following set of tests were done on a 250 MVA hydro generator at Palmiet Pump Storage:

- Initial load ramp to an active power set point.
- With Power System Stabiliser (PSS) disengaged/engaged MVAR step up/down.
- Excitation limiter check, by manually adjusting the excitation level.

With the Eskom project conducted in Vietnam, similar data was acquired with similar types of tests where certain control systems were either enabled or disabled during a load shed, which primarily consisted of VAR rejection. Rejecting active load can affect system frequency; however reactive load rejection only involves voltage regulation.

### 1.2.2. Data Processing

The acquisition of dynamic response data for a unit under test typically requires some type of multi-channel acquisition device which will capture a certain time window and save the data in a predetermined format. A Yokogawa Model DL716 recorder was used for the field measurements at the Palmiet Hydro station and for twelve units tested in the Vietnamese power system. This yielded a large amount of raw data in a rather foreign file format translated into countless hours of tedious processing and scaling to reach a point that could enable basic analysis to begin.

In the typical field measurement case, numerous tests have to be carried out in order to cover all aspects of the unit's response. This generates a huge volume of raw data that requires some degree of processing in order to make it useful for presentation in reports or for use in further parameter estimation and analysis procedures. Without some type of flexible data processing software, the engineer would have to manually process each channel of every test case using an environment such as MATLAB<sup>®</sup> scripting. Thus a large part of this master's

project was devoted to the development and evaluation of a versatile data processing software tool for the above mentioned application.

The basic processing operations required for measured field data typically involve one or more of the following:

- Amplitude scaling to account for transducer gains.
- Filtering to remove noise.
- Removal of dc offsets created by the measurement instrumentation.

This application would have to be able to import various predetermined data file formats and export the processed data to different file formats so that other analysis programs such as Microsoft Excel, MATLAB<sup>®</sup> or DIgSILENT<sup>®</sup> Power Factory can import the processed results.

Figure 1.1 shows that the data processing stage also emphasizes the presentation of raw or processed data in a graphical format in order to allow conclusions to be drawn about the responses of the generator and the associated control systems. To ensure that the application is user-friendly, the tool should have a Graphic User Interface (GUI).

### **1.2.3. Modelling and Analysis**

Figure 1.1 shows that the third and final stage of the model identification process involves determination of a suitable model topology and estimation of the associated parameter values. For the purposes of the work described in this thesis, determination of the model topology amounted to correctly representing the model using existing libraries in the Simulink Power System Blockset. The model parameters can be classified as steady-state parameters or dynamic parameters respectively. Due to time constraints, the research conducted in this project focused on estimation of the steady-state parameters as suggested in Figure 1.1. Estimation of the dynamic parameters is not covered in this text and is subject for any future study.

### **1.2.4. The Parameter estimation process**

It is vital to ensure that the conditions and the generator itself are represented as realistically as possible in the simulation environment. If this is not the case the optimiser will not converge, and if it does the parameters will not be correct. The Power System Blockset in

Simulink provides a vast library from which one can create a viable network topology. However there are some shortfalls which do often hamper realistic modelling of the system, these problems are highlighted later on in the text.

Once the system is modelled as realistically as possible, the parameter identification toolbox is put to work. Parameter estimation is not the prime focus of this project, however it is needed to identify parameter values, thus through trial and error a cost function will be developed that will be adequate for the purposes of this study.

### 1.3. Conclusion

Figure 1.1 presents the road map that is generally followed in analyzing a typical generator. The overall process involves the following main components:

- Accurate measurements of the dynamic responses of the unit under field conditions.
- Importing of the recorded data to an environment which is suitable for mathematical processing and performing the initial data analysis, i.e. scaling, filtering, plotting, etc. The development of a user-friendly software application for this purpose formed the first part of the project.
- Determining an appropriate model topology and estimating the steady-state and dynamic parameters of the model from the measured responses. The development of an estimation procedure to determine the steady-state parameters of a synchronous generator formed the second part of this project.

## 2. Literature Survey

### 2.1. Power System Stability Studies

Since Thomas Edison built the first complete electrical system in 1882 [1], power systems have evolved into an industry of unrivalled proportions and complexity spanning entire continents. The robustness of a power system is measured by the ability of the system to operate in a state of equilibrium under normal and perturbed conditions. Power system stability studies investigate how power systems behave when faced with sudden changes in load, generation or system topology [5].

### 2.2. Overview of the Power System Stability Problem

This section presents an overview of the basic definitions of power system stability as outlined by Kundur [1]. The study of stability is mainly concerned with system behaviour when faced with a transient disturbance, be it a large or small perturbation. Small disturbances in the form of load changes happen on a continuous basis and the system generally compensates for the changing conditions automatically. However, the system must also be able to cope with a large scale disturbance, such as a transmission line fault or loss of generation.

The following sections briefly introduce the important characteristics of a typical generator and then explore the different forms of power system instability.

#### 2.2.1. The Basic Synchronous Generator

A synchronous machine essentially consists of two electromagnetic components, namely the field and the armature. The field is located on the rotor, which is driven by a prime mover or turbine. The field winding is supplied with a DC current, thus creating a rotating magnetic field, which induces alternating voltages in the stator winding or armature. The frequency of the voltages induced in the armature is determined by the speed of rotation and number of pole-pairs of the rotor [1].

An electromagnetic torque is produced when the stator and rotor fields attempt to align themselves. In generation mode the electromagnetic torque retards the speed of the rotor, thus mechanical power in terms of prime mover input must be applied to ensure that the angle between the stator and rotor fields remains constant. Increasing the electromagnetic torque or output power involves increasing the angle between the rotor and the stator, which implies that the prime mover must increase the mechanical input torque [1].

### 2.2.2. Power versus angle relationship

The relationship between the angle of the rotors of a generator and a motor, and the power transfer between them has a highly non-linear relationship. In fact the power transfer is proportional to the sine of the angle separating the rotors of both machines. In the generator's case the rotor leads the stator, whilst in the case of the motor the rotor will lag the stator. If the power transfer is as a result of the angle between the rotor and stator field then the power transfer will be the sine of the sum of the angles of both machines (including the impedance of the line that separates them). Figure 2.1 , shows the idealized model with both machine internal voltages behind an effective reactance. The voltage vector for the generator is represented by  $E_g$  whilst the vector for the voltage at the motor is given by  $E_m$ . The equivalent inductance on the generation side is defined as  $X_g$ , the line inductance given by  $X_l$ , whilst the motor inductance is  $X_m$ . Two other voltage vectors are also included in the figure; these represent the voltage angle before and after line impedance and are defined as  $E_{t1}$  and  $E_{t2}$  [1].

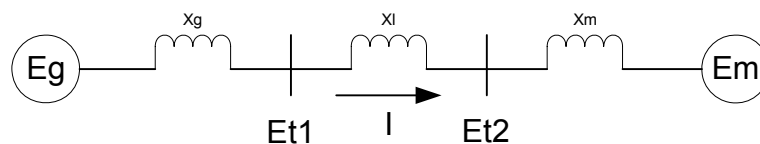


Figure 2.1 Idealized model [1]

Figure 2.2 gives the equivalent PI circuit for a lossless line, from which the  $ABCD$  parameters can be found [1].

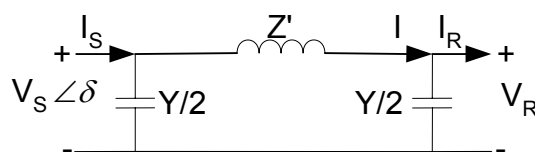


Figure 2.2 Equivalent PI circuit for lossless line [1]

The current in the series branch,  $I$  is given by the relationship

$$I = I_R + \frac{V_R Y}{2} \quad (2-1)$$

Kirchoff's voltage law (KVL) yields the following equation,

$$\begin{aligned} V_S &= V_R + \left( I_R + \frac{V_R Y}{2} \right) Z' \\ &= \left( I + \frac{YZ'}{2} \right) V_R + Z' I_R \end{aligned} \quad (2-2)$$

Using Kirchoff's current law (KCL) and summing the currents at the sending end yields

$$\begin{aligned} I_S &= I_R + \frac{V_R Y}{2} + \frac{V_S Y}{2} \\ &= YV_R \left( 1 + \frac{YZ'}{4} \right) + \left( I + \frac{YZ'}{2} \right) I_R \end{aligned} \quad (2-3)$$

Thus writing (2-2) and (2-3) in matrix format yields the  $ABCD$  parameters for the equivalent PI circuit as follows:

$$\begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} \left( 1 + \frac{YZ'}{2} \right) Z' \\ Y \left( 1 + \frac{YZ'}{4} \right) \left( 1 + \frac{YZ'}{2} \right) \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} \quad (2-4)$$

The equivalent PI circuit shown in Figure 2.2 can be used to obtain the equation for the real power delivered by a lossless line. Assume that the voltage magnitudes,  $V_R$  and  $V_S$  remain constant with the angle between the sending and receiving-end voltage denoted by  $\delta$ .

From  $KVL$ , the receiving-end current  $I_R$  is given by the relationship

$$\begin{aligned} I_R &= \frac{V_S - V_R}{Z'} - \frac{V_R Y}{2} \\ &= \frac{V_S e^{j\delta} - V_R}{jX'} - \frac{j\omega C' l}{2} V_R \end{aligned} \quad (2-5)$$

The complex power  $S_R$  delivered to the receiving end is given by

$$\begin{aligned}
S_R &= V_R I_R^* \\
&= V_R \left( \frac{V_S e^{j\delta} - V_R}{jX'} \right)^* + \frac{j\omega C' l}{2} V_R^2 \\
&= V_R \left( \frac{V_S e^{-j\delta} - V_R}{-jX'} \right) + \frac{j\omega C' l}{2} V_R^2 \\
&= \frac{jV_R V_S \cos \delta + V_R V_S \sin \delta + -jV_R^2}{X'} + \frac{j\omega C' l}{2} V_R^2
\end{aligned} \tag{2-6}$$

Since the line is lossless  $P_R = P_S$  and thus the real power P delivered is given by the relationship

$$P = P_S = P_R = \text{Re}(S_R) = \frac{V_R V_S}{X'} \sin \delta. \tag{2-7}$$

Figure 2.3 shows the phasor diagram depicting the load angle between the generator and motor [1]. The max power delivered will be at  $\delta = 90^\circ$ . Figure 2.1 shows the idealized model which has its equivalent phasor diagram displayed below.

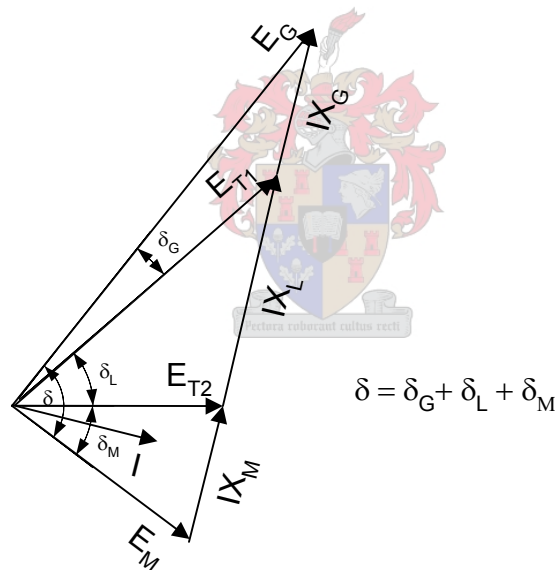


Figure 2.3 Phasor diagram of the power transfer characteristics of a two-machine system [1].



### 2.2.3. Small signal stability

Small signal stability refers to the ability of a power system to maintain synchronism for small disturbances such as variations in load or generation [1]. An example of small signal instability is increasing oscillation of the rotor due to a lack of damping torque [1]. Figure 2.4 gives a breakdown of the various types of oscillations that one can typically encounter [1].

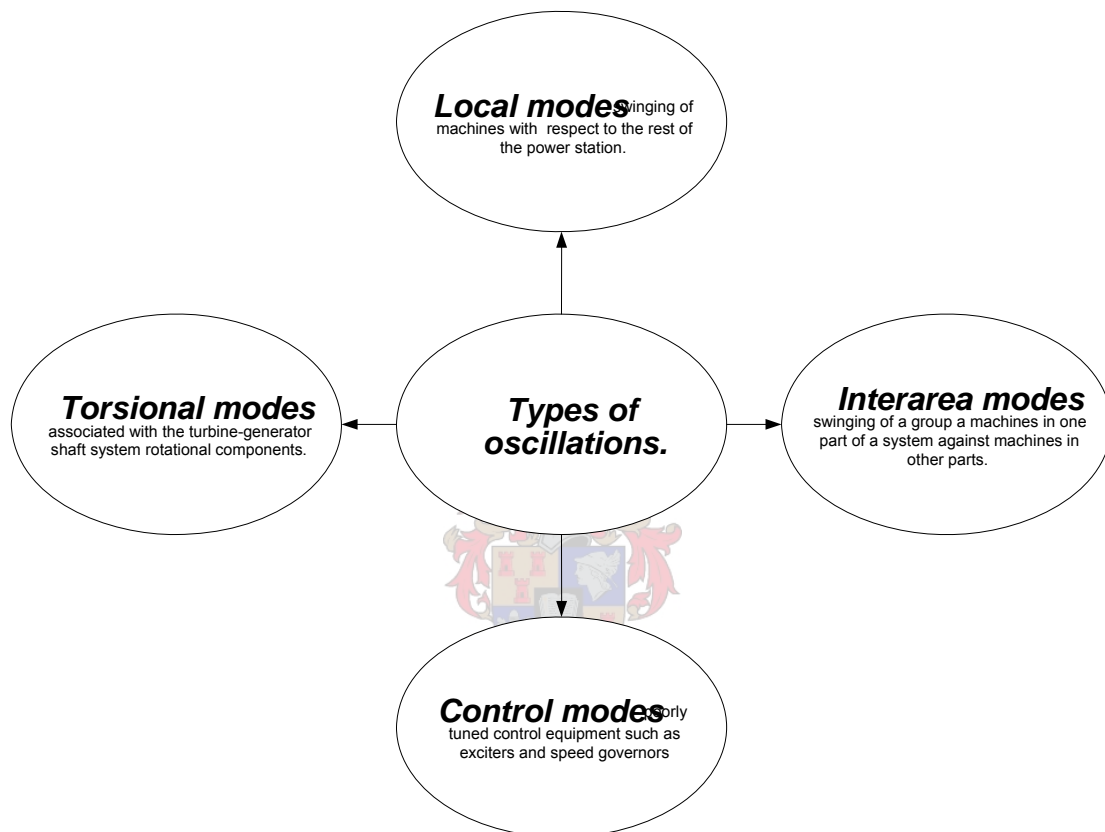


Figure 2.4 Types of oscillations

### 2.2.4. Transient stability

Severe disturbances usually in the form of transmission line faults (phase-to-phase shorts etc) induce severe changes in the rotor angle of the machine. Due to the non-linear power-angle relationship the stability of a system will not only depend on its initial operating point but also the severity of the disturbance, and when stability is finally maintained the post-fault steady state is usually different from its pre-disturbance point of operation [1].

Figure 2.5 illustrates the machine behaviour under a transient disturbance. The machine regains stability in case 1, while in the second case the machine exhibits ever increasing oscillatory behaviour, mainly due to the machine lacking adequate damping torque [1]. The final case exhibits a genuine pole slip due to the machine not having enough synchronising torque; this is often referred to as first-swing instability [1].

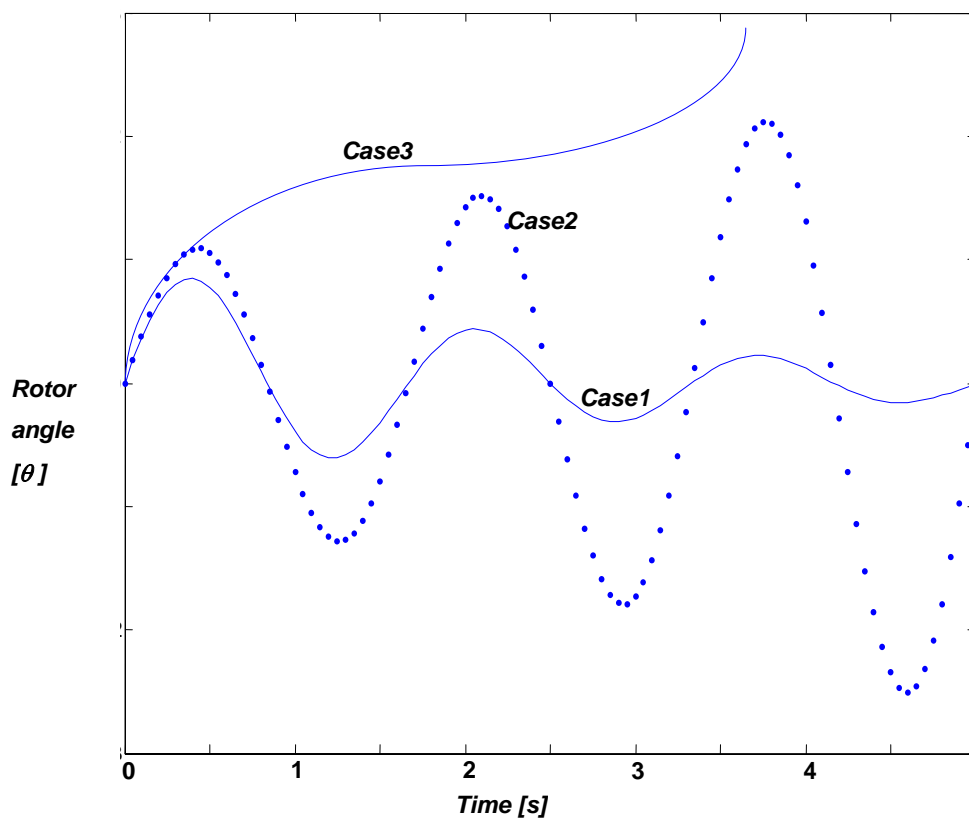


Figure 2.5 Rotor angle responses to a transient disturbance [1].

### 2.3. The Synchronous Machine

This section gives an overview of the synchronous machine construction and also briefly presents theory, mainly from the work of Krause *et al.* [5], concerning the nature of the MMF<sup>1</sup> waveform that is essentially the key to how induction and synchronous machines operate. Finally, the relevant mutual and self inductance characteristics are discussed.

As stated before a synchronous machine comprises of two essential elements, namely the field and the armature. The field is essentially a dc-excited winding driven by a prime mover. This creates a rotating magnetic field, which induces alternating voltages in the stator (armature).

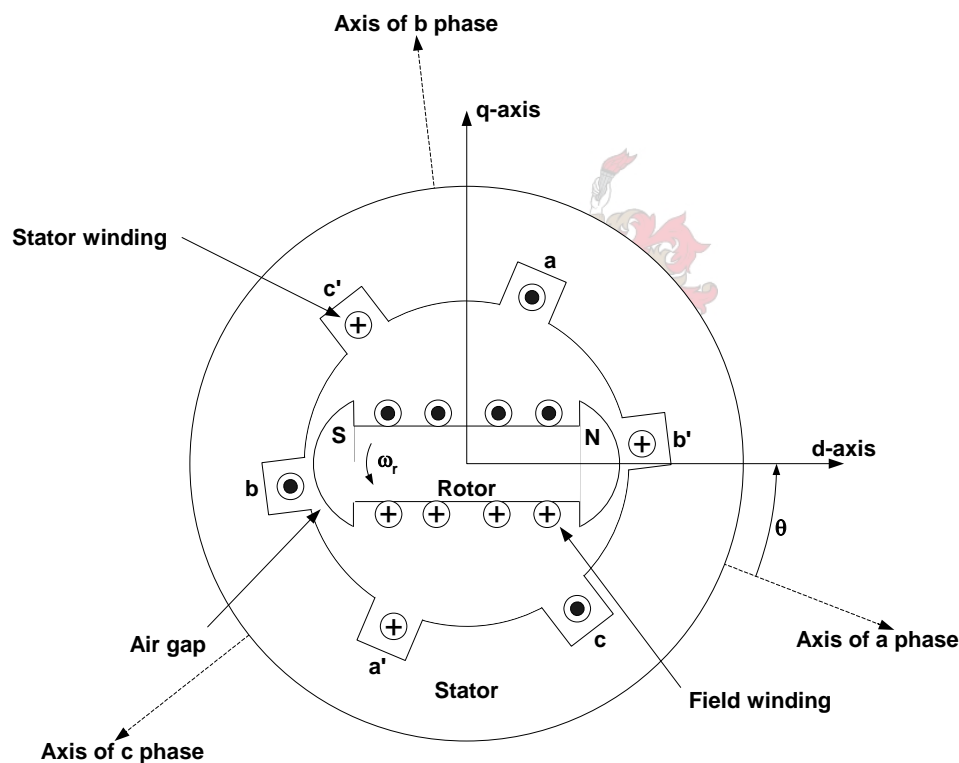


Figure 2.6 Two pole, three phase salient pole machine [1].

<sup>1</sup> The reader is referred to Appendix B for a formal overview of MMF waveforms as outlined by Krause *et al.* [6].

The stator consists of identical windings mechanically displaced 120 electrical degrees from one another, which ensures that a balanced three-phase voltage output is obtained [2]. Figure 2.6, on the previous page shows a cross sectional view of a salient pole machine with one pair of field poles. The axes of phases a, b and c represent the magnetic field induced by the current flowing through the stator windings [1].

When balanced three phase currents are flowing in the stator windings, a magnetic field is rotating at synchronous speed in the air gap between the rotor and armature. The magnetic field due to the rotor is also rotating in this region.

In order to produce steady torque the rotor must be revolving at synchronous speed [1]. Equation (2-8) expresses the speed of the machine in terms of the number of poles and the frequency of the stator currents:

$$n_s = \frac{120 f}{p_f} \quad (2-8)$$

where  $n_s$  denotes the speed in *rev/min*,  $p_f$  denotes the number of field poles and  $f$  is the frequency of the stator currents [1].



### 2.3.1. The Rotor

There are two basic rotor structures used, the choice of which depends upon the speed of the machine. An alternator designed with a high speed prime mover such as a steam turbine will entail a rotor wound for 2 to 4 poles, however for a hydraulic generator the number of poles can range from 6 to 60 [2]. For low speed machines a large number of poles are required to produce rated frequency, thus a salient pole machine is ideal for this situation.

For high speed drives the salient pole structure is mechanically too weak to withstand the centrifugal forces and mechanical stresses developed at speeds around 3600 rpm [4]. A more detailed explanation on the specifics of the various rotor assemblies can be found in the work of Kundur [1]. Figure 2.7 shows a cross sectional view of a 6 pole alternator [1], and indicates the directions of the magnetic field established by the insulated windings.

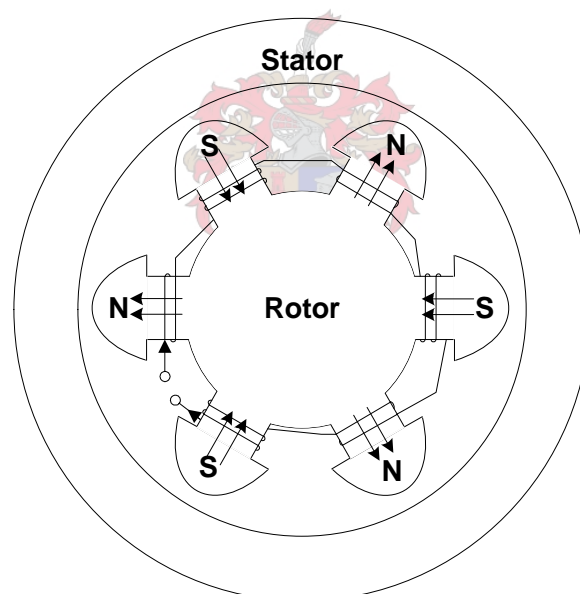


Figure 2.7 Six pole salient pole machine [1].

### 2.3.2. Magnetic Saturation

Under no-load operation and at low excitation levels the output voltage of the generator tends to have a linear relationship to the excitation current. At normal excitation levels the relationship becomes non-linear due to the saturation of the magnetic circuit.

As a typical example, Figure 2.8 shows the no-load saturation curve for the Palmiet Pump Storage Scheme unit studied in this project [3].

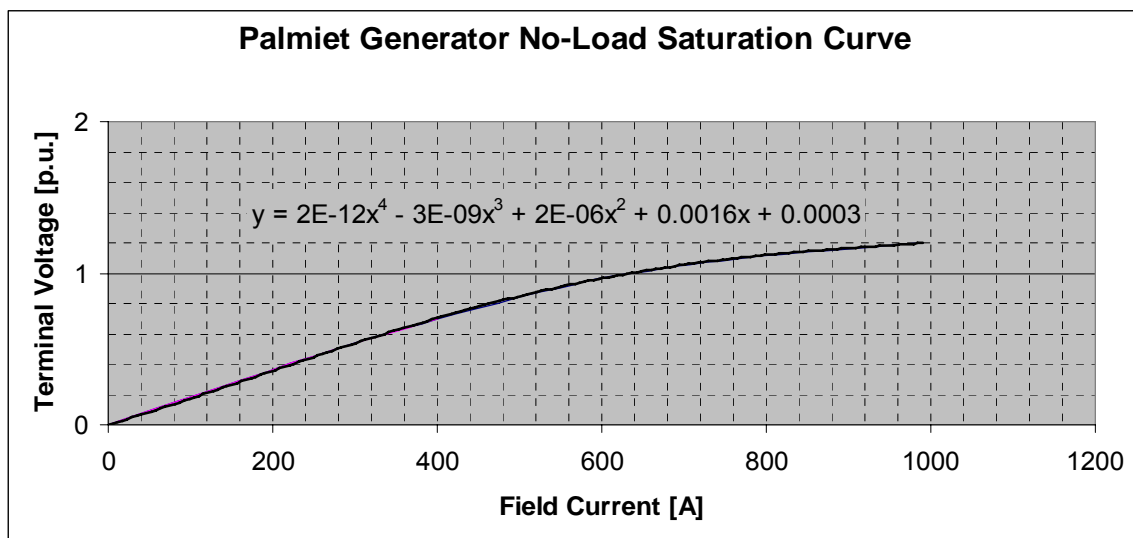


Figure 2.8 Palmiet Pump Storage Scheme no-load saturation curve [3].

One additional set of windings, known as the damper winding, is often found on the rotors of most synchronous machines. The machine does experience transients mainly during start-up and abnormal operating conditions.

### 2.3.3. Voltage and Torque Equations in Machine Variables

According to Krause et al. [5], the equations derived for a two pole, three phase, wye-connected, salient pole machine can be used to predict the behaviour of most synchronous machines. This specific machine configuration is displayed in Figure 2.9.

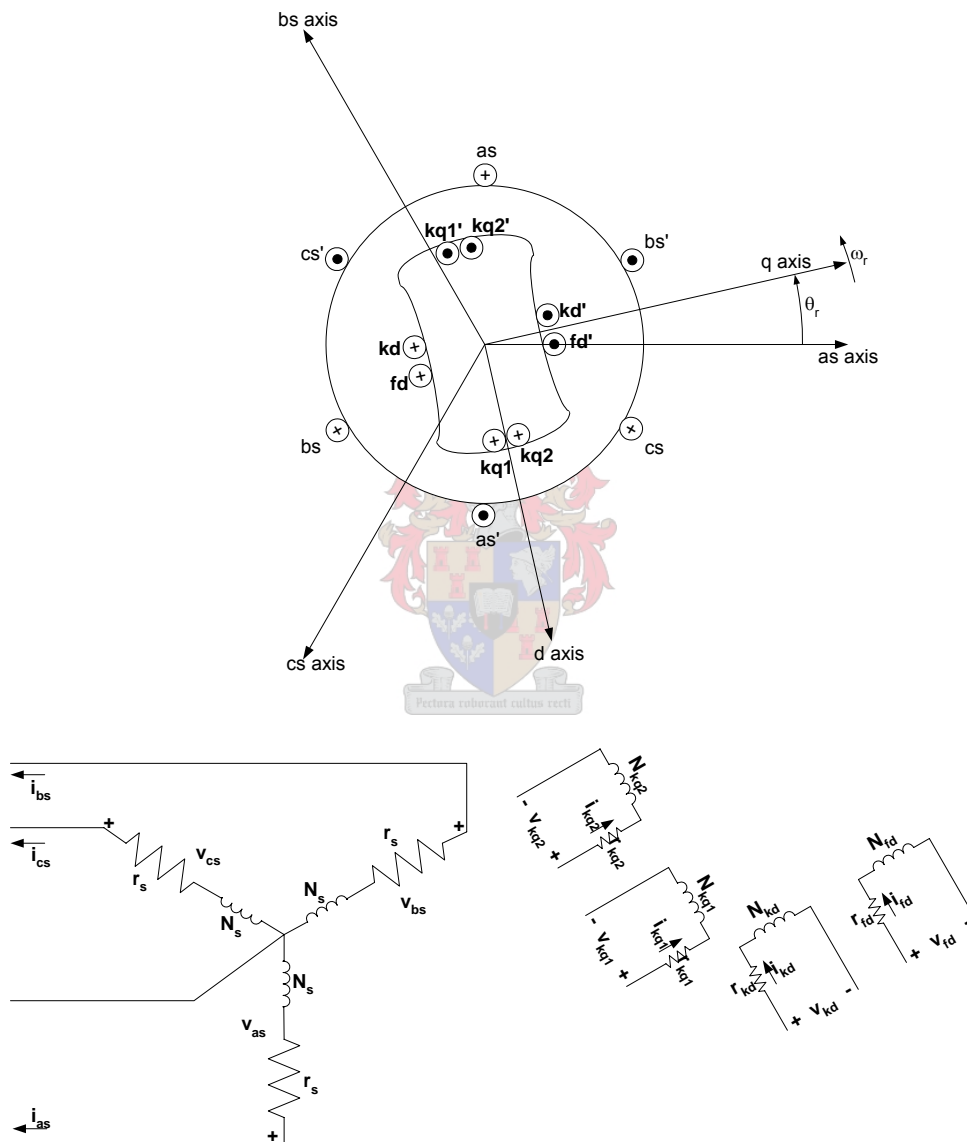


Figure 2.9 Mechanical and electrical representations for a two pole, three phase, wye connected, salient-pole synchronous machine [5].

Figure 2.9 shows the same 2-pole, wye-connected salient-pole machine that has been used in previous discussions, however the rotor winding has become slightly more complicated with the inclusion of damper windings. As before the stator windings are assumed to be identical

sinusoidally distributed windings and are mechanically displaced  $120^\circ$ , with  $N_s$  equivalent turns and resistance  $r_s$ . The field winding has  $N_{fd}$  equivalent turns with resistance  $r_{fd}$ . There is a single damper winding in the same magnetic axis as the field winding with  $N_{kd}$  equivalent turns with resistance  $r_{kd}$ . The magnetic axis of the field winding is known as the direct axis (d-axis) and the magnetic axis set  $90^\circ$  ahead of the d-axis is known as the quadrature axis (q-axis). The convention used by Krause *et al.* [5] is implemented in MATLAB<sup>®</sup>, thus it will be adopted in this text.

Two damper windings are located ahead of the magnetic axis of the field winding (d-axis), with  $N_{kq1}$  and  $N_{kq2}$  equivalent turns and resistances  $r_{kq1}$  and  $r_{kq2}$  respectively. Once again the assumption is that the rotor winding distribution is also sinusoidal. The current convention for a generator is positive current flowing out of the terminals of the stator. Thus the voltage equations in machine variables can be expressed as [5]

$$\mathbf{v}_{abc} = -\mathbf{r}_s \mathbf{i}_{abc} + p \boldsymbol{\lambda}_{abc} \quad (2-9)$$

where

$$(\mathbf{v}_{abc})^T = [v_{as} \quad v_{bs} \quad v_{cs}] \quad (2-10)$$

denotes the vector of terminal voltages and

$$(\mathbf{i}_{abc})^T = [i_{as} \quad i_{bs} \quad i_{cs}] \quad (2-11)$$

denotes the vector of three terminal currents, while

$$(\boldsymbol{\lambda}_{abc})^T = [\lambda_{as} \quad \lambda_{bs} \quad \lambda_{cs}] \quad (2-12)$$

denotes the vector of the three flux linkages of the stator windings. The subscripts as, bs and cs denote the variables associated with the three phases respectively, whilst the subscript s denote that the variables are fixed with respect to the rotor [7]. Finally the stator resistance is a diagonal square matrix expressed as [7]

$$R_s = \text{diag}[r_s + r_s + r_s] \quad (2-13)$$

Let subscript r denote variables fixed with respect to the rotor, while subscripts kq1, kq2, fd, and kd denote variables associated with rotor windings kq1, kq2, fd and kd [7].

The voltage equation describing the rotor windings may be expressed in the form

$$\mathbf{v}_{qdr} = \mathbf{r}_r \mathbf{i}_{qdr} + p \boldsymbol{\lambda}_{qdr} \quad (2-14)$$

where



$$\left( \mathbf{v}_{qdr} \right)^T = \left[ \mathbf{v}_{kq1} \quad \mathbf{v}_{kq2} \quad \mathbf{v}_{fd} \quad \mathbf{v}_{kd} \right] \quad (2-15)$$

denotes the vector of the four rotor voltages, and

$$\left( \mathbf{i}_{qdr} \right)^T = \left[ \mathbf{i}_{kq1} \quad \mathbf{i}_{kq2} \quad \mathbf{i}_{fd} \quad \mathbf{i}_{kd} \right] \quad (2-16)$$

denotes the vector of the four rotor currents. The vector of flux linkages of the rotor windings may be expressed as

$$\left( \boldsymbol{\lambda}_{qdr} \right)^T = \left[ \boldsymbol{\lambda}_{kq1} \quad \boldsymbol{\lambda}_{kq2} \quad \boldsymbol{\lambda}_{fd} \quad \boldsymbol{\lambda}_{kd} \right]. \quad (2-17)$$

The resistance matrix for the rotor circuit is given as

$$\mathbf{R}_r = \mathit{diag} \left[ r_{kq1} \quad r_{kq2} \quad r_{fd} \quad r_{kd} \right] \quad (2-18)$$

In keeping with the current convention used in the work of Krause *et al.* [5] the positive as, bs and cs axes are drawn in the direction of negative flux linkages relative to the assumed positive direction of stator currents. Thus the flux linkage equations are given as

$$\begin{bmatrix} \boldsymbol{\lambda}_{abcs} \\ \boldsymbol{\lambda}_{qdr} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_s & \mathbf{L}_{sr} \\ (\mathbf{L}_{sr})^T & \mathbf{L}_r \end{bmatrix} \begin{bmatrix} -\mathbf{i}_{abcs} \\ \mathbf{i}_{qdr} \end{bmatrix}. \quad (2-19)$$

The component of the flux linking the stator winding due to the stator currents is associated with inductance matrix  $\mathbf{L}_s$  whilst the component of the flux linkage due to the current circulating in the rotor winding is associated with the  $\mathbf{L}_{sr}$  inductance matrix. Similarly the flux linkage of the rotor circuit due to the current in the stator is associated with the transpose of the  $\mathbf{L}_{sr}$  inductance matrix, and the flux linkage due to currents in its own windings is associated with the  $\mathbf{L}_r$  inductance matrix. The formulation of the above mentioned inductance matrices is given in Appendix C.

The final step in the formulation of the voltage equations involves a change of variables which reduces the complexity of the voltage equations. Park's transformation is a key factor in the modelling of synchronous machines; in fact it is critical to the entire process in that it greatly simplifies the equations that describe the machine. Voltage equations describing the machine are filled with time-varying inductances due to in part the rotor and stator's relative motion and the varying magnetic reluctance so prevalent in salient pole machines. R.H. Park introduced an approach which effectively eliminated all time-varying parameters known as Park's transformation [6].

Thus a change of variables from the stationary circuit elements (three phase variables of the stator) to a reference frame fixed with respect to the rotor may be expressed in the form [6].

$$\mathbf{f}_{qd\theta s}^r = \mathbf{K}_s^r \mathbf{f}_{abcs}^r, \quad (2-20)$$

where

$$\left(\mathbf{f}_{qd\theta s}^r\right)^T = \left[\mathbf{f}_{qs}^r \quad \mathbf{f}_{ds}^r \quad \mathbf{f}_{\theta s}^r\right] \quad (2-21)$$

denotes a vector of transformed rotor variables fixed with respect to the rotor and

$$\left(\mathbf{f}_{abcs}^r\right)^T = \left[\mathbf{f}_{as} \quad \mathbf{f}_{bs} \quad \mathbf{f}_{cs}\right] \quad (2-22)$$

denotes the vector of stator variables. The transformation vector is given by the relationship

$$\mathbf{K}_s^r = \frac{2}{3} \begin{pmatrix} \cos \theta_r & \cos\left(\theta_r - \frac{2\pi}{3}\right) & \cos\left(\theta_r + \frac{2\pi}{3}\right) \\ \sin \theta_r & \sin\left(\theta_r - \frac{2\pi}{3}\right) & \sin\left(\theta_r + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{pmatrix}, \quad (2-23)$$

where  $\theta_r$  denotes the angle of the rotor with respect to the stator. Variable  $f$  in the above relationships represents either voltage, current or flux linkage. The subscript  $s$  denotes all stationary variables, whilst the superscript  $r$  denotes those variables that have been referred to a fixed frame of reference with respect to the rotor. Using results presented by Krause et al. [5], the voltage equations for the stator windings may be expressed in the arbitrary reference frame as [7]

$$\mathbf{v}_{qd\theta s} = -r_s \mathbf{i}_{qd\theta s} + \omega \lambda_{dqs} + \frac{d\lambda_{qd\theta s}}{dt} \quad (2-24)$$

where

$$\left(\lambda_{dqs}\right)^T = \left[\lambda_{ds} \quad -\lambda_{qs} \quad \theta\right]. \quad (2-25)$$

The above relationships are subjected to the restriction being that the phase resistances must be balanced. The rotor variables are not transformed, but are expressed in the rotor reference frame. Thus, from (2-14) with appropriate turns ratio included and the superscript  $r$  denoting rotor reference frame, the rotor voltage equation becomes [5]

$$\mathbf{v}_{qdr}^r = r_r' \mathbf{i}_{qdr}^r + \frac{d\lambda_{qdr}^r}{dt}. \quad (2-26)$$

In order to obtain the voltage equations referred to the rotor reference frame, make use equations (2-24) and (2-26) and the fact that  $\omega = \omega_r$  in these equations. Therefore

$$\mathbf{v}_{qd0s}^r = -r_s \mathbf{i}_{qd0s}^r + \omega_r \lambda_{qds}^r + \frac{d\lambda_{qd0s}^r}{dt} \quad (2-27)$$

and (2-26) remains unchanged as it has already been referred to the rotor reference frame [5]. The flux linkages in equation (2-27) are defined in equation(2-25).

As it is assumed that the system is magnetically linear, the flux linkage equations can be expressed from (2-19), with transformation of stator variables to the arbitrary reference frame as done above. For more information on the different types of reference frames that one can use, the reader can consult the work of Krause *et al.* [5].

With the previously mentioned transformation taking place the flux linkage equation of (2-19) can now be expressed as

$$\begin{bmatrix} \lambda_{qd0s} \\ \lambda_{qdr}^r \end{bmatrix} = \begin{bmatrix} \mathbf{K}_s \mathbf{L}_s (\mathbf{K}_s)^{-1} & \mathbf{K}_s \mathbf{L}'_{sr} \\ \frac{2}{3} (\mathbf{L}'_{sr})^T (\mathbf{K}_s)^{-1} & \mathbf{L}'_r \end{bmatrix} \begin{bmatrix} -\mathbf{i}_{qd0s} \\ \mathbf{i}_{qdr}^r \end{bmatrix} \quad (2-28)$$

Finally Krause *et al.* [5] shows that

$$\mathbf{K}_s^r \mathbf{L}_s (\mathbf{K}_s^r)^{-1} = \begin{bmatrix} L_{ls} + L_{mq} & 0 & 0 \\ 0 & L_{ls} + L_{md} & 0 \\ 0 & 0 & L_{ls} \end{bmatrix}, \quad (2-29)$$

$$\mathbf{K}_s^r \mathbf{L}'_{sr} = \begin{bmatrix} L_{mq} & L_{mq} & 0 & 0 \\ 0 & 0 & L_{md} & L_{md} \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2-30)$$

and

$$\frac{2}{3} (\mathbf{L}'_{sr})^T (\mathbf{K}_s^r)^{-1} = \begin{bmatrix} L_{mq} & 0 & 0 \\ L_{mq} & 0 & 0 \\ 0 & L_{md} & 0 \\ 0 & L_{md} & 0 \end{bmatrix}. \quad (2-31)$$

Writing Park's equations in expanded form yields the following [5]:

$$\mathbf{v}_{qs}^r = -r_s \mathbf{i}_{qs}^r + \omega_r \lambda_{ds}^r + \frac{d\lambda_{qs}^r}{dt}, \quad (2-32)$$

$$\mathbf{v}_{ds}^r = -r_s \mathbf{i}_{ds}^r - \omega_r \lambda_{qs}^r + \frac{d\lambda_{ds}^r}{dt}, \quad (2-33)$$

$$\mathbf{v}_{0s} = -\mathbf{r}'_s \mathbf{i}_{0s} + \frac{d\lambda_{0s}}{dt}, \quad (2-34)$$

$$\mathbf{v}'_{kq1} = \mathbf{r}'_{kq1} \mathbf{i}'_{kq1} + \frac{d\lambda'_{kq1}}{dt}, \quad (2-35)$$

$$\mathbf{v}'_{kq2} = \mathbf{r}'_{kq2} \mathbf{i}'_{kq2} + \frac{d\lambda'_{kq2}}{dt}, \quad (2-36)$$

$$\mathbf{v}'_{fd} = \mathbf{r}'_{fd} \mathbf{i}'_{fd} + \frac{d\lambda'_{fd}}{dt} \quad (2-37)$$

and

$$\mathbf{v}'_{kd} = \mathbf{r}'_{kd} \mathbf{i}'_{kd} + \frac{d\lambda'_{kd}}{dt}. \quad (2-38)$$

The expanded form of the flux linkages is obtained by substituting equations (2-29) –(2-31) and (C-43) from Appendix C into (2-28). This yields the expressions [5]

$$\lambda'_{qs} = -L_{ls} \mathbf{i}'_{qs} + L_{mq} \left( -\mathbf{i}'_{qs} + \mathbf{i}'_{kq1} + \mathbf{i}'_{kq2} \right), \quad (2-39)$$

$$\lambda'_{ds} = -L_{ls} \mathbf{i}'_{ds} + L_{md} \left( -\mathbf{i}'_{ds} + \mathbf{i}'_{kd} + \mathbf{i}'_{fd} \right), \quad (2-40)$$

$$\lambda_{0s} = -L_{ls} \mathbf{i}_{0s}, \quad (2-41)$$

$$\lambda'_{kq1} = -L'_{lkq1} \mathbf{i}'_{kq1} + L_{mq} \left( -\mathbf{i}'_{qs} + \mathbf{i}'_{kq1} + \mathbf{i}'_{kq2} \right), \quad (2-42)$$

$$\lambda'_{kq2} = -L'_{lkq2} \mathbf{i}'_{kq2} + L_{mq} \left( -\mathbf{i}'_{qs} + \mathbf{i}'_{kq1} + \mathbf{i}'_{kq2} \right), \quad (2-43)$$

$$\lambda'_{fd} = -L'_{lfd} \mathbf{i}'_{fd} + L_{md} \left( -\mathbf{i}'_{ds} + \mathbf{i}'_{fd} + \mathbf{i}'_{kd} \right) \quad (2-44)$$

and

$$\lambda'_{kd} = -L'_{lkd} \mathbf{i}'_{kd} + L_{md} \left( -\mathbf{i}'_{ds} + \mathbf{i}'_{fd} + \mathbf{i}'_{kd} \right). \quad (2-45)$$

Many of the equations put forward in this chapter are implemented in the model of the synchronous machine available in the Simulink Power Systems Blockset. For information on where each particular equation fits, the reader is referred to Appendix E.

## 2.4. Estimation of Synchronous Generator Parameters

### 2.4.1. Introduction

Since the 1930's there has been a need to develop accurate models that represent the synchronous machine in a simulation environment. With the onset of the computer age, these models have evolved into highly complex circuits. Thus various methods to determine the equivalent electrical circuit parameters have been developed over the last couple of decades. E.C. Schaffer *et al.* [8], divides the different methods into two distinct groups, namely rotary synchronous machine modelling (RSM) and linear synchronous machine modelling (LSM). Experience with the latter is not as extensive as is the case with RSM modelling. The next section gives a brief overview of all the most relevant parameter estimation procedures.

### 2.4.2. Rotary Synchronous Machine Modelling

Park's transformation [6] is the generally accepted method by which RSM circuit models are derived. RSM's generally come in the same form as the synchronous machine model example used in this text, which consists of three phase ac stator windings, rotor dc field winding, and several damper windings. As was stated in earlier sections Park's equations transforms the three phase stator quantities into a new reference frame which moves with the rotor. The new variables are expressed in terms of direct and quadrature axes quantities. Classical machine theory uses this two-axis transformation to derive the equivalent circuits and corresponding parameters [8].

### 2.4.3. Traditional tests

Terminal measurements are conducted at or close to synchronous speed, with a test arrangement as outlined in IEEE Standard 115 (1983) [9]. Winding resistances are found by dc measurements at 20°C and corrected for the system frequency. Equivalent loss resistance is taken into account by monitoring power levels at no load and short circuit conditions [8]. Time domain response measurements are detailed in the IEEE Standard 115 [9], and the following machine characteristics must be acquired:

- Open circuit curve or no-load curve (*occ*) and the air gap line.
- The short circuit saturation curve (*scc*).

- The zero power-factor saturation characteristic (*zpc*).

From the above results, parameter  $X_d$  can be found from

$$X_d = \frac{V_{t(\text{rated})\text{occ}|I_f=I_{f_0}}}{I_{a(\text{sc})|I_f=I_{f_0}}} \quad (2-46)$$

where  $I_{f_0}$  is the field current at rated terminal voltage on the *occ*, and  $I_f$  is the rated current on the *scc*.

Parameter  $X_q$  obtained by means of the slip test [8] and  $X_1$  can be deduced by Potier's method [9]. Transient and sub-transient time constants as well and reactances are derived from the current envelope for a three phase stator short-circuits [8].

#### 2.4.3.1. DC decay tests

The transient decay of dc current in the stator and field windings can be used to estimate the machine time constants. With the field short circuited two of the wye-connected stator windings are supplied with a dc current. Using a breaker the two windings are then short circuited and the time constants are thus obtained from the measured data [9]. For more on this method consult the work of P.J. Turner *et al.* [10].

#### 2.4.3.2. Decrement methods

Decrement tests involve interrupting stator currents under various load and excitation conditions to determine the d-axis and q-axis data [9]. More on this topic can be found in the text of Dandeno [21].

#### 2.4.3.3. Canay methods

Canay [16], suggests analytical refinements to calculate rotor parameters. A characteristic reactance,  $X_c$  represents the difference between the mutual reactance between the field and damper windings,  $X_f$ , and the main axis reactance,  $X_{ad}$  [8]. Thus  $X_c$  is merely the difference between  $X_f$  and  $X_{ad}$ . This "Canay Reactance" is normally found by measuring field current variations during armature short circuit tests [8].

#### 2.4.3.4. Standstill Frequency Tests

The SSFR method aligns the rotor such that d and q axis voltages can be measured directly [12]. “Operational impedances”  $Z_d(s)$ ,  $G(s)$  and  $Z(q)$  are obtained and related mathematically to the standard machine time constants [8].

#### 2.4.4. Transient Response tests

F.P de Mello *et al.* [13] suggest a test procedure to measure the parameters of the machine that will yield reliable data. The procedure involves measurement of voltage and field current transient deviations caused by staged generator breaker trips starting from loaded conditions. For a salient pole machine the d-axis is represented with field and damper winding effects allowing for saturation, whilst the q-axis is represented with one damper winding and no saturation present. With the machine at zero active power and either importing or exporting reactive power, it is possible to ensure that flux only exists in the d-axis.

##### 2.4.4.1. D-axis Parameters

The test case for the d-axis parameters entails that the machine operates at zero active power, and by controlling the excitation system the reactive power flow can be controlled. Once at the right setpoint, the breaker is opened and the deviations in field current and terminal voltage are recorded. With the machine initially underexcited saturation effects are negligible, thus the unsaturated values of  $X_d$ ,  $X_d'$ ,  $X_d''$ ,  $T'_{do}$ ,  $T''_{do}$  can be determined [12].

Figure 2.10 shows a typical variation in the terminal voltage following a rejection of a purely reactive load. The machine is initially underexcited, thus saturation effects can be neglected. De Mello *et al.* [13] proposes that the d-axis reactances can be found from the indicated response.

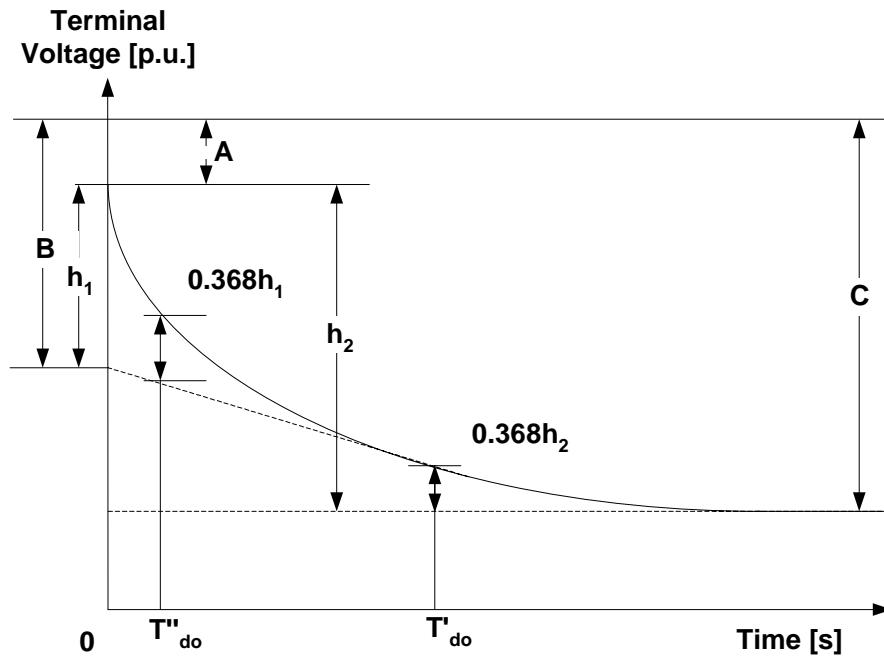


Figure 2.10 Terminal voltage response [12].

If the value of the interrupted terminal current is taken as  $i_o$ , then the d-axis reactances can be found from the following relationships:

$$X_d = \frac{C}{i_o}, \quad (2-47)$$

$$X'_d = \frac{B}{i_o} \quad (2-48)$$

and

$$X''_d = \frac{A}{i_o}, \quad (2-49)$$

where A, B and C are as indicated in Figure 2.10. The d-axis open circuit time constants can also be derived from the previous figure and the field current response as shown below in Figure 2.11 [12].



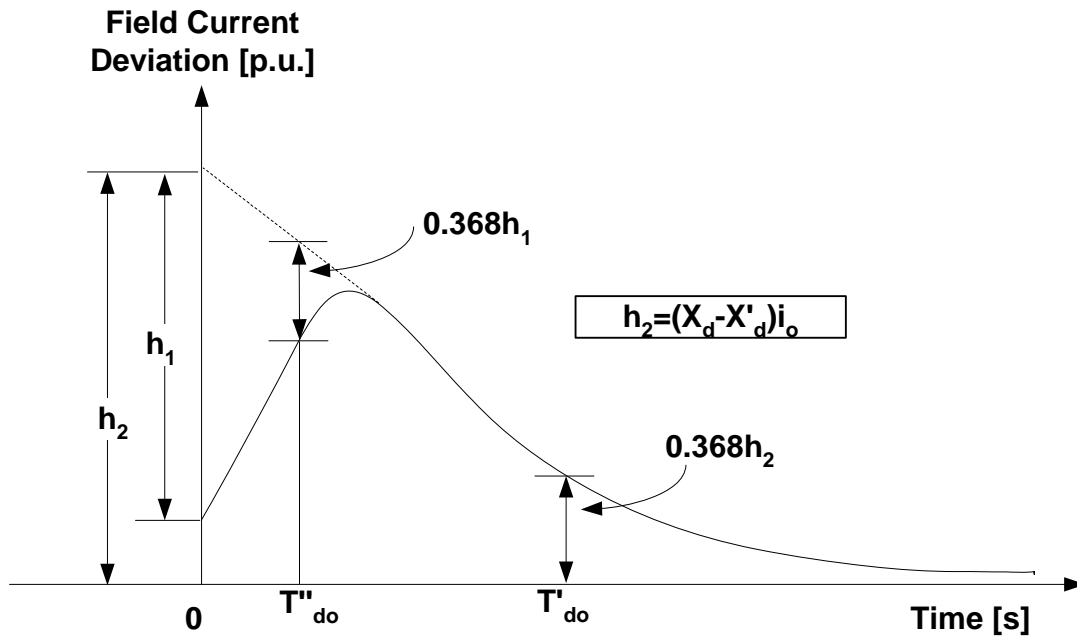


Figure 2.11 Field current deviation after rejection of a purely reactive load. Machine initially under-excited [12].

#### 2.4.4.2. Q-axis Parameters

To determine the q-axis parameters de Mello *et al.* [13] states that the ideal would be to have the armature current solely consisting of the quadrature axis component. De Mello *et al.* [13] shows with use of vectors what the loading conditions must be, however another easier way to find the desired loading condition is by running successive load rejections where the field current deviation is monitored.

The idea is to find the condition where the field current remains constant. Thus it is suggested to establish at least two loading conditions for which the deviations in field current are in opposite directions to each other. A plot of  $\Delta I_{fd}$  versus  $KVAR/i^2$  will locate the point where  $\Delta I_{fd}$  is zero. The value of  $KVAR/i^2$  at this point gives the value of  $X_q$  [13]. The value of  $KVAR/i^2$  at this point gives the value of  $X_q$  [13]. This method is illustrated in Figure 2.12.

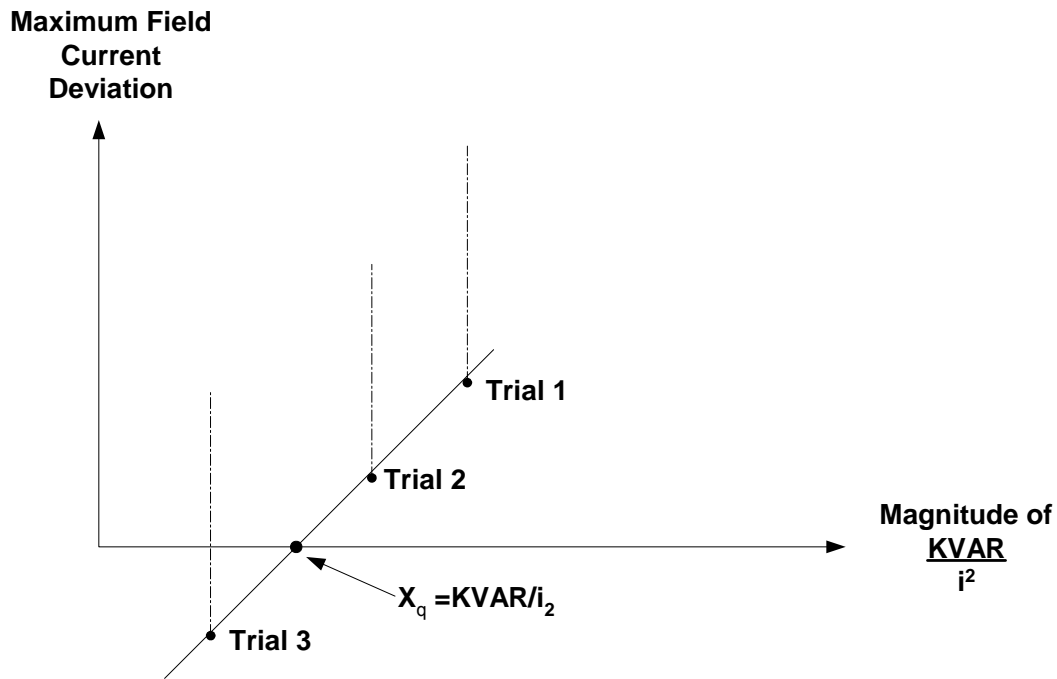


Figure 2.12 Proposed scheme to ensure that there is no d-axis component of armature current [12]. Once the appropriate loading condition has been found, the response of the terminal voltage of the machine (for the load rejection test) typically looks like that shown in Figure 2.13.

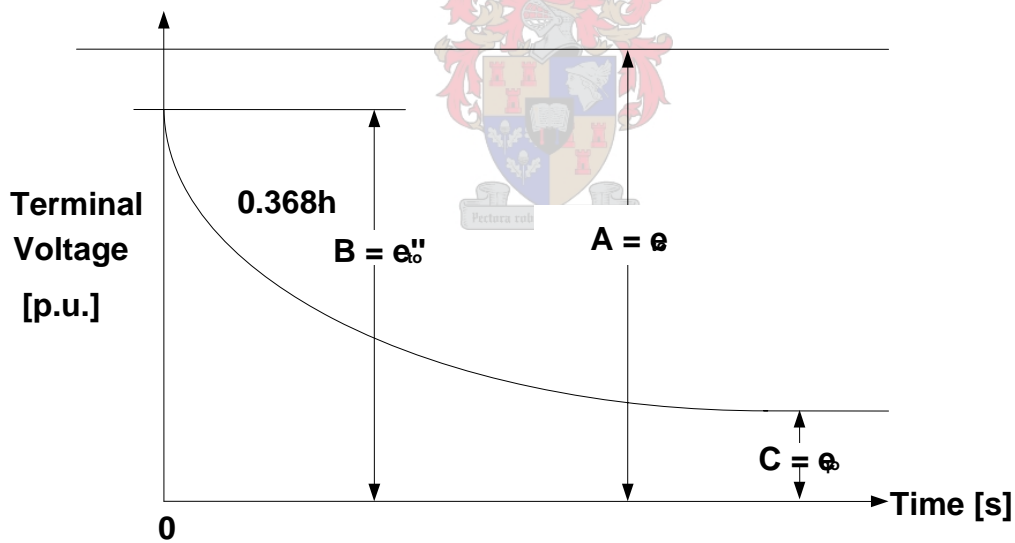


Figure 2.13 Terminal voltage variation after load rejection [12].

Using the definitions in Figure 2.13, and the geometric relationships that de Mello *et al.* [13] suggests, the synchronous and sub-transient reactances for the q-axis can be found in terms of the variables A, B and C. The expressions for  $X_q''$  and  $X_q$  and can be expressed in the form

$$X_q'' = \frac{\sqrt{A^2 - C^2} - \sqrt{B^2 - C^2}}{I_0} \tag{2-50}$$

and

$$X_q = \frac{\sqrt{A^2 - C^2}}{I_0}, \quad (2-51)$$

where  $I_0$  is the value of the interrupted current.

## 2.5. Cost Function - Least Squares Method

The method of the least squares attempts to calculate the deviation of a simulated response from a measured reference response. This is accomplished by comparing the two responses point-by-point. For a given set of values the dynamic deviation per sample is given by the relationship

$$d_n = y_n - f(x_n). \quad (2-52)$$

where  $d_n$  denotes the error per sample and  $y_n$  denotes the original field data to be matched. The equation of the curve that describes the system in question is represented by the function  $f(x)$ . Thus the best fitting curve will have the property

$$\varepsilon_{total} = \sum_{n=1}^N d_n^2 = \sum_{n=1}^N [y_n - f(x_n)]^2, \quad (2-53)$$

where  $\varepsilon_{total}$  is the total error after each estimation iteration and  $N$  is the number of samples. This error is passed back to the estimator in order for the optimiser to make further changes to the parameters of the system in order to get the minimum error.

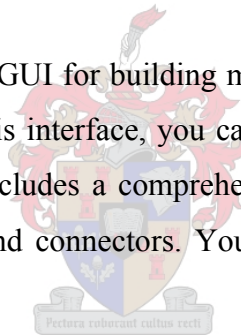
A host of different methods used to estimate machines parameters have been developed since the 1930's, this chapter has briefly touched on the various techniques and also explained the concept of the least squares method. Schaffer et. al. [8] presents a paper containing a survey, of sorts, on the various techniques used to determine machine parameters. These techniques together with the various machine models form the basis for computer analysis and simulation so prevalent in modern day industry.

### 3. The Simulink simulation environment

#### 3.1. Introduction

Simulink is a software package for modelling, simulating, and analysing dynamic systems. It supports linear and nonlinear systems, modelled in a continuous or discrete time domain. Models can be built from scratch, or existing models can be used or added to a new model. All results can be analysed either in the Simulink environment or with the use of the MATLAB<sup>®</sup> platform. Simulink allows one to model almost any system that can be found in modern day industry, in this case the generator and its control systems. One can now move away from idealised models and explore more realistic nonlinear models, factoring in friction, air resistance and any other characteristics that describe real-world phenomena. This feature comes to the fore in this thesis when rotor estimation is implemented.

For modelling, Simulink provides a GUI for building models as block diagrams, using click-and-drag mouse operations. With this interface, you can draw the models just as you would with pencil and paper. Simulink includes a comprehensive block library of sinks, sources, linear and nonlinear components, and connectors. You can also customize and create your own blocks.



After you define a model, you can simulate it, using a choice of integration methods, either from the Simulink menus or by entering commands in the MATLAB<sup>®</sup> Command Window. The menus are particularly convenient for interactive work, while the command-line approach is very useful for running a batch of simulations, in the case of this thesis M-files are used to call Simulink systems which in turn pass results back for further analysis. A prime example of this is the estimation routines that are so prevalent in this text.

One of the objectives of this project is to investigate the suitability of Simulink for simulating the dynamic responses of synchronous machines, as well as to determine the suitability of Simulink for the estimation of synchronous machine parameters. A critical eye will also be placed on the robustness of the Simulink electrical network simulation environment.

### 3.1.1. Simulink Environment

Figure 3.1 shows the workspace that one will typically encounter when designing a model for simulation purposes. When selecting a component to include in a model one can simply drag and paste the item of choice, in this example a synchronous machine has been selected from the machines library.

Once the element has been placed in the workspace the parameters of the component can easily be adjusted by just double-clicking on the item. A GUI is then activated containing all the parameter settings for the machine. Continuing in this fashion one can set up a system fairly quickly and without much trouble.

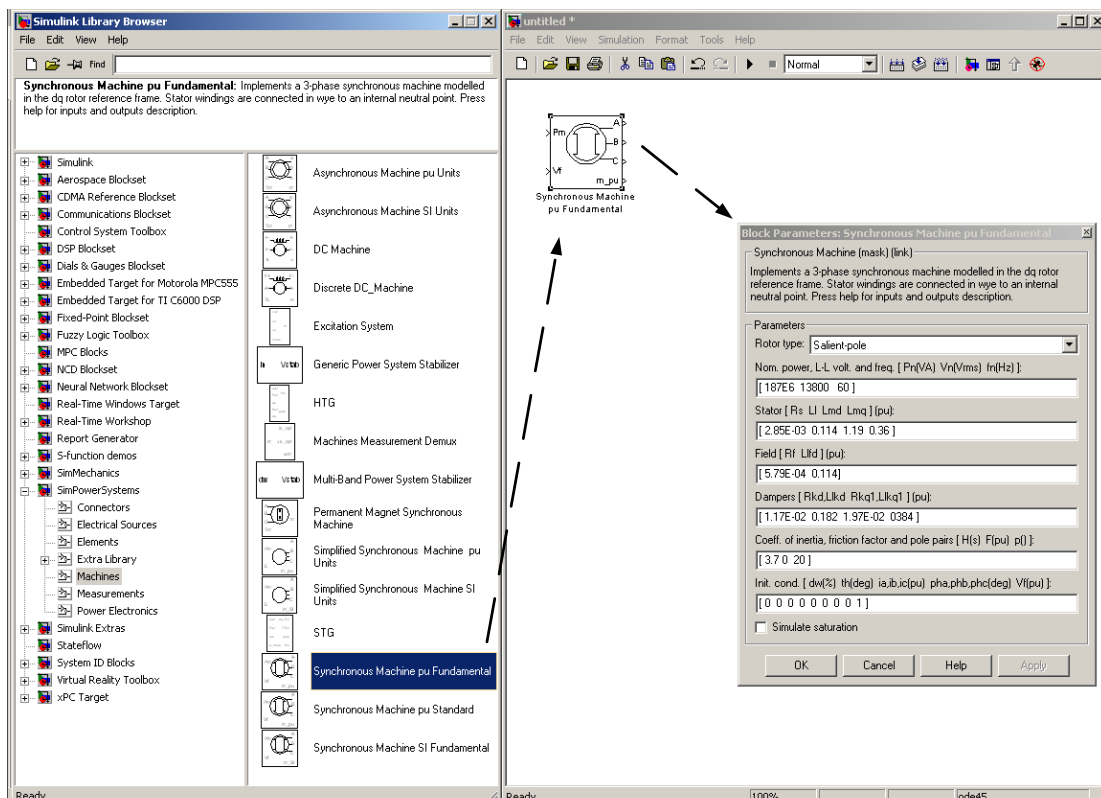


Figure 3.1 Simulink Work Environment

### 3.2. Network Topology

Figure 3.2 shows an example of a power system that has been created using the library of existing models in the Power Systems Blockset. This network topology is representative of the type of network model that is used throughout this project. This particular diagram is modelling the Palmiet Pump Storage Scheme.

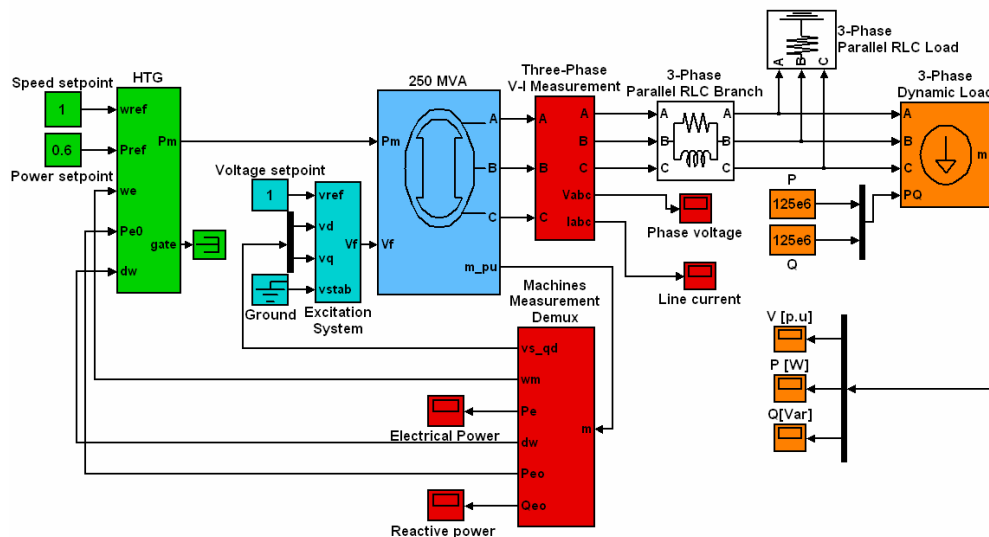


Figure 3.2 Typical Network Topology created in Simulink.

The generator is a salient pole machine and is represented by the blue block in Figure 3.2. The system is outfitted with generic system control, that is the hydraulic turbine governor system (HTG) and the automatic voltage regulator (AVR). The 250 MVA synchronous generator in Figure 3.2 is connected through a series impedance to a variable load, which acts as the infinite bus in this example.

What is interesting in this model is the active load between the series impedance and the variable load. The reason for this load is that in Simulink one cannot connect a nonlinear block in series with an inductive element due to the fact that all nonlinear blocks are modelled as current sources. Thus a resistive branch connected to ground is placed in parallel. This is crucial otherwise the solver generate an error [15]. Once the circuit is created the simulation may be started. Each time a simulation is attempted, an initialisation procedure computes the steady state model of the system and also builds the equivalent system to be run through the solver. Two functions are used to perform the entire process; the *power2sys* and *circ2ss* [15].

### 3.3. Pre-simulation initialisation

The flowchart in Figure 3.3 summarises the events happening prior to each simulation [15].

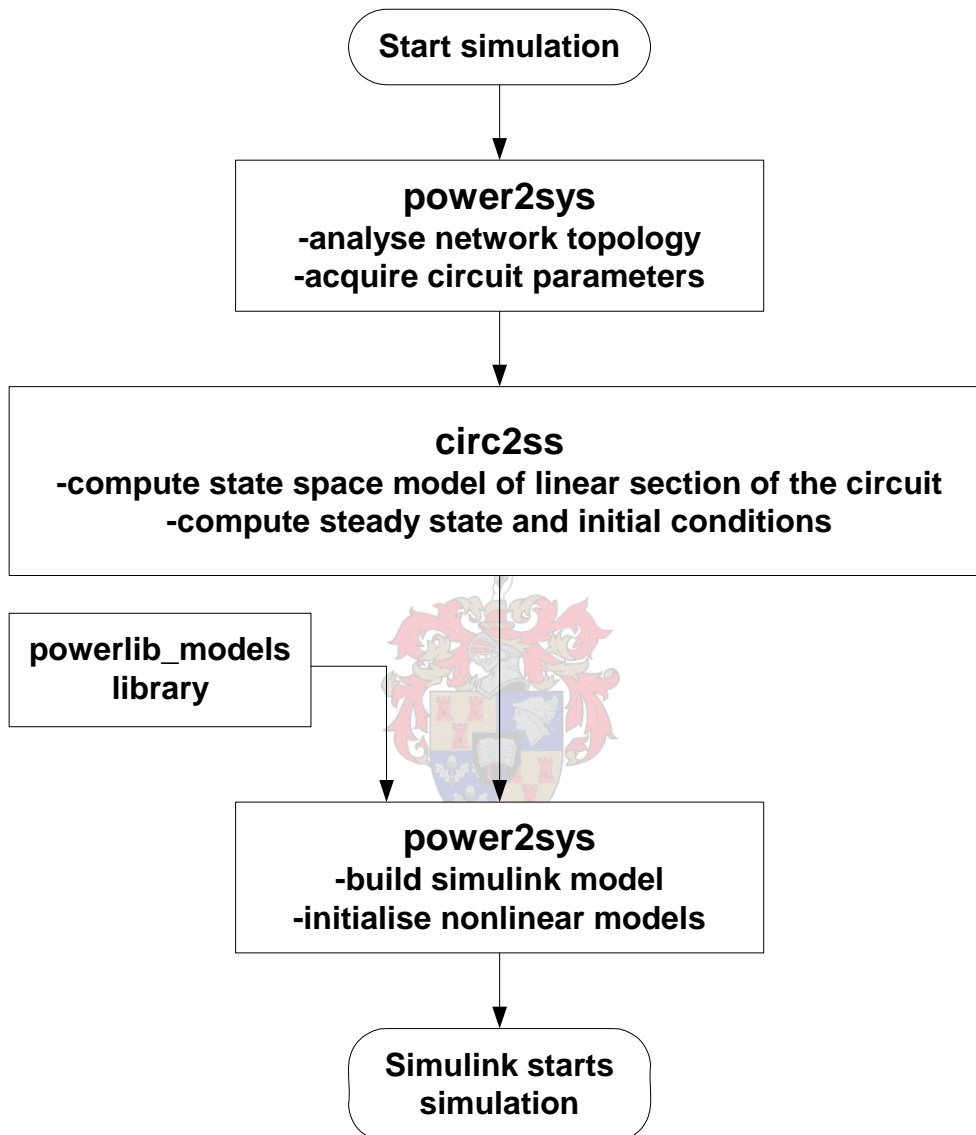


Figure 3.3 Power System Blockset Flowchart [15].

Initially the `power2sys` function sorts all the blocks, obtains parameter information and evaluates the network topology. The circuit is then separated into linear and nonlinear blocks and each electrical node is given a number. At this stage the `circ2ss` computes the state space model of the circuit and also calculates the steady state values as well as initialising the circuit.

The `power2sys` function then builds the equivalent circuit to be simulated and stores it in a measurement block. At this stage it is important that one remembers to include either a voltage, current, VI or multimeter measurement block in the model, for storage purposes. Simulink uses a state space block to represent the linear part of the circuit, whilst the nonlinear elements are replaced by their predefined equivalents, all of which can be found in the Power Systems Blockset library.

Figure 3.4 represents the interconnections between the different parts of the complete Simulink model. The nonlinear models are connected in feedback between voltage outputs and current inputs of the linear model.

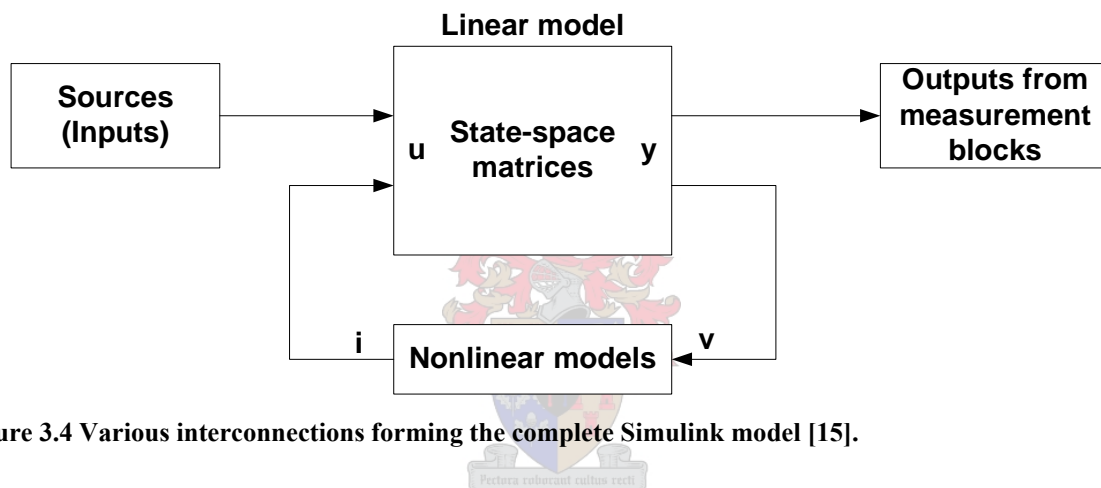


Figure 3.4 Various interconnections forming the complete Simulink model [15].

### 3.4. Nonlinear Models

The continuous section of the power systems library consists of current sources simulating continuous machine models, surge arresters and distributed parameter lines. This library typically looks like that shown in the Figure 3.5 [15]. A number of the blocks from this library were used to construct the model given in Figure 3.2, such as the HTG to model a typical governor found in hydro plants. The automatic voltage regulator was also taken from this library and used to control the synchronous machine.



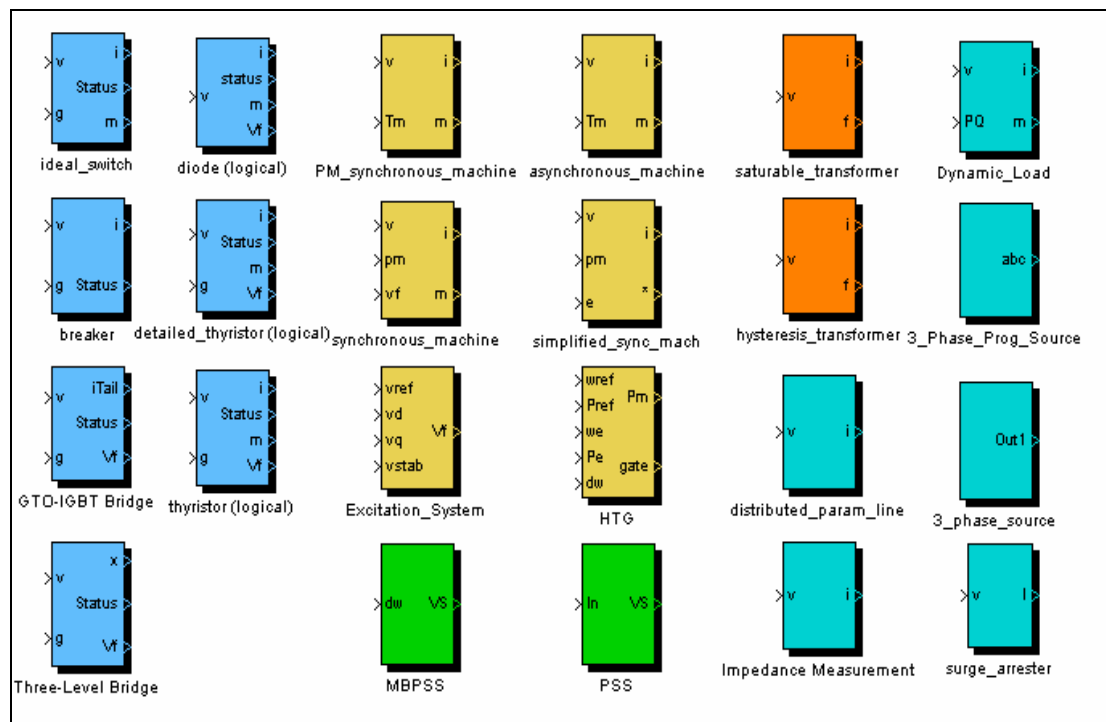


Figure 3.5 Library containing nonlinear models [15].

The nonlinear models use the outputs of the linear state space models, which are in the form of voltage signals, and return current signals in the feedback loop shown Figure 3.4. As mentioned previously, all nonlinear blocks are modelled as current sources and thus cannot be connected in series with inductances or left with their terminals open. To avoid errors one can connect a large resistance in parallel with the inductance or across the terminals of the machine [15]. This in fact represents one of the properties of the software that was found to be particularly cumbersome and gave rise to a number of difficulties.

### 3.5. Simulink Solvers

#### 3.5.1. Solver type

The solver that is used extensively in this project is known as the *ode23tb* solver. This particular solver is designed for solving stiff problems, i.e. cases where the responses can change on a time scale that is very short compared to the interval of integration, but the responses of interest changes on a much longer time scale [15]. This is typical of the topologies studied in this investigation and choosing the correct algorithm became critical to ensure successful simulation of test cases. Solvers that are not designed for stiff problems are ineffective where values change slowly over time as they use time steps small enough to resolve the fastest change. This specific solver implements a TR-BDF2 formula and at this

point the interested reader is referred to either Bank *et al.* [13] or Shampine *et al.* [14] who cover this topic in more detail.

### 3.5.2. Error tolerances

All solvers in Simulink use standard local error control techniques to monitor the error at each time step [15]. Following each time step the state values are calculated and then the estimated error of these states is also computed, known as the local error [15]. The local error is then compared to the acceptable error, which is a function of the relative (RTOL) and absolute (ATOL) tolerances. If the local error is greater than the acceptable error, the solver reduces the step size and repeats the entire exercise.

The relative tolerance is a measure of the error relative to the size of each state. The default is  $1e-3$  which translates to 0.1% of the state's actual value [15]. The ATOL is the threshold error and gives the effective error tolerance as the state approaches zero. The acceptable error is a combination of the two and is given in [15] as

$$e_i = \max(\text{rtol} \times |x_i|, \text{atol}_i), \quad (3-1)$$

where the error is for the  $i^{\text{th}}$  state and  $x_i$  is the value of this state. Figure 3.6 explains the entire procedure.

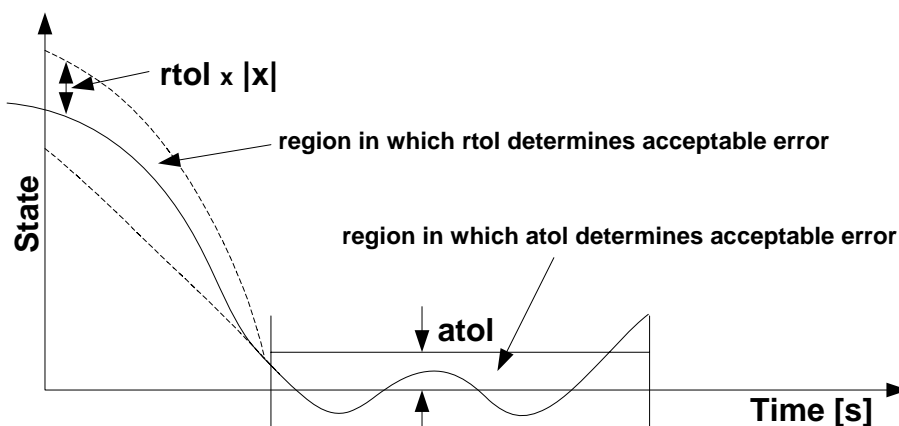


Figure 3.6 Methods by which Simulink monitors the error at each time step [15].

If you specify auto (the default), Simulink sets the absolute tolerance for each state initially to  $1e-6$ . As the simulation progresses, Simulink resets the absolute tolerance for each state to the

maximum value that the state has assumed thus far times the relative tolerance for that state. Thus, if a state goes from 0 to 1 and RTOL is  $1e-3$ , then by the end of the simulation the ATOL is set to  $1e-3$  also. If a state goes from 0 to 1000, then the ATOL is set to 1 [15]. The Simulink environment provides a useful tool with which to model and simulate systems that are not necessarily linear in nature, thus enabling more realistic phenomena to be incorporated into simulations to produce better and more reliable results. Although this software has not been tested that extensively very few shortfalls have been found.

The current blockset provides decent capability with respect to realistically modelling specific systems. Simulink does experience problems with respect to time step reduction during simulations. The main issue is that if the tolerances of the solver are too small the simulation time will be adversely affected due to the fact that the solver is trying to maintain accuracy by decreasing the time steps during large perturbations. Two nonlinear elements cannot be put in series due to the fact that they are modelled as current sources, thus additional elements had to be introduced to solve this issue, and this impacted on the realism of the model.



## 4. Estimation of the Rotor Parameters

### 4.1. Introduction

One of the objectives of this project is to research the application of Simulink for estimating the parameters of the machine. These include the rotor field parameters  $R_{fd}$  and  $L_{fd}$ , the damper winding impedances  $R_{kd1}$  and  $L_{kd1}$  and finally the equivalent airgap inductance  $L_{ad}$ . Three different models were designed and evaluated to estimate the field parameters and this chapter compares the performance of these methods. The algorithms for these approaches are compared in Figure 4.1. A point that must be stressed here is that no formal investigation into the estimation blockset has been undertaken as part of this thesis due to time constraints.

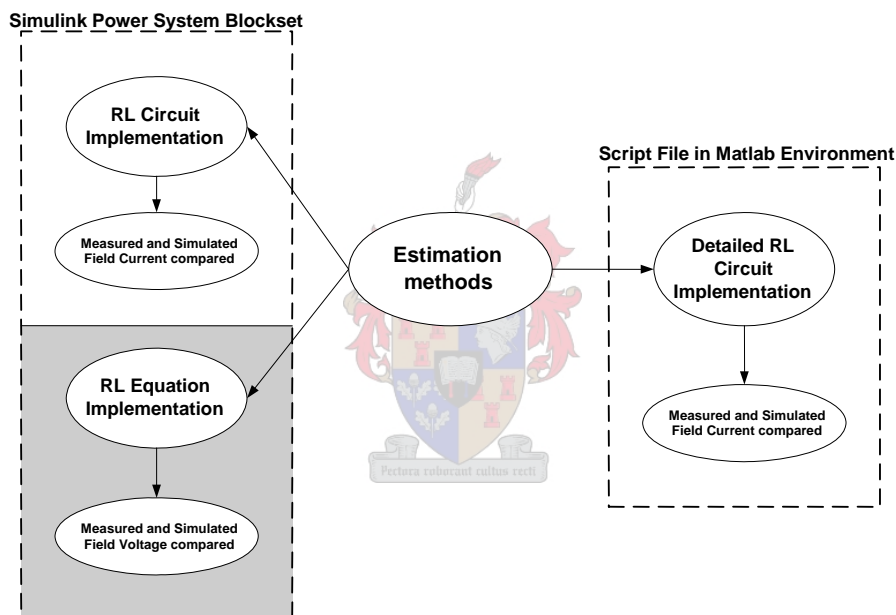


Figure 4.1 Estimation methods used to acquire rotor winding parameters.

Figure 4.1 shows that the estimation is carried out in two different environments, namely the Simulink Power Systems Blockset and in the MATLAB<sup>®</sup> workspace. The cost functions of the various models can be separated into two different categories; those that compare voltage waveforms and those that compare field current data. Finally one will notice that the detailed RL circuit could only be implemented using MATLAB<sup>®</sup> script. This was partly due to the fact that the complexity of the transfer function in the time domain allowed one to only implement a simpler version, however in the MATLAB<sup>®</sup> script environment, it is possible to work in the frequency domain.

### 4.2. Measured Field Data used for Estimation of the Rotor Parameters

The field data that is used for the estimation process comprises of the field current and field voltage responses obtained from RTDS tests. Figure 4.2 displays an enlarged view of a typical pseudo random binary sequence (PRBS) that is injected into the system.

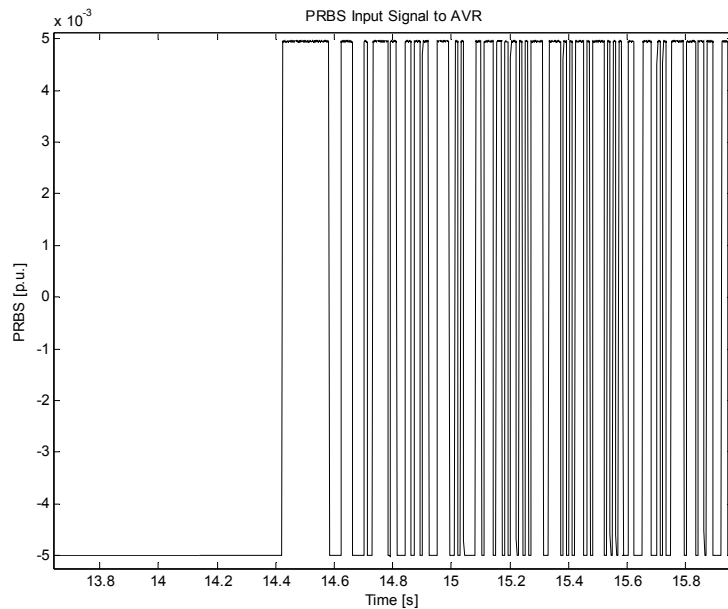


Figure 4.2 PRBS Signal.

Figure 4.2 shows the random lengths of the 1's and 0's, which is an attempt to cover a wider frequency range that will excite all elements in the circuit. The PRBS signal is injected into the AVR reference signal as shown in Figure 4.3. A PRBS signal of 1% translates into a signal consisting of band limited white noise that's amplitude is 1% of the AVR's reference value [7].

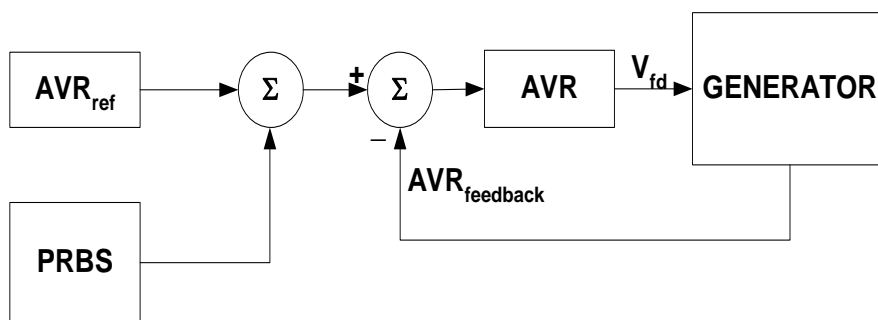
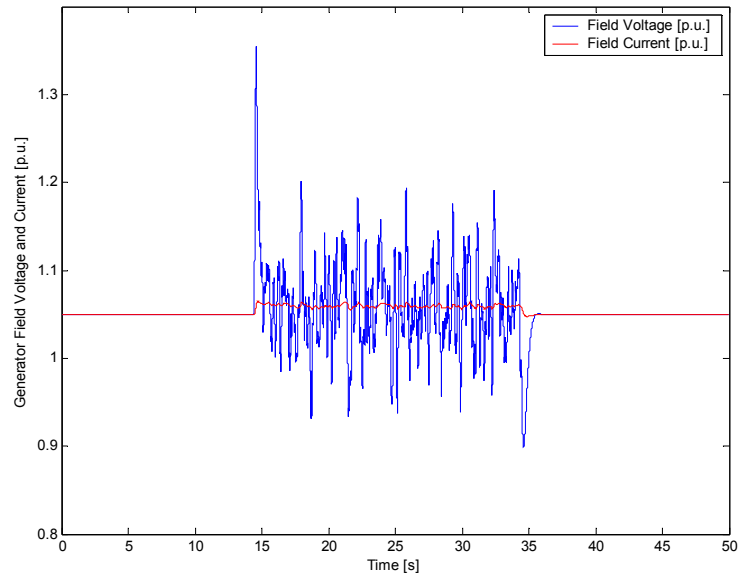


Figure 4.3 Pseudo Random Binary Sequence (PRBS) injection into system.

The input voltage signal to the AVR in this instance has a PRBS perturbation of 10% superimposed on a setpoint value of 1.05 pu. This signal is filtered by the AVR and applied to the field circuit, which will be used by the estimator to match the simulated system response to the actual measured response.



**Figure 4.4 Field voltage and current with 10 % PRBS used as input for estimated model**

The period of interest to the estimator coincides with the disturbance, thus the window of interest runs from  $t=14\text{s}$  to  $t=35\text{s}$ , as shown in Figure 4.4. Thus during the estimation routine only the data in the relevant time frame will be extracted and used.

### 4.3. Rotor Circuit Model

The estimation of the parameters of the rotor model is based on the classical rotor circuit proposed by de Mello *et al.* [12], which is shown in Figure 4.5.

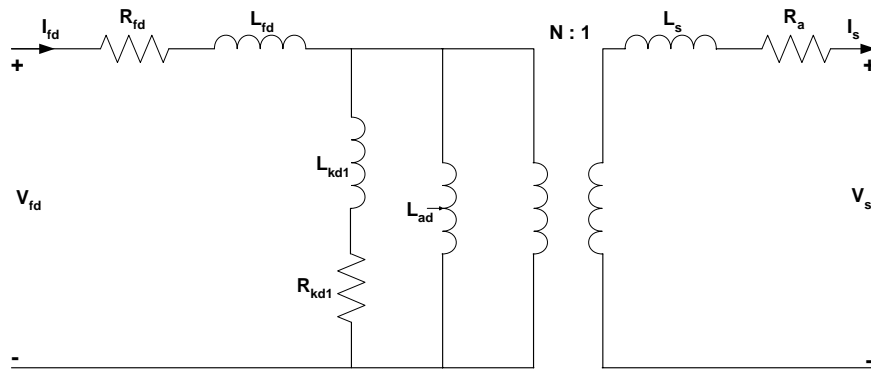


Figure 4.5 Classical Rotor Model with One Damper Winding in the D-axis [12].

Figure 4.5 shows the classical rotor model used in this analysis, which includes one damper winding in the d-axis, represented by the parameters  $L_{kd1}$  and  $R_{kd1}$ . The actual field winding parameters are given as  $L_{fd}$  and  $R_{fd}$ , while the mutual inductance between the a-phase of the stator and the d-axis is denoted by  $L_{ad}$ . The stator circuit consists of the synchronous inductance and the armature resistance given by  $L_s$  and  $R_a$  respectively. Figure 4.5 can be simplified due to the fact that test data from the field is acquired when the machine is under no-load conditions, thus the stator side of Figure 4.5 can be ignored. With all these omissions taken into account, the new model is given in Figure 4.6.

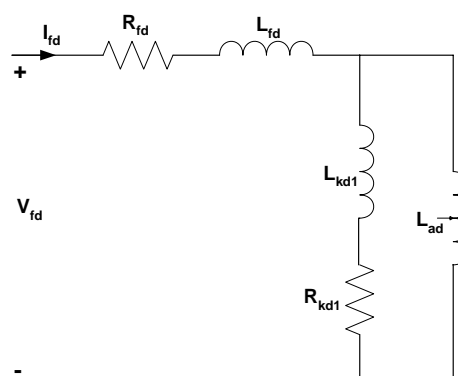


Figure 4.6 Simplified Rotor Model used for Parameter Estimation.

#### 4.4. Calculation of Field Impedances

As was stated earlier the primary test data comes out of no-load tests done on the synchronous machine. There are two calculations of importance that must be done before attempting to estimate the rotor parameters. The first task is to obtain the theoretical field parameters using equations from Kundur [1] and measured reactances given in the Palmiet Synchronous Machine Manual [3]. The second task is to determine what time constants will need to be covered to determine the field parameters.

##### 4.4.1. Field Base Values (Reciprocal per unit system)

The first step is to define base quantities for the field voltage and current. Thus from the work of Kundur [1], the base field current can be obtained by consulting the open-circuit saturation curves (OCC) obtained in Design and Technical Data Manual for Palmiet [3]. The OCC given in Figure 4.7 relates the terminal voltage  $E_t$  to the field current  $I_{fd}$ . In some cases the iron core will saturate and produce the effect portrayed in Figure 4.7. However saturation will be neglected in all the following test cases in order to simplify the problem and keep it to a first-order approximation, thus the figure below is for illustration purposes only. The perturbation in the RTDS measurements will most likely cause a certain amount of saturation, however for purposes of this task it has been decided to neglect saturation effects.

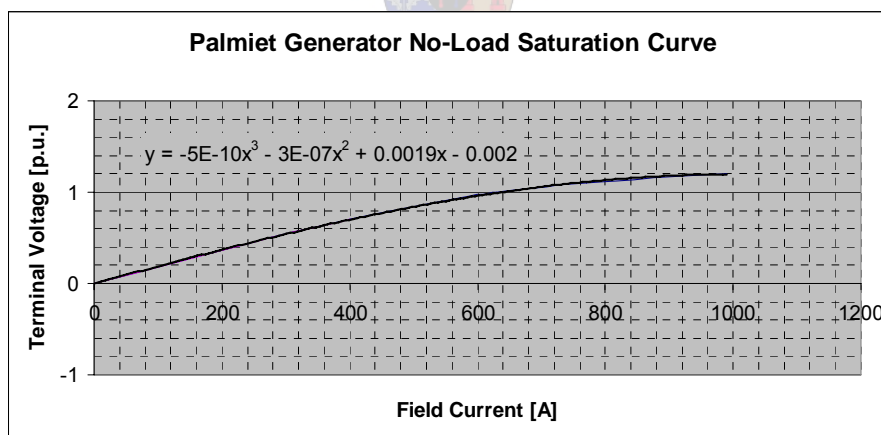


Figure 4.7 Palmiet unit open circuit saturation curve.

When the terminal voltage is operating at  $E_t = 1.0$  p.u. then the field current base in the **data manual** for Palmiet is given as

$$i_{fd_{base}} = 557.28 A \quad (4-1)$$

Using (4-1) to calculate the base value for the field voltage as



$$\begin{aligned}
 e_{fd_{base}} &= \frac{VA_{rating}}{i_{fd_{base}}} \\
 &= \frac{250MVA}{557.28A} \\
 &= 448.608kV
 \end{aligned} \tag{4-2}$$

where the machine's volt-ampere base is given as 250MVA and was obtained from the Design and Technical Manual [3]. The next step is to obtain the impedance base for the field circuit. The base impedance  $Z_{fd}$  is given as

$$\begin{aligned}
 Z_{fd_{base}} &= \frac{e_{fd_{base}}}{i_{fd_{base}}} \\
 &= \frac{448.608kV}{557.28A} \\
 &= 804.994\Omega
 \end{aligned} \tag{4-3}$$

The base value for the field inductance can be acquired by simply dividing the base impedance given in (4-3) by the synchronous speed base value  $\omega_{base}$  as follows:

$$\begin{aligned}
 \omega_{base} &= 2\pi f_{base} \\
 &= 2\pi 50 \\
 &= 314.159rad / s
 \end{aligned} \tag{4-4}$$

where the base frequency is 50 Hz.

Thus the base inductance  $L_{fd_{base}}$  can be expressed as

$$\begin{aligned}
 L_{fd_{base}} &= \frac{Z_{fd_{base}}}{\omega_{base}} \\
 &= \frac{804.994\Omega}{314.159rad / s} \\
 &= 2.5624H
 \end{aligned} \tag{4-5}$$

If the field resistance were to be converted into per unit form; its value would be very small. To negate this Kundur [1] proposes a change of base.

#### 4.4.2. Change of Base (Reciprocal to Non-Reciprocal)

The per unit system that is universally used in power system stability studies requires that the 1.0 per unit exciter output voltage be equal to the field voltage required to produce rated synchronous machine terminal voltage on the air-gap line [1], this corresponds to 1.0 per unit exciter output current as the machine field current. Kundur [1] refers to this as the *non-reciprocal per unit system* to highlight the difference between this convention and the

reciprocal system that is used for modelling the synchronous machine as shown in equations (4-1)-(4-5).

The exciter uses interfaces with the machine at two points, namely the field terminals and the stator terminals. Generally the speed and voltage of the synchronous machine are used as inputs to the exciter, while the exciter controls terminal voltage through the field circuit. In order to change base from the existing reciprocal system to the non-reciprocal convention Kundur [1] presents a relationship whereby the output voltage/current of the exciter can be linked to the field voltage and current of the machine. Kundur's work on this topic is briefly covered in the following text. If one refers to Figure 4.8, the field current required to produce 1.0 per unit armature terminal voltage on the air-gap line is given by the following relationship

$$E_t = e_q = L_{adu} i_{fd} = 1.0 \text{ p.u.} \quad (4-6)$$

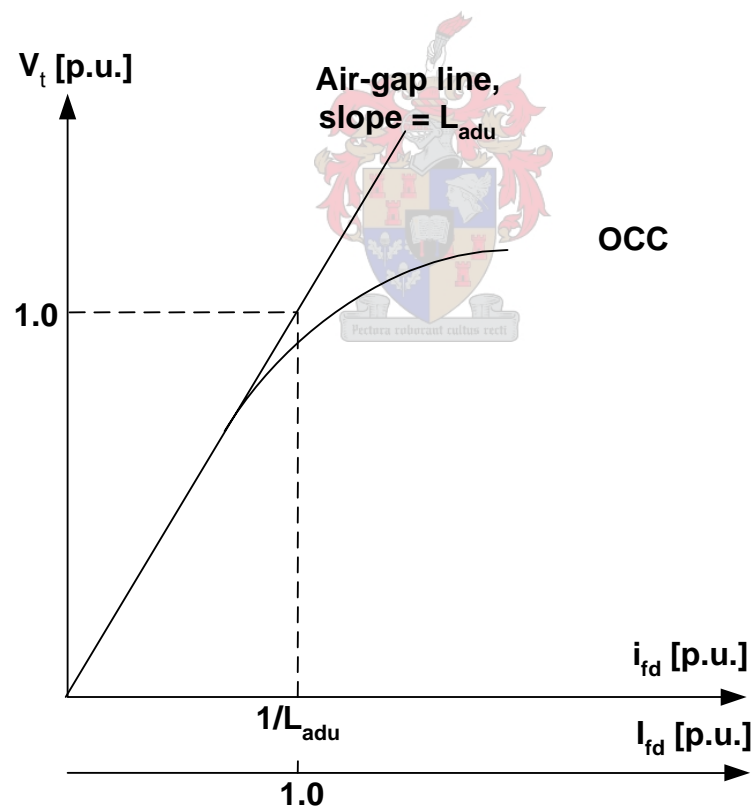


Figure 4.8 Open Circuit Characteristics of the Synchronous Machine.

where  $L_{adu}$  is the unsaturated mutual inductance across the air-gap between the field components in the d-axis and the a-phase stator windings. In the existing per unit system the field current  $i_{fd}$  required to produce rated terminal voltage on the air-gap line is given by

$$i_{fd} = \frac{I}{L_{adu}} \text{ p.u.} \quad (4-7)$$

And the field voltage can be expressed as

$$e_{fd} = R_{fd} i_{fd} = \frac{R_{fd}}{L_{adu}} \text{ p.u.} \quad (4-8)$$

As stated in earlier text the value of the exciter output current,  $I_{fd}$  is equal to 1.0 per unit, thus

$$I_{fd} = L_{adu} i_{fd} \quad (4-9)$$

And its corresponding exciter output voltage  $E_{fd}$  is given by the following relationship

$$E_{FD} = \frac{L_{adu}}{R_{fd}} e_{fd} \quad (4-10)$$

Under steady state the values of  $I_{fd}$  and  $E_{fd}$  are the same, whilst under transient conditions these values differ; the field voltage is determined by the excitation system while the field current is determined by the dynamics of the field circuit. Figure 4.9 shows the conversion interface between the two per unit systems.

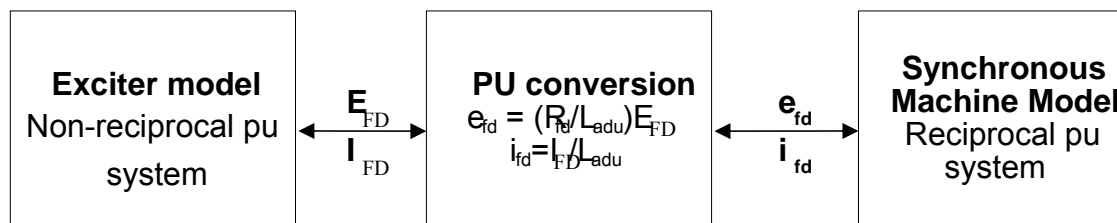


Figure 4.9 Per Unit Conversion at interface between Exciter and Synchronous Machine Field Circuit.

As stated earlier, if one is working in the non-reciprocal per unit system the steady state values for both  $E_{fd}$  and  $I_{fd}$  must be equal. Figure 4.4 exhibits such behaviour with both signals prior to the perturbation being equal. The next logical step is to calculate the base values of the per unit system that was used to normalise these values, thus the field current base that was acquired earlier is

$$i_{fd_{base}} = 557.28 \text{ A} \quad (4-11)$$

Consulting the Design and Technical Data Manual for Palmiet [3], the calculated resistance for the field winding,  $R_{fd}$ , at 75°C is

$$R_{fd} = 0.186\Omega \quad (4-12)$$

Thus under the current per unit system  $R_{fd}$  would work out as

$$\begin{aligned} R_{fd_{p.u.}} &= \frac{R_{fd}}{Z_{fd_{base}}} \\ &= \frac{0.186\Omega}{804,994\Omega} \\ &= 0.000231 \end{aligned} \quad (4-13)$$

The unsaturated value of the  $L_{adu}$ , using data given in [3] is

$$\begin{aligned} L_{adu} &= L_d - L_t \\ &= 1.23 - 0.13 \\ &= 1.1 p.u. \end{aligned} \quad (4-14)$$

The relationship given in (4-14) is taken from work done by Kundur [1]. Thus with the result from (4-14) and the field resistance in per unit expressed in (4-13) the conversion interface proposed in Figure 4.9 is now used to perform a change of base.

$$\begin{aligned} E_{fd_{base}} &= \frac{R_{fd_{p.u.}}}{L_{adu_{p.u.}}} e_{fd_{base}} \\ &= \frac{0.000231}{1.1} 448.61kV \\ &= 94.2313V \end{aligned} \quad (4-15)$$

where  $E_{fd_{base}}$  becomes the new base for the field voltage. Thus the new base for the field current is given as

$$\begin{aligned} I_{fd_{base}} &= \frac{i_{fd_{base}}}{L_{adu}} \\ &= \frac{557.28A}{1.1} \\ &= 506.62A \end{aligned} \quad (4-16)$$

Following the same procedures used in equations (4-2) to (4-5) the base values for the impedances are obtained and found to be

$$Z_{fd_{base}} = 0.186\Omega \quad (4-17)$$

$$L_{fd_{base}} = 0.592mH \quad (4-18)$$

### 4.4.3. Calculation of Theoretical Field Parameter Values

In the previous section a change of base was implemented in order to match the existing base that was used in the RTDS studies. The next step is to try and obtain the order of magnitude of the field winding impedances that must be estimated. Consulting Figure 4.6, one will notice that the d-axis transient open circuit time constant for the circuit can be given as

$$T_{do'} = \frac{L_{adu} + L_{fd}}{R_{fd}} \quad (4-19)$$

where  $T_{do'}$  is the open circuit time constant of the machine. This time constant, along with the field resistance is given in the Design and Technical Data manual [3].

### 4.5. Method 1: RL Circuit Implementation in Simulink

As shown in Figure 4.1 the first of the three methods attempts to implement a simple RL circuit using the Simulink Power Systems Blockset. The network topologies can be found in Appendix D, however the model that is being implemented here is presented below in Figure 4.10.

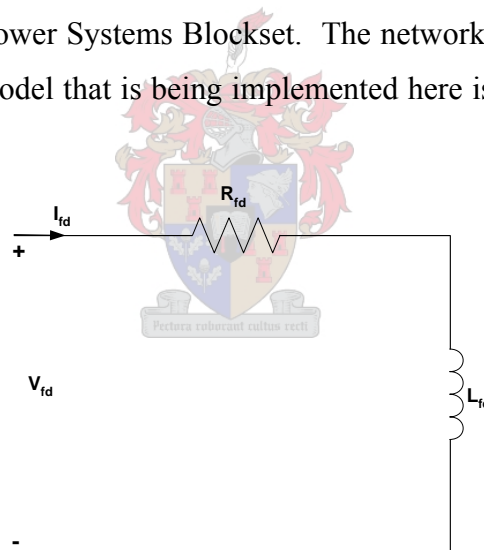


Figure 4.10 Simplified RL circuit model used in method 1.

#### 4.5.1. Cost Function: Measured and Simulated Field Current Comparison

Figure 4.11 shows the network topology of the model that is implemented in Simulink which compares measured and simulated field current waveforms. The voltage waveform is used to drive the model of the rotor circuit, which is represented by a simple RL circuit.

The simulated current is compared to the measured data and the parameters are changed accordingly by the estimator to minimise the error. For this particular setup it is assumed that the flux linkage of the rotor winding does not saturate.

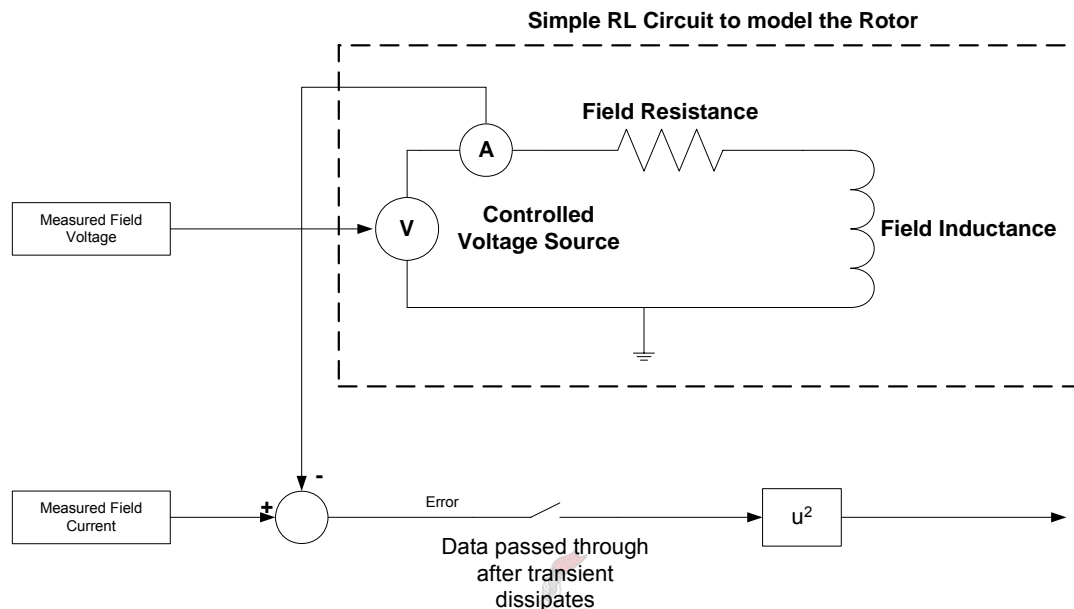


Figure 4.11 Rotor model used in Parameter Estimation: RL Circuit Method.

As explained before, the switch only allows data to be passed to the estimator for the period of the perturbation. The estimator adjusts either the resistive or inductive element, or both. The estimated current and the original measured waveforms are displayed in the Figure 4.12.

#### 4.5.2. Test: 10 % PRBS Input Signal

A number of tests were conducted for this particular model. However due to the relatively poor performance of the estimator with this cost function, only results for the 10% PRBS test are documented here. Figure 4.12 shows how the estimator tracked the measured waveform. The estimated response does not follow the original signal as well as was hoped, and part of the reason for this is that the RL model may actually be under-determined for estimation purposes. This could definitely be a factor if one compares this model to the circuit in Figure 4.5. The circuit in Figure 4.5 contains damper winding elements which are omitted in Figure 4.10. The second model which represents the equation for a RL circuit may hold the answers in both areas where this model has failed to produce satisfactory results.

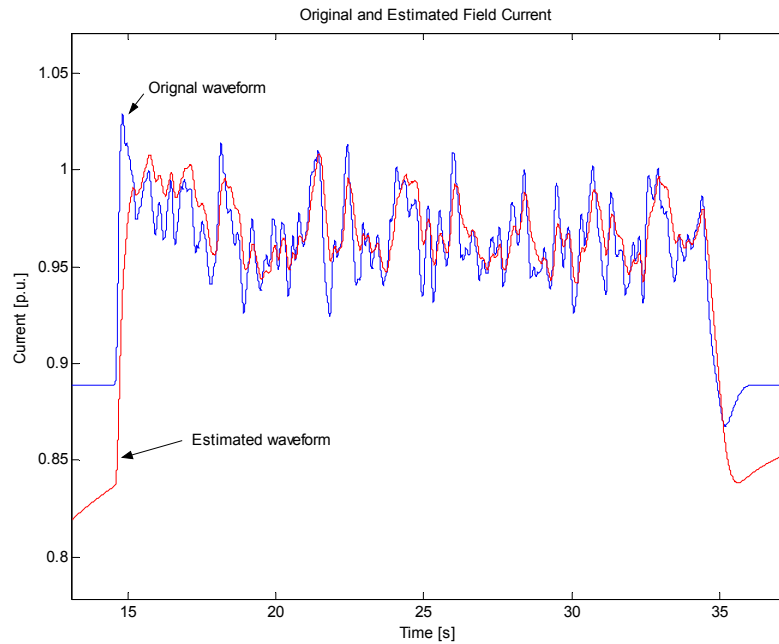


Figure 4.12 Field current estimation using RL model - 10% PRBS.

## 4.6. Method 2: RL Equation Implemented in Simulink

### 4.6.1. Network Model Topology and Cost Function

Figure 4.13 introduces the second model setup. There are two distinct differences between this model and the previous circuit. The obvious difference is that this version doesn't use an RL circuit; the equation is implemented instead. The other main difference is that the simulated field voltage is compared to the measured field voltage.

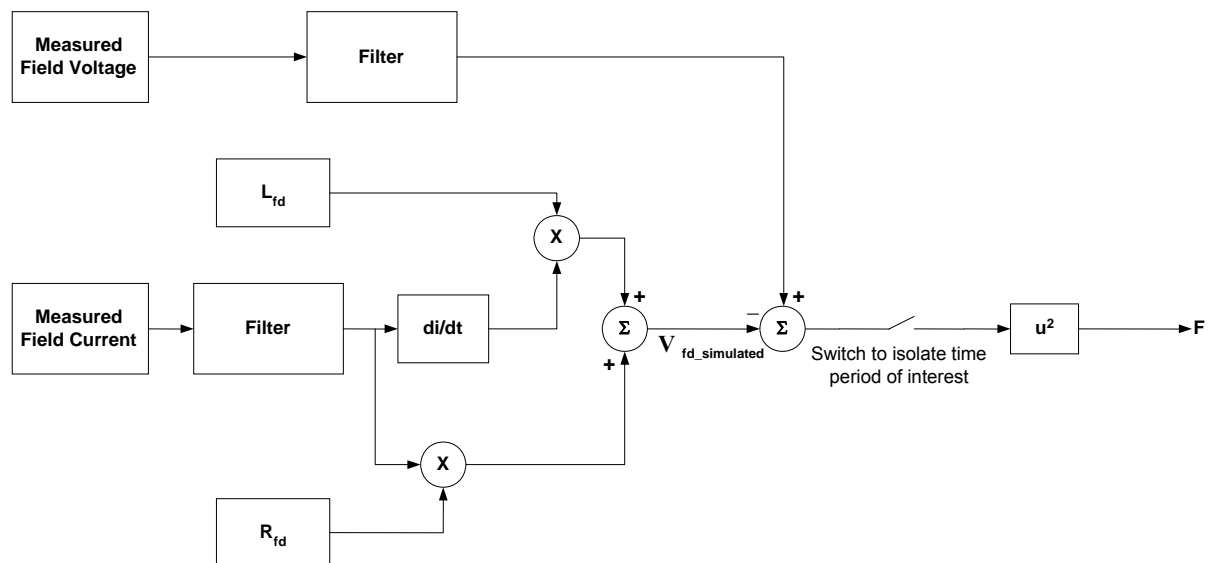
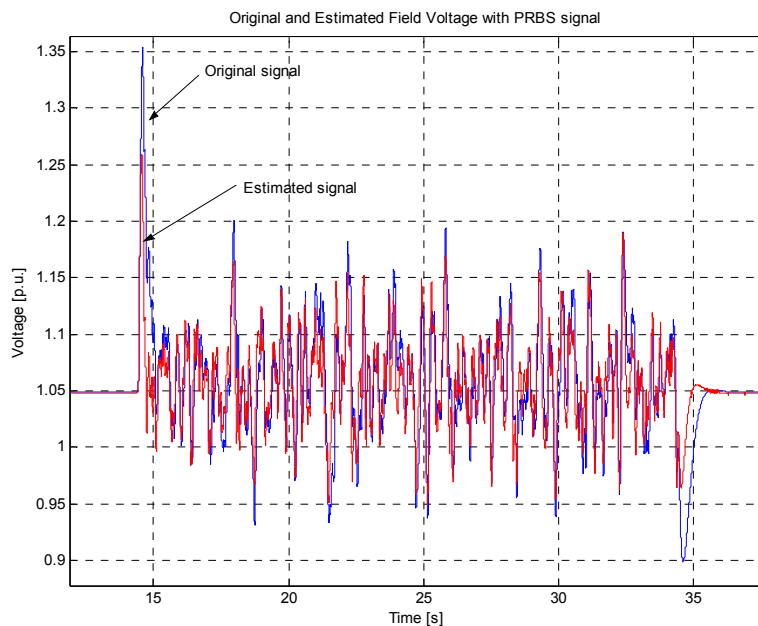


Figure 4.13 Field Rotor Model in Parameter Estimation: RL Equation method.

The field data was sampled at 1.25 kHz and this proved to be a problem when it came to estimating the inductance due to discontinuous nature of the waveform (noise), being fed through the differentiator block. However the both the incoming field voltage and current were filtered in order to single out certain time constants of the rotor circuit; thereby producing a smoothing effect on the data. The fastest time constants could be of the order of a couple of milliseconds thus a low pass filter was included to allow through perturbations on the signal that would include this information.

#### 4.6.2. Test 1 – 1% PRBS Waveforms

The initial test involves data with a 1% perturbation relative to the AVR reference. One can clearly see that the filter cut-off frequency was set such that all frequency components were passed through. The isolation of certain time constants will not be included in this text however the results from these tests are included in Appendix D.



**Figure 4.14 Original and estimated field voltage with 1% PRBS.**

Comparing Figure 4.14 and Figure 4.15 with Figure 4.12 reveals a marked improvement in the estimation model which enables the resulting waveform to follow the original signal closely.



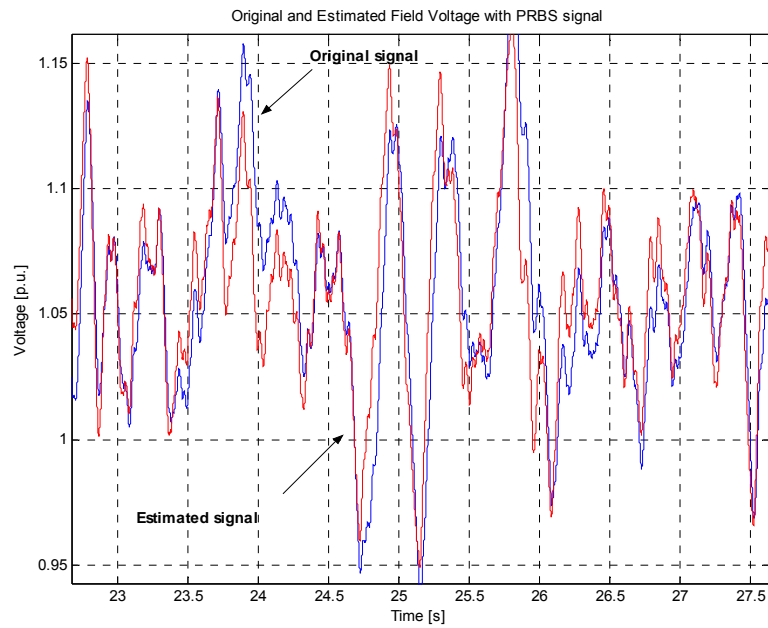


Figure 4.15 Original and estimated field voltage waveforms - zoomed view.

Figure 4.16 shows the percentage error of the estimated and original signals. The error rarely goes above 5 %, which for accuracy purposes falls in the accepted region. What is promising about this result is that the waveforms correlate closely, not only error-wise but also dynamically, thus the inductance value estimated here is expected to be close to the actual level on the rotor.

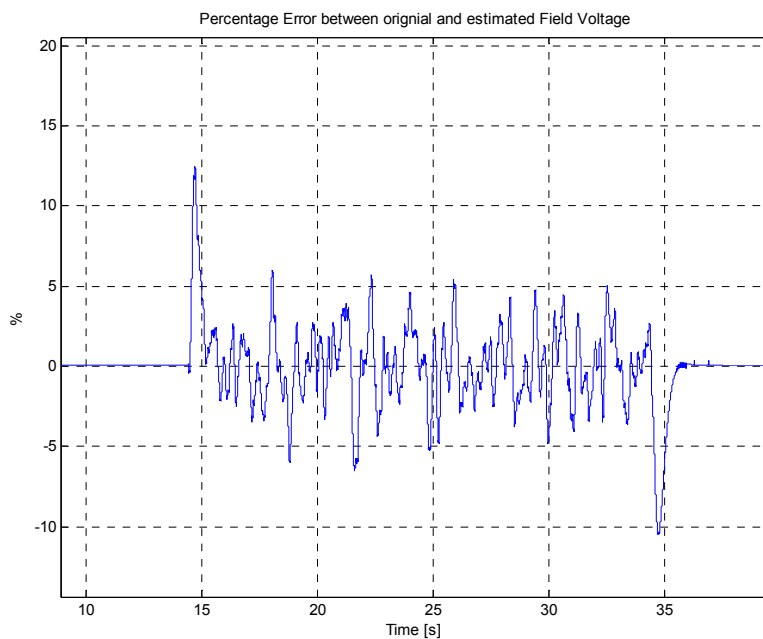


Figure 4.16 Error plot of original and estimated field voltage.

### 4.6.3. Test 2 – 10 % PRBS Waveforms

Figure 4.17 and Figure 4.18 show the estimators’ performance with a 10 % perturbation level. Once again the original signal is followed closely by the estimator.

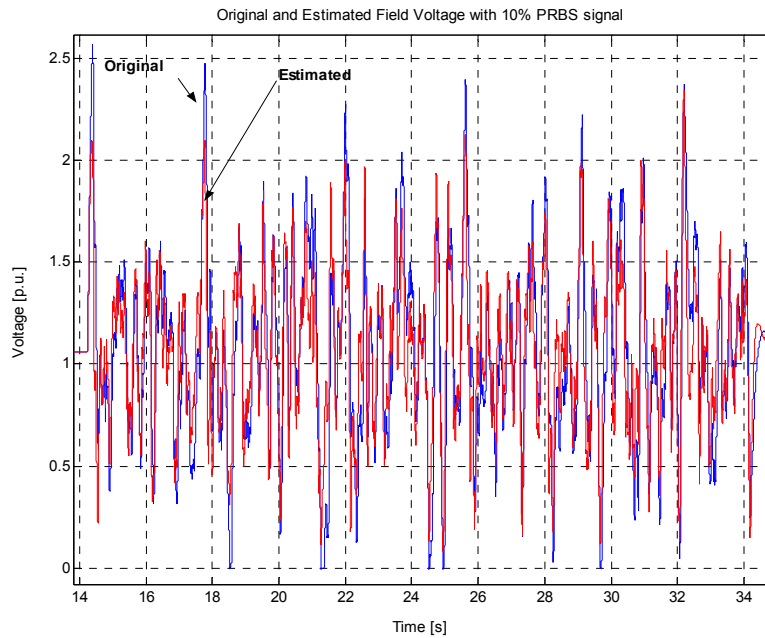


Figure 4.17 Original and estimated field voltage with 10% PRBS.

Figure 4.18 shows a close-up view of how the estimated value tracks the original.

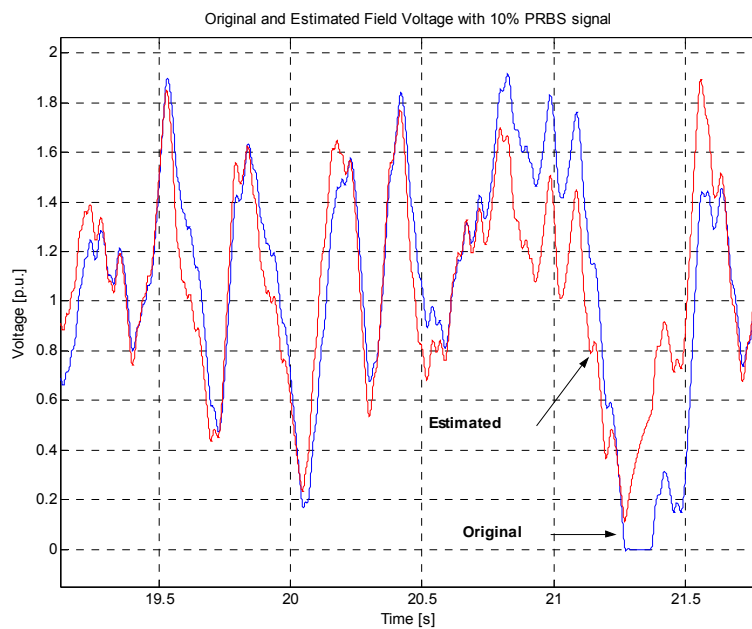


Figure 4.18 Original and estimated field voltage with 10% PRBS - zoomed view.

Figure 4.19 shows the normalized estimated plot and confirms that the error remains under 2% for most of the sample period, with the exception of a few spikes.

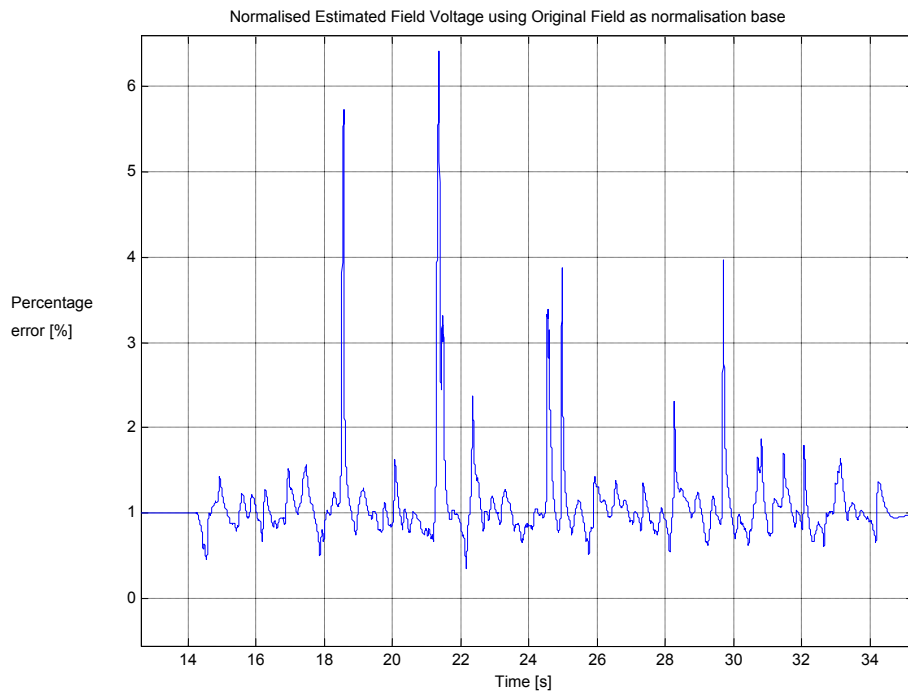


Figure 4.19 Error between original and estimated values.

#### 4.6.4. Verification of Estimator

Testing the integrity of the estimator is the final step in the rotor parameter estimation process. This entails replacing the original field voltage with the estimated value and re-estimating the parameters. Some noise is added to the simulated field voltage so that the estimator does not have zero error, this process is shown in Figure 4.20. The values that the estimator produces should be identical to those acquired in the initial process.

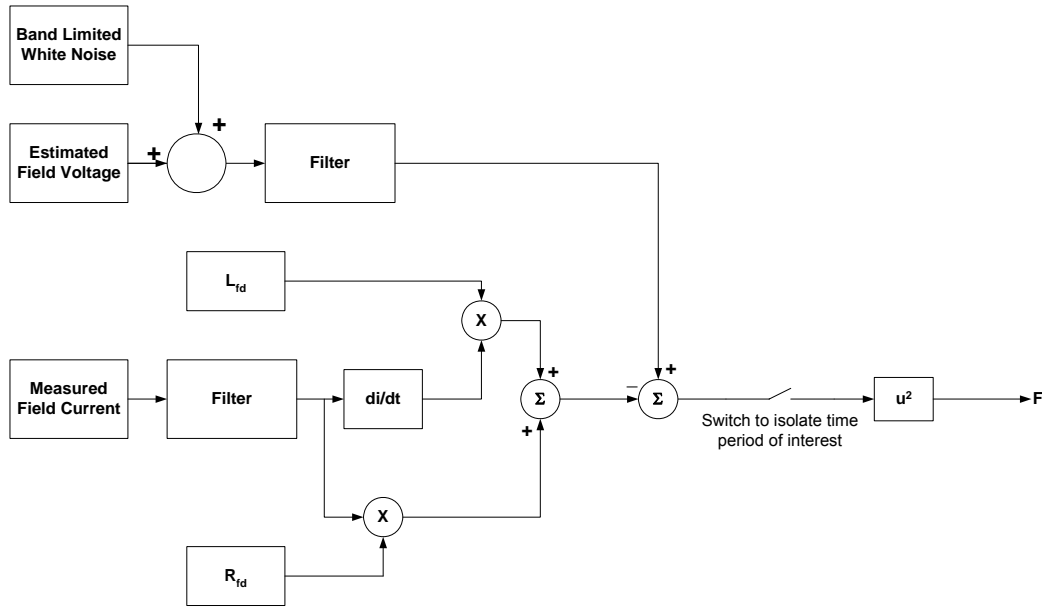


Figure 4.20 Equation model with the addition of white noise to the simulated field voltage.

Figure 4.21 shows that the simulated and re-estimated waveforms follow each other almost perfectly, which reinforces the notion that the estimator is indeed producing credible results.

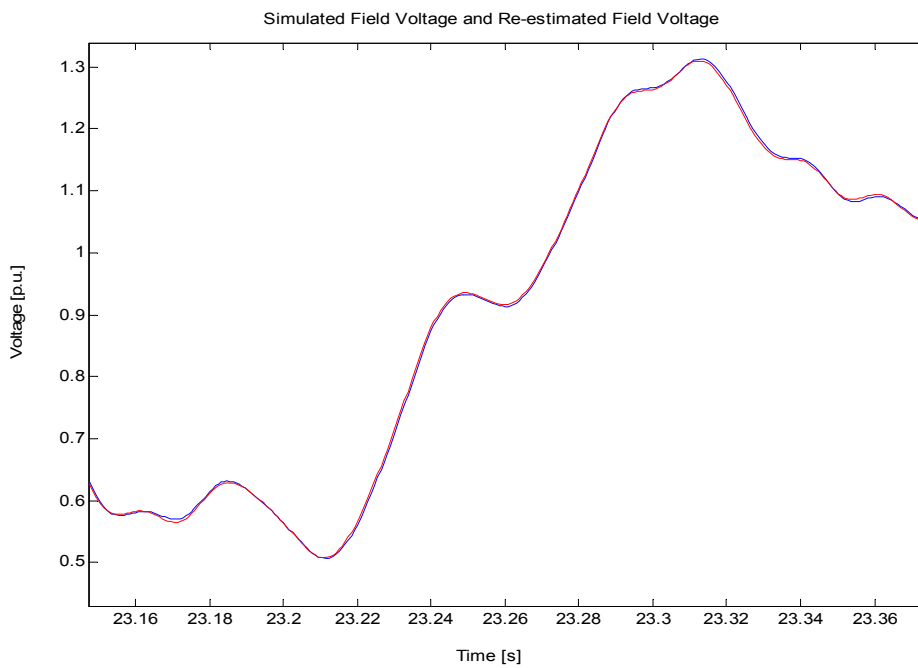


Figure 4.21 Simulated field voltage and re-estimated field voltage.

### 4.7. Method 3: Detailed RL Equation Implemented in MATLAB<sup>®</sup>

The final method of obtaining the field parameters firstly attempts to estimate a much wider range of values; these include the field impedances  $L_{fd}$  and  $R_{fd}$ , the damper winding

impedances  $L_{kd1}$  and  $R_{kd1}$  and finally the airgap inductance  $L_{ad}$ . The optimisation procedure now becomes somewhat more complicated than before, in that instead of estimating for only 2 parameters as was the case in the past two sections, the problem becomes a five dimensional estimation procedure.

With the added complexity of the problem comes the need to ensure that the circuit for the rotor is adequately modelled so that the optimiser does not have to estimate for an under-determined system. Thus the model for the rotor circuit, proposed by de Mello [12] and given in Figure 4.6 is now implemented in MATLAB<sup>®</sup>.

### 4.7.1. Derivation of Rotor Winding Transfer Function

The initial step is to derive the transfer function for the rotor circuit given in Figure 4.6. The time domain response of the system is described by differential equations, thus in order to obtain the transfer function one has to first transfer to the frequency domain with the use of the Laplace transformation [18].

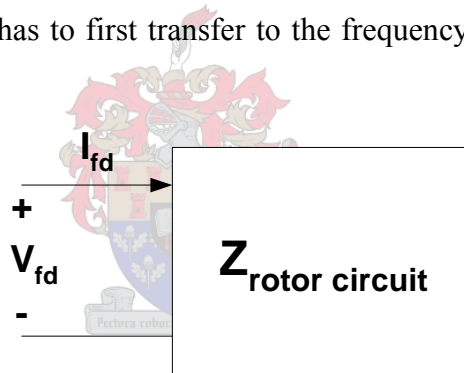


Figure 4.22 Rotor terminal definitions

Thus using common circuit analysis methods as presented in [20], the impedance as seen from the terminals of the circuit of Figure 4.22 can be expressed in the s-domain (frequency) as

$$Z_{rotor} = R_{fd} + L_{fd} + \frac{sL_{ad} (sL_{kd1} + R_{kd1})}{R_{kd1} + s(L_{ad} + L_{kd1})} \quad (4-20)$$

where the total impedance,  $Z_{rotor}$  is as seen from the terminals of the circuit in Figure 4.22.

After simplification and re-organisation the total rotor impedance is given as

$$Z_{rotor} = \frac{s^2 \left( L_{fd} + \frac{L_{ad}L_{kd1}}{L_{ad} + L_{kd1}} \right) + s \left( L_{fd} + \frac{L_{fd}R_{kd1}}{L_{ad} + L_{kd1}} + \frac{L_{ad}R_{kd1}}{L_{ad} + L_{kd1}} \right) + \frac{R_{fd}R_{kd1}}{L_{ad} + L_{kd1}}}{s + \frac{R_{kd1}}{L_{ad} + L_{kd1}}} \quad (4-21)$$

What is significant about (4-21) is that the transfer function is an improper function, thus in order to simulate in MATLAB<sup>®</sup> one has to invert this equation, thus yielding the effective admittance of the rotor circuit. Thus  $Y_{rotor}$  is given as

$$Y_{rotor} = \frac{s + \frac{R_{kd1}}{L_{ad} + L_{kd1}}}{s^2 \left( L_{fd} + \frac{L_{ad} L_{kd1}}{L_{ad} + L_{kd1}} \right) + s \left( R_{fd} + \frac{L_{fd} R_{kd1}}{L_{ad} + L_{kd1}} + \frac{L_{ad} R_{kd1}}{L_{ad} + L_{kd1}} \right) + \frac{R_{fd} R_{kd1}}{L_{ad} + L_{kd1}}} \quad (4-22)$$

Basic circuit analysis gives the voltage of a circuit in terms of the current and the impedance, thus for the rotor circuit  $V_{fd}$  is expressed as

$$V_{fd} = I_{fd} Z_{rotor} \Rightarrow I_{fd} = \frac{V_{fd}}{Z_{rotor}} \Rightarrow I_{fd} = V_{fd} Y_{rotor} \quad (4-23)$$

Thus from equation (4-23) one can see that the field current,  $I_{fd}$ , can be simulated by filtering the measured field voltage,  $V_{fd}$ , with the filter expressed by equation(4-22). MATLAB<sup>®</sup> has a function which will filter analogue signals using a digital filter, thus before using this filter; it has to be transformed into the digital domain using the *bilinear* function in MATLAB<sup>®2</sup>.

#### 4.7.2. Cost Function

Before applying the method proposed in this section as part of the estimation procedure, it is advisable to test if the optimiser can estimate for a known set of parameters if one were to measure the step response of the system. However before this test is run the setup of the estimation procedure is initially explained. Using the derivation obtained in equation (4-23) a cost function will be constructed, which will be designed to compare the simulated field current with the measured field current.

Figure 4.23 presents the arrangement of the estimator and cost function that will be used to estimate for the various parameters of the rotor circuit. The measured field voltage is passed through a digital filter, and as derived in equation (4-23) the filtered data is in fact the simulated field current  $I_{fd\_simulated}$ .

<sup>2</sup> Appendix D contains some rotor estimation figures, tables and models. It also contains the code used to estimate for the parameters.

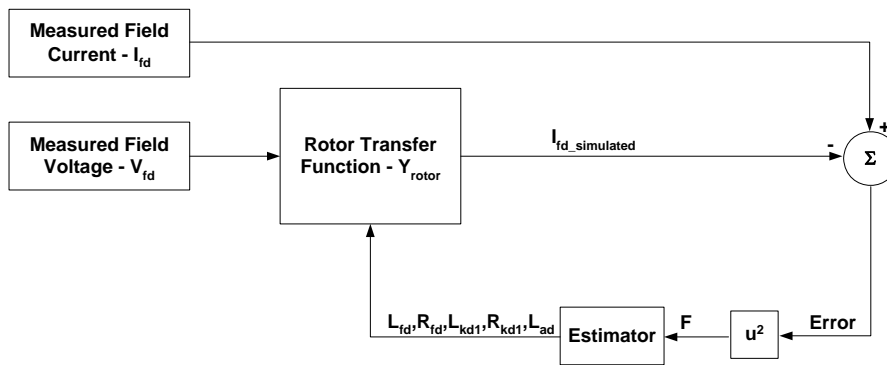


Figure 4.23 Estimator and cost function arrangement.

The cost function used for the rotor estimation is the same as that used throughout this project, namely the method of least squares presented in section 2.5. The error is then fed into the optimiser which then toggles the various parameter values in order to produce a minimum error between the simulated and measured field current waveforms.

### 4.7.3. Estimator Integrity Test

Now that the transfer function of the rotor circuit and the estimator setup has been explained the focus of this section will be centred on an integrity test imposed on the estimator in order to gauge whether the optimiser is working properly. To achieve this, arbitrary variables are set for the various rotor parameters and a step response of the system is compiled. All the parameters are then set to a starting value of 1 per unit, and then using the step response as the measured waveform, the estimator then optimises the circuit, with the hope that it will arrive at the original values. The system settings are arbitrary to a certain extent, however to try and emulate typical conditions on the rotor one must verify if the estimator can distinguish between different time constants that it will likely encounter. Thus the initial settings for the various parameters are presented below in Table 4-1, and may not be entirely realistic however they do present the estimator with a system containing two time constants.

Table 4-1 Starting values for rotor parameter estimation.

Rotor Parameters	Values [p.u.]
$L_{fd}$	1
$R_{fd}$	0.1
$L_{kd1}$	0.01
$R_{kd1}$	0.1
$L_{ad}$	3

With the filter values set to the parameters stated above in Table 4-1, the step response is obtained. The parameters are then all reset to 1 per unit and the estimator is set to work on estimating for the initial values stated in the previous table.

To do this the original step response is compared to the simulated response and parameters toggled accordingly. The resulting step response and the original waveform are shown below in Figure 4.24.

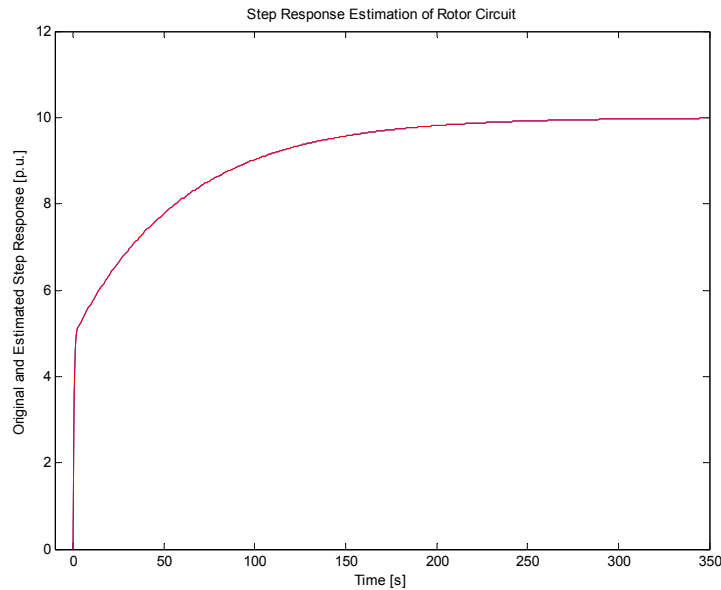


Figure 4.24 Estimated and Original Step Response of Rotor Circuit.

Considering the waveform in Figure 4.24 it is apparent that the step response consists of two major time constants. There is an initial fast response followed by a much longer time constant. This was done so that one could test whether the estimator would be able to detect the different time constants.

Table 4-2 Estimation Integrity Test - Step Response.

Estimation Integrity Test - Step Response						
Parameters [p.u.]		Lfd	Lkd1	Rkd1	Lad	Rfd
	Original	2.00	0.01	0.50	9.00	0.10
	Initial	1	1	1	1	1
	Estimated	2.00991	0.00003	0.49889	8.99014	0.10000
% Error		0.49547	-99.68678	-0.22128	-0.10961	-0.00010

Table 4-2 gives the results of the integrity test. The table consists of three sets of data; the original set points, the initial starting values for the estimator and the final convergence



values. With the exception of the damper winding inductance,  $L_{kd1}$ , the estimated parameters all are within 0.5% of their original values.

While this is good news, the damper winding inaccuracy could be a problem if the time constants are not more distinguishable. The estimator clearly struggled to acquire an accurate result for  $L_{kd1}$  due to the system being relatively insensitive to any changes in the value of this parameter.

#### **4.7.4. Estimation of Field Parameters using PRBS Waveforms**

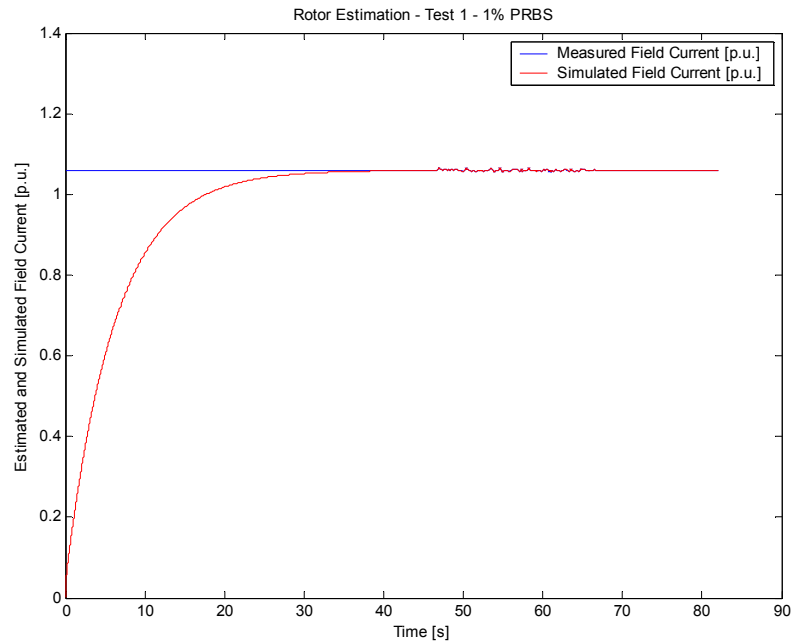
Sections 4.7.1 - 4.7.3 introduced a model that would completely describe the rotor circuit, a transfer function was then derived and a step response test carried out to gauge how the estimator would cope. It was found that to a large degree the estimator had succeeded in optimising the model; however concern was raised over the relative insensitivity of the cost function when the estimator tried to determine the damper winding inductance.

The hope is that with the introduction of the PRBS that the cost function will prove to be more responsive to parameter changes and thus ensure quality results. As was the case in the previous section, the transfer function along with the measured field voltage,  $V_{fd}$ , will be used to predict the field current. The error between the measured and simulated field current will be calculated and fed to the estimator for further optimisation.

The data used for this test case will be the same data that was used in the previous two methods. Thus the field voltage and current will look similar to those presented in Figure 4.4. Two tests will be conducted using the 1% and 10% PRBS waveforms respectively. As much detail as possible is included in this section, however for further information on the various rotor tests the reader is referred to Appendix D.

##### **4.7.4.1. Test 1 – 1 % PRBS Waveforms**

The first test involves the 1% PRBS field data. Figure 4.25 introduces the first result of the estimation tests.



**Figure 4.25 Measured and Estimated Field Current Waveforms for the 1% PRBS data.**

What is interesting about this view is the relatively long rise time of the filter due to the apparent step in the input data. The reason for this can be attributed to the fact that the filter is not given an initial steady state value, added to this is the fact that the window period of interest is only 20 seconds in duration and is situated over the perturbation.

Another point to take note of is that the input data has been padded with 32 seconds of initial steady state values in order to ensure that the value of the output to the filter coincides with the correct steady state value prior to the occurrence of the disturbance. However Figure 4.25 does not give a good indication of how well the estimator has performed, to this end the figure on the following page, Figure 4.26, presents a closer examination of the actual perturbation and displays how well the simulated signal follows the original waveform.

Figure 4.26 proves that the estimator has succeeded in tracking the measured waveform with a great deal of accuracy. The original data is not emulated perfectly and one can point to a particular area, this being the time shortly before the perturbation. This can be attributed to the fact that the relatively slow time constant of the circuit has not completely been allowed to settle to a final value.

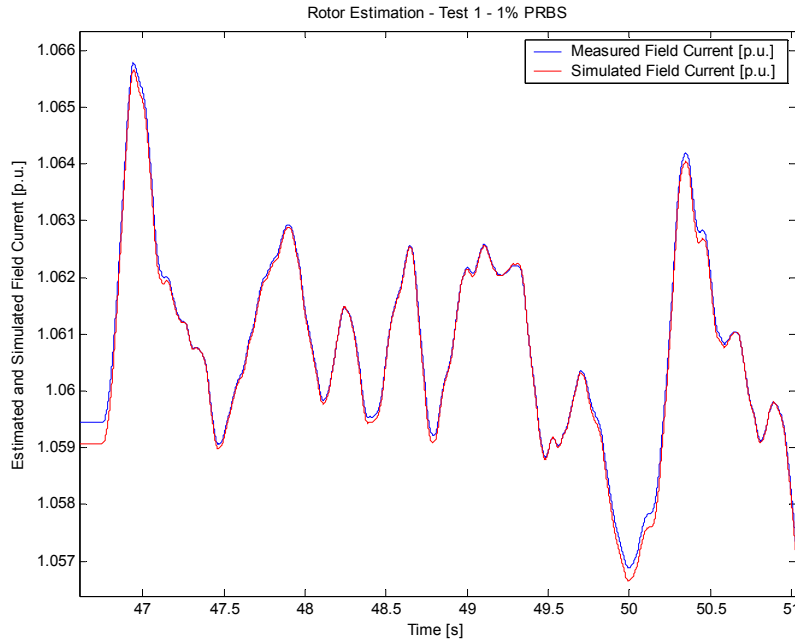


Figure 4.26 Measured and Estimated Field Current Waveforms for the 1% PRBS data – detailed view.

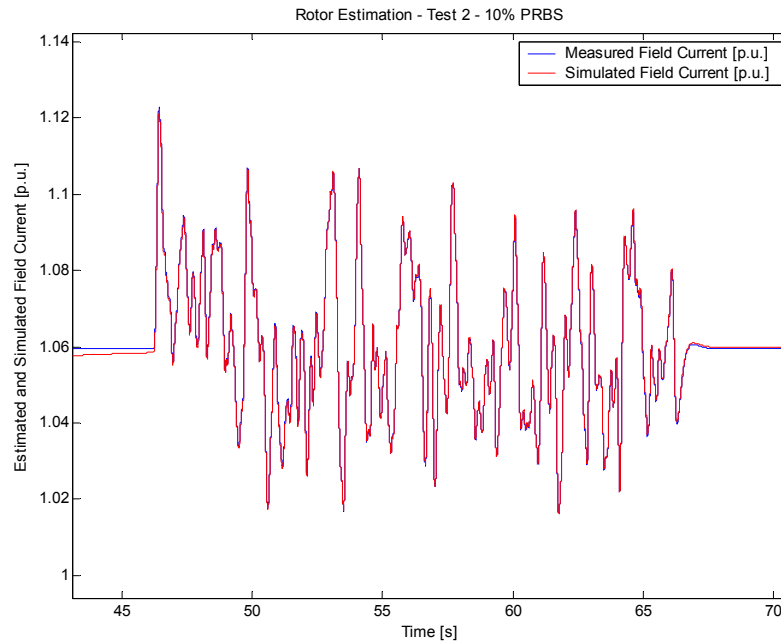
Evaluating the estimator’s performance strictly on signal tracking, would lead to the conclusion that the data in Table 4-3 should, to a large degree be representative of actual parameter values of the rotor winding circuit. However as seen in section 4.7.3 relative insensitivity of the model can lead to inaccuracies. Unfortunately the base value for the field quantities was not acquired due to time constraints not allowing further investigation into the internal workings of the RTDS. Thus the original row in the table below has remained empty.

Table 4-3 Rotor Estimation - Test 1 - 1 % PRBS.

Rotor Estimation Test 1 - 1% PRBS						
Parameters [p.u.]		Lfd	Lkd1	Rkd1	Lad	Rfd
	Original	n/a	n/a	n/a	n/a	n/a
	Initial	1	1	1	1	1
Estimated	1.6018	0.0001	15.4939	4.4318	0.9998	
% Error	n/a	n/a	n/a	n/a	n/a	

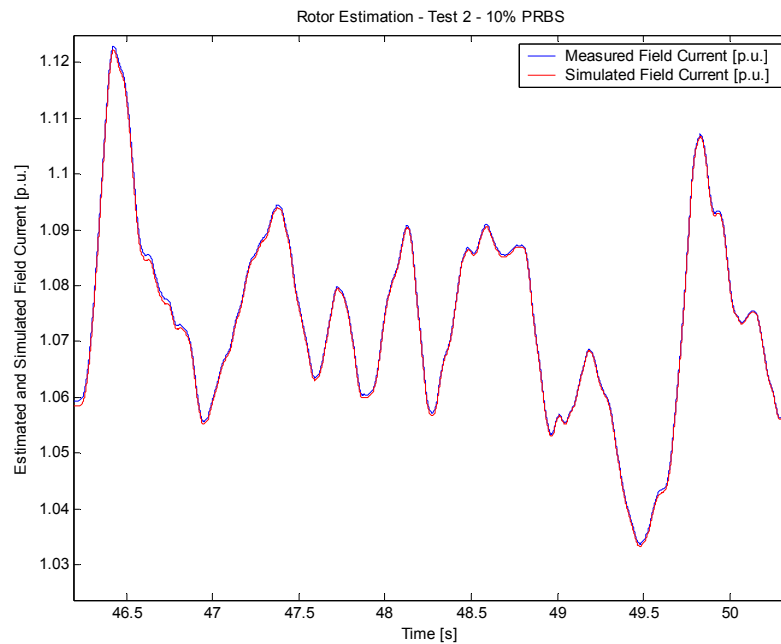
4.7.4.2. Test 2 – 10 % PRBS Waveforms

The second case is almost identical to the first however this time the PRBS signal is boosted to 10%. Figure 4.27 confirms that the result is similar to that in Test 1, and in fact exhibits that same offset error prior to the perturbation as was evident in the previous case.



**Figure 4.27 Measured and Estimated Field Current Waveforms for the 10% PRBS data.**

The improvement or lack thereof in terms of the tracking performance of the estimator when faced with a greater disturbance (magnitude-wise) is brought under scrutiny by the following figure, Figure 4.28. Comparing Figure 4.28 with Figure 4.26 shows the difference in terms of following is not very significant.



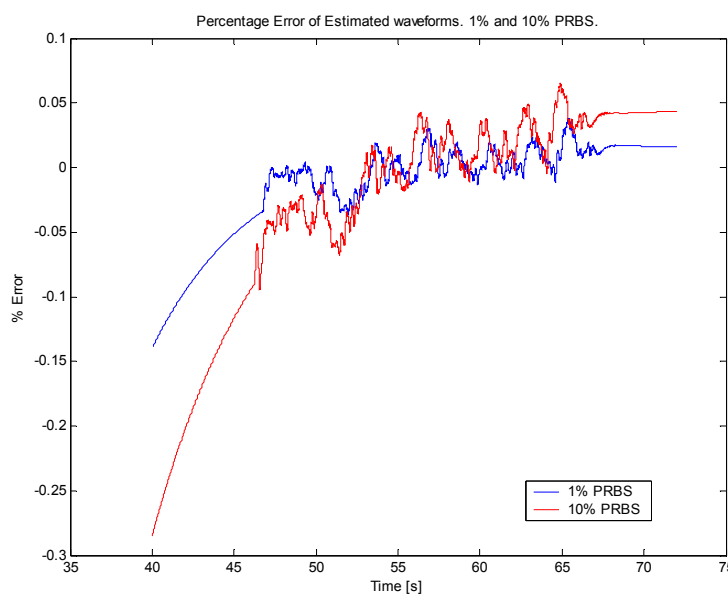
**Figure 4.28 Measured and Estimated Field Current Waveforms for the 10% PRBS data – detailed view.**

Table 4-4 presents the estimated values for this particular case. The parameters estimated for this test differ slightly from those in the previous case, however the direction of the differential change does not boast the same trend for all parameters concerned. This once again questions the validity of the estimated values due to system insensitivity. Once again the estimated value cannot be compared to the original as the field base quantities were not available due to time constraints.

**Table 4-4 Rotor Estimation - Test 1 - 10 % PRBS.**

Rotor Estimation Test 2 - 10% PRBS						
Parameters [p.u.]		Lfd	Lkd1	Rkd1	Lad	Rfd
	Original	n/a	n/a	n/a	n/a	n/a
	Initial	1	1	1	1	1
Estimated		1.7014	0.0013	14.8884	5.0944	0.9996
% Error		n/a	n/a	n/a	n/a	n/a

The relative performance of the estimator for both cases has not been properly quantified. Thus to gain more insight into how the estimator performed the reader is referred to Figure 4.29 on the following page. This figure gives one an idea of the tracking error of the estimator for both cases. The time frame begins just prior to the disturbance and ends. Consulting the legend in the bottom right-hand corner of the figure, it is clear that the tracking performance on the 1%-PRBS data was better than for the second case.



**Figure 4.29 Percentage error in tracking for both the 1% and 10% cases.**

## 4.8. Results and Conclusions

Parameter identification of the rotor winding forms part of the initial analysis of the synchronous machine, once this has been successfully achieved the next step is to focus on the d and q axis parameters, namely  $X_d$  and  $X_q$ .

Successful parameter estimation first and foremost requires that an accurate representation of the system be created. de Mello [12] proposes a circuit that accurately describes the rotor winding and also includes detail such as damper windings and air-gap inductance. The model of this circuit is shown in Figure 4.5.

MATLAB<sup>®</sup> presents one of two options in terms of platforms to model the given systems. The first is the Simulink Power System Blockset which enables the user to create a network graphically using existing built in models representing elements such as an inductor, a resistor, a derivative block and a host of other functions. The setup of models in this environment is quite straight forward, but when simulating a cost function a number of problems crop up. These are listed below.

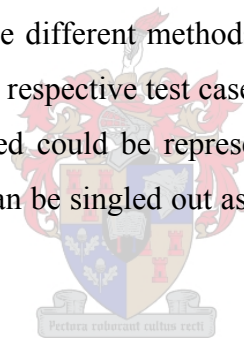
- Implementing the differential equation of a simple RL circuit such as the one shown in Figure 4.10 is simple enough, however to implement the time domain transfer function of the detailed rotor circuit is another challenge altogether. This is partly due to the fact that obtaining the time domain equation is very cumbersome, thus it is preferred to transform to the frequency domain for the simulation stage and then transform back to the time domain for the results. It is easier to implement the filter in MATLAB<sup>®</sup> Script rather than in Simulink, so this tends to beg the question, why bother with Simulink and all its built-in features.
- To ensure accurate simulations one has to lower the tolerance of the solver, however this has a negative effect on simulation time.
- The Blockset contains many different functions that have been written by many different authors, thus some of these blocks are not entirely reliable which can create problems when the system has to solve very stiff problems.
- As discussed in the chapter on the Simulink Power Systems Blockset, nonlinear systems are simulated as current sources. At times this requires the addition of a resistive branch to ensure that two of these types of blocks are not connected in series.

This detracts from the accuracy of the modelling of the circuit, which is a crucial factor contributing to the overall success of the estimation routine.

Despite the problems listed above, two different test cases were carried out using the Simulink Power Systems Blockset as a platform and sections 4.5 and 4.6 focus on the results of these two methods.

The second platform used is MATLAB<sup>®</sup> itself. The transfer function of the rotor circuit is implemented in MATLAB<sup>®</sup> script. The benefits of working in MATLAB<sup>®</sup> script makes it possible to work in the frequency domain which allows one to apply the transfer function of the detailed rotor circuit proposed by de Mello [12]. Section 4.7 gives the derivation of the complex transfer function, the integrity test of the estimator and finally the results from two different test cases.

The estimation results from the three different methods are not as important as the tracking performance of the estimator in their respective test cases. Successful emulation of input data suggests that the parameters obtained could be representative of actual field values. Thus with this in mind the third method can be singled out as being the most successful in terms of signal tracking



Estimating the parameters of the rotor concludes the initial part of steady state evaluation of the machine. Three methods were initially proposed; the first two made use of the Simulink Power Systems Blockset, while the third and most successful test case was carried out in the MATLAB<sup>®</sup> workspace.

## 5. Xd and Xq Parameters

### 5.1. Introduction

The second step in the parameter estimation process is to acquire the steady state synchronous impedances of the machine, namely  $X_d$  and  $X_q$ . Initially however, a feel of how these two parameters affect the power system must be found before estimation can begin. The setup of the system is discussed in the following section; two important considerations are initially put forward here:

- Figure 5.1 shows the two quadrants that the machine will be operating in. The arrows merely suggest the preferred setpoints of the machine, thus representing an active power and its corresponding reactive power value. This gives a rough idea of what operating points will be used, which should be sufficient enough to allow one to gain an insight into how  $X_d$  and  $X_q$  affect machine operation. The real power is controlled by setting the prime mover input to a specific per unit set point. The reactive power and speed of the machine is controlled by the variable load. In order to cover all operating points the machine must be acting in both the 1<sup>st</sup> and 4<sup>th</sup> quadrants respectively.

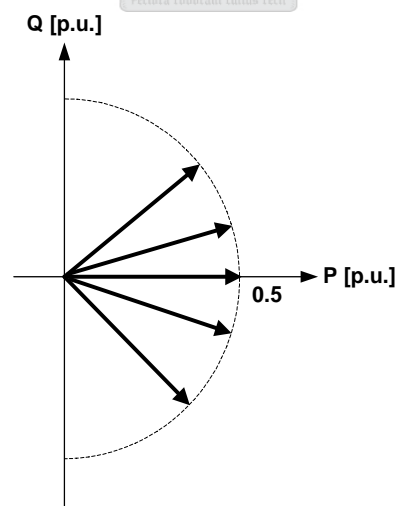


Figure 5.1 P, Q set points for the system.

- The full range of  $X_d$  and  $X_q$  values must be covered in order for this exercise to prove useful. Table 5-1 gives the range over which  $X_d$  and  $X_q$  will be simulated. These



values were obtained from Kundur [1]. The idea is to run the system at different set points and recreate the conditions as they were during testing. Then by running the simulation with every combination of  $X_d$  and  $X_q$  and feeding this through the cost function, one can create the error curve which helps visualise what the estimator will be faced with. This exercise also acts as a screening process with respect to deciding whether a cost function will enable the estimator to converge or whether it will produce local minima. If the latter is the case there is a good chance the estimator will converge to the wrong parameters.

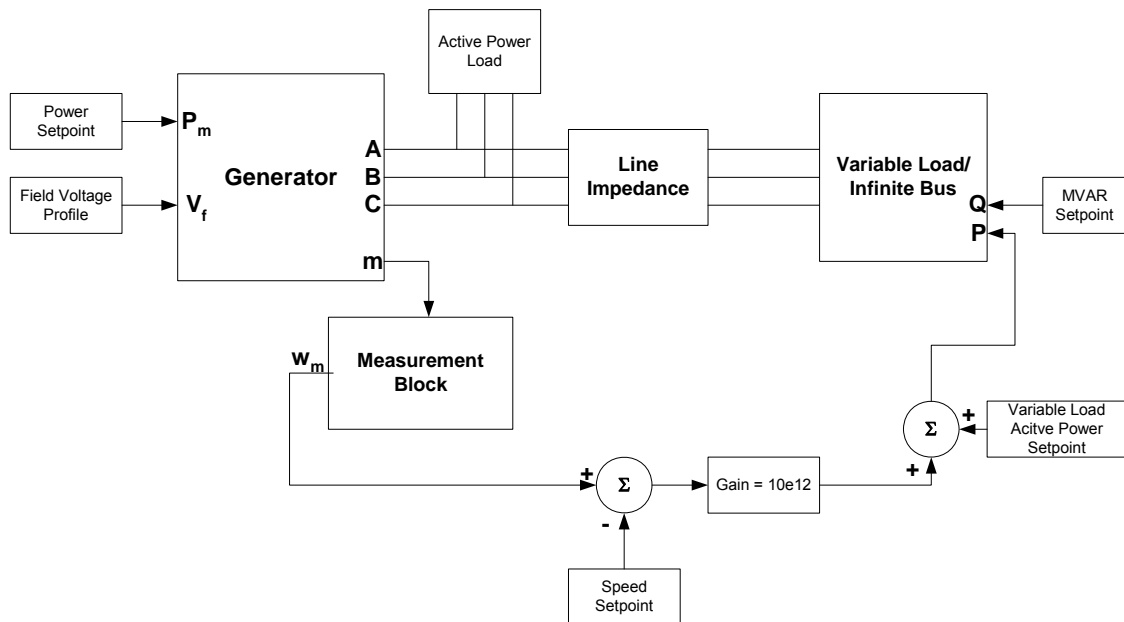
**Table 5-1 Typical values for synchronous reactances.**

Parameters	Hydraulic Units	Thermal Units
$X_d$	0.6-1.5	1.0-2.3
$X_q$	0.4-1.0	1.0-2.3

## 5.2. System Setup

The primary aim at this point of the project is to find the synchronous reactances  $X_d$  and  $X_q$ . In order to achieve meaningful results the Simulink model must be an accurate replica of the setup that was used to generate the field measurements. To achieve this, the plan is to recreate the terminal conditions as they were when testing with the RTDS.

Thus during testing it is important to monitor the terminal voltage, terminal current and the AVR signal. The setpoint and the speed of the machine must also be taken into account. Please refer to Figure 5.2. The first two blocks on the left hand side are the power setpoint and AVR signal blocks. These are fed into the controls of the generator. The generator is then connected to the variable load (approximates infinite bus) via the line impedance and the active load. The active load is included to negate the nonlinear effect previously touched on. The measurement block is also included to monitor the speed of the machine (rotor). This is then compared to the speed.



**Figure 5.2 Power system setup in Simulink.**

The variable load has two settings, namely active and reactive power. If one consults the figure it is noticeable that the error from the speed setpoint is magnified and either subtracted or added to the active power setpoint. Without prime mover control the speed of the machine is only influenced by the active power demand applied by the variable load. If the active power load on the machine is too high for the constant setpoint the machine will slow down and vice versa. The speed of the rotor is thus monitored and compared to a constant speed setting. This is then added to the active power setpoint of the variable load. The effect of this is to ensure that the active load changes its setting to ensure that the speed of the machine remains constant; this is explained topographically in the Figure 5.2. One must remember that by ensuring that the terminal conditions, AVR signal, speed and power setpoint of the machine remain the same as during testing the generator can effectively be isolated from any other influences of the network.

The voltage on the terminals of the machine has to not only be at the correct level but also has to be constant. To achieve this, the AVR is initially included to force the machine to run at the proper terminal conditions. During the estimation process the  $X_d$  and  $X_q$  parameters are changed, and the error is then evaluated by the optimizer. Whenever the system was simulated the AVR would react differently to stabilise the terminal voltage. With a dynamic field voltage one cannot expect the estimator to converge to a solution, thus the plan is to set the field voltage as it was under test conditions. As the entire set of parameters is already known

for this machine, the field voltage can easily be attained through trial and error. With the correct parameters entered, the AVR settings are adjusted so that the terminal conditions are exactly the same as in field measurements. To ensure the same rotor conditions for each case, the entire simulated field voltage profile is stored and then imported for each simulation run.

With the model setup in mind it is important to realise that during these test cases the idea is to isolate the generator, thus removing the effect of any dynamic influences that certain control systems could impose on the system during estimation. This is the primary reason that it is so important to monitor the terminal conditions as well as the AVR and speed signals. Thus with the power setpoint the governor can be removed and simply replaced by a constant. In theory the AVR signal can also be imported straight from the field, however in this case the data was corrupted. The reactive power setpoint is also set as shown in Figure 5.2. As there is no infinite bus component, the variable load is used to approximate it. Thus the active load and the variable load do detract from the model's realism and could introduce potential errors. However with  $P$ ,  $Q$  and  $V_{\text{terminal}}$  set, the terminal current should also be close to testing conditions.

As explained in the chapter on the Simulink Blockset the solver splits the system up into linear and nonlinear blocks. The nonlinear blocks are simulated as current sources. Both the generator and variable load are nonlinear blocks, thus the solver cannot converge to a solution if a current source is placed in series with an inductive load, or another current source. To avoid this problem a small active load is placed between the two elements to secure an alternative path to ground in the event of power flow discontinuities. The line impedance would typically consist of a series inductance for each phase.

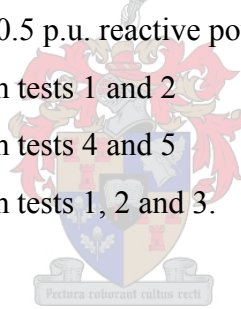
A summary of the initial tasks is stipulated below:

- Network is set up with the emphasis focused on isolating the machine
- Constant input power meant that the governor system could be ignored at the expense of the speed control
- The speed is controlled by setting the variable load's active power demand
- The AVR's dynamic influence counteracts the estimator thus it had to be replaced by a constant field voltage profile

### 5.3. Simulation Test Cases

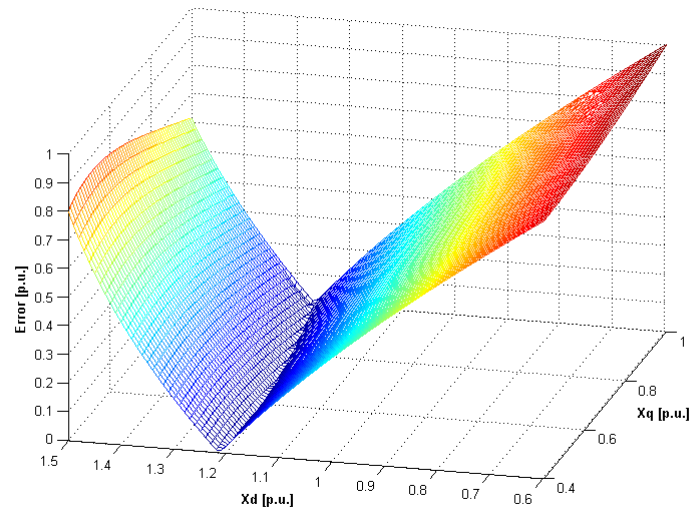
As mentioned in the introduction of this chapter, eight different test situations will be simulated in order to produce a broad spectrum of results. Export and import is with respect to the synchronous machine. It is important to note that for the single power setpoint cases the model will be similar to that shown in Figure 5.2, whilst for the combined case the system in Figure 5.2 is duplicated and run in parallel in the same workspace with the only difference being the setpoints and terminal data. The cost function for the combined cases is simply the addition of the two separate cost functions of each model. The eight conditions are as follows:

- 0.8 p.u. real power exported, 0.256 p.u. reactive power exported
- 0.8 p.u. real power exported, 0.2 p.u. reactive power imported
- 0.8 p.u. real power exported, 0 p.u. reactive power exported
- 0.5 p.u. real power exported, 0.5 p.u. reactive power exported
- 0.5 p.u. real power exported, 0.5 p.u. reactive power imported
- Combined cost functions from tests 1 and 2
- Combined cost functions from tests 4 and 5
- Combined cost functions from tests 1, 2 and 3.



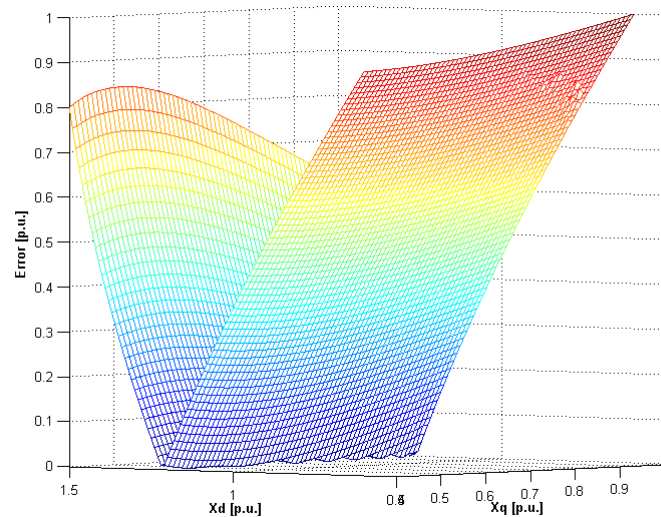
#### 5.3.1. Test 1: 0.8 p.u. active power, 0.256 p.u. reactive power export

The first test involved a system exporting real and reactive power. The shape of the cost function is shown in Figure 5.3. The error between the measured terminal voltage and the simulated value is plotted for all combinations of the synchronous impedances.



**Figure 5.3 Error curve produced by cost function (0.8 p.u. active and 0.256 p.u. reactive).**

What is quite clear from this view is that there is a valley of minima running parallel to the  $X_q$ -axis, roughly in the vicinity of 1.23 (p.u.) on the  $X_d$ -axis. Figure 5.4 shows the side view of the cost function; it also introduces one to the problem of local minima. To ensure that the optimizer converges to the correct answer one must provide the estimator with a curve producing far fewer local minima than in this case.



**Figure 5.4 Error curve produced by cost function - side view.**

### 5.3.2. Test 2: 0.8 p.u. active power, 0.2 p.u. reactive power import

The second test is opposite to the first simulation in terms of reactive power flow; here the machine is importing MVAR's. This specific case introduces the second problem that the estimator will face and this is a surface that has discontinuities as shown in Figure 5.5.

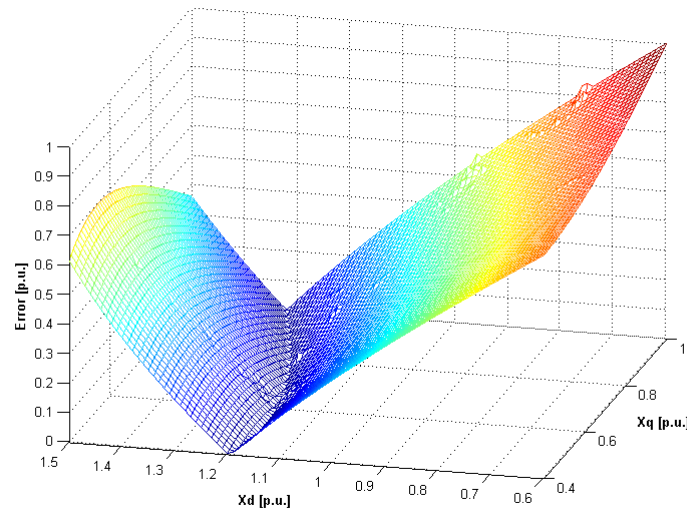


Figure 5.5 Error curve produced by cost function (0.8 pu active, -0.2 pu reactive).

A simple solution to smooth the curve would be to decrease the error tolerances on the solver. This however will create a trade-off between accuracy and simulation time.

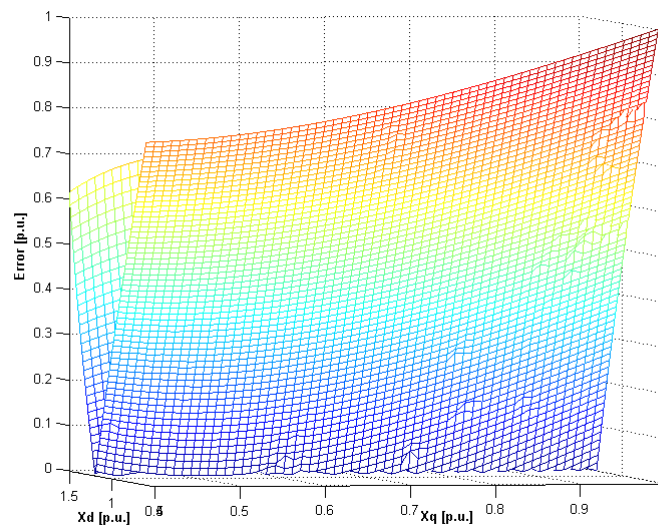


Figure 5.6 Error curve produced by cost function (0.8 pu active, -0.2 pu reactive). Side view.

In Figure 5.6 one can clearly see that there are far too many points of local minima situated along the trough of this curve to expect meaningful results from the optimiser.

### 5.3.3. Test 3: 0.8 p.u. active power, 0 reactive power

The third test proposes a machine only generating active power. As with the previous two test cases the valley of minima runs parallel to the  $X_q$ -axis. Thus using this curve as the sole cost function for the estimator will not produce credible results.

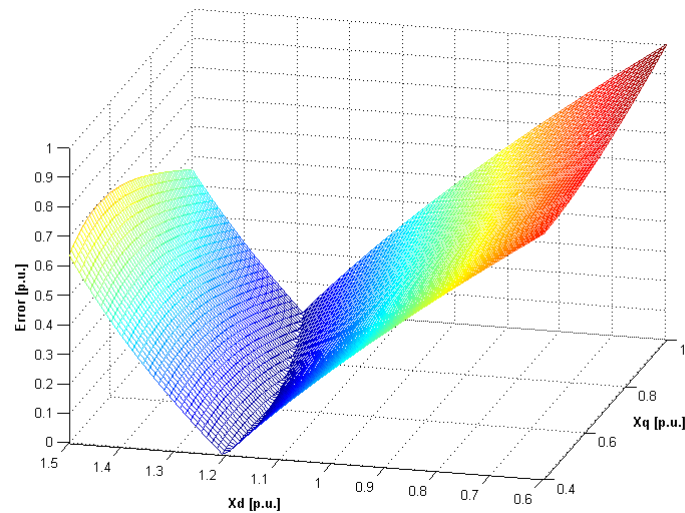


Figure 5.7 Error curve produced by cost function (0.8 pu active, 0 pu reactive).

The problem of local minima pops up yet again as shown in Figure 5.8 and a solution will have to be sought in order for the estimator to converge to a solution.

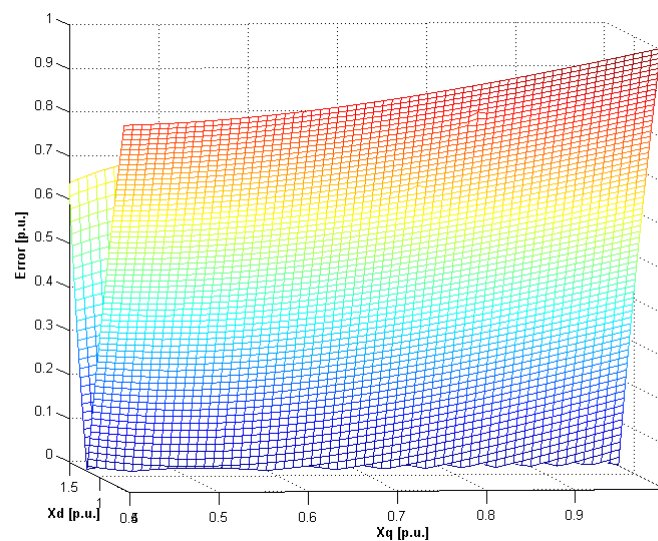


Figure 5.8 Error curve produced by cost function (0.8 pu active, 0 pu reactive). Side view.



### 5.3.4. Test 4: 0.5 p.u. active power, 0.5 reactive power export

An investigation into the error function curve of a machine producing 50% active and reactive power seems necessary in light of the shape of the previous three curves.

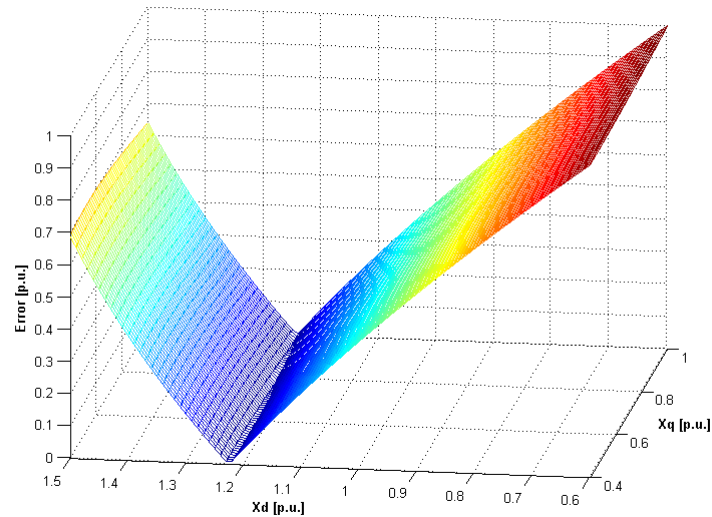


Figure 5.9 Error curve produced by cost function (0.5 pu active, 0.5 pu reactive).

However once again the valley of minima runs parallel to the  $X_q$ -axis and this result doesn't offer much more compared to the previous simulations as witnessed in Figure 5.9.

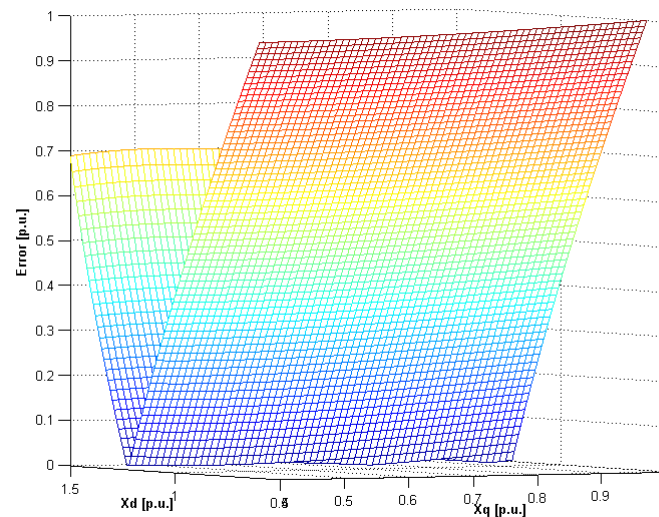


Figure 5.10 Error curve produced by cost function (0.5 pu active, 0.5 pu reactive). Side view.

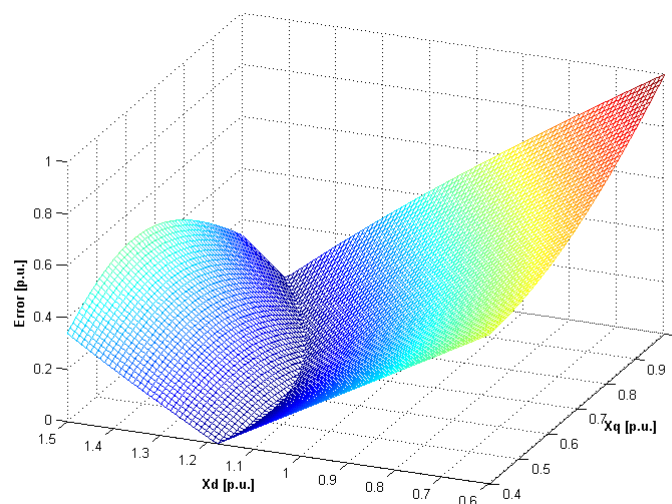


Figure 5.10 shows the side view of the cost function and what is significant about this particular set of results is that the local minima that were so prevalent in the previous curves have now disappeared.

From an estimator's point of view this offers a far better chance of convergence to correct values then with the previous cases.

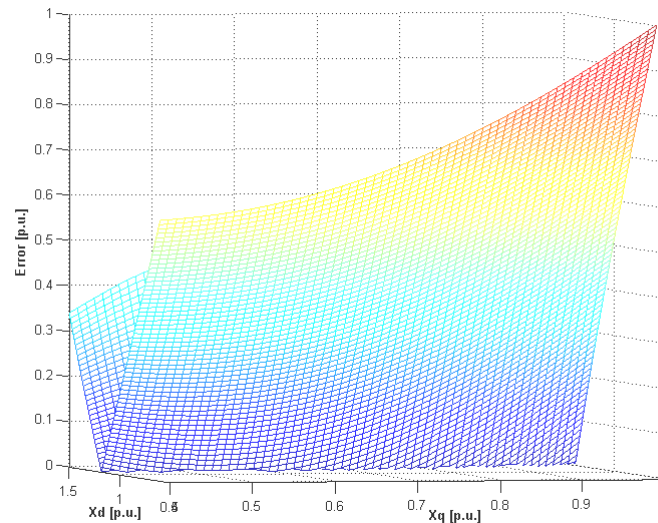
### 5.3.5. Test 5: 0.5 p.u. active power, 0.5 reactive power import

The final individual test involves a machine generating 50% active power while importing 50% reactive power. The valley of minima shows a trend very different from the previous four simulations and is shown in Figure 5.11.



**Figure 5.11 Error curve produced by cost function (0.5 pu active, -0.5 pu reactive).**

Figure 5.12 shows the side view of Figure 5.11 where once again the local minima are present which will only serve to inhibit the estimator's progress.

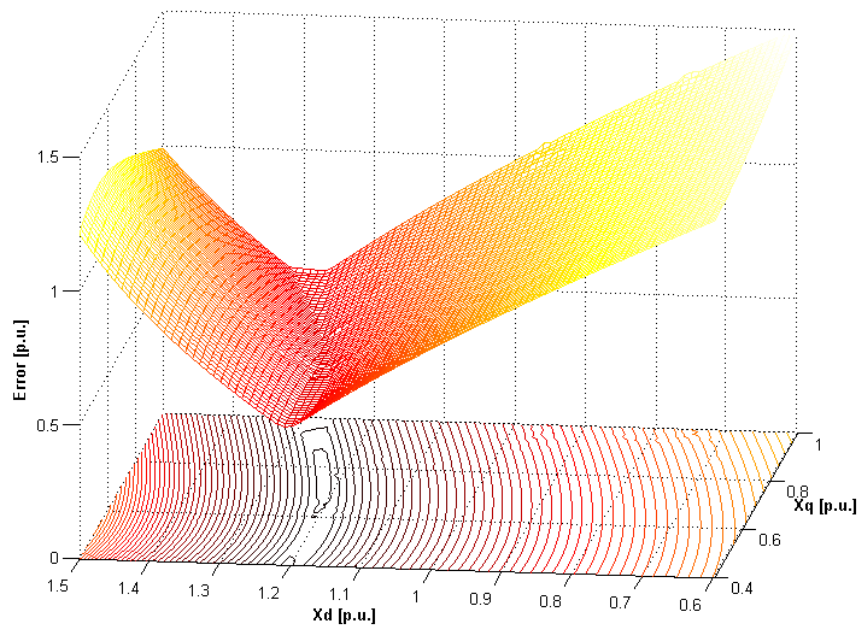


**Figure 5.12 Error curve produced by cost function (0.5 pu active, -0.5 pu reactive). Side view.**

Of the 5 test cases so far, the valley of minima has all generally been orientated in the same direction with the exception of this case. Local minima have also appeared in all cases however test case 4 has produced fewer of these phenomenons. Based on these results it would suggest that the optimiser will struggle to find the absolute minimum using a single cost function, thus the next step is to investigate how the combination of different operating points will influence the shape of the curve and perhaps improve estimation prospects.

### 5.3.6. Test 6: Combined Cost Functions: Tests 1 and 2

Test 6 is the first case where operating points are combined to produce a composite error function. In this particular case the machines simulated in the first two test cases are run in parallel and the error in their terminal voltages is added together to produce the curve shown below in Figure 5.13. What is significant about this curve is that the shape of the cost function is tending towards a cone which will improve the performance of the estimator.

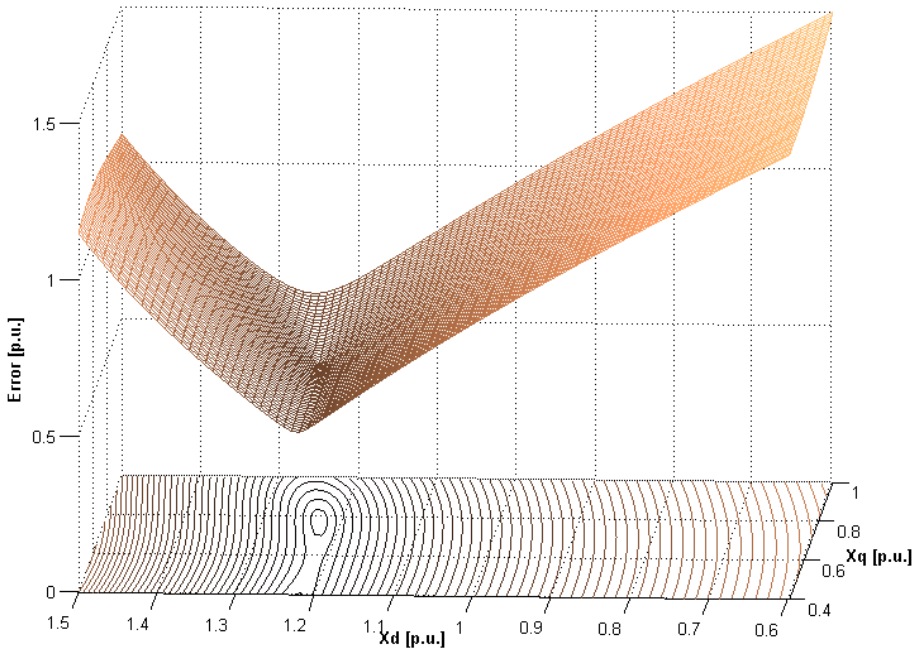


**Figure 5.13 Error produced by combination of cost functions from Tests 1 & 2.**

In Figure 5.13 the contour plot suggests that there is a definite region of minima. Compared to the individual cost function curves the combined case renders a huge improvement in terms of shape of the curve and this ultimately improves the overall performance of the estimator.

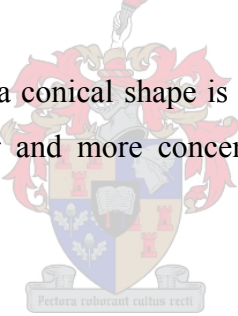
### **5.3.7. Test 7: Combined Cost Functions: Tests 4 and 5**

Case 7 is exactly the same as 6, however the operating points used in simulations 4 and 5 are applied in parallel to produce the cost function in Figure 5.14.



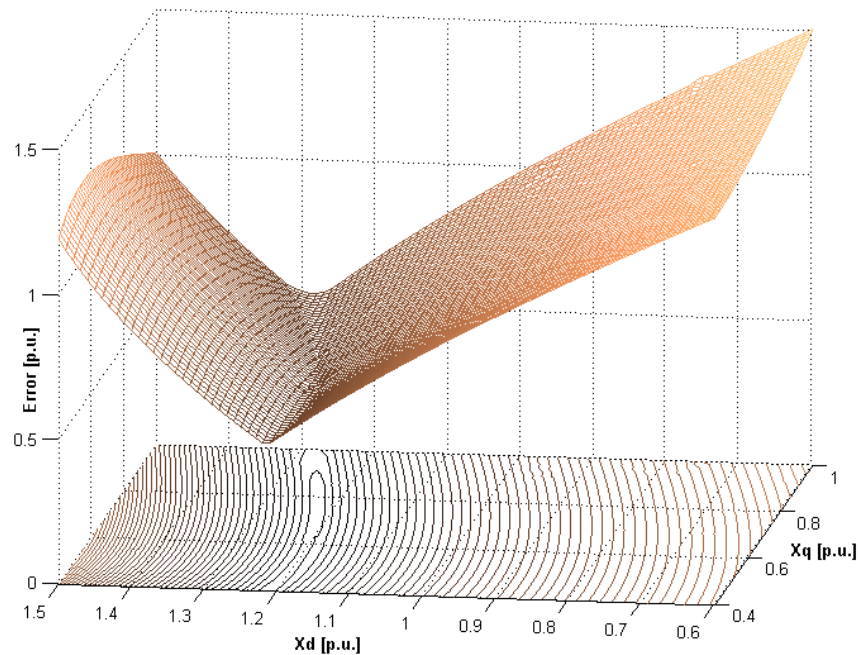
**Figure 5.14 Error produced by combination of cost functions from Tests 4 & 5.**

Once again a curve tending towards a conical shape is produced. Unlike the previous result the area of local minima is smaller and more concentrated which should enable quicker convergence to a result.



### 5.3.8. Test 8: Combined Cost Function: Test Cases 1-3

Test 8 introduces a curve that is created by combining 3 operating points namely the set points followed in cases 1 – 3. Here a single area of minima is produced; however this is not as pronounced as in cases 6 and 7.



**Figure 5.15 Error produced by combining all setpoints in the cost function.**

This concludes the investigation on the type of cost function curve that one can generate from the data available.

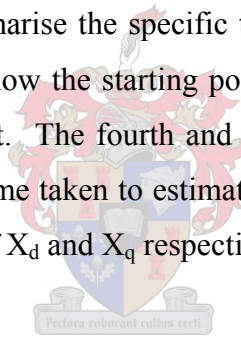
It is quite clear from these figures that simply supplying the estimator with a single individual operating point will not suffice, instead a combination of setpoints should be identified that will ensure the best performance from the estimator.

## 5.4. Estimator performance results

The previous section enabled one to gain insight into how the cost function curve would look for systems operating under particular conditions. On simple evidence supplied to one by the numerous test cases covered in the previous section one could conclude that in the case of a single cost function the estimator would be faced with a valley of local minima, and hence would struggle to find a unique result.

The next logical step was to combine certain operating conditions to form a more definitive curve, and as cases 7 – 9 suggest the valley-shaped curve was transformed into a shallow but much more effective cone. What now follows will be a summary of the performance of the estimator under different condition and with different types of cost functions.

The various tables that follow summarise the specific test condition and performance of the estimator. The first two columns show the starting points for the estimator; the second two columns contain the estimated result. The fourth and fifth columns contain the error at the convergence point and the overall time taken to estimate. Finally the last two columns show the percentage error for the results of  $X_d$  and  $X_q$  respectively.



### 5.4.1. System with single machine exporting MVARs

The table below shows the performance of the estimator in the case where the error function is made up of one operating condition.

**Table 5-2 Estimator performance chart - single set point Q export.**

Estimator performance chart: Single set-point (Export MVARs) Relative Tolerance : 1e-3							
Starting values [p.u]		Estimated values [p.u]		F-Value	Time [s]	Percentage Error [%]	
$X_d$	$X_q$	$X_d$	$X_q$			$X_d$	$X_q$
0.8	0.5	1.209716	0.465037	0.00659678	3367.093	-1.6491332	-40.3798499
0.9	0.6	1.209498	0.564292	0.00000413	721.829	-1.6668033	-27.6548129
1.0	0.7	1.216135	0.663219	0.00001134	608.078	-1.1272729	-14.9719482
1.1	0.8	1.227226	0.759590	0.00006780	576.359	-0.2255418	-2.6166673
1.2	0.9	1.245002	0.877341	0.00000333	514.000	1.2196549	12.4796461
1.3	1.0	1.264122	0.986456	0.00000694	518.656	2.7741427	26.4687409

Judging by the results, one will initially notice that the points of convergence never reached the correct values and instead the estimator was caught in areas of local minima.

### 5.4.2. Two machines in parallel; Import and Export set points

The following three tables show the performance of the estimator using two different operating points to create the cost function. The first table shows the results with the cost function that is not equally weighted. The contribution from each individual error is not evenly spread. With the exception of the first estimation result the optimiser falls within 0.1% of the correct values.

**Table 5-3 Estimator performance chart - combined set point - Q export and import.**

Estimator performance chart: Combined set-point (Import, export MVARs) Relative Tolerance : 1e-3							
Starting values [p.u]		Estimated values [p.u]		F-Value	Time [s]	Percentage Error [%]	
Xd	Xq	Xd	Xq			Xd	Xq
0.8	0.5	1.210149	0.400239	0.00659678	3367.093	-1.6138912	-48.6873061
0.9	0.6	1.229975	0.779718	0.00000413	721.829	-0.0020458	-0.0361860
1.0	0.7	1.229971	0.779660	0.00001134	608.078	-0.0023893	-0.0436513
1.1	0.8	1.230100	0.780516	0.00006780	576.359	0.0081033	0.0661436
1.2	0.9	1.229993	0.779807	0.00000333	514.000	-0.0005938	-0.0247969
1.3	1.0	1.229993	0.779773	0.00000694	518.656	-0.0005830	-0.0291040

The second chart displays the results for the same system, the only difference being that the two machines make equal contributions to the cost function. The curve of this particular function is displayed in Test 6 in the previous section. When one compares the percentage error of the previous table with this example, it is noteworthy that the performance of the estimator has improved, if only slightly.

**Table 5-4 Estimator performance chart - combined set point Q export.**

Estimator performance chart: Combined set-point (Import, export MVARs) Relative Tolerance : 1e-3. Equal weighted.							
Starting values [p.u]		Estimated values [p.u]		F-Value	Time [s]	Percentage Error [%]	
Xd	Xq	Xd	Xq			Xd	Xq
0.8	0.5	1.207663	0.400072	0.00792516	931.750	-1.8160183	-48.7087177
0.9	0.6	1.229984	0.779767	0.00000001	465.063	-0.0013260	-0.0298405
1.0	0.7	1.229988	0.779778	0.00000229	535.141	-0.0009590	-0.0285011
1.1	0.8	1.230000	0.779790	0.00001083	420.390	-0.0000062	-0.0268904
1.2	0.9	1.229993	0.779798	0.00000392	608.969	-0.0005447	-0.0259429
1.3	1.0	1.229991	0.779795	0.00000297	430.266	-0.0007091	-0.0262217

The table below shows the results from the final test of this section. Here the set points are set at 50% active power and 50% reactive power (both importing and exporting). The percentage error in the estimation of the impedances still falls within 0.2%; however these statistics are not as good as in the previous test.



**Table 5-5 Estimator performance chart - combined set point 50% Q export and import.**

Estimator performance chart: Combined set-point (50% Import, export MVARs) Relative Tolerance : 1e-3. Equal weighted.							
Starting values [p.u]		Estimated values [p.u]		Percentage Error [%]			
Xd	Xq	Xd	Xq	F-Value	Time [s]	Xd	Xq
0.8	0.5	1.215760	0.400000	0.01521095	376.546	-1.1576838	-48.7179487
0.9	0.6	1.230099	0.778875	0.00000728	810.312	0.0080181	-0.1441856
1.0	0.7	1.230096	0.778859	0.00000550	754.875	0.0077843	-0.1462794
1.1	0.8	1.230070	0.778829	0.00001201	726.828	0.0057290	-0.1501646
1.2	0.9	1.230091	0.778861	0.00000252	670.032	0.0074194	-0.1459769
1.3	1.0	1.230075	0.778837	0.00000941	523.390	0.0061010	-0.1491344

With the exception of the first single operating condition, the estimator manages to converge to a solution within 1%. The next step is to see how the estimator will manage with a cost function consisting of three operating points.

### 5.4.3. Three machines in parallel; Import, Export and Zero MVAR's

The table below presents information on the performance of the optimiser when presented with a cost function consisting of three inputs. The other change that was made was to decrease the relative tolerance of the solver from 0.1% to 0.001%.

**Table 5-6 Estimator performance chart - combined set point Q export, import and zero.**

Estimator performance chart: Combined set-point (Import, export, zero MVAR's) Relative Tolerance : 1e-5							
Starting values [p.u]		Estimated values [p.u]		Percentage Error [%]			
Xd	Xq	Xd	Xq	F-Value	Time [s]	Xd	Xq
0.8	0.5	1.209148	0.400000	0.00676856	905.719	-1.6952921	-48.7179487
0.9	0.6	1.229928	0.779537	0.00002916	1410.781	-0.0058656	-0.0593367
1.0	0.7	1.229942	0.779546	0.00002174	1540.016	-0.0046815	-0.0582440
1.1	0.8	1.229956	0.779599	0.00001797	1610.703	-0.0035724	-0.0513550
1.2	0.9	1.229996	0.779768	0.00001238	1406.250	-0.0003142	-0.0297813
1.3	1.0	1.229988	0.779736	0.00001238	1637.734	-0.0009392	-0.0338202

As can be seen the effect of this did not significantly improve the results if one compares these to the previous section's statistics. It did however manage to almost double the simulation time, which is by no means a step forward.

When weighing up the results produced by the three different types of cost functions, one can easily conclude that the system running two machines in parallel is the best performer of the group, both in terms of accuracy and simulation time.

Investigations testing the robustness and general performance characteristics of the estimation toolbox fall outside of the scope of this project, and thus limited changes were made to the default settings. Rather the cost function for the specific model was tailored to produce credible results.



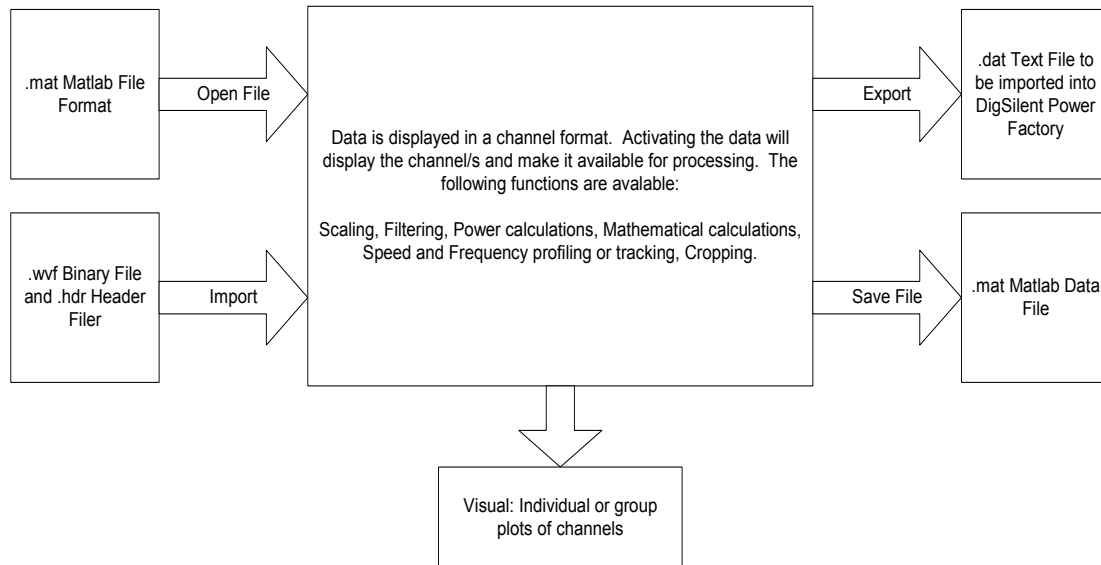
## 6.1. Software Specifications

An effective software tool which empowers the user to import, analyse, manipulate and finally store or export captured data is the topic of this chapter. In order for the measured data to be of good use for estimation purposes it has to first undergo a preparation process. This could involve all or some of the following procedures:

- Data scaling
- Filtering data to clean up noisy signals
- Capture of the speed or frequency profile
- RMS calculations
- Power calculations
- Facility to create new channels
- Import function which recognises .wvf file format
- Export utility enabling the user to create .dat files for use in DIgSILENT® Power Factory
- File handling options
- And other data manipulation options

The requirements stated above enables one to get a vague idea of what the program should be able to accomplish. As this thesis is mainly focused on developing an alternative platform with which to estimate generator parameters, MATLAB® is thus the language of choice when it comes to programming this piece of software.

Writing graphic user interfaces in MATLAB<sup>®</sup> Script is a tedious and often cumbersome job, but despite this the access to the mathematical toolboxes that MATLAB<sup>®</sup> possesses is a major plus factor. Figure 6.1 illustrates the basic functionality of the Data Analyser.



**Figure 6.1** Diagram describing the main functions of the GPIS Data Analyser.

## 6.2. Data Handling: Importing of measured data

The first and most important function of the GPIS Data Analyser is obviously the ability to import data from various sources such as Yokogawa Waveform data, known as the WVF file format. A second format is also catered for, which enables data to be imported from a LabView<sup>®</sup> application.

### 6.2.1. YOKOGAWA ASCII FILE FORMAT

When data is saved with the DL 716 the following is created:

Header file: Filename.HDR

Binary file: Filename.WVF

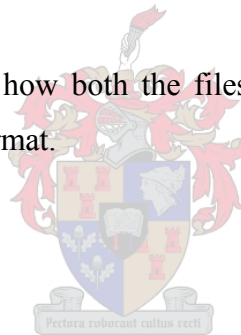
Using a MATLAB<sup>®</sup> formatted read command one must first extract information from the header file, once this is done the WVF file can be successfully imported into the workspace. The format of such a header file is shown in Figure 6.2.

```
//YOKOGAWA ASCII FILE FORMAT
$PublicInfo
FormatVersion      1.01
Model              DL708
Endian             Big
DataFormat         Trace
GroupNumber        1
TraceTotalNumber   4
DataOffset         0

$Group             1
TraceNumber        4
BlockNumber        1
TraceName          Ch1          Ch2          Ch3          Ch4
BlockSize          10032        10032        10032        10032
VResolution        1.56250e-04   3.12500e-03   6.25000e-04   1.56250
e-05
VOffset           0             0             0             0
VDataType          IS2          IS2          IS2          IS2
VUnit              V            V            V            V
VPlusOverData     ?            ?            ?            ?
VMinusOverData    ?            ?            ?            ?
VIllegalData      -32768  -32768  -32768  -32768
VMaxData          32640          32640          32640          32640
VMinData          -32640  -32640  -32640  -32640
HResolution        1.00000e-06   1.00000e-06   1.00000e-06   1.00000
e-06
HOffset           0             0             0             0
HUnit              s            s            s            s
Date              97/12/04       97/12/04       97/12/04       97/12/04
Time              16:40:33       16:40:33       16:40:33       16:40:33
```

**Figure 6.2 Format of Yokogawa Header file.**

The flowchart in Figure 6.3 shows how both the files are read and used to save the final product in a MATLAB<sup>®</sup> structure format.



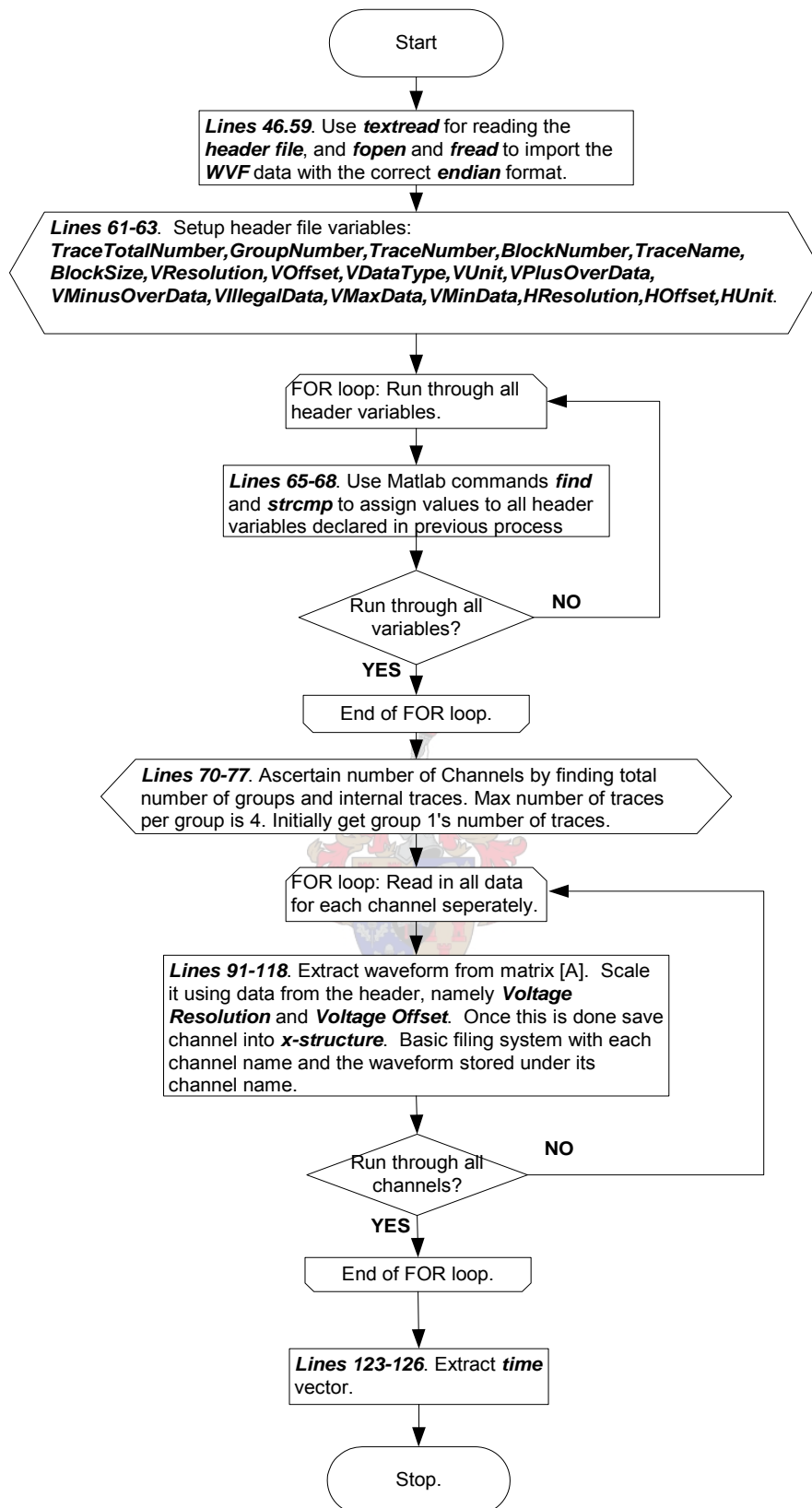


Figure 6.3 ImportFile.m flowchart: WVF importing procedure.

The MATLAB<sup>®</sup> command ‘**Textread**’ is used to import formatted text from the header file which is used to scale and eventually store the channel data into a structured array in the

MATLAB<sup>®</sup> workspace. The following variables found in the header file are highlighted as the most important for this purpose:

- **"Endian"** - The high-low byte order for data differs depending on CPU. The "Endian" is provided to absorb this difference.

680XX (Motorola): Big Endian

80X86 (Intel): Little Endian

These two are "Endian" modes. If the "Endian" mode of a saved file differs from the computer to process the data, reverse the high and low order.

- **"DataOffset"** - This shows how many bytes there are from the head of the data file saved in binary from (with extension "WVF") to the start of the captured data. If the captured data begins immediately in the file, "DataOffset" is "0".
- **"BlockSize"** - This shows the number of data points in each block of trace names. Note that this size means the number of data points and NOT the number of total bytes.
- **"VResolution"** - This indicates the coefficients of Y axis conversion equations for each trace. The values of the Y axis (values of the vertical axis) can be determined using the following conversion equation:

$$Y \text{ axis value} = "VResolution" \times \text{Data value} + "VOffset"$$

- **"VOffset"** - This indicates the Y axis offset values for each trace.
- **"HResolution"** & **"HOffset"** - This indicates coefficients of the X axis conversion equation for each trace.

$$X \text{ axis value} = "HResolution" \times (\text{data no.} - 1) + "HOffset"$$

The data number starts at 1 and "HOffset" can be calculated from the equation by assigning 0 to the "X axis value" and the trigger point no. to the "data no."

The information above was taken from the technical document on the structure of files created by the DI series recorders. The rest of this manual can be found in the appendix. Figure 6.4 illustrates the format of the data file. As shown in the flowchart all the data is read in simultaneously using the correct endian format. This ensures that the byte ordering is correct. A FOR loop is then instituted to run through the full length of data and using the block size value each channel is extracted, scaled and saved into a structured array in the MATLAB<sup>®</sup> workspace.

Block format style

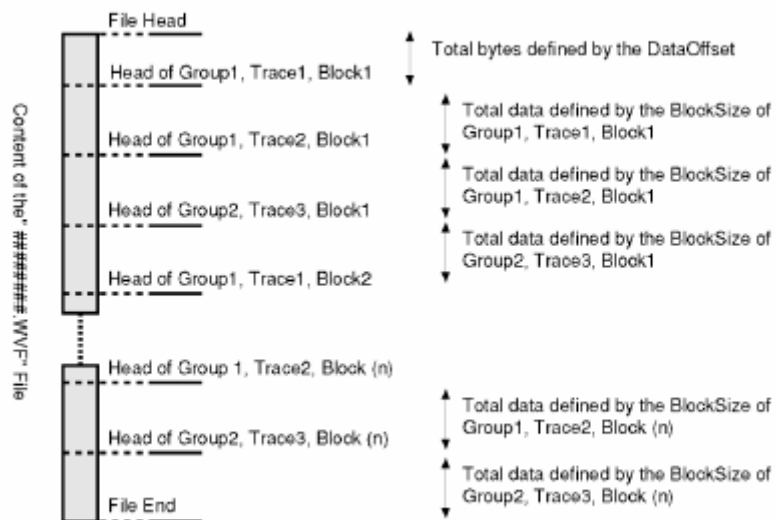


Figure 6.4 WVF file format.

### 6.2.2. LabView file format

Table 6-1 explains the layout of the file that has to be imported from the LabView<sup>®</sup> application.

**Table 6-1 Format of LabView file.**

Structure	Variable	Sub-variable	Representation	Length	
Header	Header length		Longint	4 bytes	
	Header length		Longint	4 bytes	
	Memo	Memo data length		LongInt	4 bytes
		Memo data		Char	Memo data length bytes
	Sample frequency		Single precision floating point	4 bytes	
	Sample frequency index		Unsigned word	2 bytes	
	No. of Scans		Unsigned LongInt	4 bytes	
No. of Channels		Unsigned LongInt	4 bytes		
Data	Scan values	No. of Channels by No. of Scans	Single precision floating point	4 bytes	

The MATLAB<sup>®</sup> command '**fread**' is used instead of '**textread**' to acquire header information, before channel data is read in. Importing this type of file as opposed to WVF file types is made easier by the fact that it is only necessary to read in a single file as both header and data information is located in the same file.

Figure 6.5 presents the flowchart detailing the method to read in the file format explained on the previous page in Table 6-1.

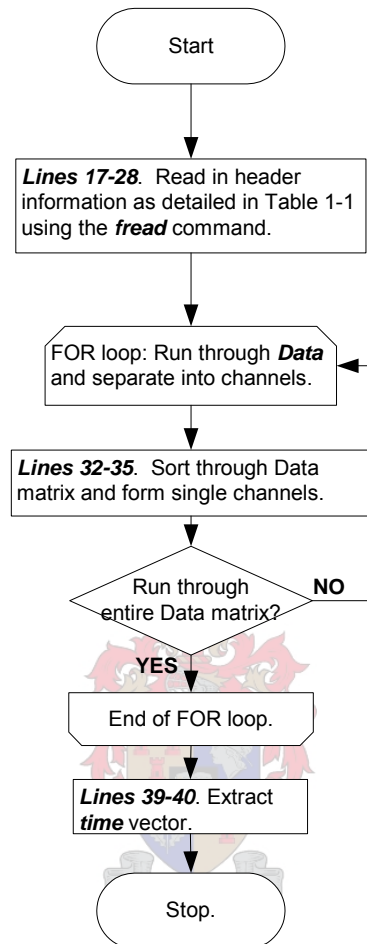


Figure 6.5 ImportFile.m flowchart: Procedure to import LabView data.



### 6.2.3. Data structure in the MATLAB® Workspace

Figure 6.6 shows how the data is stored once it has been imported into the MATLAB® workspace.

Name	Size	Bytes	Class
x	1x1	70354596	struct array
Grand total is 8794136 elements using 70354596 bytes			
x =			
CH1:	[676471x1]	double	
CH2:	[676471x1]	double	
CH3:	[676471x1]	double	
CH4:	[676471x1]	double	
CH5:	[676471x1]	double	
CH6:	[676471x1]	double	
CH7:	[676471x1]	double	
CH8:	[676471x1]	double	
CH9:	[676471x1]	double	
CH10:	[676471x1]	double	
CH11:	[676471x1]	double	
CH12:	[676471x1]	double	
CH13:	[676471x1]	double	

Figure 6.6 Structure used to store channel data in MATLAB® workspace.

A structured array is best suited for this purpose, and as shown by the diagram the filing system is both logical and easy to access. The time vector is left out of the array due to the possibility of future cropping of data. The reason for this is that MATLAB® will produce an error when given two vectors of different lengths to plot.

Initially when data was cropped and saved one would have had to crop all channels the same length in order to skirt the vector-length issue, however it was decided in order to add more flexibility, that the time vector be excluded. Thus every time a channel is processed or displayed a time vector of the same dimensions is created to avoid errors.

### 6.3. Graphic User Interface (GUI) Layout

To design the optimum layout one has to imagine how the data processing is to be handled and then evaluate how the layout of buttons, graphs etc can be done to speed up the process and generally make this application as user-friendly as possible. Figure 6.7 explains what is required to be in the GUI.

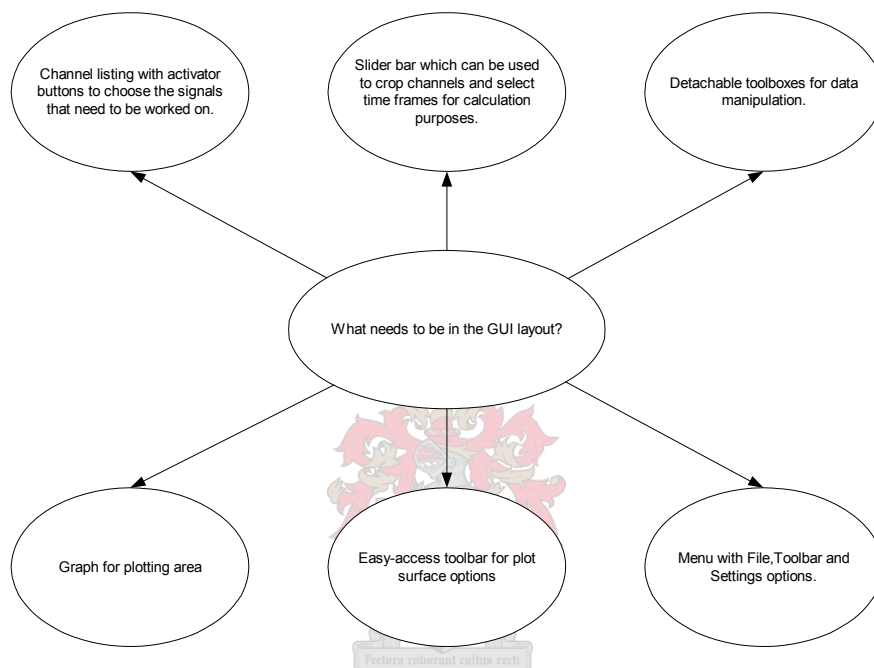


Figure 6.7 Chart detailing features of the GUI layout.

With Figure 6.7 in mind the following layout was decided upon:

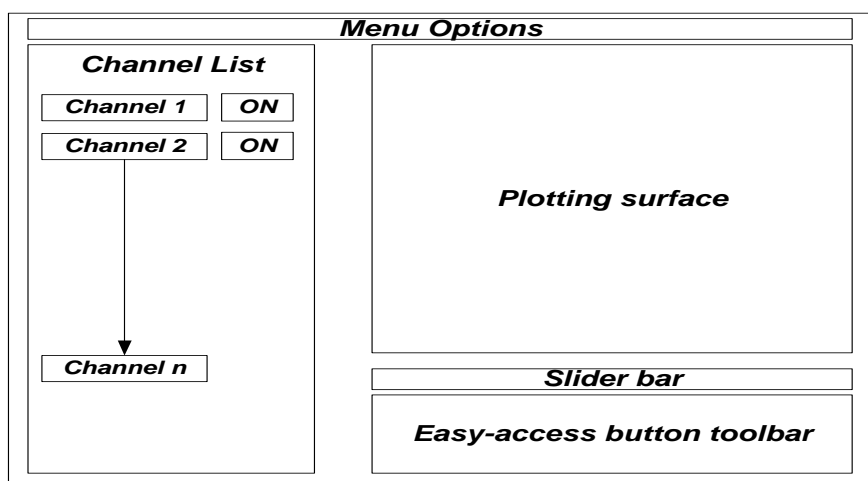


Figure 6.8 GUI layout: Main window.

The final product is shown below in Figure 6.9:

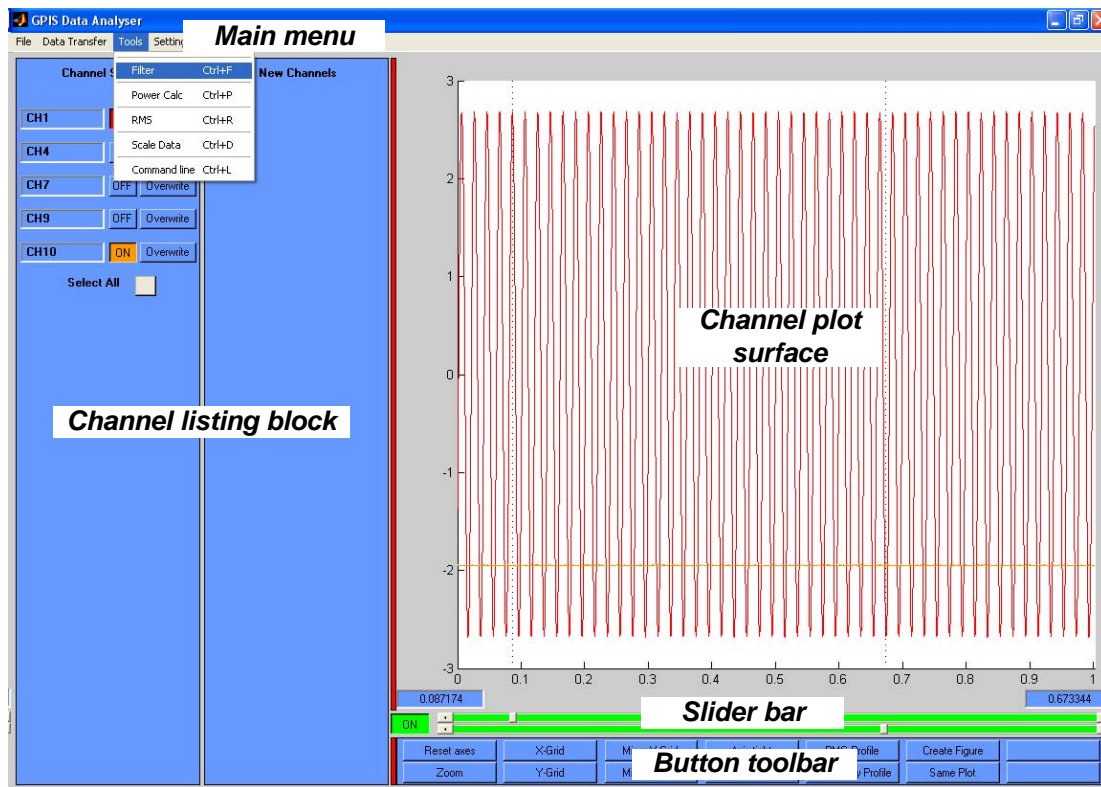


Figure 6.9 Main GUI Window.

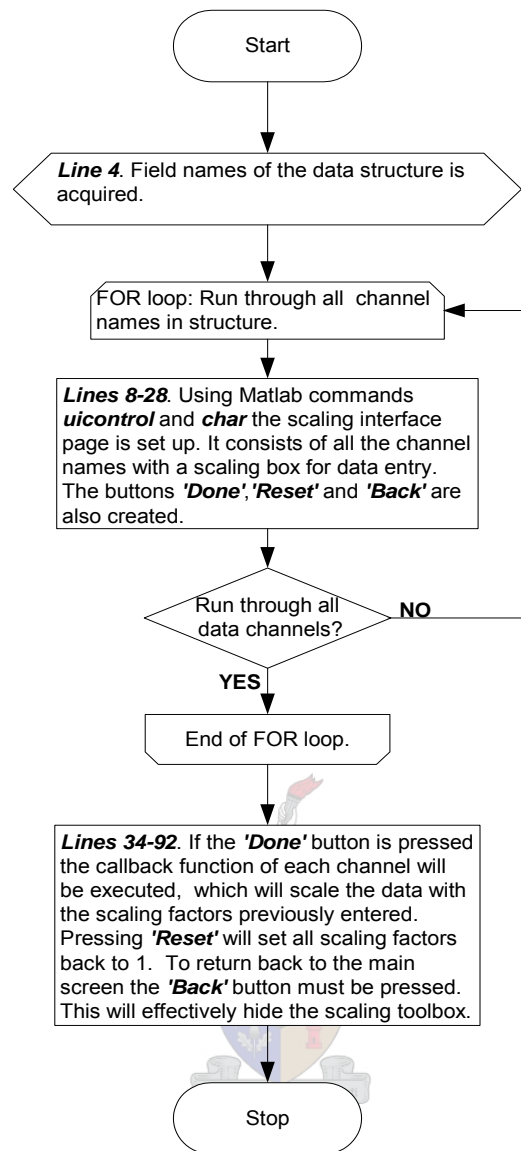
Now that the main layout has been explained, the individual toolboxes will be introduced and scrutinised in detail in the following section.

## 6.4. GPIS Data Analyser - Toolboxes.

The GPIS Data Analyser is outfitted with a number of useful functions, which can be used to crop, scale, and filter and manipulate raw data. This section will examine each tool individually.

### 6.4.1. Scaling Toolbox

The scaling of the data is essentially a simple, but important task in the entire process. The flowchart below in Figure 6.10 shows how the scaling toolbox is created.



**Figure 6.10** Scaling.m-flowchart explaining program code.

Figure 6.11 shows the Scaling Toolbox interface. It is a basic setup with channel names followed by edit boxes and finally option buttons at the bottom of the screen. The option buttons and their functions were discussed in the previous flowchart.

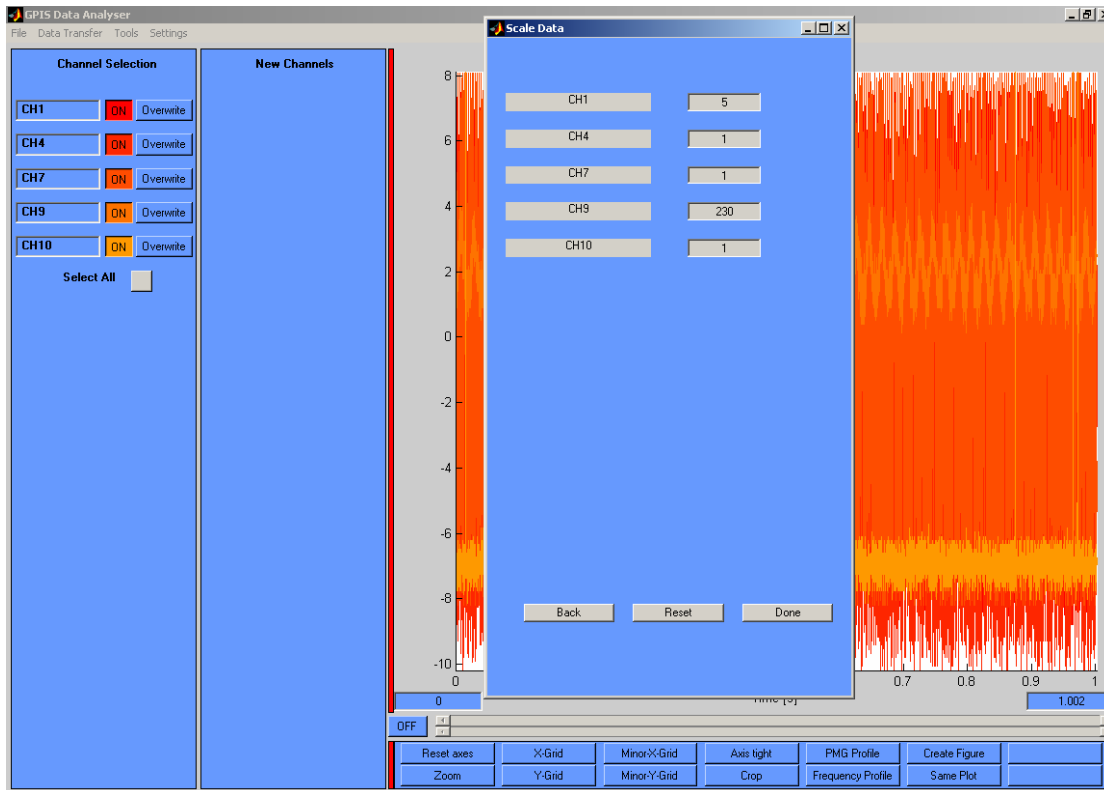


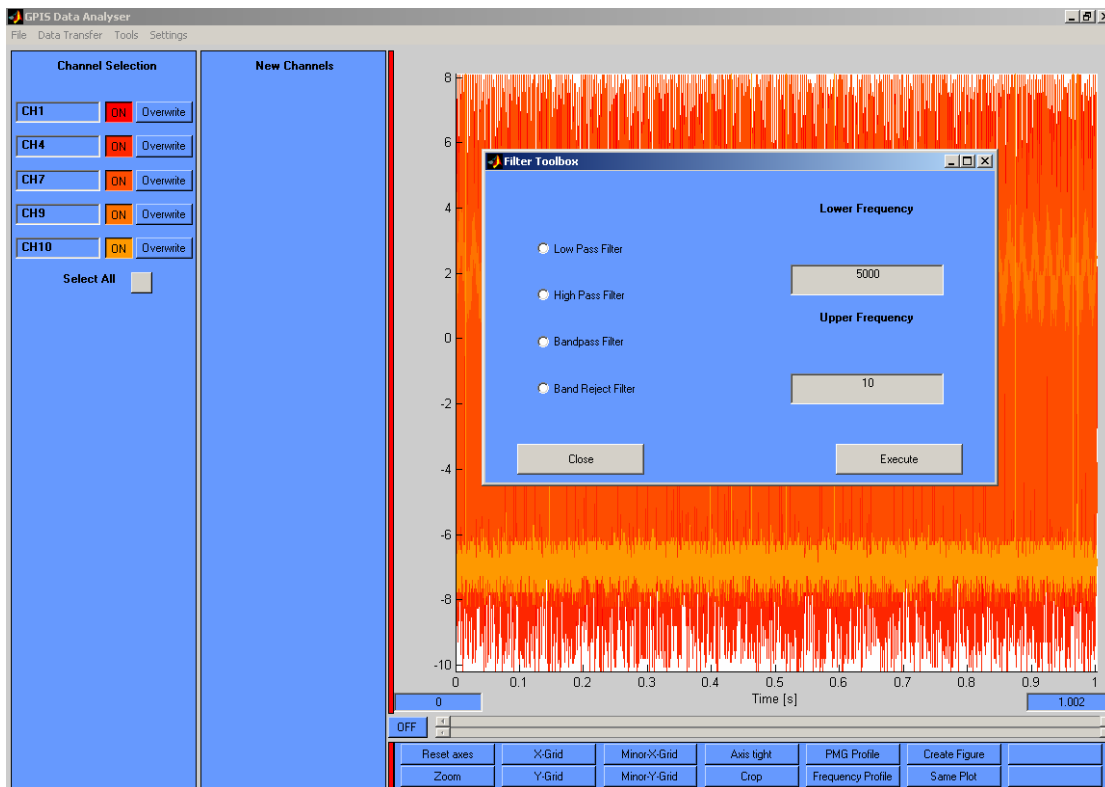
Figure 6.11 Scaling toolbox layout.

## 6.4.2. Filter Toolbox

The filter toolbox is primarily used to clean up noisy signals. This toolbox boasts four different filter modes:

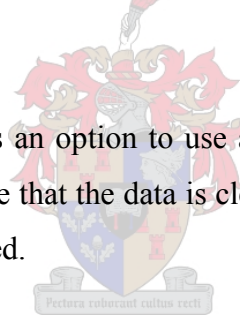
- Low pass filter
- High pass filter
- Band pass filter
- Band stop filter

Figure 6.12, showing the Filter Toolbox is given below. On the left are the different types of filters offered. The required filter mode can be selected by clicking on the corresponding radio button. The 3dB point of the filter can be set in the 'Lower Frequency' edit box; this is the only parameter needed for low and high pass filtering. Obviously for the other two modes the 'Upper Frequency' must also be set. All cut-off values are entered in as Hertz.



**Figure 6.12 Filter toolbox layout.**

The Frequency Tracker Toolbox has an option to use a pre-filter before performing its own calculations. This is simply to ensure that the data is clean for optimum performance, and for this purpose a low pass filter is needed.



The flowchart for the filter toolbox code is shown below in Figure 6.13. Both the dedicated filtering and the pre-filter code is handled in one script file, this is all clarified in the diagram.

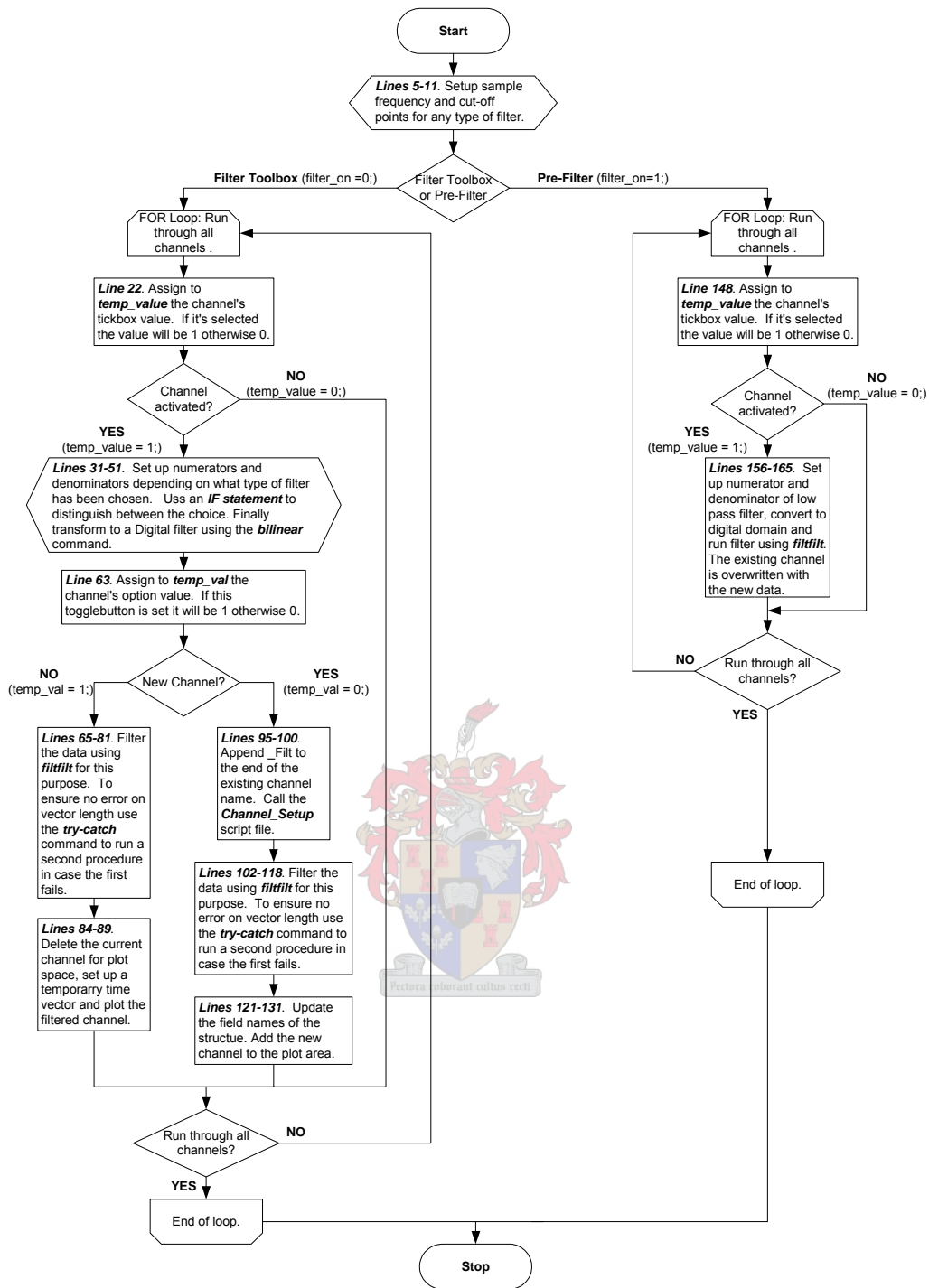


Figure 6.13 Filter\_Program.m flowchart: program code.

### 6.4.3. The Frequency (Speed) Tracker Toolbox: Interpolation Method

The frequency tracker toolbox's aim is to capture the frequency profile (or PMG speed) of a selected signal. The RMS Toolbox also has a feature to include frequency tracking in its calculations, thus this code performs a similar role to the filtering toolbox in terms of pre-calculation.

The frequency profile and PMG speed can be calculated by clicking on the corresponding button in the toolbar below the plot space. There is an options menu as shown below in Figure 6.14 where one can select what tracking method one would like to use. The pre-filter option is also included here.

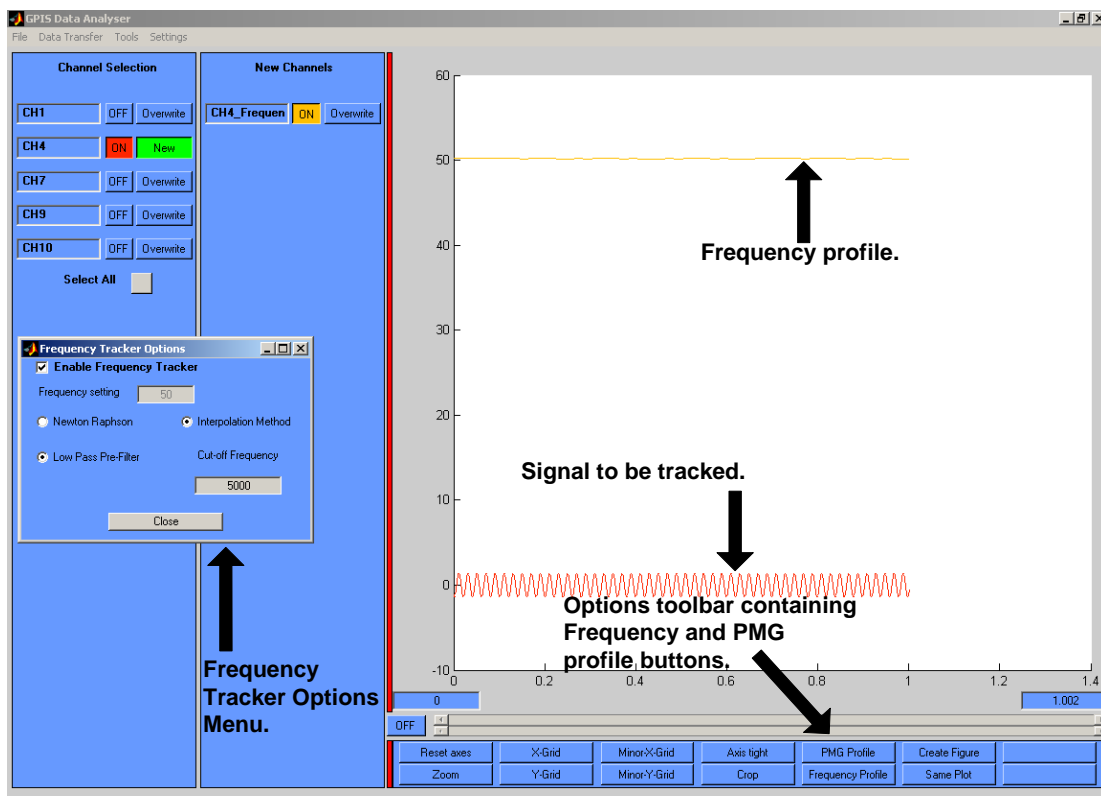


Figure 6.14 Frequency toolbox layout.

The flowchart on the following page in Figure 6.15 page shows how the code is implemented in MATLAB<sup>®</sup> Script. The program initially checks to see for what purpose the tracking is being used for, then it checks if the pre-filter is needed and then finally calculates the frequency profile for the selected signal.



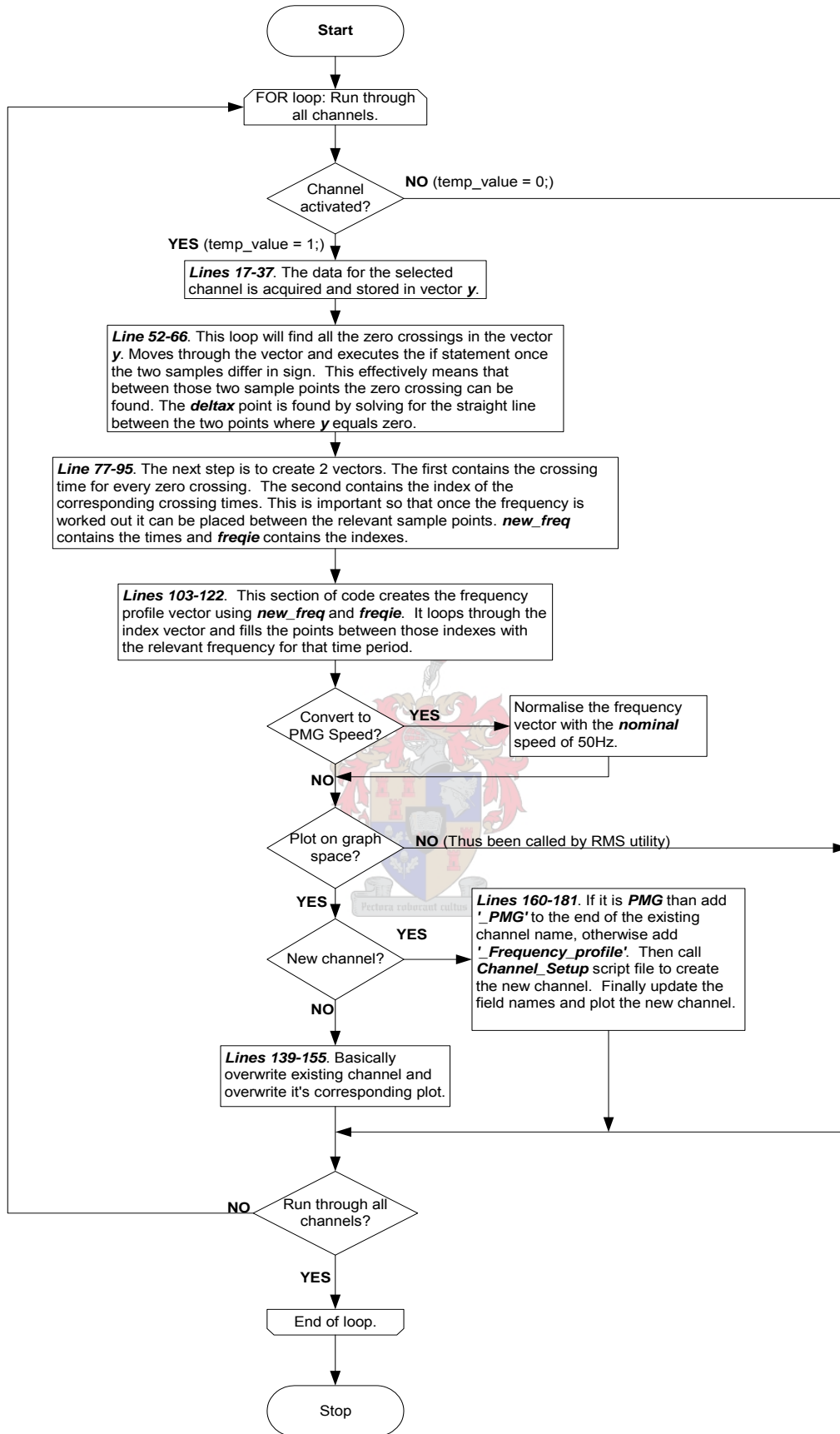
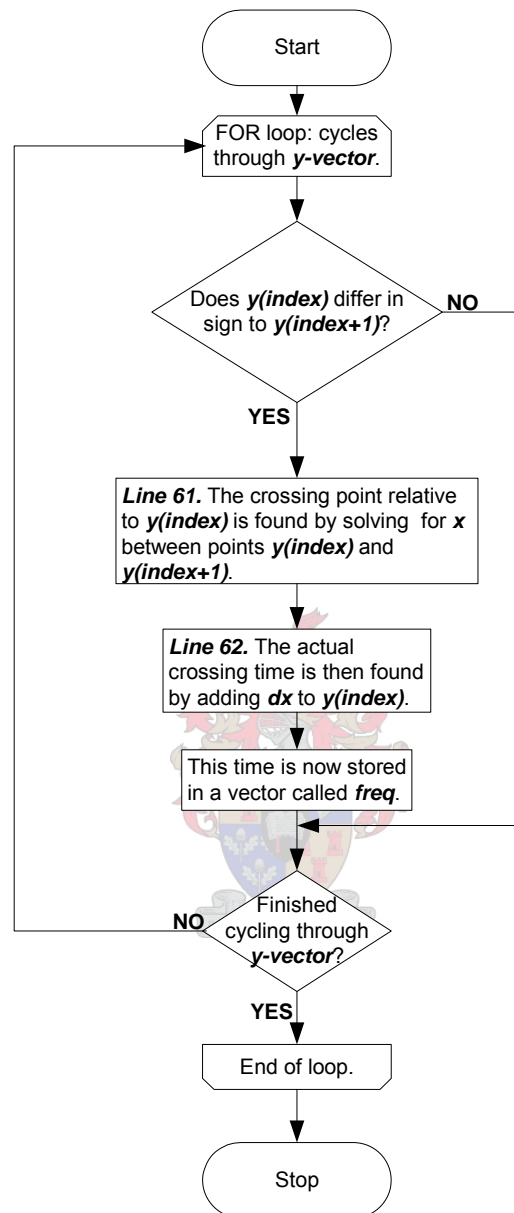


Figure 6.15 Freq2.m flowchart: program code.

As stated earlier there are two different techniques which one can use to track the frequency, this section discusses the Interpolation method. Figure 6.16 presents the flowchart explaining how the zero-crossing points are found and their associated times stored.



**Figure 6.16 Interpolation process flowchart: finding the zero crossings.**

For clarification purposes, Figure 6.17, on the following page depicts a sine wave plotted in point form. The signal is representative of the data that one might encounter during processing. The boxed region shows a typical crossing point region. Initially the algorithm cycles through the data until two consecutive points differ in sign.

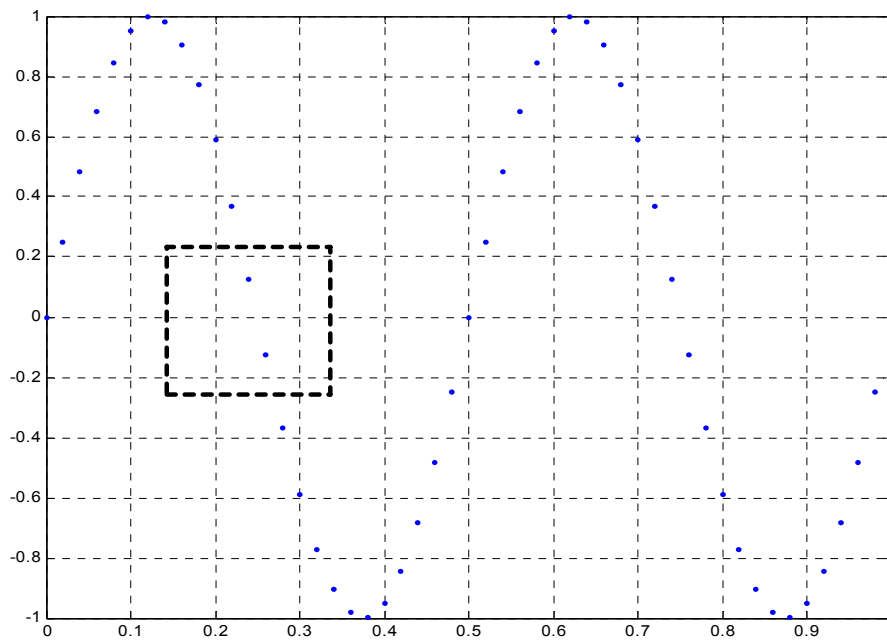


Figure 6.17 Plot of a sine wave explaining zero crossing detection procedure.

The next step is to find the crossing point relative to the first data point. Figure 6.18, below shows how this is achieved. Once  $dx$  is found the actual time of the zero-crossing is easily calculated by adding  $dx$  to the value of  $y(\text{index})$  at that stage. The final step in this loop is to put each time stamp at its corresponding index in a new vector called **freq**.

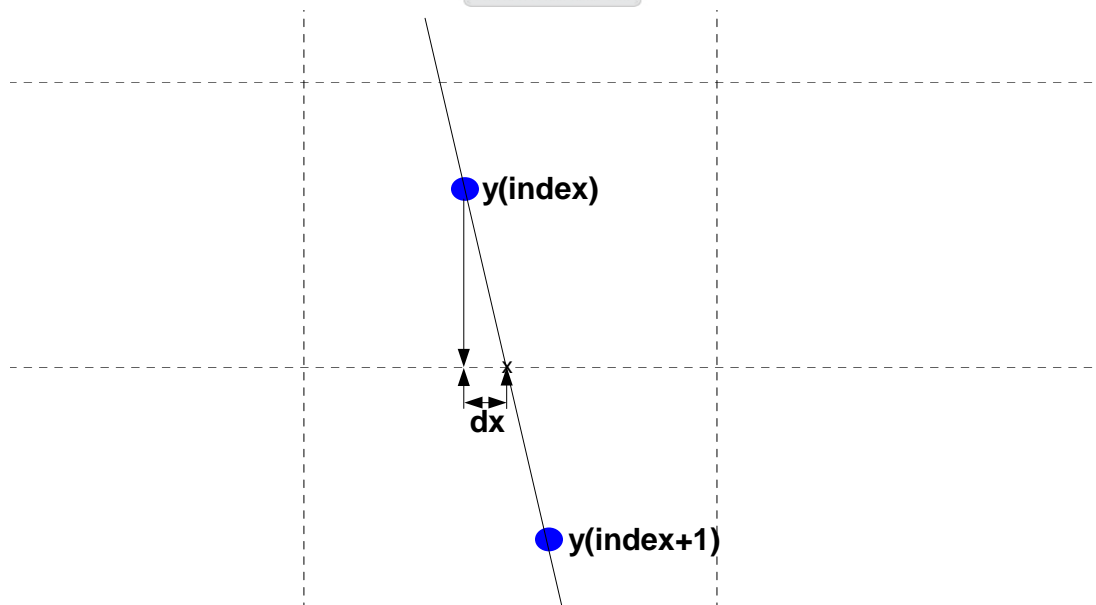


Figure 6.18 Time detection process.

Finally, the entire procedure is summarised graphically in Figure 6.19:

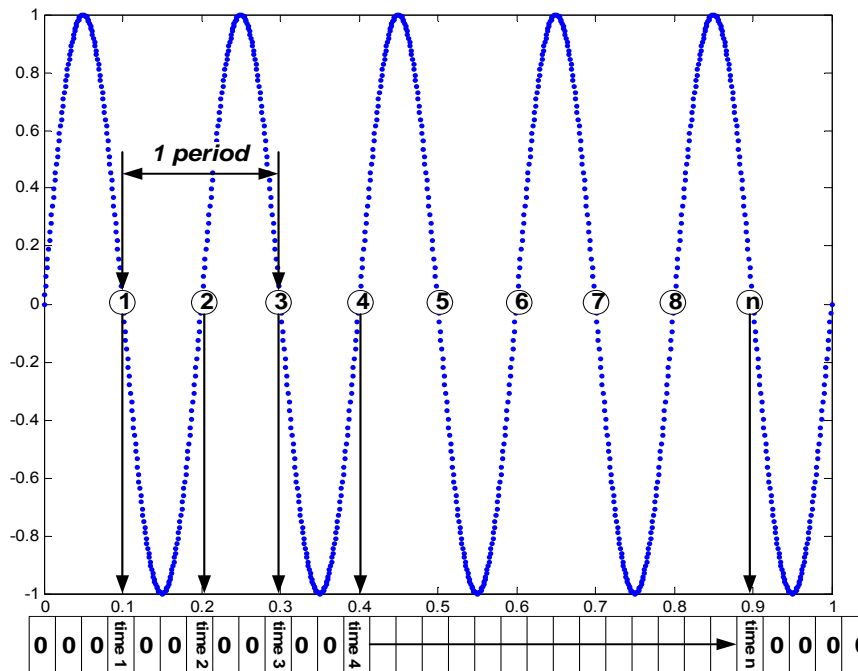


Figure 6.19 Summary of zero crossing detection process.

The next step is to put the *freq* vector into a more compact form. Once this is done, a loop is used to cycle through both the *time stamp* vector and *index* vector starting at the third crossing point. The frequency is calculated between three crossing points to cover an entire period as shown in the previous figure. This frequency is then placed at all points in the vector between the  $n^{\text{th}}$  and  $n+1$  index point. Figure 6.20 and Figure 6.21 clarifies this process in more detail.

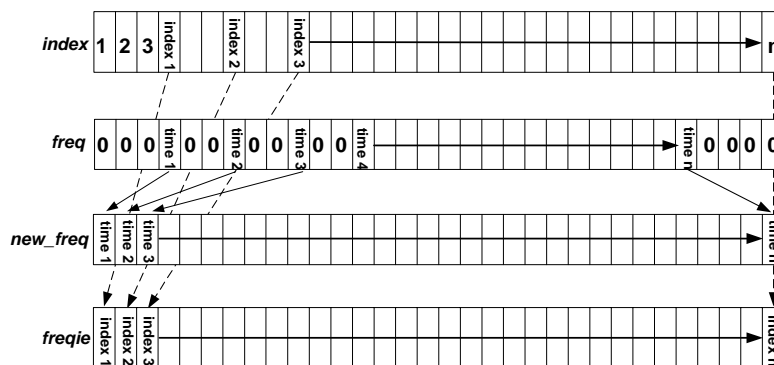


Figure 6.20 Compression of vectors.

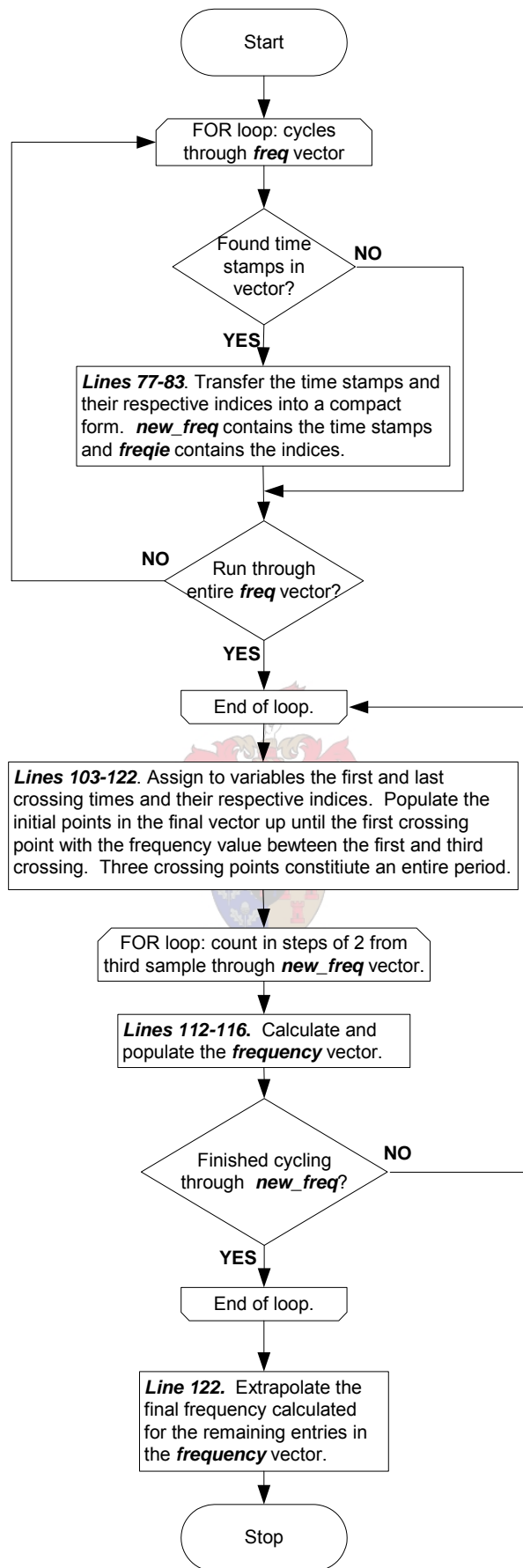


Figure 6.21 Interpolation procedure: final phase of process.

#### 6.4.4. The Frequency (Speed) Tracker Toolbox: Newton-Raphson Method

A second tracking technique is also available which makes use of the Newton-Raphson method to find the zero-crossing.

Initially a ten point buffer is set up effectively forming a window which scans through the entire piece of data. As soon as the crossing point is found to be located somewhere between the middle two sample points in the buffer (5<sup>th</sup> and 6<sup>th</sup>), the window is halted. A 2<sup>nd</sup> order curve fit is then performed on all data point within the window, producing a polynomial approximation.

The Newton-Raphson method is now implemented to find the zero-crossing. The iterative loop continues until the difference between consecutive iterations falls within a value deemed to be small enough to produce an accurate result. Figure 6.22 shows how the moving window is implemented.

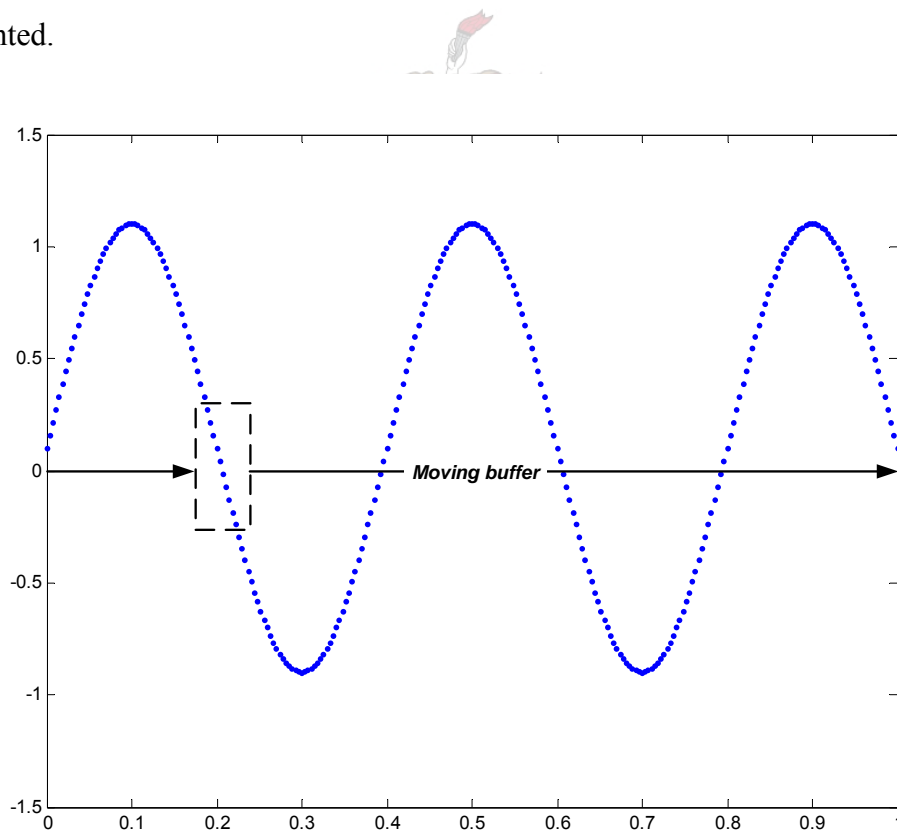
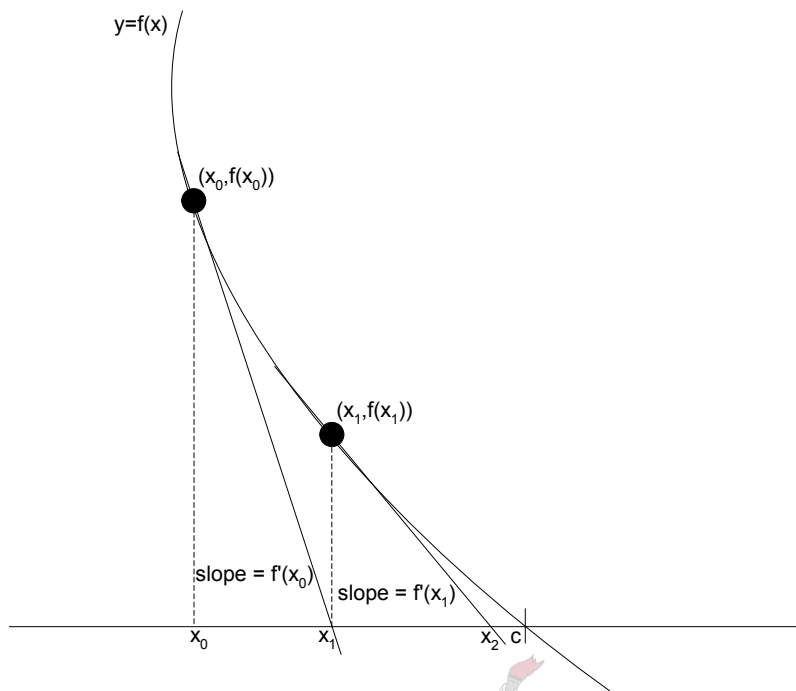


Figure 6.22 Moving window used in Newton Raphson tracking method.

Figure 6.23 gives some insight into how the Newton-Raphson method is implemented.



**Figure 6.23 Newton Raphson method.**

An initial x-value is first assumed; in this case it will always be the 5 time sample in the moving buffer, as the crossing lies somewhere between sample points 5 and 6. To solve for  $x_1$  the gradient of the tangent to point  $x_0$  must be calculated, and it is from this equation that the value of  $x_1$  can be then isolated and solved. The following equation solves the iteration,

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \quad (6-1)$$

The flowchart in Figure 6.24 on the following page explains the entire procedure for the Newton-Raphson frequency tracker utility for more information on the Newton-Raphson method the interested reader is referred to the work of R.L. Finney *et al.* [17].

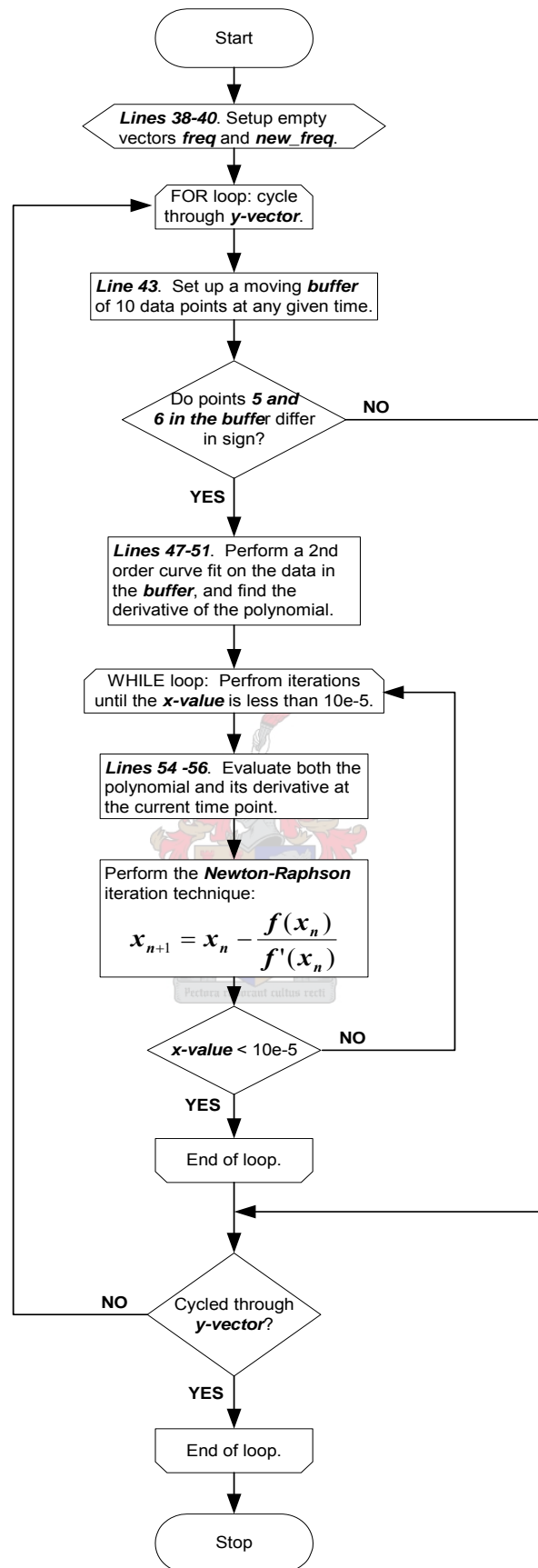


Figure 6.24 Freq4.m flowchart: Newton Raphson implemented in code.



### 6.4.5. Root Mean Square (RMS) Toolbox

This toolbox calculates the RMS value of a signal, which is useful in simplifying the data when it comes to the estimation process. Figure 6.25 shows how the RMS toolbox is presented in the program. The toolbox details the process from start to finish. In this case the signal is initially put through a low pass filter to clean the data. The frequency tracker is then employed so that variations in frequency can be taken into account when calculating the RMS of the waveform.

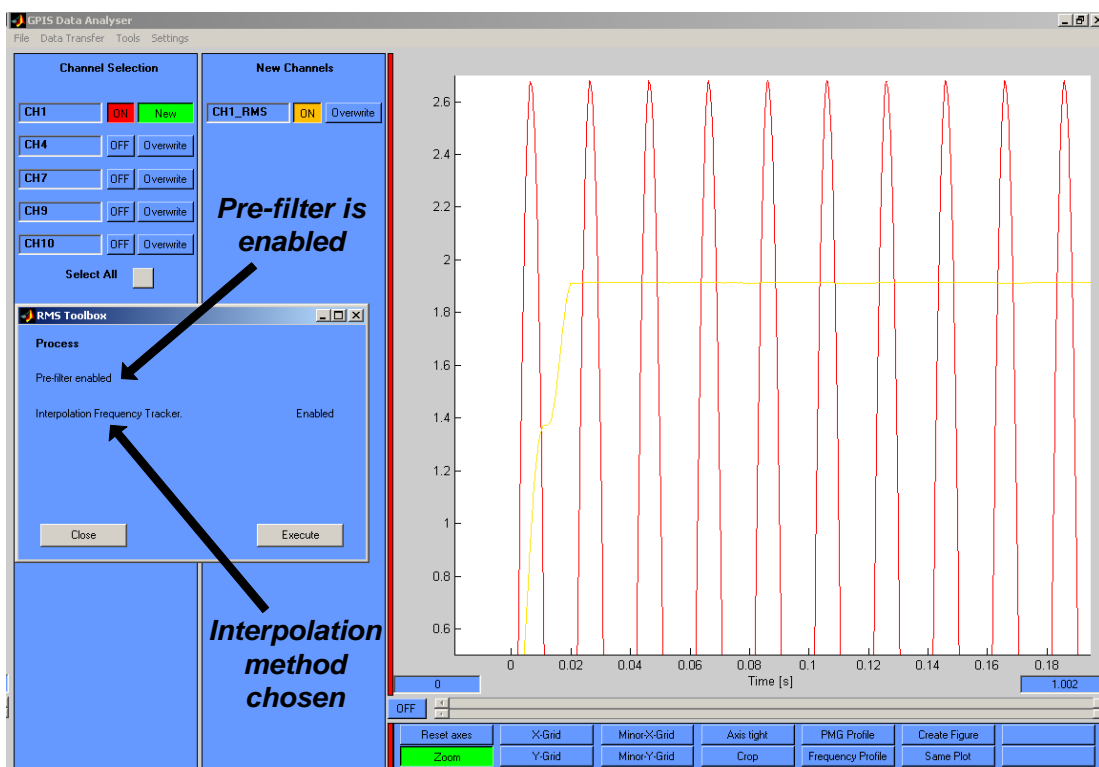


Figure 6.25 RMS toolbox layout.

The flowchart explaining the source code for the RMS toolbox is given on the following page in Figure 6.26. This particular module can calculate with a variable or fixed frequency input. Figure 6.26 discusses the code that is responsible for handling variable frequency calculations. The code for the fixed frequency computation is almost identical and thus is left out at this stage.

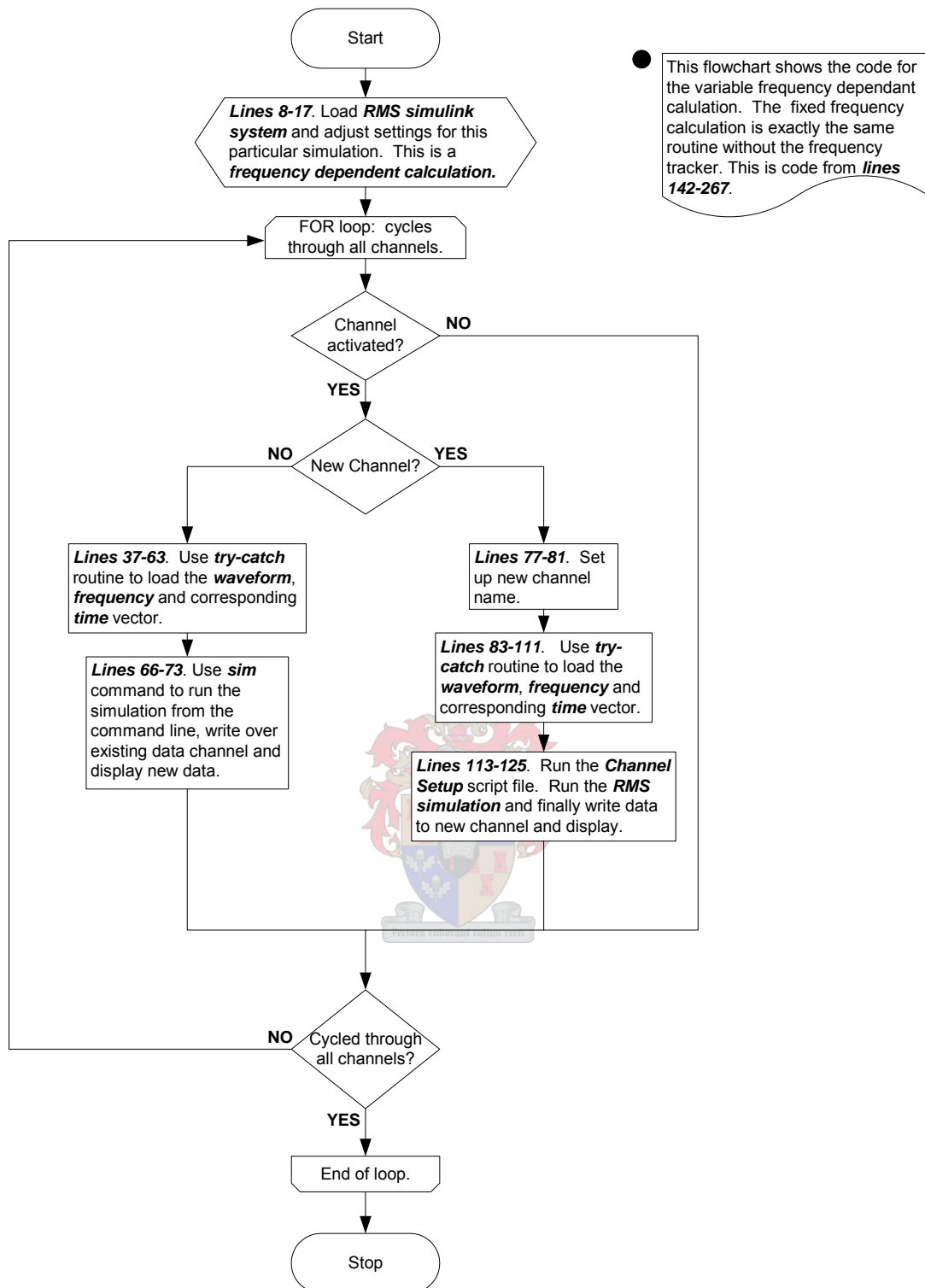


Figure 6.26 RootMeanSquare.m flowchart: RMS code implementation.

Figure 6.27 explains the breakdown of the RMS Simulink block used to carry out all RMS simulations with a fixed frequency.

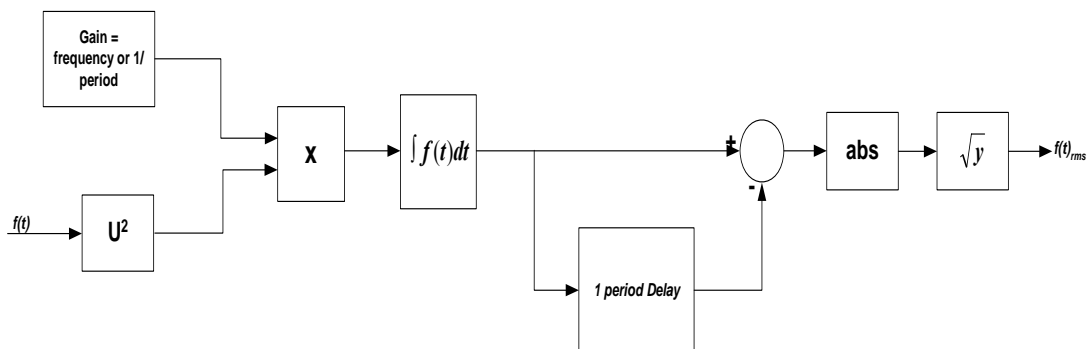


Figure 6.27 RMS block implementation in Simulink. Fixed frequency.

The setup in Figure 6.28 represents the RMS Simulink block used for all variable frequency simulations.

The structure for Figure 6.28 is quite similar to the diagram in Figure 6.27, the only difference being that the alternating frequency vector is streamed in simultaneously with the signal values.

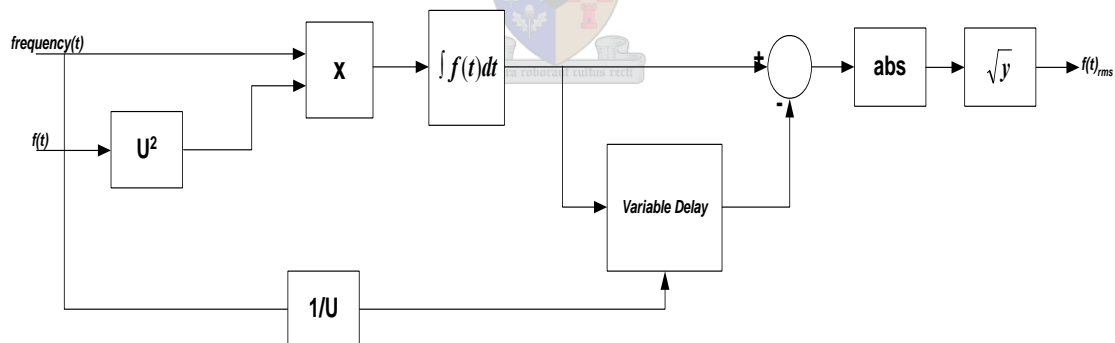
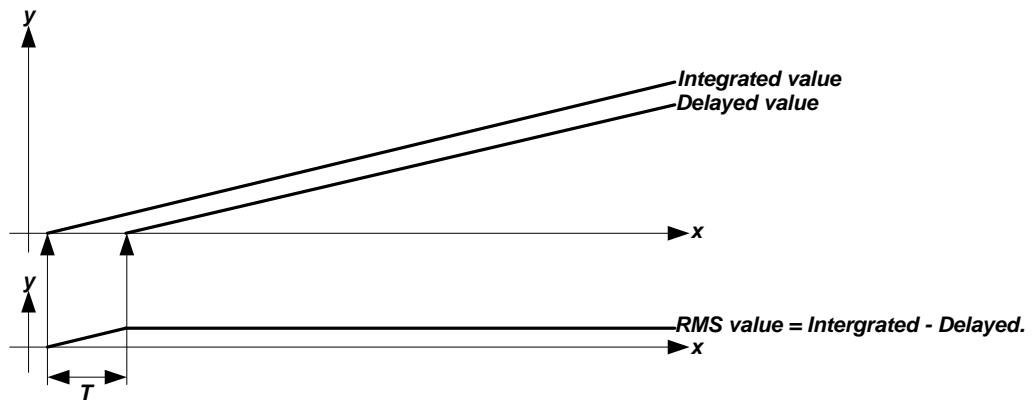


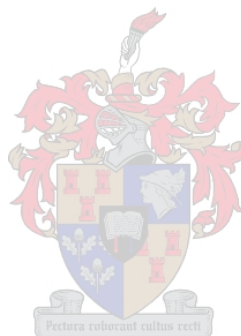
Figure 6.28 RMS block implementation in Simulink. Variable frequency.

Integrating the incoming signal continuously will cause the result to increase to infinity, thus the delay block in the above two figures is included to make sure that the starting value of the integrator is always zero; this process is explained in Figure 6.29.



**Figure 6.29** Delay block used in RMS computations.

What essentially happens is that the integrator has memory, this memory is always equal to the last integration value, thus what the delay block does is to negate this offset.



### 6.4.6. Power Toolbox

The final toolbox available calculates the power using current and voltage signals, the layout is shown in Figure 6.30. A drop-down box listing all channels allows the user to select which signals will be used for the calculation. A toggle button also allows one to switch between 3-phase and single phase power calculations.

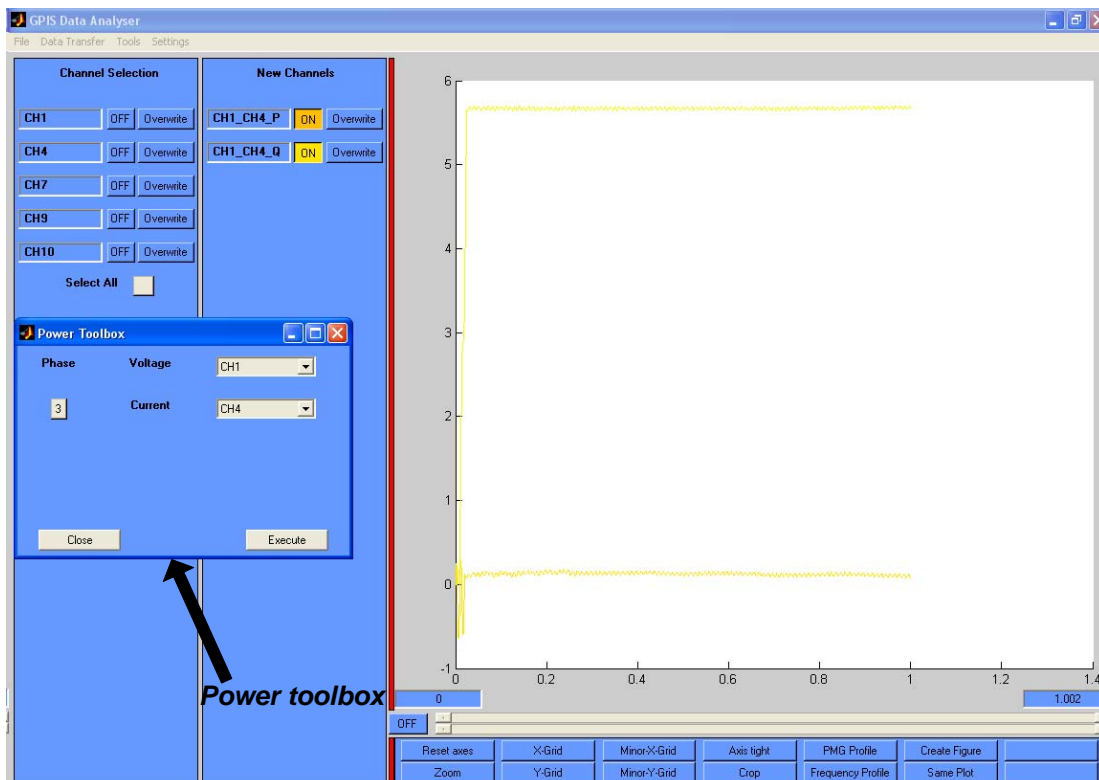


Figure 6.30 Power toolbox layout.

When power is calculated two new channels are set up in the 'New Channels' block, one for the reactive power and the other for the real power. The channels are created apart from one another for future use in the estimation process.

Figure 6.31 shows how the power calculation is implemented. The procedure is simplified by making use of Simulink's Power System Blockset, which has a built-in power calculation block.

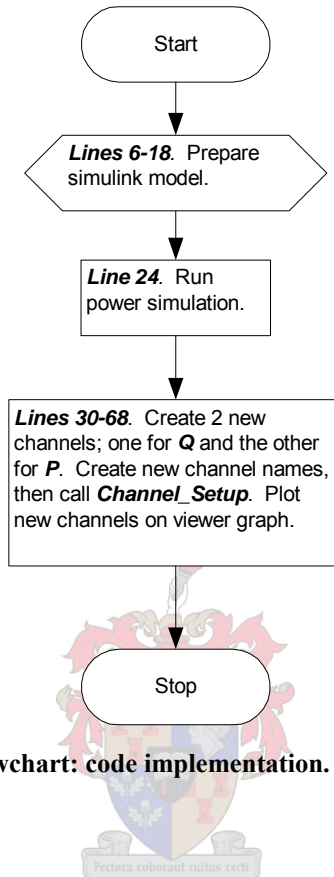


Figure 6.31 `Power_Calculation.m` flowchart: code implementation.

## 7. Conclusions and Recommendations

### 7.1. Conclusion

With the gap between demand and supply of energy ever diminishing, the operational integrity of present day power systems is of increasing concern. Software packages currently being used by industry to simulate the stability of networks can be cumbersome to work with especially when transforming field data into an acceptable form with which to estimate parameters. The primary goal of this project was to find a way to ease the transfer from the field to the simulation environment with respect to data conditioning. The secondary aim was to make use of Simulink Power Systems Blockset in an attempt to estimate the parameters of the synchronous machine.

As with any project involving system analysis, the initial stage involves the collection of field data. Once the data has been acquired it must now be processed, and this is where the first issue arises. An excellent example of this is when measurements had been taken of various units in Vietnam. The first problem was that a Yokogawa viewer was used as the recording device, thus a program had to be constructed to import the data into the MATLAB<sup>®</sup> environment. Once imported the data had to be processed and changed to another acceptable form so that it could be exported to DIGSILENT<sup>®</sup> for further analysis. A considerable amount of time was needed for the entire process; this was before any analysis could begin.

Using a single platform for the entire process seemed to be the next logical step. To attempt this, a suitable application that could import and process raw data was created. The software was programmed using MATLAB<sup>®</sup> script and proved to be useful in processing field data acquired from many units such as Palmiet and all those tested in Vietnam. Thus the first problem was solved with respect to data handling.

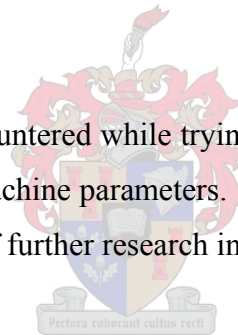
In keeping with previous experience data would now have to be exported to another application such as DIGSILENT<sup>®</sup> for further system analysis; however the idea of this project is to keep the entire process on a single platform. Thus the next step was to implement estimation of the rotor circuit using the Simulink Power Systems Blockset, as well as the

MATLAB<sup>®</sup> environment. Chapter 4 covers the progress of this section in much detail. Three models were proposed and tested. Although the validation of the estimated parameters could unfortunately not be acquired due to a lack of field base quantities, the results do look promising in that simulated data closely follows field measurements.

The focus then shifted to steady state parameters of the machine. Chapter 5 explains how the cost function was acquired. It presents a number of tests which helped shaped the final product and the eventual success thereof. A combination of various setpoints of the machine was found to be most successful in determining the steady state parameters of the machine. This was another example of the power of MATLAB<sup>®</sup> in that Simulink was effortlessly integrated with MATLAB<sup>®</sup> Script to simulate the models. Appendix D presents a model which is representative of the various that were implemented.

## 7.2. Recommendations

A number of pitfalls have been encountered while trying to use MATLAB<sup>®</sup> and its associated tools to estimate the synchronous machine parameters. A number of recommendations follow which could be undertaken as part of further research into this topic:

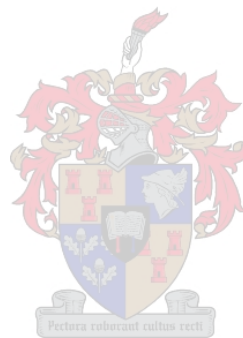


- While Simulink offers an environment where building and implementing models is relatively easy, the results weren't as realistic as could have been the case. This may just be teething problems and could be attributed to the fact that this type of project has never been fully implemented using MATLAB<sup>®</sup>. Thus to sort this out, the libraries in the Power Systems Blockset should be expanded to include a wider range of equipment that is currently employed in the field.
- The current source issue must also be sorted out so that the network can be more realistic.
- The rotor estimation results must be validated, thus the hope is to clear up the field base issue in order to resolve this problem. Should the results be close to actual values then this would account for the majority of the machine parameters, with the exception of the transient behaviour. If all goes according to plan then obtaining the transient parameters would be the logical topic for any future research.



- Further investigation into the estimation blockset of MATLAB® should also be an option in the continuation of this project. If the performance of the optimiser can be enhanced this can only improve outputs.

Acquiring the transient parameters of the synchronous machine will complete the entire process. This would then conclude the entire project and render it a success in terms of the single platform strategy. With MATLAB® ever evolving into one of the most powerful applications of its kind in the world, the room for improvement is endless and it would be fitting to conclude that “the sky is the limit” if these few problems can be ironed out.



## 8. References

- [1] P. Kundur, *Power System Stability and Control*, McGraw-Hill, 1994.
- [2] Leonard R. Anderson, *Electric Machines and Transformers*, Reston Publishing, Virginia, 1981.
- [3] *Design and Technical Data Manual of Palmiet Synchronous Generators*, Fuji Electric Co.,Ltd.
- [4] Syed A. Nasar, *Electric Energy Systems*, Prentice Hall, Upper Saddle River, New Jersey, 1996.
- [5] Paul C. Krause, Oleg Wasynczuk, Scott D. Sudhoff, *Analysis of Electric Machinery*, IEEE Press, 1995.
- [6] R.H. Park, *Two-Reaction Theory of Synchronous Machines – Generalized Method of Analysis – Part I*, AIEE Trans, Vol 48, July 1929, pp.716-727.
- [7] J.M. Strauss, *On-Line Estimation of Synchronous Generator Parameters using Pseudo-Random Binary Sequence Perturbations*, MSc.Eng Thesis, University of Stellenbosch, 1998.
- [8] E.C. Schaffer, C.A. Gross, *Methods for Determining Linear Synchronous Machine Parameters*, IEEE Press, 1994.
- [9] ANSI/IEEE Standard 115-1983, *IEEE Guide: Test Procedures for Synchronous Machines*.
- [10] P.J. Turner, *The D.C. Decay Test for Determining Synchronous Machine Parameters: Measurement and Simulation*, IEEE Trans. Energy Conversion, Dec. 1989, pp. 616-622.
- [11] ANSI/IEEE Standard 115A-1987, *Procedure for Obtaining Synchronous Machine Parameters by Standstill Frequency Response Testing*.

- [12] F.P. de Mello, and J.R. Ribeiro, *Derivation of Synchronous Machine Parameters from Tests*, *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-96, no. 4, July/August 1977, pp.1211-1218.
- [13] R. E. Bank, W. C. Coughran, Jr., W. Fichtner, E. Grosse, D. Rose, and R. Smith, *Transient Simulation of Silicon Devices and Circuits*, *IEEE Transactions on Power Apparatus and Systems*, CAD, 4 (1985), pp 436-451.
- [14] Shampine, L. F. and M. E. Hosea, *Analysis and Implementation of TR-BDF2*, *Applied Numerical Mathematics* 20, 1996.
- [15] *MATLAB® Reference Guide*, The Mathworks Inc., Natwick MA, 1995.
- [16] L.M. Canay, *Determination of the Model Parameters of Machines from the Reactance Operators  $x_d(p)$ ,  $x_q(p)$  (Evaluation of Standstill Frequency Response Test)*, *IEEE Trans. Energy Conversion*, Vol8, No. 2, June 1993, pp. 272-279.
- [17] R.L. Finney, G.B. Thomas, Jr, M.D. Weir, *Calculus 2<sup>nd</sup> Edition*, Addison-Wesley Publishing Company, Inc., 1994.
- [18] J.W. Nilsson, S. A. Riedel, *Electric Circuits 5<sup>th</sup> Edition*, Addison-Wesley Publishing Company, Inc., 1996.
- [19] RTDS Technologies, *Real-Time Digital Simulation Users Manual*, Rev 2, August 2003.
- [20] Robert I. Boylestad, *Introductory Circuit Analysis 8th Edition*, Prentice-Hall International, Inc., 1997.
- [21] P.L.Dandeno, "Current Usage and Suggested Practices in Power System Stability Simulations for Synchronous Machines", *IEEE Trans. Energy Conversion*, Vol. EC-1, March 1986, pp 77-91.

## Appendix A - Magnetically Coupled Circuits

In the case of electric machines, circuits in relative motion are magnetically coupled for the purpose of transferring energy between mechanical and electrical systems, otherwise known as electromechanical energy conversion.

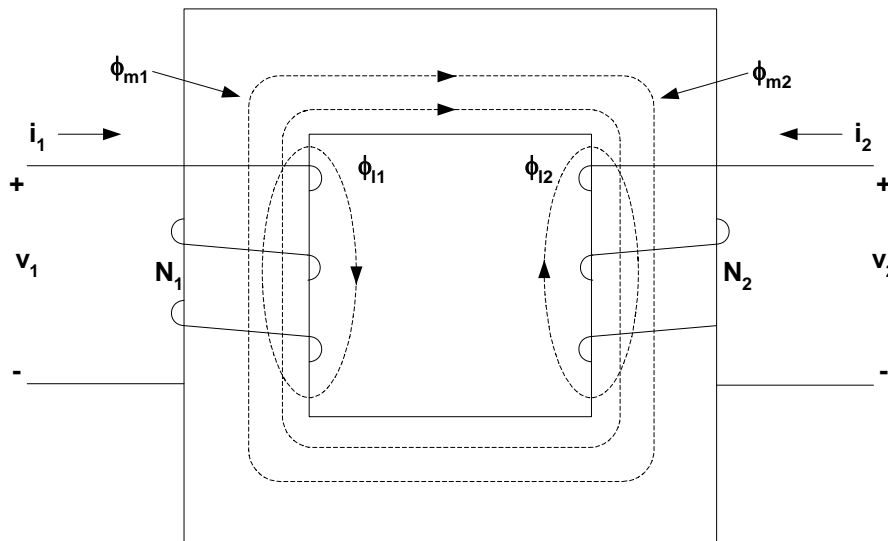


Figure A-1 Magnetically Coupled Circuits

Figure A-1 shows two stationary electric circuits magnetically coupled. The coils consist of turns  $N_1$  and  $N_2$  respectively and are wound on a common core. The currents flowing in the coils are denoted by  $i_1$  and  $i_2$  respectively. Four flux circuits are also shown and these are denoted by  $\phi_{m1}$ ,  $\phi_{m2}$ ,  $\phi_{l1}$ , and  $\phi_{l2}$  respectively. In general the flux produced by each coil can be broken up into a leakage component denoted by the subscript  $l$  and a magnetizing component denoted by the subscript  $m$ . Thus the flux linking each coil may be expressed as

$$\phi_1 = \phi_{l1} + \phi_{m1} + \phi_{m2} \quad (\text{A-1})$$

$$\phi_2 = \phi_{l2} + \phi_{m2} + \phi_{m1} \quad (\text{A-2})$$

The leakage flux  $\phi_{l1}$  is created by the current flowing in coil 1 and only links that coil, whilst the leakage flux produced by coil 2 is  $\phi_{l2}$  and only links the turns of coil 2. The magnetizing flux produced by coil 1 is denoted by  $\phi_{m1}$  and links both the turns of coil 1 and 2. Likewise

the magnetizing flux produced by coil 2 namely  $\phi_{m2}$  also links both coils. An important point to consider is that this is an idealized situation in that in practice only part of the leakage flux links the turns of its own coil, whilst a portion of the magnetizing flux produced by the first coil will not link the turns of the second coil. Thus the turns  $N_1$  and  $N_2$  are the equivalent number of turns and not the actual number.

The voltage equations may be expressed in a matrix form as

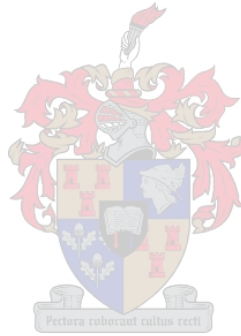
$$\mathbf{v} = \mathbf{ri} + \frac{d\boldsymbol{\lambda}}{dt} \quad (\text{A-3})$$

where

$$\mathbf{R} = \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \end{bmatrix} \quad (\text{A-4})$$

$$(\mathbf{v})^T = [v_1 \quad v_2] \quad (\text{A-5})$$

$$(\mathbf{i})^T = [i_1 \quad i_2] \quad (\text{A-6})$$



and

$$(\boldsymbol{\lambda})^T = [\lambda_1 \quad \lambda_2] \quad (\text{A-7})$$

The resistances  $r_1$  and  $r_2$  and the flux linkages  $\lambda_1$  and  $\lambda_2$  are all related to coils 1 and 2 respectively. As it is assumed that  $\phi_1$  links the equivalent turns of coil 1 and  $\phi_2$  links the equivalent turns of coil 2 the flux linkages can be given as

$$\lambda_1 = N_1 \phi_1 \quad (\text{A-8})$$

$$\lambda_2 = N_2 \phi_2 \quad (\text{A-9})$$

If saturation is neglected the fluxes may be expressed as

$$\phi_{l1} = \frac{N_1 i_1}{R_{l1}} \quad (\text{A-10})$$

$$\phi_{l2} = \frac{N_2 i_2}{R_{l2}} \quad (\text{A-11})$$

$$\phi_{m1} = \frac{N_1 i_1}{R_m} \quad (\text{A-12})$$

$$\phi_{m2} = \frac{N_2 i_2}{R_m} \quad (\text{A-13})$$

where  $R_{l1}$  and  $R_{l2}$  are the reluctances of the leakage paths and  $R_m$  is the reluctance of the magnetizing path of the magnetizing fluxes. The magnetomotive force (MMF) is determined by the product  $N$  times  $i$ . The reluctance of the leakage path is difficult to express and is impossible to measure, thus leakage flux can only be calculated or determined from design considerations [5]. However the reluctance of the magnetizing core can be approximated with the following relationship

$$R = \frac{l}{\mu A} \quad (\text{A-14})$$

Where  $l$  is the equivalent length of the magnetic path,  $A$  is the cross-sectional area and  $\mu$  is the permeability.

$$\phi_1 = \frac{N_1 i_1}{R_{l1}} + \frac{N_1 i_1}{R_m} + \frac{N_2 i_2}{R_m} \quad (\text{A-15})$$

$$\phi_2 = \frac{N_2 i_2}{R_{l2}} + \frac{N_1 i_1}{R_m} + \frac{N_2 i_2}{R_m} \quad (\text{A-16})$$

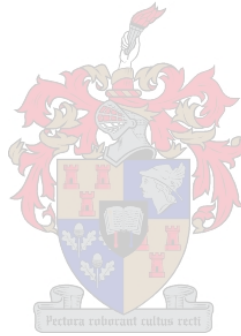
$$\lambda_1 = \frac{N_1^2}{R_{l1}} i_1 + \frac{N_1^2}{R_m} i_1 + \frac{N_1 N_2}{R_m} i_2 \quad (\text{A-17})$$

$$\lambda_2 = \frac{N_2^2}{R_{l2}} i_2 + \frac{N_2^2}{R_m} i_2 + \frac{N_2 N_1}{R_m} i_1 \quad (\text{A-18})$$

When the magnetic system is free from saturation the flux linkages can be expressed in terms of inductances and currents. The coefficients of the first two terms on the right-hand side of the previous two equations depend upon the turns of coils 1 and 2 respectively, thus the self inductances are defined as

$$\begin{aligned} L_{11} &= \frac{N_1^2}{R_{l1}} + \frac{N_1^2}{R_m} \\ &= L_{l1} + L_{m1} \end{aligned} \quad (\text{A-19})$$

$$\begin{aligned} L_{22} &= \frac{N_2^2}{R_{l2}} + \frac{N_2^2}{R_m} \\ &= L_{l2} + L_{m2} \end{aligned} \quad (\text{A-20})$$



Where  $L_{l1}$  and  $L_{l2}$  are the leakage inductances and  $L_{m1}$  and  $L_{m2}$  are the magnetizing inductances of coils 1 and 2 respectively. The mutual inductances are defined as

$$L_{12} = \frac{N_1 N_2}{R_m} \quad (\text{A-21})$$

$$L_{21} = \frac{N_2 N_1}{R_m} \quad (\text{A-22})$$

which essentially describes the mutual inductances as being equal.

With this in mind the mutual inductance is related to the magnetizing inductance with the relationship

$$L_{12} = L_{21} = \frac{N_2}{N_1} L_{m1} = \frac{N_1}{N_2} L_{m2} \quad (\text{A-23})$$

As was mentioned before, in a linear magnetic system the flux linkages may be expressed as

$$\lambda = Li \quad (\text{A-24})$$

where

$$\begin{aligned} L &= \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix} \\ &= \begin{bmatrix} L_{11} + L_{m1} & \frac{N_2}{N_1} L_{m1} \\ \frac{N_1}{N_2} L_{m2} & L_{12} + L_{m2} \end{bmatrix} \end{aligned} \quad (\text{A-25})$$

The next step is to perform a change of variables which will yield the equivalent T circuit of two magnetically coupled coils. The flux linkages can be expressed in the expanded form as

$$\lambda_1 = L_{11}i_1 + L_{m1} \left( i_1 + \frac{N_2}{N_1} i_2 \right) \quad (\text{A-26})$$

$$\lambda_2 = L_{12}i_2 + L_{m2} \left( i_2 + \frac{N_1}{N_2} i_1 \right) \quad (\text{A-27})$$

The derivation of the T circuit begins by referring the current of either of the coils to the other. If one refers the current of coil 2 to coil 1, one effectively is substituting a variable  $i'_2$  which is the equivalent current flowing through coil 1 that will produce the same MMF as that created by the actual  $i_2$  flowing through coil 2. Thus with this in mind the substitute variable is expressed as



$$i'_2 = \frac{N_2}{N_1} i_2 \quad (\text{A-28})$$

If the power is to remain constant with the change of variables then

$$v'_2 = \frac{N_1}{N_2} v_2 \quad (\text{A-29})$$

which ensures that  $v_2 i_2 = v'_2 i'_2$ .

Flux linkages, which have the units of volt-second, are expressed as

$$\lambda'_2 = \frac{N_1}{N_2} \lambda_2 \quad (\text{A-30})$$

$$\lambda_1 = L_{l1} i_1 + L_{m1} (i_1 + i'_2) \quad (\text{A-31})$$

$$\lambda'_2 = L'_{l2} i'_2 + L_{m1} (i_1 + i'_2) \quad (\text{A-32})$$



where

$$L'_{l2} = \left( \frac{N_1}{N_2} \right)^2 L_{l2} \quad (\text{A-33})$$

The voltage equations then become

$$v_1 = r_1 i_1 + \frac{d\lambda_1}{dt} \quad (\text{A-34})$$

$$v'_2 = r'_2 i'_2 + \frac{d\lambda'_2}{dt} \quad (\text{A-35})$$

where

$$r'_2 = \left( \frac{N_1}{N_2} \right)^2 r_2 \quad (\text{A-36})$$

The above voltage equations describe the T circuit shown below. It is apparent that this method can be extended to include any number of coils wound on the same core.

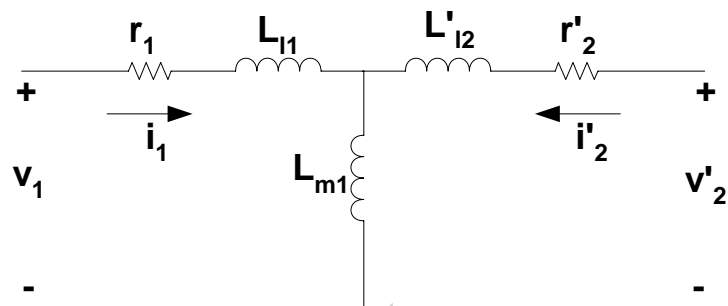


Figure A-2 Equivalent T circuit with coil 1 as reference coil.

## Equation Chapter 2 Section 2 Appendix B – Air Gap MMF

An overview of the work put forward by Krause et al. [5] on the topic of the air gap magnetomotive force (MMF) generated in the air gap is the subject of this section..

An elementary 2-pole, 3-phase, wye-connected salient pole synchronous machine will be used as an example for the forthcoming discussion. The schematic of this alternator is shown below in Figure B-1.

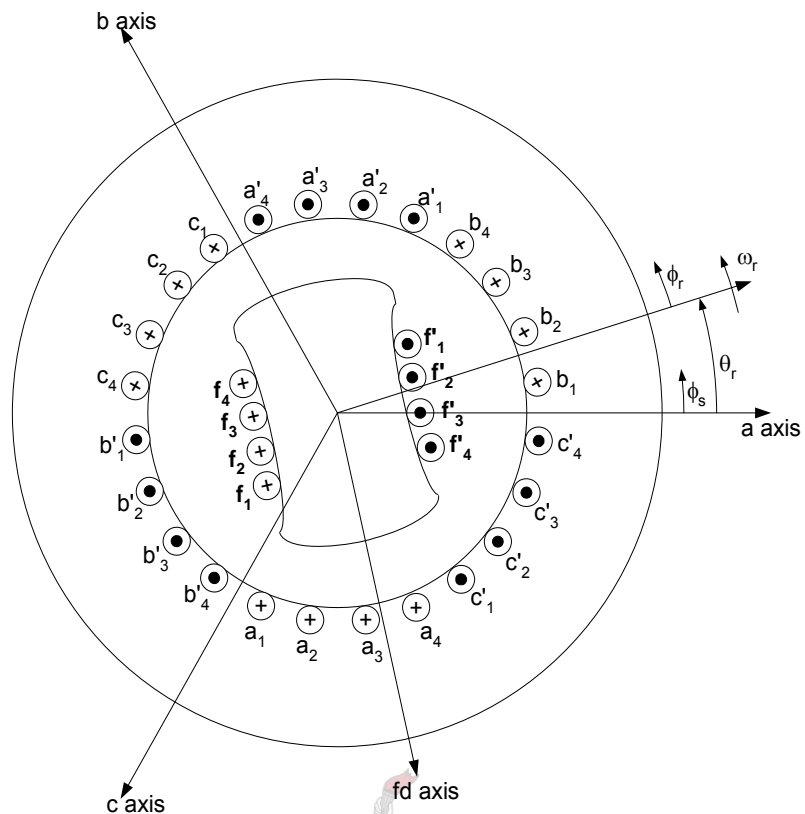


Figure B-1 Elementary 2-pole, 3-phase, wye-connected, salient pole synchronous machine.

The axes represent the positive direction of the flux produced by each of the windings. This machine has symmetrical stator windings as it is assumed that the a, b, and c windings are identical in terms of resistance and number of turns.

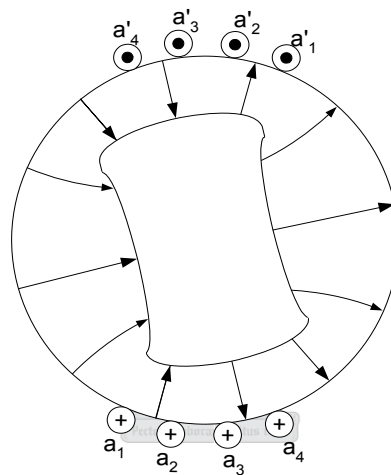
The coil sides are considered to be evenly distributed over  $60^\circ$  of the stator circumference, practically however, the coil sides are distributed over a much larger area, even as much as  $120^\circ$ . In this case it would be necessary for coil sides of two phases to overlap. Another practical consideration involves the uneven distribution of the coil sides along the periphery of the stator, for example coil sides  $a_2$  and  $a_3$  could consist of more turns than  $a_1$  and  $a_4$  thus creating an air-gap MMF that more closely approximates a sinusoidal distribution with respect to angular displacement along the air-gap [5].

Now that the schematic of the salient pole machine has been clarified it will be used to derive an expression for the air-gap MMF as in Krause *et al.* [5].

Following from the developed diagram in the work of Krause et al. [5], the magnetic field intensity and flux density are derived as functions of the angle  $\phi_s$ . Initially the angle,  $\phi_s$  is related to the rotor angular displacement  $\theta_r$ , and the rotor angular velocity  $\omega_r$  by the following relationship:

$$\phi_s = \phi_r + \theta_r \quad (\text{B-1})$$

Only one phase, the a-phase will be considered. Due to the relative distance from the rotor to stator and the high permeability of the steel of both, the magnetic field essentially exists in the air gap and is positioned in a radial direction.



**Figure B-2 Direction of magnetic field intensity due to current in phase a winding.**

Figure B-2, on the previous page, shows the direction of the flux through the 2-pole rotor, due to current direction in the windings of the a-phase positioned on the stator. Figure B-3 shows the various areas over which one must integrate to formulate equations for developed air-gap MMF.

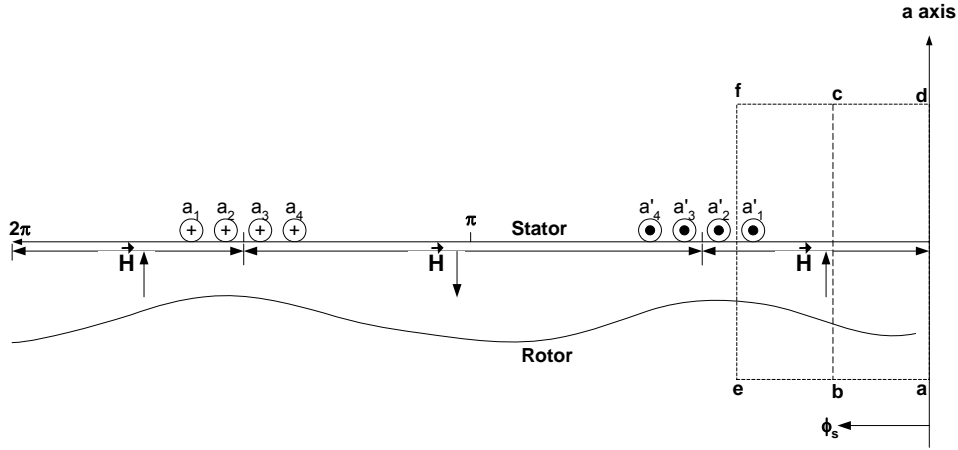


Figure B-3 Diagram displaying rotor 'rolled out'.

Both the magnetic field intensity,  $\vec{H}$  and the flux density,  $\vec{B}$  only have a component in the radial direction  $\vec{a}$  and are a function of  $\phi_s$  [5]. The magnetic field intensity  $\vec{H}$  has an assumed direction shown in Figure B-3 due to the current in phase a. Amperes law states that

$$\int \vec{H} \cdot d\vec{L} = i \tag{B-2}$$

where  $i$  is the net current enclosed within the closed path of integration (Krause *et al.* use the convention that current flowing out of the page is negative). Integrating around the path  $abcda$  (where  $\phi_s = \pi/4$ ) one can write Ampere's law as

$$\int_{r(\pi/4)}^{r(\pi/4)+g(\pi/4)} H\left(\frac{\pi}{4}\right) dL + \int_{r(0)+g(0)}^{r(0)} H(0) dL = 0$$

$$H_r\left(\frac{\pi}{4}\right)g\left(\frac{\pi}{4}\right) - H_r(0)g(0) = 0 \tag{B-3}$$

where  $r(\pi/4)$ ,  $r(0)$ ,  $g(\pi/4)$  and  $g(0)$  are the rotor radii and air-gap lengths respectively. . The MMF is defined as the line integral of  $\vec{H}$ , thus all terms on the left hand side can be rewritten in terms of MMFs. Thus the previous equation becomes

$$MMF\left(\frac{\pi}{4}\right) + MMF(0) = 0 \tag{B-4}$$

with  $MMF(0) = -H_r(0)g(0)$ . If one considers the second path of integration denoted figure A-3 as  $aefda$  a similar equation can be derived and is shown to be

$$MMF\left(\frac{7}{16}\pi\right) + MMF(0) = -n_c i_a \quad (B-5)$$

with  $\phi_s$  now shifted to  $7\pi/16$ , the enclosed path also contains a net current equal to the number of conductors carrying the a-phase current. The right-hand side of the previous equation is negative in accordance with the right-hand rule [5].

However the MMF at  $\phi_s=0$  cannot be zero as previously assumed due to the fact that Gauss's law specifically states that

$$\int_s \vec{B} \cdot d\vec{S} = 0 \quad (B-6)$$

This effectively requires that no net flux may cross the air gap, thus forming the equation (keeping in mind that the flux density in the machine context only has a radial component) one is confronted with the following:

$$\int_0^{2\pi} B_r(\phi_s) r l d\phi_s = 0 \quad (B-7)$$

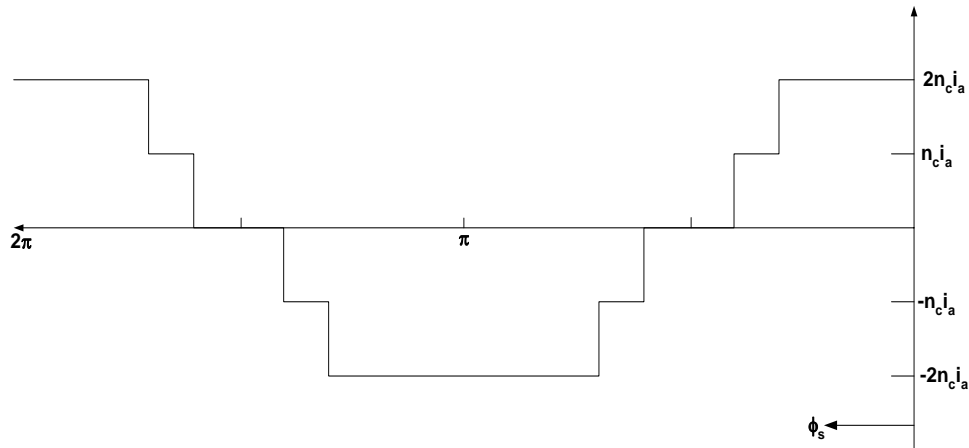
With the previous assumptions about zero MMF at the origin of the a-phase, this would create a unidirectional MMF waveform which would not satisfy Gauss's condition. Another factor that must be taken into account is that due to the shape of the salient pole machine a periodic air-gap is created such that

$$g(\phi_r) = g(\phi_r + \pi) \quad (B-8)$$

By satisfying Gauss's law and the previous constraint, the MMF waveform would have to have a zero average value with

$$MMF(\phi_s) = MMF(\phi_s + \pi) \quad (B-9)$$

and thus the shape of the waveform becomes that of a badly approximated sine wave as shown in Figure B-4.



**Figure B-4 Air gap MMF due to current in a-phase winding.**

The MMF due to the other two phase windings will be displaced  $120^\circ$  to either side of the MMF waveform created by the a-phase winding. In most practical applications the machine windings will be distributed over the periphery of the stator in order to produce a truly sinusoidal MMF waveform. The details of how the conductors are distributed can be found in the work of Krause *et al.* [5]. However the equations describing the MMF waveforms are given below:

$$MMF_a = \frac{N_s}{2} i_a \cos \phi_s \quad (\text{B-10})$$

$$MMF_b = \frac{N_s}{2} i_b \cos \left( \phi_s - \frac{2\pi}{3} \right) \quad (\text{B-11})$$

$$MMF_c = \frac{N_s}{2} i_c \cos \left( \phi_s + \frac{2\pi}{3} \right) \quad (\text{B-12})$$

Thus the total air-gap MMF is produced by adding all the individual MMFs given by the previous three equations

$$MMF_s = \frac{N_s}{2} \left[ i_a \cos \phi_s + i_b \cos \left( \phi_s - \frac{2\pi}{3} \right) + i_c \cos \left( \phi_s + \frac{2\pi}{3} \right) \right] \quad (\text{B-13})$$

Finally if one considers that at balanced steady state conditions the current may be expressed as

$$I_a = \sqrt{2} I_s \cos \left[ \omega_e t + \theta_{ei}(\theta) \right] \quad (\text{B-14})$$

$$I_b = \sqrt{2} I_s \cos \left[ \omega_e t - \frac{2\pi}{3} + \theta_{ei}(\theta) \right] \quad (\text{B-15})$$

$$I_c = \sqrt{2} I_s \cos \left[ \omega_e t + \frac{2\pi}{3} + \theta_{ei}(\theta) \right] \quad (\text{B-16})$$

where  $\theta_{ei}$  is the phase angle at time zero. Substituting (B-14), (B-15) and (B-16) into (B-13) which expresses the total MMF for the stator, one will obtain the following:

$$MMF_s = \left( \frac{N_s}{2} \right) \sqrt{2} I_s \left( \frac{3}{2} \right) \cos \left[ \omega_e t + \theta_{ei}(\theta) - \phi_s \right] \quad (\text{B-17})$$

This concludes a basic overview of the MMF waveform in the salient pole synchronous machine, for more information on how the space sinusoid is created for machines with P number of poles consult the work of Krause *et al.* [5].

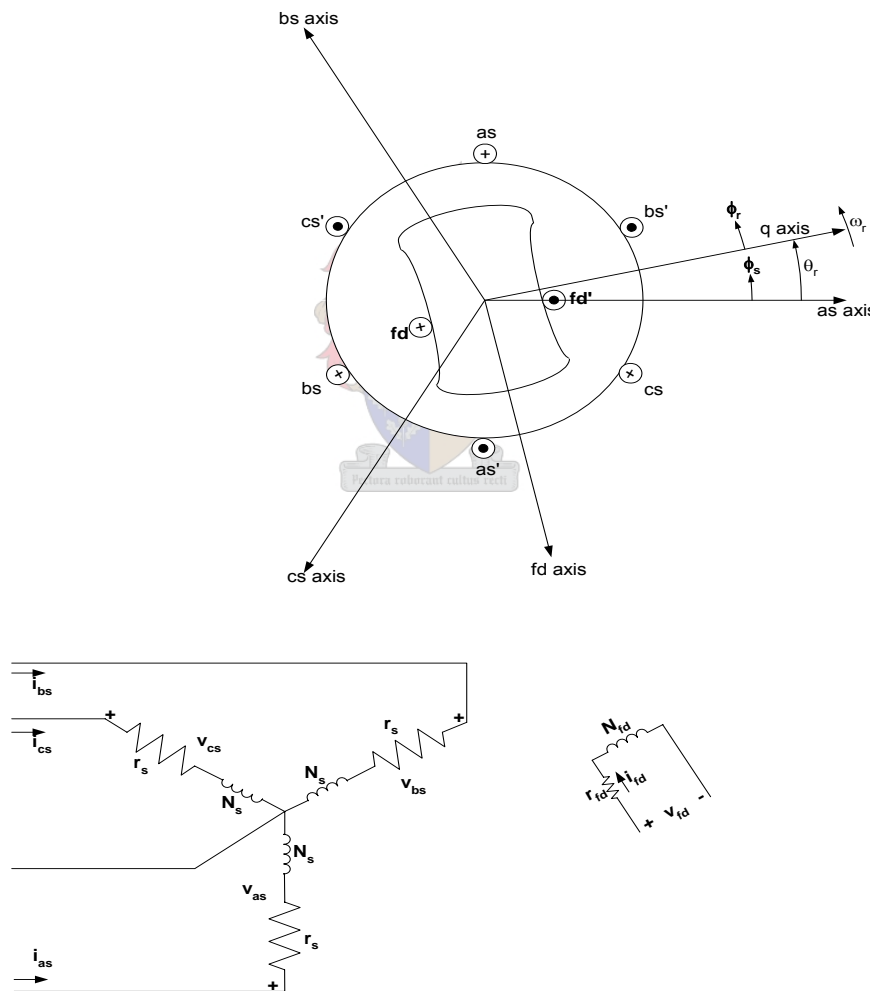
Equation Chapter 2 Section 2 Equation Chapter 3 Section 3



## Appendix C – Synchronous Generator Inductance Matrices

The theory of magnetically coupled circuits is dealt with in Appendix A, whilst Appendix B covers the topic of air-gap MMF. This section presents work covered by Krause *et al.* [5], in the development of equations describing the self and mutual inductances of the machine windings.

Figure C-1 introduces an elementary 2-pole, salient pole machine which will be used in further discussions. This development can easily be modified to account for additional windings (damper windings) placed on the rotor of the synchronous



machine.

**Figure C-1 Elementary 2-pole, 3-phase, wye-connected, salient-pole synchronous machine.**

Self inductance of a winding is the ratio of the flux linked by a winding to the current flowing through it, with all other winding currents zero. Mutual inductance is the ratio of flux linked by one winding due to current flowing in the neighbouring winding, with all other winding

currents zero including the target winding [5]. The air gap for a 2-pole machine can be approximated as

$$g(\phi_s) = \frac{l}{\alpha_1 - \alpha_2 \cos 2(\phi_s - \theta_r)} \quad (\text{C-1})$$

One can now express the flux density as a function of rotor angular displacement  $\theta_r$  and stator angle  $\phi_s$ ,

$$B_r(\phi_s, \theta_r) = \mu_0 \frac{MMF(\phi_s)}{g(\phi_s - \theta_r)} \quad (\text{C-2})$$

Thus with all currents zero except  $i_{as}$  the flux density can be expressed as

$$B_r(\phi_s, \theta_r) = \mu_0 \frac{N_s}{2} i_{as} \cos \phi_s [\alpha_1 - \alpha_2 \cos 2(\phi_s - \theta_r)] \quad (\text{C-3})$$

The equations describing the flux density as a result of current flowing only in the other two windings is the same as the previous derivation obviously with two changes, the first is the current source and secondly the 120° phase shift left or right depending upon which phase is being used. The  $N_s/2i_{as}\cos\phi_s$  is present due to the fact that the windings of each phase are assumed to be sinusoidally distributed.

As was the case with the stator windings the field windings are also assumed to be distributed to form a sinusoidal air-gap flux due to current flowing only in the field winding itself. Thus the air-gap MMF due to the  $i_{fd}$  current flowing in the field winding  $fd$  can be expressed as

$$MMF_{fd} = -\frac{N_f}{2} i_{fd} \sin \phi_r \quad (\text{C-4})$$

With  $N_f$  being the equivalent number of turns in the field winding. The flux density is given as

$$B_r(\phi_s, \theta_r) = -\mu_0 \frac{N_f}{2} i_{fd} \sin \phi_r [\alpha_1 - \alpha_2 \cos 2\phi_r] \tag{C-5}$$

The flux linkage is now determined by integrating over the open surface of a single turn of a stator winding which spans  $\pi$  radians. Thus the flux linking a single turn orientated at  $\phi_s$  radians from the axis of the as winding is given as

$$\Phi(\phi_s, \theta_r) = \int_{\phi_s}^{\phi_s + \pi} B_r(\xi, \theta_r) r l d\xi \tag{C-6}$$

The radius to the inside circumference of the stator is given by  $r$  and the axial length is represented by  $l$ . The dummy variable of integration is denoted by  $\xi$ . The next step is to acquire the flux linkages created by the entire phase winding to this end the flux created by each single turn must be summed. The assumption has been made that the windings are sinusoidally distributed and that there is no magnetic saturation, thus the total flux linkage is found by integrating over all coil sides carrying current in the same direction.

Before the equation for the total flux linkage is given a brief definition of how the sinusoidally distributed winding would be pertinent. The shaded portion in Figure C-2 shows the equivalent distribution of a sinusoidally distributed winding, and the accompanying MMF waveform created used this construction.

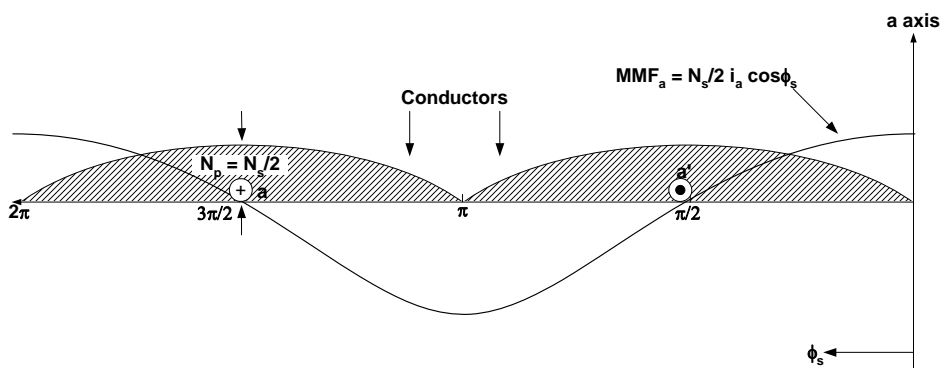


Figure C-2 Equivalent distribution of the as winding.

Once again note that the convention used by Krause et al. [5] is left to right on the graph simulating the stator angle moving anti-clockwise from the a axis.  $N_s$  represents the number of turns for the equivalent sinusoidally distributed winding, thus  $N_s$  can be expressed as

$$\begin{aligned}
N_S &= \int_0^\pi N_p \sin \phi_S d\phi_S \\
&= 2N_p
\end{aligned} \tag{C-7}$$

As stated before  $N_s$  is the equivalent number of turns for a coil to produce the same fundamental component as the actual winding distribution. Also note that the coil side only spans  $\pi$  radians of the stator. With this in mind the total flux linking a coil is expressed as

$$\begin{aligned}
\lambda_a &= L_l i_{as} + \int N_S(\phi_S) \Phi(\phi_S, \theta_r) d\phi_S \\
&= L_l i_{as} + \int N_S(\phi_S) \int_{\phi_S}^{\phi_S+\pi} B_r(\xi, \theta_r) r l d\xi d\phi_S
\end{aligned} \tag{C-8}$$

with the stator leakage inductance denoted by  $L_l$  (end turn leakage flux). Finally, to derive the expression for the self-inductance one must divide the previous equation by  $i_{as}$  which arrives at the following definition for the self-inductance of the  $as$  winding:

$$L_{aa} = L_l + \left(\frac{N_S}{2}\right)^2 \pi \mu_0 r l \left(\alpha_1 - \frac{\alpha_2}{2} \cos 2\theta_r\right) \tag{C-9}$$

The mutual inductance between the  $as$  and  $bs$  windings is determined by finding the flux linking the  $as$  winding due to current only flowing in the  $bs$  winding. The equation describing this inductance is given as

$$L_{ab} = -\left(\frac{N_S}{2}\right)^2 \frac{\pi}{2} \mu_0 r l \left[\alpha_1 + \alpha_2 \cos 2\left(\theta_r - \frac{\pi}{3}\right)\right] \tag{C-10}$$

In much the same way the mutual inductance between the stator windings and the field windings are derived and can be found in the work of Krause *et al.* [5]. For the convenience of the reader the mutual inductance between the  $as$  and  $fd$  windings can be expressed as

$$L_{asfd} = \left(\frac{N_s}{2}\right) \left(\frac{N_f}{2}\right) \pi \mu_0 r l \left(\alpha_1 + \frac{\alpha_2}{2}\right) \sin \theta_r \tag{C-11}$$

With the air gap density given in (C-5) the self inductance of the fd winding can be derived by first evaluating the flux linking the winding with all other currents zero except  $i_{fd}$ . Thus the self inductance is given as

$$L_{fd} = L_{lfd} + \left( \frac{N_f}{2} \right)^2 \pi \mu_0 r l \left( \alpha_1 + \frac{\alpha_2}{2} \right) \quad (C-12)$$

with  $L_{lfd}$  denoting the leakage inductance of the field winding. The remaining self and mutual inductances, as well as the machine inductances can be found in Krause *et al.* [5]. The voltage equations for the synchronous machine given in Figure C-1 are now given as

$$v_{as} = r_s i_{as} + \frac{d\lambda_{as}}{dt} \quad (C-13)$$

$$v_{bs} = r_s i_{bs} + \frac{d\lambda_{bs}}{dt} \quad (C-14)$$

$$v_{cs} = r_s i_{cs} + \frac{d\lambda_{cs}}{dt} \quad (C-15)$$

$$v_{fd} = r_{fd} i_{fd} + \frac{d\lambda_{fd}}{dt} \quad (C-16)$$

Where the field resistance is denoted by the subscript fd and the stator resistance is denoted by the subscript s. Due to the fact that some inductances in the voltage equations are a function of  $\theta_r$ , adds to the complexity level.

As opposed to the simple model of the synchronous machine that was used to develop the equations for the self and mutual inductances of the machine windings, a second machine setup is presented in Figure C-3, which has additional windings on the rotor. One other slight change is that the conventional current directions used for generator operation have been applied to the stator windings unlike in Figure C-1. The figure below is also given in the Literature Survey, however for convenience sake it has been added to the appendix.

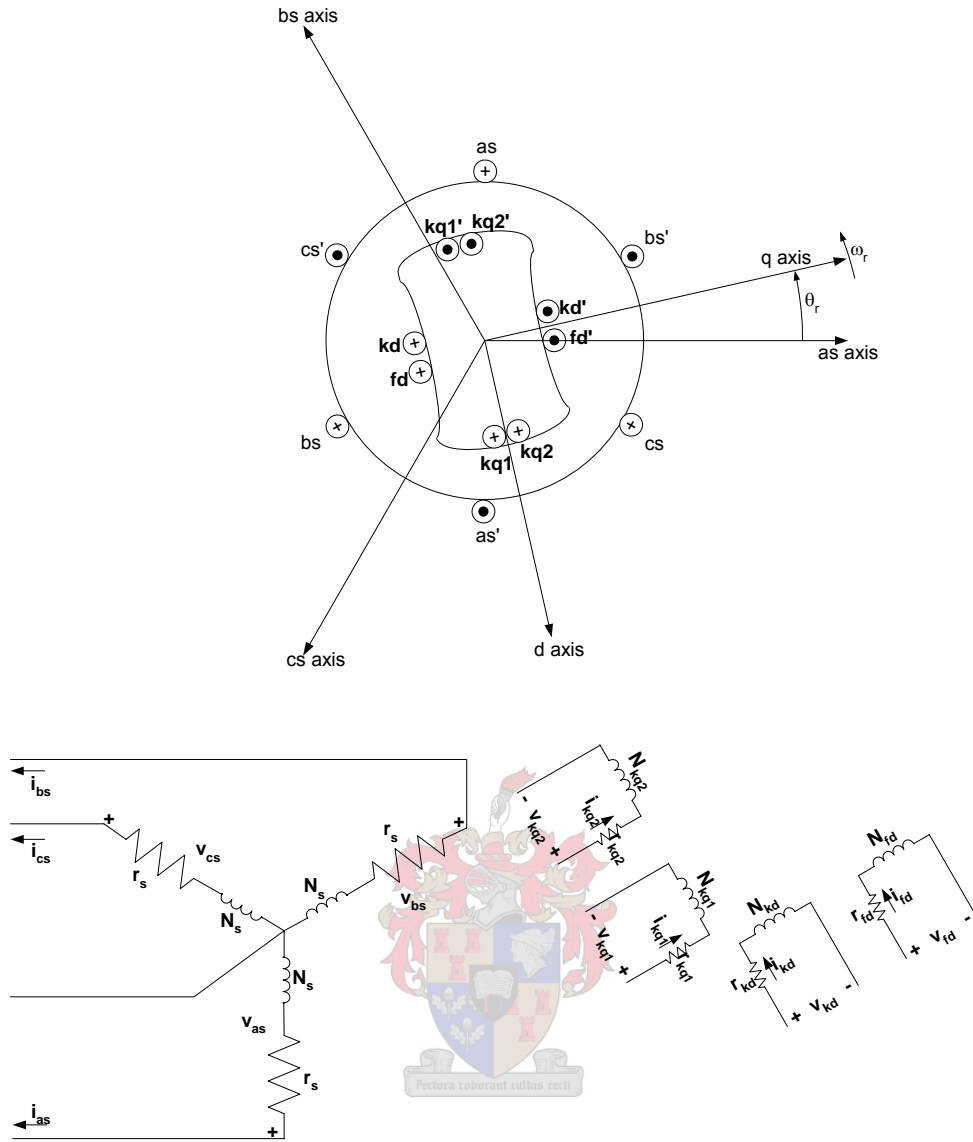


Figure C-3 Two-pole, 3-phase, wye-connected salient pole synchronous machine.

According to Krause *et al.* [5], the flux linkage equations for a 3 phase, wye-connected salient pole synchronous machine with a field winding (fd winding) a d-axis damper winding (kd winding) and two q-axis damper windings (kq1 and kq2 windings) may be given as

$$\begin{bmatrix} \lambda_{abc} \\ \lambda_{qdr} \end{bmatrix} = \begin{bmatrix} L_s & L_{sr} \\ (L_{sr})^T & L_r \end{bmatrix} \begin{bmatrix} -i_{abc} \\ i_{qdr} \end{bmatrix} \quad (C-17)$$

where

$$(\lambda_{abc})^T = [\lambda_{as} \quad \lambda_{bs} \quad \lambda_{cs}] \quad (C-18)$$

$$\left(\lambda_{qdr}\right)^T = \left[\lambda_{kq1} \quad \lambda_{kq2} \quad \lambda_{fd} \quad \lambda_{kd}\right] \quad (\text{C-19})$$

$$\left(i_{abc}\right)^T = \left[i_{as} \quad i_{bs} \quad i_{cs}\right] \quad (\text{C-20})$$

and

$$\left(i_{qdr}\right)^T = \left[i_{kq1} \quad i_{kq2} \quad i_{fd} \quad i_{kd}\right] \quad (\text{C-21})$$

The flux linkages  $\lambda_{as}$ ,  $\lambda_{bs}$  and  $\lambda_{cs}$  all denote the flux linking the stator windings as, bs and cs respectively. The flux linking the rotor windings kq1, kq2, fd and kd are denoted by  $\lambda_{kq1}$ ,  $\lambda_{kq2}$ ,  $\lambda_{fd}$  and  $\lambda_{kd}$ . The current flowing in the stator windings are given by  $i_{as}$ ,  $i_{bs}$  and  $i_{cs}$ , whilst  $i_{kq1}$ ,  $i_{kq2}$ ,  $i_{fd}$  and  $i_{kd}$  denote the currents flowing in the rotor windings.

The inductance matrices  $L_s$ ,  $L_{sr}$  and  $L_r$  give the different inductance relationships between the various currents flowing in the windings and the flux linkages linking the stator and rotor windings. The followings equations include the angle  $\theta_r$  which is defined as the angle between the magnetic axis of the as windings and the q-axis.

$$L_s = \begin{bmatrix} L_{ls} + L_A - L_B \cos 2\theta_r & -\frac{1}{2}L_A - L_B \cos 2\left(\theta_r - \frac{\pi}{3}\right) & -\frac{1}{2}L_A - L_B \cos 2\left(\theta_r + \frac{\pi}{3}\right) \\ -\frac{1}{2}L_A - L_B \cos 2\left(\theta_r - \frac{\pi}{3}\right) & L_{ls} + L_A - L_B \cos 2\left(\theta_r - \frac{2\pi}{3}\right) & -\frac{1}{2}L_A - L_B \cos 2\left(\theta_r + \pi\right) \\ -\frac{1}{2}L_A - L_B \cos 2\left(\theta_r + \frac{\pi}{3}\right) & -\frac{1}{2}L_A - L_B \cos 2\left(\theta_r + \pi\right) & L_{ls} + L_A - L_B \cos 2\left(\theta_r + \frac{2\pi}{3}\right) \end{bmatrix} \quad (\text{C-22})$$

The leakage inductance is denoted by  $L_{ls}$ . Magnetising inductances of the stator windings and their mutual inductances with respect to each other are expressed in terms of the inductances  $L_A$  and  $L_B$ . These two inductances originate from the nature of the salient pole machine which has a varying path of reluctance. When the path of reluctance is at its lowest then the magnetising inductance of a stator winding is equal to  $L_A + L_B$ , when the path of reluctance is

at its greatest the value of the magnetising inductance becomes  $L_A - L_B$ . For the above equation  $L_A > L_B$ , and for a round rotor machine  $L_B$  is zero.

The inductance matrix  $L_{sr}$  gives all the mutual inductances of the rotor and stator windings, and is expressed as

$$L_{sr} = \begin{bmatrix} L_{skq1} \cos \theta_r & L_{skq2} \cos \theta_r & L_{sfd} \sin \theta_r & L_{skd} \sin \theta_r \\ L_{skq1} \cos \left( \theta_r - \frac{2\pi}{3} \right) & L_{skq2} \cos \left( \theta_r - \frac{2\pi}{3} \right) & L_{sfd} \sin \left( \theta_r - \frac{2\pi}{3} \right) & L_{skd} \sin \left( \theta_r - \frac{2\pi}{3} \right) \\ L_{skq1} \cos \left( \theta_r + \frac{2\pi}{3} \right) & L_{skq2} \cos \left( \theta_r + \frac{2\pi}{3} \right) & L_{sfd} \sin \left( \theta_r + \frac{2\pi}{3} \right) & L_{skd} \sin \left( \theta_r + \frac{2\pi}{3} \right) \end{bmatrix} \quad (C-23)$$

where the subscripts skq1, skq2, sfd and skd denote the mutual inductances between the two windings (stator and rotor) [5]. Finally, the inductance matrix  $L_r$  is expressed in terms of the leakage and magnetising inductances of the field and damper windings. This is given as

$$L_r = \begin{bmatrix} L_{lkq1} + L_{mkq1} & L_{kq1kq2} & 0 & 0 \\ L_{kq1kq2} & L_{lkq2} + L_{mkq2} & 0 & 0 \\ 0 & 0 & L_{lfd} + L_{mfd} & L_{fdkd} \\ 0 & 0 & L_{fdkd} & L_{lkd} + L_{mkd} \end{bmatrix} \quad (C-24)$$

The magnetising inductances are defined as

$$L_{mq} = \frac{3}{2} (L_A - L_B) \quad (C-25)$$

$$L_{md} = \frac{3}{2} (L_A + L_B) \quad (C-26)$$

with the subscripts md and mq referring to the d-axis and q-axis components respectively. Whereupon it is easy to show that



$$L_{skq1} = \left( \frac{N_{kq1}}{N_s} \right) \left( \frac{2}{3} \right) L_{mq} \quad (\text{C-27})$$

$$L_{skq2} = \left( \frac{N_{kq2}}{N_s} \right) \left( \frac{2}{3} \right) L_{mq} \quad (\text{C-28})$$

$$L_{sfd} = \left( \frac{N_{fd}}{N_s} \right) \left( \frac{2}{3} \right) L_{md} \quad (\text{C-29})$$

$$L_{sfd} = \left( \frac{N_{fd}}{N_s} \right) \left( \frac{2}{3} \right) L_{md} \quad (\text{C-30})$$

$$L_{skd} = \left( \frac{N_{kd}}{N_s} \right) \left( \frac{2}{3} \right) L_{md} \quad (\text{C-31})$$

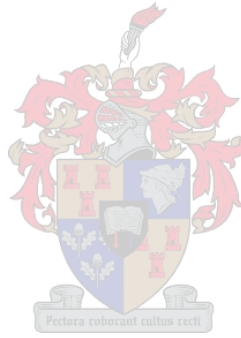
$$L_{mkq1} = \left( \frac{N_{kq1}}{N_s} \right)^2 \left( \frac{2}{3} \right) L_{mq} \quad (\text{C-32})$$

$$L_{mkq2} = \left( \frac{N_{kq2}}{N_s} \right)^2 \left( \frac{2}{3} \right) L_{mq} \quad (\text{C-33})$$

$$L_{mfd} = \left( \frac{N_{fd}}{N_s} \right)^2 \left( \frac{2}{3} \right) L_{md} \quad (\text{C-34})$$

$$L_{mkd} = \left( \frac{N_{kd}}{N_s} \right)^2 \left( \frac{2}{3} \right) L_{md} \quad (\text{C-35})$$

$$\begin{aligned} L_{kq1kq2} &= \left( \frac{N_{kq2}}{N_{kq1}} \right) L_{mkq1} \\ &= \left( \frac{N_{kq1}}{N_{kq2}} \right) L_{mkq2} \end{aligned} \quad (\text{C-36})$$



$$\begin{aligned} L_{fdkd} &= \left( \frac{N_{kd}}{N_{fd}} \right) L_{mfd} \\ &= \left( \frac{N_{fd}}{N_{kd}} \right) L_{mkd} \end{aligned} \quad (C-37)$$

$$i'_j = \left( \frac{2}{3} \right) \left( \frac{N_j}{N_s} \right) i_j \quad (C-38)$$

$$v'_j = \left( \frac{N_s}{N_j} \right) v_j \quad (C-39)$$

$$\lambda'_j = \left( \frac{N_s}{N_j} \right) \lambda_j \quad (C-40)$$

where j may be kq1, kq2, fd or kd [5].

The flux linkages may now be written as



$$\begin{bmatrix} \lambda_{abcs} \\ \lambda'_{qdr} \end{bmatrix} = \begin{bmatrix} L_s & L'_{sr} \\ \frac{2}{3}(L'_{sr})^T & L_r \end{bmatrix} \begin{bmatrix} -i_{abcs} \\ i'_{qdr} \end{bmatrix} \quad (C-41)$$

where  $L_s$  is as defined before, thus

$$L'_{sr} = \begin{bmatrix} L_{mq} \cos \theta_r & L_{mq} \cos \theta_r & L_{mq} \sin \theta_r & L_{md} \sin \theta_r \\ L_{mq} \cos \left( \theta_r - \frac{2\pi}{3} \right) & L_{mq} \cos \left( \theta_r - \frac{2\pi}{3} \right) & L_{md} \sin \left( \theta_r - \frac{2\pi}{3} \right) & L_{md} \sin \left( \theta_r - \frac{2\pi}{3} \right) \\ L_{mq} \cos \left( \theta_r + \frac{2\pi}{3} \right) & L_{mq} \cos \left( \theta_r + \frac{2\pi}{3} \right) & L_{md} \sin \left( \theta_r + \frac{2\pi}{3} \right) & L_{md} \sin \left( \theta_r + \frac{2\pi}{3} \right) \end{bmatrix}$$

$$(C-42)$$

$$\mathbf{L}'_r = \begin{bmatrix} L'_{lkq1} + L_{mq} & L_{mq} & 0 & 0 \\ L_{mq} & L'_{lkq2} + L_{mq} & 0 & 0 \\ 0 & 0 & L'_{lfd} + L_{md} & L_{md} \\ 0 & 0 & L_{md} & L'_{lkd} + L_{md} \end{bmatrix} \quad (\text{C-43})$$

The voltage equations expressed in terms of machine variables referred to the stator windings are

$$\begin{bmatrix} v_{abc} \\ v'_{qdr} \end{bmatrix} = \begin{bmatrix} r_s + \frac{dL_s}{dt} & \frac{dL'_{sr}}{dt} \\ \frac{2}{3} \frac{d(L'_{sr})^T}{dt} & r'_r + \frac{dL'_r}{dt} \end{bmatrix} \begin{bmatrix} -i_{abc} \\ i'_{qdr} \end{bmatrix} \quad (\text{C-44})$$

In (C-43) and (C-44)

$$r'_j = \left(\frac{3}{2}\right) \left(\frac{N_s}{N_j}\right)^2 r_j \quad (\text{C-45})$$

$$L'_{lj} = \left(\frac{3}{2}\right) \left(\frac{N_s}{N_j}\right)^2 L_{lj} \quad (\text{C-46})$$

where j may be kq1, kq2, fd, or kd. The voltage equations given in (C-44) are valid for the positive direction of stator current assumed out of the stator terminals [5].

## Appendix D – MATLAB® Script Files and Simulink Models

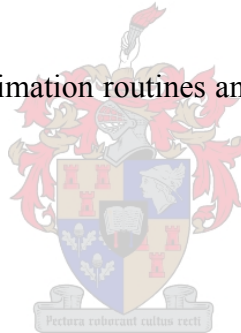
### RL Circuit Method

### RL Equation Method

The M-file script for both the RL Equation and RL Circuit methods are basically the same. They consist of a main file that initiates the estimator. The estimation routine then calls the second script file, a function, which runs the model in Simulink. Once the simulation is complete the function hands back various results to the estimator for further scrutiny.

Various models follow the script files. The first uses the field voltage to simulate the field current and compare this with measured values. The second model uses field current to implement the differential equation and compare simulated voltage with measured field data.

All further code pertaining to the estimation routines and the Data Analyser is contained on a CD that comes with this thesis.



## RL Circuit and Equation Method

### Field Main

*This program is the main program used to estimate the rotor field parameters  $L_{fd}$  and  $R_{fd}$ . It initially declares all variables needed in the process, most importantly Rfield and Lfield. The upper and lower bounds are then declared which informs the estimator of its boundary settings. The rest of the settings for the optimizer are then declared in the options array. Finally the estimator is called which in turn calls the FieldEst function. This function returns the error value created by the cost function in Simulink.*

**clc;**

**global Rfield Lfield a b c F FValue Parvec;**

**Solution\_Log = [];**

**Data\_Record = [];**

**Prmtrinitval = [0.9 0.11];**

*Lower and upper bounds are set for the estimation procedure*

**LowerB = [0.0001 0.0000000001];**

**UpperB = [20 20];**

*The MinOptions settings determine how the estimator attempts the optimization procedure.*

*The DiffMax and Min change is with respect to the maximum and minimum step change which one allows the estimator to take. Following each iteration various values are displayed including the error and parameter values.*

**MinOptions = optimset('DerivativeCheck','off',...**

**'Diagnostics','off',...**

**'DiffMaxChange',1e-1,...**

**'DiffMinChange',1e-5,...**

**'Display','iter',...**

**'GoalsExactAchieve',0,...**

**'GradConstr','off',...**

**'GradObj','off',...**

**'Hessian','off',...**

**'HessPattern',[],...**

**'HessUpdate','bfgs',...**

**'Jacobian',[],...**

**'JacobPattern',[],...**

**'LargeScale','off',...**

**'LevenbergMarquardt','off',...**

**'LineSearchType','cubicpoly',...**

```

'MaxFunEvals',500,...
'MaxIter',50,...
'MaxPCGIter',50,...
'MeritFunction','multiobj',...
'MinAbsMax',0,...
'PrecondBandWidth',0,...
'TolCon',0.0000000001,...
'TolFun',0.0000000001,...
'TolX',0.0000000001,...
'TypicalX',[]);

```

```

[Parvec,FValue,ExitReason,OutputData] =
fmincon('FieldEst',Prmtrinitval,[],[],[],[],LowerB,UpperB,[],MinOptions,0,9)

```

```

if ExitReason > 0
    disp('Exit reason: FMINUNC converged to a solution X');
elseif ExitReason == 0
    disp('Exit reason: The maximum number of function evaluations was reached');
else
    disp('Exit reason: FMINUNC did not converge to a solution');
end;

```



### Field Est Function

*This function calls the passes variables on to the Simulink model, runs the models and then hands back to Field Main.*

```

function F = FieldEst(X,StartTime,EndTime);

global Rfield Lfield F FValue Parvec;

tic;

Rfield = X(1,1);
Lfield = X(1,2);

[TimeSteps,States,Error] = sim('FieldParameterEstimation4',[],[],[]);

close all;
F = sum(Error);
display(Rfield);
display(Lfield);
display(F);

time_taken = toc

```

## de Mello Method

**RotorEstimationMain**

**clc;**

**Global variables.**

```
global Lfd Lkd1 Lad Rfd Rkd1 F sim_field_current new_field_voltage
new_field_voltage_filtered new_field_current new_field_current_filtered FValue
Parvec;
```

**Define initial parameters for the system.**

```
Prmtrinitval = [1 1 1 1 1];
```

**%Lower and upper bounds are set for the estimation procedure**

```
LowerB = [0.0001 0.0001 0.0001 0.0001 0.0001];
```

```
UpperB = [100 100 100 100 100];
```

```
MinOptions = optimset('DerivativeCheck','off',...
```

```
'Diagnostics','off',...
```

```
'DiffMaxChange',1e-1,...
```

```
'DiffMinChange',1e-5,...
```

```
'Display','iter',...
```

```
'GoalsExactAchieve',0,...
```

```
'GradConstr','off',...
```

```
'GradObj','off',...
```

```
'Hessian','off',...
```

```
'HessPattern',[],...
```

```
'HessUpdate','bfgs',...
```

```
'Jacobian',[],...
```

```
'JacobPattern',[],...
```

```
'LargeScale','off',...
```

```
'LevenbergMarquardt','off',...
```

```
'LineSearchType','cubicpoly',...
```

```
'MaxFunEvals',20000,...
```

```
'MaxIter',20000,...
```

```
'MaxPCGIter',50,...
```

```
'MeritFunction','multiobj',...
```

```
'MinAbsMax',0,...
```

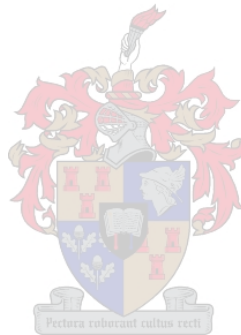
```
'PrecondBandWidth',0,...
```

```
'TolCon',0.000000000001,...
```

```
'TolFun',0.000000000001,...
```

```
'ToIX',0.000000000001,...
```

```
'TypicalX',[]);
```



```
A = [1 0 0 -1 0];
```

```
b = [0];
```

**Call estimation routine.**

```
[Parvec,FValue,ExitReason,OutputData] =  
fmincon('FieldEstRotorFunction',Prmtrinitval,A,b,[],[],LowerB,UpperB,[],MinOptions)
```

```
if ExitReason > 0  
    disp('Exit reason: FMINUNC converged to a solution X');  
elseif ExitReason == 0  
    disp('Exit reason: The maximum number of function evaluations was reached');  
else  
    disp('Exit reason: FMINUNC did not converge to a solution');  
end;
```

**Plot original and estimated waveform.**

```
figure;  
plot(new_time,new_field_current);  
hold;  
plot(new_time,sim_field_current,'r');
```

**function F = FieldEstRotorFunction(X);**

*Implements a digital filter of the voltage waveform using the transfer function of the rotor circuit. The end result, the filed current is then compared to the original. The error is sent back to the estimator.*

*Global variables.*

```
global Lfd Lkd1 Lad Rfd Rkd1 F sim_field_current new_field_voltage  
new_field_voltage_filtered new_field_current new_field_current_filtered Parvec  
FValue;
```

*Assign values for the circuit.*

```
Lfd = X(1,1);  
Lkd1 = X(1,2);  
Rkd1 = X(1,3);  
Lad = X(1,4);  
Rfd = X(1,5);
```

*Define coefficients of the equation describing the filter.*

```
a0 = Rfd*Rkd1/(Lad+Lkd1);  
a1 = Rfd + Lfd*Rkd1/(Lad+Lkd1) + Lad*Rkd1/(Lad+Lkd1);  
a2 = Lfd + Lad*Lkd1/(Lad+Lkd1);
```

```
b0 = Rkd1/(Lad+Lkd1);  
b1 = 1;
```

*Setup transfer function*

```
num = [a2 a1 a0];  
den = [b1 b0];
```



*The sampling frequency of the PRBS waveform.*

**fs = 1.25e3;**

*Transforms the filter from s-domain to digital. The numerator and denominator are reversed to make the filter a proper function, thus the filter represents the admittance of the ROTOR Circuit, Thus  $I=1/Z.V \Rightarrow I=A.V$ .*

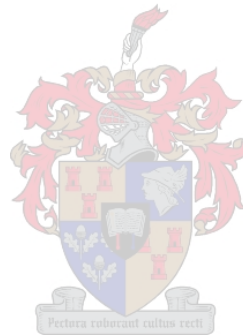
**[numd,dend] = bilinear(den,num,fs);**

*Apply the digital filter.*

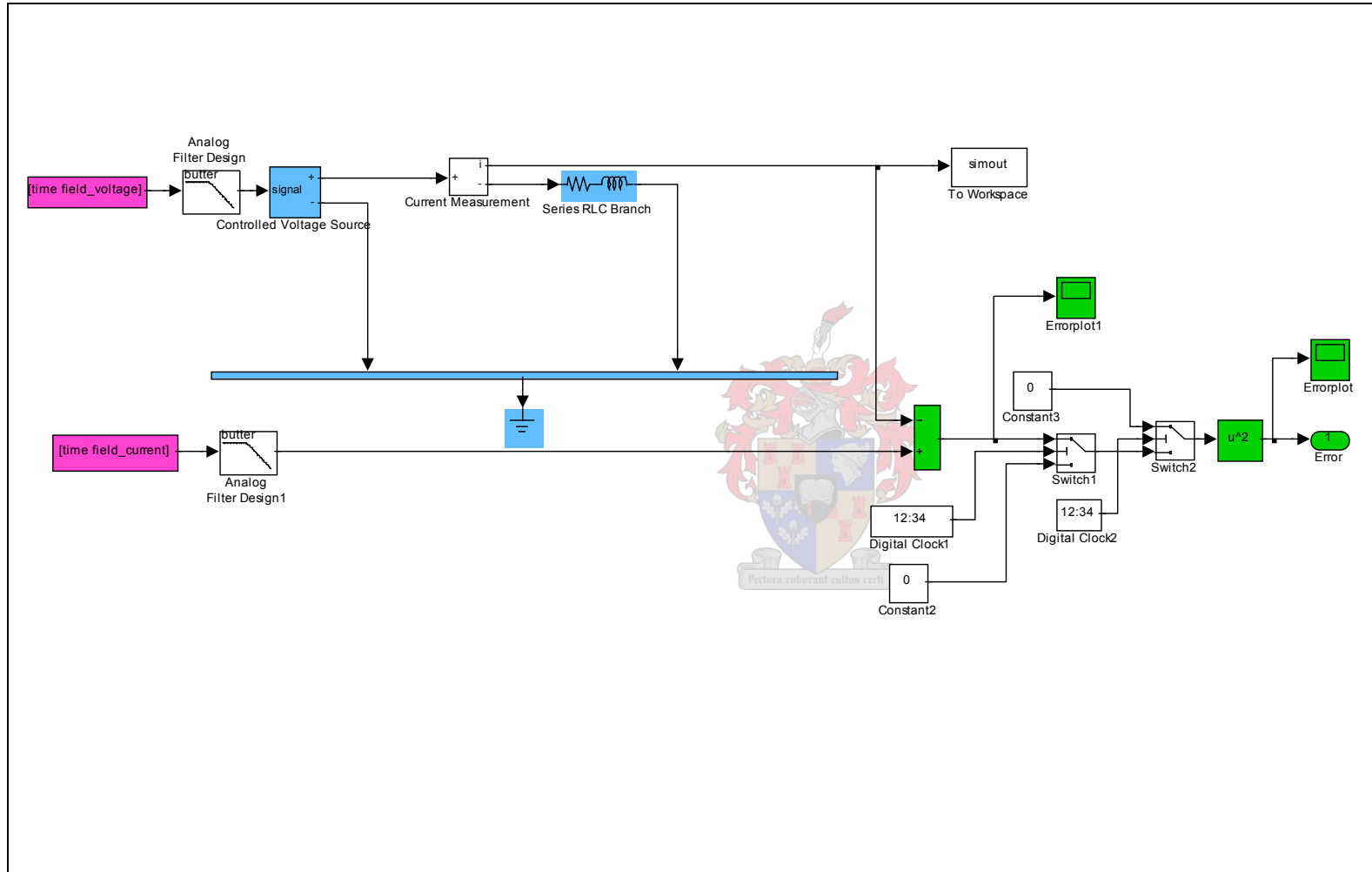
**sim\_field\_current = filter(numd,dend,new\_field\_voltage);**

*Cost Function to be returned to the estimator.*

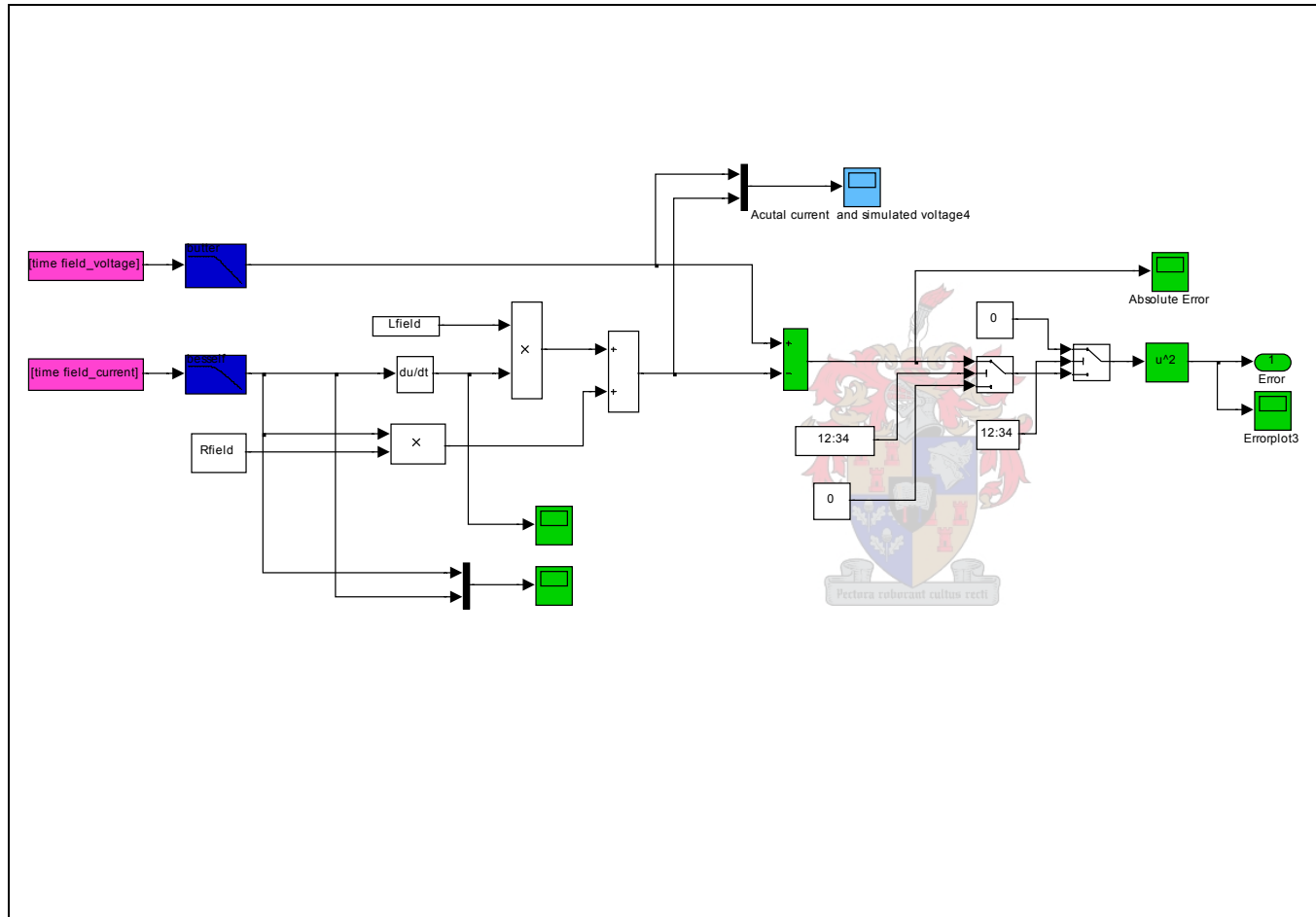
**F = sum((new\_field\_current(58500:83500) - sim\_field\_current(58500:83500)).^2);**



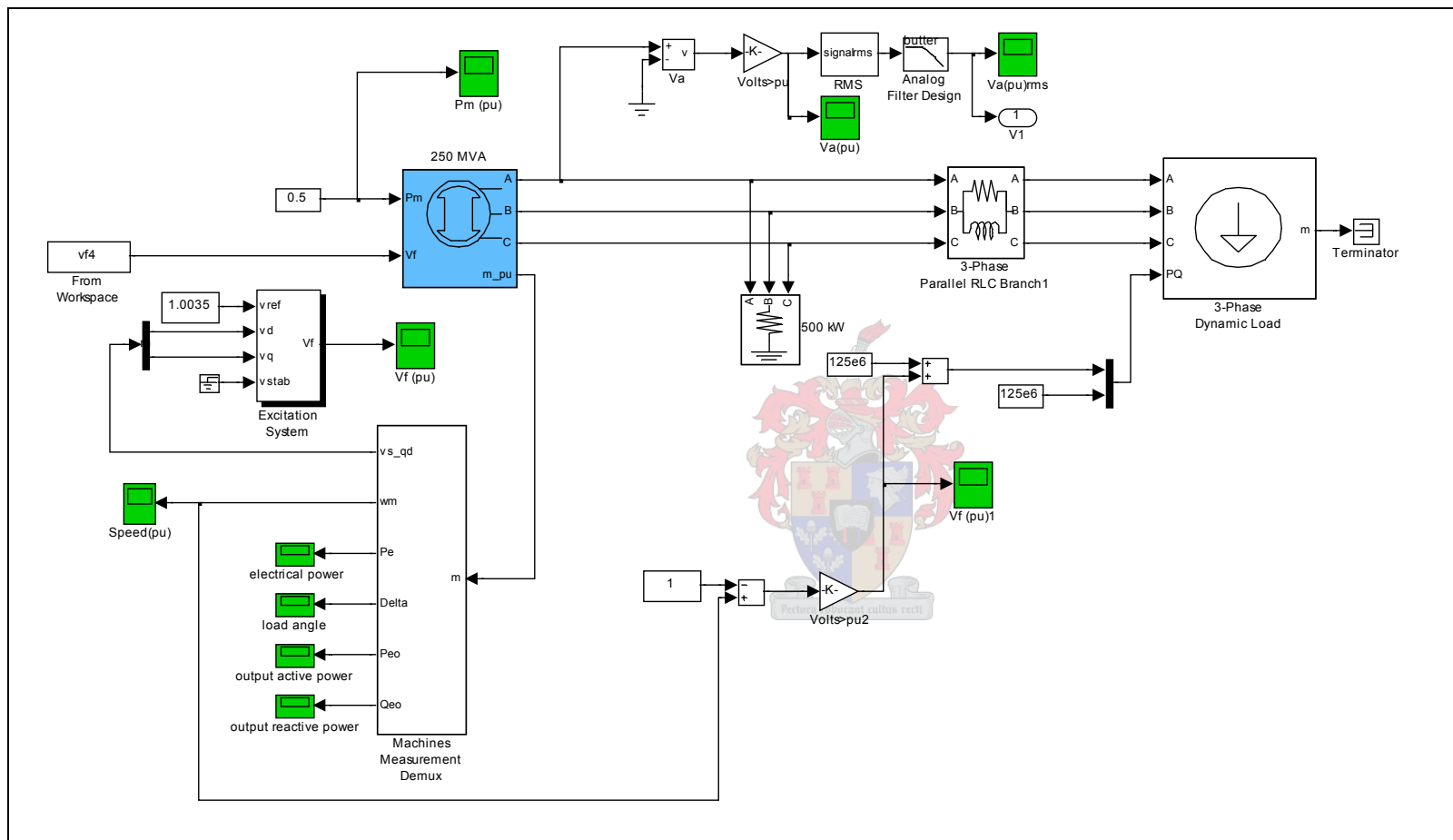
**RL Circuit method**



**RL Equation method**



50% Power export (P,Q)



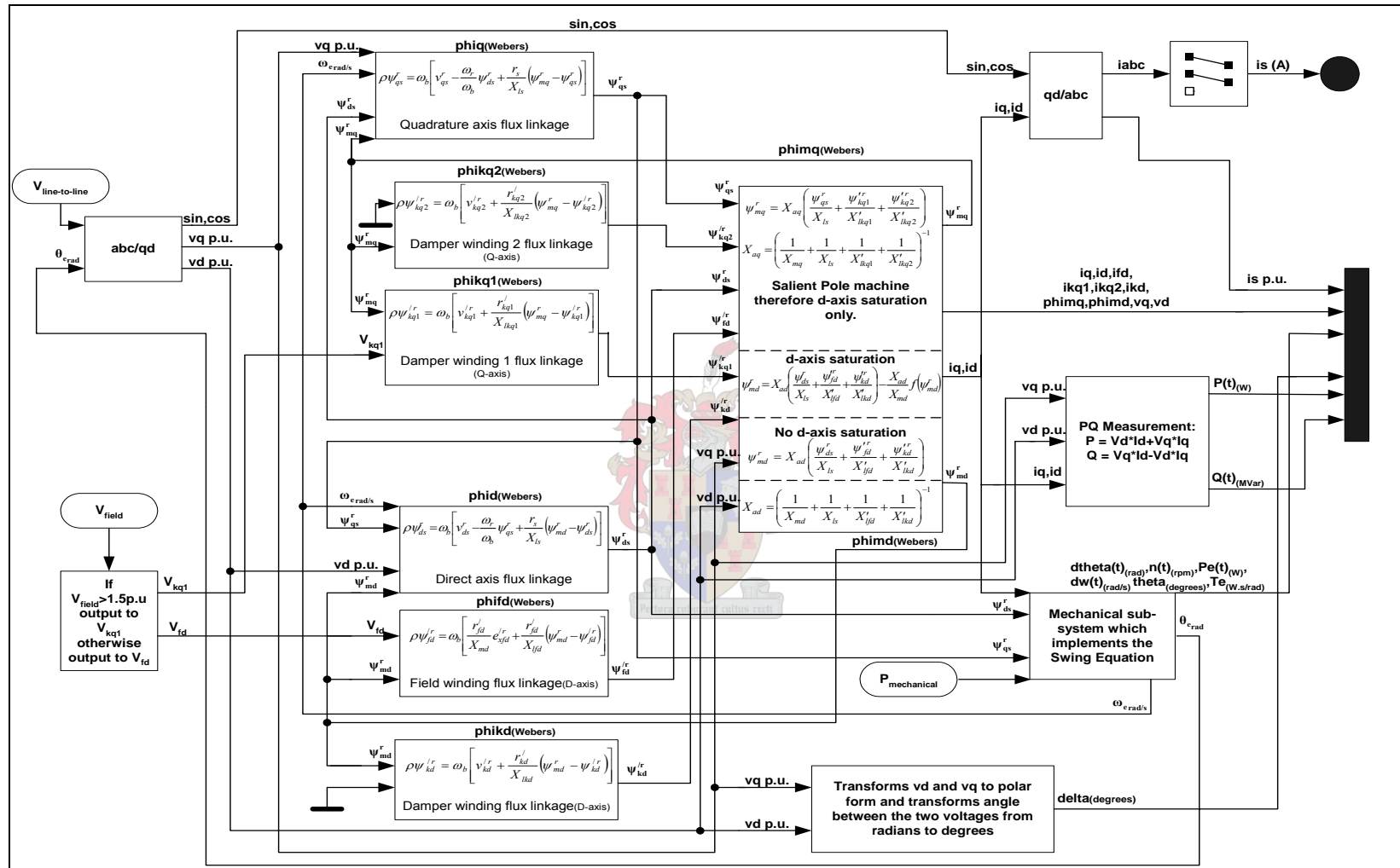
## Appendix E - System Components

### The Synchronous Machine

This component implements a 3-phase synchronous machine modeled in the dq-reference frame, it is connected in wye with an internal neutral point. The conversion from abc-dq-abc reference frames is implemented in Simulink and is described in the diagram on the following page. The model takes into account the dynamics of the stator, field, and damper windings. The equivalent circuit of the model is represented in the rotor reference frame (dq frame).

Development of the dq reference frame to be used when modeling the synchronous machine is discussed in the literature survey. This section primarily discusses its implementation in the Simulink environment.





The synchronous machine model accepts a number of parameters in its dialog box. The following table lists all the data needed to fully describe the machine in the MATLAB<sup>®</sup> environment:

**Table E.1 Synchronous machine parameters.**

	Parameters	Test Case - Palmiet
<b>Ratings</b>	3 Phase MVA [MVA]	250
	Stator line voltage [RMS]	1.65E+04
	System Frequency [Hz]	50
<b>Reactances [p.u.]</b>	Synchronous d axis - $x_d$	1.23
	Synchronous q axis - $x_q$	0.78
	Transient d axis - $x_d'$	0.23
	Subtransient d axis - $x_d''$	0.16
	Subtransient q axis - $x_q''$	0.16
	Armature-leakage - $x_l$	0.13
<b>Time constants [s]</b>	Field open circuit	14.1
	Subtransient d axis open circuit time constant	0.09
	Subtransient q axis open circuit time constant	0.22
<b>Other</b>	Stator resistance [ $\Omega$ ]	1.62E-03
	Coefficient of inertia [s]	5.6
	Pole pairs	20

The synchronous reactances for the machine model consist of the d-axis synchronous reactance  $X_d$ , transient reactance  $X_d'$ , and subtransient reactance  $X_d''$ , the q-axis synchronous reactance  $X_q$ , transient reactance  $X_q'$  (only if round rotor), and subtransient reactance  $X_q''$ , and finally the leakage reactance  $X_l$  (all in p.u.).

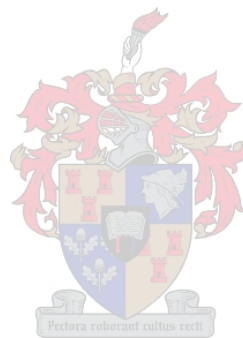
The figure on the previous page describes how MATLAB<sup>®</sup> models the synchronous machine. Using the line voltage specified in the table, above, the conversion from the abc to the dq reference frame is initially carried out. The electrical angle between the d axis and the reference phase is also fed back from the mechanical subsystem in order to calculate the conversion from the abc-dq reference frame.

The next stage requires the dq voltages, speed of the machine (rad/s) and the mutual flux linkages in the various axes to convert the flux linkages from the stator to the rotating reference frame. Thus the quadrature axis, direct axis and field winding flux linkages are amongst the most important values to be calculated here. The flux linkages for the damper circuits are also determined in this stage.

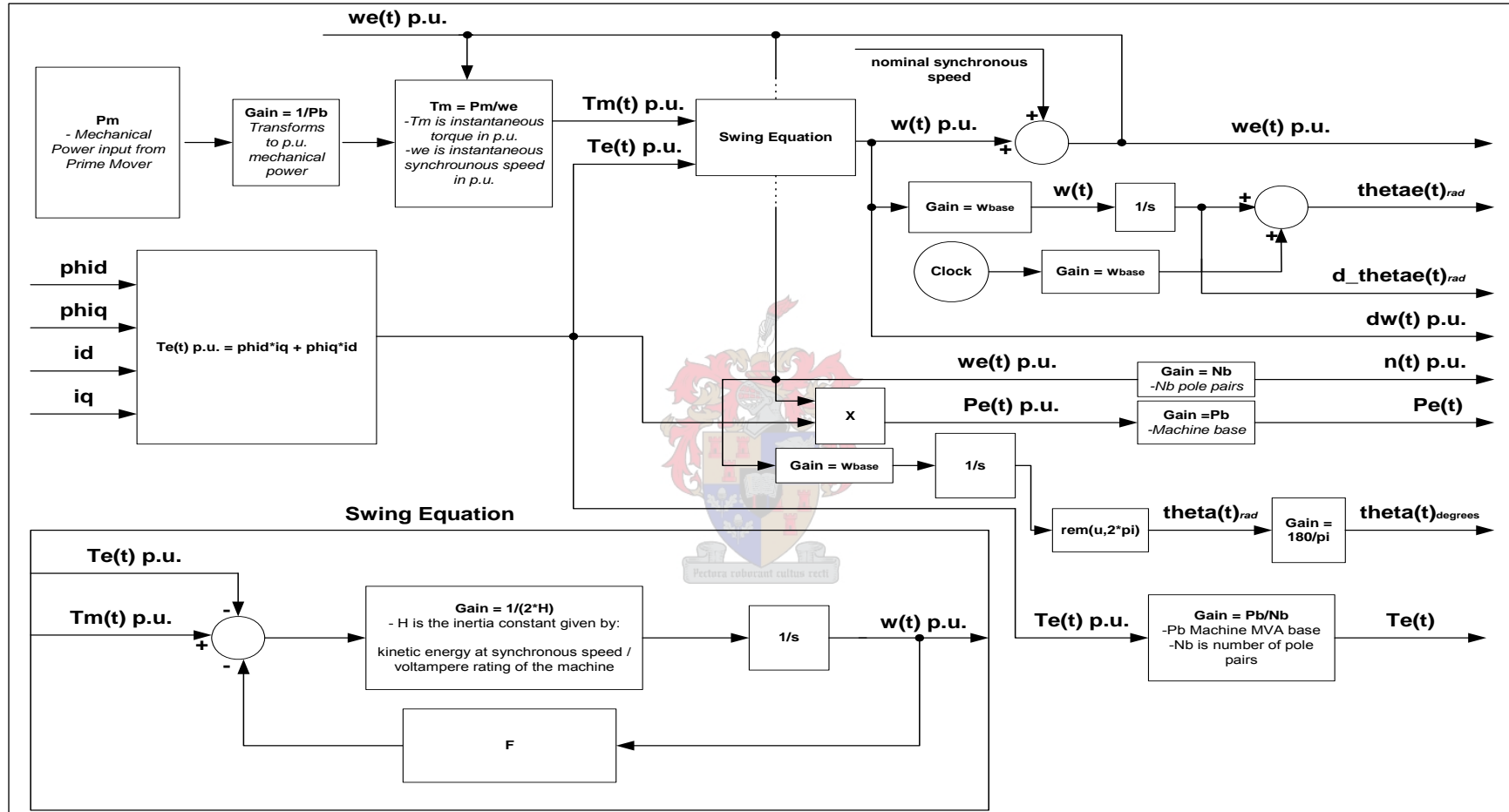
The third step calculates the the mutual inductances in both axes as well as the current in the dq axes. In order for this to be done all flux linkages calculated in the previous stage as well as  $v_q$  and  $v_d$  are needed. As this test case involves a salient pole machine there is only saturation in the d axis.

The final stage prepares all data for outputting. The currents in the dq reference frame are converted to line currents, the active and reactive output power is calculated and the mechanical subsystem also determines a number of parameters for output to the main system. This is discussed in the next section.

The other two inputs to the machine model are the field voltage and the mechanical power supplied by the prime mover.







The block diagram on the previous page describes how the mechanical subsystem is implemented. This specific system can be found in the synchronous machine model block diagram.

The mechanical subsystem essentially implements the swing equation and then outputs important data used in the parent block. Mechanical power inputted from a prime mover (governor system or constant block) is transformed to mechanical torque and together with the electrical torque the swing equation is implemented to produce the instantaneous synchronous speed of the machine in per unit quantities. The electrical torque is dependent upon the combination of the flux linkages and currents in the dq axis reference frame.

A number of essential parameters are calculated from the synchronous speed. These range from the instantaneous electrical torque and electrical power to the change in synchronous speed.

