

Graduate Theses, Dissertations, and Problem Reports

2015

Parameter identification of induction motor

Yousef Abdel-Rehim

Follow this and additional works at: https://researchrepository.wvu.edu/etd

Recommended Citation

Abdel-Rehim, Yousef, "Parameter identification of induction motor" (2015). *Graduate Theses, Dissertations, and Problem Reports.* 5018. https://researchrepository.wvu.edu/etd/5018

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

PARAMETER IDENTIFICATION OF INDUCTION MOTOR

by

Yousef Abdel-Rehim

Thesis submitted to the College of Engineering and Mineral Resources at West Virginia University In partial fulfillment of the requirements for the degree of

> Masters of Science in Electrical Engineering

Prof. Muhammad A. Choudhry, PhD, Chairperson Prof. Hany H. Ammar, PhD Prof. Natalia A. Schmid, PhD

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia 2015

Keywords: Induction motor, genetic algorithm, particle swarm optimization, parameter identification, nonlinear modeling, LabVIEW software

Copyright 2015 Yousef Abdel-Rehim

Abstract

PARAMETER IDENTIFICATION OF INDUCTION MOTOR

by Yousef Abdel-Rehim

Numerous recent techniques of induction motor parameters calculating are hard to be done and expensive. Accurate calculations of the parameters of these motors would allow savings in different prospective like energy and cost. The major problem in calculating induction motor parameters is that it's hard to measure output power precisely and without harm during the operation of the machines. It will be better to find other way to find out the output power with certain amount of inputs like input voltage and current.

Particle swarm optimization (PSO) and genetic algorithms (GAs) are often used to estimate quantities from limited information. They belong to a class of "weak" search procedures, that is, they do not provide the "best" solution, but one close to it. It is a randomized process in which follows the principles of evolution.

In this thesis genetic algorithm and partial swarm optimization are used to identify induction motor parameters. The inputs used to estimate electrical and mechanical parameters are measured stator voltages and currents. The estimated parameters compare well with the actual parameters. Data Acquisition (DAQ) is used to obtain these variable with the help of LABVIEW software. The induction motor used is a 7.5-hp with a constant frequency and in free acceleration. IEEE standard test of 7.5-hp induction motor is used to compare with performance of the simulated and measured data obtained. According to the output results, method of optimizing induction machine can be used in different models of induction motor

Acknowledgments

I would like to take this opportunity to express my sincere gratitude to my advisor Dr. Muhammad A. Choudhry whose guidance and encouragement was instrumental in the successful completion of my work.

I would also like to thank my committee members Dr. Hany H. Ammar and Dr. Natalia A. Schmid for supportive advice and discussion. Furthermore, I am very thankful to professors and staff at Lane Department of Computer Science and Electrical Engineering (LCSEE) for making my work such an enjoyable and memorable experience.

Special thanks goes to my family for their inspiration and encouragement during this work. None of my accomplishments would have been possible without their endless love.

Contents

Abstract		ii
Acknowle	dgments	iii
List of Fig	gures	6
List of Tal	bles	
Chapter 1:	Introduction	9
1.1	Thesis Outline	9
1.2	Literature Survey	
Chapter 2:	Induction motor	
2.1	Principle of operation of induction motor:	
2.2	Types of tests used	
2.2	2.1 Locked rotor test:	12
2.3	Iron core loss Effect <i>Rc</i> : Error! Bookmark not	defined.
2.4	System Model of Induction Motor:	14
2.4	4.1 Objective	18
Chapter 3:	Particle Swarm optimization	
3.1	Theory	
3.2	Parameters identification	
3.3	PSO steps of operation:	
3.4	Advantage and disadvantage of PSO:	
Chapter 4:	Genetic Algorithm	
4.1 Mu	utation and Crossover	
4.1	1.1 Crossover	23
4.2	2.2 Mutation:	27
4.3 Fit	ness Function:	
Chapter 5:	Data Acquisition System	
5.1	Data Acquisition System (DAQ)	
5.2	Lab VIEW Software	
5.2	2.1 Front Panel User Interface	31
5.2	2.2 Block Diagram	

5.2.3 Lab VIEW Features	.31
Chapter 6: Case Study and Results	.35
6.1 Case Study	. 35
6.2 Experimental Test Procedure	. 35
6.2.1 Calculations:	36
6.4 Results	. 38
6.4 Results of 7.5-hp Motor	.43
6.5 Comparison between GA and PSO Algorithms	.46
Chapter 7 Conclusion and future work	.47
Appendix A: Hardware & Software	.48
Appendix B: Code	.51
GA Code	.51
FLUX	.55
PSO Code	. 57
References	. 62

List of Figures

Figure 2.1: Equivalent model for Induction machine12	2
Figure 2.2 effect with and without <i>Rc</i> 14	4
Figure 2.3 <i>Rc</i> Effect14	4
Figure 2.4 dqo equivalent circuit for induction motor10	б
Figure 5: One point Crossover	3
Figure 4.6: One point crossover for offspring	3
Figure 4.7: Two Point Crossover	4
Figure 4. 8: Continuous two point crossover for offspring24	4
Figure 4.9: Uniform crossover2	5
Figure 4.10: Continuous Uniform Crossover2	5
Figure 4.11: Least popular crossover	б
Figure 4.12: Arithmetic Crossover	7
Figure 4.13: Single Point Crossover and Mutation2	7
Figure 4.14: Multi-point Crossover and Mutation23	8
Figure 4.15: Example for fitness function	9
Figure 5.16: Data acquisition block diagram for a three-phase system	0
Figure 5.17: GUI in LabVIEW	2
Figure 5.18: Three phase voltage and current Output from LabVIEW for 7.5-hp machine 3.	3
Figure 5.19: Rotor speed data from LabVIEW for 7.5-hp machine	4
Figure 6.20: Rotor Speed for 3-hp machine40	0
Figure 6.21: Speed vs Torque characteristics for 3-hp machine40	0
Figure 6.22: Stator current Ias, for 3-hp machine	1
Figure 6.23: Stator current Ias, for 2250-hp machine	1
Figure 6.24: Torque, Te for 2250-hp machine42	2
Figure 6.25: Rotor Speed vs Torque, for 2250-hp machine42	2
Figure 6. 26: Rotor Speed, for 2250-hp machine4	3
Figure 6.27 Simulated and measured stator currents of 7.5-hp induction motor44	4
Figure 6.28. GAs estimated and no-load and blocked-rotor test data estimated stator current of 7.5-hp induction motor4	s 5

Figure 6. 29. Estimated and measured rotor speeds of 7.5-hp induction motor	45
Figure A. 30: Name plate for 7.5-hp Induction machine	48
Figure A.31: Physical Connection for Transducers	49
Figure A.32: Voltage Transducer	50
Figure A.33: Current Transducer	50

List of Tables

Table 1: Induction Machine Model Parameters	15
Table 2: No load, blocked rotor and DC resistance test results	35
Table 3: Standard test parameters for 7.5-hp induction motor	37
Table 4: Test parameters for 3-hp induction motor $(J = 0.089 \text{kgm}^2)$	38
Table 5: Test parameters for 50-hp induction motor $(J = 1.662 \text{kgm}^2)$	38
Table 6: Test parameters for 2250-hp induction motor (J = 63.87 kgm ²)	
Table 7 Results for 7.5-hp machine $(J = 0.5 \text{kgm}^2)$	44

Chapter 1: Introduction

In the United States, electric motors use most of the electricity produced. Most of these motors are induction motors because induction motors are efficient. In order to increase the efficiency of induction motor, it has to be upgraded with some features. Information, like an accurate motor parameters and machine model, must be known in order to improve its function. In case of inaccurate parameters, the predicted output will be inaccurate. Examples of outputs are Torque and Flux.

Knowing that the induction motor is nonlinear, variables of the rotor are unknown which in turn makes most of the parameters unknown too. Therefore, in order to get all of these parameter, we have to get it through another way by using known values and control it as well. On doing this, we can control other parameters that are unknown. These known parameters used are input voltage of stator, current and speed of motor.

Great methods are created to generate the best values for the unknown parameter through using Genetic algorithm (GA) and Particle optimization Swarm (PSO) and different methods of estimation. In this thesis, two optimization methods will be used, Genetic Algorithm (GA) and Particle Swarm Optimization. Identification of induction motor parameters will be performed by using DAQ card which will be connected to a personal computer (PC) to observe and calculate the unknown parameters.

1.1 Thesis Outline

This thesis is designed to identify the parameters of induction motor by different methods.

Chapter 2 will present what is induction motor and nonlinear induction motor model and how to solve it through Simulink/Matlab.

Chapter 3 will introduce PSO and how it will be used to estimate parameters of the induction motor.

Chapter 4 will introduce GA and how it will be used to estimate parameters of the induction motor.

Chapter 5 will show the results and case study done by the use of Data Acquisition card and LabVIEW to know the output parameters from induction motor.

And finally, Chapter 6 will show the implementation of PSO and GA to optimize parameters of induction motor, estimated results and evaluation.

1.2 Literature Survey

To be able to identify the unknown parameters of induction motor, two main tests will be done on the induction motor according to IEEE test procedure. These tests are no load test and block rotor test [36].

In [3], a balanced three phase induction motor was used as a model. From this model, parameters were estimated. Some measurements were expected to be able to identify procedures like stator current and stator voltage.

In early 90's, two scientists, Richard R. Bishop and Gill G.Richards, introduced GA to identify the parameter of induction motor by using steady state mode. They used some known parameter like stator resistance and rotor reactance from the induction motor user manual to find other parameters. For these tests they used a three phase induction motor with 0.5 hp. These tests measure the motor terminal impedance Zm and Torque Tm at a rated voltage. These scientists succeeded to prove that GA can be used to identify unknown parameters [31].

[47] Stated that by using steady state models, motor parameter can be identified. Fitness function was introduced while trying to do identify the parameter. Fitness function is defined as the least error between the torques created and input torque. Here came the big advantage of GA which was used to get this least error .In this situation the parameters of the motor were assumed to be constant. This aimed to use the parameters where precision wasn't necessary. This method showed that performance of genetic algorithm was affected by those constant value. By applying the parameter as constant on 5-hp, a huge error (20%) was noticed. In case of deep bar model, it reduced from 3% to 6%. They concluded that accurate values must be used to get accurate outs with low errors.

[17] and [18] talked about creating a solution to identify parameter of induction motor by using qdo axis for modeling of induction motor. The induction motor that was used was 1kw. Genetic algorithm was applied to identify the parameters. The genetic algorithm was compared with a technique called analytical least square to identify induction motor parameter. It was noticed that quality of the parameters depend on accuracy of equipment used in the test for the motor.

[48] Demonstrated the use of genetic algorithm to identify induction motor by adding 4 different levels of noise. The mathematical model used was Kron's voltage equation. To check the efficacy of Genetic algorithm, another method, simple Random search (SRS), was used under the same condition used in genetic algorithm. It was noticed that the values of Genetic algorithm parameter where more accurate and efficient than SRS.

The fast increase in calculating power by using personal computers allowed researchers to implement several optimization algorithms and verify their efficiency. Researchers developed many algorithms that mimic natural phenomena. Examples of these algorithms include the Simulated Annealing Genetics Algorithms (GA), Ant Colony Optimization Algorithms. As genetic algorithm was spring an estimation method called Particle Swarm optimization (PSO) was created in 1995 by Dr. Eberhart and Dr. Kennedy based on the animal society which didn't have leader like fish schooling and birds flocking [37].

Chapter 2: Induction motor

This thesis will talk about parameter identification of induction motor using genetic algorithm and particle swarm optimization. Before we get involved more in the complicated detail, we have to talk more about some fundamentals to understand the complicated parts in this thesis.

First we have to know what the difference between motors and generators is and how they operate differently, and what is needed to able to perform the thesis in the right way.

Motor is a device used to convert electricity to mechanical movement. Most mechanical motions happen due to interaction between electric motor's magnetic field and winding current to create force of motion and generator is the vice versa.

In our research we will do our experiment on motor more specifically on induction motor which is an AC electric motor. When current enters the rotor, it produces torque due to interaction between current and magnetic field created by stator winding.

2.1 Principle of operation of induction motor:

In induction motor, when supplying an AC power to the motor's stator, a magnetic field will be created. The rotor speed will not be the same as field created from the stator. So, the rotor of the induction motor will have a lower speed than stator magnetic field. That will lead of a creation current in the rotor which will affect the secondary winding of the motor. Currents will be created in the rotor winding when the magnetic flux induced. The currents created in the rotor will lead to creation of magnetic field in the rotor which will react against the stator field .

Induction motors are the most used in the industry as they are not difficult to build. They also have great advantage as they are reliable and have self-starting capability. To be able to control induction motor, some of the parameters should be known. These known parameters must be very accurate. Inaccuracy may lead to worsening of the motor performance.



Figure 2.1: Equivalent model for Induction machine

It was found that the easiest way to get parameters of induction motor is to execute locked rotor test and no-load test. The idea of using linear parameter estimation technique is to find out ' R_1 ' which is stator resistance, ' X_1 ' which is stator leakage-reactance, ' X_m ' magnetizing reactance, ' X'_2 ' which is rotor self-reactance and ' R'_2 ' rotor resistance.

2.2 Types of tests used

- Locked Rotor Test
- No Load Test

2.2.1 Locked rotor test:

Locked rotor test is also known as blocked rotor test and stalled toque test. In this test, voltage applied can't be rated voltage, as the rated voltage is 220. This voltage is very high in case of locked rotor test, as when this voltage is applied to the motor while it is locked, a huge current will pass in the stator winding which will lead to overheat and damage the motor. Therefore, the applied voltage must be low and test time should be short time.

In block rotor test, the rotor will be locked to avoid its motion, leading to production of slip factor s=1 that will make $R'_2/s=R'_2$. This will make R'_2 become much less resistance than R_c and therefore, R_c and X_m parts of the circuit will be removed or ignored due to their huge resistance and reactance.

No load Test:

No load test is also known as open circuit test. A balanced rated voltage will be applied to the stator winding for a No load test. The motor speed will be almost at synchronous speed. Synchronous

speed will lead to s= very small value, and therefore according to figure, $R'_2/s=$ Very big value, removal of the part which include X'_2 and R'_2 will occur as their value is huge. In the end, all current will pass only in R_c and X_m

From both test locked rotor test and no load test certain parameter can be calculated like R_1 , X_1 , X_m , X_2 and R'_2 which are stator resistance, stator leakage-reactance, magnetizing reactance, rotor leakage -reactance and rotor resistance respectively

For identifying induction motor parameters, Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) are used [5], [6], [37] .The aim of using those two algorithms to get electrical and mechanical parameters of the nonlinear motor model. To identify the parameters 3, 50 and 2250-hp models are used. In lab experiment the model used is 7.5-hp induction motor.

2.3 Iron core loss effect:

- 1. R_c is added to account for hysteresis and eddy current losses
- 2. Iron core losses causes a reduction in performance.
- 3. The use of static converter for electrical drives leads to iron losses increase.
- 4. Magnetic saturation can be viewed as a reduction in the value of mutual induction and thus will increase magnetizing current.
- 5. The addition of R_c will make the model more accurate than a simplified model.

2.3.1How to decrease the R_c effect:

Eddy current losses are minimized by using lamination. By using lamination the core area decreases and resistance increases, which results in decrease in eddy currents. Hysteresis losses are minimized by using high grad silicon steel.

2.3.2 The R_c effect on 7.5-hp induction motor model:



 R_c value calculated was 63.5 Ω . when we applied the R_c to simulink model the output was almost the same as without R_c and that can be see below in figure 2.2 and 2.3.

Figure 2.3 R_c Effect

2.4 System Model of Induction Motor:

In this research the study will be done on a balanced three phase induction motor. The voltage and current are at synchronous rotating reference frame. The rotor parameters are referred to the stator. From stator current and rotor current, flux can be measured [45]. Table 1, 2 summarizes the name of each parameter that will to be used in our research.

$R_1 = R_s$	stator resistance
$X_1 = X_{ls}$	stator leakage-reactance
X _m	magnetizing reactance
$X'_2 = X'_{lr}$	rotor leakage -reactance
$R'_2 = R'_r$	rotor resistance

Table 1: Induction Machine Model Parameters

Table 2: Induction Machine Model Parameters

	2002.000	d avis stator voltage
Uds		d-axis stator voltage
Vqs		q-axis stator voltage
v_{dr}		d-axis rotor voltage
Vqr		q-axis rotor voltage
ids		d-axis stator current
iqs		q-axis stator current
idr		d-axis rotor current
igr		q-axis rotor current
ψ_{ds}		d-axis stator flux
ψ_{qs}		q-axis stator flux
ψ_{dr}		d-axis rotor flux
ψ_{qr}		q-axis rotor flux
r_s		stator resistance per phase
r_r		rotor resistance per phase
Xmd		d-axis magnetizing reactance
X_{mq}		q-axis magnetizing reactance
ω_r		rotor electric angular speed
ω_b		nominal rotor electric angular speed
T_e		electrical torque
T_L		load torque
H		inertia moment



Figure 2.4 dqo equivalent circuit for induction motor

The stator currents for induction machine are given by;

$$i_{qs} = \frac{1}{X_{ls}} (\psi_{qs} - \psi_{mq})$$
$$i_{ds} = \frac{1}{X_{ls}} (\psi_{ds} - \psi_{md})$$

The rotor currents are given by:

$$i'_{qr} = \frac{1}{X'_{lr}}(\psi'_{qs} - \psi_{mq})$$
$$i'_{dr} = \frac{1}{X'_{lr}}(\psi'_{ds} - \psi_{md})$$

The mutual flux is;

$$\psi_{mq} = X_m \left(i_{qs} + i'_{qr} \right)$$

$$\psi_{md} = X_m \left(i_{ds} + i'_{dr} \right)$$

The stator and rotor flux are given by;

$$\frac{d\psi_{qs}}{dt} = \omega_b \left[v_{qs} - \frac{\omega}{\omega_b} \psi_{ds} + \frac{r_r}{X_{ls}} (\psi_{mq} - \psi_{qs}) \right]$$
$$\frac{d\psi_{ds}}{dt} = \omega_b \left[v_{qs} + \frac{\omega}{\omega_b} \psi_{qs} + \frac{r_r}{X_{ls}} (\psi_{md} - \psi_{ds}) \right]$$
$$\frac{d\psi'_{qr}}{dt} = \omega_b \left[v'_{qs} - \frac{\omega - \omega_r}{\omega_b} \psi'_{ds} + \frac{r'_r}{X'_{lr}} (\psi_{mq} - \psi'_{qr}) \right]$$
$$\frac{d\psi'_{dr}}{dt} = \omega_b \left[v'_{ds} + \frac{\omega - \omega_r}{\omega_b} \psi'_{qs} + \frac{r'_r}{X'_{lr}} (\psi_{md} - \psi'_{dr}) \right]$$
$$\psi_{mq} = X_{aq} \left(\frac{\psi_{qs}}{X_{ls}} + \frac{\psi'_{qr}}{X'_{lr}} \right)$$
$$\psi_{md} = X_{ad} \left(\frac{\psi_{ds}}{X_{ls}} + \frac{\psi'_{dr}}{X'_{lr}} \right)$$

$$\psi_{aq} = X_{ad} = \frac{1}{\frac{1}{\frac{1}{X_m} + \frac{1}{X_{ls}} + \frac{1}{X_{lr}'}}}$$

The Torque, T_e is given by

$$T_e = 2H \frac{d}{dt} \frac{\omega_r}{\omega_b} + T_L$$

$$T_e = \psi_{ds} i_{qs} - \psi_{qs} i_{dr}$$

The input voltage applied to stator are balanced and rotor voltage is zero, which lead to bring up speed and stator current

2.4.1 Objective

Identification of induction motor parameters are the main objective of our experiment .The basic idea is to compare time dependent response of the model and model estimated by using error criterion. The objective function can be obtained from the output error and estimated model. It tells how good the estimated model response is and detects the match with system model response. The structure of the estimated model is same as the system model except that the estimated model has unknown parameter

Chapter 3: Particle Swarm optimization

Particle Swarm Optimization theory was created in 1995 by Dr. Eberhat and Dr. Kenny .The idea was inspired from nature of the social behavior and dynamic movement in the communication of birds flocking or fish schooling [37].

3.1 Theory

Particle swarm optimization was used to solve several problems .The idea of PSO algorithm came from the animal society that didn't have leader in their group, like fish schooling and bird flocking. For example, each bird in the flock will search randomly for food, when a bird finds food at a certain position, it leads the whole flock to the closet food place. The success of the team depends on the best direction and shortest distance in finding food, which in turn depends on good communication between them. The process of searching food happens continuously until food is discovered. The technique of partial swarm optimization is used to find the optimal value and best solution.

Particle swarm optimization consists of a number of particles. Each particle has a certain position in space [38]. The particle status act:

- 1) To keep its inertia
- 2) To change condition according to its most optimization position
- 3) To change condition according to the swarm's most optimist position.

Each particle in space adjusts its movement according to its own moving experience with another particle.

There are two experiences in particle swarm optimization, the 'individual experience' and the 'near experience'. The 'individual experience' is the position of each particle in swarm and it is affected by the most optimist position during its movement. The 'near experience' is the position of the most optimist particle in its surrounding.

Each particle keeps track of where it was at the best fitness. This is called 'Pbest'. Another "best" value is tracked by the particle swarm optimizer. It is obtained by any particle in the population. This best value is the global best and is called gbest. When a particle takes part of the population as its topological neighbor, the best value is then the local best and is called lbest. Each particle has a position and velocity. The information can be updated by changing the velocity and position.

In the process, we initiate the population including the value and velocity, we look at the fitness then we modify the velocity based on the pervious pbest, global best and neighborhood best and we loop around terminating a good round solution.

$$v = K[v + c1 * rand(pbest - Partical) + c2 * rand(gbest - Particle)]$$

 $\theta = \theta + v\phi = c1 + c2$

$$K = \frac{2}{\left|2 - \phi - \sqrt{\phi^2 - 4\phi}\right|}$$

The states for measuring for induction machine are:

$$\hat{\Theta} = \begin{bmatrix} \hat{r}_s & \hat{X}_{ls} & \hat{X}_m & \hat{X}'_{lr} & \hat{r}'_r & \hat{H} \end{bmatrix}.$$

3.2 Parameters identification

Particle update rule: p = p + vWhere p: particle's position v: velocity c1: weight of local information c2: weight of global information pbest: best position of the particle gbest: best position of the swarm rand: random variable

3.3 PSO steps of operation:

Algorithm of PSO:

- **Step 1:** Define fitness function.
- Step 2: Generate initial population of N particles with random position and velocities.
- **Step 3:** Calculate fitness: Evaluate the fitness values of current particle using the objective function
- **Step 4:** Update pbest- Compare the fitness values of each particle with its pbests. If current value is better than pbest then set it to current value.
- **Step 5:** Update gbest- Compare the fitness values of each particle with its pbests. If current value is better than gbest then set it to current value.
- **Step 6:** If the number of iterations reaches the maximum, then go to step 8 otherwise go to step 4.
- Step 7: The particle that generates the latest gbest is the solution of the problem.
- Step 8: Stop

3.4 Advantage and disadvantage of PSO:

Advantages:

- 1) Scaling insensitive of design
- 2) Its implementation is very simple
- 3) Few algorithms are used
- 4) Very Efficient

Disadvantages:

1) Slow convergence in refined search stage (weak local search ability)

Chapter 4: Genetic Algorithm

Dr. John Holland introduced Genetic Algorithm in1970's with the help of his colleagues and some students at university of Michigan. The idea of GA came from genetic and reproduction behavior observed in biological population. The ideology of GA came from 'survival of the fitness'. It searches a process to select and create individual that can be able to survive on the environment. After a number of iterations, design characteristic evolves and remains in the genome composition of the population over traits with weaker undesirable characteristics. GA has been known for its success in solving sophisticated optimization as it can work with nonlinear objective and can work with discrete and continues variables.

In IT application, GA are used widely in most applications and especially parallel genetic algorithm. The prospective that parallels Genetic Algorithm and deals with existing information technology is huge, which may control very large and variable data in real world. GA is idealistic to do this.

GA is the best method for optimization especially when little data is available and large size problems are present. GA uses the method of direct analogy of a natural evolution. It is assumed that any problem can be solved by having a number of parameters. These parameters are like genes of a chromosome which has a string of value, which correlates with the objective function of the problem.

In 1970, Dr. Fraser came up with idea of replicating a genetic system by using genetic operators. Dr. John Holland first introduced a simple Genetic Algorithm and discovered the usage of genetic operators schemes which lead to production of a new generation and the old generation died. Dr. Lawrence Davis highlighted the significance of evaluation function as the connection between Genetic Algorithm and problem to be solved. Dr. Lawrence recommended the following:

- 1. Initialization of a population of chromosomes.
- 2. Creation of new chromosomes by mating current chromosomes.
- 3. Mutation and reproduction as the parent chromosomes mate.
- 4. Evaluation of the new chromosomes and insertion into the population.
- 5. If time is up, stop and return the best chromosome. If not, go to step three.

Genetic Algorithm is divided into two operators which makes it unique and successful. These operators are 'mutation' and 'crossover' .It is predicted from these procedures that a better chromosome will create a large number of offspring which at the end is guaranteed a higher chance of surviving in subsequent generation. The sequence of evolution is repeated until desired execution measured is achieved.

4.1 Mutation and Crossover

It was mentioned before that Genetic Algorithm has 2 operators, the first is mutation and second is crossover.

Mutation: Is to modify the value of one or more gene in a chromosome.

Crossover: Is to reproduce propagation where two parents give birth to a new Offspring

4.1.1 Crossover

Crossover is a genetic operator used to do some changes in chromosomes to make it better. The methodology of Crossover is to take genes from different parents to create a child with better functionality and with no defects. Parents must be healthy There are different ways to approach this.

Some types are popular and used a lot in real like one-point crossover, two-point cross over, and uniform Crossover. Other least used crossovers are cut and splice and half uniform crossover. More over Crossover is also expressed in binary encoder method.

Most popular:

• One-point crossover:

It is a single crossover point from both parents. All points after certain point are identically swapped and the result is a child of both parents



Figure 4.6: One point crossover for offspring

Parent 1:	$X X \mid X XXXX$
Parent 2:	Y Y Y YYYY
	V V V VVVV
Offspring 1:	ΧΧΥΥΥΥΥ
Offspring 2:	Y Y X XXXX

- Encoding Part
- Parent A: 10010|11101
 Parent B: 01011|10110
 Child AB: 1001010110
 Child BA: 0101111101
- Two Point Crossover:

It is the choice of a certain partition from parents and then identically swapped it.



Figure 4. 8: Continuous two point crossover for offspring

Or

- Parent 1: X X | X XX | X X

- Parent 2: Y Y | Y YY | Y Y
- Offspring 1: X X Y YY X X
- Offspring 2: Y Y X XX Y Y

Encoding Part

- Parent A: 1 0 0|1 0 1 1| 1 0 1
- Parent B: 0 1 0* 1 1 1 0* 1 1 0
- Child AB: 1001110101
- Child BA: 0101011110

• Uniform crossover:

Different partitions will be taken from both parents with a probability of 0.5.



With a probability of 0.5, children have 50% genes from first parent and 50% of genes from second parent even with randomly chosen crossover points.

Figure 4.9: Uniform crossover

Continuous Uniform crossover:



Figure 4.10: Continuous Uniform Crossover

- Parent 1: XXXXXXX

		- Parent 2:	Y Y Y YY YY	
		- Offspring 1:	XYXYYXY	
		- Offspring 2:	YXYXXYX	
0	Encoding Part			
	Mask:	0110011000	(Randomly generated	1)
	Parents:	1010001110	0011010010	

Offspring: 0011001010 1010010110

• Least popular Crossover:

Cut and slice:

This is another type of crossover which changes in Length of children. This happen due to taking different length of data from each parent.



Figure 4.11: Least popular crossover

• Arithmetic Crossover:

It is a combination of different data from different parent which lead to totally different Offspring



11001011 + 11011111 = 11001011 (AND)

Figure 4.12: Arithmetic Crossover

4.2.2 Mutation:

Genetic algorithm is a binary code .Each bit in the chromosome is mutated with probability Pbm known as the mutation rate

Parent1	1	0	0	0	0	1	0	
Parent2	1	1	1	0	0	0	1	
Child1	1	0	0	1	0	0	1	
Child2	0	1	1	0	1	1	0	

An Example of Single-point Crossover Between the Third and Fourth Genes with a Mutation Rate of 0.01 Applied to Binary Coded Chromosomes



Figure 4.13: Single Point Crossover and Mutation



Figure 4.14: Multi-point Crossover and Mutation

The mutation operator ensure that gene pool does not become too restricted. GA is carried out by randomly changing one or more of bits in an individual. This may lead to creation of a totally new solution from the previous one, this means that GA can be better by using mutation

There are different types of mutation:

- Bit string mutation

- One of bits is chosen randomly and be flipped
- Flip Bit

•

- One genome will be chosen and change all its bits, which mean if a bit is 1 it will be flipped to 0 and vice versa.

- Boundary

The genome will be replaced with a random upper or lower bounds

4.3 Fitness Function:

The fitness Function measures the quality of represented solution over Genetic Representation. Also it is problem dependent. The illustration of the result might be an array of bits, where each bit represent a different object. The bit value donates an objective but not every illustration is valid.

Fitness function is used in each iteration of algorithm to value the superiority of all projected solution to the problem in current population. The fitness function rule is to evaluate the goodness of each solution in a population.



Figure 4.15: Example for fitness function

Chapter 5: Data Acquisition System

In this chapter a programmed method to find the input and output variables of an induction motor is labeled. The process starts by finding the hardware devices that are used for the data acquisition system (DAQ). The technique, created as a virtual instrument (VI), is done in the Lab VIEW setting to measure and save the signals of the rotor speed, three - phase currents and three - phase voltages, of the induction motor. The technical features of the Lab VIEW and VI observing system supply huge help in parameter identification of an induction machine since it gives material on developed signals needed for the identification problem to be cracked.

5.1 Data Acquisition System (DAQ)

The data acquisition (DAQ) VI library attains waveforms and produces signals with all National Instruments (NI) plug-in and remote data acquisition products. The data acquisition system used here has a NI DAQ Card (6040E Families) [35]). The Input/output Connector of the DAQ card has 68 pins for the analog and digital signals. Six analog inputs in the range of 10 volts To Measure three-phase voltages and three-phase current. So, certain type of transducers must be employed to convert high amplitude signals within the range. For voltage VT7 and two different types of transducers are used for currents, CTL - 51 and CTA201R. Analog output is provided by a digital tachometer (speed transducer) to measure the speed, which converts speed in the range of 50 – 10000 rpm to 0 to 5 volts. The way in which analog inputs are connected to the DAQ card is given in Appendix A.31..



Figure 5.16: Data acquisition block diagram for a three-phase system

DAQ systems are used in a different ranges of applications in the research workroom, in the field, and in manufacturing plants. A general DAQ system includes:

Transducers - connected to induction motor being measured. For three phase voltages of induction motor – transducers convert input voltage in the range of 600 V to analog output voltage in the range of 10 Volts.

5.2 Lab VIEW Software

Lab VIEW is a visual programming language for display of measured data and analysis. The software was firstly produced by NI. These programs are called Virtual Instruments (VI) since they actually imitate physical instruments. VIs look like the functions in textual programming language, where the functions are used instead of icons. Each VI in Lab VIEW consists of:

- Front Panel
- Block Diagram

5.2.1 Front Panel User Interface

Front panel is the GUI for creating commands for showing instruments and measurements results. This interface gathers user input and shows program output. Control objects are used for data input, while indicator objects are used for data output. Front panel is very suitable to create and to work with. One example of Front Panel and VI designed in the power laboratory for the goal of obtaining stator currents, stator voltages and speed of the rotor shaft of induction motor is shown in the Figure 5.18. Figure 5.19 shows front panel configurations for reading and displaying saved data from the file.

5.2.2 Block Diagram

The *block diagram* has the graphical source code of VIs. All the functions shown in the *front panel* must be controlled and performed with in *block diagram*. The *block diagrams* designed for obtaining signals from the induction motor are shown in the Figures (5.17

5.2.3 Lab VIEW Features

Lab VIEW VIs are modular in design, so any VI can run on its own or be used as part of another VI (subVI). With this modularity, it gives the operator flexibility to design a whole hierarchy of VIs and subVIs that serve as building blocks in any number of applications. Also, it provides easy change, interchange, and combination of VIs in order to provide accommodations for other applications. Furthermore enhance performance [38].



Figure 5.17: GUI in LabVIEW



Figure 5.18: Three phase voltage and current Output from LabVIEW for 7.5-hp machine



Figure 5.19: Rotor speed data from LabVIEW for 7.5-hp machine

Chapter 6: Case Study and Results

In this chapter the results for parameter estimation of induction machine using GA and PSO are presented. Also the results from the hardware test of machine are presented.

6.1 Case Study

This chapter discusses the experimental testing to the problem identification of electric parameters of induction motor. The standard test procedure of DC resistance, no load and blocked rotor is applied to the 7.5-hp induction motor to obtain parameters. Then, a free acceleration test is performed, and with the help of DAQ and Lab VIEW Software three - phase voltages, three - phase currents and speed are recorded. Next, the GA method and the PSO method is used to estimate parameters of induction motors. Thus, the method is based on the following:

- Experimental Test Procedure.
- Implementation of GA to the problem identification.
- Implementation of PSO to the problem identification.

6.2 Experimental Test Procedure

The first test applied for determination of parameters is based on IEEE standard procedure. The following table gives information on parameters obtained after performing DC resistance, no – load and blocked rotor tests of a three – phase, 7.5-hp, 60 Hz, 220 V, induction motor.

No Load Test	Blocked Rotor Test	DC Resistance
Vnl = 126.5V	Vbl = 18.63 V	Vdc = 11.74 V
Inl = 8.8 A	Ibl = 18.62 A	Idc = 20 A
Pnl = 274.4 W	Pbl = 221.33 KW	

Table 2: No load, blocked rotor and DC resistance tes	st
results	

Both no load and blocked rotor test are performed at 60 Hz frequency. According to the IEEE Standard Test Procedure given in [36], this case applies for the situation when the slip is near unity

thus, right after the machine start – up, when the starting currents have the largest values. No attempt was made in this work to compose the test at reduced frequency since there was no adjustable frequency drive in equipment setup.

The detailed methodology for calculating the parameters of the induction machine is discussed below:

6.2.1 Calculations:

No load test:

$$\cos(\phi_o) = \frac{P_{ph}}{V_{ph}I_o} = \frac{274.4}{126.3 * 8.8} = 0.24$$
$$\phi_o = 76.17$$

$$I_m = I_o \sin(\phi_o) = 8.8 \sin(76.17) = 8.74 A$$
$$I_c = I_o \cos(\phi_o) = 8.8 \cos(76.17) = 2.16 A$$
$$L_m = \frac{V_{ph}}{2 \pi f_s I_m} = \frac{126.3}{2 * \pi * 60 * 8.74} = 0.039 H$$
$$X_m = 14.35 \Omega$$

Rc calculation:

$$P_{fwc} = P_{nl} - P_{I_0 {}^{2}R_s}$$

$$P_{I_0 {}^{2}rs} = I_0 {}^{2}R_s$$

$$P_{fwc} = \frac{3 * |V_{ph}|^2}{R_c}$$

$$P_{nl} = 823.2 W$$

$$R_s = 0.3 \Omega$$

$$I_0 = 8.8A$$

$$P_{I_0 {}^{2}r_s} = 3 * I_0 {}^{2}R_s = 3 * (8.8)^2 * 0.3 = 69.696W$$

$$P_{fwc} = P_{nl} - P_{I_0 {}^{2}R_s} = 823.2 - 69.696 = 753.504W$$

$$R_c = 63.5 \Omega$$

Locked rotor test

$$\cos(\phi_{sc}) = \frac{P_{ph}}{V_{ph}I_{sc}} = \frac{221.33}{18.63 * 18.62} = 0.64$$
$$\phi_{sc} = 50.35$$
$$Z_{sc} = \frac{V_{ph}}{I_{sc}} = \frac{18.63}{18.62} \approx 1\Omega$$
$$R_{r=}Z_{sc}\cos(\phi_{sc}) - R_{s} = (1 * 0.64) - 0.3 = 0.34 \Omega$$
$$X_{eq} = Z_{sc}\sin(\phi_{sc}) = 1\sin(50.13) = 0.77 \Omega$$
$$X_{eq} = X_{ls} + X_{lr} = 0.77\Omega$$

$$X_{ls} = X_{lr} = 0.385 \Omega$$

Using the said methodology, the parameters of the induction machine were calculated and are tabulated in following table.

Table 3: Standard test parameters for 7.5-hp induction motor.

R _s	<i>X</i> ₁	X _m	<i>X</i> ₂	R _r
0.3	0.385	14.35	0.385	0.34

The next step is then applied to measure three - phase voltages, three – phase currents and rotor speed. Machine variables are recorded during the start- up conditions. Three phase voltages are generated from the Motor – Generator Set (MG-Set 1) available in Power Lab. A three-phase wound rotor induction motor rated at 7.5-hp is tested.

The configuration test for the experiment can be found in Chapter 5, where Data Acquisition System and Lab VIEW Software are employed for recording the waveforms. Voltage and current signals in the range of [-10 V, +10 V] are acquired by the means of voltage and current transducers. A digital transducer – tachometer is used to generate analog voltage input at the range of 0 to 5 volts.

After employing Lab VIEW Software, acquired signals are saved. Data collected are transformed to the files suitable for the identification method by the GAs. Source code of the GAs is listed in Appendix.

The boiler – plate data from the test of induction motor is given in Appendix. No mechanical load, except for the rotor inertia and windage, was attached to the motor.

6.4 Results

Before applying the genetic algorithm to the real experimental test data, a simulation for induction machine was implemented to prove the validity of the approach. The theoretical examples are chosen with widely different horsepower ratings in order to confirm applicability of the method to motors with various sizes. The following tables summarize the results obtained from running the GA and PSO for each machine. The estimated parameters agree very closely with the actual value.

	R _s	X ₁	X _m	X ₂	R _r
From GA	0.40	0.5615	25.3539	0.8161	0.7353
	-0.09	-0.25	-0.03	0.08	-0.098
From PSO	0.4445	0.7754	26.9727	0.7997	0.7990
	0.02	0.03	0.03	0.06	0.02
Actual	0.435	0.754	26.13	0.754	0.816

Table 4: Test parameters for 3-hp induction motor $(J = 0.089 \text{kgm}^2)$

Table 5: Test parameters for 50-hp induction motor $(J = 1.662 \text{kgm}^2)$

Data Set	R _s	<i>X</i> ₁	X _m	<i>X</i> ₂	R _r
From GA	0.019	0.3558	12.2320	0.3374	0.4507
	-0.78	0.2	-0.06	0.1	1
From PSO	0.0113	0.4992	12.7521	0.4967	0.2742
	-0.89	0.67	-0.015	0.67	0.18
Actual	0.087	0.302	13.08	0.302	0.228

Data Set	R _s	<i>X</i> ₁	X _m	X ₂	R _r
From GA	0.0200	0.1085	13.5626	0.2096	0.0317
Percentage error for GA	-0.31	-0.52	0.04	-0.07	0.44
From PSO	0.0209	0.1025	14.9895	0.2364	0.0372
Percentage error for PSO	-0.28	-0.5	0.15	0.09	0.5
Actual	0.029	0.226	13.04	0.226	0.022

Table 6: Test parameters for 2250-hp induction motor (J = 63.87kgm²)

From the tables given above it is obvious that the GA was able to identify parameters with a good accuracy. The errors between the responses of the induction motor with the actual parameters and the genetic algorithm identified parameters are small.

Figures (6.19 - 6.21) compare the actual parameter response with that predicted by using the GA and PSO identified parameters for 3-hp machine. Rotor speed, three-phase stator current and torque during transients are shown. The Stator voltages are the only input to the induction motor for these results.

Figures (6.22 - 6. 25) compare the actual parameter response with that predicted by using the GA and PSO identified parameters for 2250-hp machine. Rotor speed, three-phase stator current and torque during transients are shown. The Stator voltages are the only input to the induction motor for these results.



Figure 6.20: Rotor Speed for 3-hp machine



Figure 6.21: Speed vs Torque characteristics for 3-hp machine



Figure 6.22: Stator current Ias, for 3-hp machine



Figure 6.23: Stator current Ias, for 2250-hp machine



Figure 6.24: Torque, Te for 2250-hp machine



Figure 6.25: Rotor Speed vs Torque, for 2250-hp machine



Figure 6. 26: Rotor Speed, for 2250-hp machine

6.4 Results of 7.5-hp Motor

Free acceleration of 7.5-hp induction machine was performed and the data for 3-phase voltage and currents was recorded. Also the data for speed was measured for the said machine. The data was stored in Excel file and then transferred to Matlab. Using the data, the parameters of the machine were calculated and verified against the standard parameters of 7.5-hp induction machine. The results came very close to the values calculated using the blocked rotor test and no load test.

	R _s	<i>X</i> ₁	X _m	X ₂	R _r
From No-load and blocked-rotor test	0.3	0.385	14.35	0.385	0.34
From GA	0.2506	0.3616	14.1444	0.3811	0.2440
Percentage error for GA	-0.1647	-0.0608	-0.0143	-0.0102	-0.2824
From PSO	0.2770	0.3607	14.7466	0.3682	0.2115
Percentage error for PSO	-0.0767	-0.0631	0.0276	-0.0436	-0.377

Table 7 Results for 7.5-hp machine $(J = 0.5 \text{kgm}^2)$



Figure 6.27 Simulated and measured stator currents of 7.5-hp induction motor



Figure 6.28. GAs estimated and no-load and blocked-rotor test data estimated stator currents of 7.5-hp induction motor



Figure 6. 29. Estimated and measured rotor speeds of 7.5-hp induction motor.

6.5 Comparison between GA and PSO Algorithms

GA	PSO	
 Advantages: 1) Very large number of parameters 2) Solves problems with multiple solutions 3) GA is very easy to understand. 4) The Percentage error for GA is better in case of 50-hp,2250-hp and 7.5-hp 	 Advantages: 1) Scaling insensitive of design 2) Its implementation is very simple 3) Few algorithms are used 4) Very Efficient 5) The Percentage error for PSO is better in case of 3-hp 	
Disadvantages: GA may not find the global optimum	Disadvantages: Slow convergence in refined search stage (weak local search ability)	

Chapter 7 Conclusion and future work

This thesis has presented intelligent approaches to estimate the Induction Machine electrical parameters based on Particle Swarm optimization and Genetic Algorithm heuristic optimization approach, a simulation model and experimental transient measurements. Based on both the simulation and experimental measured results, it is concluded that the use of the proposed strategy is an effective and reliable method for induction motor parameter identification. The reliability of this technique in estimating the machine parameters of different rating of induction motor has been presented successfully.

Future work:

- 1. R_c should be included in dqo induction motor model and its effect should be investigated through simulation.
- 2. Induction motor parameters should be estimated using Nonlinear least square approach and its results should be compared with Genetic algorithm (GA), Particle swarm optimization (PSO) and real experiment.

Appendix A: Hardware & Software

Information on hardware tools and equipment's, which are used for acquiring the signals in the experiment is given in this section. The second part defines an overall purpose of Lab VIEW Software intended to simply obtain outside signals. It is included here because of the ability to further work on this subject.

Setup

The identification technique labeled in the previous chapters performed a 7.5-hp Induction Motor. The next figure demonstrations the image of the motor along with the plate information from the motor. When the machine is at halt without mechanical load, the three – phase AC power is supplied to the stator stations. The experimental setup along with current and voltage transducers is presented in the following Figure A.31. The three - phase stator voltages, three – phase stator currents and the rotor speed are recorded by the means of voltage and current transducers respectively and tachometer, which is used to measure the speed of the rotor shaft.



Figure A. 30: Name plate for 7.5-hp Induction machine

DAQ card used is 6040E .this DAQ has 68 pins as shown in the figure 28



Figure A.31: Physical Connection for Transducers

DAQ card in this experiment is used to measure three-phase voltage, three-phase current and speed. The maximum voltage that can be applied to the DAQ card is 10 volt. If this input voltage exceed 10 Volts it may damage DAQ card. Transducers used in our experiment step down voltage from 220 to 5v maximum. This output of transducer will be measured through the DAQ card.

As it was said before, there are three phase voltage and currents are measured. So 6 pins are going to be used.

Pins for 3-phase voltage

Ai0 pins (+68, -34) voltage 1

Ail pins (+33, -66) voltage 2

Ai2 Pins(+65,-31) voltage 3

Pins for 3phase current

Ai3 pins (+30,-63) Current 1

Ai4 pins(+28,-61) Current 2

Ai5 pins(+60,-26) Current 3

Pin For speed

Ai6 pin(+25,-58)

Transducer:

Two different types of transducers are used

- Voltage transducer
- Current transducer

Voltage transducer:

Voltage transducer are used to step down high voltage (0-220) volts to (0 to 5) volts. Connections will be in figure A.32 are shown.



*AC instrument power-terminals 3, 4.

Figure A.32: Voltage Transducer

Current transducer:

Current transducer are used to measure high currents in the range of -35/+35 A and produce an output signal proportional to the input. The output range is a -10/+10 volts. The connections are shown in figure A.33



Figure A.33: Current Transducer

Appendix B: Code

GA Code

```
%Objective is to solve nonlinear differential equation (3.10) and (3.14). The
file "Flux" calculated the derivative of the state
%variables, which are defined as fluxes.
globalfo Vs w wbWrCoeffTeCoeff
globalrsXrrp D XmXssXlrpXlsrrp% Machine parameters
choice = menu('Select Induction Machine','3-hp Induction Motor','2250-hp
Induction Motor', '50-hp Induction Motor', '500 hp Induction Motor');
if choice == 1
% Parameters for 3-hp machine
    P = 4; fo = 60;
   Vs = 220/sqrt(3);
rs = 0.435; X1s = 0.754; Xm = 26.13; X1rp = 0.754; rrp = 0.816; J = 0.089;
Pn=3*746.7;
Tf = 0.5;
   Mach = '3-hp machine';
elseif choice == 2
% Parameters for 2250-hp machine
    P = 4; fo = 60;
   Vs = 2300/sqrt(3);
%rs = 0.029; Xls = 0.226; Xm = 13.04; Xlrp = 0.226; rrp = 0.022;
rs=0.0359; Xls= 0.2003; Xm=13.0694; Xlrp=0.2020; rrp=0.0215; J = 63.87; Tf =
3.5;
    Mach = '2250-hp machine';
elseif choice ==3
% Parameters for 50-hp machine
    P = 4; fo = 60;
   Vs = 460/sqrt(3);
rs = 0.087; X1s = 0.302; Xm = 13.08; X1rp = 0.302; rrp = 0.228; J = 1.662; Tf
= 1.8;
   Mach = '50-hp machine';
elseif choice ==4
% Parameters for 500-hp machine
```

```
P = 4; fo = 60;
    Vs = 2300/sqrt(3);
rs = 0.262; Xls = 1.206; Xm = 54.02; Xlrp = 1.206; rrp = 0.187; J = 11.06; Tf
= 1.8;
    Mach = '500-hp machine';
end
we=2*pi*fo;
                        % Work in the synchronous /stationary reference frame
w = we;
Tstep = Tf/2000;
tv = 0:Tstep:Tf;
                        % Time vector for Vqdos
%Define voltages of three phases a, b, c as input to the model
Vas = sqrt(2) *Vs*cos(we*tv);
Vbs = sqrt(2)*Vs*cos(we*tv - (2*pi/3));
Vcs = sqrt(2) *Vs*cos(we*tv + (2*pi/3));
Vabcs = [Vas; Vbs; Vcs];
% Convert Vabcs to the ODO reference frame.
Vqdos = zeros(3, length(tv));
forpt = 1:length(tv);
th = w tv(pt);
    Ks = 2/3*[cos(th) cos(th-(2*pi/3)) cos(th+(2*pi/3)); ...
            sin(th) sin(th-(2*pi/3)) sin(th+(2*pi/3)); .5 .5 .5];
Vqdos(:,pt) = Ks*Vabcs(:,pt);
end
*Solution for the Equations (3.10) and (3.14) will follow directly after
calling ode solver. Results of the state space equation are
%flux variables.
Xss = Xls + Xm;
Xrrp = Xlrp + Xm;
D = Xss * Xrrp - Xm^2;
wb = we;
WrCoeff = P/(2*J);
TeCoeff = .75*P*Xm/(D*wb);
% matrix to find currents
C = 1/D*[Xrrp 0 0 -Xm 0 0;
```

```
0 Xrrp 0 0 -Xm 0;
          0 0 D/Xls 0 0 0;
          -Xm 0 0 Xss 0 0;
          0 -Xm 0 0 Xss 0;
          0 0 0 0 0 D/Xlrp];
FL0 = zeros(8, 1);
[t, FL] = ode15s('Flux', tv, FL0);
Iqdos = C(1:3,:)*FL(:,1:6).'; % Equation (3.14)
Iqdor = C(4:6,:)*FL(:,1:6).'; % Equation (3.14)
wr = FL(:, 7);
theta = FL(:,8); % Theta-r
rpm = wr*(2/P)*60/(2*pi); % Mechanical rpm
Te = TeCoeff*(FL(:,1).*FL(:,5) - FL(:,4).*FL(:,2)); % equation
% Convert Iqdo back to Iabc
Iabcs = zeros(3, length(t));
Iabcr = zeros(3, length(t));
th1=w*t;
forpt = 1:length(t);
th = th1(pt);
Ks inv = [\cos(th) \sin(th) 1; \cos(th-(2*pi/3)) \sin(th-(2*pi/3)) 1;
cos(th+(2*pi/3)) sin(th+(2*pi/3)) 1];
Iabcs(:,pt) = Ks inv * Iqdos(:,pt);
thr = theta(pt);
   b = th-thr;
Kr inv = [\cos(b) \sin(b) 1; \cos(b-(2*pi/3)) \sin(b-(2*pi/3)) 1; ...
   cos(b+(2*pi/3)) sin(b+(2*pi/3)) 1];
Iabcr(:,pt) = Kr inv * Iqdor(:,pt);
end
bits = [10 10 10 10 10]; %Define the number of bits for GA
vlb=[0.1 0.2 20 0.5 0.5];
```

```
vub=[1 1 30 1 1];
options = foptions([1 le-4]); % First:printingsecond:Terminate
options(11)=30; % Number of population - before it was 30
options(13) = 0.033; %Mutation probability
options(12)=0.70; %Crossover probability
options(14)=300; % Max number of gen
% load numeric3-hp.mat % 3-hp machine data
x = [rs, Xls, Xm, Xlrp, rrp];
%load numeric2250-hp.mat %2250-hp machine data
%load numeric50-hp.mat %50-hp machine data
%load numeric7.5-hp.mat %7.5-hp machine data
% we = 2*pi*60; % Omega-e, the angular frequency of the synchronous frame
% % w=we; % w
% Vqdos = zeros(3, length(tv)); % Initialize to 0.
% for pt = 1:length(tv);
% th = w*tv(pt);
% % If omega is not constant, theta = integral of omega
% Ks = 2/3*[\cos(th) \cos(th-(2*pi/3)) \cos(th+(2*pi/3)); \dots
% sin(th) sin(th-(2*pi/3)) sin(th+(2*pi/3)); .5 .5 .5];
% Vqdos(:,pt) = Ks*Vabcs(:,pt);
% end
FitnessFunction = @IMfitness;
nvars = 5;
[x,stats,options,bf,fgen,lgen]=ga(FitnessFunction,nvars,[],[],[],[],vlb,vub)
% x = ga(FitnessFunction, nvars);
% x % best fitness
```

FLUX

```
function Flux = Flux(t,FL);
P = 4; fo = 60;
we = 2*pi*fo; %Omega-e, the angular frequency of the synchronous frame
w=we; % Work in the synchronous reference frame
wb = we;
Vs = 220/sqrt(3);
rs = 0.435; Xls = 0.754; Xm = 26.13; Xlrp = 0.754;
rrp = 0.816; J = 0.089; Pn=3*746.7;
Tf = 0.5;
Xss = Xls + Xm;
Xrrp = Xlrp + Xm;
D = Xss*Xrrp - Xm^2;
WrCoeff = P/(2*J);
TeCoeff = .75*P*Xm/(D*wb);
% FL = a 8-vector = [The 6 flux linkages in (3.10); Omega-r; Theta-r];
\% This function calls ode15s (one of the differential solvers) to
%obtain solution for the fluxes Equation (3.10).
fo = 60;
Vs = 1;
we = 2*pi*fo; %Omega-e, the angular frequency of the synchronous frame
w=we; % Work in the synchronous reference frame
wb = we;
% Conversion of three phase voltages to synchronous QDO frame
Vas = sqrt(2) *Vs*cos(we*t);
Vbs = sqrt(2) *Vs*cos(we*t - (2*pi/3));
Vcs = sqrt(2) *Vs*cos(we*t + (2*pi/3));
Vabcs = [Vas; Vbs; Vcs];
th = w^*t;
Ks = 2/3*[cos(th) cos(th-(2*pi/3)) cos(th+(2*pi/3)); ...
sin(th) sin(th-(2*pi/3)) sin(th+(2*pi/3)); .5 .5 .5];
```

```
Vqdos = Ks*Vabcs;
Vqdor = [0;0;0];
% Equation (3.10) in the state space form:
wr=FL(7);
A = -wb*[rs*Xrrp/D w/wb 0 -rs*Xm/D 0 0;
-w/wbrs*Xrrp/D 0 0 -rs*Xm/D 0;
0 0 rs/Xls 0 0 0;
-rrp*Xm/D 0 0 rrp*Xss/D (w-wr)/wb 0;
0 -rrp*Xm/D 0 - (w-wr)/wbrrp*Xss/D 0;
0 0 0 0 0 0 rrp/Xlrp];
Flux = zeros(8,1);
Flux(1:6) = A*FL(1:6) + wb*[Vqdos; Vqdor];
Te = TeCoeff*(FL(1)*FL(5) - FL(4)*FL(2)); % (equation 3.16)
Flux(7) = WrCoeff*Te; % The equation for d(wr)/dt is (3.17)
Flux(8) = FL(7); % Theta-r = integral(wr)
```

```
function [rs,Xls,Xm,Xlrp,rrp] = IMfitness(x);
%IM fitness simulates the induction machine and as an output gives
%stator currents and rotor speed. It calls the main genetic function
%which further calls additional sub functions (reproduc, mate, xover
%etc.) It calculates the fitness at every step after calling genetic
%main function.
% global fo Vs w wbWrCoeffTeCoeff we Iabcs actual
% global rsXrrp D XmXssXlrpXlsrrpindexxVabcstvwr actual J P Vqdostvwr
% indexx =0
% Declare parameters as variables to be optimized:
rs = x(1);
Xls=x(2);
Xm=x(3);
Xlrp=x(4);
rrp=x(5) ;
% J=x(6);
% fitness = [rs, Xls, Xm, Xlrp, rrp,J];
```

PSO Code

```
% This file descibes the setup parameters for PSO.
% Partical Swarm Optimization for Matlab
% Copyright (C) 2003 AmerAlhinai
2
% This program is free software; you can redistribute it and/or modify
% it under the terms of the GNU General Public License as published by
% the Free Software Foundation; either version 1, or (at your option)
% any later version.
2
% This program is distributed in the hope that it will be useful,
% but WITHOUT ANY WARRANTY; without even the implied warranty of
% MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
% GNU General Public License for more details. A copy of the GNU
% General Public License can be obtained from the
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
% This configuration depends in the problem, the given example is to design
% Ppower System Stabilizer parameters: K1 T11 T12 K2 T21 T22
global K1 T11 T12 K2 T21 T22
응응
c1=2.0; c2=2.01;
n=50; M=5; tmax=50;
bounds = \begin{bmatrix} 0 & 60 \end{bmatrix};
0.1 0.5;
0 0.1;
0 60;
0.1 0.5;
0 0.1]; % Parameters max and min intervels
evalFN = 'objectives';
[Jss,xss,Jsss,t] = psomax(bounds,evalFN,tmax,c1,c2,n,M);
function [Jss,xss,Jsss,t] = psomax(bounds,evalFN,tmax,c1,c2,n,M)
% PSOmax run a Partical Swarm Optimization to maximize the objective
function.
% function [x,endPop,bPop,traceInfo]=ga(bounds,evalFN,tmax,c1,c2,n,M)
8
% Output Arguments:
% Jss - the best solution found during the course of the run.
```

```
% xss - the final population.
% Jsss - a trace of the best population.
% t - number of iteration.
2
% Input Arguments:
% bounds - a matrix of upper and lower bounds on the variables.
% evalFN - the name of the evaluation .m function.
% tmax - maximum number of iteration.
% c1 - weighting factor.
% c2 - weighting factor.
% n - number of popualtion.
% M - maximum number of iteration of unchanged soltion.
% Partical Swarm Optimization for Matlab
% Copyright (C) 2003 AmerAlhinai
2
% This program is free software; you can redistribute it and/or modify
% it under the terms of the GNU General Public License as published by
% the Free Software Foundation; either version 1, or (at your option)
% any later version.
8
% This program is distributed in the hope that it will be useful,
% but WITHOUT ANY WARRANTY; without even the implied warranty of
% MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
% GNU General Public License for more details. A copy of the GNU
% General Public License can be obtained from the
x int= bounds;
t=1; % 1st iteration.
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
phi=c1+c2;
K=2/(abs(2-phi-sqrt(phi^2-4*phi)));
fori=1:length(x int) % initial population position
x(i,:)=x int(i,1)+(x int(i,2)-x int(i,1))*rand(1,n);
end
fori=1:length(x int) % initial velocities
Vmax(i) = max(x(i,:)); Vmin(i) = min(x(i,:));
vk(i) = (Vmax(i) - Vmin(i)) / n;
```

```
V(i,:) = -vk(i) + (vk(i) + vk(i)) * rand(1,n);
end
vmax=vk;vmin=-vk;
size V=size(V);
xs=x; %vs=V;
fori=1:size V(2) % evaluate the initial population
xc=x(:,i);
elstr=['jc='evalFN'(xc);'];
eval(e1str);
J(i)=jc;
end
Js=J; % initial best individual
Jss=max(Js);xss=x(:,find(Jss==Js));
Jsss(t)=Jss; % initial best global
m=0;
while t<tmax% check maximum # of iteration</pre>
if m>M \ check the number of iterations since the last change
break% of the best solution "m" is greater than "M"
end
t=t+1
m
r1=rand(1);r2=rand(1);
fori=1:size V(1) % update the velocity
for k=1:size V(2)
v cal=K*(V(i,k)+c1*r1*(xs(i,k)-x(i,k))+c2*r2*(xss(i)-x(i,k)));
if (v cal>=vmin(i) &v cal<=vmax(i))</pre>
      V(i,k)=v cal;
else
end
end
end
fori=1:size_V(1) % update position
for k=1:size V(2)
x \text{ cal=V}(i,k)+x(i,k);
if (x cal>=x int(i,1) &x cal<=x int(i,2))</pre>
      x(i,k)=x cal;
else
```

```
end
end
end
fori=1:size_V(2) % evaluation
xc=x(:,i);
elstr=['jc='evalFN'(xc);'];eval(elstr);
J(i)=jc;
end
fori=1:size V(2) % update best individual
if J(i)>Js(i)
Js(i) = J(i);
xs(:,i)=x(:,i);
else
end
end
Jmax=max(Js);
ifJmax>Jss% update best global
    m=0;
Jss=Jmax
xss=xs(:,find(Jss==Js));xss=xss(:,1);
else
   m=m+1;
fprintf('%2.0f ',m)
end
Jsss(t)=Jss;
end
return
function [val] = objectives(sol)
% This function compute the Partical Swarm Optimization objective function.
% function [val] = objectives(sol)
00
% Output Arguments:
% val - PSO performance index value
8
% Input Arguments:
% sol - PSO candidate solution
% Partical Swarm Optimization for Matlab
```

```
% Copyright (C) 2003 AmerAlhinai
9
% This program is free software; you can redistribute it and/or modify
% it under the terms of the GNU General Public License as published by
% the Free Software Foundation; either version 1, or (at your option)
% any later version.
0
% This program is distributed in the hope that it will be useful,
% but WITHOUT ANY WARRANTY; without even the implied warranty of
% MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
% GNU General Public License for more details. A copy of the GNU
% General Public License can be obtained from the
% This configuration depends in the problem, the given example is to design
% Ppower System Stabilizer parameters:
% K1 Tw1 T11 T12 T13 T14 K2 Tw2 T21 T22 T23 T24 K3 Tw3 T31 T32 T33 T34
global K1 Tw1 T11 T12 T13 T14 K2 Tw2 T21 T22 T23 T24 K3 Tw3 T31 T32 T33 T34
K1=sol(1); Tw1=sol(2);
T11=sol(3);T12=sol(4);
T13=sol(5); T14=sol(6);
K2=sol(7); Tw2=sol(8);
T21=sol(9);T22=sol(10);
T23=sol(11);T24=sol(12);
K3=sol(13);Tw3=sol(14);
T31=sol(15);T32=sol(16);
T33=sol(17);T34=sol(18);
୧୧୧୧୧୧୧୧୧୧୧୧୧୧୧୧
sim('m IEEE37 2MT') % Nonlinear Simulations
if tout <7 % If the guided constrains interepted the simulation ignor this
solution
val=0
else% Good candidate solution
j=min(sum(J)); % J is the simulation performance index computed within PAT
val=1e6/j
end
```

```
61
```

References

[1] ZbigniewMichalewicz. Genetic Algorithms + Data Structures = Evolution Programs. Springer -

Verglar 3rd edition, 1996.

[2] David E. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. Addison – Wesley Publishing Company, Inc. 1989.

[3] Steven R. Shaw. Numerical Methods for Identification of Induction Motor Parameters. Master's thesis, MIT, January 1997.

[4] Dennis J. Evangelista. Parameter Estimation of Induction Motors using PWM Inverters. Master's thesis, MIT, May 1999.

[5] A. E. Fitzgerald, C. Kingsley, S. D. Umans. Electric Machinery. McGraw-Hill, fourth edition, 1983.

[6] Paul C. Krause, Oleg Wasynczuk, Scott D. Sudhoff. Analysis of Electric Machinery. IEEE Press, 1995.

[7] Lawrence Davis. Handbook of Genetic Algorithms. International Thomson Computer Press, 1996.

[8] P. C. Krause, C. H. Thomas. Simulation of Symmetrical Induction Machinery. IEEE Transactions on Power Apparatus and Systems, November 1965.

[9] Richard R. Bishop, Gill G. Richards. Identifying Induction Machine Parameters Using a Genetic Optimization Algorithm. IEEE Proceedings – 1990 Southeastcon.

[10] P. Ju, E. Handschin, Z. N. Wei. Application of genetic algorithm to nonlinear dynamic modeling. 1995 Elsevier Science B. V.

[11] Seung-Ill Moon, Ali Keyhani, Srinivas Pillutla. Nonlinear Neural – Network Modeling of an Induction Machine. IEEE Transactions on Control Systems Technology, March 1999.

[12] H.B. Karayaka, A. Keyhani. Induction Machine Parameter Tracking from Test Data via PWM Inverters. IEEE Industry Applications Society, October 1997.

[13] JA de Kock, FS van der Merwe, HJ Vermeulen. Induction Motor Parameter Estimation through an Output Error Technique. IEEE Transactions on Energy Conversion, March 1994.

[14] Simon Hart, AbolfazlRanjbar, B.E. Mulhall. Practical Combined Parameter Identification and State Estimation of Machines. Proceedings volume from the 8th IFAC Symposium, UK, September 2000.

[15] Robin Biesbroek. GA Tutorial. http://www.estec.esa.nl/outreach/gatutor/Default.htm.

[16] MATLAB, High Performance Computation and Visualization Software, The Mathworks Inc., 2001.

[17] Genetic Algorithms. Implementation based on GAs from David E. Goldberg. Andrew F Potvin. The MathWorks, Inc. January 1994.

[18] Basics on GAs. http://www.d.umn.edu/ece/lis/GA/ga_1_basics.html.

[19] LabVIEW Data Acquisition Basics Manual. National Instruments. January 1998.

[20] LabVIEW User Manual. National Instruments. January 1998.

[21] DAQ PCI E Series User Manual. National Instruments. July 1997.

[22] KouroshSeghisigarchi. DAQ User Manual. WVU 1999.

[23] IEEE Standard Test Procedure for Polyphase Induction Motors and Generators. Electric Machines Committee of the IEEE Power Engineering Society, USA. IEEE Std 112-1996. INSPEC Accession Number: 5621352.

[24] Yu-hua-Wang, Birdwell-JD. Dynamic identification of the model parameters for an induction motor. Conference Proceedings of IEEE SOUTHEASTCON, 1982.

[25] National Instrument and LAB View Software. http://www.ni.com/.

[26] P. C. Krause, O. Wasynczuk, S. D. Sudhoff, Analysis of Electric Machinery, Institute of Electrical & Electronics Engineer; , January 1995.

[27] Y. Koubaa, "Recursive identification of induction motor parameters," Journal of Simulation Modeling Practice and Theory, Vol. 12, Issue 5, pp. 363-381, August 2004.

[28] J. Stephan, M. Bodson, and J. Chiasson, "Real Time Estimation of Induction Motor Parameters," IEEE Trans. on Industrial Applications, Vol. 30, No. 3, pp. 746-759, May-June 1994.

[29] K. Wang, J. Chiasson, M. Bodson, L. M. Tolbert, "A Nonlinear Least-Squares Approach for Identification of the Induction Motor Parameters," IEEE Trans. on Automatic Control, Vol. 50, No. 10, pp. 1622-1628, October 2005.

[30] H. A. Toliyat, E. Levi, and M. Raina, "A Review of RFO Induction Motor Parameter Estimation Techniques," IEEE Trans. on Energy conversion, Vol.18, No. 3, pp. 271-283, June 2003.

[31] R. R. Bishop, G. G. Richards, "Identifying Induction Machine Parameters Using A Genetic Optimization Algorithm," Proceedings. Southeastcon IEEE, Vol.2, pp. 476-479, New Orleans, LA, USA, April 1990.

[32] F. Alonge, F. D' lppolito, G. Ferrante, F. M. Raimondi, "ParameterIdentification of Induction Motor Model Using Genetic Algorithms," IEE Proc. Control Theory Applications, Vol. 145, No. 6, November 1998.

[33] K. S. Huange, Q. H. Wu, D. R. Turner, "Effective Identification of Induction Motor Parameters Based on Fewer Measurements," IEEE Trans. on Energy Conversion, Vol. 17, No. 1, March 2002.

[34] E. Bajrektarevi'c, M. A. Choudhry, "Parameter Identification of Induction Motor using a Genetic Algorithm and Lab View Software," Proceedings of 34th North American Power Symposium, October 14-15, 2002.

[35] R. K. Ursem, P. Vadstrup, "Parameter Identification of Induction Motors Using Differential Evolution," The 2003 Congress on Evolutionary Computation CEC '03, Vol. 2, pp. 790-796, 8-12 December 2003.

[36] IEEE Standard Test Procedure for Polyphase Induction Motors and Generators. *Electric Machines Committee of the IEEE Power Engineering Society, USA*. IEEE Std112-1996. INSPEC Accession Number: 5621352.

[37] D.P.Rini,S,M.shamsuddin,S.S.Yuhaniz,"Particle Swarm Optimization: Technique, System and Challenges" Volume14-No.1,January2011

[38].Qinghai Bai"Analysis of Particle Swarm Optimization Algorithm"Vol.3 No.1 February 2010

[39] J. Kennedy, R. C. Eberhart, Y. Y. Shi, Swarm Intelligence, The Morgan Kaufmann Series in Artificial Intelligence, March 2001.

[40] S. R. Shaw, S. B. Leeb "Identification of Induction Motor Parameters from Transient Stator Current Measurements," IEEE Trans. on Industrial Electronics, Vol. 46, No. 1, February 1999.

[41]NadeekaYapa "Genetic Algorithm in Induction Motor Efficiency Detemination" May 2004

[42]Rania Hassan,Babakcohanim,Olivier De Weck "A comparison of Particle Swarm Optimization And and the Genetic Algorithm"

[43] http://www.talkorigins.org/faqs/genalg/genalg.html#strengths

[45]Ali Karimi, M.A,Choudhry , Ali Feliachi "PSO-based Evolution Optimization for Parameter Idenetification of an induction Motor" IEEE,2007

[46] Edina Bajrektarevic "Parameter identification of induction motor using a Genetic Algorithm" Masters thesis ,2002

[47] Phaderm Nangsue, Pragasen Pillay, Susan E. Conry. Evolutionary Algorithms for Induction Motor Parameter Determination.

[48 K.S. Huang, W. Kent, Q.H. Wu, D. R. Turner, Parameter Identification of an Induction Machine Using a Genetic Algorithms. Proceedings of the 1999 IEEE, August 1999.]