MASTER OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING

Development of a smart controller for a Switched Reluctance Motor drives

by

Md. Rezaul Hasan

Department of Electrical and Electronic Engineering
Islamic University of Technology (IUT)
Gazipur-1704, Bangladesh.

July, 2012.

CERTIFICATE OF APPROVAL

The thesis titled "Development of a smart controller for a switched reluctance motor drives" submitted by Md. Rezaul Hasan bearing Student No. 092613 of Academic Year 2009-2010 has been found as satisfactory and accepted as partial fulfillment of the requirement for the degree of Master of Science in Electrical and Electronic Engineering on 26 July 2012.

BOARD OF EXAMINERS

---

1. **Dr. Md. Ashraful Hoque** — Chairman (Supervisor)
   Professor
   Department of Electrical and Electronic Engineering
   Islamic University of Technology (IUT)
   Gazipur, Bangladesh.

---

2. **Dr. Md. Shahid Ullah** — Member (Ex-Officio)
   Professor and Head
   Department of Electrical and Electronic Engineering
   Islamic University of Technology (IUT)
   Gazipur, Bangladesh.

---

3. **Dr. Kazi Khairul Islam** — Member
   Professor
   Department of Electrical and Electronic Engineering
   Islamic University of Technology (IUT)
   Gazipur, Bangladesh.

---

4. **Dr. Muhammad Fayyaz Khan** — Member (External)
   Professor
   Department of Electrical and Electronic Engineering
   United International University (UIU)
   Dhaka, Bangladesh.

DECLARATION OF CANDIDATE

It is hereby declared that this Thesis or any part of it has not been submitted elsewhere for the award of any Degree or Diploma.

(Signature)
Dr. Md. Ashraful Hoque
Supervisor and Professor
Department of Electrical and Electronic Engineering
Islamic University of Technology (IUT)
Gazipur-1704, Bangladesh.

(Signature)
Md. Rezaul Hasan
Student No. 092613
Academic Year: 2009-2010

DEDICATION

This thesis is dedicated to my beloved parents and all my well wishers helping me to accomplish this work.

TABLE OF CONTENTS

Chapter 6
Theory of basic Genetic Algorithm (GA)

Chapter 7
Performance analysis of genetically tuned PI controller

Chapter 8
Conclusion

LIST OF TABLES

LIST OF FIGURES

Chapter 6

Chapter 7

ACKNOWLEDGEMENTS

Every honor and every victory on earth is due to the Great Almighty, descended from Him and must be ascribed to Him. He has given me the capability to do all this work. This thesis is a result of research of one and a half year and this is by far the most significant scientific accomplishment in my life. It would be impossible without support and appreciation of those who mattered most. I would like to express my heartiest gratitude to my supervisor, Prof. Dr. Md. Ashraful Hoque for his great supervision, inspiration and unbounded support for doing this thesis. I would also like to pay my heartiest gratitude to Asstt. Prof. Mr. Ashik Ahmed, for his great support in understanding the basics of linear control system. I am also thankful to the Head of the Department Prof. Dr. Md. Shahid Ullah for his guideline and support for my thesis. Last but not the least I am thankful to my parents and my respected elder brothers, colleagues, friends and relatives who have always supported me in my work.

ABSTRACT

Nowadays, switched reluctance motors (SRMs) attract more and more attention. Switched reluctance machines have emerged as an important technology in industrial automation. They represent a real alternative to conventional variable speed drives in many applications. It not only features a salient pole stator with concentrated coils, which allows earlier winding and shorter end turns than other types of motors, but also features a salient pole rotor, which has no conductors or magnets and is thus the simplest of all electric machine rotors. Simplicity makes the SRM inexpensive and reliable, and together with its high speed capacity and high torque to inertia ratio, makes it a superior choice in different applications.

However, the control of the SRM is not an easy task. The motor flux linkage appears to be a nonlinear function of stator currents as well as rotor position, as does the generated electric torque. Apart from the complexity of the model, the SRM should be operated in a continuous phase-to-phase switching mode for proper motor control. This makes the control of SRM a tough challenging. This thesis attempts to first create a MATLAB model of multiphase SRM using the equations governing the dynamic behavior of linear inductance profile SRM. Based on this model, an example case study of single phase SRM operation has been. Small signal analysis of linearized single phase SRM was simulated. Performance analysis of the speed control loop, current control loop and overall SRM drive using PI controller was simulated in MATLAB environment and later all those were simulated again after PI controller was tuned by genetic algorithm. Performance improvement of genetically tuned PI controller is proved in this thesis.

CHAPTER 1

INTRODUCTION

1.1 Background

The switched reluctance motor (SRM) represents one of the oldest electric motor designs around. A variation on the conventional reluctance machine has been developed and is known as the "switched reluctance" (SR) machine. This development is partly due to recent demand for variable speed drives and partly as a result of development of power electronic drives. The name "switched reluctance", describes the two features of the machine configuration: (a), switched, the machine must be operated in a continuous switching mode, which is the main reason for the machine development occurred, only after good power semiconductors became available; (b), reluctance, it is the true reluctance machine in the sense that both rotor and stator have variable reluctance magnetic circuits or more properly, it is a doubly salient machine. The switched reluctance motor is basically a stepper motor with fewer poles and has had many applications as both rotary and linear steppers. The idea of using the SR configuration in a continuous mode (in contrast to a stepper mode) with power semiconductor control is due primarily to Nasar [1-2], at that time, only thyristor power semi-conductors were available for the relatively high-current, high-voltage type of control needed for SR machines.

The reluctance motor operates on the principle that a magnetically salient rotor is free to move to a position of minimum reluctance to the flow of flux in a magnetic circuit. Improved magnetic materials and advances in machine design have brought the switched reluctance motor into the variable speed drive market. The simple brushless construction of the motor makes it cheap to build and very reliable in operation. The unipolar current requirements of the phase windings

results in a simple and very reliable power converter circuit. The researchers are now focusing on switched reluctance motors and drives with only one or two phase windings so that applications for the technology are being created in low cost, high volume markets such as domestic appliances, heating ventilation and air conditioning and automotive auxiliaries. In recent years, power transistors, GTOs, IGBTs, and power MOSFETs have been developed in the power ranges required for SRM control [3]. SRM's eliminate permanent magnets (PMs), brushes and commutators. The stator consists of steel laminations forming salient poles. A series of coil windings, independently connected in phase pairs, envelops the stator poles. With no rotor winding, the rotor is basically a piece of steel (and laminations) shaped to form salient poles. It is the only motor type with salient poles in both the rotor and stator (double salient). As a result, and also because of its inherent simplicity, the SR machine promises a reliable and low-cost variable-speed drive and will undoubtedly take the place of many drives using the cage induction and DC commutator machines in the near future. The switched reluctance motor is a new entrant in domestic appliance applications. Many electrical machine researchers are investigating the dynamic behaviour of switched reluctance motor (SRM) by monitoring the dynamic response (torque and speed), monitoring and minimizing the torque ripple, building different types of controllers to reduce the cost, to increase the general performance of SRM like high reliability and high practicability, to build a better controller for SRM [6-14]. The switched reluctance motor's (SRM) principle of operation has been known for more than a century, under general name of the doubly salient variable reluctance motor. However, an intensive research on SRM began about thirty years ago, mainly due to the progress in power electronics and microprocessors. Its principal advantages are simple and robust construction, possibility to work

at very high rotation speeds, high mechanical torque at low speeds, and simple power electronics driver [15-18].

SR motors motors offer numerous benefits, such as:

- Performance: much greater torque output and with the same (or slightly higher) efficiencies than "premium efficiency" induction motors. Efficiency is flat over a wider speed range;
- Small unit size: makes efficient use of materials and low inertia;
- Low cost: low manufacturing cost, low material cost and low maintenance cost. It does not use magnets;
- High speed and acceleration capability: 100,000RPM (Rotation Per Minute), with the proper drive; [19-21]
- Cooling: most of the heat is generated in stationary stator which is relatively easy to cool;
- Rugged construction suitable for harsh environments such as high temperature and vibration.

SRM controllers add to the benefits. Since they do not need bipolar (reversed) currents, the number of power-switching devices can be reduced by 50%, compared to bridge-type inverters of adjustable-speed drives. An SRM drive has inherent reliability and fault tolerance, it can run in a "limp-home" mode with diminished performance with one failed transistor in a phase, unlike standard motor drives. As control techniques developed, applications of SRMs include:

(a). general purpose industrial drives;

(b). application specific drives: compressors, fans, pumps, centrifuges;

(c). domestic drives: food processors, washing machines, vacuum cleaners;

(d). electric vehicle application;

(e). aircraft applications;

(f). servo-drives.

The switched reluctance motor, which was originally conceived in the early 1800's [3], recently has gained considerable attention. It has the advantages of being inexpensive and rugged. Its simple construction makes it easy to manufacture but rugged enough to be worthy of consideration for powering traction applications such as automobiles [4]. But, it also has its drawbacks. The switched reluctance motor is inherently subject to torque ripple and acoustic noise [5]. This makes a more complex means of control necessary. Until recently, it was not considered a viable candidate for traction applications, but with improved methods of control it may be possible to design a method which would allow the use of the reluctance motor where smoother torque is required. Research into this application requires computer simulation and so a computer model is required. This work includes one such model for the multiphase switched reluctance motor. The model was created in MATLAB and the code is included as an appendix. Then a genetically tuned PI controller is simulated and the improvement of the performance is shown with respect to a normal open loop operation and a universal PI controller.

1.2 Related Work

Many researchers worked on switched reluctance motor modeling and control. Lawrenson *et al.* [22] has laid general foundation for the basic modes of operation, analysis, design considerations and experimentation from a family of prototype motors. Fulton *et al.* [23] has presented a timely

review of the different design methods, which have been adopted for the SRM up to 1988 and broadly classified the design methods into 1) linear method; 2) nonlinear method; and 3) finite element method. Ray and Davis *et al.* [24] suggested a superior approach which depended on linearizing the inductance that allowed the voltage to be switched at any point in the cycle and enabled control strategies to be examined with sufficient accuracy inclusive of component ratings. An evaluation of the capabilities of the switched reluctance motor drive [25], particularly in small integral-horsepower sizes, has been presented and was compared with those of typical induction motor drives [25]. Higuchi *et al.* [26] has designed and developed a single phase 2/2 switched reluctance motor as a cost-effective alternative to multiphase SRM in fan applications and overcame the starting problem in single phase SRM by successful application of magnetic saturation effect. Chari *et al.* [27] has well formulated the finite element method as a suitable technique for electrical design, performance evaluation and device optimization in low frequency applications. Arumugam *et al.* [28] for the first time used a finite element model for 2-D magnetic field analysis of SRM to predict the steady state motor performance accurately. The intelligence controller also brings about improved system performance such as torque ripple reduction [29], [30] and mitigation of acoustic noise. Bolognani *et al.* [31] has designed a fuzzy logic control for a SRM drive in terms of state evaluation control rules derived from a rough formulation of sliding mode- control of the drive. Several authors [32]–[35] and [36] have proposed schemes for torque ripple reduction based on fuzzy logic control in situations where it is difficult to obtain accurate mathematical model or when the model is severely nonlinear. The major research issues in the speed control scheme of a SRM drive are the fast tracking capability, less steady state error and robustness to load disturbance. The proposed hybrid controller by Paramasivam *et al.* [37] has reduced the steady state error as compared with PI-type fuzzy logic

control (FLC), while keeping the merits of PI-type FLC. Reay *et al.* [38] proposed neural network based SRM drive.

1.3 Motivation

As an intelligent control technology, the genetic algorithm (GA) can give robust adaptive response of a drive with nonlinearity, parameter variation and load disturbance effect. From the above literature review, we saw that intelligent controllers like fuzzy logic based approach and Neural network based approach has already been widely applied in SRM drive. No research work has so far used genetically tuned PI controller in MATLAB modeling of Switched Reluctance motor. Selecting the proper value of Proportional gain constant, K_p and Integration gain constant, K_i is a difficult job for PI controller and most often researchers go for trial and error method. But in this regard, genetic algorithm can tune the best possible value for K_p and K_i. Good result obtained from the simulation of this research can instigate further experimental work using the model of SRM and genetically tuned PI controller in this research work.

1.4 Research Objectives

The objectives of the work can be listed as follows:

- ✓ MATLAB modeling of multiphase Switched Reluctance Motor (SRM).
- ✓ Small signal analysis of a specific single phase linearized switched reluctance motor (SRM).
- ✓ Performance analysis of speed control loop, current control loop and overall SRM drive with PI controller only.
- ✓ Performance analysis of speed control loop, current control loop and overall SRM drive with genetically tuned PI controller.

- ✓ Drawing the conclusion based on the comparison of the results.

1.5 Outline of the Thesis

- ✓ **Chapter 1** represents the background of the present work, motivation and objectives and related work with this project.
- ✓ **Chapter 2** elaborates the working principle of Switched Reluctance Motor (SRM), It's dynamic behavior and governing equation.
- ✓ **Chapter 3** gives the MATLAB modeling of multiphase SRM along with the simulated characteristics curve of the SRM.
- ✓ **Chapter 4** provides the detail of small signal analysis of a specified linearized SRM, calculation of all the characteristic parameters and plot of the curves from the MATLAB simulation.
- ✓ **Chapter 5** describes the performance of current control loop and speed control loop and the overall SRM drive block including both current loop and speed loop inside of a single phase SRM using PI controller.
- ✓ **Chapter 6** presents the theory of genetic algorithm.
- ✓ **Chapter 7** describes the performance of current loop, speed loop and overall block using a genetically tuned PI controller and comparison with the performance of only PI controller.
- ✓ **Chapter 8** draws the summary and conclusion based on the compared results.

CHAPTER 2

MATHEMATICAL MODELING OF SWITCHED RELUCTANCE MOTOR

2.1 Elementary operation of the Switched Reluctance Motor

Even though this machine is a type of synchronous machine, it has certain novel features. It has wound field coils of a dc motor for its stator windings and has no coils or magnets on its rotor. Both the stator and rotor have salient poles, hence the machine is referred to as a doubly salient machine. The rotor is aligned whenever diametrically opposite stator poles are excited. In a magnetic circuit, the rotating member prefers to come to the minimum reluctance position at the instance of excitation. While two rotor poles are aligned to the two stator poles, another set of rotor poles is out of alignment with respect to a different set of stator poles. Then, this set of stator poles is excited to bring the rotor poles into alignment. Likewise, by sequentially switching the currents into the stator windings, the rotor is rotated. The movement of the rotor, hence the production of torque and power, involves switching of currents into stator windings when there is a variation of reluctance; therefore, this variable speed motor drive is referred to as a switched reluctance motor drive. Consider that the rotor poles r_1 and r_1' and stator poles c and c′ are aligned. Apply a current to phase a with the current direction as shown in Figure 2.1a. A flux is established through stator poles a and a′ and rotor poles r_2 and r_2' and which tends to pull the rotor poles r_2 and r_2' and toward the stator poles a and a′ respectively. When they are aligned, the stator current of phase a is turned off and the corresponding situation is shown in Figure 2.1b. Now the stator winding b is excited, pulling r_1 and r_1' toward b and b′ , respectively, in a

clockwise direction. Likewise, energization of the c phase winding results in the alignment of r_2 and r_2' with c and c′, respectively.

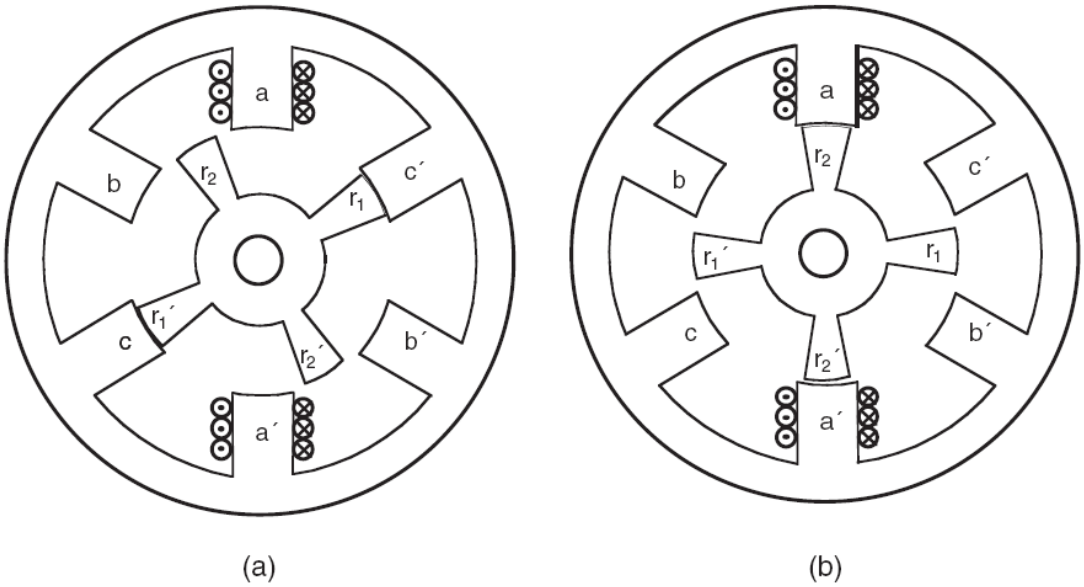

Fig. 2.1 Operation of an SRM. (a) phase c aligned. (b) phase a aligned.[39]

Hence, it takes three phase energizations in sequence to move the rotor by 90°, and one revolution of rotor movement is affected by switching currents in each phase as many times as there are number of rotor poles. The switching of currents in the sequence acb results in the reversal of rotor rotation is seen with the aid of Figures 2.1a and b. The SRM must obey the laws of physics. The torque in a reluctance motor is developed by virtue of the change in the reluctance with respect to the rotor position. Based on this principle, a reluctance motor is different from other types of electric machines such as the DC machine, synchronous machine and induction machine. The theory of conventional reluctance machines evolved from synchronous machine theory developed in the early 20th century, based on the well-known Park Equations. The basic torque or force production in reluctance machines results from the variation

of the stored magnetic energy as a function of the rotor position. This relationship also applies to most electromagnetic relays, holding magnets, solenoid actuators, and other devices where force is produced between two magnetic surfaces, including all machines with saliency.

2.2 Mathematical modeling of SRM

To derive the basic torque equation of the SRM, consider an elementary reluctance machine as shown in figure 2.2. The machine is single phase exited; and the excited winding is

$$\phi(\theta) = L(\theta)\, I \tag{2.1}$$

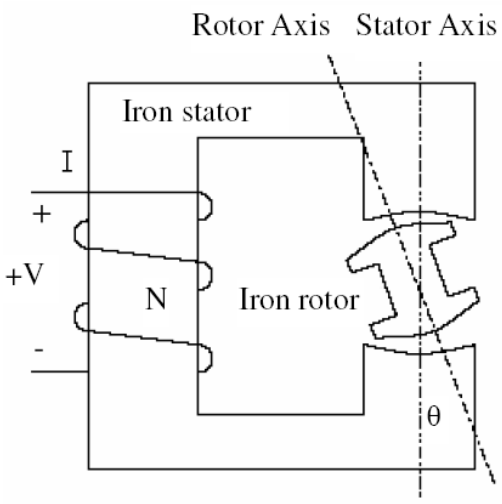

Fig. 2.2 A single phase SRM

wound on the stator and the rotor is free to rotate [41-42]. The flux linkage is where I is the independent input variable, i.e. the current is flowing through the stator and L is the inductance.

A mathematical model of an SRM can be developed, based on the electrical diagram of the motor, incorporating phase resistance and phase inductance. The diagram for one phase is illustrated in figure 2.3. The voltage applied to a phase of the SRM can be described as a sum of voltage drops in the phase resistance and induced voltages on the phase inductance:

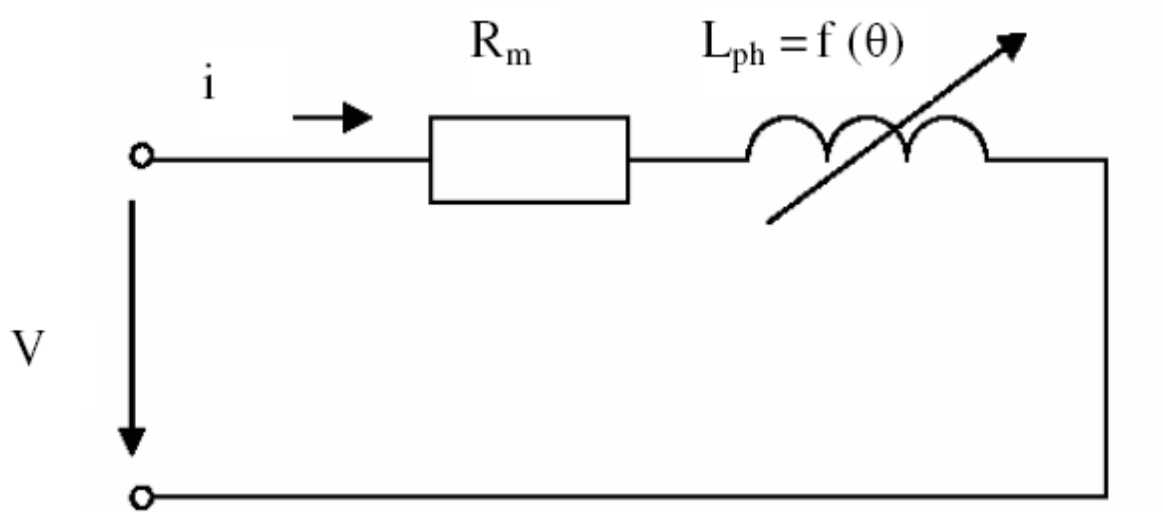

Fig. 2.3 Equivalent Circuit Diagram of Single Phase SRM

Although SR motor operation appears simple, an accurate analysis of the motor's behaviour requires a formal, and relatively complex, mathematical approach. The instantaneous voltage across the terminals of a single phase of an SR motor winding is related to the flux linked in the winding as illustrated in equation 2.2,

$$V = i R_m + \frac{d\phi}{dt} \tag{2.2}$$

where, V is the terminal voltage, i is the phase current, Rm is the motor resistance, and φ is the flux linked by the winding. Because of the doubly salient construction of the SR motor (both the rotor and the stator have salient poles) and because of magnetic saturation effects, in general, the

flux linked in an SRM phase varies as a function of rotor position, φ, and the motor current. Thus, equation (2.2) can be expanded as:

$$V = i R_m + \frac{\partial \phi}{\partial i}\frac{di}{dt} + \frac{\partial \phi}{\partial \theta}\frac{d\theta}{dt} \tag{2.3}$$

where,— is defined as L(θ,i), the instantaneous inductance, — is the instantaneous back EMF. Equation (2.3) governs the transfer of electrical energy to the magnetic field. In this section, the equations which describe the conversion of the field's energy into mechanical energy are developed. Multiplying each side of equation (2.2) by the electrical current i, gives an expression for the instantaneous power in an SRM:

$$V i = i^2 R_m + i\frac{d\phi}{dt} \tag{2.4}$$

The left-hand side of equation (2.4) represents the instantaneous electrical power delivered to the SRM. The first term in the right-hand side (RHS) of equation (2.4) represents the ohmic losses in the SRM winding. If power is to be conserved, then the second term in the RHS of equation (2.4) must represent the sum of the mechanical power output of the SRM and any power stored in the magnetic field. Thus,

$$i\frac{d\phi}{dt} = \frac{dW_m}{dt} + \frac{dW_f}{dt} \tag{2.5}$$

where——, the instantaneous mechanical power, and —— is the instantaneous power. Because power, by its own definition, is the time rate of change of energy, Wm is the mechanical energy

and Wf is the magnetic field energy. It is well known that mechanical power can be written as the product of torque and speed,

$$\frac{dW_m}{dt} = T\omega = T\frac{d\theta}{dt} \tag{2.6}$$

where T is the torque, and w=— is the rotational velocity of the shaft. Substitution of equation (2.6) into equation (2.5) gives,

$$i\frac{d\phi}{dt} = T\frac{d\theta}{dt} + \frac{dW_f}{dt} \tag{2.7}$$

Solving equation (2.7) for torque yields the following equation,

$$T(\theta,\phi) = i(\theta,\phi)\frac{d\phi}{d\theta} - \frac{dW_f(\theta,\phi)}{d\theta} \tag{2.8}$$

For constant flux, equation (2.8) simplifies to,

$$T = -\frac{\partial W_f}{\partial \theta} \tag{2.9}$$

Since it is often desirable to express torque in terms of current rather than flux, it is common to express torque in terms of co-energy Wc, instead of energy. To introduce the concept of co-energy, first consider a graphical interpretation of field energy. For constant shaft angle, —=0, integration of equation (2.7) shows that the magnetic field energy can be shown by a shaded area in figure 2.4 and equation (2.10) [42-43].

$$W_f = \int_0^{\phi} i(\theta,\phi)\, d\phi \tag{2.10}$$

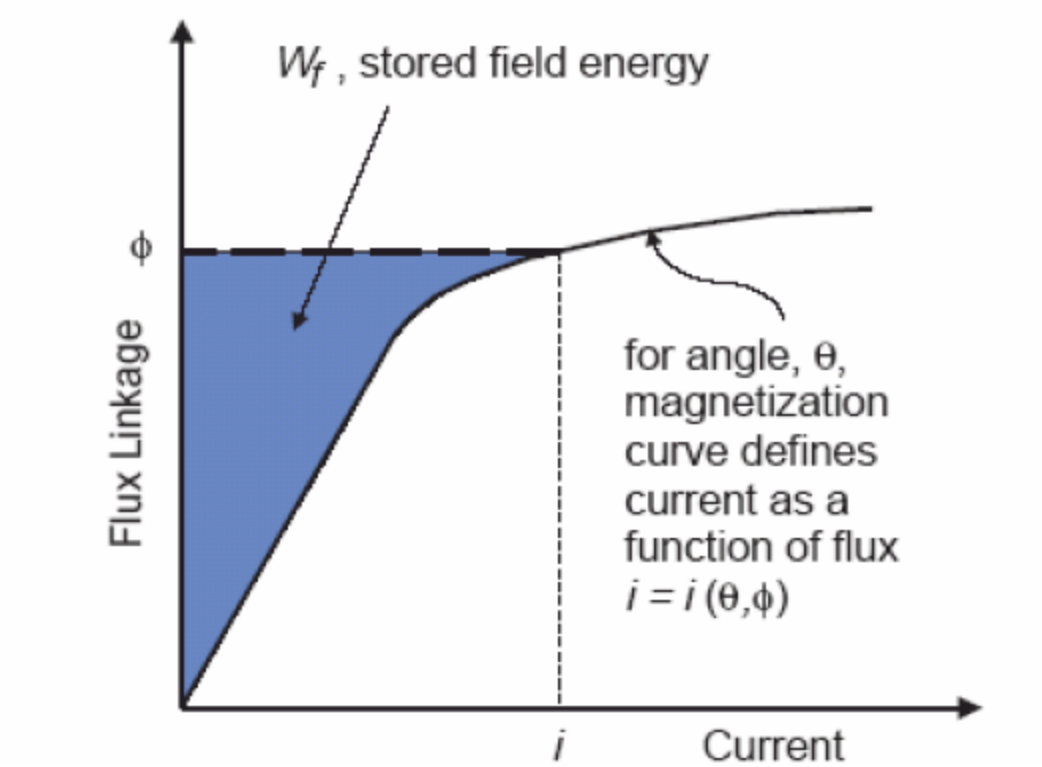

Fig. 2.4 Graphical Interpretation of Magnetic Field Energy [41].

For the fixed angle, θ, let the magnetization curve define flux as a function of current, instead of current defined as a function of flux. The shaded area below the curve is defined as the magnetic field co-energy, and shown in figure 2.5 and equation (2.11) [41-43].

$$W_c = \int_0^i \phi(\theta, i)\, di \tag{2.11}$$

From figures 2.4 and 2.5, we see that the area defining the field energy and co-energy can be described by the relation,

$$W_c + W_f = i\phi \tag{2.12}$$

Differentiating both sides of equation (2.12) yields

$$dW_c + dW_f = \phi\, di + i d\phi \tag{2.13}$$

Solving for the differential field energy in equation (2.13) and substituting back into equation (2.8) gives,

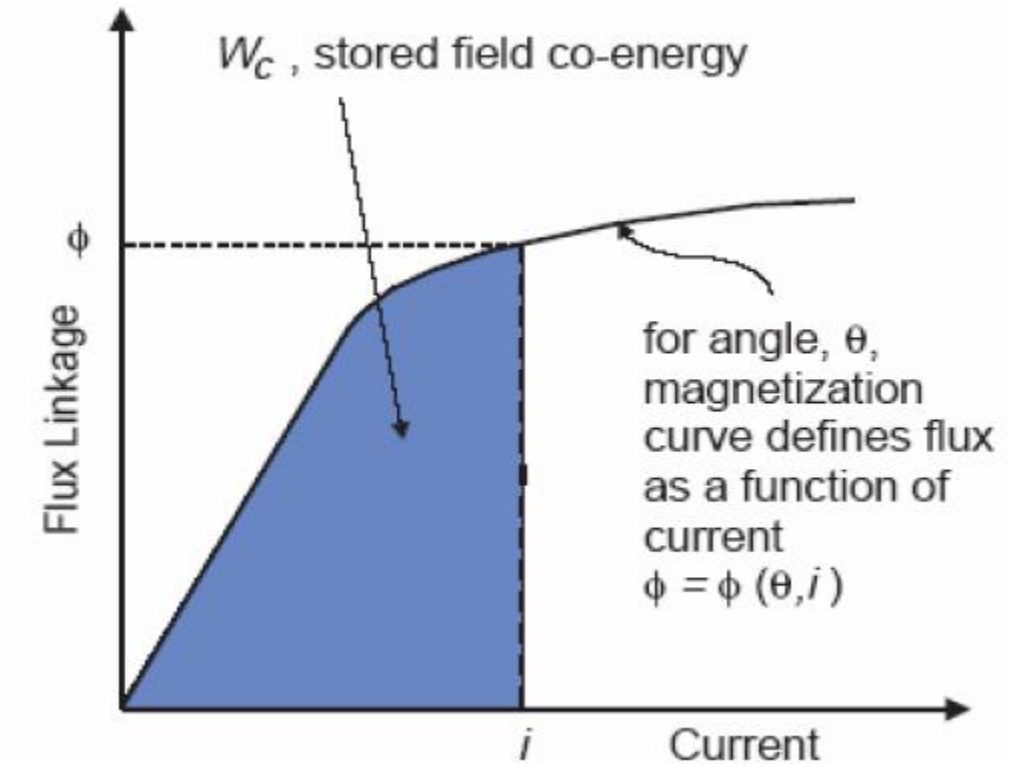

Fig. 2.5 Graphical Interpretation of Magnetic Field Co-energy [41].

$$T = \frac{i d\phi - (\phi di + i d\phi - dW_c(\theta, i))}{d\theta} \tag{2.14}$$

For simplification, the general torque equation, equation (2.14), is usually simplified for constant current. The differential co-energy can be written in terms of its partial derivatives as,

$$dW_c(\theta, i) = \frac{\partial W_c}{\partial \theta} d\theta + \frac{\partial W_c}{\partial i} di \tag{2.15}$$

From equation (2.14) and equation (2.15), it is fairly easy to show that under constant current,

$$T = \frac{\partial W_c}{\partial \theta}, \ \text{i constant} \tag{2.16}$$

Often, SRM analysis proceeds with the assumption that, magnetically, the motor remains unsaturated during operation. When magnetic saturation is neglected, the relationship from flux to current is given by,

$$\phi = L(\theta) \times i \tag{2.17}$$

And the motor inductance varies only as a function of rotor angle. Substituting equation (2.17) into equation (2.11) and evaluating the integral yields,

$$W_c = \frac{i^2}{2} L(\theta) \tag{2.18}$$

And then substituting equation (2.18) into equation (2.16) gives the familiar simplified relationship for SRM torque,

$$T = \frac{i^2}{2} \frac{dL}{d\theta} \tag{2.19}$$

The reluctance of the flux path between the two diametrically opposite stator poles varies as a pair of rotor poles rotates into and out of alignment. The inductance of a phase winding is a maximum when the rotor is in the aligned position and a minimum when the rotor is in the nonaligned position. Since inductance is inversely proportional to reluctance a pulse of positive torque is produced if a current flow in a phase winding as the inductance of that phase winding is increasing. A negative torque contribution is avoided if the current is reduced to zero before the

inductance starts to decrease again. The rotor speed can be varied by changing the frequency of the phase current pulses while retaining synchronism with the rotor position [44-49].

The absence of permanent magnets or coils on the rotor means that the torque is produced purely by the saliency of the rotor laminations. The direction of torque produced is irrespective of the direction of the flux through the rotor, and hence the direction of current flow in the stator phase windings is not important. The unipolar phase current in the reluctance motor results in simpler and more reliable power converter circuits. By choosing a combination where there are two more stator poles than rotor poles, high torque and low switching frequency of the power converter can be achieved. Figure 2.6 shows the three positions for the SRM.

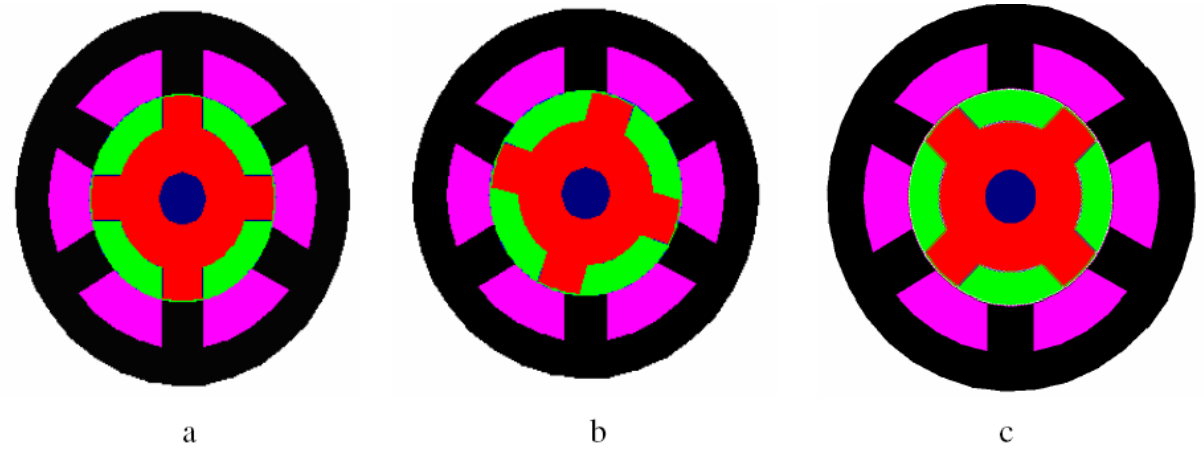

Fig. 2.6 (a) Aligned Position; (b) Misaligned [Overlap] Position; (c) Unaligned Position.

The rotor of an SRM is said to be at the aligned position with respect to a fixed phase if the reluctance has the minimum value; and the rotor is said to be at the unaligned position with respect to a fixed phase if the reluctance reaches its maximum value; otherwise the rotor is said to be at the misaligned position.

For an SRM with symmetric structure, i.e. both the stator and rotor poles are distributed symmetrically, respectively; the positions defined above with respect to phase 1 are shown in figure 2.7 [50-51].

CHAPTER 3

MATLAB MODELING OF SRM

3.1 Modeling of linear inductance profile based SRM

Most studies concerning dynamic simulation of switched reluctance machines (SRMs) [8] have been achieved from the programming, either in C language, Fortran, and also employing differential equation-based languages such as ACSL [55—57]. Even software designed to simulate electric network systems as the EMTDC and EMTP have been used. These techniques, although very useful, have lack of flexibility if new elements are brought, causing the increase of cost because of supplementary programming effort. On the other hand, very few simulation studies of the SRM have been achieved with circuit-based languages such as Spice, Simulink, Matrix, Tutsim, Vissim, and even Mathcad. The first simulations have been made thanks to the software Spice [53]. Unfortunately, this technique is not “elegant” because Spice is especially adapted to electronic circuit simulation [54]. Lately, there has been considerable progress in simulation software such as Matlab/Simulink, which allows a high flexible modeling environment to electrical machinery, as shown in [58], and in particular for SRMs as shown in [52].

In this research, Linear inductance profile based SRM was simulated using MATLAB. The sequences which are followed in this research is shown in the flow chart below-

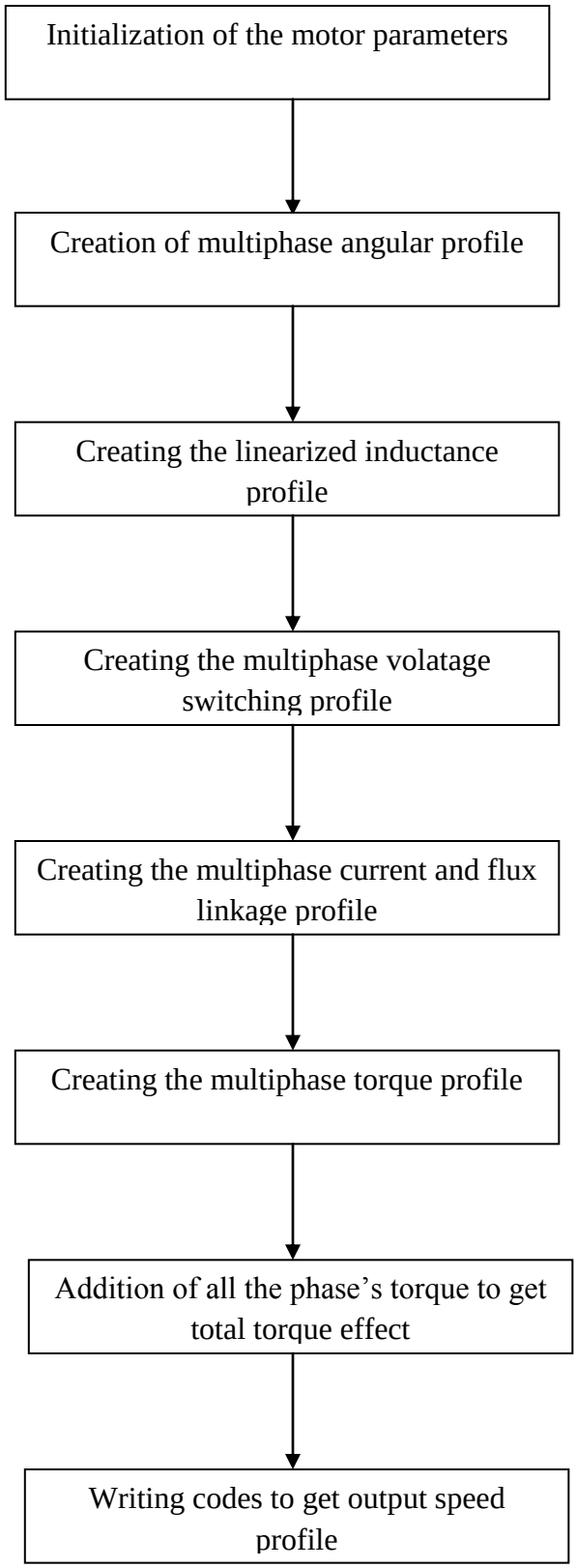

Fig. 3.1 Flow chart of modeling of SRM using MATLAB

All the necessary mathematical equations which govern the behavior of SRM are already elaborated in chapter 2. A brief introduction of MATLAB coding technique of the above flow chart is given below-

3.1.1 Initialization

Value of all the motor parameters such as number of stator and rotor poles, stator arc angle, rotor arc angle, turn on angle, turn off angle, commutation angle, separation of subsequent angle etc are defined. These constants values of the parameters can be changed for different motors or for different data sheets.

3.1.2 Creation of multiphase angular profile

To create different phase angle, MATLAB command *rem* is used and subsequent separation of angle for different phase is created in the model.

3.1.3 Creation of linear inductance profile

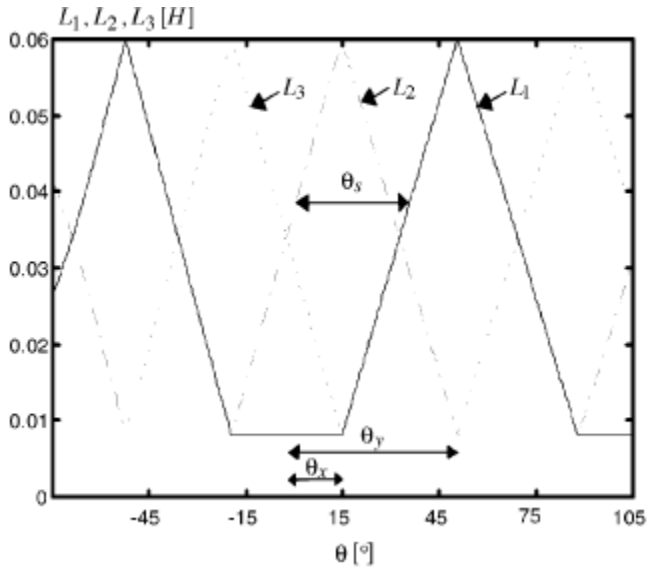

Fig. 3.2 Linear inductance profile of each phase of SRM [40]

Due to magnetic saturation, inductance profile is generally nonlinear. But if nonlinearity is included, then computational burden also increases. So, in this research to just understand the impact of genetic algorithm in tuning of PI controller, linearized inductance profile of SRM was simulated. Figure 3.2 shows the linear profile of inductance.

3.1.4 Creation of multiphase voltage switching profile

For SRM drive, many converter topologies have already been proposed and in this modeling, we assume H-bridge asymmetric converter while simulating the machine model. In the following figure, H-bridge asymmetric converter is shown.

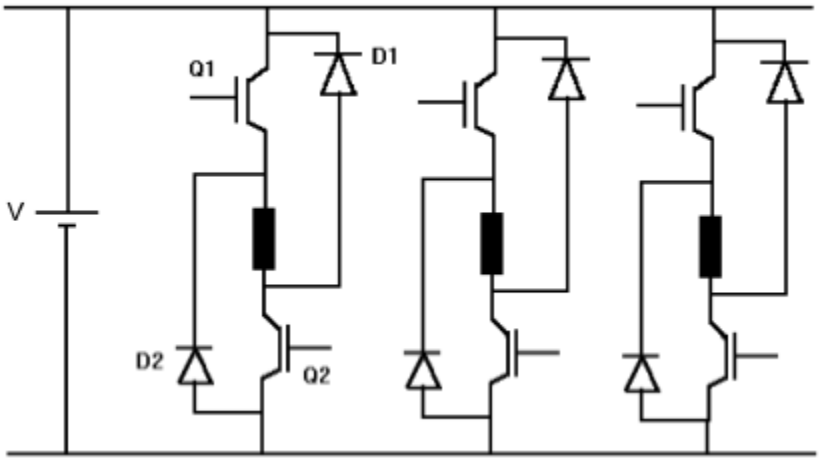

Fig. 3.3 H-bridge asymmetric converter.

The conditions for voltage switching are-

i. When 0° < Rotor angle < Turn on angle, then Voltage = 0;

ii. When Turn on angle <= Rotor angle < Turn off angle, then Voltage = +V;

iii. When Turn off angle <= Rotor angle < commutation angle (θ_d) then voltage= -V

The control takes place applying the voltage source to a phase coil at turn-on angle θ_{on} until a turn-off angle θ_{off}. After that, the applied voltage is reversed until a certain demagnetizing angle θ_d, which allows the return of the magnetic flux toward zero. To apply voltage V in one phase, the two IGBTs Q1 and Q2 in Fig. 3.3 must be ON. On the contrary, to apply the -V voltage and assure the current continuity, the two diodes D1 and D2 are used. We verify in more detail the phase energizing. From the phase voltage relation from equation 2.2,

$$V = i R_m + \frac{d\phi}{dt} \tag{2.2}$$

3.1.5 Creation of the multiphase current and flux linkage profile

Equation 2.2 can be rewritten as . Many well known methods are already established to implement integral equation in MATLAB. In this thesis, we followed the Euler's method to implement the voltage, current and flux linkage relation. An iterative loop was created for this purpose.

3.1.6 Creation of the multiphase torque profile

For linearized inductance profile SRM, the torque equation (2.19) is already proved in second chapter. Depending on the varying slope of the inductance for varying angle, torque profile will be created.

$$T = \frac{i^2}{2} \frac{dL}{d\theta}$$

3.1.7 Creation of total torque profile of the motor

A simple addition operation of all the individual phase's torque was done to get the overall torque of the machine.

3.1.8 Output speed profile

Mechanical equations for SRM are

$$J\frac{d\omega}{dt} = \Gamma - \Gamma_l - f\omega \tag{3.1}$$

Where ω is speed and T_l is the load torque. Load torque can be varied before running the simulation to see the stability of the speed profile for different load torque.

In this research, output characteristic curves are shown only for 3 phases SRM. But this MATLAB model can be assumed as a versatile one, which can be edited a bit to implement model of 4 phase or even more. The complete MATLAB code is given in the appendix A.

3.2 Simulation of Characteristic curves of 3-phase SRM in MATLAB

The simulated plots of voltage, current, Back EMF, torque and speed is shown below-

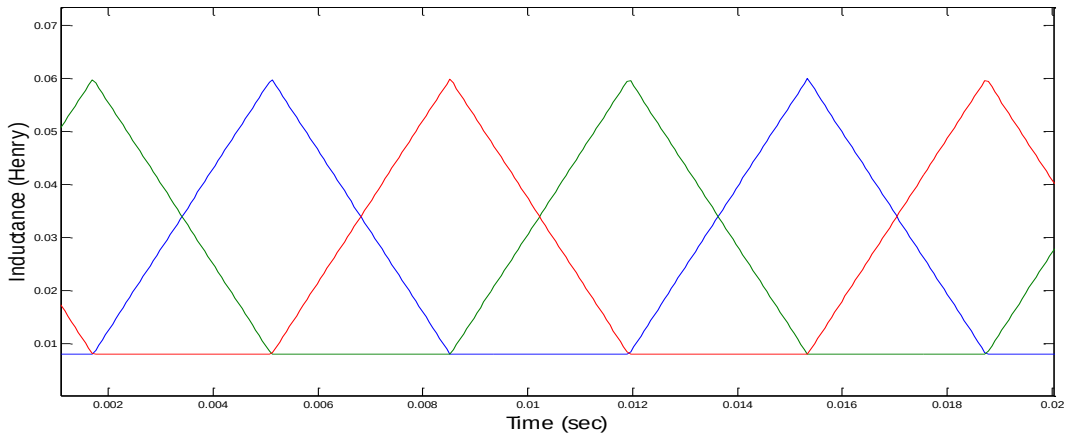

Fig 3.4 Linear inductance profile for 3-phase of SRM

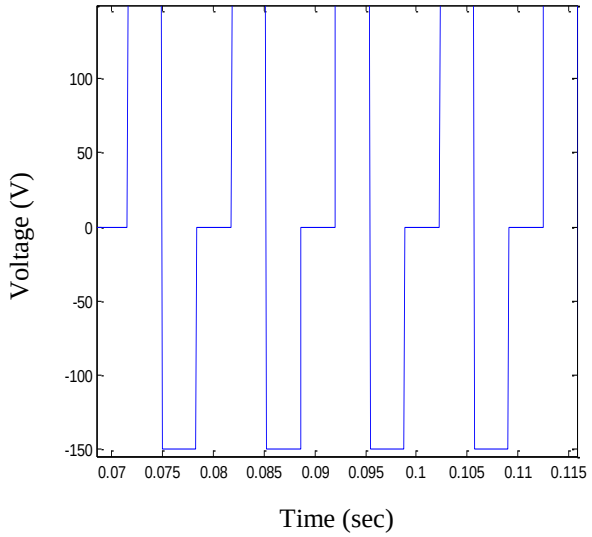

Fig. 3.5 Voltage profile for one phase of SRM

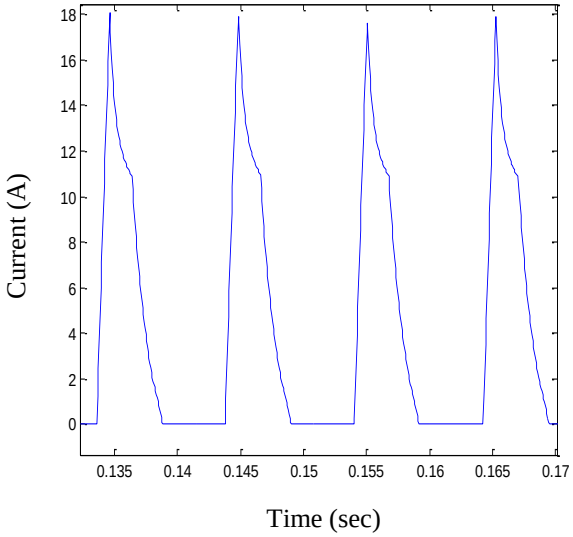

Fig. 3.6 Current profile for one phase of SRM

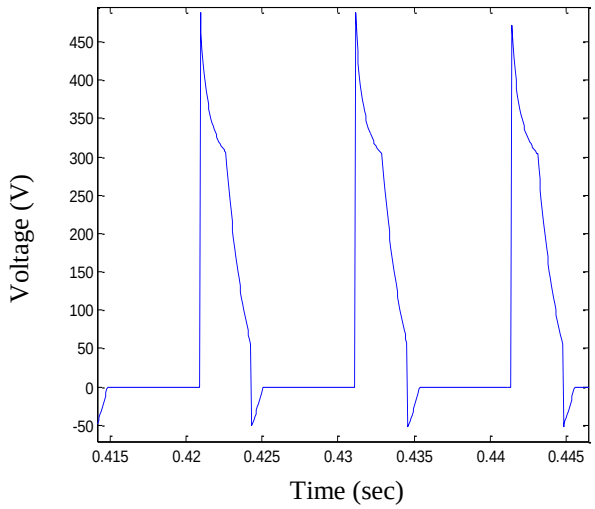

Fig. 3.7 Back EMF of one phase of SRM

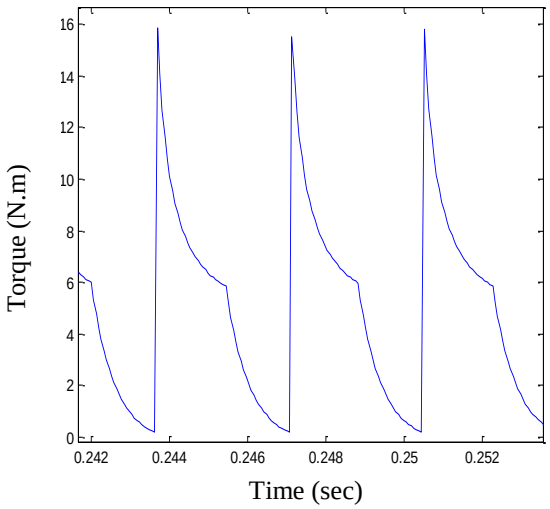

Fig 3.8 Overall torque profile of 3-phase SRM

All the graphs were obtained for turn on angle 30° and turn off angle 60° and for No load torque.

Equation 2.3 can be rewritten as below-

$$V = RI + L(\theta)\frac{dI}{dt} + I\omega\frac{dL}{d\theta} \tag{3.2}$$

In (3.2), term Iω— is the Back EMF induced voltage, which will be high for high speeds. To increase the current growth and avoid a high Back EMF opposition, the Turn on angle must be chosen in the same way as in Fig. 3.5-3.8, which means chosen when both inductance and the Back EMF are minimum. Using the linear model, the minimum Back EMF value will be zero since — = 0, as shown in Fig. 3.6. However, when the rotor position is in the zone where the inductance increases, the FEM voltage appears.

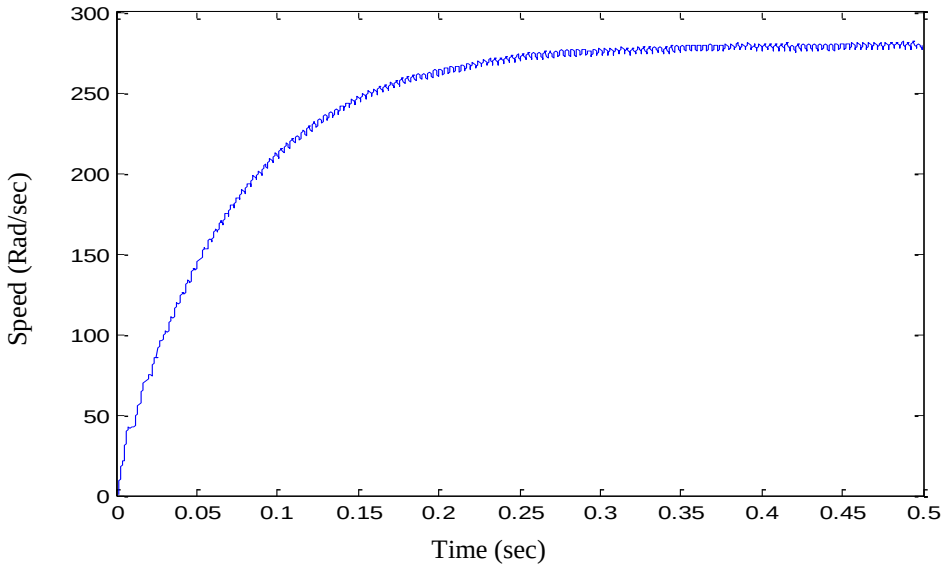

Fig 3.9 Speed profile of a 3-phase SRM

After, when the Back EMF surpasses voltage V, phase current starts to decrease until turn off angle is reached, as shown in Fig. 3.6. The sharp switching effects present in the voltage energizing strategy clearly introduce harmonics in the torque signal, by phase current signal, that increase the motor speed ripple. Since this energized strategy is usually applied only when the motor reaches high speed values, the mechanical system will attenuate these harmonics from the motor speed signal.

3.2.1 Impact of providing a disturbance for a small time interval

If for a very small time interval, an additional load torque as a disturbance is given to the system, then Speed profile and back EMF profile changes. The figures are show below-

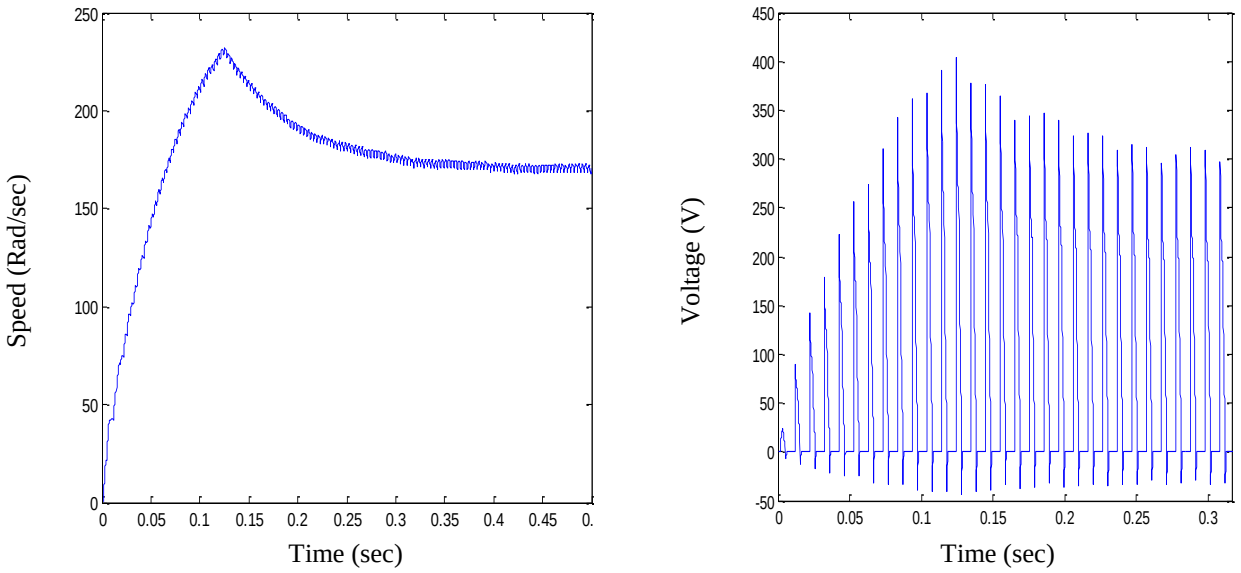

Fig. 3.10 Impact of a sudden disturbance on (a) Speed profile (b) Back EMF profile

This model is plotting all the characteristic parameters as it should have been and can also incorporate the impact of disturbance. Hence, it can be said that, this is a very useful model for SRM.

However, from this multiphase model, any number of phases can be implemented by doing some simple editing in the codes. So, in the next few chapters, small signal analysis of a linearized single phase, performance analysis of speed control loop, current control loop and overall SRM drive is simulated in MATLAB environment using the model and a comparison between genetically tuned PI controller and only PI controller is evaluated based on the MATLAB simulation.

CHAPTER 4

SMALL SIGNAL ANALYSIS OF A SINGLE PHASE LINEARIZED SRM

4.1 Derivation of the SRM small signal model

The design of the speed controller is an integral part of any drive system development. Due to the nonlinear nature of the SRM, the development of a block diagram is not as straightforward as in the case of the dc motor. Realizing that the SRM is very much similar to the series-excited dc machine (as seen from the torque and equivalent circuit development earlier in our text), it is feasible to proceed with linearization of the system equations to obtain a small signal model and a block diagram from which the transfer functions are developed. The transfer functions could be used, as in the case of other motor drives, to derive a proportional-plus-integral controller. From equation 2.19 and 3.1, the linearized mechanical equation can also be written as

$$\frac{1}{2} i^2 \frac{dL(\theta, i)}{d\theta} - T_e = J\frac{d\omega_m}{dt} + B\omega_m \tag{4.1}$$

Where B is the rotor friction constant and ω_m is the mechanical speed. Voltage equation (2.3) can also be rewritten as

$$v = R_s i + \frac{d[L(\theta, i)i]}{dt} = R_s i + L(\theta, i)\frac{di}{dt} + \frac{dL(\theta, i)}{d\theta}\omega_m i \tag{4.2}$$

The states of the SRM plant are the rotor speed, ω_m, and the phase current, i. By examining the SRM voltage and torque equations, there are terms where states are multiplied together resulting

in a nonlinear system. It is desirable to derive a linearized model to utilize a vast amount of knowledge on linear systems to synthesize the current and speed controllers. This section contains the derivation of a linearized model of the SRM. The inductance is assumed to be constant for the sake of simplification. The inductance is chosen as the mean value between the aligned inductance and the unaligned inductance at the rated current. The derivative of inductance with respect to rotor position is also assumed to be a constant and calculated between the conduction angles at the rated current value. This derivative has only a small change over the operating range of the motor.

Perturbing the system around a steady-state operating point with small signals, the new system states and inputs are-

$$i = i_o + \delta i \tag{4.3}$$

$$\omega_m = \omega_{mo} + \delta\omega_m \tag{4.4}$$

$$v = v_o + \delta v \tag{4.5}$$

$$T_l = T_{lo} + \delta T_l \tag{4.6}$$

where the extra subscript *o* indicates steady-state values of the states and inputs, and the small signals are indicated by δ preceding the variables. Substituting the perturbed variables in the system equations, it is seen that the steady-state terms cancel and the residual of these equations gives:

$$\frac{d\delta i}{dt} = \left(-\frac{R_s}{L} - \frac{1}{L}\frac{dL}{d\theta}\omega_{mo}\right)\delta i - \frac{1}{L}\frac{dL}{d\theta} i_o \delta\omega_m + \frac{\delta V}{L} \tag{4.7}$$

$$\frac{d\delta\omega_m}{dt} = \left(+\frac{1}{J}\frac{dL}{d\theta}i_o\right)\delta i - \frac{B}{J}\delta\omega_m - \frac{\delta T_\ell}{J} \tag{4.8}$$

Hereafter, the following substitutions are used:

$$R_{eq} = R_s + \frac{dL}{d\theta}\omega_{mo} \tag{4.9}$$

$$K_b = \frac{dL}{d\theta}i_o \tag{4.10}$$

$$\delta e = \frac{dL}{d\theta}i_o\delta\omega_m \tag{4.11}$$

where R_{eq} is the equivalent resistance, K_b is the emf constant, and δe is the induced emf. By using the small signal voltage and torque equations, the following block diagram is derived for the linearized SRM plant model. Note that this model is similar to the separately excited dc machine model. The block diagram of the linearized SRM is shown in Figure 4.1. The load is assumed to be frictional; that way, the load torque is treated as an integral component of the system but not as a disturbance. For the sake of simplicity, only one current feedback loop is shown in Figure 4.1, even though for a q-phase SRM there will be q current feedback loops.

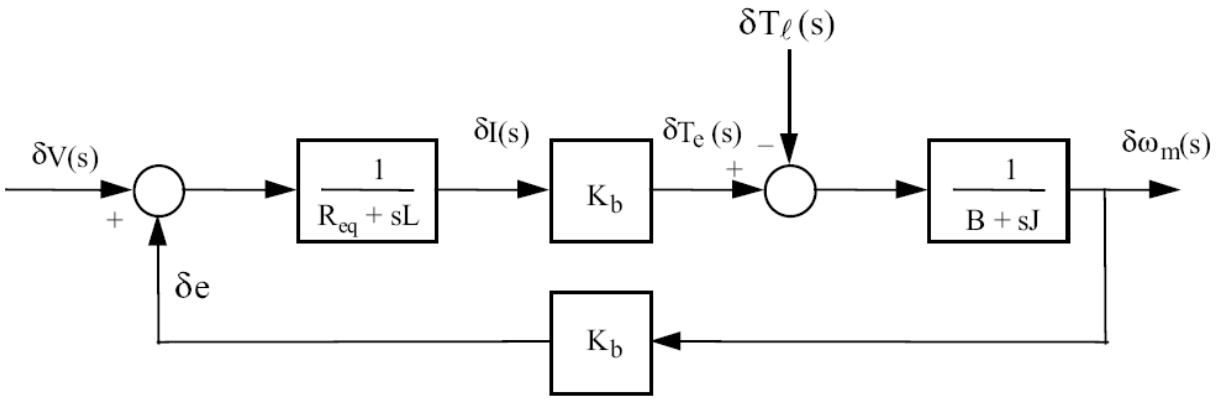

Fig. 4.1 Block diagram of the linearized SRM.

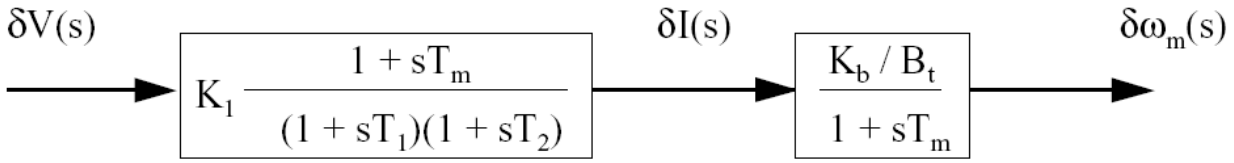

Fig. 4.2 Reduced block diagram of the SRM.

These current loops are identical but shifted in phase, so there is no need to consider more than one phase for control modeling, analysis, and design. The back emf and current feedback loops cross each other, resulting in cross coupling of these loops. Further, it makes the task of designing a current controller and later the speed controller very difficult. For this reason, the SRM block diagram is cast in a different form by removing the back emf feedback loop but absorbing it in a form which leads to a two-stage transfer function as shown in Figure 4.2, very much similar to dc machines, where

$$B_t = B + B_\ell \tag{4.12}$$

$$K_1 = \frac{B_t}{K_b^2 + R_{eq}B_t} \tag{4.13}$$

$$T_m = \frac{J}{B_t} \tag{4.14}$$

$$-\frac{1}{T_1}, -\frac{1}{T_2} = -\frac{1}{2}\left[\frac{B_t}{J} + \frac{R_{eq}}{L}\right] \pm \sqrt{\frac{1}{4}\left(\frac{B_t}{J} + \frac{R_{eq}}{L}\right)^2 - \frac{K_b^2 + R_{eq}B_t}{JL}} \tag{4.15}$$

4.2 Steady state stability of the speed curve due to a sudden disturbance

From equation (4.7) and (4.8), change of speed and change of current was made state of the linearized SRM plant. A sudden disturbance was given to check the stability of the SRM without having any speed controller in the feedback path. Value of slope of the inductance was changed to see the impact of parameter value change.

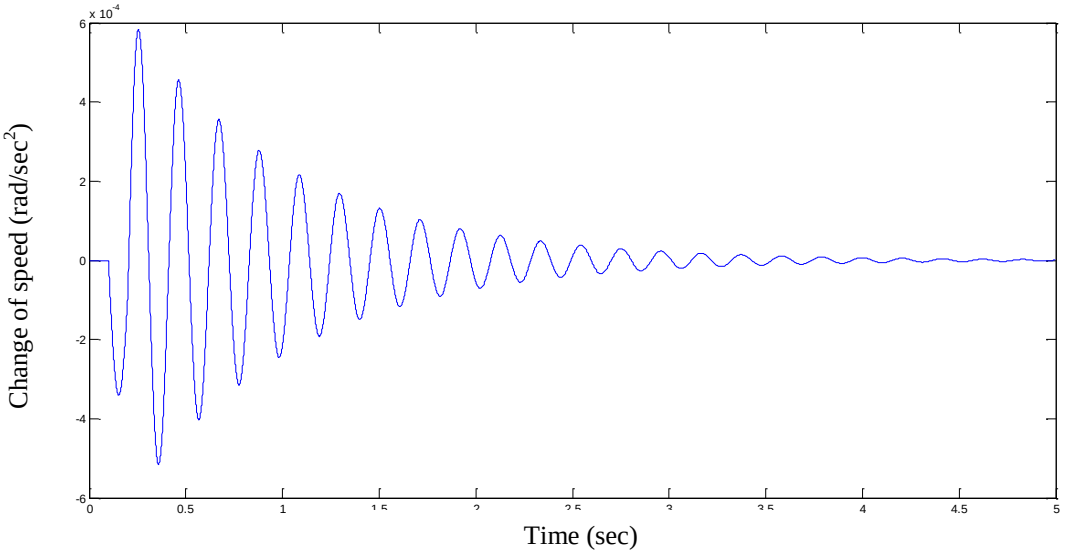

Fig. 4.3 Change of speed with respect to time for DL=0.234 H/rad and $dW_{m=0}$

As the input $dW_m=0$, change of speed profile should also trace 0 line. But as a disturbance is given for a small time interval, there is some oscillation in the curve. But the pattern clearly shows that, it is going to be declined and at one stage, it will again be zero. So, for slope (DL) = 0.234 h/rad, the SRM linearized plant will reach stability after a certain time. Now, it can be checked for another value of DL.

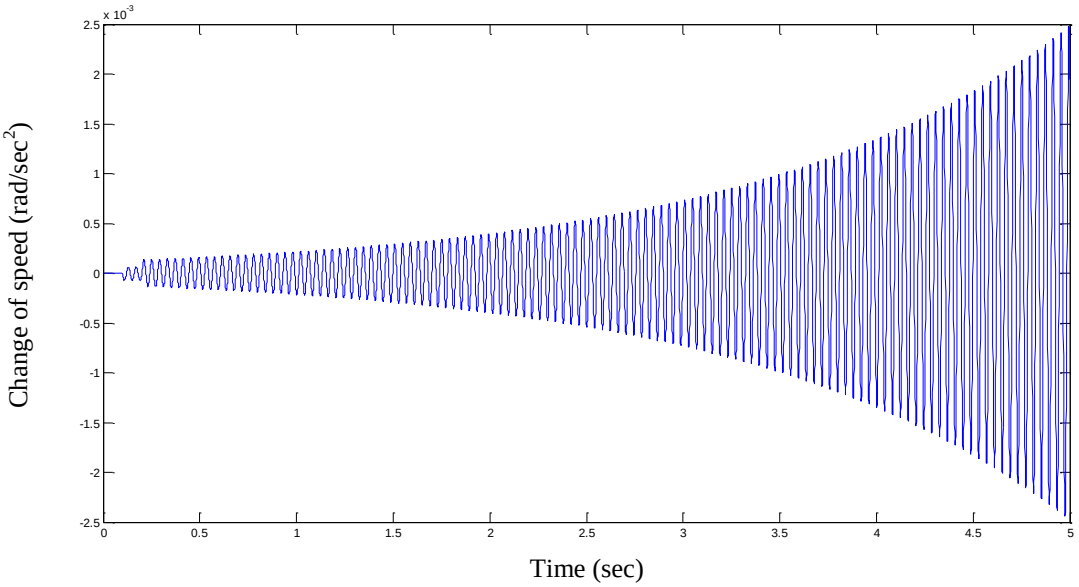

Fig. 4.4 Change of speed with respect to time for DL=1.2 H/rad and $dW_{m=0}$

In this curve, it is very much clear that the pattern is rising and it will never trace zero line. So, this linearized SRM without having any speed controller cannot give absolute stability irrespective of any parameter value change. From these simulation results, it was understood that this model is parameter sensitive and in the next few chapters, this thing was kept in mind before doing further simulation based on this model.

Details of the MATLAB code for this linearized small signal model SRM simulation is given in Appendix-B.

CHAPTER 5

PERFORMANCE ANALYSIS OF SRM DRIVE WITH PI CONTROLLER

5.1 Design of Current controller

The SRM is nonlinear as there is a term in the voltage equation containing the product of rotor speed and phase current. Nonlinearity in the system makes the controller design difficult. If the system is linearized, then the knowledge base of well-known linear control systems theory can be applied to the controller design. Further, such an approach is analytical and capable of providing insight into the system unlike the computer-aided design approach. This controller is not capable of high performance as its design was based on an operating point, whereas in a variable speed SRM drive system the operating point continually changes. A nonlinear controller enables linearization and decoupling of the current loop, resulting in high performance. Decoupling is essential as the current in one phase will affect other phase current due to the presence of mutual coupling. Normally two phases conduct for part of a phase conduction period; when the outgoing phase current is being commutated, the incoming phase current is in the process of rising to its required level. These currents contribute to mutual flux linkages, even though the mutual inductances are minimized in the machine design. The design of the current controllers is an integral part of any drive system development. Due to the nonlinear nature of the SRM, the development of a block diagram is not as straightforward as in the case of the dc motor. Realizing that the SRM is very much similar to the series-excited dc machine (as seen from the torque and equivalent circuit development earlier in our text), it is feasible to proceed

with linearization of the system equations to obtain a small signal model and a block diagram from which the transfer functions are developed. The transfer functions could be used, as in the case of other motor drives, to derive a proportional-plus-integral controller.

A speed-controlled SRM drive system is shown in Figure 5.1. Rotor speed is converted to a voltage signal through a tachogenerator which then is filtered to provide ω_r^*, which is then compared with its reference The speed error signal is amplified and conditioned with the speed controller which normally is a proportional-plus integral type. The output of this speed controller is a voltage signal proportional to current command signal I^*. A current feedback signal in volts is compared with this command signal to generate a current error. The current error is processed through a PI controller to produce a command signal for the power converter. The power converter is

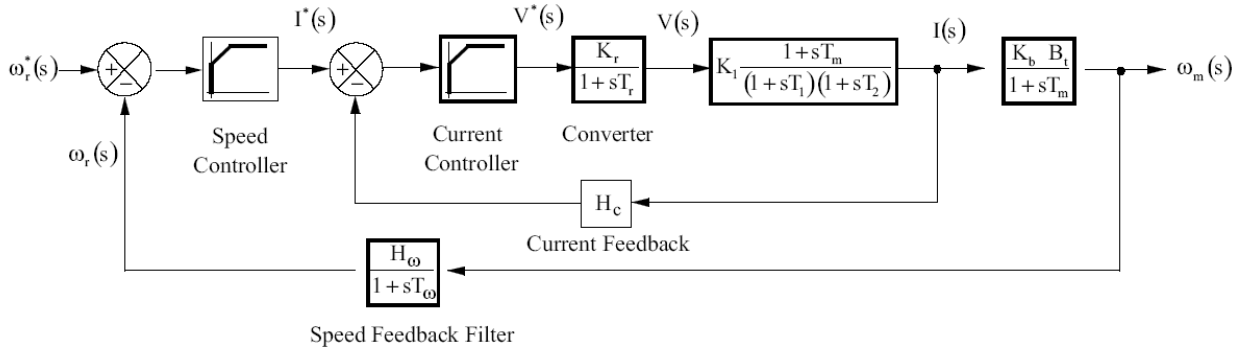

Fig. 5.1 Block diagram of the SRM drive.

Modeled as a gain with a first-order lag, and both of these constants may be measured or evaluated in the design stage. The power converter gain is

$$K_r = \frac{V_{dc}(\text{nominal})}{V_{cm}} \tag{5.1}$$

where V_{cm} is the maximum control voltage. The time constant of the converter, T_r, assuming PWM control of the converter with a carrier frequency of f_c, is given by:

$$T_r = \frac{T_c}{2} = \frac{1}{2f_c} \tag{5.2}$$

5.2 Design of current and speed loop

In this thesis, speed loop and current loop was designed based on all the value of motor constants obtained from a specification sheet. The specification sheet of a practical motor is shown below-

5.2.1 Example 1: 5-hp SRM Drive System

To validate the design technique using the linearized model, a 5-hp SRM is considered for the current and speed controller designs [59]. The specifications for the 5-hp SRM drive are listed below:

Table 5.1 Motor and system parameters

Motor and system parameters	
Command signal levels	+-10
Dc link voltage	400 V
Max. current	15 A
PWM chopping frequency	8 kHz
Phase resistance	0.931 Ω
Power	5 hp
Rated current	10 A (1 p.u.)
Rated speed	2500 rpm

Rotor friction constant	0.001 N · m/rad/sec
Rotor inertia	0.006 kg/m^2
Speed feedback gain	0.0383 V/rad/sec
Speed feedback time constant	0.1 sec

A PI controller is selected for the current controller because of its simplicity to implement and its widespread industrial use. The block diagram of the current loop is shown in Figure 5.2. The transfer function of the current controller is

$$G_c(s) = \frac{K_c(1 + sT_{cc})}{sT_{cc}} \tag{5.3}$$

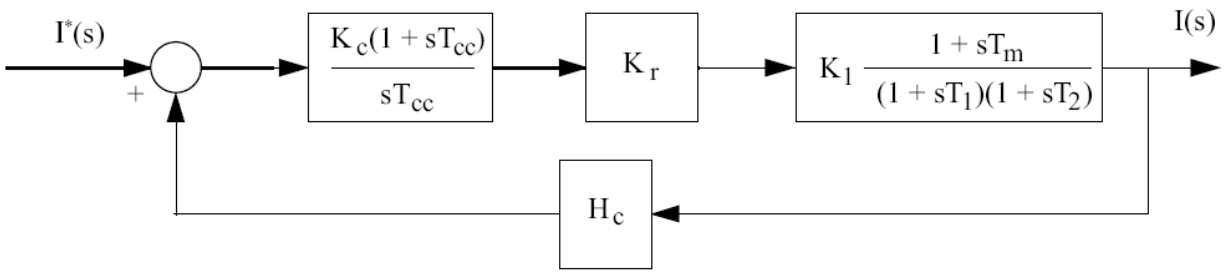

Fig. 5.2 Block diagram of the current control loop.

The time delay of the converter is neglected here due to the assumption that the switching frequency is at least 10 times greater than that of the electrical time constant in the current loop. Further, such an assumption simplifies the block diagram. Since the mechanical time constant of the system, T_m, is large, $(1 + sT_m)$ can be approximated as sT_m. With this approximation, the current loop becomes a second order system. The approximated system is given below:

$$\frac{I(s)}{I^*(s)} = \frac{K_c K_r K_1 T_m (1 + sT_{cc})}{T_c (1 + sT_1)(1 + sT_2) + H_c K_c K_r K_1 T_m (1 + sT_{cc})} \quad (5.4)$$

In designing the current controller gain and time constant, it is desirable to specify a bandwidth for the current loop based on the switching frequency of the converter. In order to approximate the converter as a simple gain, the bandwidth of the converter must be ten times faster than the bandwidth of the current loop.

To design the current controller using the bandwidth method, the characteristic equation of the approximated current loop is used, as shown below:

$$s^2 + s\left(\frac{T_1 + T_2 + H_c K_c K_r K_1 T_m}{T_1 T_2}\right) + \frac{H_c K_c K_r K_1 T_m + T_{cc}}{T_{cc} T_1 T_2} \quad (5.5)$$

Since it is a second-order equation, the natural frequency of oscillation, ω_n, and damping ratio, ζ, of a second-order system may be used to obtain the current controller gain and time constant. Given below are the equations which specify the damping (5.6) and the natural frequency of the approximated system (5.7):

$$2\zeta\omega_n = \frac{T_1 + T_2 + H_c K_c K_r K_1 T_m}{T_1 T_2} \quad (5.6)$$

$$\omega_n^2 = \frac{H_c K_c K_r K_1 T_m + T_{cc}}{T_{cc} T_1 T_2} \quad (5.7)$$

The gain, K_c, and the time constant, T_c, may be solved from Eqs. (5.6) and (5.7), respectively, for a given natural frequency and damping ratio. The following are equations for Kc and T_c:

$$K_{cc} = \frac{2\zeta T_1 T_2 \omega_n - T_1 - T_2}{H_c K_r K_1 T_n} \quad (5.8)$$

$$T_{cc} = \frac{H_c K_c K_r K_1 T_m}{T_1 T_2 \omega_n^2 - 1} \tag{5.9}$$

Inductance of the SRM phases is assumed to be the mean value of the unaligned inductance and aligned inductance at the rated current. This value turns out to be 22.1 mH. The slope of the inductance curve is needed in order to calculate the linearized torque/back emf constant. Using the inductance values at the rated current, the approximate slope of the inductance profile is 0.234 H/rad. By using earlier equations, the linearized torque/back emf constant, K_b, and the linearized phase resistance, R_{eq}, are calculated:

$$R_{eq} = R_s + \frac{dL}{d\theta}\omega_{mo} = 0.931 + (0.234)(261) = 62\,\Omega \tag{5.10}$$

$$K_b = \frac{dL}{d\theta} i_o = (0.234)(10) = 2.34 \tag{5.11}$$

The following constants are calculated in order to begin the design of the controllers:

1. Converter gain:

$$K_r = \frac{V_{dc}}{v_c} = \frac{400}{10} = 40 \tag{5.12}$$

2. Current transducer gain:

$$H_c = \frac{v_c}{i_{max}} = \frac{10}{15} = 0.667\ \text{V/A} \tag{5.13}$$

3. Motor transfer function:

$$B_t = B + B_l = 0.001 + 0 = 0.001 \tag{5.14}$$

$$K_1 = \frac{B_t}{K_b^2 + RB_t} = 0.000182 \tag{5.15}$$

$$T_m = \frac{J}{B_t} = \frac{0.006}{0.001} = 6\ \text{s} \tag{5.16}$$

$$-\frac{1}{T_1}, -\frac{1}{T_2} = -\frac{1}{2}\left[\frac{B_t}{J} + \frac{R_{eq}}{L}\right] \pm \sqrt{\frac{1}{4}\left(\frac{B_t}{J} + \frac{R_{eq}}{L}\right)^2 - \frac{K_b^2 + R_{eq}B_t}{JL}} \tag{5.17}$$

$$T_1 = 0.0668 \text{ s} \quad T_2 = 0.000358 \text{ s} \tag{5.18}$$

To design the current controller gain and time constant, Eqs. (5.6) - (5.9) are used. For a bandwidth of 1600 Hz and damping ratio of 0.707 for the current loop performance, the controller gains are:

$$K_c = \frac{1.414T_1T_2\omega_n - T_1 - T_2}{H_cK_rK_1T_m} = 9.36 \tag{5.19}$$

$$T_{cc} = \frac{H_cK_cK_rK_1T_m}{T_1T_2\omega_n^2 - 1} = 0.000113 \text{ s} \tag{5.20}$$

5.2.2 Design of speed loop

To simplify the design of the speed control loop, it is assumed that the delay of the current loop is negligible due to the fact that usually the speed of response of the current loop is at least ten times faster than the response of the speed loop. To further simplify the design equations, the current loop gain is approximated as unity and its time delay is neglected as it is very, very small compared to all other time constants. Normally, the delay due to the speed feedback may be neglected, which would reduce the system to a second-order system, but when

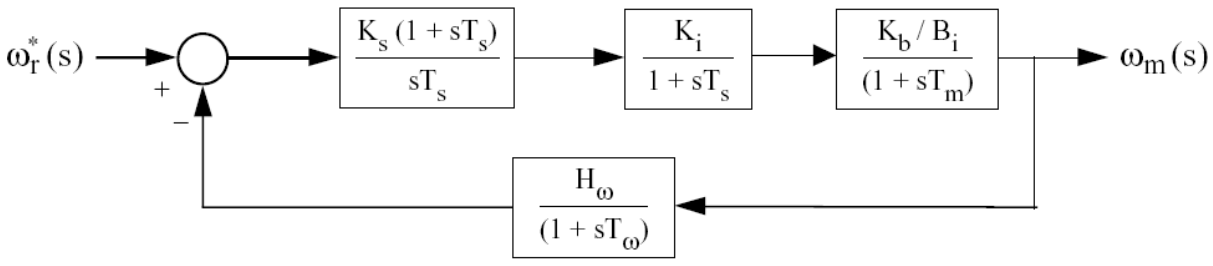

Fig. 5.3 Approximated speed loop block diagram.

the speed feedback delay is comparable to the delay of the other subsystems it must be considered in the design process. Given in Figure 5.3 is the block diagram of the approximated Speed loop.

Consider the speed loop transfer function given as:

$$GH_{\omega}(s) = \frac{K_s(1+sT_s)}{sT_s} \cdot \frac{K_b/B_t}{(1+sT_m)} \cdot \frac{H_{\omega}}{(1+sT_{\omega})} \tag{5.21}$$

Near the gain crossover frequency, as T_m is large compared to other time constants, the following approximation is made:

$$(1+sT_m) \cong sT_m \tag{5.22}$$

This results in the speed loop transfer function as:

$$GH_{\omega}(s) \cong \frac{K_2 \cdot K_s}{T_s} \cdot \frac{(1+sT_s)}{s^2(1+sT_{\omega})} \tag{5.23}$$

Where

$$K_2 = \frac{K_b H_{\omega}}{B_t T_m} \tag{5.24}$$

From this, the closed loop speed-transfer function is approximately derived as:

$$\frac{\omega_m(s)}{\omega_r^*(s)} \cong \frac{1}{H_{\omega}}\left[\frac{GH_{\omega}(s)}{1+GH_{\omega}(s)}\right] = \frac{1}{H_{\omega}} \cdot \frac{a_0 + a_1 s}{a_0 + a_1 s + a_2 s^2 + a_3 s^3} \tag{5.25}$$

where the coefficients of the polynomials are

$$a_0 = \frac{K_2 K_s}{T_s} \tag{5.26}$$

$$a_1 = K_2 K_s \tag{5.27}$$

$$a_2 = 1 \tag{5.28}$$

$$a_3 = T_{\omega} \tag{5.29}$$

To optimize the gain of the closed-loop, speed-transfer function, the denominator of the magnitude response function is minimized to provide a flat frequency response (i.e., symmetric optimum). Such a condition gives a relationship between the coefficients of the characteristic polynomial leading to evaluation of the speed controller constants as given below:

$$\left|\frac{\omega_m(j\omega)}{\omega_r^*(j\omega)}\right| = \frac{1}{H_\omega}\sqrt{\frac{a_0^2 + a_1^2\omega^2}{[a_0^2 + a_1^2 - (2a_0a_2)\omega^2 + (a_2^2 - 2a_1a_3)\omega^4 + a_3\omega^6]}} \tag{5.30}$$

from which the coefficients are related by:

$$a_1^2 = 2a_0a_2 \tag{5.31}$$

$$a_2^2 = 2a_1a_3 \tag{5.32}$$

to give a flat frequency response. From the above two conditions, the speed controller constants are evaluated as:

$$K_s = \frac{1}{2K_2T_\omega} \tag{5.33}$$

$$T_s = 4T_\omega \tag{5.34}$$

Substituting for K_s and T_s we can obtain the closed-loop, speed-transfer function in terms of T_ω as:

$$\frac{\omega_m(s)}{\omega_r^*(s)} = \frac{1}{H_\omega} \cdot \frac{1 + 4T_\omega s}{1 + 4T_\omega s + 8T_\omega^2 s^2 + 8T_\omega^3 s^3} \tag{5.35}$$

For the open-loop gain function, the corner points are $1/4T_\omega$ and $1/T_\omega$ with a gain crossover frequency of $1/2T_\omega$. In the vicinity of the gain crossover frequency the slope of the magnitude response is –20 dB/decade, which is the most desirable characteristic for good dynamic behavior. Because of its symmetry at the gain crossover frequency, this transfer function is known as a symmetric optimum function. Further, this transfer function has the following features:

1. Approximate time constant of the system is $4T\omega$.

2. The step response is given by:

$$\omega_r(t) = \frac{1}{H_\omega}\{(1 + e^{-t/2T_\omega} - 2e^{-t/4T_\omega}\cos(\sqrt{3}t/4T_\omega))\} \tag{5.36}$$

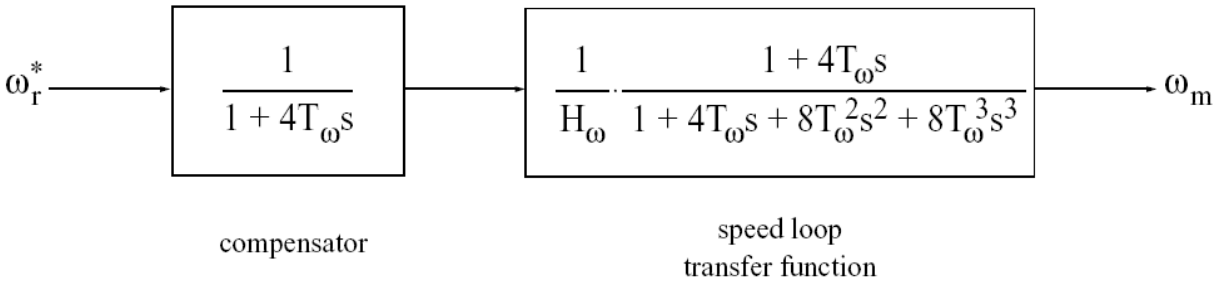

Fig. 5.4 Smoothing of the overshoot with a compensator.

with a rise time of $3.1T_\omega$, maximum overshoot of 43.4%, and a settling time of $16.5T_\omega$.

3. Because the overshoot is high, it can be reduced by compensating for its cause (i.e., the zero by a pole in the speed command path, as shown in the Figure 5.4). The resulting transfer function of the speed to its command is

$$\frac{\omega_m(s)}{\omega_r^*(s)} = \frac{1}{H_\omega}\left[\frac{1}{1 + 4T_\omega s + 8T_\omega^2 s^2 + 8T_\omega^3 s^3}\right] \tag{5.37}$$

whose step response is

$$\omega_r(t) = \frac{1}{H_\omega}\left\{\left(1 - e^{-t/4T_\omega} - \frac{2}{\sqrt{3}}e^{-t/4T_\omega}\sin(\sqrt{3}t/4T_\omega)\right)\right\} \tag{5.38}$$

with a rise time of $7.6T_\omega$, maximum overshoot of 8.1%, and a settling time of $13.3T_\omega$. Even though the rise time has increased, the overshoot has been reduced to approximately 1/5 th of its previous value, and the settling time has come down by 19%.

4. The poles of the transfer function are

$$s = -\frac{1}{2T_\omega}; \quad -\frac{1}{4T_\omega} \pm j\frac{\sqrt{3}}{4T_\omega} \tag{5.39}$$

As the real part of the poles are negative and there are no repeated poles at the origin, the system is asymptotically stable. Hence, in the symmetric optimum design, the system stability is guaranteed and there is no need to check for it in the design process. Whether this is true for the original system without approximation will be explored in the following example.

5. Symmetric optimum eliminates the effects due to the disturbance very rapidly compared to other optimum techniques employed in practical systems such as linear and modulus optimum, etc. This approach indicates one of the possible methods to synthesize the speed controller. The judicious choice of approximation is based on the physical constants of the motor, converter, and transducer gains and delays.

5.3 Simulation results

Step response of current control loop and speed control loop best describes the behaviour of the system. From the MATLAB model of current loop and speed loop, step response was plotted and is shown below-

Step response of the current control loop with PI controller only

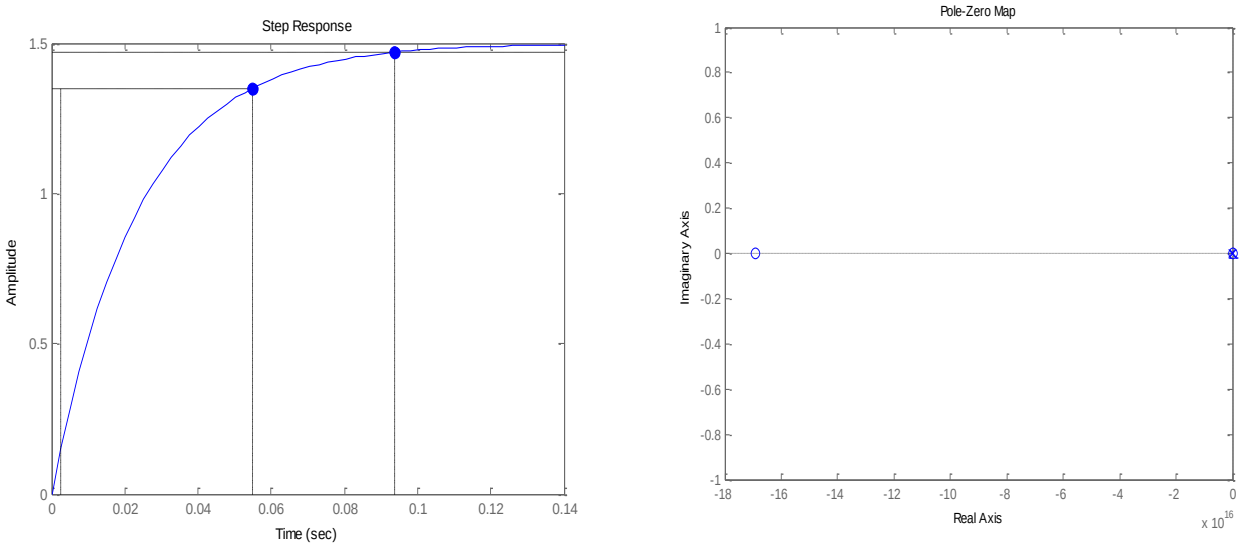

Fig. 5.5 (a) Step response, (b) pole-zero mapping of current control loop with PI controller

From the above figure, we see the following parameter values as:

Rise time = 0.0524 sec

Settling time = 0.0939 sec

Overshoot = 0% (over damped)

Step and impulse response of the speed control loop with PI controller only

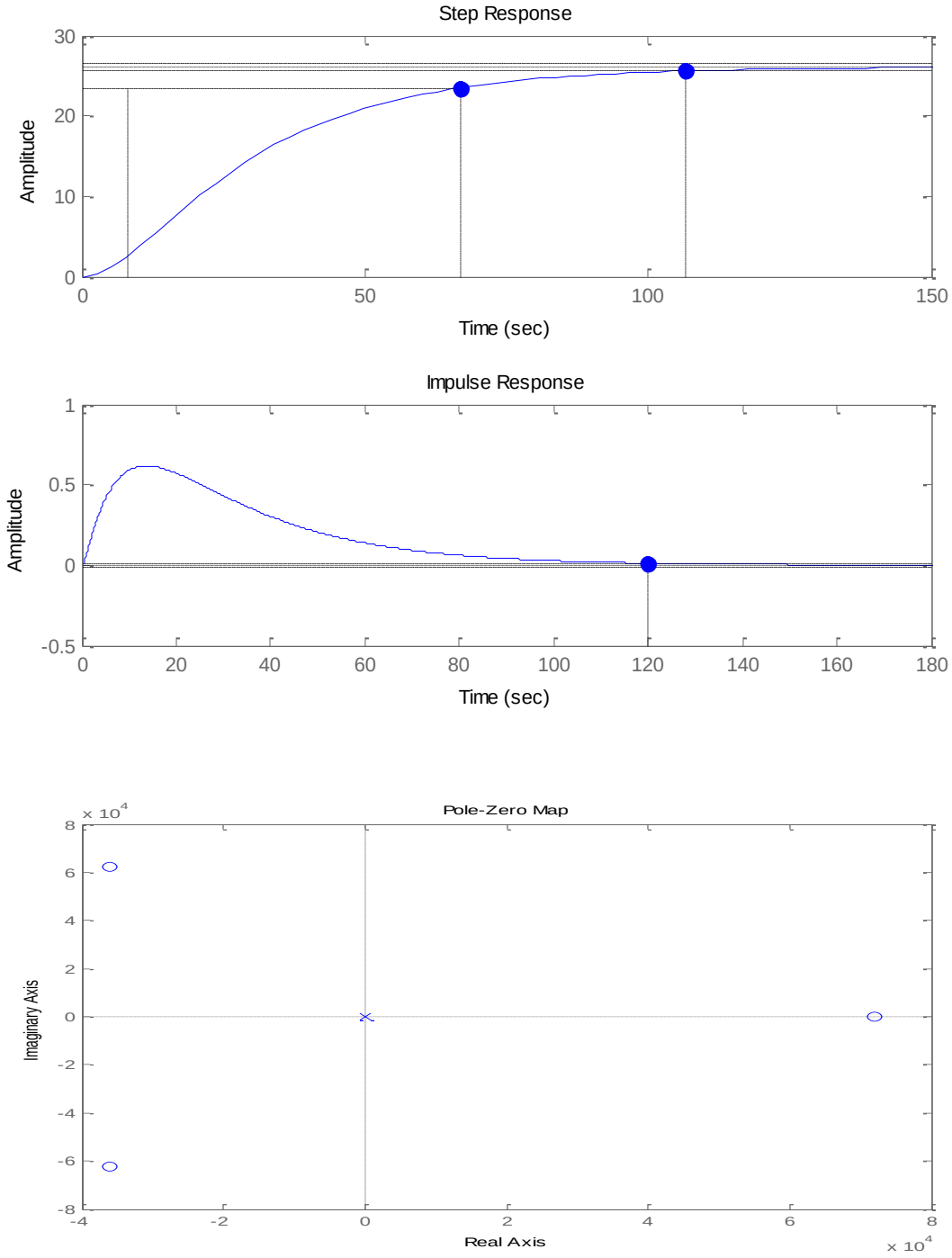

Fig. 5.6 (a) Step and impulse response, (b) Pole-zero mapping of speed control loop with PI controller

Rise time = 58.8 sec,

Settling time = 107 sec and Overshoot = 2e-14 %.

Step response of the overall speed controlled SRM drive including current control loop also (Fig. 5.1)

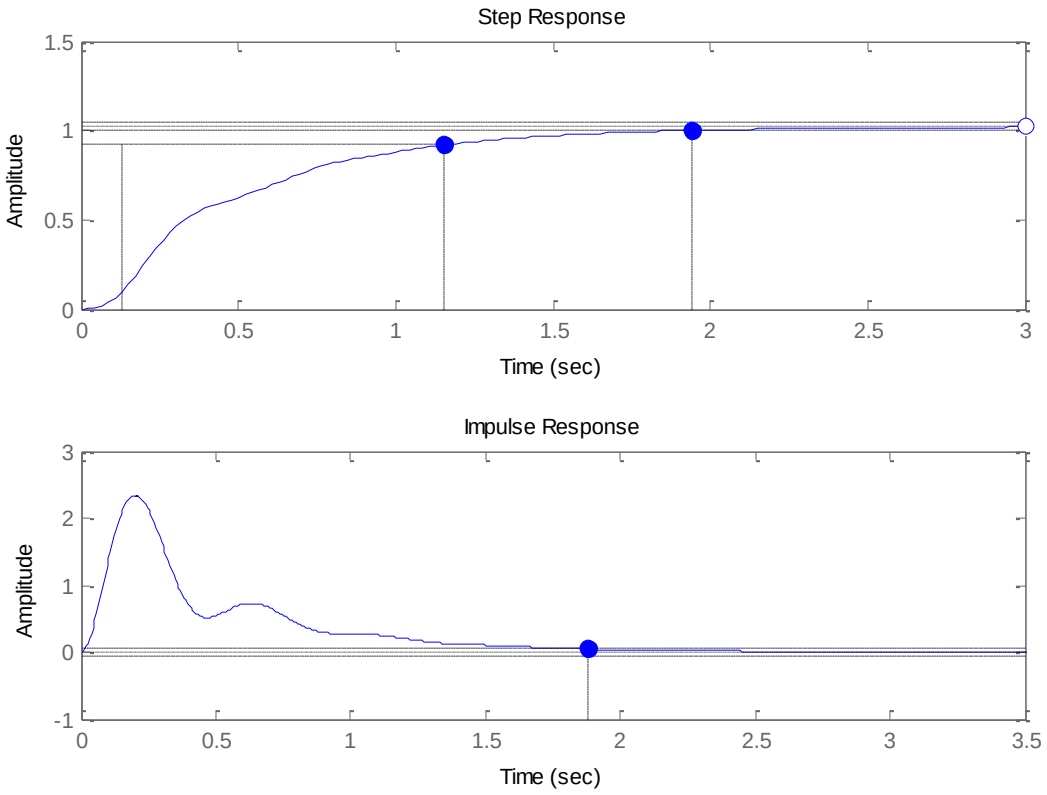

Fig. 5.7 Step and impulse response of speed controlled SRM drive including current loop & speed loop

Rise Time = 1.02 sec

Settling time = 1.94 sec

Overshoot = 0%

Step response genearlly depicts the behavior of any dynamic system. And performance analysis of any system includes the observation of rise time, settling time, steady state value and overshoot. In this chapter, rise time and settling time of individual current control loop, speed control loop and also overall drive system including both the loops inside has been noted. In the next couple of chapters, these charateristics such as rise time and settling time will be observed after the constants of the PI controller are tuned using the genetic algorithm.

CHAPTER 6

THEORY OF BASIC GENETIC ALGORITHM

Genetic Algorithms (GAs) are adaptive heuristic search algorithm based on the evolutionary ideas of natural selection and genetics. Genetic algorithms (GAs) are a part of Evolutionary computing, a rapidly growing area of artificial intelligence. GAs are inspired by Darwin's theory about evolution - "survival of the fittest". GA represent an intelligent exploitation of a random search used to solve optimization problems. GAs, although randomized, exploit historical information to direct the search into the region of better performance within the search space. In nature, competition among individuals for scanty resources results in the fittest individuals dominating over the weaker ones. In engineering and mathematics, Genetic algorithms is considered as a process of optimization. The problems are first formulated as mathematical models expressed in terms of functions and then to find a solution, discover the parameters that optimize the model or the function components that provide optimal system performance. It is more robust than the artificial intelligent techniques []. Unlike older AI systems, the GA's do not break easily even if the inputs changed slightly, or in the presence of reasonable noise. While performing search in large state-space, or multi-modal state-space, or n-dimensional surface, a genetic algorithms offer significant benefits over many other typical search optimization techniques like - linear programming, heuristic, depth-first, breath-first.

6.1 Parameter optimization

Optimization is a process that finds a best, or optimal, solution for a problem. The Optimization problems are centered around three factors:

1. **An objective function** which is to be minimized or maximized;

Examples: # In manufacturing, we want to maximize the profit or minimize the cost .

In designing an automobile panel, we want to maximize the strength.

2. **A set of unknowns or variables** that affect the objective function,

Examples:

In manufacturing, the variables are amount of resources used or the time spent.

In panel design problem, the variables are shape and dimensions of the panel.

3. **A set of constraints** that allow the unknowns to take on certain values but exclude others;

Examples: # In manufacturing, one constrain is, that all "time" variables to be non-negative.

In the panel design, we want to limit the weight and put constrain on its shape.

An optimization problem is defined as finding values of the variables that minimize or maximize the objective function while satisfying the constraints.

6.1.1 Search Optimization Algorithms

Fig. below shows different types of Search Optimization algorithms.

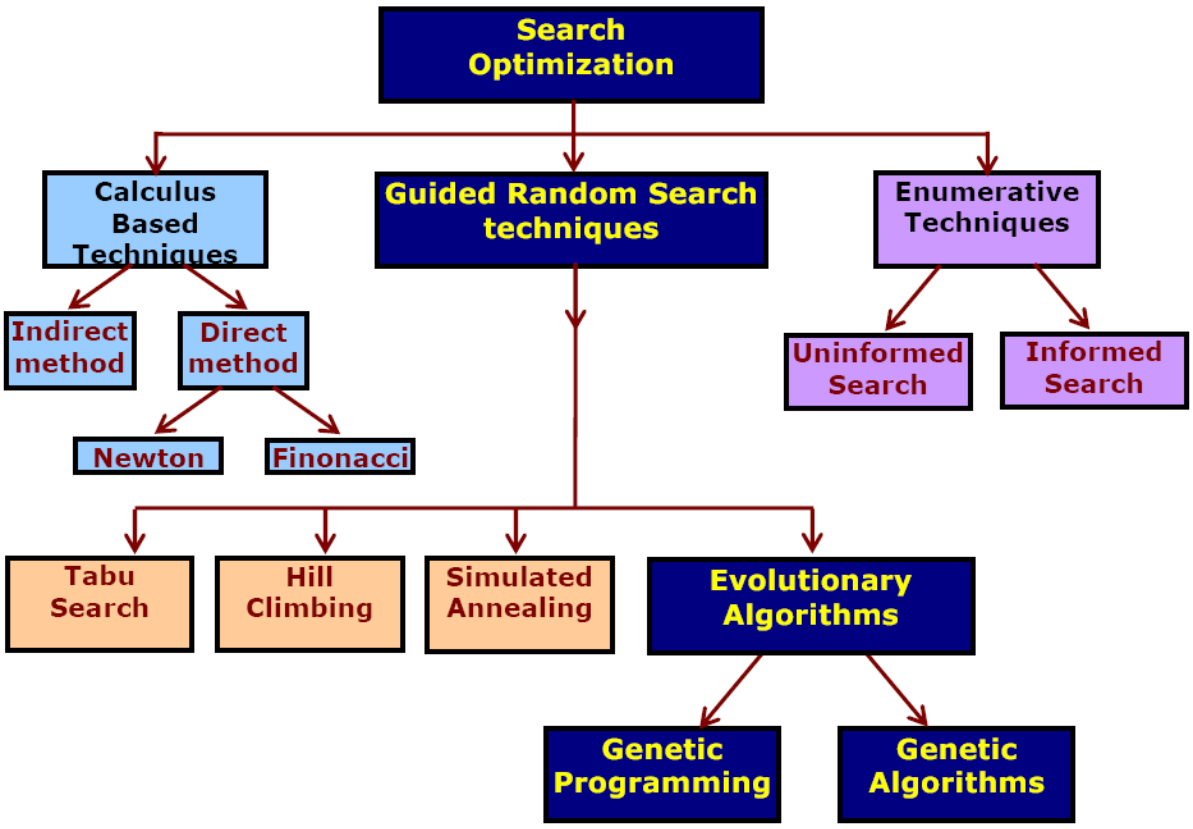

Fig. 6.1 Taxonomy of Search Optimization techniques

The Evolutionary Algorithms include Genetic Algorithms and Genetic Programming. Evolutionary Algorithm (EA) is a subset of Evolutionary Computation (EC) which is a subfield of Artificial Intelligence (AI). Evolutionary Computation (EC) is a general term for several computational techniques. Evolutionary Computation represents powerful search and optimization paradigm influenced by biological mechanisms of evolution: that of natural selection and genetic. Evolutionary Algorithms (EAs) refers to Evolutionary Computational models using randomness and genetic inspired operations. EAs involve selection, recombination,

random variation and competition of the individuals in a population of adequately represented potential solutions. The candidate solutions are referred as chromosomes or individuals. Genetic Algorithms (GAs) represent the main paradigm of Evolutionary Computation. GAs simulate natural evolution, mimicking processes the nature uses such as Selection, Crosses over, Mutation and Accepting. GAs simulates the survival of the fittest among individuals over consecutive generation for solving a problem. Genetic algorithms (GAs) are the main paradigm of evolutionary computing. GAs are inspired by Darwin's theory about evolution – the "survival of the fittest". In nature, competition among individuals for scanty resources results in the fittest individuals dominating over the weaker ones. GAs are adaptive heuristic search based on the evolutionary ideas of natural selection and genetics. GAs are intelligent exploitation of random search used in optimization problems. GAs, although randomized, exploit historical information to direct the search into the region of better performance within the search space [60].

6.2 Biological Background – Basic Genetics

- Every organism has a set of rules, describing how that organism is built. All living organisms consist of cells.
- In each cell there is same set of chromosomes. Chromosomes are strings of DNA and serve as a model for the whole organism.
- A chromosome consists of genes, blocks of DNA.
- Each gene encodes a particular protein that represents a trait (feature), e.g., color of eyes.
- Possible settings for a trait (e.g. blue, brown) are called alleles.
- Each gene has its own position in the chromosome called its locus.

- Complete set of genetic material (all chromosomes) is called a genome.
- Particular set of genes in a genome is called genotype.
- The physical expression of the genotype (the organism itself after birth) is called the phenotype, its physical and mental characteristics, such as eye color, intelligence etc.
- When two organisms mate they share their genes; the resultant offspring may end up having half the genes from one parent and half from the other. This process is called recombination (cross over).
- The new created offspring can then be mutated. Mutation means, that the elements of DNA are a bit changed. These changes are mainly caused by errors in copying genes from parents.
- The fitness of an organism is measured by success of the organism in its life (survival).

Below is the general scheme of evolutionary process in genetic along with pseudo-code: [61]

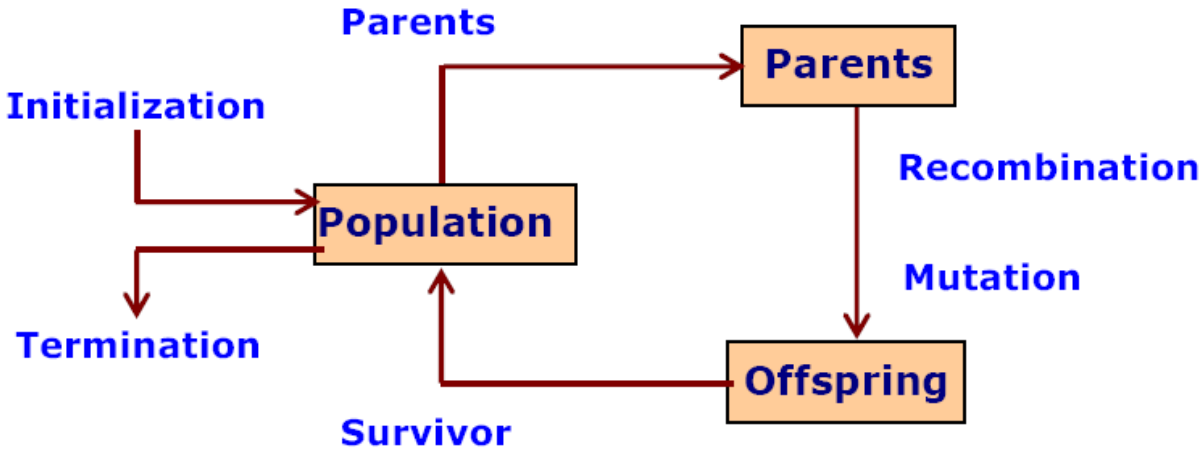

Fig. 6.2 General Scheme of Evolutionary process [61]

6.2.1 Pseudo-Code

- BEGIN
- INITIALISE population with random candidate solution.
- EVALUATE each candidate;
- REPEAT UNTIL (termination condition) is satisfied DO
    1. SELECT parents;
    2. RECOMBINE pairs of parents;
    3. MUTATE the resulting offspring;
    4. SELECT individuals or the next generation;
- END.

6.3 Outline of the basic genetic programming

Genetic algorithm begins with a set of solutions (represented by chromosomes) called the population. Solutions from one population are taken and used to form a new population. This is motivated by the possibility that the new population will be better than the old one. Solutions are selected according to their fitness to form new solutions (offspring); more suitable they are, more chances they have to reproduce. This is repeated until some condition (e.g. number of populations or improvement of the best solution) is satisfied.

1. [Start] Generate random population of **n** chromosomes (i.e. suitable solutions for the problem).

2. [Fitness] Evaluate the fitness **f(x)** of each chromosome **x** in the population.

3. [New population] Create a new population by repeating following steps until the new population is complete.

(a) [Selection] Select two parent chromosomes from a population according to their fitness (better the fitness, bigger the chance to be selected).

(b) [Crossover] With a crossover probability, cross over the parents to form new offspring (children). If no crossover was performed, offspring is the exact copy of parents.

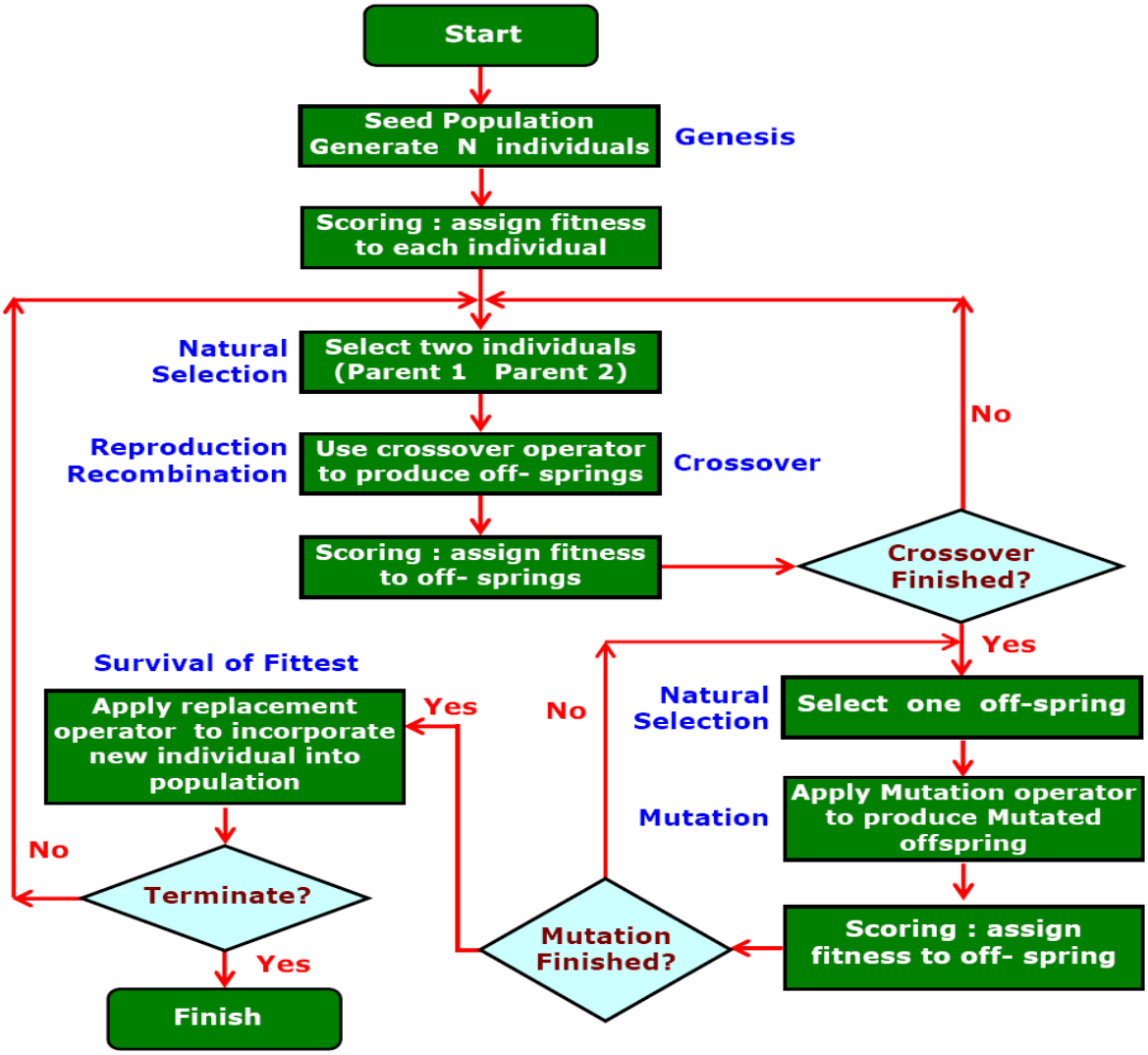

Fig 6.3 Genetic algorithm – program flow chart [62]

(c) [Mutation] With a mutation probability, mutate new offspring at each locus (position in chromosome).

(d) [Accepting] Place new offspring in the new population

The genetic algorithm's performance is largely influenced by two operators called crossover and mutation. These two operators are the most important parts of GA. [62]

6.4 Operators of Genetic Algorithm

Genetic operators used in genetic algorithms maintain genetic diversity. Genetic diversity or variation is a necessity for the process of evolution. Genetic operators are analogous to those which occur in the natural world [63] :

- Reproduction (or Selection) ;
- Crossover (or Recombination); and
- Mutation.

In addition to these operators, there are some parameters of GA. One important parameter is Population size. Population size says how many chromosomes are in population (in one generation). If there are only few chromosomes, then GA would have a few possibilities to perform crossover and only a small part of search space is explored. If there are many chromosomes, then GA slows down. Research shows that after some limit, it is not useful to increase population size, because it does not help in solving the problem faster. The population size depends on the type of encoding and the problem.

6.4.1 Reproduction, or Selection

Reproduction is usually the first operator applied on population. From the population, the chromosomes are selected to be parents to crossover and produce offspring. The problem is how to select these chromosomes? According to Darwin's evolution theory "survival of the fittest" – the best ones should survive and create new offspring. The Reproduction operators are also called Selection operators. Selection means extract a subset of genes from an existing population, according to any definition of quality. Every gene has a meaning, so one can derive from the gene a kind of quality measurement called fitness function. Following this quality (fitness value), selection can be performed. Fitness function quantifies the optimality of a solution (chromosome) so that a particular solution may be ranked against all the other solutions. The function depicts the closeness of a given 'solution' to the desired result. Many reproduction operators exists and they all essentially do same thing. They pick from current population the strings of above average and insert their multiple copies in the mating pool in a probabilistic manner. The most commonly used methods of selecting chromosomes for parents to crossover are:

- Roulette wheel selection
- Rank selection
- Boltzmann selection
- Steady state selection
- Tournament selection

The Roulette wheel and Boltzmann selections methods are illustrated next.

6.4.1.1 Roulette wheel selection (Fitness-Proportionate Selection)

Roulette-wheel selection, also known as Fitness Proportionate Selection, is a genetic operator, used for selecting potentially useful solutions for recombination. In fitness-proportionate selection [63]:

- the chance of an individual's being selected is proportional to its fitness, greater or less than its competitors' fitness.
- Conceptually, this can be thought as a game of Roulette.

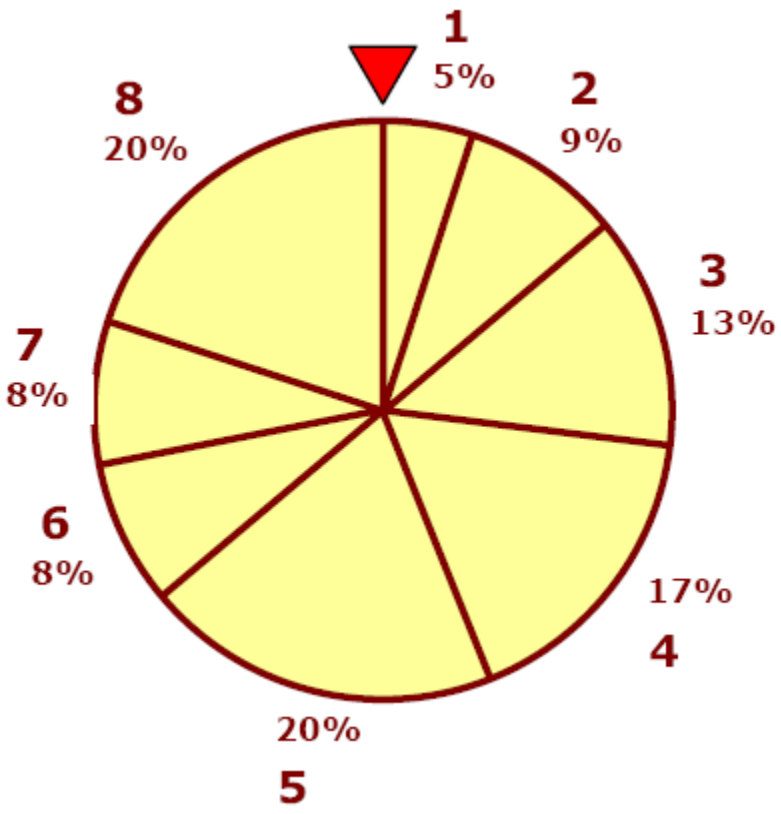

Fig. 6.4 Fig. Roulette-wheel Shows 8 individual with fitness [63]

The Roulette-wheel simulates 8 individuals with fitness values F_i, marked at its circumference; e.g.,

- the 5th individual has a higher fitness than others, so the wheel would choose the 5th individual more than other individuals .
- the fitness of the individuals is calculated as the wheel is spun n = 8 times, each time selecting an instance, of the string, chosen by the wheel pointer.

Probability of ith string is $P_i = F_i / (\quad)$, where n = no of individuals, called population size; P_i = probability of ith string being selected; F_i = fitness for ith string in the population. Because the circumference of the wheel is marked according to a string's fitness, the Roulette-wheel mechanism is expected to make F/F copies of the ith string. Average fitness = $\overline{F}$ F j / n; Expected count = (n =8) x P_i

Cumulative Probability_5 =

6.4.1.2 Boltzmann Selection

Simulated annealing is a method used to minimize or maximize a function [63].

- This method simulates the process of slow cooling of molten metal to achieve the minimum function value in a minimization problem.
- The cooling phenomena is simulated by controlling a temperature like parameter introduced with the concept of Boltzmann probability distribution.
- The system in thermal equilibrium at a temperature T has its energy distribution based on the probability defined by P(E) = exp (- E / kT) were k is Boltzmann constant.

- This expression suggests that a system at a higher temperature has almost uniform probability at any energy state, but at lower temperature it has a small probability of being at a higher energy state.
- Thus, by controlling the temperature T and assuming that the search process follows Boltzmann probability distribution, the convergence of the algorithm is controlled.

In this research, Roulette-wheel selection was used for simulation.

6.4.2 Crossover

Crossover is a genetic operator that combines (mates) two chromosomes (parents) to produce a new chromosome (offspring). The idea behind crossover is that the new chromosome may be better than both of the parents if it takes the best characteristics from each of the parents. Crossover occurs during evolution according to a user-definable crossover probability. Crossover selects genes from parent chromosomes and creates a new offspring. The Crossover operators are of many types [63].

- One simple way is, One-Point crossover.
- The others are Two Point, Uniform, Arithmetic, and Heuristic crossovers.

The operators are selected based on the way chromosomes are encoded.

6.4.2.1 One-Point Crossover

One-Point crossover operator randomly selects one crossover point and then copy everything before this point from the first parent and then everything after the crossover point copy from the second parent. The Crossover would then look as shown below. Consider the two parents selected for crossover.

Parent 1 **1 1 0 1 1 | 0 0 1 0 0 1 1 0 1 1 0**

Parent 2 **1 1 0 1 1 | 1 1 0 0 0 0 1 1 1 1 0**

Interchanging the parents chromosomes after the crossover points - The Offspring produced are:

Offspring 1 **1 1 0 1 1 | 1 1 0 0 0 0 1 1 1 1 0**

Offspring 2 **1 1 0 1 1 | 0 0 1 0 0 1 1 0 1 1 0**

The symbol, a vertical line, | is the chosen crossover point.

6.4.2.2 Two-Point Crossover

Two-Point crossover operator randomly selects two crossover points within a chromosome then interchanges the two parent chromosomes between these points to produce two new offspring. Consider the two parents selected for crossover:

Parent 1 **1 1 0 1 1 | 0 0 1 0 0 1 1 | 0 1 1 0**

Parent 2 **1 1 0 1 1 | 1 1 0 0 0 0 1 | 1 1 1 0**

Interchanging the parents chromosomes between the crossover points -

The Offspring produced are:

Offspring 1 **1 1 0 1 1 | 0 0 1 0 0 1 1 | 0 1 1 0**

Offspring 2 **1 1 0 1 1 | 0 0 1 0 0 1 1 | 0 1 1 0**

6.4.2.3 Uniform Crossover

Uniform crossover operator decides (with some probability – known as the mixing ratio) which parent will contribute how the gene values in the offspring chromosomes. The crossover operator allows the parent chromosomes to be mixed at the gene level rather than the segment level (as with one and two point crossover).

Consider the two parents selected for crossover.

Parent 1 **1 1 0 1 1 0 0 1 0 0 1 1 0 1 1 0**

Parent 2 **1 1 0 1 1 1 1 0 0 0 0 1 1 1 1 0**

If the mixing ratio is **0.5** approximately, then half of the genes in the offspring will come from parent **1** and other half will come from parent **2**. The possible set of offspring after uniform crossover would be:

Offspring 1 $\mathbf{1}_1\ \mathbf{1}_2\ \mathbf{0}_2\ \mathbf{1}_1\ \mathbf{1}_1\ \mathbf{1}_2\ \mathbf{1}_2\ \mathbf{0}_2\ \mathbf{0}_1\ \mathbf{0}_1\ \mathbf{0}_2\ \mathbf{1}_1\ \mathbf{1}_2\ \mathbf{1}_1\ \mathbf{1}_1\ \mathbf{0}_2$

Offspring 2 $\mathbf{1}_2\ \mathbf{1}_1\ \mathbf{0}_1\ \mathbf{1}_2\ \mathbf{1}_2\ \mathbf{0}_1\ \mathbf{0}_1\ \mathbf{1}_1\ \mathbf{0}_2\ \mathbf{0}_2\ \mathbf{1}_1\ \mathbf{1}_2\ \mathbf{0}_1\ \mathbf{1}_2\ \mathbf{1}_2\ \mathbf{0}_1$

The subscripts indicate which parent the gene came from.

6.4.2.4 Arithmetic Crossover

Arithmetic crossover operator linearly combines two parent chromosome vectors to produce two new offspring according to the equations:

Offspring1 = a * Parent1 + (1- a) * Parent2

Offspring2 = (1 – a) * Parent1 + a * Parent2

where **a** is a random weighting factor chosen before each crossover operation. Consider two parents (each of 4 float genes) selected for crossover:

Parent 1 (0.3) (1.4) (0.2) (7.4)

Parent 2 (0.5) (4.5) (0.1) (5.6)

Applying the above two equations and assuming the weighting factor **a = 0.7**, applying above equations, we get two resulting offspring. The possible set of offspring after arithmetic crossover would be:

Offspring 1 **(0.36) (2.33) (0.17) (6.87)**

Offspring 2 **(0.402) (2.981) (0.149) (5.842)**

6.4.2.5 Heuristic Crossover

Heuristic crossover operator uses the fitness values of the two parent chromosomes to determine the direction of the search. The offspring are created according to the equations:

Offspring1 = BestParent + r * (BestParent − WorstParent)

Offspring2 = BestParent

where **r** is a random number between **0** and **1**. It is possible that **offspring1** will not be feasible. It can happen if **r** is chosen such that one or more of its genes fall outside of the allowable upper or lower bounds. For this reason, heuristic crossover has a user defined parameter **n** for the number of times to try and find an **r** that result in a feasible chromosome. If a feasible chromosome is not produced after **n** tries, the worst parent is returned as offspring1.

6.4.3 Mutation

After a crossover is performed, mutation takes place. Mutation is a genetic operator used to maintain genetic diversity from one generation of a population of chromosomes to the next. Mutation occurs during evolution according to a user-definable mutation probability, usually set to fairly low value, say **0.01** a good first choice. Mutation alters one or more gene values in a chromosome from its initial state. This can result in entirely new gene values being added to the gene pool. With the new gene values, the genetic algorithm may be able to arrive at better solution than was previously possible. Mutation is an important part of the genetic search, helps to prevent the population from stagnating at any local optima. Mutation is intended to prevent the search falling into a local optimum of the state space. The Mutation operators are of many types

[63]. One simple way is, **Flip Bit**. the others are **Boundary, Non-Uniform, Uniform, and Gaussian**. The operators are selected based on the way chromosomes are encoded.

6.4.3.1 Flip Bit Mutation

The mutation operator simply inverts the value of the chosen gene. i.e. **0** goes to **1** and **1** goes to **0**. This mutation operator can only be used for binary genes. Consider the two original off-springs selected for mutation.

Original offspring 1 1 1 0 1 1 1 1 0 0 0 0 1 1 1 1 0

Original offspring 2 1 1 0 1 1 0 0 1 0 0 1 1 0 1 1 0

Invert the value of the chosen gene as **0** to **1** and **1** to **0.** The Mutated Off-spring produced are:

Mutated offspring 1 1 1 0 0 1 1 1 0 0 0 0 1 1 1 1 0

Mutated offspring 2 1 1 0 1 1 0 1 1 0 0 1 1 0 1 0 0

6.4.3.2 Boundary Mutation

The mutation operator replaces the value of the chosen gene with either the upper or lower bound for that gene (chosen randomly). This mutation operator can only be used for integer and float genes.

6.4.3.3 Non-Uniform

The mutation operator increases the probability such that the amount of the mutation will be close to **0** as the generation number increases. This mutation operator prevents the population from stagnating in the early stages of the evolution then allows the genetic algorithm to fine tune

the solution in the later stages of evolution. This mutation operator can only be used for integer and float genes.

6.4.3.4 Uniform

The mutation operator replaces the value of the chosen gene with a uniform random value selected between the user-specified upper and lower bounds for that gene. This mutation operator can only be used for integer and float genes.

6.4.3.5 Gaussian

The mutation operator adds a unit Gaussian distributed random value to the chosen gene. The new gene value is clipped if it falls outside of the user-specified lower or upper bounds for that gene. This mutation operator can only be used for integer and float genes.

CHAPTER 7

PERFORMANCE ANALYSIS OF GENETICALLY TUNED PI CONTROLLER

In chapter 6, theory of basic genetic algorithm has been discussed elaborately. Following are the basic steps of operation in short

1. **[Start]** Generate random population of *n* chromosomes (suitable solutions for the problem)
2. **[Fitness]** Evaluate the fitness *f(x)* of each chromosome *x* in the population
3. **[New population]** Create a new population by repeating following steps until the new population is complete
    1. **[Selection]** Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)
    2. **[Crossover]** With a crossover probability cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.
    3. **[Mutation]** With a mutation probability mutate new offspring at each locus (position in chromosome).
    4. **[Accepting]** Place new offspring in a new population
4. **[Replace]** Use new generated population for a further run of algorithm
5. **[Test]** If the end condition is satisfied, **stop**, and return the best solution in current population
6. **[Loop]** Go to step **2**

In this research, roulette wheel selection was used for population selection. And also arithmetic crossover and uniform mutation techniques were applied. A Genetic PI controller for the SRM

drives is shown in Fig. 7.1. The GA uses the principles of evolution and genetics to select and adapt the controller parameters (*Kp* and *Ki*). The controller parameters are coded by decimal numbers in chromosome. The candidate controllers of the Genetic PI controller are defined as members of the population. During time step, each member of the population is evaluated on how well it minimizes the ITAE.

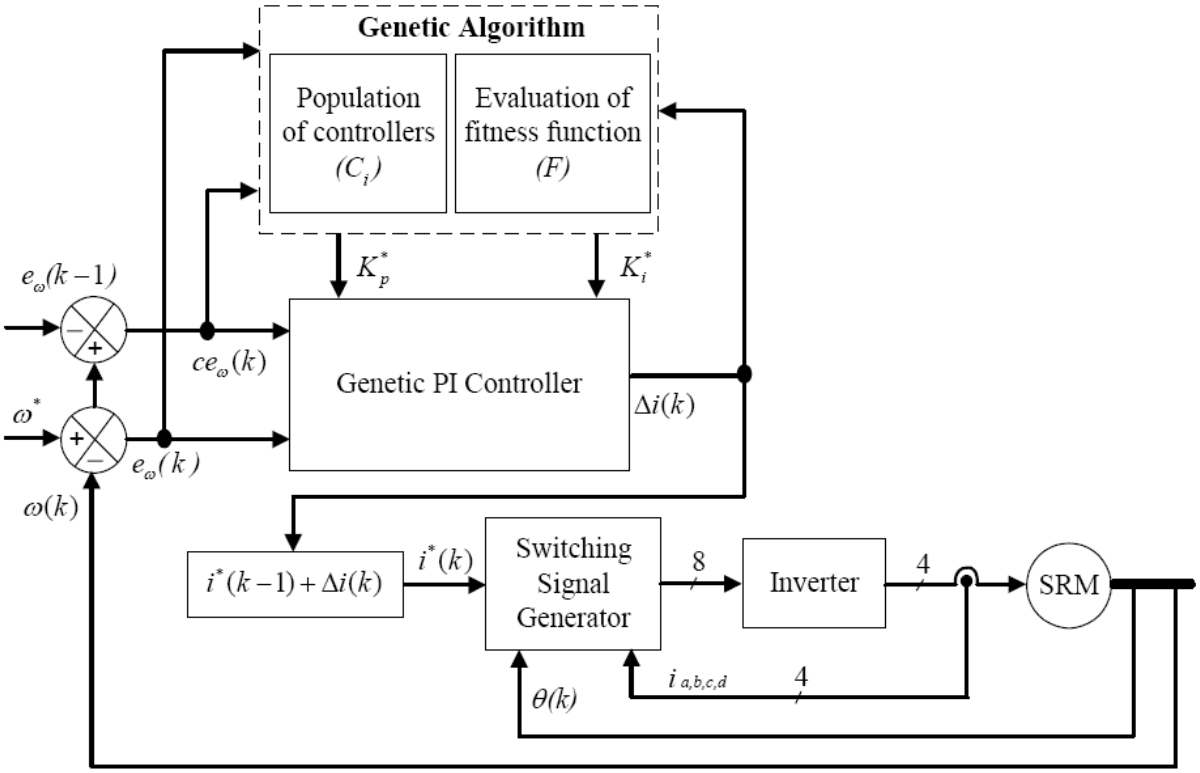

Fig. 7.1 Genetically tuned PI controller for the SRM drives.

For each member of the population, the GA computes the speed error (*e*ω) and change in the speed error (*ceω*). The output variable of controller is change in the reference current ($\Delta i(k)$). The ω *e* and ω *ce* are defined as:

$$e_\omega(k) = \omega^* - \omega(k) \tag{7.1}$$

$$ce_\omega(k) = e_\omega(k) - e_\omega(k-1) \tag{7.2}$$

where ω* is the reference speed. Also (k) and (k-1) denote actual and previous values, respectively.

The steps for speed control are summarized as follows:

a) Sample the speed signal of the SRM

b) Calculate the speed error and change in speed error.

c) Chose the number of digits to represent each controller parameter *Kp* and *Ki*. Chose crossover probability (*pc)* and mutation probability (*pm*).

d) Generate an initial population of *Kp* and *Ki* gains (we make a random selection) Initialize sample time *T* and set time *t.*

e) Generate Δ*i(k),* for each population member *Ci, i=1,2,...n* using the PI control laws.

$$\Delta i(k) = K_p.ce_\omega(k) + K_i.e_\omega(k).T \tag{7.3}$$

f) Assign fitness to each element of the population *Ci, i=1,2,3,....n,*

$$p_1 = e_\omega(k) \tag{7.4}$$

$$p_2 = ce_\omega(k) \tag{7.5}$$

$$p_3 = \Delta i(k) \tag{7.6}$$

$$F = \frac{1}{(\alpha_1.p_1^2 + \alpha_2.p_2^2 + \alpha_3.p_3^2)} \tag{7.7}$$

g) Produce the next generation using GA operators and let $t=t+T$ go to step (d)

h) The maximally fit *Ci* becomes *C** and send the change of control action (*i*(k)*) to control the drive. Where *i*(k)* is the inferred change of reference current by the controller at the *k*th sampling time and defined as

$$i^*(k) = i^*(k-1) + \Delta i(k) \tag{7.8}$$

where, i^* (k-1) is the previous reference current.

7.1 Simulation results

Only current control loop with GA tuned PI controller

By using genetic algorithm, value of K_c and T_c were tuned, where $K_c = K_p$ (Proportional gain) and $K_c/T_c = K_i$ (Integral gain)

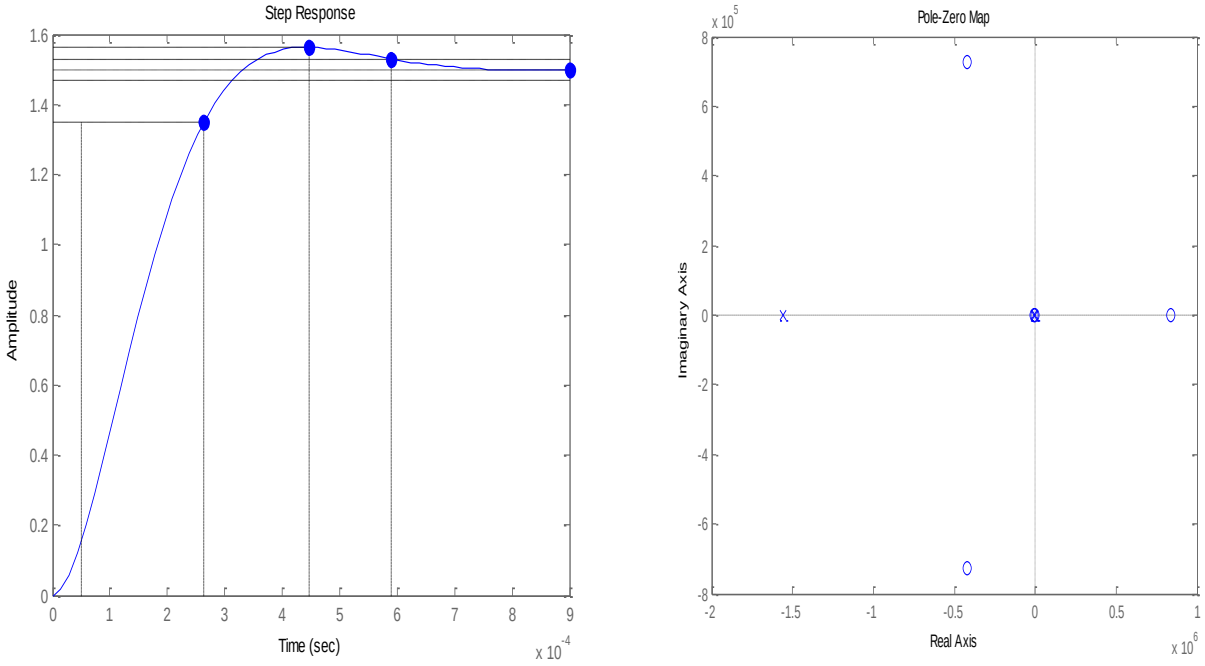

Fig.7.2 (a) Step response, (b) pole zero mapping of current loop with GA tuned PI controller

Rise Time = 0.000214 sec

Settling time = 0.00059 sec

Overshoot = 4.24 %

Eigen value of A matrix = -0.2, -7124.3+7108i, -7124.3+7108i (3^{rd} Order system). From the result, we see that the rise time and settling time of the system is improved compared to the

result obtained in chapter 5 with only PI controller. Also, poles are shifted towards left compared to pole zero mapping of current loop with only PI controller, which increases the stability

Approximated Speed control loop only with GA tuned PI controller

By using genetic algorithm, value of K_s and T_s were tuned, where $K_s = K_p$ (Proportional gain) and $K_s/T_s = K_i$ (Integral gain)

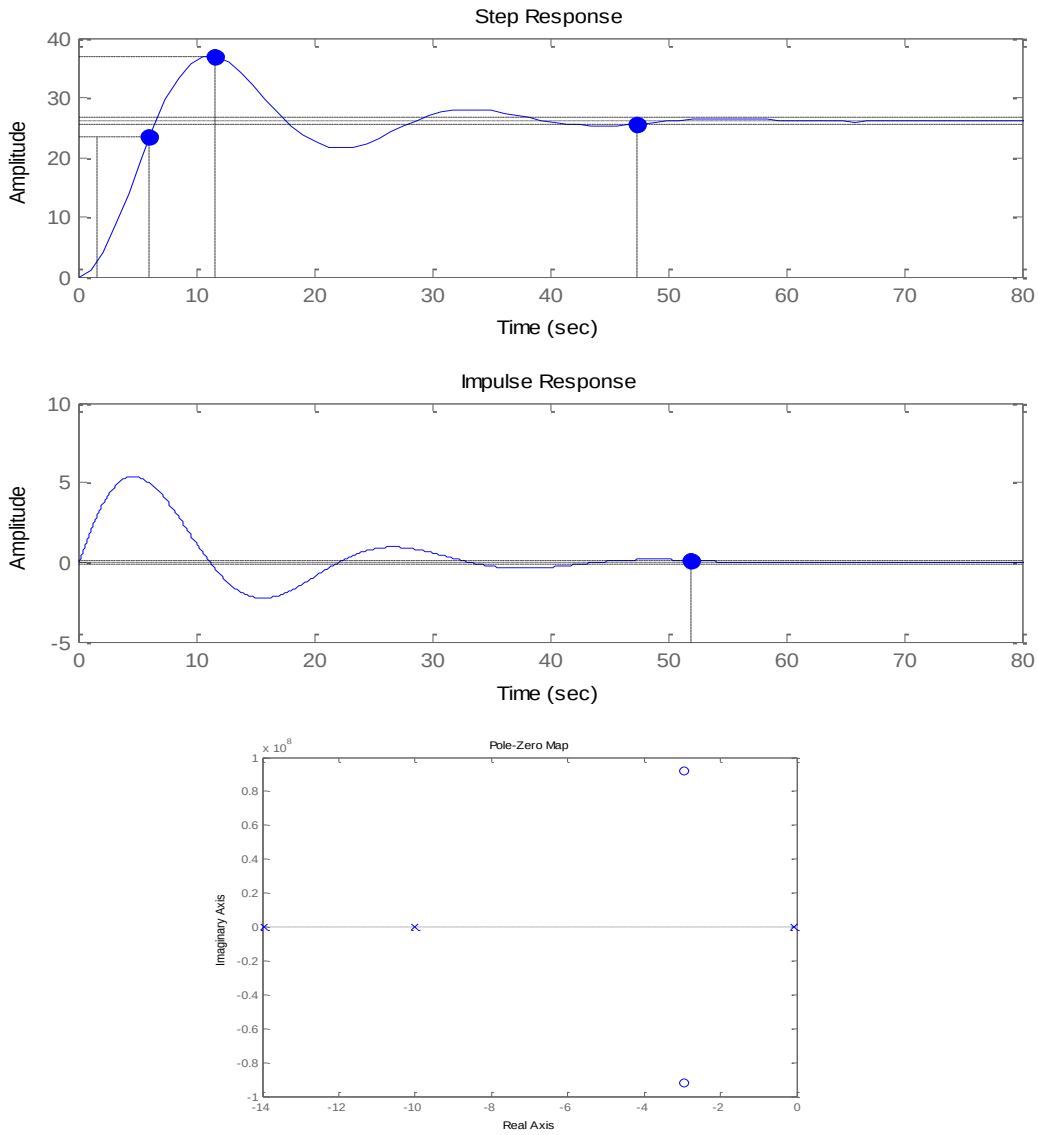

Fig. 7.3 Step and impulse response and pole zero map of speed loop with GA tuned PI controller

Rise time = 4.42 sec

Settling time = 47.4 sec

Overshoot = 41.4%

From the result, we see that the rise time and settling time of the system is improved compared to the result obtained in chapter 5 with only PI controller. Also, the poles are shifted towards left compared to pole zero mapping of speed loop with only PI controller, which increases stability.

Overall SRM drive with GA tuned PI controller

By using genetic algorithm, value of K_s and T_s were tuned, where $K_s = K_p$ (Proportional gain) and $K_s/T_s = K_i$ (Integral gain). Value of K_c and T_c were given manually following equations 5.19 and 5.20.

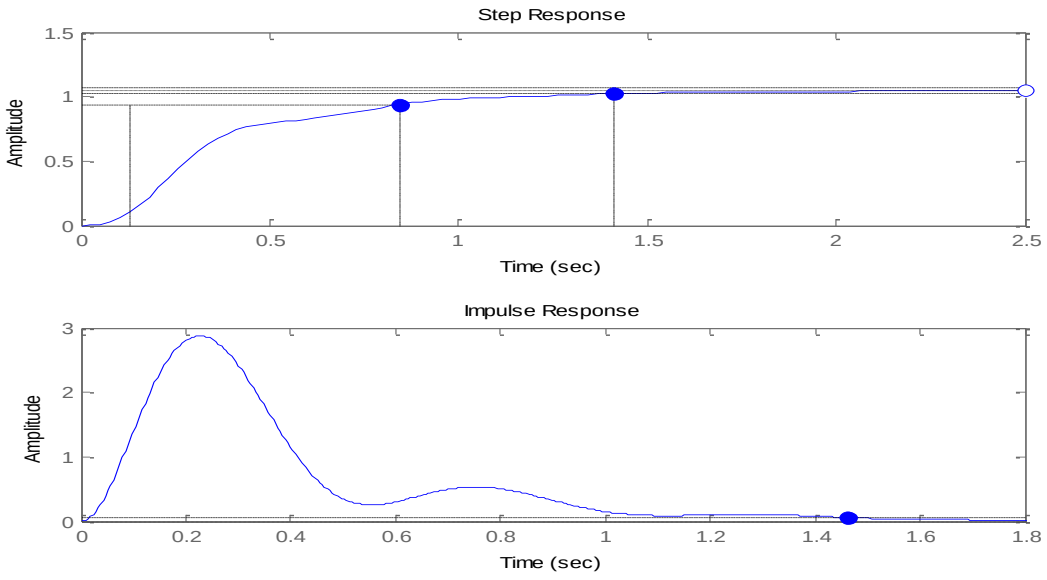

Fig. 7.4 Step and impulse response of SRM drive with GA tuned PI controller

Rise time = 0.717 sec, Settling time = 1.41 sec, Overshoot = 2.22e-14 % (over damped). From the result, we see that the rise time and settling time of the system is improved compared to the result obtained in chapter 5 with only PI controller.

CHAPTER 8

CONCLUSION

8.1 Summary and result comparison

Fast response and quick recovery from load disturbances and insensitivity to parameter variations are some of the principal criteria in designing and implementing a high performance variable speed electric motor drive system. Review of relevant literature enables us to conclude that the Switched Reluctance Motor (SRM) with a suitable speed controller has the potential to fulfill the required criteria of high performance motor drive system. The researchers and scientists in the field of electrical machines have significant researches to introduce the switched reluctance motor (SRM) to the electrical machines family with affordable cost. These machines are eliminating rotor losses due to the absence of rotor conductors.

Conventional PI controller based motor drive systems need accurate mathematical models to describe the system dynamics. Sophisticated system models incorporating unavoidable conditions such as saturation, disturbances, parameter drifts and temperature variations are often unavailable in the real world. Thus the performance of the PI controller based drive system, whose constants are not optimized are unpredictable under abnormal operating conditions. Furthermore, uncertainty and non-linearity from the motor mechanical load sometimes cause the drive system to become unstable in the absence of proper control. Hence, an adaptive controller is essential in a high performance SRM drive system. As an intelligent control technology, the genetic algorithm (GA) can give robust adaptive response of a drive with nonlinearity, parameter variation and load disturbance effect. In this thesis, the Genetic PI speed controller was applied to the speed loop by replacing the conventional PI speed controller. The Genetically tuned PI

controller was simulated in a MATLAB environment. The results show that Genetic PI control is suitable for control of the SRM drive systems consisting uncertainties and nonlinearities and gives smooth dynamic response. The Genetic PI gives reduced rise time as well as small overshoot with or without load and also deferent reference speeds.

In chapter 1, an extensive literature survey was carried out on variable speed switched reluctance motor drives have been carried out. Problems involving the precise speed control of the switched reluctance motors have been identified and a solution using genetically tuned PI controller has been proposed.

In chapter 2, working principle of SRM has been elaborated. As an integral part of the control structure, the mathematical model is derived from the dynamic behavior and governing equations.

In chapter 3, a step by step process for creating a MATLAB model has been described. The coding procedure of all different characteristics profile has been elaborated and also, the simulation result has been demonstrated. From the simulation result, it was shown that, all different characteristics curve of motor is exactly matching with the simulation result.

Chapter 4 presents small signal analysis of single phase linearized SRM. All the results are extracted from MATLAB simulation. A small perturbation was given around an operating point to linearize the system model so that linear control techniques can be exploited to see the effect of genetic PI controller. From this model simulation, it was shown that, without control drive is very much sensitive to parameter variation.

Chapter 5 describes the performance of current control loop and speed control loop and the overall SRM drive block including both current loop and speed loop inside of a single phase

SRM using PI controller in terms of rise time, settling time and overshoot of the step and impulse response.

Chapter 6 elaborates the theory of genetic algorithm. The basic concept of genetic programming, steps associated with genetic algorithm and operators of genetic algorithm has been shown with example in details. This chapter generates a clear idea about how to use genetic algorithm to tune the constants of PI controller.

Chapter 7 describes the performance of current loop, speed loop and overall block using a genetically tuned PI controller and comparison with the performance of only PI controller. From the result, it is observed that, settling time and rise time of individual current control loop and speed control loop and also the overall drive system block improves after the constants K_p and K_i are tuned using the genetic algorithm. This result clearly demonstrates the efficiency of genetically tuned PI controller over normal PI controller.

The table shown below summarizes the result and shows the comparison between normal PI controller and GA tuned PI controller.

Table 8.1 Result Comparison

System	Rise time (sec)		Settling time (sec)	
	GA tuned PI controller	Only PI controller	GA tuned PI controller	Only PI controller
Speed control loop	4.42	58.8	47.4	107
Current control loop	0.000214	.0524	0.00059	0.0939
Overall SRM drive block	0.717	1.02	1.41	1.94

From the above table, it is understood very clearly that, genetically tuned PI controller is improving the performance in terms of rise time and settling time of all the individual loops and also overall block of the SRM drive. In all the three cases, genetically tuned PI controller gives early rising time and also early steady state in the step response of the system.

8.2 Contribution of the thesis

Contributions of this thesis are pointed below-

1. Creating a MATLAB model for linear inductance profile multiphase SRM.
2. Simulation of the genetically tuned PI controller and proof of performance improvement of PI controller when proportional gain and integral gain are tuned by genetic algorithm

8.3 Future work

Performance of the GA tuned PI on SRM can be observed incorporating the mutual coupling effect of the phases, which was ignored in this thesis to just observe the basic impact of GA tuned PI controller on SRM behavior. Moreover, this technique can be applied to non linear SRM to get more accurate picture of the dynamic response of SRM due to the impact of GA tuned PI controller. In this thesis, as genetic algorithm operator, Roulette wheel selection, arithmetic cross over and uniform mutation criteria was used. There are many other types of GA operator. The impact of other operators may be observed in future work. And in this thesis, the impact of GA tuned PI controller on single phase SRM has been discussed only. In future research, this impact can be extended to 6/4, 8/6 or 12/10 SRM.

Reference

[1] S. A. Nasar, “Electromagnetic Energy Conversion Devices and Systems”. Englewood Cliffs, Prentice-Hall, 1970.

[2] S.A. Nasar, “DC Switched Reluctance Motor", Proceedings of the Institution of Electrical Engineers, vol.166, no.6, June, 1996.

[3] Miller TJE : Brushless Permanent-Magnet and Reluctance Motor Drives, Clarendon Press (Oxford), 1989

[4] Cheok AD, Fukuda Y : A New Torque and Flux Control Method for Switched Reluctance Motor Drives, IEEE Transactions on Power Electronics, Vol. 17, no. 4 July 2002

[5] Staton DA, Soong WL, Miller TJE : Unified Theory of Torque Production in Switched Reluctance and Synchronous Reluctance Motors, IEEE

[6] P. French and A.H. Williams, “A New Electric Propulsion Motor", Proceedings of AIAA, Third Propulsion Joint Specialist Conference, Washington, July, 1967.

[7] L.E. Unnewehr and H.W. Koch, “An Axial Air-Gap Reluctance Motor for Variable Speed Application", IEEE Transactions on Power Apparatus and Systems, vol. PAS- 93, no.1, January, 1974.

[8] P.J. Lawrenson, J.M. Stephenson, P.T. Blenkinson, J. Corda and N.N. Fulton, “Variable-Speed Switched Reluctance Motors”, IEE Proc., vol. 127, July 1980.

[9] P.J. Lawrenson, “Switched Reluctance Motor Drives", Electronics and Power Conference, 1983.

[10] T.E.J. Miller, and McGilp," Non-Linear Theory of the Switched Reluctance Motor for Rapid Computer-Aided Design", IEE Proceedings B 137, 1990.

[11] R. Colby, F. Mottier, and T. Miller, “Vibration Nodes and Acoustic Noise in a 4- Phase Switched Reluctance Motor”, IEEE Trans. on Industry Applications 32, 1996.

[12] J. Faiz, and J.W. Finch, “Aspects of Design Optimisation for Switched Reluctance Motors”, IEEE Trans. on Energy Conversion, 1993.

[13] D.A. Torrey, X-M. Niu, and E.J. Unkauf, “Analytical Modelling of Variable- Reluctance Machine Magnetisation Characteristics”, IEE Proceedings Electric Power Application, 1995.

[14] A.V. Radun, “Design Considerations for the Switched Reluctance Motor”, IEEE Trans. on Industry Applications 31, 1995.

[15] S. Brisset, and P. Brochet, “Optimisation of Switched Reluctance Motors using Deterministic Methods with Static and Dynamic Finite Element Simulations”, IEEE Trans. on Magnetics 34, 1995.

[16] I. Boldea and S.A. Nasar, “Electric Drives”, Ch. 12, CRC Press, 1999.

[17] C. Roux, and M.M. Morcos, “On the Use of a Simplified Model for Switched Reluctance Motors”, IEEE Trans. on Energy Conversion 17, 2002.

[18] P. J. Lawrenson, "Switched Reluctance Drives - A Fast Growing Technology", Electric Drives and Controls, April/May 1985.

[19] W.F. Ray and R.M. Davis, " Inverter Drive for Doubly Salient Reluctance Motor: Its Fundamental Behaviour, Linear Analysis and Cost Implications", Electric Power Applications, vol. 2, no. 6, December 1979.

[20] H. Klode, A. Omekanda, A. Khalil, S. Underwood, I. Husain "The Potential of Switched Reluctance Motor Technology for Electro-Mechanical Brake Applications", SAE International Simulation & Modelling Mechatronics, April, 2006.

[21] W.F. Ray, P.J. Lawrenson, R.M. Davis, J.M. Stephenson, N.N. Fulton and R.J. Blake, "High Performance Switched Reluctance Brush Less Drives", in Conf. Rec. of IEEE Ind. Appl. Soc. Annual Meeting, 1985.

[22] P. J. Lawrenson, J. M. Stephenson, P. T. Blenkinsop, J. Corda, and N. N. Fulton, "Variable-speed switched reluctance motor," *Proc. Inst. Elect. Eng.*, vol. 127, pp. 253–265, Jul. 1980.

[23] N. N. Fulton and J. M. Stephenson, "A review of switched reluctance machine design," in *Proc. IEE Electric Machines Conf.*, 1988, pp. 120–126.

[24] W. F. Ray, R.M.Davis, P. J. Lawrenson, J. M. Stepenson, N. N. Fulton, and R. J. Blake, "Switched reluctance motor drives for rail traction: A second view," *Proc. Inst. Elect. Eng.*, vol. 131, no. 5, pt. B, pp. 220–264, Sep. 1984.

[25] M. R. Harris, J. W. Finch, J. A. Mallick, and T. J. E. Miller, "A review of the integral-horsepower switched reluctance motor drive," *IEEE Trans. Ind. Appl.*, vol. IA-22, no. 4, pp. 716–721, Jul./Aug. 1986.

[26] T. Higuchi, J. O. Fiedler, and R. W. D. Doncker, "On the design of a single-phase reluctance motor," in *Proc. IEEE Electric Machines and Drives Conf.*, 2003, pp. 561–567.

[27] M. V. K. Chari, G. Bedrosian, J. D. Angelo, and A. Konrad, “Finite element applications in electrical engineering,” *IEEE Trans. Magn.*, vol. 29, no. 2, pp. 1306–1314, Mar. 1993.

[28] R. Arumugam, D. A. Lowther, R. Krishnan, and J. F. Lindsay, “Magnetic field analysis of a switched reluctance motor using a two dimensional finite element model,” *IEEE Trans. Magn.*, vol. MAG-21, no. 5, pp. 1883–1885, Sep. 1985.

[29] D. S. Reay, M. M. Moud, T. C. Green, and B. W.Williams, “Switched reluctance motor control via fuzzy adaptive systems,” in *Proc. IEEE Control Systems Conf.*, Jun. 1995, pp. 527 531.

[30] S. Bolognani and M. Zigliotto, “Fuzzy logic control of a switched reluctance motor drive,” *IEEE Trans. Ind. Appl.*, vol. 32, no. 5, pp. 1063–1068, Sep./Oct. 1996.

[31] S. Bolognani and M. Zigliotto, “Fuzzy logic control of a switched reluctance motor drive,” *IEEE Trans. Ind. Appl.*, vol. 32, no. 5, pp. 1063–1068, Sep./Oct. 1996.

[32] M. G. Rodrigues, W. I. Suemitsu, P. Branco, J. A. Dente, and L. G. B. Rolim, “Fuzzy logic control of a switched reluctance motor,” in *Proc. IEEE ISIE*, Guimaraes, Portugal, 1997, pp. 527–531.

[33] P. C. Kjaer, J. J. Gribble, and T. J. E. Miller, “High-grade control of switched reluctance machines,” *IEEE Trans. Ind. Appl.*, vol. 33, no. 6, pp. 1585–1593, Nov./Dec. 1997.

[34] S. Mir, M. E. Elbuluk, and I. Husain, “Torque-ripple minimization in switched reluctance motors using adaptive fuzzy control,” *IEEE Trans. Ind. Appl.*, vol. 35, no. 2, pp. 461–468, Mar./Apr. 1999.

[35] K. Russa, I. Husain, and M. E. Elbuluk, "A self-tuning controller for switched reluctance motors," *IEEE Trans. Power Electron.*, vol. 15, no. 3, pp. 545–552, May 2000.

[36] S. Mir, M. S. Islam, T. Sebastian, and I. Husain, "Fault-tolerant switched reluctance motor drive using adaptive fuzzy logic controller," *IEEE Trans. Power Electron.*, vol. 19, no. 2, pp. 289–295, Mar. 2004.

[37] S. Paramasivam and R. Arumugam, "Real time hybrid controller implementation for switched reluctance motor drive," *Amer. J. Appl. Sci.*, vol. 1, no. 4, pp. 284–294, 2004.

[38] D.S.Reay. T.C Green and B.W.williams,"Neural Network used for torque ripple minimization for SRM", EPE,Briton,Sep 1993,pp 1-5.

[39] R. Krishnan "Switched Reluctance Motor Drives: Modelling, Simulation, Analysis, Design, and Applications", CRC Press, 2001.

[40] F. Soares, P.J. Costa Branco "Simulation of a 6/4 Switched Reluctance Motor Based on Matlab/Simulink Environment, *IEEE Trans. On Aerospace and Electronic Systems vol. 37, no. 3 July 2001*"

[41] M. T. DiRenzo "Switched Reluctance Motor Control – Basic Operation and Example Using the TMS320F240", Application Report SPRA420A – February 2000.

[42] I. Husain and M. Ehsani," Torque Ripple Minimization in Switched Reluctance Motor Drives by PWM Current Control", Proc. APEC'94, 1994.

[43] P. C. Kjaer, J. Gribble, and T. J. E. Miller, "High-grade Control of Switched Reluctance Machines", IEEE Trans. Industry Electronics, Vol. 33, Nov. 1997.

[44] J. Stephenson, A. Hughes, and R. Mann, “Online Torque-Ripple Minimisation in a Switched Reluctance Motor over a Wide Speed Range”, IEE Proceedings Electric Power Application 149, 2002.

[45] D. Cameron, J. Lang, and S. Umans, "The Origin and Reduction of Acoustic Noise in Doubly Salient Variable-Reluctance Motors", IEEE Trans. on Industry Applications 28, 1992.

[46] C. Wu and C. Pollock, "Analysis and Reduction of Vibration and Acoustic Noise in the Switched Reluctance Drive", IEEE Trans. on Industry Applications 31, 1995.

[47] W. Wu, J. B. Dunlop, S. J. Collocott and B. A. Kalati, “Design Optimisation of a Switched Reluctance Motor by Electromagnetic and Thermal Finite Element Analysis”, IEEE Transactions on Magnetics, USA, March 30 - April 3, 2003.

[48] R. De Doncker, J. Fiedler, N. Fuengwarodsakul, S. Bauer, C. Carstensen “State-ofthe- Art of Switched Reluctance Drives for Hybrid and Electric Vehicles”, in Proc. 5th International Power Electronics Conference IPEC05, Nigata, Japan, April 2005.

[49] V. Török, K. Loreth, “The World's Simplest Motor for Variable Speed Control, the Cyrano Motor, a PM-Biased SR-Motor of High Torque Density”, Proceedings of EPE '93, 5th European Conference on Power Electronics and Applications, IEE Conference Publication No. 377, Vol. 6, September, UK, 1993.

[50] N. Fuengwarodsakul, J. Fiedler, S. Bauer, R. De Doncker “New Methodology in Sizing and Predesign of Switched Reluctance Machines Using Normalized Flux- Linkage Diagram”, in Proc. 40th IEEE-IAS Annual Meeting IAS05, Hong Kong, Oct 2005.

[51] J. Smart, “Single-Phase Variable Reluctance Motor Having Permanent Magnets Embedded within a Phase Winding”, U.S. patent 5.650.682, 1997.

[52] Elliot, C. R., Stephenson, J. M., and McClelland, M. L. (1995) Advances in switched reluctance drive system dynamic simulation, *Proceedings of EPE ’95*, **3** (1995), 622—626.

[53] Franceschini, G., Pirani, S., Rinaldi, M., and Tassoni, C. (1991) SPICE assisted simulation of controlled electric drives: An application to switched reluctance drives. *IEEE Transactions on Industry Applications*, **27**, 6 (Nov./Dec. 1991), 1103—1110.

[54] Ichinokura, O., Onda, T., Kimura, M., Watanabe, T., Yanada, T., and Guo, H. J. (1998) Analysis of dynamic characteristics of switched reluctance motor based on SPICE. *IEEE Transactions on Magnetics*, **34**, 4 (1998), 2147—2149.

[55] Skvarenina, Wasynczuk, Krause (1996) Simulation of a switched reluctance generator/more electric aircraft power system. In *Proceedings of 1996 IECEC*, paper 96398.

[56] Radun, X. (1995) Switched reluctance starter/generator modeling results. In *Proceedings of SAE Aerospace Atlantic Conference*, 1995, paper 951407.

[57] Pekarek, S. D., Wasynczuk, O., and Hegner, H. J. (1998) An efficient and accurate model for the simulation and analysis of synchronous machine/converter systems. *IEEE Transactions on Energy Conversion* (Mar. 1998), 42—49.

[58] Ong, C-M. (1998) *Dynamic Simulation of Electric Machinery using Matlab/Simulink*. Englewood Cliffs, NJ: Prentice-Hall, 1998.

[59] Jackson, T.W., "*Design and development of a low-cost controller for SRM drive, M.S. Thesis*", The Bradley Department of Electrical and Computer Engineering, Virginia Tech., Blacksburg, VA, July 1996.

[60] Haupt, R.L., Haupt,S.E., "*Practical Genetic Algorithms*",A.John. Wiley & Sons Publication, 1998

[61] Mitchell Melanie, "*An Introduction to Genetic Algorithms*", MIT press, 1998.

[62] Gen, M., Cheng, R., "Genetic Algorithms and Engineering Optimization", John Wiley & Sons, Inc, 2000.

[63] Reeves, C.R., Rowe, J.E.,"Genetic Algorithms - Principles and Perspectives, A guide to GA theory", Kluwer Academic publishers, 2002.

Appendix A

MATLAB code for linear inductance based multiphase SRM

```
clc;clear all;
close all;
NS=6
NR=4
p=3;
BETAS=30*(pi/180);
BETAR=30*(pi/180);
TETAS=(2*pi)*((1/NR)-(1/NS))
TETAX=(pi/NR)-((BETAR+BETAS)/2)
TETAY=(pi/NR)-((BETAR-BETAS)/2)
TETAZ=(BETAR-BETAS)/2
TETAXY=(TETAY+TETAZ+TETAS)
TETAON=0.1*(pi/180)
TETAOFF=30*(pi/180)
TETAQ=60*(pi/180)
TETAIN=20.1*(pi/180)            % not needed
V=150
TL=0;
W=0.0;
ts=0.000065;
% theta=16*pi/180;
% th_rem1=rem(theta,pi/2);
% th_rem2=rem(theta+pi/6,pi/2);
% th_rem3=rem(theta+pi/3,pi/2);
R=1.30;
J=0.0013;
F=0.0183;
DELTAI=0.2;
DELTAVMIN=0;
DELTAVMAX=150;
LMIN=8e-3;
LMAX=60e-3;
% Program below computes from the giving minimum and maximum inductance
% values, the equations of the linear inductance profile for the increasing
% and decreasing part

G=(inv([TETAX 1;TETAY 1]))*([LMIN;LMAX]);
AUP=G(1);           %used by the program 1.m
BUP=G(2);           %used by the program 1.m
H=(inv([(TETAY+TETAZ) 1;TETAXY 1]))*([LMAX;LMIN]);
ADOWN=H(1);
BDOWN=H(2);
DL=AUP;
flux1=0;
flux2=0;
flux3=0;
Va1=V;
Va2=V;
```

```
Va3=V;
I1=0;
I2=0;
I3=0;
tsim=0.5;
t=zeros(1,tsim/ts);
theta=0:pi/314.6:pi/2;

for i=1:tsim/ts

    th_rem1(i)=rem(theta(i),pi/2);
    th_rem2(i)=rem(theta(i)+pi/6,pi/2);
    th_rem3(i)=rem(theta(i)+pi/3,pi/2);

if (0<=th_rem1(i)&(th_rem1(i)<=TETAX))
    L1(i)=LMIN;
end;

if (TETAX<th_rem1(i)&(th_rem1(i)<=TETAY))
    L1(i)=(AUP*th_rem1(i)+BUP);
end;

if ((TETAY<th_rem1(i))&(th_rem1(i)<=TETAXY))
    L1(i)=((ADOWN*th_rem1(i))+BDOWN);
end;

if (th_rem1(i)>TETAXY)
    L1(i)=LMIN;
end;

if (0<=th_rem2(i)&(th_rem2(i)<=TETAX))
    L2(i)=LMIN;
end;

if (TETAX<th_rem2(i)&(th_rem2(i)<=TETAY))
    L2(i)=(AUP*th_rem2(i)+BUP);
end;

if ((TETAY<th_rem2(i))&(th_rem2(i)<=TETAXY))
    L2(i)=((ADOWN*th_rem2(i))+BDOWN);
end;

if (th_rem2(i)>TETAXY)
    L2(i)=LMIN;
end;

if (0<=th_rem3(i)&(th_rem3(i)<=TETAX))
    L3(i)=LMIN;
end;

if (TETAX<th_rem3(i)&(th_rem3(i)<=TETAY))
    L3(i)=(AUP*th_rem3(i)+BUP);
end;
```

```
if ((TETAY<th_rem3(i))&(th_rem3(i)<=TETAXY))
    L3(i)=((ADOWN*th_rem3(i))+BDOWN);
end;

if (th_rem3(i)>TETAXY)
    L3(i)=LMIN;
end;
if (TETAON<=th_rem1(i)&(th_rem1(i)<=TETAOFF))
    Va1(i)=V;
end;

if (TETAOFF<th_rem1(i)&(th_rem1(i)<=TETAQ))
    Va1(i)=-V;;
end;

if (th_rem1(i)>TETAQ)
    Va1(i)=0;
end;

if (0<=th_rem1(i))&(th_rem1(i)<TETAON)
    Va1(i)=0;
end;

if (TETAON<=th_rem2(i)&(th_rem2(i)<=TETAOFF))
    Va2(i)=V;
end;

if (TETAOFF<th_rem2(i)&(th_rem2(i)<=TETAQ))
    Va2(i)=-V;;
end;

if (th_rem2(i)>TETAQ)
    Va2(i)=0;
end;

if (0<=th_rem2(i))&(th_rem2(i)<TETAON)
    Va2(i)=0;
end;

if (TETAON<=th_rem3(i)&(th_rem3(i)<=TETAOFF))
    Va3(i)=V;
end;

if (TETAOFF<th_rem3(i)&(th_rem3(i)<=TETAQ))
    Va3(i)=-V;;
end;

if (th_rem3(i)>TETAQ)
    Va3(i)=0;
end;

if (0<=th_rem3(i))&(th_rem3(i)<TETAON)
    Va3(i)=0;
end;
```

```
% compute current and flux

if ((0<=th_rem1(i))&(th_rem1(i)<=TETAX))
    I1(i)=flux1(i)/LMIN;
end;

if ((TETAX<th_rem1(i))&(th_rem1(i)<=TETAY))
    I1(i)=flux1(i)/((AUP*th_rem1(i))+BUP);
end;
if ((TETAY<th_rem1(i))& (th_rem1(i)<=TETAXY))
    I1(i)=flux1(i)/((ADOWN*th_rem1(i))+BDOWN);
end;

if (th_rem1(i)>TETAXY)
    I1(i)=flux1(i)/LMIN;
end;

if ((0<=th_rem2(i))&(th_rem2(i)<=TETAX))
    I2(i)=flux2(i)/LMIN;
end;

if ((TETAX<th_rem2(i))&(th_rem2(i)<=TETAY))
    I2(i)=flux2(i)/((AUP*th_rem2(i))+BUP);
end;
if ((TETAY<th_rem2(i))& (th_rem2(i)<=TETAXY))
    I2(i)=flux2(i)/((ADOWN*th_rem2(i))+BDOWN);
end;

if (th_rem2(i)>TETAXY)
    I2(i)=flux2(i)/LMIN;
end;



if ((0<=th_rem3(i))&(th_rem3(i)<=TETAX))
    I3(i)=flux3(i)/LMIN;
end;

if ((TETAX<th_rem3(i))&(th_rem3(i)<=TETAY))
    I3(i)=flux3(i)/((AUP*th_rem3(i))+BUP);
end;
if ((TETAY<th_rem3(i))& (th_rem3(i)<=TETAXY))
    I3(i)=flux3(i)/((ADOWN*th_rem3(i))+BDOWN);
end;

if (th_rem3(i)>TETAXY)
    I3(i)=flux3(i)/LMIN;
end;
% compute torque
if ((0<=th_rem1(i))&(th_rem1(i)<=TETAX))
    T1(i)=0;
end;

if ((TETAX<th_rem1(i))&(th_rem1(i)<=TETAY))
```

```
    T1(i)=0.5*(DL)*(I1(i)*I1(i));
end;
if ((TETAY<th_rem1(i))&(th_rem1(i)<=TETAXY))
    T1(i)=-0.5*(DL)*(I1(i)*I1(i));
end;
if (th_rem1(i)>TETAXY)
    T1(i)=0;
end;
while T1(i)<0
    T1(i)=0;
end;

if ((0<=th_rem2(i))&(th_rem2(i)<=TETAX))
    T2(i)=0;
end;

if ((TETAX<th_rem2(i))&(th_rem2(i)<=TETAY))
    T2(i)=0.5*(DL)*(I2(i)*I2(i));
end;
if ((TETAY<th_rem2(i))&(th_rem2(i)<=TETAXY))
    T2(i)=-0.5*(DL)*(I2(i)*I2(i));
end;
if (th_rem2(i)>TETAXY)
    T2(i)=0;
end;
while T2(i)<0
    T2(i)=0;
end;

if ((0<=th_rem3(i))&(th_rem3(i)<=TETAX))
    T3(i)=0;
end;

if ((TETAX<th_rem3(i))&(th_rem3(i)<=TETAY))
    T3(i)=0.5*(DL)*(I3(i)*I3(i));
end;
if ((TETAY<th_rem3(i))&(th_rem3(i)<=TETAXY))
    T3(i)=-0.5*(DL)*(I3(i)*I3(i));
end;
if (th_rem3(i)>TETAXY)
    T3(i)=0;
end;
while T3(i)<0
    T3(i)=0;
end;

    flux1(i+1)=flux1(i)+(Va1(i)-(R*I1(i)))*ts;
    flux2(i+1)=flux2(i)+(Va2(i)-(R*I2(i)))*ts;
    flux3(i+1)=flux3(i)+(Va3(i)-(R*I3(i)))*ts;
    while I1(i)<0
        I1(i)=0;
    end;
    while I2(i)<0
        I2(i)=0;
    end;
    while I3(i)<0
```

```
        I3(i)=0
    end;

    % compute FEM
if ((0<=th_rem1(i))&(th_rem1(i)<=TETAX))
    DL1(i)=0;
end;

if ((TETAX<th_rem1(i))&(th_rem1(i)<=TETAY))
    DL1(i)=DL;
end;
if ((TETAY<th_rem1(i))&(th_rem1(i)<=TETAXY))
    DL1(i)=-DL;
end;
if (th_rem1(i)>TETAXY)
    DL1(i)=0;
end;

if ((0<=th_rem2(i))&(th_rem2(i)<=TETAX))
    DL2(i)=0;
end;

if ((TETAX<th_rem2(i))&(th_rem2(i)<=TETAY))
    DL2(i)=DL;
end;
if ((TETAY<th_rem2(i))&(th_rem2(i)<=TETAXY))
    DL2(i)=-DL;
end;
if (th_rem2(i)>TETAXY)
    DL2(i)=0;
end;

if ((0<=th_rem3(i))&(th_rem3(i)<=TETAX))
    DL3(i)=0;
end;

if ((TETAX<th_rem3(i))&(th_rem3(i)<=TETAY))
    DL3(i)=DL;
end;
if ((TETAY<th_rem3(i))&(th_rem3(i)<=TETAXY))
    DL3(i)=-DL;
end;
if (th_rem3(i)>TETAXY)
    DL3(i)=0;
end;

    FEM1(i)=I1(i)*W(i)*DL1(i);
    FEM2(i)=I2(i)*W(i)*DL2(i);
    FEM3(i)=I2(i)*W(i)*DL3(i);

T(i)=T1(i)+T2(i)+T3(i);
t(i+1)=t(i)+ts;
if (t(i+1)>tsim/4)&(t(i+1)<tsim/3.9);
    TL=2;
end;
```

```
W(i+1)=ts/J*(T(i)-TL-F*W(i))+W(i);
theta(i+1)=theta(i)+(pi/314.6);
end;
figure(1)
plot (t(1:end-1),L1,t(1:end-1),L2,t(1:end-1),L3)
figure(2)
plot (t(1:end-1),Va1)
figure(3)
plot (t(1:end-1),I1)
figure(4)
plot (t(1:end-1),T)
figure(5)
plot (t,W)
figure(6)
plot (t(1:end-1),FEM1)
figure(7)
plot (t,flux1)
```

Appendix B

MATLAB code for small signal analysis of a linearized SRM

```
%%%% Small signal model of linearized SRM%%%
Rs=0.931;
DL=0.234;     %slope of the inductance profile is 0.234H/rad.
Wmo=261;    %speed is in rad/sec
Req=Rs+(DL*Wmo);
L=22.1;     %average of minimum and maximum inductance.
dI=0;
dWm=0;
Ir=10;
dV=0;
Kb=DL*Ir;
J=0.006;    %Rotor inertia
B=0.001;    %Rotor friction constant
Bl=0;
Bt=B+Bl;
% dTl=0;

A_SRM=[-Req/L -Kb;1/J*Kb -(Bt/J)];
eig(A_SRM)

ts=0.00065
tsim=1;
n=tsim/ts;
t=zeros(1,n);
```

```
dist_start=0.1;
dist_end=0.2;
for i=1:tsim/ts
    t(i+1)=t(i)+ts;
    dI(i+1)=dI(i)+ts*((-Req/L)*dI(i)-Kb*dWm(i)+(dV/L))
    if i<dist_start/ts
        dTl=0;
    elseif i>dist_start/ts && i<dist_end/ts
        dTl=0.00006
    else dTl=0;
    end
    dWm(i+1)=dWm(i)+ts*((1/J)*Kb*dI(i)-(((B/J)+(Bl/J))*dWm(i))-(dTl/J))

end;
plot(t,dWm)
```

Appendix C1

MATLAB code for speed control loop simulation with PI controller only

```
%%%% Speed loop with PI controller only%%%
clc; clear all;
Tw=0.1;          % Speed feedback time constant
Hw=0.0383;       %Speed feedback gain
Vdc=400;         %DC link voltage
Vc=10;           %Command signal level
kr=Vdc/Vc;       %Converter gain
B=0.001;         %Rotor friction constant
Bl=0;            %Load friction constant
Bt=B+Bl;
DL=0.234;        %inductance slope=0.234 rad/sec
Ir=10;           %Rated current
kb=DL*Ir;        %Back e.m.f constant
Rs=0.931;        %phase resistance
k1=(Bt/(kb^2+Rs*Bt));
B=0.001;     %Rotor friction constant
Bl=0;
Bt=B+Bl;
J=0.006;         %Rotor inertia
Tm=(J/Bt);
Ts=4*Tw;
k2=((kb*Hw)/(Bt*Tm));
ks=(1/2*k2*Tw);
E=[-1/Tw 0 -(kb*Hw*ks)/(Bt*Tm*Tw) (Hw*ks/Tw)*(1/Tm-1/Ts);1 0 0 0;0 k1/Ts -
1/Ts 0;0 0 kb/(Bt*Tm) -1/Tm]
F=[ks/(Tw*Ts);0;0;0]
G=[0 0 0 1]
H=0
%%% Putting initial value to the states
x1=0;
```

```
x2=0;
x3=0;
Wm=0;
[num,den]=ss2tf(E,F,G,H)
y=tf(num,den)
figure(1)
step(y);
ITAE=0.0;
ts=0.0065
tsim=100;
n=tsim/ts;
t=zeros(1,n);
dist_start=0.1;
dist_end=0.2;
Wr=1;
error=0;
for i=1:tsim/ts
    t(i+1)=t(i)+ts;
%     if i<dist_start/ts
%         Wr=0;
%     elseif i>dist_start/ts && i<dist_end/ts
%         Wr=0.01;
%     else
%         Wr=0;
%     end
    x1(i+1)=x1(i)+ts*((-1/Tw)*x1(i)-
(kb*Hw*ks)/(Bt*Tm*Tw)*x3(i)+(Hw*ks/Tw)*(1/Tm-1/Ts)*Wm(i)+(ks/(Tw*Ts))*Wr);
    x2(i+1)=x2(i)+ts*x1(i);
    x3(i+1)=x3(i)+ts*(k1/Ts*x2(i)-x3(i)/Ts);
    Wm(i+1)=Wm(i)+ts*(kb*x3(i)/(Bt*Tm)-Wm(i)/Tm);

end;




figure(2)
plot(t,Wm)
```

Appendix C2

MATLAB code for calculating fitness function of Genetic algorithm

```
function fitness=ga_srm(ks,Ts)
global Tw Hw Vdc Vc kr B Bl Bt DL Ir kb Rs J k1 Tm

Ts=4*Tw;
```

```
k2=((kb*Hw)/(Bt*Tm));
ks=(1/2*k2*Tw);

%%% Putting initial value to the states
x1=0;
x2=0;
x3=0;
Wm=0;


ITAE=0.0;
ts=0.065
tsim=50;
n=tsim/ts;
t=zeros(1,n);
dist_start=0.0;
dist_end=0.1;
Wr=1;
for i=1:tsim/ts
    t(i+1)=t(i)+ts;

    x1(i+1)=x1(i)+ts*((-1/Tw)*x1(i)-
(kb*Hw*ks)/(Bt*Tm*Tw)*x3(i)+(Hw*ks/Tw)*(1/Tm-1/Ts)*Wm(i)+(ks/(Tw*Ts))*Wr);
    x2(i+1)=x2(i)+ts*x1(i);
    x3(i+1)=x3(i)+ts*(k1/Ts*x2(i)-x3(i)/Ts);
    Wm(i+1)=Wm(i)+ts*(kb*x3(i)/(Bt*Tm)-Wm(i)/Tm);
    ITAE(i+1)=ITAE(i)+abs(Wm(i+1)-Wr)*ts;
end;
fitness=1/ITAE(end)
```

Appendix C3

MATLAB code for genetically tuned PI controller for speed loop

```
clc;
clear all;
global Tw Hw Vdc Vc kr B Bl Bt DL Ir kb Rs J k1 Tm

Tw=0.1;          % Speed feedback time constant
Hw=0.0383;       %Speed feedback gain
Vdc=400;         %DC link voltage
Vc=10;           %Command signal level
kr=Vdc/Vc;       %Converter gain
B=0.001;         %Rotor friction constant
Bl=0;            %Load friction constant
Bt=B+Bl;
DL=0.234;        %inductance slope=0.234 rad/sec
```

```
Ir=10;          %Rated current
kb=DL*Ir;       %Back e.m.f constant
Rs=0.931;       %phase resistance
k1=(Bt/(kb^2+Rs*Bt));
B=0.001;        %Rotor friction constant
Bl=0;
Bt=B+Bl;
J=0.006;        %Rotor inertia
Tm=(J/Bt);
% Ts=4*Tw;
k2=((kb*Hw)/(Bt*Tm));
% Setting boundaries
bounds1 = [1 20];
bounds2=[0.001 0.1];
% bounds3=[0.001 0.1];
bk =[0.1 5];
% pop size
n = 50;
% number of iterations
numits = 15;
% numer of mutations per it
nummut = 2;
param=2; % no. of parameters to be optimized

f=@ga_srm;

bkl = bk(2)-bk(1);
blength1 = bounds1(2)-bounds1(1);
blength2 = bounds2(2)-bounds2(1);
% blength3 = bounds3(2)-bounds3(1);

% Initializing population
pop1 = rand(1,n)*blength1 + bounds1(1);
pop2=rand(1,n)*blength2 + bounds2(1);
% pop3=rand(1,n)*blength3 + bounds3(1);
% pop(1,:)=rand*bkl + bk(1);
pop=[pop1; pop2];
% GA Cycle
for it=1:numits

    % fitness eval
    for i=1:n,
    fpop(i) = feval(f, pop(1,i),pop(2,i));
    end


      maxf(it) = max(fpop);
      meanf(it) = mean(fpop);

    % subtract worst fitness in order to normalize
    m=min(fpop);
    fpop=fpop-m;
    cpop(1) = fpop(1);
```

```
    for i=2:n,
        cpop(i) = cpop(i-1) + fpop(i);
    end

% SELECTION
    total_fitness = cpop(n);

    % use roulette selection (-> need pos.fitness!)
    for i=1:n
        p=rand*total_fitness;

        % now find first index
        j=find(cpop-p>0);
        if isempty(j)
            j=n;
        else
            j=j(1);
        end
        parent(1,i)=pop(1,j);
        parent(2,i)=pop(2,j);
%         parent(3,i)=pop(3,j);
    end


    % REPRODUCTION
    % parents 2i-1 and 2i make two new      children 2i-1 and 2i

    % crossover
    % use arithmetic crossover
    for i=1:2:n
        r=rand;
        pop(1,i)   =     r*parent(1,i) + (1-r)*parent(1,i+1);
        pop(1,i+1) = (1-r)*parent(1,i) +     r*parent(1,i+1);
        pop(2,i)   =     r*parent(2,i) + (1-r)*parent(2,i+1);
        pop(2,i+1) = (1-r)*parent(2,i) +     r*parent(2,i+1);
%         pop(3,i)   =     r*parent(3,i) + (1-r)*parent(3,i+1);
%         pop(3,i+1) = (1-r)*parent(3,i) +     r*parent(3,i+1);
    end

    % mutation
    % use uniform mutation
    for i=1:nummut
%         pop(1,ceil(rand*n)) = bk(1) +     rand*bkl;
        pop(1,ceil(rand*n)) = bounds1(1) +  rand*blength1;
        pop(2,ceil(rand*n)) = bounds2(1) +  rand*blength2;
%         pop(3,ceil(rand*n)) = bounds3(1) +    rand*blength3;
    end

end

% Final population & its fitness values
pop
for i=1:n,
    fpop(i) = feval(f, pop(1,i),pop(2,i));
end
```

```
fpop

% Getting fitness value and indes of best individual
[f,i]=max(fpop)

% Assigning best values to k, T, L
ks=pop(1,i)
Ts=pop(2,i)


% Final Model & its Simulation
x1=0;
x2=0;
x3=0;
Wm=0;
E=[-1/Tw 0 -(kb*Hw*ks/(Bt*Tm*Tw)) (Hw*ks/Tw)*(1/Tm-1/Ts);1 0 0 0;0 k1/Ts -
1/Ts 0;0 0 kb/(Bt*Tm) -1/Tm]
F=[ks/(Tw*Ts);0;0;0]
G=[0 0 0 1]
H=0
[num,den]=ss2tf(E,F,G,H)
Z=tf(num,den)
figure(1)
step(Z);
ITAE=0.0;
ts=0.065
tsim=50;
n=tsim/ts;
t=zeros(1,n);
dist_start=0.0;
dist_end=0.1;
Wr=1;

for i=1:tsim/ts
    t(i+1)=t(i)+ts;

    x1(i+1)=x1(i)+ts*((-1/Tw)*x1(i)-
(kb*Hw*ks)/(Bt*Tm*Tw)*x3(i)+(Hw*ks/Tw)*(1/Tm-1/Ts)*Wm(i)+(ks/(Tw*Ts))*Wr);
    x2(i+1)=x2(i)+ts*x1(i);
    x3(i+1)=x3(i)+ts*(k1/Ts*x2(i)-x3(i)/Ts);
    Wm(i+1)=Wm(i)+ts*(kb*x3(i)/(Bt*Tm)-Wm(i)/Tm);

end;
figure(2)
plot(t,Wm)
```