A MAGNETIC MODEL OF THREE-PHASE SWITCHED RELUCTANCE MACHINE USING CUBIC SPLINE INTERPOLATION TECHNIQUE

SISAVATH KHOTPANYA

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING

SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2006

ISBN 974-15-2285-1

หัวข้อวิทยานิพนธ์	การจำลองคุณลักษณะทางแม่เหล็กไฟฟ้าของเครื่องจักรกลไฟฟ้าแบบสวิทซ์รีลัคแตนซ์โดยใช้เทคนิคแบบคิวบิคสไปลน์
ชื่อนักศึกษา	นาย สีสะหวาด โคดปันยา
รหัสประจำตัว	47060230
ปริญญา	วิศวกรรมมหาบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2549
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ผศ. ดร. สุพัฒน์ กิตติรัตน์สัจจา
อาจารย์ผู้ควบคุมวิทยานิพนธ์ร่วม	ศ. ดร. อิชิบะชิ คะซิฮิสะ

บทคัดย่อ

เครื่องจักรกลแบบสวิทซ์รีลัคแตนซ์เป็นอีกทางเลือกหนึ่งในการใช้งานระบบขับเคลื่อนเครื่องจักรกล ซึ่งปัจจุบันมีอยู่หลายชนิด อย่างไรก็ตามในการออกแบบเพื่อที่ทำการจำลองพฤติกรรมของเครื่องจักรกลดังกล่าวนี้ไม่ได้เป็นที่แพร่หลายเหมือนกับเครื่องจักรกลไฟฟ้าชนิดอื่นๆ ดังนั้นในวิทยานิพนธ์นี้จึงได้นำเสนอการจำลองคุณลักษณะทางด้านแม่เหล็กไฟฟ้าของเครื่องจักรกแบบสวิทซ์รีลัคแตนซ์ที่เป็นแบบไม่เป็นเชิงเส้น โดยเน้นการประมาณหาค่าเส้นแรงแม่เหล็กได้อย่างรวดเร็ว ซึ่งแบบจำลองที่ได้นำเสนอนี้ประกอบไปด้วยสมการของเส้นแรงแม่เหล็กที่เปลี่ยนแปลงตามค่ากระแสเฟส และ ตำแหน่งโรเตอร์ ดังนั้นเทคนิคคิวบิคสไปลน์จึงได้ถูกนำมาใช้เพื่อประมาณหาค่าแต่ละจุดของเส้นแรงแม่เหล็ก ในปัจจุบันสังเกตได้ว่าวิธีที่ใช้กันส่วนมากแล้วต้องการข้อมูลที่เป็นจุดของเส้นแรงแม่เหล็กซึ่งเปลี่ยนแปลงตามค่ากระแสเฟส และตำแหน่งโรเตอร์ เพื่อใช้สร้างเป็นแบบจำลอง ดังนั้นโปรแกรมไฟไนต์อิลิเมนต์จึงได้ถูกนำมาใช้ในการเก็บข้อมูลของแบบจำลองที่ได้นำเสนอ และได้ถูกพัฒนาจนกระทั่งสามารถหาค่าของแรงบิด และ ค่าอินดักแตนซ์โพลไฟล์ได้อีกด้วย นอกจากนี้ผลที่ได้จากการจำลองก็ยังได้ถูกนำมาเปรียบเทียบกับค่าทีจากโปรแกรมไฟไนต์อิลิเมนต์เพื่อยืนยันความถูกต้องของแบบจำลอง

Thesis Title	A Magnetic Model of Three-Phase Switched Reluctance Machine using Cubic Spline Interpolation Technique
Student	Mr. Sisavath Khotpanya
Student I. D	47060230
Degree	Master of Engineering
Program	Electrical Engineering
Year	2006
Thesis Advisor	Asst. Prof. Dr. Supat Kittiratsatcha
Co-Thesis Advisor	Prof. Dr. Ishibashi Kazuhisa

ABSTRACT

Switched reluctance motors (SRM) provide an alternative for many application drives. However, the modeling and designing of SRMs are not as mature as other conventional electric machines. This work presents a simplified non-linear magnetic analysis model of a SRM in order to rapidly compute the non-linear magnetizing curves for the given SRM. The propose model comprises of polynomial flux linkage function which depends on phase current and rotor position. Cubic spline interpolation technique has been implemented to estimate the coefficients of the candidate function. At present, most of the conventional methods require numerous data points of flux-linkage corresponding to rotor positions and phase currents to build a model therefore Finite Element Analysis (FEA) has been used for data collection. The model was then used to model torque production and inductance profile of the SRM. The effectiveness of the proposed model is verified by comparing with 2D-FEA in term of the accuracy.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Asst. Prof. Dr. Supat Kittiratsatcha for his valuable advice on the framework of this thesis, his continued guidance and support during the completion of the work, and his careful review of the results. Due to his expertise in the SRM, I could start the work in the right direction and complete the work successfully.

I would like to thank Prof. Dr. Ishibashi Kazuhisa from School of Engineering II, Department of Mechanical Engineering, Tokai University, Japan, my co-advisor, for his encouragement and support throughout the work.

I would like to acknowledge the Department of Electrical Engineering, Faculty of Engineering, KMITL for studied equipment and work place.

I would like to express my sincere gratitude to all KMITL's teachers for giving me a very useful knowledge and experience.

I would also like to thank my friend Mr. Sittichai Keawthai, who assisted me in working and was always available for discussions on my thesis ideas.

Finally, I would like to acknowledge AUN/SEED-Net (JICA) for financial support and giving me an opportunity to carry out my study.

Sisavath Khotpanya

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (Cont)

CHAPTER 1

INTRODUCTION

1.1 Background

The principle of variable reluctance machines has been known for a long time, but the advent of power electronics has made it useful especially in the form of switched-reluctance machines (SRMs). A typical, commonly used 6/4-pole configuration of an SRM is shown in Figure 1.1. SRMs are mechanically robust. This is due to their simple mechanical structures. Both stator and rotor have salient poles. The rotor has no windings. The stator coils are concentrated on individual poles. Concentrated coils are an important feature of the SRM because they strongly reduce the SRM's cost. Another important feature that appears in the SRM is a strong magnetic saturation of SRM stator teeth. The SRM is operated principally as a motor. Its coils are supplied by a power electronic supply with one-directional currents, while only one phase is energized by each current pulse. For example, the SRM shown in Figure 1.1 has three stator phases with two poles per phase. It is seen that the numbers of poles in the stator (six) are different than the numbers of poles in the rotor (four). This special configuration, quite unique for the SRM, would permit the starting and operation of the SRM without pole clamping in both rotation directions [1]. The power electronic supplies are simpler in construction and contain less active solid-state power devices than the power supplies connected with induction or brushless motors [2]. This fact makes the SRM quite attractive for mass production home appliances, as washing machines, refrigerators, and air conditioners. Other fields of applications are robotics and electric vehicles. The SRM is able to develop high mechanical torques at low speeds, making them suitable for direct-drive configurations, with the SRM directly connected to the robot arm or electric vehicle wheel.

The principal disadvantages of SRM are the acoustical noise and the mechanical torque which is less uniform than that of brushless motors or induction motors. The principle of operation of the SRM is based on the tendency of an energized magnetic circuit to change mechanically until the minimum magnetic reluctance is attained. As a result, when one SRM phase is energized, the rotor rotates until two of its poles are aligned with two stator poles of the energized phase. After that, a following phase is energized, with a further rotation of the rotor,

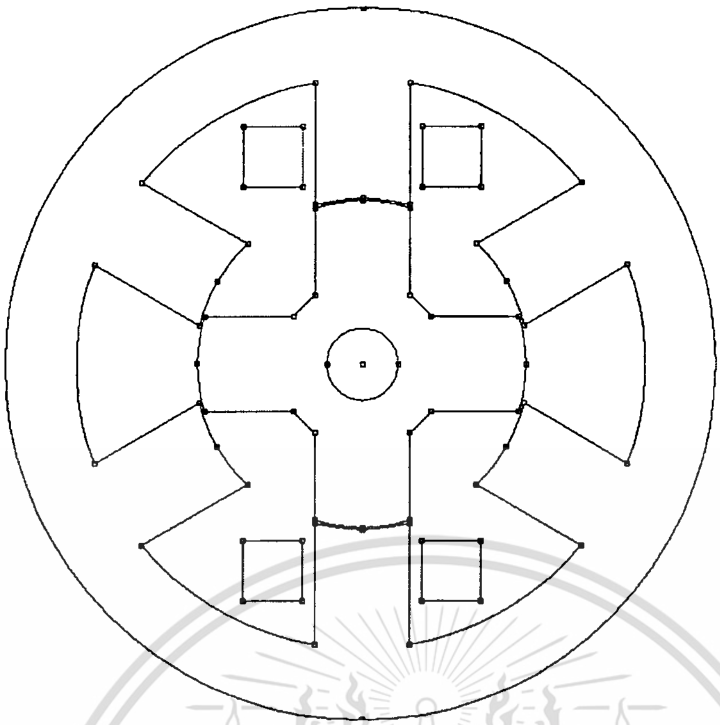


Figure 1.1 Structure of a 6/4 switched-reluctance motor, showing a rotor, a stator and the phase windings.

and so on. The motor torque is obtained while the rotor tends to be aligned with the stator poles and the magnetic reluctance of the magnetic circuit decreases [3]. The operation of SRM as generator is opposite the phase coils are energized when the rotor tends to unaligned from the stator poles and the magnetic reluctance of the magnetic circuit increases. For more operation of SRM will be described in Chapter 2.

1.2 Objective of the study

The objective of this research is to develop a model of the three-phase switched reluctance motor in order to rapidly compute the non-linear magnetizing curves for the given SRM. The non-uniform air gap, salient poles on stator and rotor, and the strong magnetic saturation of the stator poles provide a strong nonlinear behavior of the magnetic reluctance in SRM. These make the process of simulation and analysis quite difficult. Therefore, this research is concerned about a magnetic model of a SRM which based on polynomial flux linkage modeling. Cubic spline method is applied to the proposed model. The proposed model is very simple and can apply in torque production and inductance profile of SRM with the same cubic spline coefficients.

1.3 Assumption of this study

One of the disadvantages of SRM is that the nonlinear nature of SRM and torque ripple. These become the characteristics of the SRM. Thus, in term of model, it makes the calculation and simulation complexity. In this study, the assumptions are to propose the nonlinear function of magnetization curves, torque production and inductance profile by polynomial equations. The reason is that to make the mathematic analytical model less complicated. The cubic spline is implemented in this research as well.

1.4 Theory or concept to be used in the research

Magnetic saturation is very important to the high-performance operation of the SRM drive and is difficult to model analytically. The piecewise-linear model does a good job of predicting time average torque, but because of errors imbedded in the relationship between flux linkage and current, peak currents and instantaneous torque are not predicted with accuracy.

Because the piecewise linearizations of the magnetic characteristic has accuracy limitations, and the SRM drive modeling is required, it makes sense to try to find an analytic expression for the flux linkage, current, position data. The goal of this analytic expression is to provide all of the flux linkage-current information for every rotor position in one summary equation that is simple, matches the experimental (or numerical) data and can be connected to a physical interpretation. Thus, the theory or the concepts that used in this research are considered about the non-linearization model of the SRM. In terms of the analytical model, third order polynomials function and cubic spline interpolation method for estimation the magnetization properties of a SRM are used. This method will be explained more clearly in Chapter 3 and the implementation will be discussed in Chapter 4.

1.5 Scopes of this work

This research has been proposed about a magnetic model of the three-phase switched reluctance machine. The model starts from model flux linkage of SRM which depends on rotor position and phase current. The model was then expensed to implement in torque production and inductance profile of the SRM. The analysis model tool is developed on the basis of MATLAB program. Cubic spline interpolation method is used to estimate the value of candidate coefficients model at particular position. By using this technique, we can use less data and spend less time for data collection and interpolation, respectively.

1.6 Outline of the thesis

Chapter 2 discusses about the characteristic of SRM. The magnetization curves are presented. Equivalent circuit and torque production of SRM are formulated.

Chapter 3 presents the completed cubic spline interpolation technique. The theory and process are explained. The four constrains of cubic spline are proposed. In this chapter, the example of three types of cubic spline is given. Curve fitting with spline is also described.

Chapter 4 discusses the nonlinear modeling of SRM. The models consist of polynomial flux linkage, torque production and inductance profile.

Chapter 5 emphasizes the work and suggestions for the future work.

CHAPTER 2

CHARACTERISTICS OF A SRM

2.1 Definitions

A switched reluctance machine is an electric machine in which torque is produced by the tendency of its movable part to move to a position where the inductance of the excited winding is maximized [2]. The switched reluctance motor has salient poles on both stator and rotor, and operates like a controlled stepped motor.

The switched reluctance machine (SRM) drives for industrial applications are of recent origin. This chapter contains of any machine is its torque expression, which is derived from principle. The torque expression requires a relationship between machine flux linkages or inductance and the rotor position. The characteristic of magnetization curves of SRM is presented, i.e., aligned, unaligned, and intermediate positions. Equivalent circuit for SRM is formulated.

2.2 Magnetization curves

Figure 2.1 shows a 6/4 motor, i.e., one having six stator and four rotor poles. The 6/4 motor has three phases. Each phase comprises two coils wound on opposite poles and connected. They may be in series or in parallel, but we will assume for now that they are in series.

2.2.1 Aligned position

When any pair of rotor poles is exactly aligned with the stator poles of phase A, that phase is said to be in aligned position, as shown Figure 2.1(a). [The phase A poles are on the horizontal axis]. When current is flowing in phase A, there is no torque at this position because the rotor is in a position of maximum inductance. If the rotor is displaced to either side of the aligned position, as in Figure 2.1 (c) and (d), there is a restoring torque that tends to return the rotor towards the aligned position.

In the aligned position, the phase inductance is at its maximum and it is susceptible to saturation, especially in the stator and rotor yokes.

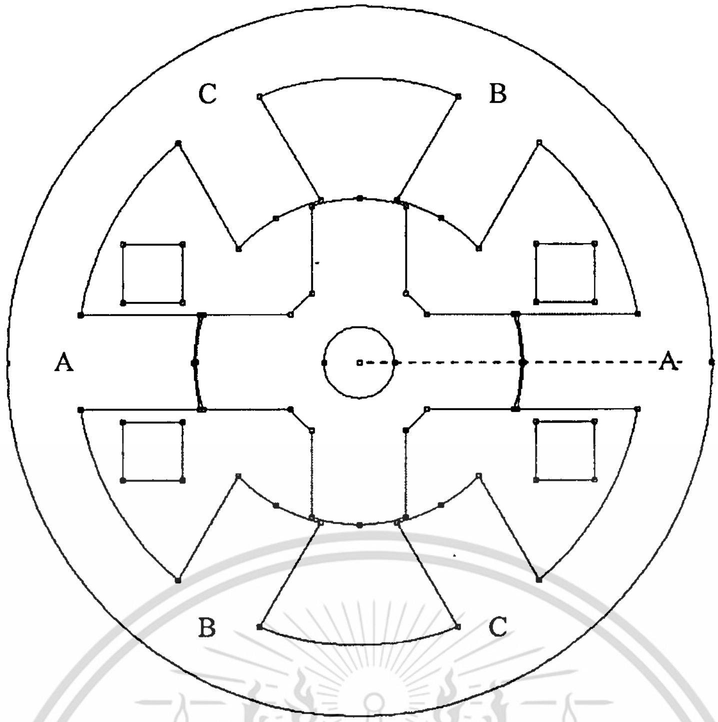


Figure 2.1 (a) 6/4 SRM – aligned position of phase A, the poles of phase A are on the horizontal axis.

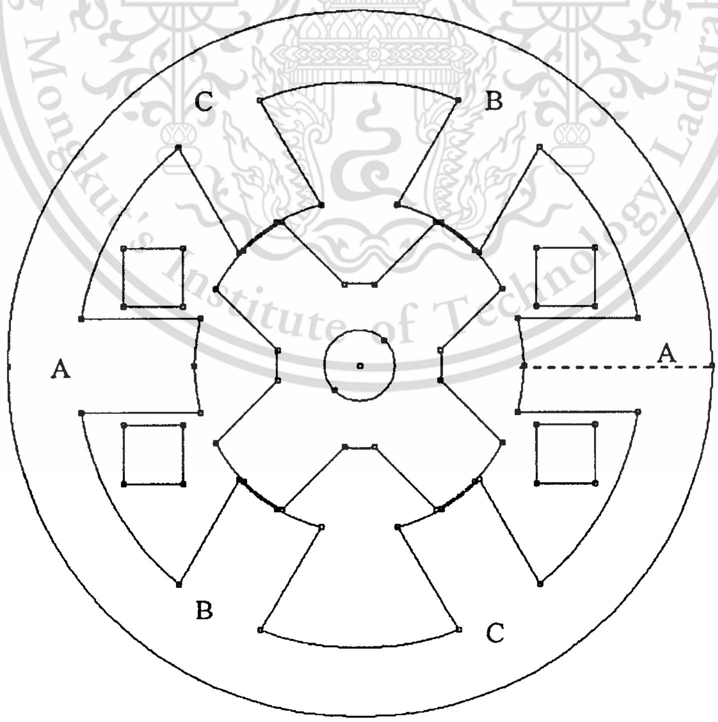


Figure 2.1 (b) 6/4 SRM – unaligned position on phase A.

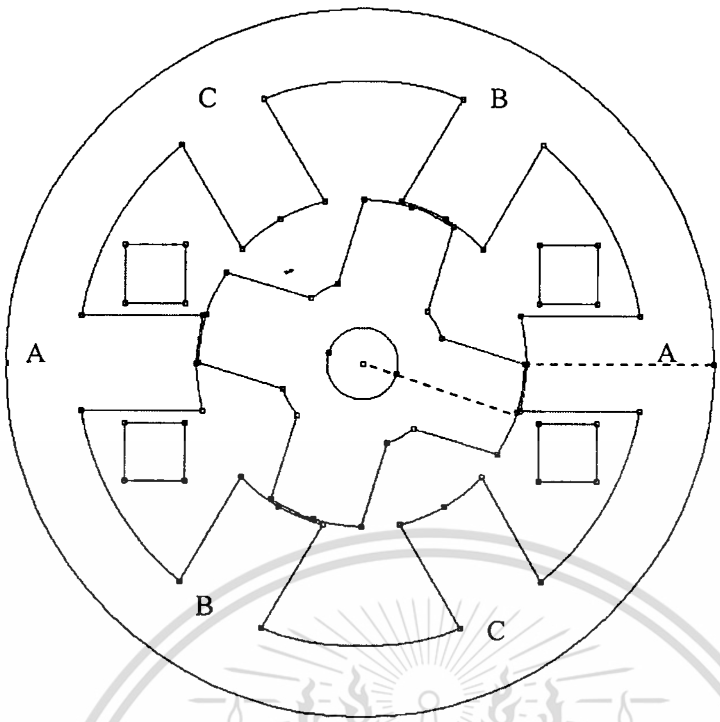


Figure 2.1 (c) 6/4 SRM – partial overlap position of phase A while motoring in the counter clockwise direction.

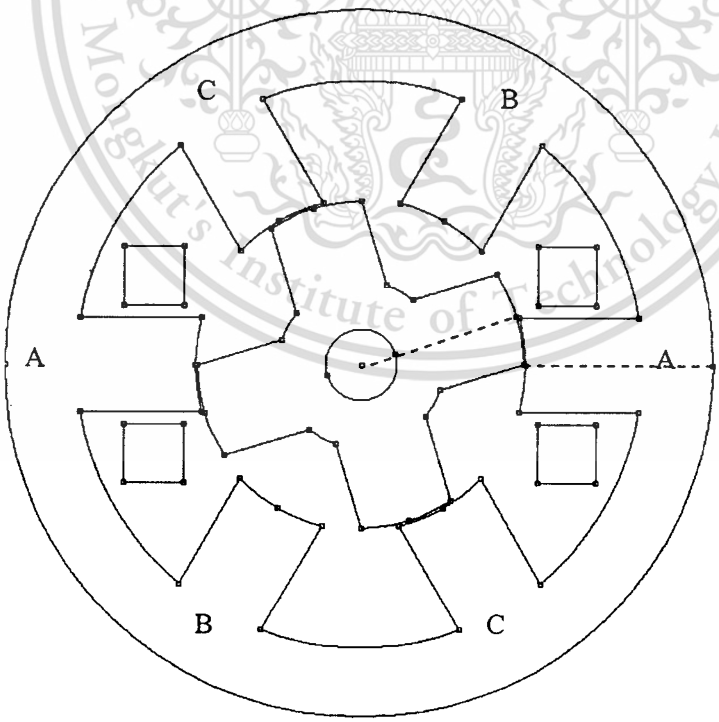


Figure 2.1 (d) 6/4 SRM – partial overlap position of phase A while generating in the counter clockwise direction.

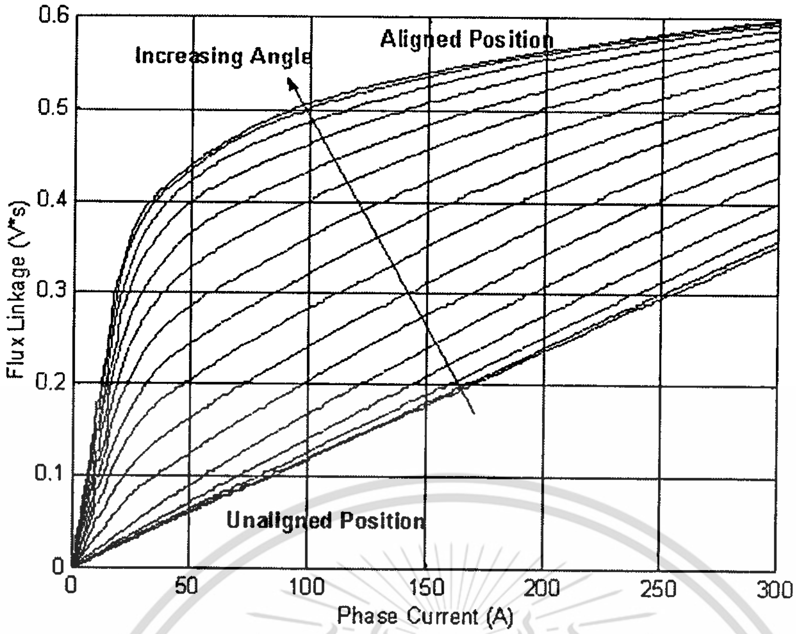


Figure 2.2 Complete set of magnetization curves, showing flux linkage versus current for one phase, with different rotor positions between the unaligned and aligned positions. The aligned curve is the highest, and the unaligned curve is the lowest.

2.2.2 Unaligned position

When the interpolar axis of the rotor is aligned with the poles of phase A, phase A is in the unaligned position, as shown in Figure 2.1 (b). When current is flowing in phase A, there is no torque at this position. If the rotor is displaced to either side of the unaligned position, there appears a torque that tends to displace it still further and attract it towards the next aligned position. The unaligned position is one of unstable equilibrium.

In the unaligned position, the phase inductance (the slope of the unaligned curve in Figure 2.2) is at its minimum, because the magnetic reluctance of the flux path is at its highest as a result of the large air gap between the stator and the rotor. The unaligned magnetization curve is not susceptible to saturation as the aligned curve.

2.2.3 Intermediate rotor position

At intermediate rotor positions such as the ones shown in Figure 2.1 (c) and (d), the magnetization curve is intermediate between the aligned and unaligned curves. The magnetization curves at intermediate rotor positions having the form shown in Figure 2.2. Because of the sharpness of the pole-corner, it would not be surprising if there were a sudden change in magnetization characteristics at the start of overlap, i.e. near the position shown in Figure 2.1 (c). This is indeed the case, between the unaligned position and the start of overlap, the magnetization curves do not change very rapidly. As the start of overlap is approach, the curves begin to sweep upwards and rapidly assume a shape closer to that of the aligned curve. In last few degrees before alignment, again there is little change in the curves.

2.3 Variation of inductance with rotor position

In the simple machine shown in Figure 2.1, the coil inductance L varies with rotor position θ as shown in Figure 2.3. Positive direction of rotation is in the counter clockwise direction. Assume that the coil carries a constant current. Positive motoring torque is produced only while the inductance is increasing as the rotor approaches the aligned position; that is, between position J and A. At J, the leading edge of the rotor pole is aligned with the first edge of a stator pole; at A, the rotor and stator poles are fully aligned. Thus J defines the start of overlap position, A is the maximum overlap, and K is the end of overlap.

The torque changes direction at the aligned position. If the rotor continues past A, the attractive force between the poles produces a retarding (braking) torque. If the machine rotates with constant current in the coil, the negative and positive torque impulses cancel, and therefore the average over a complete cycle is zero. To eliminate the negative torque impulse, the current must be switched off while the poles are separating, i.e. during intervals A-K, as shown in Figure 2.4.

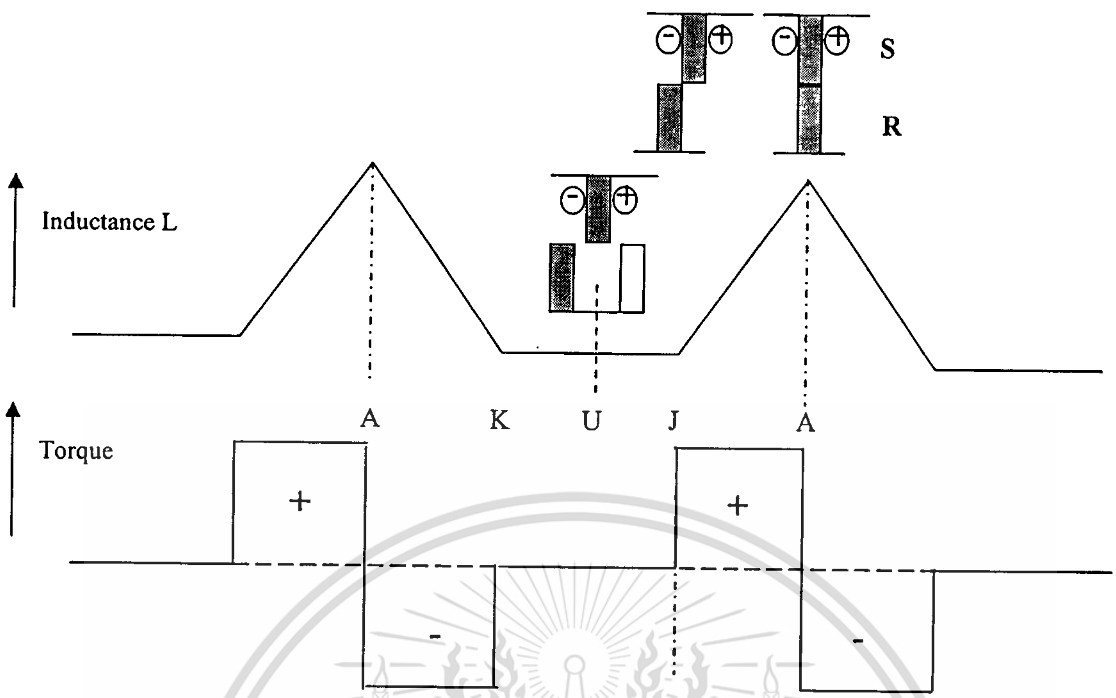


Figure 2.3 Variation of inductance and torque with rotor position, coil current is constant. The small icons show the relative positions of the rotor and stator poles, with the rotor moving to the right. A = aligned position; U = unaligned position; J = start of overlap; K = end of overlap.

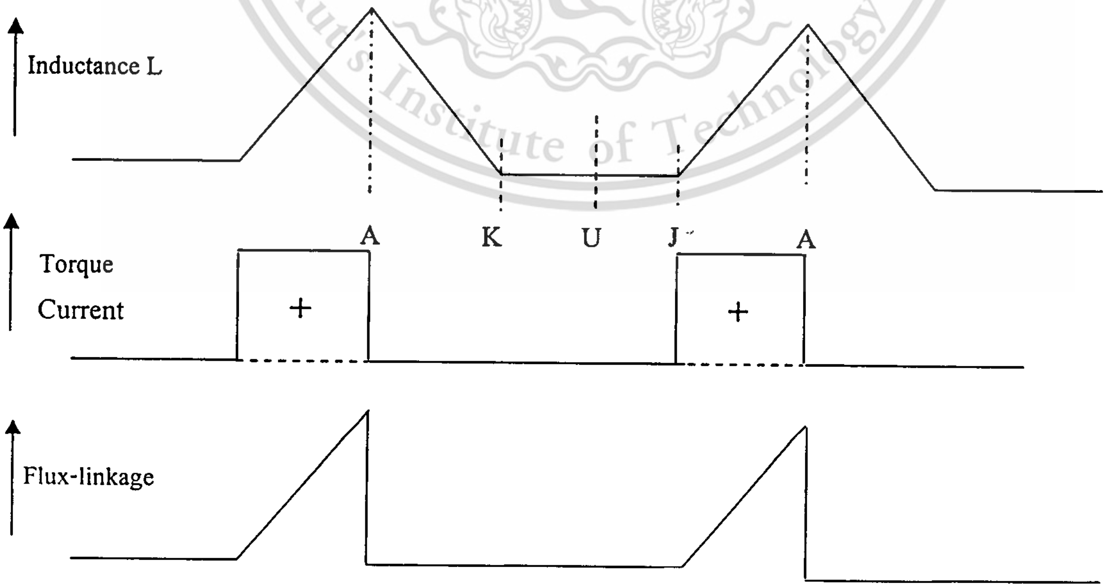


Figure 2.4 Variation of inductance, flux linkage, and torque with rotor position with ideal pulsed unidirectional current.

2.4 Equivalent circuit

An elementary equipment circuit for the SRM can be derived neglecting the mutual inductance between the phases as follows. The applied voltage to a phase is equal to the sum of the resistive voltage drop and the rate of the flux linkages and is given as:

$$v = R_s i + \frac{d\lambda(\theta,i)}{dt} \tag{2.1}$$

where R_s is the resistance per phase, and λ is the flux linkage per phase given by:

$$\lambda(\theta,i) = L(\theta,i)i \tag{2.2}$$

where L is phase inductance dependent on the rotor position and phase current. The phase voltage equation, then, is

$$\begin{aligned} v &= R_s i + \frac{d\{L(\theta,i)i\}}{dt} = R_s i + L(\theta,i)\frac{di}{dt} + i\frac{d\theta}{dt}.\frac{dL(\theta,i)}{d\theta} \\ &= R_s i + L(\theta,i)\frac{di}{dt} + \frac{dL(\theta,i)}{d\theta}\omega_m i \end{aligned} \tag{2.3}$$

In this equation, the three terms on the right-hand side represent the resistive voltage drop, inductive voltage drop, and induced emf, respectively, and the result is similar to the series excited dc motor voltage equation.

The induced emf, e , is obtained as:

$$e = \frac{dL(\theta,i)}{d\theta}\omega_m i = K_b \omega_m i \tag{2.4}$$

where K_b may be construed as an emf constant similar to that of the dc series excited machine and given here as:

$$K_b = \frac{dL(\theta,i)}{d\theta} \tag{2.5}$$

Note that the emf constant is dependent on operating point and is obtained with constant current at the point. From the voltage equation and the induced emf expression, the equivalent circuit for one phase of the SRM is derived and shown in Figure 2.5.

Substituting for the flux linkages in the voltage equation and multiplying with current results in instantaneous input power given by:

$$p_i = vi = R_s i^2 + i^2 \frac{dL(\theta,i)}{dt} + L(\theta,i) i \frac{di}{dt} \tag{2.6}$$

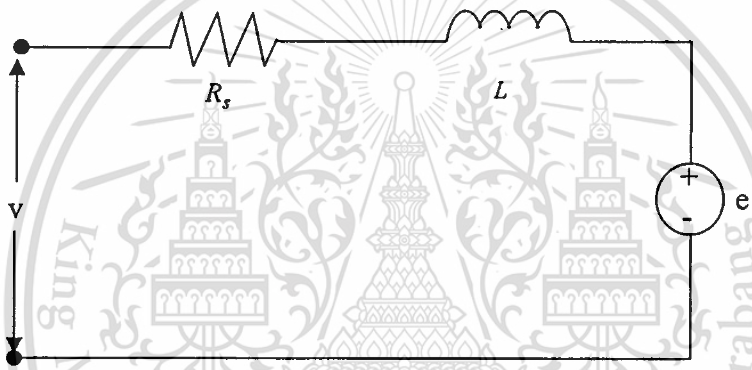


Figure 2.5 Single-phase equivalent circuit of the SRM.

Here, we can change the last term of equation (2.6) to draw a meaningful inference, it may be cast in term of known variables as in the following:

$$\frac{d}{dt}\left(\frac{1}{2} L(\theta,i) i^2\right) = L(\theta,i) i \frac{di}{dt} + \frac{1}{2} i^2 \frac{dL(\theta,i)}{dt} \tag{2.7}$$

Substituting the above equation (2.6) gives:

$$p_i = vi = R_s i^2 + \frac{d}{dt}\left(\frac{1}{2} L(\theta,i) i^2\right) + \frac{1}{2} i^2 \frac{dL(\theta,i)}{dt} \tag{2.8}$$

where p_i is the instantaneous input power. This implies that the sum of the winding resistive losses given by $R_s i^2$, the rate of change of the field energy given by $p[\frac{1}{2}L(\theta,i)i^2]$, and the air gap power, p_a, which is identified by the term $[i^2 pL(\theta,i)]/2$, where p is the differential operator, d/dt. Substituting for time in terms of the rotor position and speed, with

$$t = \frac{\theta}{\omega_m} \tag{2.9}$$

in the air gap power results in:

$$p_a = \frac{1}{2}i^2\frac{dL(\theta,i)}{dt} = \frac{1}{2}i^2\frac{dL(\theta,i)}{d\theta}\cdot\frac{d\theta}{dt} = \frac{1}{2}i^2\frac{dL(\theta,i)}{d\theta}\omega_m \tag{2.10}$$

The air gap power is the product of the electromagnetic torque and rotor speed given by:

$$p_a = \omega_m T_e \tag{2.11}$$

from which the torque is obtained by equating these two equations as:

$$T_e = \frac{1}{2}i^2\frac{dL(\theta,i)}{d\theta} \tag{2.12}$$

This completes development of the equivalent circuit and equation for evaluating electromagnetic torque, air gap power, and input power to the SRM both for dynamic and steady-state operations.

Equation (2.12) has in implication:

1. The torque is proportional to the square of the current; hence, the current can be unipolar to produce unidirectional torque.
2. The torque constant is given by the slope of the inductance versus rotor position characteristic. It is understood that the inductance of a stator winding is a function of both rotor position and current, thus making it nonlinear. Because of its

nonlinear nature, a simple equivalent circuit development for this motor is not possible.

3. Since the torque is proportional to the square of the current, this machine resembles a dc series motor; hence, it has a good starting torque.
4. A generating action is made possible with unipolar current due to its operation on the negative slope of the inductance profile.
5. The direction of rotation can be revered by changing the sequence of stator excitation, which is a simple operation.
6. Due to features 1 and 5, this machine is suitable for four-quadrant operation with a converter. Thus, torque and speed control is achieved with converter control.
7. This machine required a controllable converter for its operation and can not be operated directly from a three-phase line supply.
8. Because of its dependence on a power converter for its functioning, this motor drive is an inherently variable-speed motor drive system.
9. There is very little mutual inductance between machine phase windings in SRM, and for all practical purposes it is considered to be negligible. Since mutual coupling is absent, each phase is electrically independent of other phases. This is a feature unique to this machine only. Due to this feature, note that a short-circuit fault in one phase winding has no effect on other phases.

2.5 Conclusion

This chapter considers the characteristics of SRM such as: magnetization curves, aligned, unaligned, and intermediate position. The operation of SRM is considered. The derived of torque production is concluded and this will be described more in the latter chapter.

CHAPTER 3

CUBIC SPLINE INTERPOLATION

3.1 Introduction

A cubic spline is a smooth curve constructed to go through a set of points. The curve between each pair of points is a third-degree polynomial, which is computed to provide a smooth curve between the two points and to provide a smooth transition from the third-degree polynomial between the previous two points [4]. Real world numerical data is usually difficult to analyze. Any function which would effectively correlate the data would be difficult to obtain and highly unwieldy. To this end, the idea of the cubic spline was developed. Using this process, a series of unique cubic polynomials are fitted between each of the data points, with the constrains that the curve obtained be continuous and appear smooth. These cubic splines can then be used to determine rates of change and cumulative change over an interval. In this brief introduction, we will only discuss splines which interpolate equally spaced data points.

This chapter deals with the analysis cubic spline. The theory and process. The method of splines interpolation. The estimation of cubic spline coefficients is given, and curve fitting with cubic spline is presented.

3.2 Theory and process

The fundamental idea behind cubic spline interpolation is based on the engineer's tool used to draw smooth curves through a number of points. This spline consists of weights attached to a flat surface at the points to be connected. A flexible strip is then bent across each of these weights, resulting in a pleasingly smooth curve.

The mathematical spline is similar in principle. The points, in this case, are numerical data. The weights are the coefficients on the cubic polynomials used to interpolate the data. These coefficients 'bend' the line so that it passes through each of the data points without any erratic behavior or breaks in continuity.

The essential idea is to fit a piecewise function of the form:

$$S(x) = \begin{cases} s_1(x) & \text{if } x_1 \le x \le x_2 \\ s_2(x) & \text{if } x_2 \le x \le x_3 \\ \vdots & \vdots \\ s_{n-1}(x) & \text{if } x_{n-1} \le x \le x_n \end{cases} \tag{3.1}$$

where s_i is a third degree polynomial defined by

$$s_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \tag{3.2}$$

for $i = 1,2,...,n-1$.

The first and second derivatives of these $n-1$ equations are functional to this process, and they are

$$s_i'(x) = 3a_i(x - x_i)^2 + 2b_i(x - x_i) + c_i \tag{3.3}$$

$$s_i''(x) = 6a_i(x - x_i) + 2b_i \tag{3.4}$$

for $i = 1,2,...,n-1$.

3.3 The four properties of cubic spline

The spline will need to conform to the following constrains.

1. The piecewise function $S(x)$ will interpolate all data points.
2. $S(x)$ will be continuous on the interval $[x_1, x_n]$.
3. $S'(x)$ will be continuous on the interval $[x_1, x_n]$.
4. $S''(x)$ will be continuous on the interval $[x_1, x_n]$.

Since the piecewise function $S(x)$ will interpolate all of the data points, we can conclude that

$$S(x_i) = y_i \tag{3.5}$$

for $i = 1,2,...,n-1$. Since $x_i \in [x_i, x_{i+1}]$, $S(x_i) = s_i(x_i)$ and we can used equation (3.2) to produce

$$y_i = s_i(x_i)$$
$$y_i = a_i(x_i - x_i)^3 + b_i(x_i - x_i)^2 + c_i(x_i - x_i) + d_i \tag{3.6}$$
$$y_i = d_i$$

for each $i = 1,2,\ldots,n-1$.

Since the curve $S(x)$ must be continuous across its entire interval, it can be concluded that each sub-function must join at the data points, so

$$s_i(x_i) = s_{i-1}(x_i) \tag{3.7}$$

for $i = 1,2,\ldots,n-1$.

From equation (3.2),

$$s_i(x_i) = d_i \tag{3.8}$$

and

$$s_{i-1}(x_i) = a_{i-1}(x_i - x_{i-1})^3 + b_{i-1}(x_i - x_{i-1})^2 + c_{i-1}(x_i - x_{i-1}) + d_{i-1}$$

so

$$d_i = a_{i-1}(x_i - x_{i-1})^3 + b_{i-1}(x_i - x_{i-1})^2 + c_{i-1}(x_i - x_{i-1}) + d_{i-1} \tag{3.9}$$

for $i = 1,2,\ldots,n-1$. Letting $h = x_i - x_{i-1}$ in equation (3.8), we have

$$d_i = a_{i-1}h^3 + b_{i-1}h^2 + c_{i-1}h + d_{i-1} \tag{3.10}$$

for $i = 1,2,\ldots,n-1$.

Also, to make the curve smooth across the interval, the derivatives must be equal at the data points; that is,

$$s_i'(x_i) = s_{i-1}'(x_i) \tag{3.11}$$

However, by equation (3.3),

$$s_i'(x_i) = c_i$$

and

$$s'_{i-1}(x_i) = 3a_{i-1}(x_i - x_{i-1})^2 + 2b_{i-1}(x_i - x_{i-1})^1 + c_{i-1}$$

so

$$c_i = 3a_{i-1}(x_i - x_{i-1})^2 + 2b_{i-1}(x_i - x_{i-1}) + c_{i-1} \tag{3.12}$$

Again, letting $h = x_i - x_{i-1}$, we will get

$$c_i = 3a_{i-1}h^2 + 2b_{i-1}h + c_{i-1} \tag{3.13}$$

for $i = 1,2,...,n-1$.

From equation (3.4), $s''_i(x) = 6a_i(x - x_i) + 2b_i$, so

$$\begin{aligned} s''_i(x) &= 6a_i(x - x_i) + 2b_i \\ s''_i(x_i) &= 6a_i(x_i - x_i) + 2b_i \\ s''_i(x_i) &= 2b_i \end{aligned} \tag{3.14}$$

for $i = 1,2,...,n-1$.

Lastly, since $s''_i(x)$ has to be continuous across the interval $s''_i(x_i) = s''_{i+1}(x_i)$ for $i = 1,2,...,n-1$. This equation (3.14) leads us to the equation

$$s''_i(x_{i+1}) = 6a_i(x_{i+1} - x_i) + 2b_i \tag{3.15}$$

$$s''_{i+1}(x_{i+1}) = 6a_i(x_{i+1} - x_i) + 2b_i \tag{3.16}$$

And, letting $h = x_{i+1} - x_i$ and using the conclusion from equation (3.14) and (3.16),

$$s''_{i+1}(x_{i+1}) = 6a_i(x_{i+1} - x_i) + 2b_i \tag{3.17}$$

$$2b_{i+1} = 6a_i h + 2b_i \tag{3.18}$$

These equations can be much simplified by substituting M_i for $s''_i(x_i)$ and expressing the above equations in terms of M_i and y_i. This makes the determination of the weight a_i, b_i, c_i, and d_i be easy task. Each b_i can be represented by

$$s_i''(x_i) = 2b_i \tag{3.19}$$

$$M_i = 2b_i$$

$$b_i = \frac{M_i}{2}$$

and d_i has already been determined to be

$$d_i = y_i \tag{3.20}$$

Similarly, using equation a_i can be re-written as

$$2b_{i+1} = 6a_i h + 2b_i \tag{3.21}$$

$$6a_i h = 2b_{i+1} - 2b_i$$

$$a_i = \frac{2b_{i+1} - 2b_i}{6h}$$

$$a_i = \frac{2(\frac{M_{i+1}}{2}) - 2(\frac{M_i}{2})}{6h}$$

$$a_i = \frac{M_{i+1} - M_i)}{6h}$$

and c_i can be re-written as

$$d_{i+1} = a_i h^3 + b_i h^2 + c_i h + d_i \tag{3.22}$$

$$c_i h = -a_i h^3 - b_i h^2 - d_i + d_{i+1}$$

$$c_i = \frac{-a_i h^3 - b_i h^2 - d_i + d_{i+1}}{h}$$

$$c_i = \frac{-a_i h^3 - b_i h^2}{h} + \frac{-d_i + d_{i+1}}{h}$$

$$c_i = (-a_i h^2 - b_i h) - \frac{d_i - d_{i+1}}{h}$$

$$c_i = -(\frac{M_{i+1} - M_i}{6h} h^2 + \frac{M_i}{2} h) - \frac{y_i - y_{i+1}}{h}$$

$$c_i = \frac{y_{i+1} - y_i}{h} - (\frac{M_{i+1} - M_i}{6}h + \frac{3M_i}{6}h)$$
$$c_i = \frac{y_{i+1} - y_i}{h} - (\frac{M_{i+1} - M_i + 3M_i}{6})h$$
$$c_i = \frac{y_{i+1} - y_i}{h} - (\frac{M_{i+1} + 2M_i}{6})h$$

We now have the equations for determining the weights for $n-1$ equations as

$$a_i = \frac{M_{i+1} - M_i}{6h} \tag{3.23}$$
$$b_i = \frac{M_i}{2}$$
$$c_i = \frac{y_{i+1} - y_i}{h} - (\frac{M_{i+1} + 2M_i}{6})h$$
$$d_i = y_i$$

These systems can be handled more conveniently by putting them into matrix form as follows

$$c_{i+1} = 3a_i h^2 + 2b_{i1} h + c_i \tag{3.24}$$
$$3(\frac{M_{i+1} - M_i}{6h})h^2 + 2(\frac{M_i}{2})h + \frac{y_{i+1} - y_i}{h} - (\frac{M_{i+1} + 2M_i}{6})h = \frac{y_{i+2} - y_{i+1}}{h} - (\frac{M_{i+2} + 2M_{i+1}}{6})h$$
$$3(\frac{M_{i+1} - M_i}{6h})h^2 + 2(\frac{M_i}{2})h - (\frac{M_{i+1} + 2M_i}{6})h + (\frac{M_{i+2} + 2M_{i+1}}{6})h = -\frac{y_{i+1} - y_i}{h} + \frac{y_{i+2} - y_{i+1}}{h}$$
$$h(\frac{3M_{i+1} - 3M_i}{6h} + \frac{6M_i}{6} - (\frac{M_{i+1} + 2M_i}{6}) + (\frac{M_{i+2} + 2M_{i+1}}{6})) = \frac{y_i - 2y_{i+1} + y_{i+2}}{h}$$
$$\frac{h}{6}(M_i + 4M_{i+1} + M_{i+2}) = \frac{y_i - 2y_{i+1} + y_{i+2}}{h}$$
$$(M_i + 4M_{i+1} + M_{i+2}) = 6(\frac{y_i - 2y_{i+1} + y_{i+2}}{h^2})$$

for $i = 1, 2, \ldots, n-1$.

which leads to the matrix equation

$$\begin{bmatrix} 1 & 4 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 1 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 4 & 1 \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \\ \vdots \\ M_{n-2} \\ M_{n-1} \\ M_n \end{bmatrix} = \frac{6}{h^2} \begin{bmatrix} y_1 - 2y_2 + y_3 \\ y_2 - 2y_3 + y_4 \\ y_3 - 2y_4 + y_5 \\ \vdots \\ y_{n-4} - 2y_{n-3} + y_{n-2} \\ y_{n-3} - 2y_{n-2} + y_{n-1} \\ y_{n-2} - 2y_{n-1} + y_n \end{bmatrix}. \quad (3.25)$$

Note that this system has $n-2$ rows and n columns, and is therefore under-determined. In order to generate a unique cubic spline to other condition must be imposed upon the system. This lead us the next section.

3.4 Three type of Splines

3.4.1 Natural spline

This first spline includes the condition that the second derivative be equal to zero at the endpoints.

$$M_1 = M_n = 0 \quad (3.26)$$

This results in the spline extending as a line outside the endpoints. The matrix for determining the $M_1 - M_n$ values can be adapted accordingly.

$$\begin{bmatrix} 1 & 4 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 1 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 4 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ M_2 \\ M_3 \\ M_4 \\ \vdots \\ M_{n-2} \\ M_{n-1} \\ 0 \end{bmatrix} = \frac{6}{h^2} \begin{bmatrix} y_1 - 2y_2 + y_3 \\ y_2 - 2y_3 + y_4 \\ y_3 - 2y_4 + y_5 \\ \vdots \\ y_{n-4} - 2y_{n-3} + y_{n-2} \\ y_{n-3} - 2y_{n-2} + y_{n-1} \\ y_{n-2} - 2y_{n-1} + y_n \end{bmatrix} \quad (3.27)$$

For reasons of convenience, the first and last columns of this matrix can be eliminated, as they correspond to the M_1 and M_n values, which are both zero.

$$\begin{bmatrix} 4 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 4 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 4 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 4 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 4 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} M_2 \\ M_3 \\ M_4 \\ \vdots \\ M_{n-2} \\ M_{n-1} \end{bmatrix} = \frac{6}{h^2} \begin{bmatrix} y_1 - 2y_2 + y_3 \\ y_2 - 2y_3 + y_4 \\ y_3 - 2y_4 + y_5 \\ \vdots \\ y_{n-4} - 2y_{n-3} + y_{n-2} \\ y_{n-3} - 2y_{n-2} + y_{n-1} \\ y_{n-2} - 2y_{n-1} + y_n \end{bmatrix} \tag{3.28}$$

This results in and $n-2$ by $n-2$ matrix, which will determine the remaining solution for M_2 through M_{n-1}. The spline is now unique.

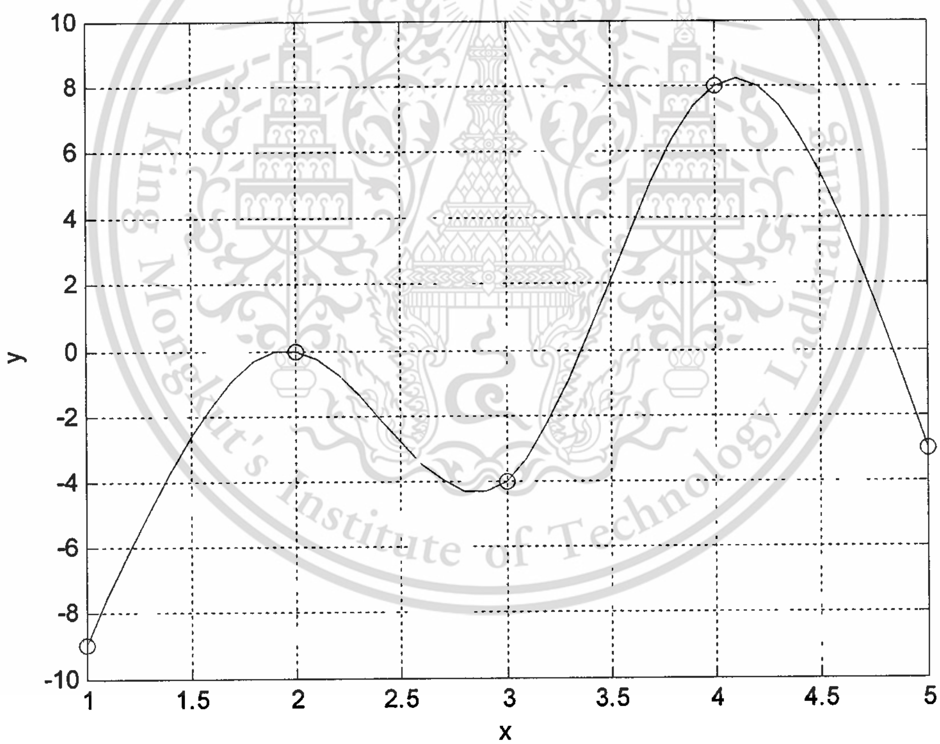


Figure 3.1 Natural interpolating curve.

3.4.2 Parabolic Runout Spline

The parabolic spline imposes the condition that the second derivative at the endpoints, M_1 and M_n be equal to M_2 through M_{n-1} respectively.

$$M_1 = M_2 \qquad (3.29)$$
$$M_n = M_{n-1}$$

The result of this condition is that the curve becomes a parabolic curve at the endpoints. This type of cubic spline is useful for periodic and experiential data. The matrix equation for this type of spline is

$$\begin{bmatrix} 5 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 4 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 4 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 4 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 4 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 5 \end{bmatrix} \begin{bmatrix} M_2 \\ M_3 \\ M_4 \\ \vdots \\ M_{n-2} \\ M_{n-1} \end{bmatrix} = \frac{6}{h^2} \begin{bmatrix} y_1 - 2y_2 + y_3 \\ y_2 - 2y_3 + y_4 \\ y_3 - 2y_4 + y_5 \\ \vdots \\ y_{n-4} - 2y_{n-3} + y_{n-2} \\ y_{n-3} - 2y_{n-2} + y_{n-1} \\ y_{n-2} - 2y_{n-1} + y_n \end{bmatrix} \qquad (3.30)$$

We can now determine the values of M_2 through M_{n-1}, with the values for M_1 and M_n already determined.

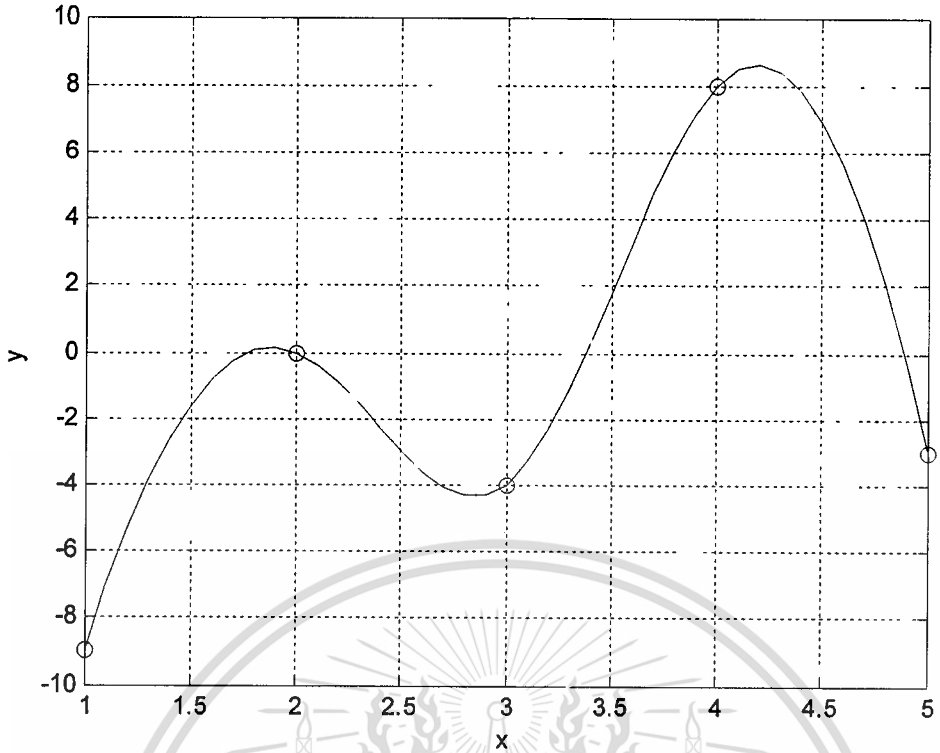


Figure 3.2 Parabolic Runout curve.

Note that the endpoints behavior is a little more extreme than the natural spline option.

3.4.3 Cubic Runout Spline

This last type of spline has the most extreme endpoints behavior. It assigns

$$M_1 = 2M_2 - M_3 \qquad (3.31)$$

$$M_n = 2M_{n-1} - M_{n-2}$$

This causes the curve to degrade to a single cubic curve over the last two intervals, rather than two separate functions. The matrix equation for this type is

$$\begin{bmatrix} 6 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 4 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 4 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 4 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 4 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 6 \end{bmatrix} \begin{bmatrix} \\ M_2 \\ M_3 \\ M_4 \\ \vdots \\ M_{n-2} \\ M_{n-1} \end{bmatrix} = \frac{6}{h^2} \begin{bmatrix} y_1 - 2y_2 + y_3 \\ y_2 - 2y_3 + y_4 \\ y_3 - 2y_4 + y_5 \\ \vdots \\ y_{n-4} - 2y_{n-3} + y_{n-2} \\ y_{n-3} - 2y_{n-2} + y_{n-1} \\ y_{n-2} - 2y_{n-1} + y_n \end{bmatrix} \tag{3.32}$$

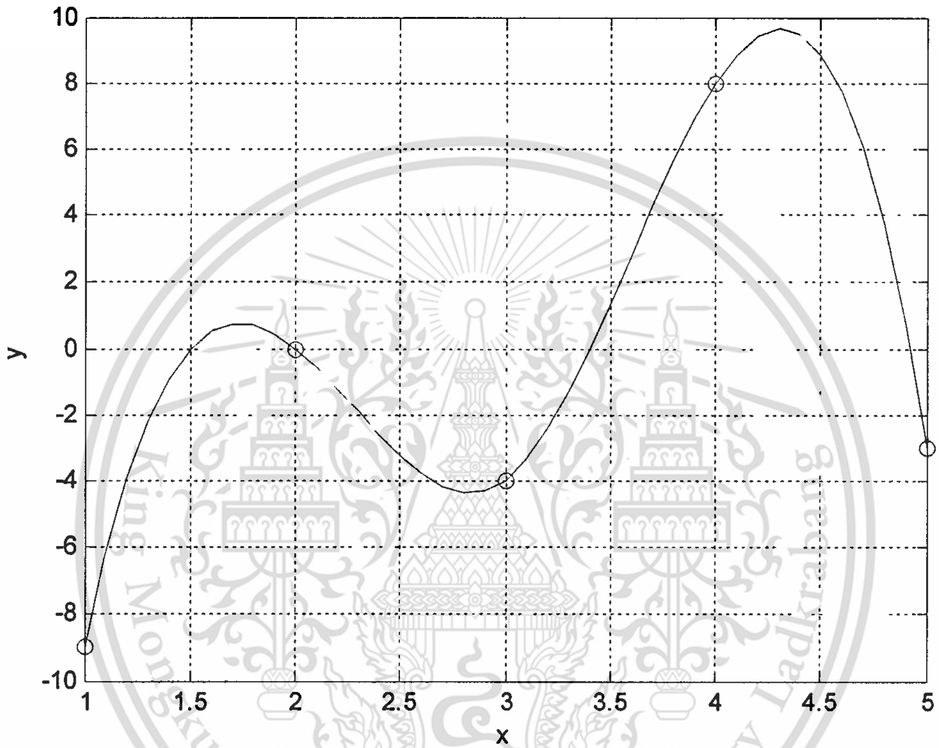


Figure 3.3 Cubic Runout curve.

Note the pronounced curvature at the endpoints.

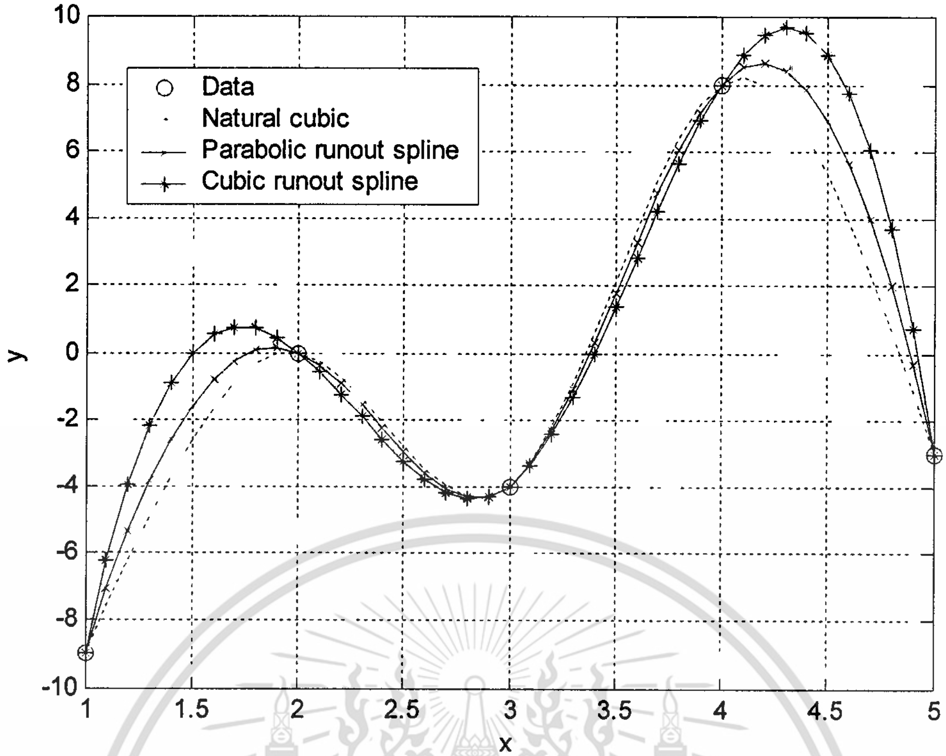


Figure 3.4 Comparison between three types of splines.

Keep in mind that there are many other types of interpolating spline curves, such as the Periodic Spline and the Clamped Spline. The three discussed in this work are simply the ones which we have chosen to examine; they are not intrinsically superior to, or more widely used than these other types of splines.

Example 3.1

Find the natural cubic spline that passes through (0, 0), (1, 0.5), (2, 2), and (3, 1.5) with the free boundary conditions $s''(0)=0$ and $s''(3)=0$.

Solution:

Use equation (3.28), we will find M_2 and M_3 as

$$\begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix}\begin{bmatrix} M_2 \\ M_3 \end{bmatrix} = \frac{6}{1^2}\begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

Thus,

$$\begin{bmatrix} M_2 \\ M_3 \end{bmatrix} = \begin{bmatrix} 2.4 \\ -3.6 \end{bmatrix}$$

Since the characteristic of natural spline $M_1 = M_4 = 0$, we can find the spline coefficients by using equation (3.23) and the piecewise function of natural spline is

$$\begin{aligned} s_1(x) &= 0.4x^3 + 0.1x && \text{for} \quad 0 \le x \le 1 \\ s_2(x) &= -(x-1)^3 + 1.2(x-1)^2 + 1.3(x-1) + 0.5 && \text{for} \quad 1 \le x \le 2 \\ s_3(x) &= 0.6(x-2)^3 - 1.8(x-2)^2 + 0.7(x-2) + 2 && \text{for} \quad 2 \le x \le 3 \end{aligned}$$

This natural spline is shown in Figure 3.5.

Example 3.2

Find the parabolic runout spline that passes through (0, 0), (1, 0.5), (2, 2), and (3, 1.5).

Solution:

We can calculate the value of M_2 and M_3 by using equation (3.30) as

$$\begin{bmatrix} 5 & 1 \\ 1 & 5 \end{bmatrix} \begin{bmatrix} M_2 \\ M_3 \end{bmatrix} = \frac{6}{1^2} \begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

Thus,

$$\begin{bmatrix} M_2 \\ M_3 \end{bmatrix} = \begin{bmatrix} 1.75 \\ -2.75 \end{bmatrix}$$

Since, the second derivative at the endpoints of parabolic runout spline, M_1 and M_4 be equal to M_2 through M_3 respectively. We will get the parabolic runout spline coefficients by using (3.23) and the piecewise of its functions as

$$s_1(x) = 0.875x^2 - 0.375x \qquad \text{for} \quad 0 \le x \le 1$$
$$s_2(x) = -0.75(x-1)^3 + 0.875(x-1)^2 + 1.375(x-1) + 0.5 \qquad \text{for} \quad 1 \le x \le 2$$
$$s_3(x) = -1.375(x-2)^2 + 0.875(x-2) + 2 \qquad \text{for} \quad 2 \le x \le 3$$

This parabolic spline is shown in Figure 3.6.

Example 3.3

Find the cubic runout spline through (0, 0), (1, 0.5), (2, 2), and (3, 1.5).

Solution:

By using equation (3.32), we can calculate the M_2 and M_3 as

$$\begin{bmatrix} 6 & 0 \\ 0 & 6 \end{bmatrix} \begin{bmatrix} M_2 \\ M_3 \end{bmatrix} = \frac{6}{1^2} \begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

Thus,

$$\begin{bmatrix} M_2 \\ M_3 \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

We can calculate the second derivatives of cubic runout spline, M_1 and M_4 as

$$M_1 = 2M_2 - M_3 = 4$$

and

$$M_4 = 2M_3 - M_2 = -5$$

By using equation (3.23), we obtained the cubic runout spline coefficients and the piecewise polynomial becomes

$$s_1(x) = -0.5x^3 + 2x^2 - x \qquad \text{for} \quad 0 \le x \le 1$$
$$s_2(x) = -0.5(x-1)^3 + 0.5(x-1)^2 + 1.5(x-1) + 0.5 \qquad \text{for} \quad 1 \le x \le 2$$
$$s_3(x) = -0.5x^3 - (x-2)^2 + (x-2) + 2 \qquad \text{for} \quad 2 \le x \le 3$$

This cubic runout spline is shown in Figure 3.7.

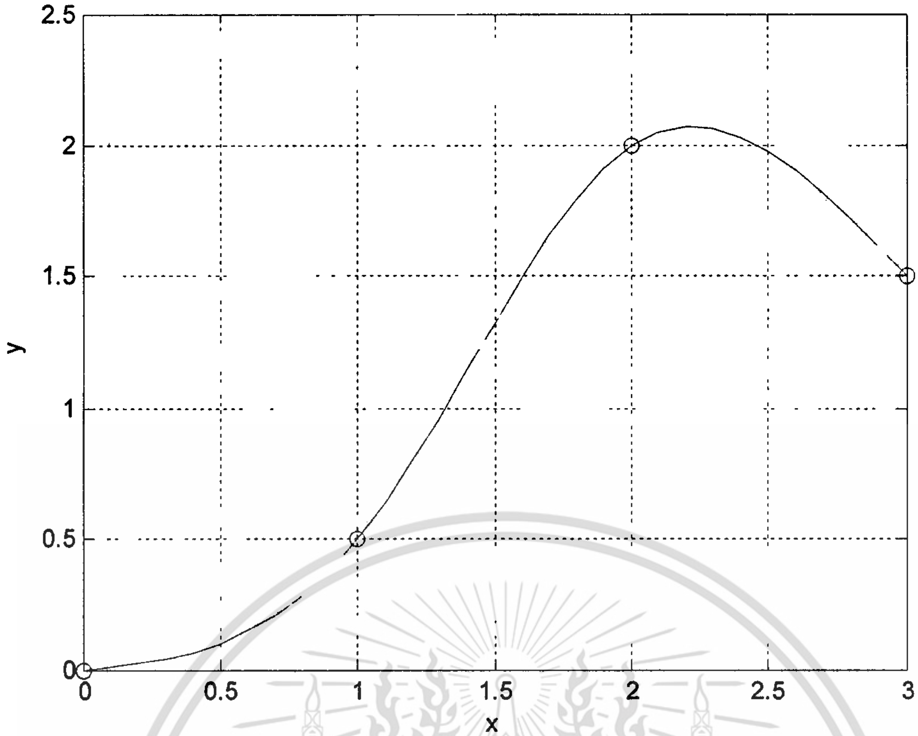


Figure 3.5 The natural spline with $s''(0) = 0$ and $s''(3) = 0$.

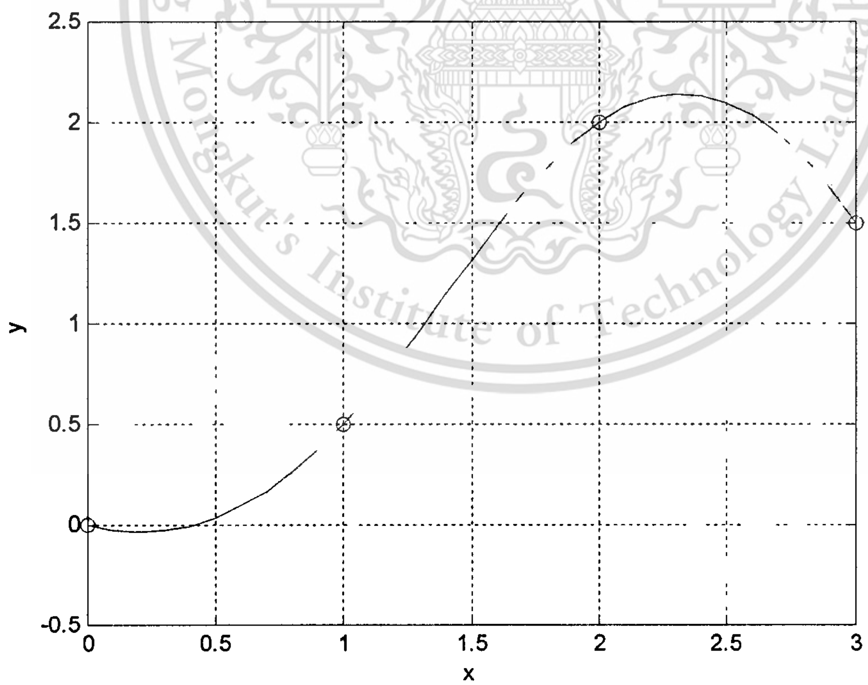


Figure 3.6 Parabolic Runout spline with $M_1 = M_2 = 1.75$ and $M_4 = M_3 = -2.75$.

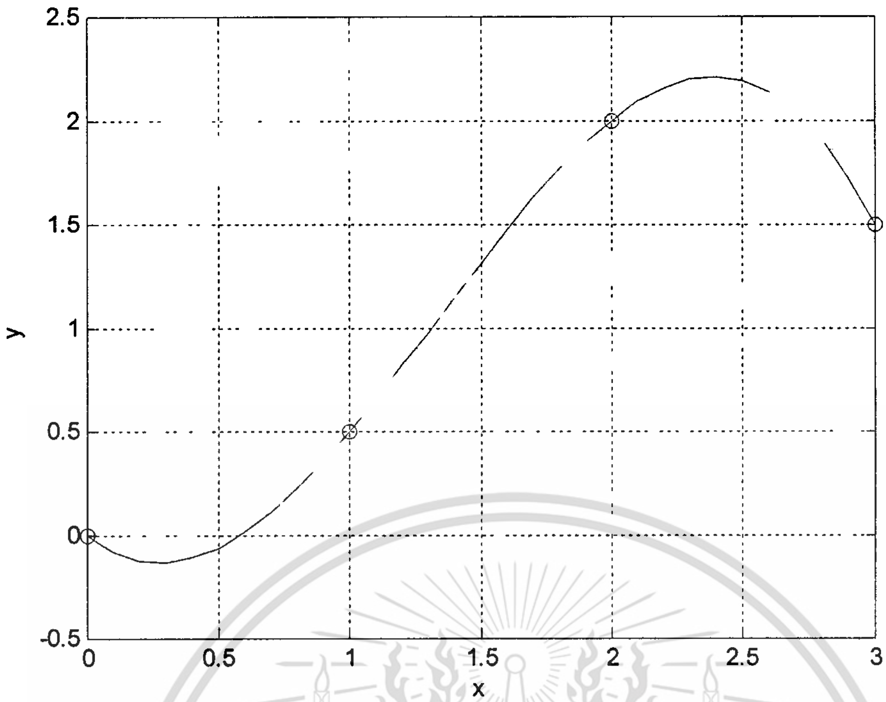


Figure 3.7 Cubic Runout spline curve with $M_1 = 4$ and $M_4 = -5$.

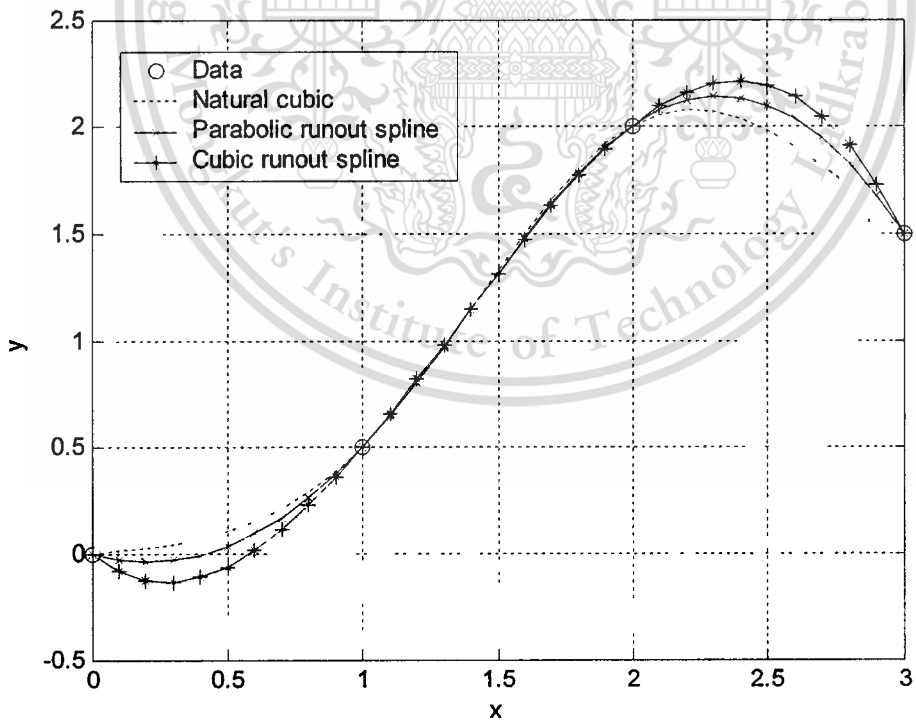


Figure 3.8 Comparison between three types of splines.

3.5 Curve fitting with splines

The main application of cubic spline interpolation techniques is, of course, curve fitting. To this end, the consistency and efficiency of the spline as a data correlation tool will be demonstrated.

3.5.1 Function Imitation

Cubic splines would not be necessary were it simple to determine a well-behaved function to fit any data set. This is, however, usually not the case. Thus, the cubic spline technique is used to generate a function to fit data. Moreover, it can be shown that data generated by a particular function is interpolated by a spline which behaves more or less like the original function. This is testimony to the consistency of splines.

For example, the following figure was generated using the function $y = \sin(x)$.

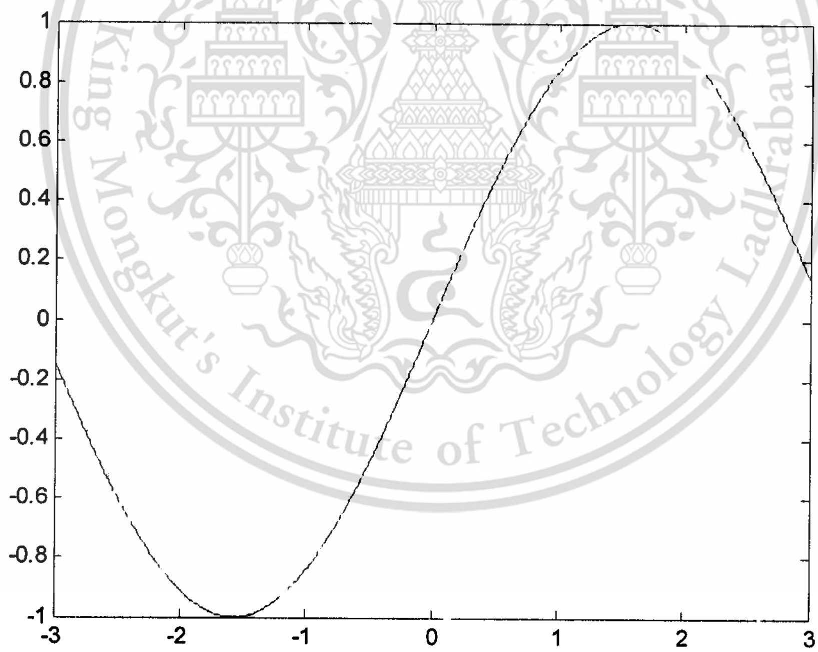


Figure 3.9 The graph of $y = \sin(x)$.

This next figure was generated by connecting seven data points along the above line with a cubic spline function.

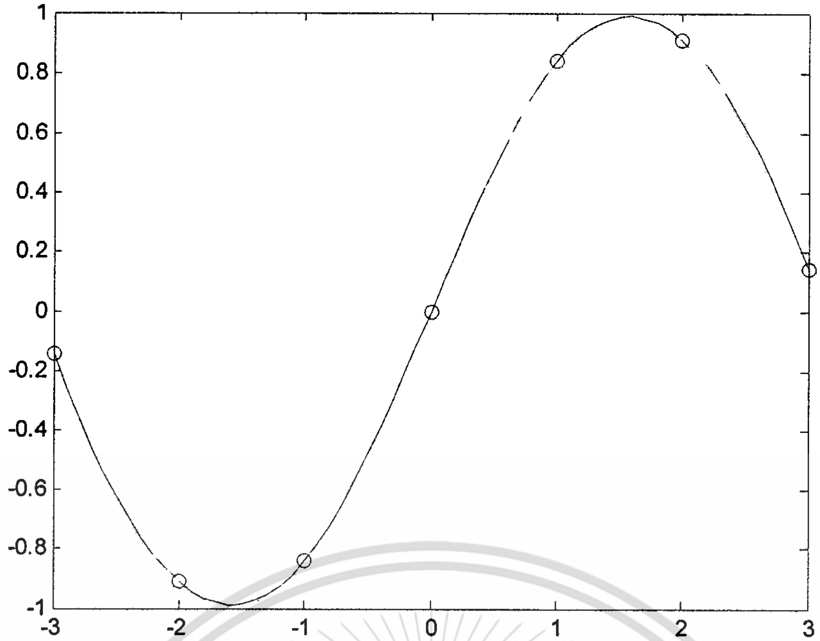


Figure 3.10 Spline through seven points on a sine curve.

By superimposing Figure 3.10 on Figure 3.9, we can see the degree to which the cubic curve imitates the original function $y = \sin(x)$.

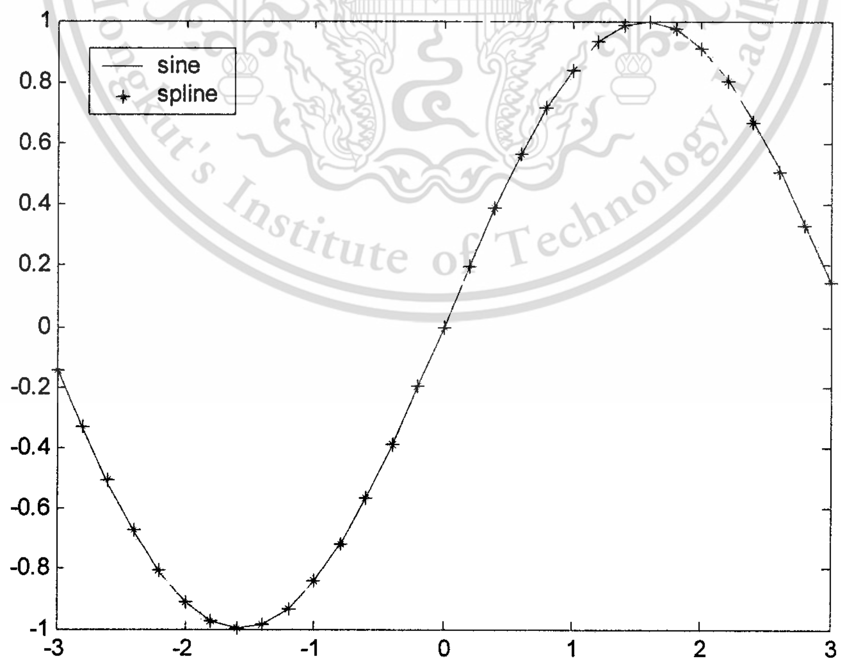


Figure 3.11 Superimposing of a spline curve on a sine function.

Clearly, the cubic spline curve closely imitates the sine curve. It has no extreme behavior between data points, and it effectively correlates the points.

3.6 Conclusion

Cubic spline interpolation is a powerful data analysis tool. Splines correlate data efficiently and effectively, no matter how data may seem. Once the algorithm for spline generation is produced, interpolating data with a spline becomes a simply task.

CHAPTER 4

NONLINEAR MODELING OF A SRM

4.1 Introduction

The switched reluctance machine (SRM) is considered to be a good competitor to conventional motors because of its simplicity, brushless, low cost and high efficiency. A great deal of research and development on the SRM has been reported during the last years [1]-[14]. Problems have been found in describing and controlling the motors because of the nonlinear nature of the SRM and high saturation of phase windings during high load. The flux linkage and phase inductance of SRM change with both the rotor position and the phase current. Therefore, the nonlinear model of SRM must be identified as a function of the phase current and rotor position. A primitive example is shown in Figure 4.1. This machine is denoted '12/8' because it has twelve stator poles and eight rotor poles. The two coils wound on opposite stator pole are excited simultaneously, and generate magnetic flux as shown.

In this chapter, a model of switched reluctance machine is presented. The model is based on polynomial function. The model was then used cubic spline interpolation technique to estimate the candidate coefficients function. Torque production and inductance profile is also proposed.

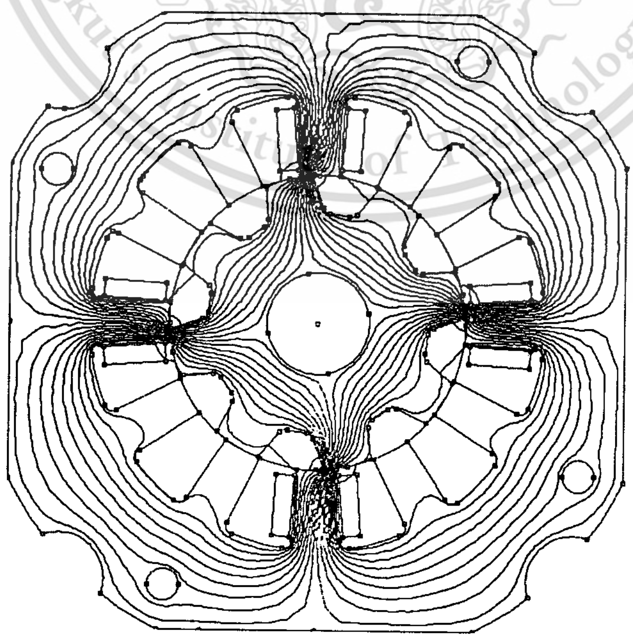
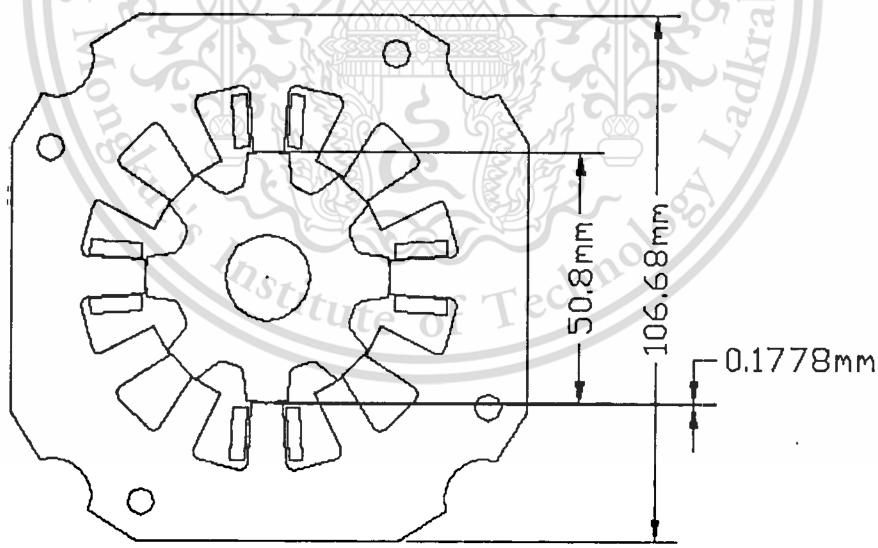


Figure 4.1 Simple switched reluctance machine with three-phase on twelve stator and eight rotor poles.

4.2 Flux linkage model

This section presents a nonlinear analysis of the three-phase SRM. Figure 4.2 shows the schematic diagram of a 12/8 SRM. The pole numbers of stator and rotor are 12 and 8 respectively, the core material is a stack of soft magnetic M19 silicon steel sheet with the length of 24.5 mm. Let the rotor position θ is 0° at aligned position and 22.5° mechanical degree when rotor is unaligned position. The magnetizing curves contain the data of flux linkage versus current at different rotor positions. A polynomial function of flux linkage is used to model these curves.

Figure 4.3 shows the flowchart of nonlinear analysis modeling. Finite Element Analysis (FEA) has been used to obtain the magnetic properties of the candidate machine. The area enclosed between the curve of flux linkage (λ) versus phase current (i) for unaligned and aligned positions of the rotor poles that shown in Figure 4.4 gives the maximum work done for one stroke of the motor. The electromagnetic torque can be calculated if the angular movement is known from the mechanical work. The data required for this procedure are the flux linkage versus stator excitation currents for discrete positions of the rotor.



Stator pole number	12	poles
Rotor pole number	8	poles
Stator pole arc, β_s	15	deg
Rotor pole arc, β_r	19	deg
Stack length	25.4	mm
Number of windings/pole	24	turns

Figure 4.2 The structure of a 12/8 SRM and geometry parameters.

Thus, the nonlinear model is developed from these data. Figure 4.4 shows that the curves are linear for small applied phase current or when the rotor is somewhere around the unaligned position. Then the candidate flux linkage function should contain a linear term. Similarly, the function also has nonlinear terms to represent the saturate effect. There are a few papers proposed nonlinear flux linkage function [5], [7], [8]. But the simplest form of polynomial function is shown in equation (4.1).

$$\lambda(i,\theta) = a_1(\theta)i + a_2(\theta)i^2 + a_3(\theta)i^3 \tag{4.1}$$

where the coefficients a_1, a_2 and a_3 are functions of rotor positions. The data of Figure 4.4 has been fit to equation (1) using MATLAB. The figures obtained from curve fitting may seen in Figure 4.5 – 4.7, respectively. The coefficient a_1 represents the unsaturated inductance of the coil while a_2 and a_3 represent the nonlinearity of the core.

Since a_1, a_2 and a_3 are functions of rotor positions, they can each be represented by a Fourier series [5], [9], but using this technique we have found that the finite number of harmonics used in the Fourier series has the effect of producing fictitious torque ripple on the static torque. The harmonics should be low enough so that there are no sign of significant torque ripple on the dynamic calculation. This is hard to find the suitable number of harmonics. Therefore, this paper proposed a model which uses cubic spline interpolation technique to solve this problem which is described in section 4.2.2.

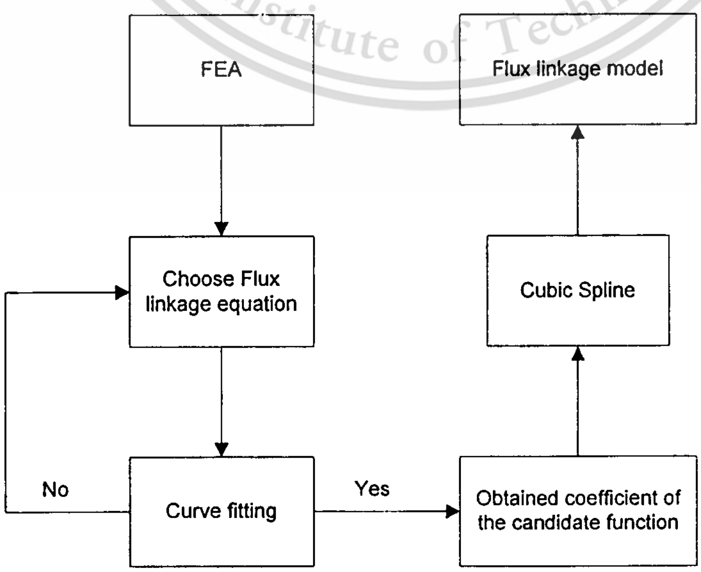


Figure 4.3 Flowchart of nonlinear modeling.

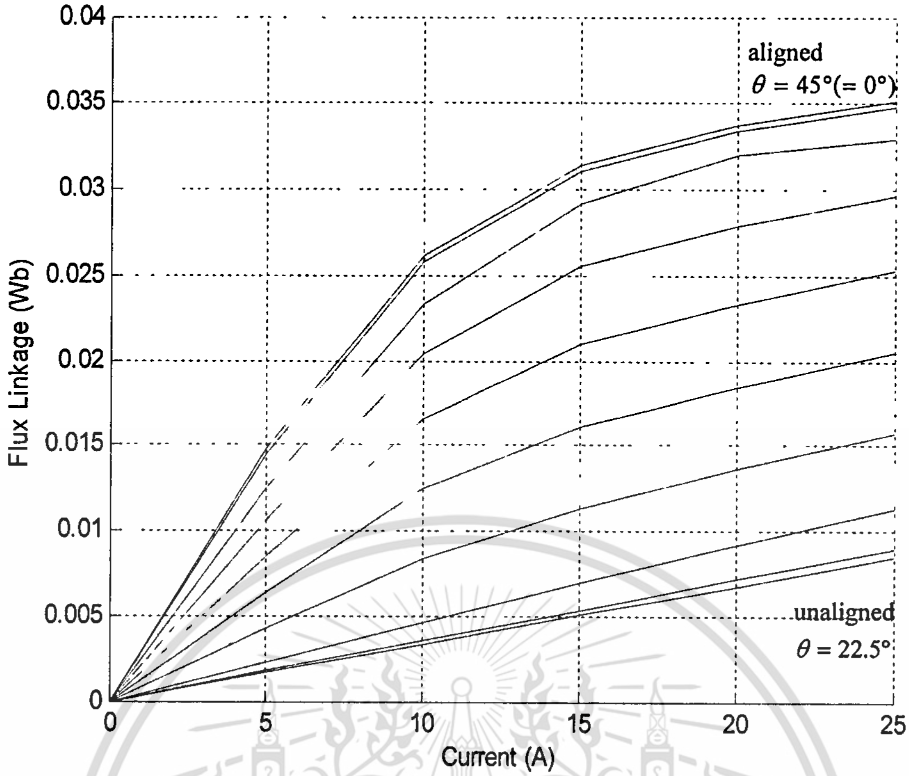


Figure 4.4 Magnetizing curves of a 12/8 SRM obtained from FEA.

4.2.1 Curve fitting

The art and science of creating a function definition that passes through a set of data points is often called curve fitting. The curve fitting process fits equations of approximating curves to the raw field data. Nevertheless, for a given set of data, the fitting curves of a given type are generally not unique. Thus, a curve with a minimal deviation from all data points is desired. This proposed-fitting curve can be obtained by the method of least squares which uses the minimal sum of the deviations squared from a given set of data. Least square error method determines the curve that best describes the relationship between expected and observed sets of data by minimizing the sums of the squares of deviation between observed and expected values. For more explanation of least square error, can been seen in [10].

Figures 4.5–4.7 show the result of the coefficients that of obtained from curve fitting while equation (4.1) that fits the magnetization curves. The comparison between curve fitting and magnetization curves are given in Figure 4.8.

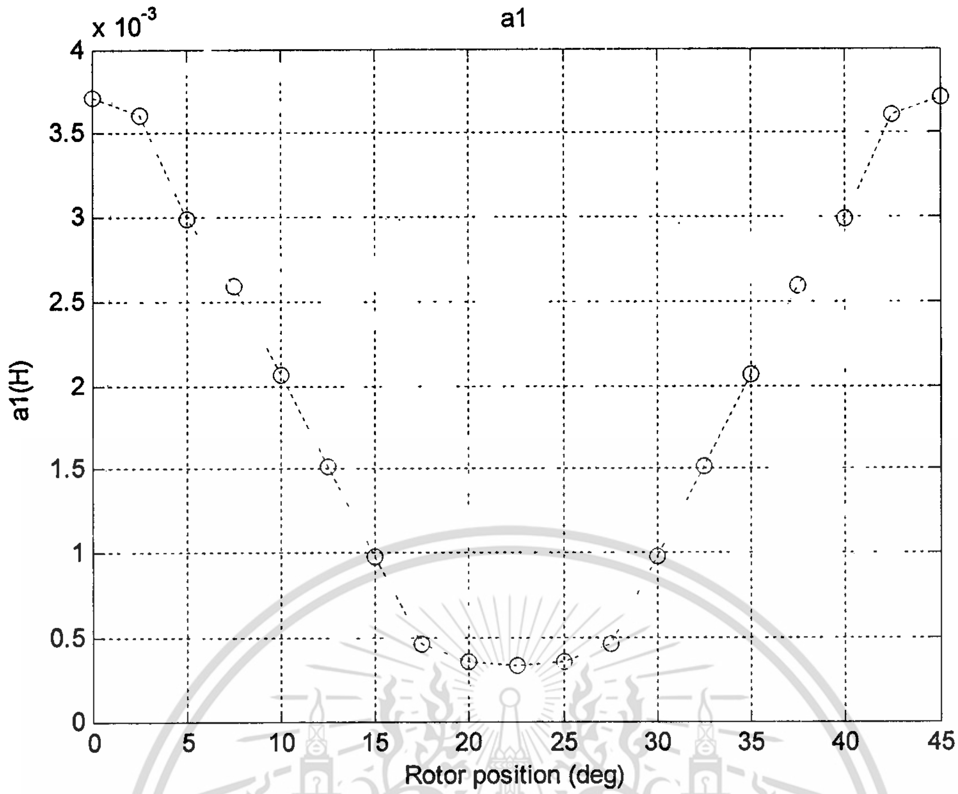


Figure 4.5 a_1 curve obtained from curve fitting.

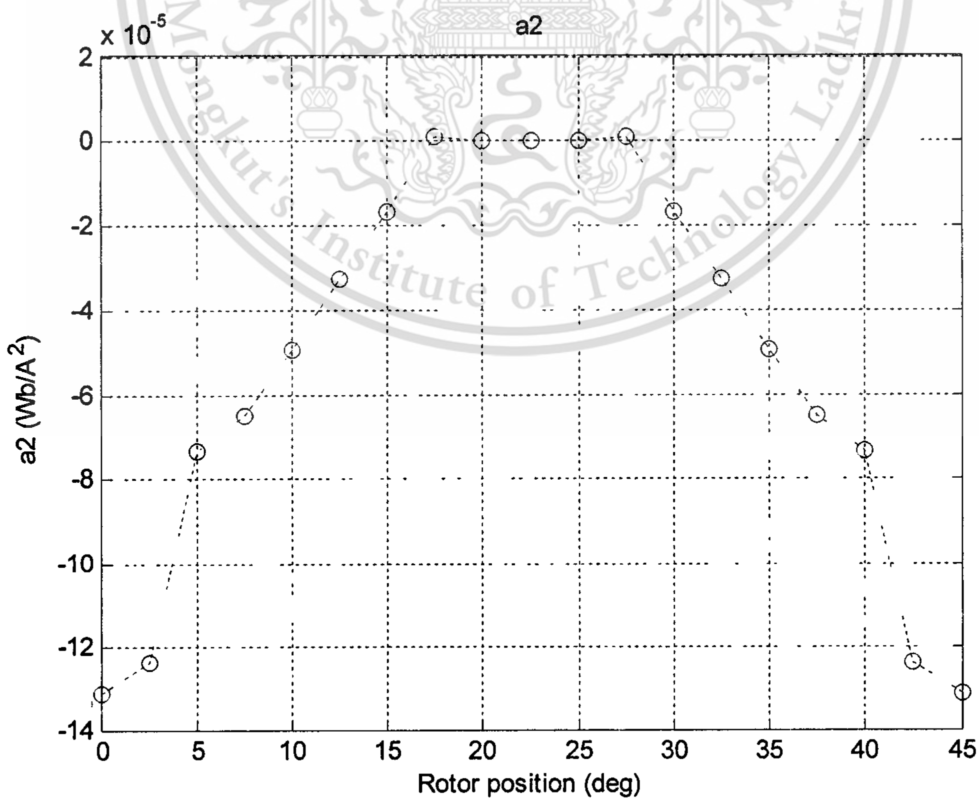


Figure 4.6 a_2 curve obtained from curve fitting.

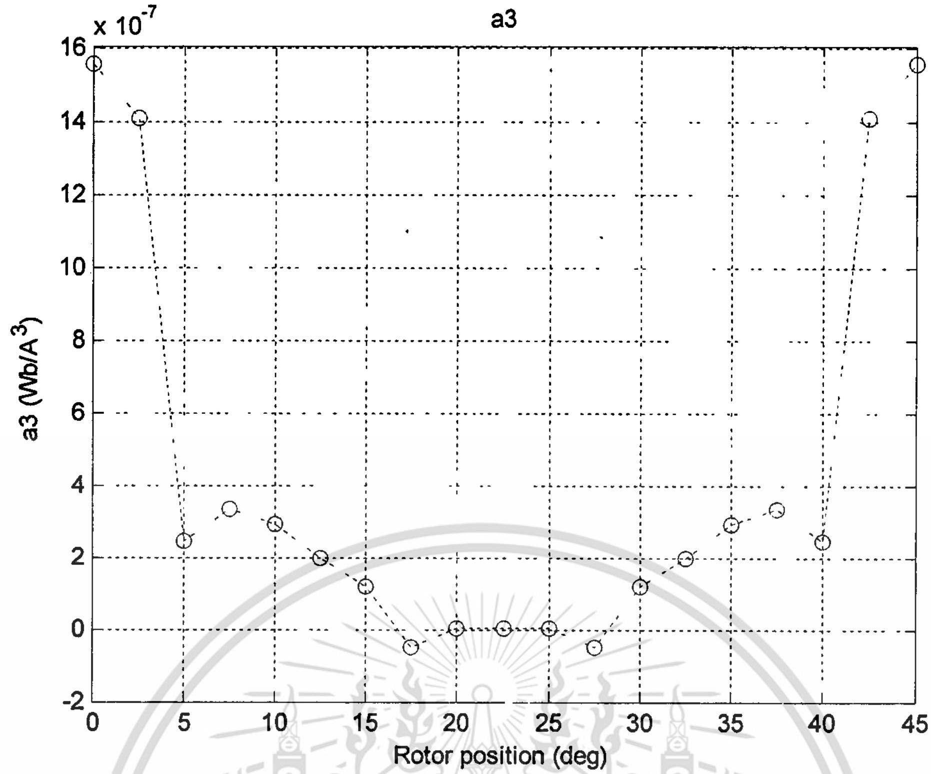


Figure 4.7 a_3 curve obtained from curve fitting.

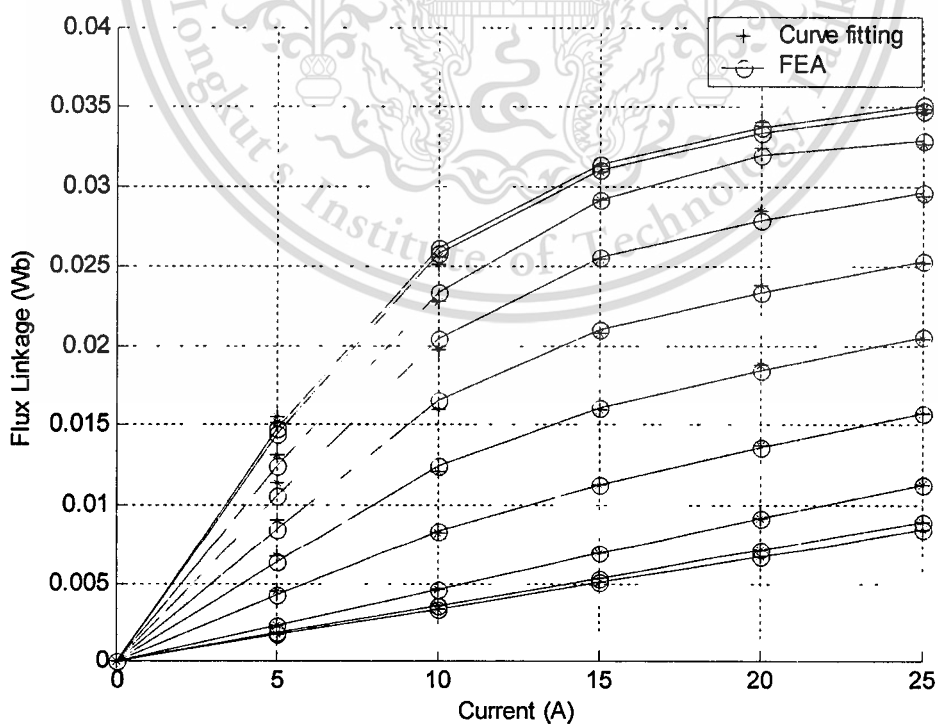


Figure 4.8 Comparison between curve fitting and raw data from FEA.

4.2.2 Cubic spline interpolation

The approach of estimation of a_1, a_2 and a_3 is introduced in [5]. However it is not good enough to implement. So, this section has introduced a method of cubic spline interpolation technique. The cubic spline interpolation is already explained in Chapter 3. So, for this section, it will be described throughout briefly. For this implementation, we use natural spline to interpolate the flux linkage model. The reason is that the curvatures at the endpoints of coefficients a_1, a_2 and a_3 are not high and the error is acceptable.

Let the i_{th} piece of the spline be represented by:

$$Y_i(x) = A_i + B_i x + C_i x^2 + D_i x^3 \tag{4.2}$$

where

A_i, B_i, C_i, D_i are spline coefficients.

x is parameter $\in [0, 1]$.

If the curve has m points, so it would have m-1 pieces. The matrix size of the coefficients A_i , B_i , C_i and D_i is $(m-1)\times 4$. Using this technique applies to the coefficients a_1, a_2 and a_3 curves that obtained from curve fitting. These curves are divided into 2.5° mechanical degree interval from aligned position to the next aligned position. There are total 18 pieces and the spline coefficients are (18×4) for each coefficient a_1, a_2 and a_3 as shown in Table 4.1.

Table 4.1 Spline coefficients of $a_1(\theta)$, $a_2(\theta)$, $a_3(\theta)$.

$a_1(\theta)$				$a_2(\theta)$				$a_3(\theta)$			
A_i	B_i	C_i	D_i	A_i	B_i	C_i	D_i	A_i	B_i	C_i	D_i
-1.01E-05	0	2.33E-05	3.71E-03	9.54E-07	0	-3.08E-06	-1.31E-04	-2.34E-08	0	8.82E-08	1.55E-06
1.73E-05	-7.61E-05	-1.67E-04	3.61E-03	-1.97E-06	7.15E-06	1.48E-05	-1.24E-04	5.16E-08	-1.75E-07	-3.50E-07	1.41E-06
-1.13E-05	5.38E-05	-2.23E-04	2.99E-03	1.40E-06	-7.62E-06	1.36E-05	-7.31E-05	-3.77E-08	2.12E-07	-2.59E-07	2.44E-07
4.66E-06	-3.06E-05	-1.65E-04	2.59E-03	-4.50E-07	2.88E-06	1.78E-06	-6.48E-05	1.08E-08	-7.11E-08	9.37E-08	3.33E-07
-6.62E-08	4.40E-06	-2.30E-04	2.06E-03	3.13E-08	-4.92E-07	7.75E-06	-4.94E-05	-6.77E-10	9.87E-09	-5.93E-08	2.92E-07
-2.27E-06	3.90E-06	-2.09E-04	1.51E-03	1.70E-07	-2.57E-07	5.87E-06	-3.26E-05	-3.34E-09	4.79E-09	-2.26E-08	1.95E-07
9.13E-06	-1.31E-05	-2.32E-04	9.78E-04	-5.06E-07	1.02E-06	7.78E-06	-1.69E-05	7.37E-09	-2.03E-08	-6.14E-08	1.16E-07
-8.23E-06	5.54E-05	-1.27E-04	4.58E-04	4.94E-07	-2.77E-06	3.41E-06	1.07E-06	-6.91E-09	3.50E-08	-2.46E-08	-4.93E-08
1.91E-06	-6.35E-06	-4.13E-06	3.58E-04	-1.87E-07	9.36E-07	-1.17E-06	7.17E-09	3.36E-09	-1.68E-08	2.10E-08	1.82E-10
-1.91E-06	8.00E-06	0.00E+00	3.38E-04	1.87E-07	-4.67E-07	1.39E-22	3.72E-09	-3.36E-09	8.38E-09	1.46E-24	2.11E-10
8.23E-06	-6.35E-06	4.13E-06	3.58E-04	-4.94E-07	9.36E-07	1.17E-06	7.17E-09	6.91E-09	-1.68E-08	-2.10E-08	1.82E-10
-9.13E-06	5.54E-05	1.27E-04	4.58E-04	5.06E-07	-2.77E-06	-3.41E-06	1.07E-06	-7.37E-09	3.50E-08	2.46E-08	-4.93E-08
2.27E-06	-1.31E-05	2.32E-04	9.78E-04	-1.70E-07	1.02E-06	-7.78E-06	-1.69E-05	3.34E-09	-2.03E-08	6.14E-08	1.16E-07
6.62E-08	3.90E-06	2.09E-04	1.51E-03	-3.13E-08	-2.57E-07	-5.87E-06	-3.26E-05	6.77E-10	4.79E-09	2.26E-08	1.95E-07
-4.66E-06	4.40E-06	2.30E-04	2.06E-03	4.50E-07	-4.92E-07	-7.75E-06	-4.94E-05	-1.08E-08	9.87E-09	5.93E-08	2.92E-07
1.13E-05	-3.06E-05	1.65E-04	2.59E-03	-1.40E-06	2.88E-06	-1.78E-06	-6.48E-05	3.77E-08	-7.11E-08	-9.37E-08	3.33E-07
-1.73E-05	5.38E-05	2.23E-04	2.99E-03	1.97E-06	-7.62E-06	-1.36E-05	-7.31E-05	-5.16E-08	2.12E-07	2.59E-07	2.44E-07
1.01E-05	-7.61E-05	1.67E-04	3.61E-03	-9.54E-07	7.15E-06	-1.48E-05	-1.24E-04	2.34E-08	-1.75E-07	3.50E-07	1.41E-06

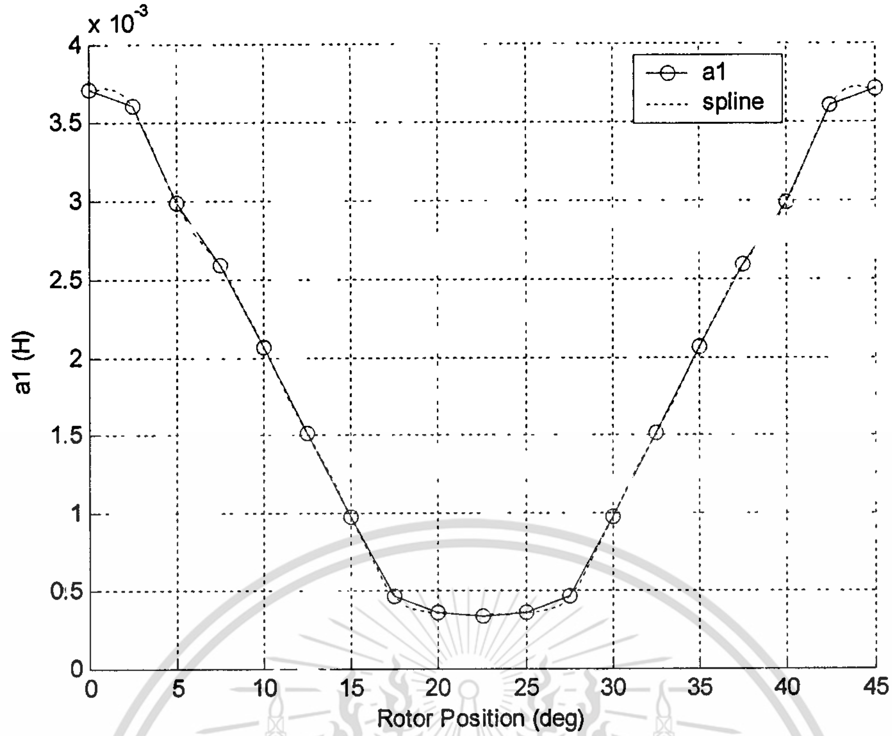


Figure 4.9 Comparison between cubic spline curve (dashed line) and $a_1(\theta)$ curve (solid line).

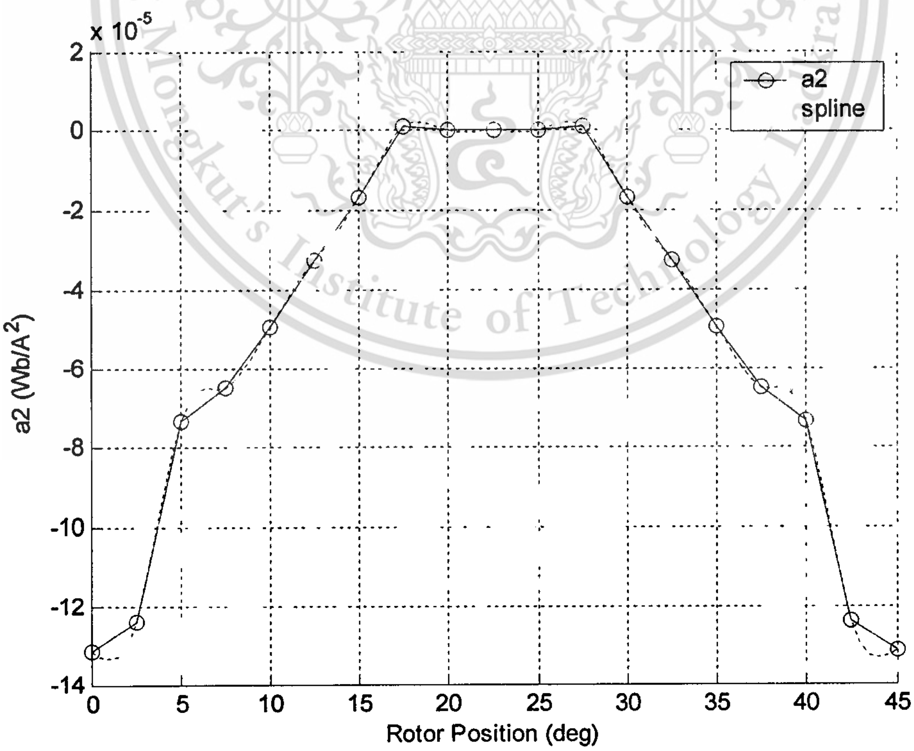


Figure 4.10 Comparison between cubic spline curve (dashed line) and $a_2(\theta)$ curve (solid line).

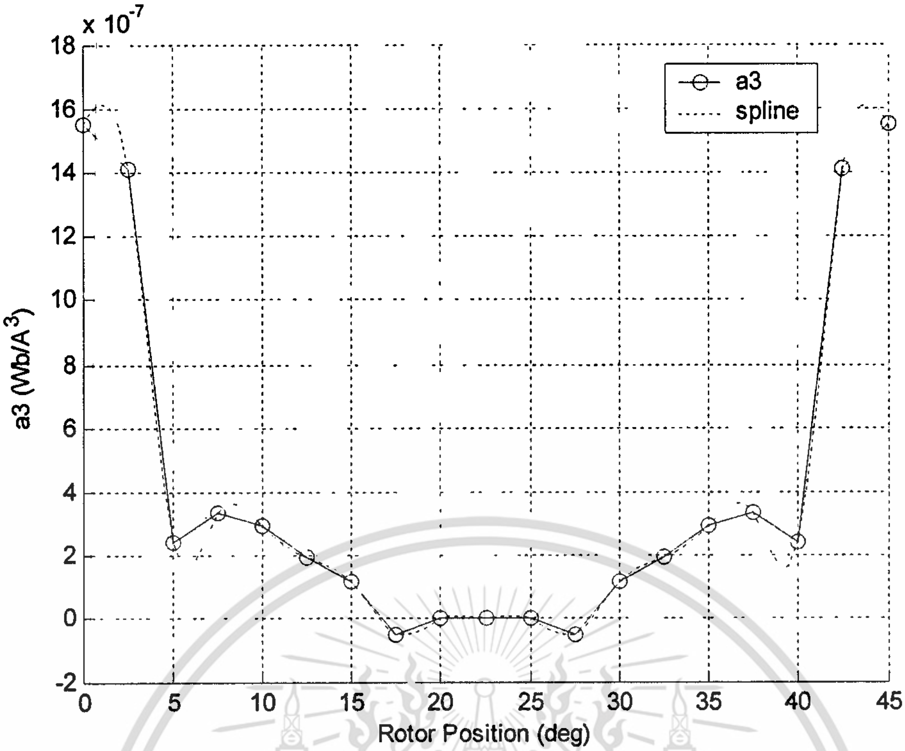


Figure 4.11 Comparison between cubic spline curve (dashed line) and $a_3(\theta)$ curve (solid line).

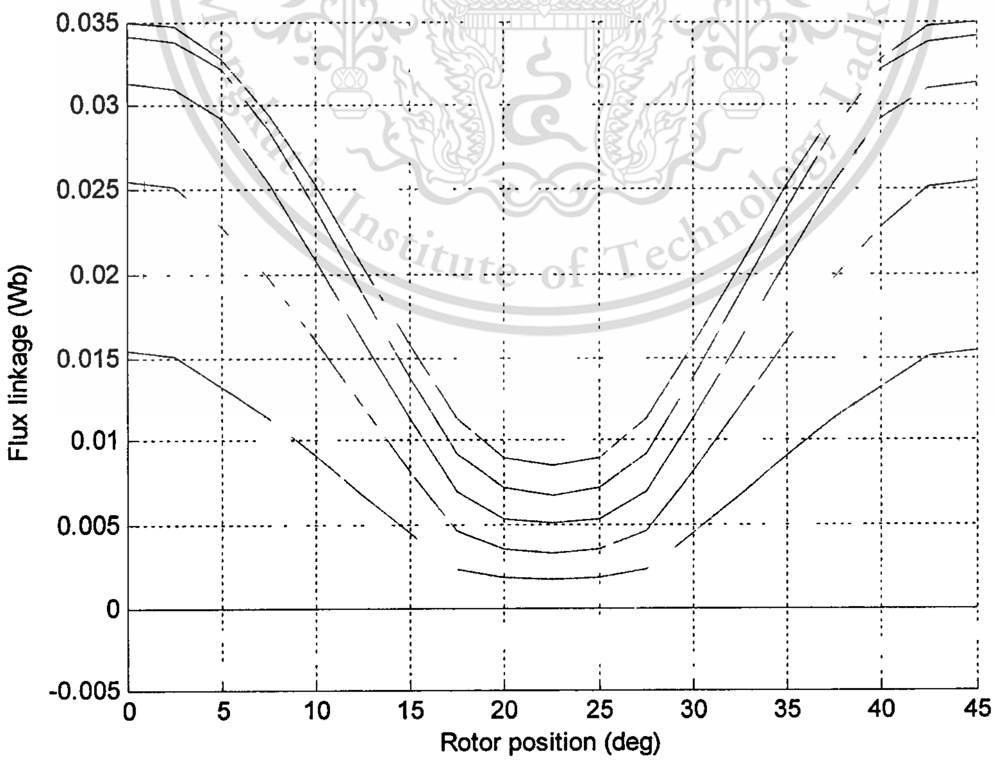


Figure 4.12 Flux linkage obtained from flux linkage model.

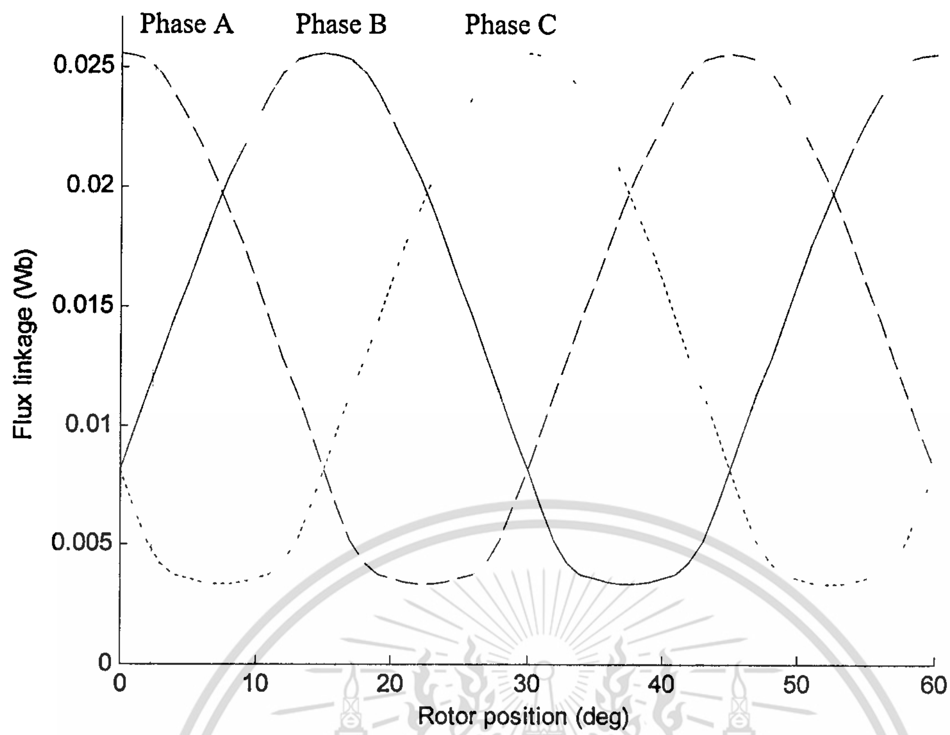


Figure 4.13 Plot of three-phase flux linkage at 15A.

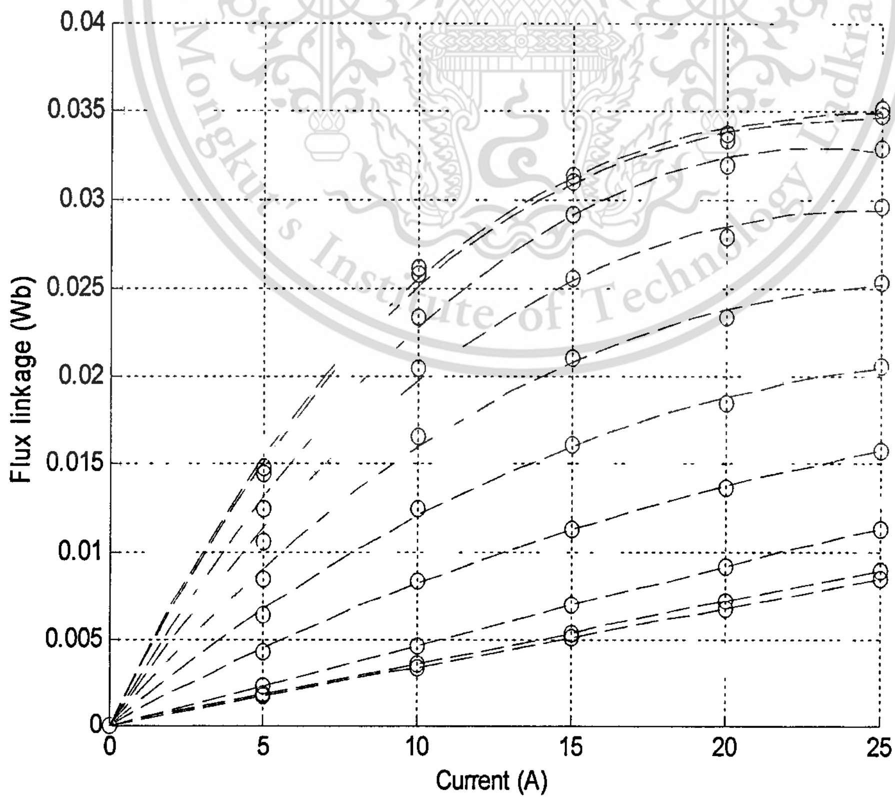


Figure 4.14 Comparison between results from FEA and calculated magnetizing curves at different rotor positions.

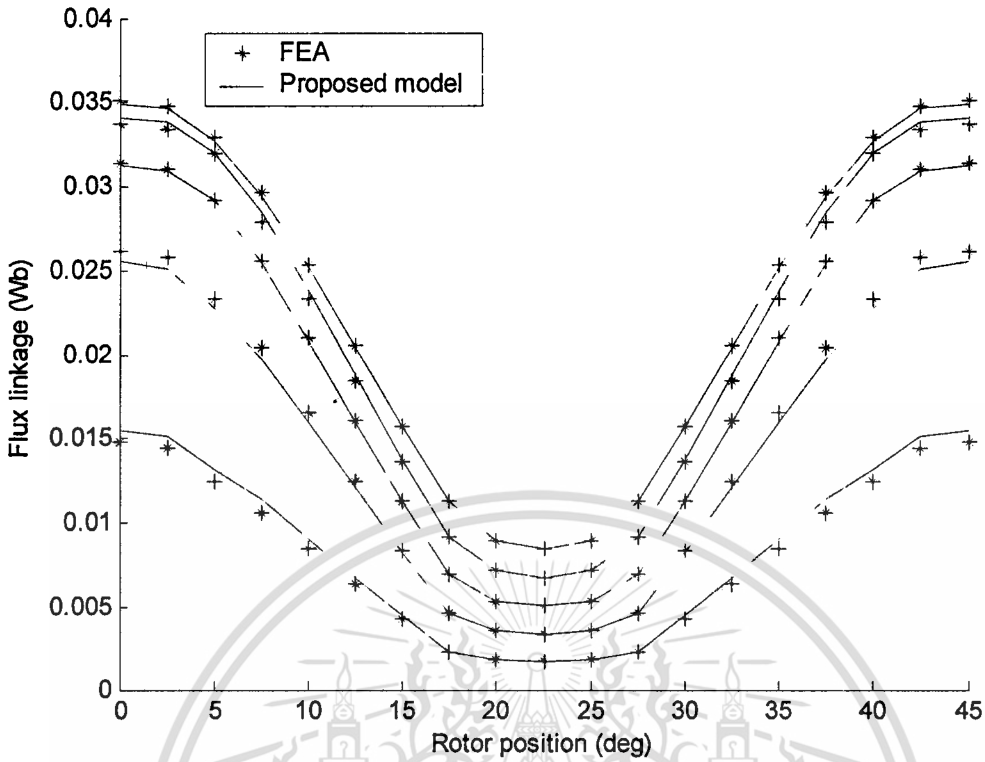


Figure 4.15 Comparison between proposed model and FEA corresponding to the rotor positions.

Figures 4.9, 4.10 and 4.11 depict the comparison between the curves of coefficients a_1, a_2 and a_3 from curve fitting model and the reconstruction of those coefficients from cubic spline interpolation technique. Figure 4.12 shows the proposed flux linkage model versus rotor position from aligned position to next aligned position while Figure 4.13 depicts the overlap between phase flux linkages at 15A. For this proposed machine, the overlap between phases is 7.5 mechanical degrees. The measured magnetizing curve from FEA with step 2.5° mechanical degree in rotor position and the reconstructed curves from cubic spline interpolation are shown in Figure 4.14 and Figure 4.15. The results are good agreement. However, there is small noticeable error which is less than 5%.

4.3 Torque modeling of SRM

Non-linear mapping techniques like lookup tables or artificial neural networks can be used to propose SRM torque in terms of the phase current and rotor position, for estimating SRM online. Lookup table requires large amounts of online storage spaces. On the other hand, artificial neural networks require large amounts of online computation. Hence, such methods are not appropriate for real time controller implementation [11]. A compact analysis torque expression based on physical principles of SRM operation would be the most efficient for real time implementation. However, looking at the measured torque data at different phase currents and rotor position, no intuitive model can be thought of for representing the torque as an analysis function [12].

Torque production of a SRM is represented as the position rate of change of co-energy at given current and rotor position. For the nonlinear SRM, torque production for each phase is given by:

$$T = \left.\frac{\partial W_c(i,\theta)}{\partial \theta}\right|_{i=const} \tag{4.3}$$

where

$$W_c = \left.\int_0^i \lambda(i,\theta).di\right|_{\theta=const} \tag{4.4}$$

Thus, the phase torque can be represented as:

$$T = \frac{\partial}{\partial \theta}\int_0^i \lambda(i,\theta)di \tag{4.5}$$

Substitution of equation (4.1) to (4.5) gives:

$$T = \frac{1}{2}i^2\frac{da_1(\theta)}{d\theta} + \frac{1}{3}i^3\frac{da_2(\theta)}{d\theta} + \frac{1}{4}i^4\frac{da_3(\theta)}{d\theta} \tag{4.6}$$

This torque equation represents only one phase. For the total SRM torque, it is simply summing up individual phase torque with appropriate adjustment for spatial displacement.

From equation (4.6), the torque equation comprises of the derivative of the coefficients a_1, a_2 and a_3. The cubic spline interpolation technique is implemented by taking derivative of equation (4.2). The derivative of each piece of spline becomes equation (4.7).

$$Y_i(x) = B_i + 2C_i x + 3D_i x^2 \tag{4.7}$$

Using equation (4.7) and the coefficients in Table 1 with equation (4.6) yield the torque value at particular position. The result of calculating of static torque is shown in Figure 4.16. The static torque was computed using equation (4.6) for the rotor position starting from aligned position to the next aligned position. Figure 4.17 shows the static torque when using the torque equation from equation (4.6) and FEA. The results show that there are very good fit.

Figure 4.18 shows the total electromagnetic torque of SRM from different currents level which starts form 0A – 25A respectively. It shows that the curves contribute high ripples, especially at high current.

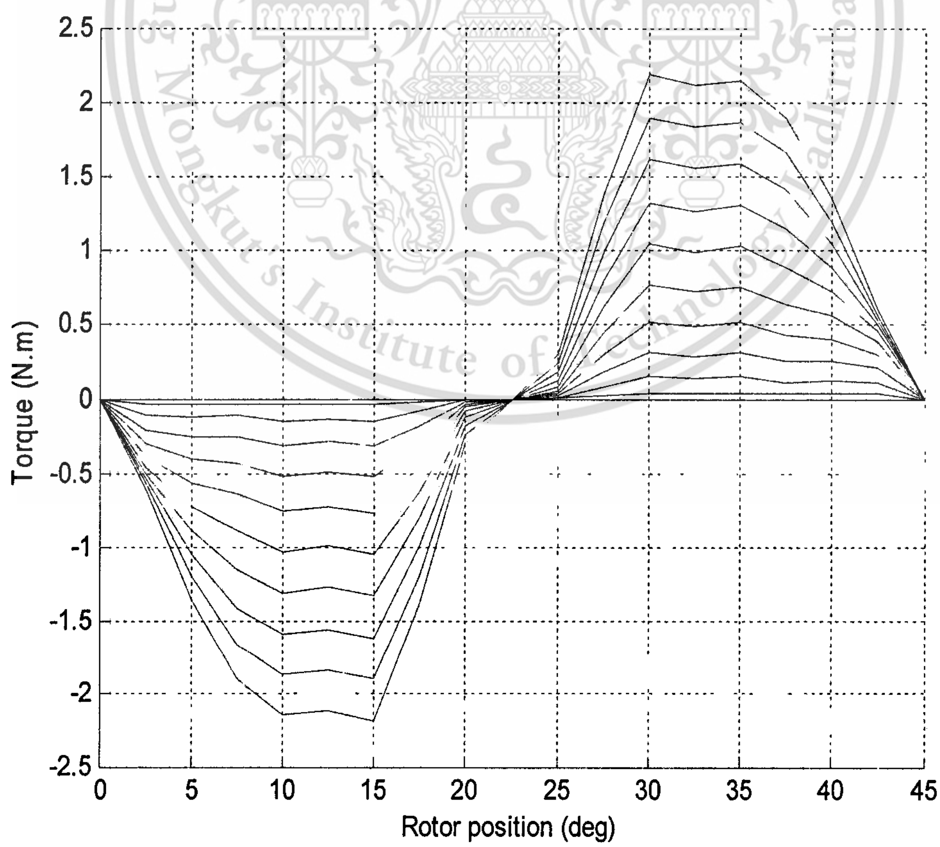


Figure 4.16 Static torques for different phase current levels.

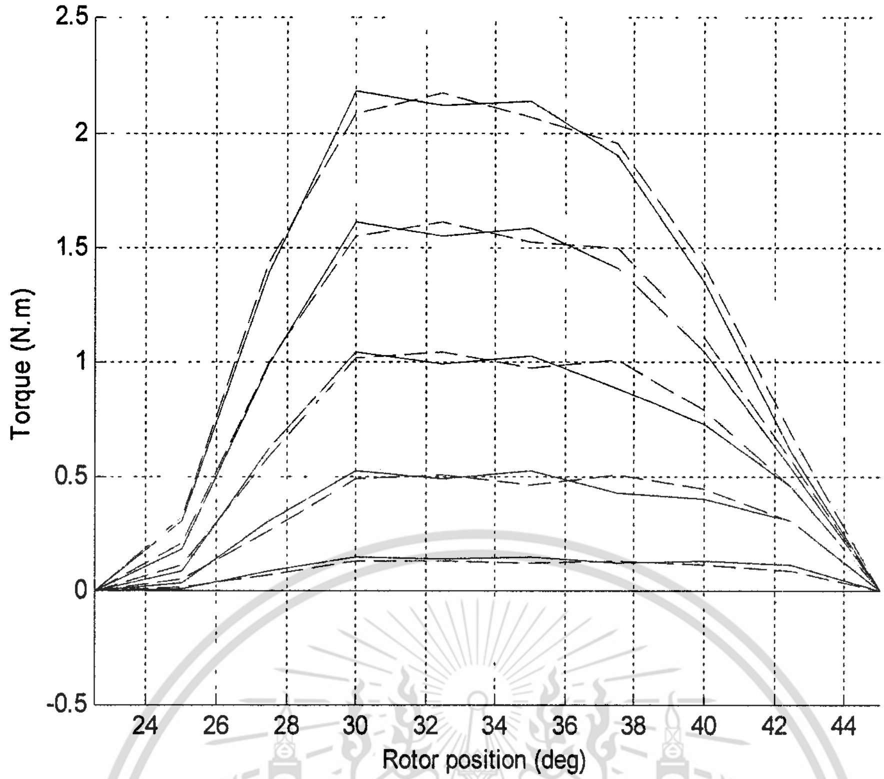


Figure 4.17 Static torque obtained from FEA (dashed line) and from torque equation (solid line) at different current levels.

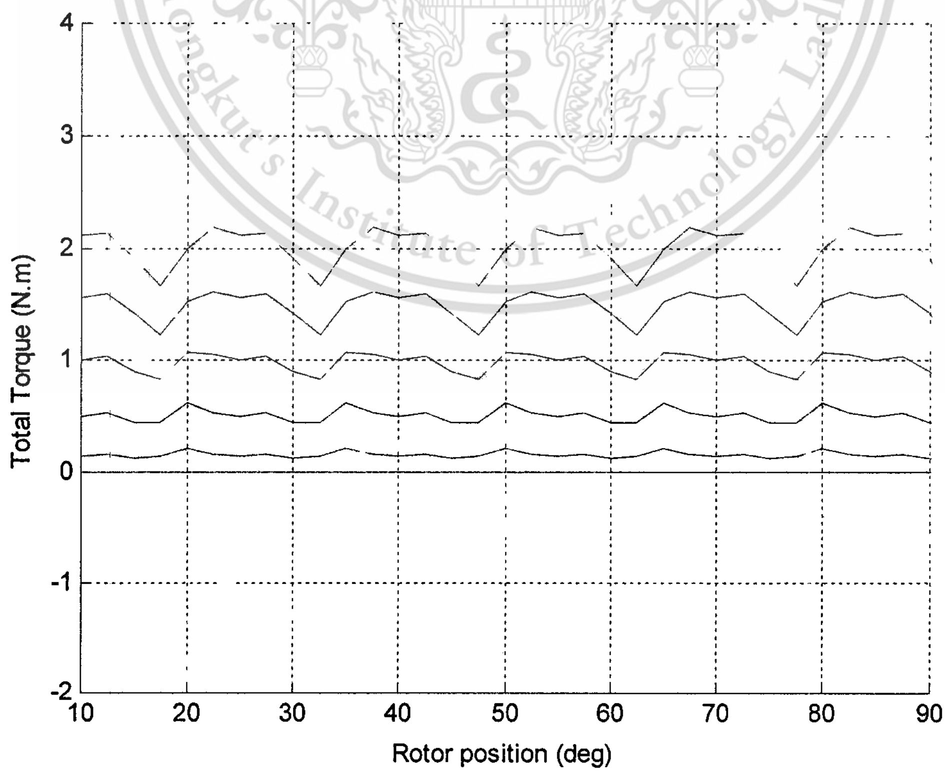


Figure 4.18 Total electromagnetic torque at different currents levels.

4.4 Inductance profile modeling

The voltage across the terminals of a phase of an SRM winding is related to the flux linkage in the winding by Faraday's laws as

$$v = R.i + \frac{d\lambda(i,\theta)}{dt} \tag{4.8}$$

where v is the terminal voltage, i is the phase current, R is the phase winding resistance, and λ is the phase flux linked by the winding.

Because of the double salient construction of the SRM and the magnetic saturation effects, the flux linked in a SRM phase varies as a function of rotor position θ and the phase current. Equation (4.8) can be rewrite as

$$v = R.i + \frac{\partial\lambda(i,\theta)}{\partial i}.\frac{di}{dt} + \frac{\partial\lambda(i,\theta)}{\partial\theta}.\frac{d\theta}{dt} \tag{4.9}$$

$$v = R.i + L(i,\theta).\frac{di}{dt} + e(i,\theta,\omega) \tag{4.10}$$

where

$$e = \omega\frac{\partial\lambda(i,\theta)}{\partial\theta}$$
$$e = \omega.i.\frac{dL(i,\theta)}{d\theta} \tag{4.11}$$

where L is the phase inductance, θ is the rotor position, ω is the angular velocity, and e is the back electromotive force (EMF). Since the phase inductance can be calculated from the equation (4.12)

$$L(i,\theta) = \frac{\partial\lambda(i,\theta)}{\partial i} \tag{4.12}$$

Substitution equation (4.1) to (4.10) gives a new inductance as

$$L(i,\theta) = a_1(\theta) + 2ia_2(\theta) + 3i^2a_3(\theta) \tag{4.13}$$

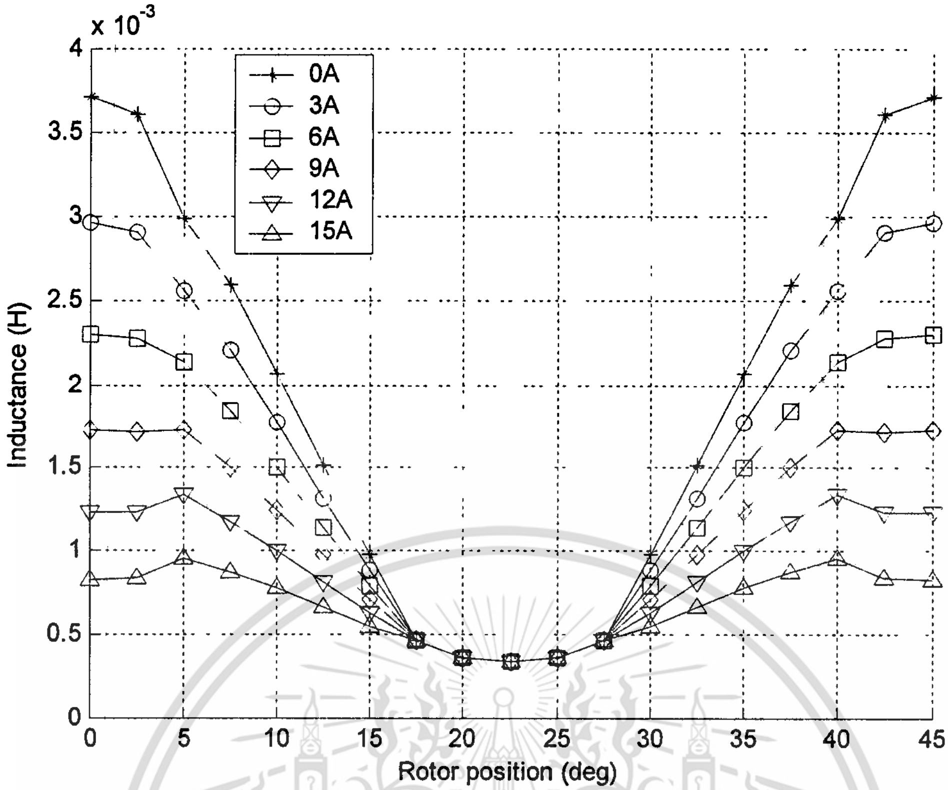


Figure 4.19 Calculated valued of inductance profile.

From (4.13), the inductance profile of SRM comprises of the coefficient a_1, a_2 and a_3. Figure 4.19 shows the result of calculating of inductance using (4.13). It is shown that the inductance in the high currents lower than inductance in low currents.

4.5 Conclusion

In this work, polynomial flux linkage modeling of SRM was presented that used cubic spline interpolation technique to find the value of the candidate coefficients of flux linkage function. The effectiveness of proposed model was verified by comparison with FEA. The model was then extended to be torque production and inductance profile of SRM. The polynomial functions of flux linkage, torque and inductance profile are very simple and can be used in various applications such as: sensor less control and simulation.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

A simplified nonlinear model of SRM is developed under reasonable assumptions to characterize the inherent relationship among current, position, flux, inductance and torque. The model only requires input data which may obtain by measurement or finite element analysis (FEA) to be generated as model. Thus, in this thesis, FEA is implemented for data collection.

This research covers the design procedure of a 12/8 switched reluctance machine from the first principles. The nonlinear designed procedure is based on magnetization curves of the SRM. The design is focus on the polynomial flux linkage model which is simple for person with basic knowledge of electromagnetic to achieve an intuitive procedure. The methodical approach to design the model of the machine is one of the major contributions of this research.

The procedure for design the model of flux linkage corresponding to phase current and rotor position has been developed. It follows the basic principle of electromagnetic and does not rely on any previous sources whose documentation is hard to verify. Cubic spline method is implemented. Thus, makes the flux linkage model become easy task. The result shows a good accuracy when compares with the calculated flux linkage model and other method as finite element analysis.

The process of torque production is described. The function of torque production is one of the signification contributions of this research. We found that both flux linkage and torque functions are polynomials. Thus, the calculation of them is not complicated as described in [13], [14]. Moreover, one of the advantages of this research is that the spline coefficients can implement into several models that presented in this thesis such as: flux linkage, electromagnetic torque, and inductance profile.

An improved analytical method for determining the inductance profile of a switched reluctance motor has been proposed. This will be the guidance for a person who does about sensorless control of SRM.

5.2 Future work

The SRM is increasingly gained attention to researchers in recent years. There has been intensive research work on this type of motor, but there are still a lot of issues to be addressed.

The SRM is regarded as a high torque ripple motor. A smooth torque can be achieved but with complicated algorithms, and in a dynamic response simulation, a smooth torque is difficult to achieve. The develop of simple and reliable control strategies will allow SRM drives to further penetrate marketplace.

In addition, all models proposed in this thesis have been implemented in MATLAB simulation. So, to verify the model accuracy, the model should be implemented in experimental conditions.

LITERATURE CITED

[1] Radimov. N, Ben-Hail. N, Rabinovici. R, "Simple Model of Switched-Reluctance Machine Based Only on Aligned and Unaligned Position Data" IEEE Transactions on Magnetic, Volume 40, Issue 3, May 2004. Page(s):1562-1572.

[2] T. J. E. Miller, "Switched Reluctance Motors and Their Control," Oxford, U.K. Clarendon, 1993.

[3] T. J. E. Miller, "Electronic Control of Switched Reluctance Machines," Oxford.

[4] Delores M. Etter, "Engineering Problem Solving with MATLAB," University of Colorado at Boulder. Second Edition.

[5] S. Kittiratsatcha, D. A. Torrey, "A Design Method of a Two-phase Switched-Reluctance Machine," Power Electronics and Drive Systems, 2003. PEDS 2003. The Fifth International Conference on Volume 2, 17-20 Nov. 2003 Page(s):1599 – 1604.

[6] Wenzhe Lu, Keyhani. A, Fardoun. A, "Neural Network-Based Modeling and Parameter Identification of Switched Reluctance Motors," Energy Conversion, IEEE Transactions on Volume 18. Issue 2, June 2003 Page(s):284 – 290.

[7] Liu Shanshan, Zhao Zhengming, Meng Shuo, Chai Jianyun, "A Non-Linear "Analysis Model for Switched Reluctance Motor," TENCON '02. Proceedings. 2002 IEEE Region 10 Conference on, Communications, Control and Power Engineering Volume 3, 28-31 Oct. 2002. Page(s):2034 - 2037 vol.3.

[8] D. A. Torrey, "Switched Reluctance Generators and Their Control," Industrial Electronics, IEE Transactions on Volume 49, Issue 1, Feb. 2002 Page(s):3 – 14.

[9] Kano. Y, Kosaka. T, Matsui. N, "Magnetization characteristics analysis of SRM by simplified non-linear magnetic analysis," Power Conversion Conference, 2002. PCC Osaka 2002. Proceedings of the Volume 2, 2-5 April 2002 Page(s):689 - 694 vol.2.

[10] Steven C. Chapra, Raymond P. Canale, "Numerical Methods for Engineers," International Edition, fourth edition.

[11] S. K Sahoo, S. K. Panda, and J. X. Xu "Model-base Torque Estimator for Switched Reluctance Motors," Power Electronics and Drive Systems, 2003. PEDS 2003. The Fifth International Conference on Volume 2, 17-20 Nov. 2003 Page(s):959 - 963 Vol.2.

[12] Stiebler. M, Ke Liu, "An analytical model of switched reluctance machines," Energy Conversion, IEEE Transactions on Volume 14, Issue 4, Dec. 1999 Page(s):1100 – 1107.

[13] Mir. S, Husain. I, Elbuluk. M.E, "Switched Reluctance Motor Modeling with On-Line Parameter Identification," Industry Applications, IEEE Transactions on volume 34, Issue4, July - Aug, 1998.Page(s):776-783.

[14] Torrey. D.A, Lang. J.H, "Modelling a nonlinear variable-reluctance motor drive," Electric Power Applications, IEE Proceedings B. Volume 137, Issue 5, Sept, 1990 Page(s):314 – 326.

APPENDICES

Appendix A

The Publication

S. Khotpanya, S. Kittiratsatcha, Ishibashi Kazuhisa, "A Magnetic Model of a Three-Phase Switched Reluctance Machine using Cubic Spline Interpolation Technique," IEEE 2005 International Conference on Power Electronics and Drive Systems (PEDS 2005), 28 Nov – 1 Dec, 2005. Kuala Lumpur, Malaysia. Page(s): 1167-1170.

The Sixth International Conference on
Power Electronics and Drive Systems
PEDS 2005

PEDS

Proceedings

28 Nov – 1 Dec 2005
Renaissance Hotel
Kuala Lumpur, Malaysia

Organiser:

Universiti Teknologi Malaysia

Co-organisers:

IEEE PELS/IAS/IES Joint Chapter, Malaysia Section
and
IEEE IAS/PELS Joint Chapter, Singapore Section

Technical Co-Sponsors:

IEEE Industry Applications Society

and

IEEE Power Electronics Society

A Magnetic Model of a Three-Phase Switched-Reluctance Machine using Cubic Spline Interpolation Technique

S. Khotpanya, S. Kittiratsatcha
Department of Electrical Engineering. Faculty of Engineering.
King Mongkut's Institute of Technology Ladkrabang,
Bangkok 10520, Thailand.

Ishibashi Kazuhisa
Department of Electrical Engineering. Faculty of Engineering.
Tokai University.
2-28 Tomigaya, Shibuya-ku, Tokyo Japan.

Abstract— **This paper presents a simple model of three phase switched reluctance machine by using polynomial flux linkage equation. The nonlinear function of flux linkage depends on phase current and the rotor position. The model requires pre-calculated or measured input data therefore Finite Element Analysis (FEA) has been used for data collection. Curve fitting method is implemented to model the magnetizing curves. Since the magnetizing curves are smooth and continuous then flux linkage at particular position and phase current can be estimated by using cubic spline interpolation technique. Moreover, this technique can be used to model torque of the SRM. The comparison between proposed model and FEA reveals the model accuracy.**

***Index Terms*—A magnetic modeling, switched-reluctnace machin, Cubic spline..**

I. INTRODUCTION

The switched reluctance machine (SRM) is considered to be a good competitor to conventional motors because of tits simplicity, brushless, low cost and high efficiency. A great deal of research and development on the SRM has been reported during the last years [1]-[8]. Problems have been found in describing and controlling the motors because of the nonlinear nature of the SRM and high saturation of phase windings during high load. The flux linkage and phase inductance of SRM change with both the rotor position and the phase current. Therefore, the nonlinear model of SRM must be identified as a function of the phase current and rotor position.

From previous work [1], the authors proposed an analysis of SRM based on Fourier Series to find the Fourier coefficients of a_1, a_2 an a_3 which are the coefficients of polynomial function. The calculated results qualitatively agree with the measurement, but it has a problem of evaluating the order of harmonics of Fourier series. This paper presents a new magnetic model of a SRM, which uses cubic spline interpolation technique. By using this technique, we could easily find out the values of flux linkages at any rotor positions. The calculation method will be discussed in section II and III, torque model will be found in section IV, and section V is conclusion.

II. NONLINEAR MODELING

This section presents a nonlinear analysis of the three-phase SRM. Fig. 1 shows the schematic diagram of 12/8 SRM used in this paper. The pole numbers of stator and rotor are 12 and 8 respectively, the core material is soft magnetic M19 silicon steel with a thickness of 24.5 mm. Let the rotor position θ is 0 at aligned position and 22.5 when rotor is unaligned position. The magnetizing curves contain the data of flux linkage versus current at different rotor positions. A polynomial function of flux linkage is used to model these curves.

FEA has been used to obtain the magnetic properties of the candidate machine. The nonlinear model is developed from these data. Fig. 2 shows that the curves are linear for small applied phase current or when the rotor is somewhere around the unaligned position. Then the candidate flux linkage function should contain a linear term. Similarly, the function also has nonlinear terms to represent the saturate effect.

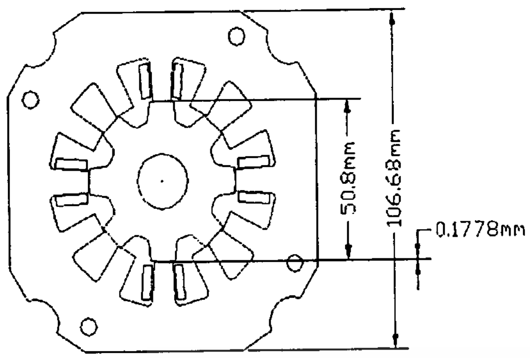
There are a few papers proposed nonlinear flux linkage function [1], [3], [4]. But the simplest form of polynomial function is shown in (1).

$$\lambda(i,\theta) = a_1(\theta)i + a_2(\theta)i^2 + a_3(\theta)i^3 \quad (1)$$

where the coefficients a_1, a_2 and a_3 are functions of rotor positions. The data of Fig. 2 has been fit to (1) using MATLAB. The coefficient a_1 represents the unsaturated inductance of the coil while a_2 and a_3 represent the nonlinearity of the core.

Since a_1, a_2 and a_3 are functions of rotor positions, they can each be represented by a Fourier Series [1], [5], but using this technique we have a problem of evaluating the order of harmonics of Fourier Series. So, this paper proposed a new

0-7803-9296-5/05/$20.00 © 2005 IEEE



Stator pole number	12	poles
Rotor pole number	8	poles
Stator pole arc, β_s	15	deg
Rotor pole arc, β_r	19	deg
Stack length	25.4	mm
Number of windings/pole	24	turns

Figure 1. The structure of a 12/8 SRM and geometry parameters.

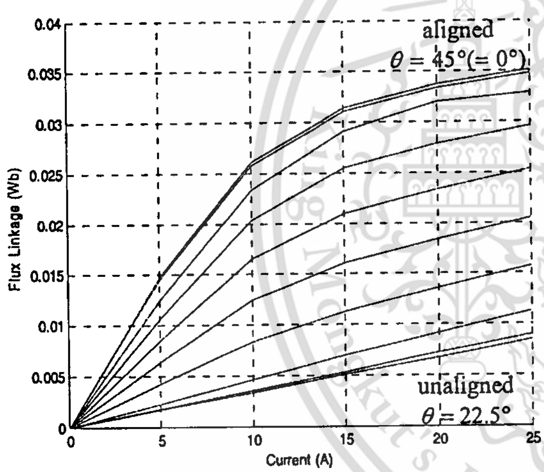


Figure 2. Magnetizing curves of a 12/8 RSM obtained from FEA.

model which uses cubic spline interpolation technique to solve this problem.

III. CUBIC SPLINE

The approach of inductance estimation is introduced in [1]. However it is not good enough to implement finding inductance. So, this paper has introduced a technique of inductance estimation by using cubic spline interpolation technique.

A cubic spline is a spline constructed of piecewise third-order polynomials which pass through a set of m control points [6]. The second derivative of each polynomial is commonly set to zero at the endpoints.

Let the i_{th} piece of the spline be represented by:

$$Y_i(x) = a_i + b_i x + c_i x^2 + d_i x^3 \qquad (2)$$

where a_i, b_i, c_i, d_i are spline coefficients and x is parameter $\in [0,\ 1]$. If the curve has m points, so it would have m-1 pieces. The matrix size of the coefficients a_i, b_i, c_i and d_i is $(m-1)\times 4$. Using this technique applies to our coefficients a_1, a_2 and a_3 curves. These curves are divided into 2.5 interval from aligned position to the next aligned position. There are total 18 pieces and the spline coefficients are (18×4) for each coefficients a_1, a_2 and a_3 as shown in Table 1.

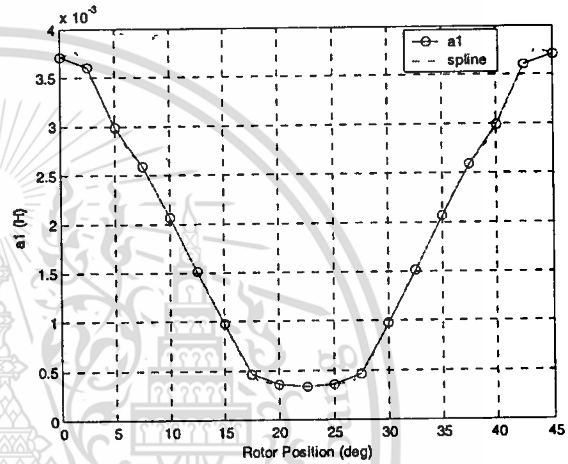


Figure 3. Comparison between cubic spline curve (dashed line) and $a_1(\theta)$ curve (solid line).

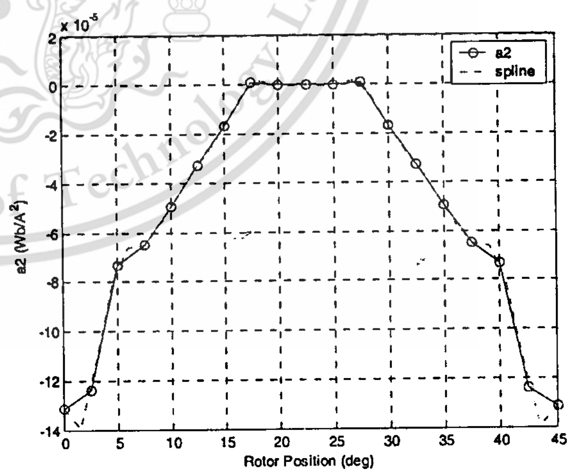


Figure 4. Comparison between cubic spline curve (dashed line) and $a_2(\theta)$ curve (solid line).

TABLE 1: SPLINE COEFFICIENTS OF $a_1(\theta)$, $a_2(\theta)$, $a_3(\theta)$.

$a_1(\theta)$				$a_2(\theta)$				$a_3(\theta)$			
d_i	c_i	b_i	a_i	d_i	c_i	b_i	a_i	d_i	c_i	b_i	a_i
1.15e-05	-0.00013	0.000208	0.003709	-1.35e-06	1.36e-05	-2.28e-05	-0.00013	3.58e-08	-3.50e-07	5.93e-07	1.55e-06
1.15e-05	-4.18e-05	-0.00022	0.003608	-1.35e-06	3.50e-06	2.01e-05	-0.00012	3.58e-08	-8.16e-08	-4.86e-07	1.41e-06
-9.70e-06	4.46e-05	-0.00021	0.002986	1.23e-06	-6.64e-06	1.22e-05	-7.31e-05	-3.35e-08	1.87e-07	-2.22e-07	2.44e-07
4.24e-06	-2.81e-05	-0.00017	0.00259	-4.05e-07	2.62e-06	2.16e-06	-6.48e-05	9.65e-09	-6.43e-08	8.40e-08	3.33e-07
4.73e-08	3.73e-06	-0.00023	0.00206	1.94e-08	-4.22e-07	7.65e-06	-4.94e-05	-3.72e-10	8.06e-09	-5.67e-08	2.92e-07
-2.30e-06	4.09e-06	-0.00021	0.001512	1.74e-07	-2.76e-07	5.90e-06	-3.26e-05	-3.43e-09	5.27e-09	-2.33e-08	1.95e-07
9.14e-6	-1.32e-05	-0.00023	0.000978	-5.06e-07	1.03e-06	7.78e-06	-1.69e-05	7.40e-09	-2.04e-08	-6.12e-08	1.16e-07
-8.23e-06	5.54e-05	-0.00013	0.000458	4.94e-07	-2.77e-06	3.41e-06	1.07e-06	-6.91e-09	3.51e-08	-2.46e-08	-4.93e-08
1.92e-06	-6.36e-06	-4.13e-06	0.000358	-1.87e-07	9.37e-07	-1.17e-06	7.17e-09	3.36e-09	-1.68e-08	2.10e-08	1.82e-10
-1.92e-06	8.01e-06	1.27e-21	0.000338	1.87e-07	-4.68e-07	-9.08e-23	3.72e-09	-3.36e-09	8.39e-09	-7.09e-25	2.11e-10
8.23e-06	-6.36e-06	4.13e-06	0.000358	-4.94e-07	9.37e-07	1.17e-06	7.17e-09	6.91e-09	-1.68e-08	-2.10e-08	1.82e-10
-9.14e-06	5.54e-05	0.000127	0.000458	5.06e-07	-2.77e-06	-3.41e-06	1.07e-06	-7.40e-09	3.51e-08	2.46e-08	-4.93e-08
2.30e-06	-1.32e-05	0.000232	0.000978	-1.74e-07	1.03e-06	-7.78e-06	-1.69e-05	3.43e-09	-2.04e-08	6.12e-08	1.16e-07
-4.73e-08	4.09e-06	0.000209	0.001512	-1.94e-08	-2.76e-07	-5.90e-06	-3.26e-05	3.72e-10	5.27e-09	2.33e-08	1.95e-07
-4.24e-06	3.73e-06	0.000229	0.00206	4.05e-07	-4.22e-07	-7.65e-06	-4.94e-05	-9.65e-09	8.06e-09	5.67e-08	2.92e-07
9.70e-06	-2.81e-05	0.000168	0.00259	-1.23e-06	2.62e-06	-2.16e-06	-6.48e-05	3.35e-08	-6.43e-08	-8.40e-08	3.33e-07
-1.15e-05	4.46e-05	0.000209	0.002986	1.35e-06	-6.64e-06	-1.22e-05	-7.31e-05	-3.58e-08	1.87e-07	2.22e-07	2.44e-07
-1.15e-05	-4.18e-05	0.000217	0.003608	1.35e-06	3.50e-06	-2.01e-05	-0.00012	-3.58e-08	-8.16e-08	4.86e-07	1.41e-06

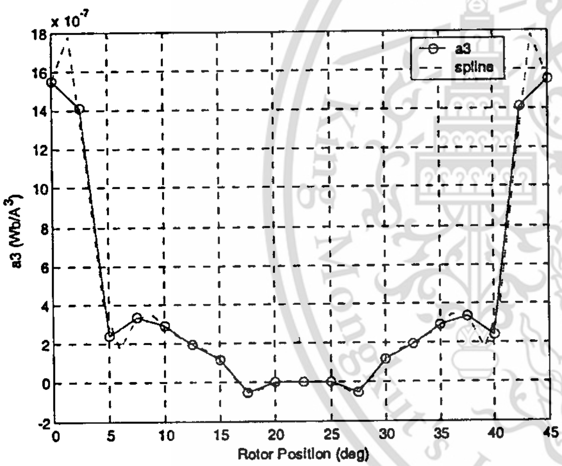


Figure 5. Comparison between cubic spline curve (dashed line) and $a_3(\theta)$ curve (solid line).

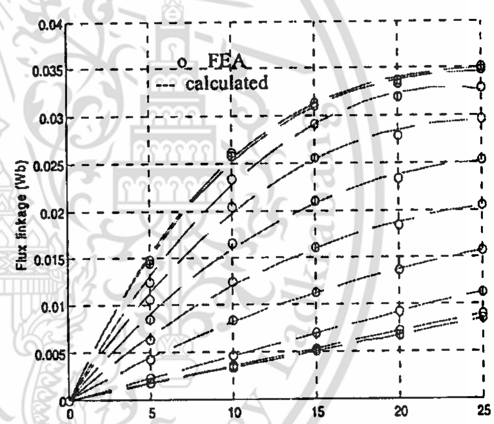


Figure 6. Comparison between results from FEA and calculated magnetizing curve.

Fig. 3, 4 and 5 depict the comparison between the curves of coefficients a_1, a_2 and a_3 from curve fitting model and the reconstruction of those coefficients from cubic spline interpolation technique. The measured magnetizing curve from FEA starting from unaligned to aligned position with step 2.5 in rotor position and the reconstructed curves from cubic spline interpolation are shown in Fig. 6. The results are good agreement. However, there is small noticeable error which is less than 5%.

IV. TORQUE MODELING

Non-linear mapping techniques like lookup tables or artificial neural networks can be used to propose SRM torque in terms of the phase current and rotor position, for estimating SRM online. Lookup table requires large amount of online storage space. On the other hand, artificial neural networks require large amount of online computation. Hence, such methods are not appropriate for real time controller implementation [7]. A compact analysis torque expression based on physical principles of SRM operation would be the most efficient for real time implementation. However, looking at the measured torque data at different phase currents and rotor position, no intuitive model can be thought of for representing the torque as an analysis function [8].

Torque production of a SRM is represented as the position rate of change of co-energy at given current and rotor position. For the nonlinear SRM, torque production for each phase is given by:

$$T = \frac{\partial}{\partial \theta} \int_0^i \lambda(i,\theta) di \qquad (3)$$

Substitution of (1) to (3) gives:

$$T = \frac{1}{2} i^2 \frac{da_1(\theta)}{d\theta} + \frac{1}{3} i^3 \frac{da_2(\theta)}{d\theta} + \frac{1}{4} i^4 \frac{da_3(\theta)}{d\theta} \qquad (4)$$

This torque equation represents only one phase. For the total SRM torque, it is simply summing up individual phase torque with appropriate adjustment for spatial displacement. From (4), the torque equation comprises of the derivative of the coefficients a_1, a_2 and a_3. The cubic spline interpolation technique is implemented by taking derivative of (2). The derivative of each piece of spline becomes (5).

$$Y_i(x) = b_i + 2c_i x + 3d_i x^2 \qquad (5)$$

Using (5) and the coefficients in Table 1 with (4) yield the torque value at particular position. The result of calculating of static torque is shown in Fig. 7. The static torque was computed using (4) for the rotor position starting from aligned position to the next aligned position. This torque is well suited for both position (motoring mode) and negative (generating mode). The peak of negative torque and positive torque are well balancing and derisible for engineering proposed. Fig. 8 shows the static torque when using the torque equation from (4) and FEA. The results show that there are very good fit.

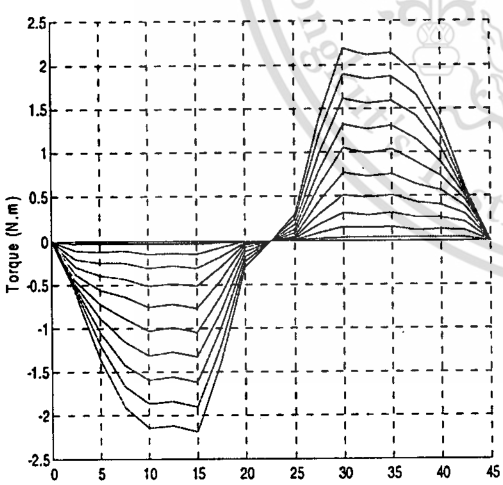


Figure 7. Static torque for different current levels showing both positive and negative torque.

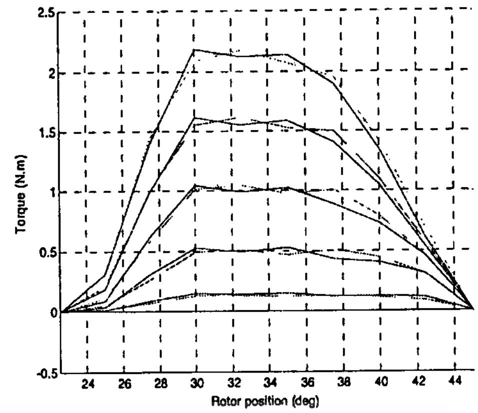


Figure 8. Static torque obtained from FEA (dashed line) and from torque equation (solid line) at different current levels.

V. CONCLUSION

In this paper, the simplified non-linear magnetic analysis for evaluating the magnetizing curves of SRM has been proposed. The validity of the presented analysis has been verified by the comparison with finite element analysis. The polynomial function of cubic spline has been implemented to model the magnetic property of the machine. The polynomial flux linkage and torque functions are very simple for further analysis and implementation.

ACKNOWLEDGEMENT

The Authors would like to thank you to the AUN/SEED-Net, JICA for the financial support on this project.

REFERENCES

[1] S. Kittiratsatcha, D. A. Torrey, "A Design Method of a Two-phase Switched-Reluctance Machine," Power Electronics and Drive Systems, 2003. PEDS 2003. The Fifth International Conference onVolume 2, 17-20 Nov. 2003 Page(s):1599 – 1604.

[2] Wenzhe Lu, Keyhani. A, Fardoun. A, "Neural Network-Based Modeling and Parameter Identification of Switched Reluctance Motors," Energy Conversion, IEEE Transactions on Volume 18, Issue 2, June 2003 Page(s):284 – 290.

[3] Liu Shanshan, Zhao Zhengming, Meng Shuo, Chai Jianyun, "A Non-Linear "Analysis Model For Swithched Reluctance Motor," TENCON '02. Proceedings. 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering Volume 3, 28-31 Oct. 2002 Page(s):2034 - 2037 vol.3.

[4] D. A. Torrey, "Switched Reluctance Generators and Their Control, " Industrial Electronics, IEEE Transactions on Volume 49, Issue 1, Feb. 2002 Page(s):3 – 14.

[5] Kano. Y, Kosaka. T, Matsui. N, "Magnetization characteristics analysis of SRM by simplified non-linear magnetic analysis," Power Conversion Conference, 2002. PCC Osaka 2002. Proceedings of the Volume 2, 2-5 April 2002 Page(s):689 - 694 vol.2.

[6] Eri W. Weisstein, "Cubic Spline," >*From Math World.* Available: http://mathworld.wolfram.com/CubicSpline.html

[7] S. K Sahoo, S. K. Panda, and J. X. Xu "Model-base Torque Estimator for Switched Reluctance Motors," Power Electronics and Drive Systems, 2003. PEDS 2003. The Fifth International Conference on Volume 2, 17-20 Nov. 2003 Page(s):959 - 963 Vol.2.

[8] Stiebler. M, Ke Liu, "An analytical model of switched reluctance machines," Energy Conversion, IEEE Transactions on Volume 14, Issue 4, Dec. 1999 Page(s):1100 – 1107.

Appendix B

Application Software: MATLAB 5.6.0 18913a Release 13

```
% Sisavath Khotpanya 25/8/2004
% File Megnetic_Model.m
% This m-file has been written to simulate the magnetization curve of SRM. This program is
% the main program that run the model of SRM. It consists of flux linkage model, torque
% production and inductance profile.

clc;            % clear window screen
clear all;      % clear all parameters
close all;      % close all figures
t=[zeros(1,10)
-4.18E-05     0.0126487  0.0654355  0.123648  0.12774  0.115666  0.129189  0.11268   0.0846235  -0.00122949
-0.000167046  0.0506078  0.261929   0.490095  0.506913  0.461235  0.506571  0.438983  0.306615  -0.00325852
-0.000376015  0.113915   0.587555   1.0101    1.03874   0.967668  1.00001   0.786749  0.451829  -0.0022125
-0.000668791  0.202608   0.996515   1.54827   1.60753   1.51872   1.4979    1.09823   0.570044  -0.0014149
-0.00104537   0.316689   1.43834    2.08493   2.17393   2.06315   1.94859   1.4184    0.678006  -0.00110299];
t=[ zeros(6,9) ,t] ;  %Matrix t is the static torque that measured form FEA, used to compare with
% proposed torque model.
magdata=[   zeros(1,20)
0       0         0         0         0         0         0         0         0         0         0         0 ↵
        0         0         0         0         0         0         0         0
5       0.014755  0.014387  0.012512  0.010545  0.0084639 0.0063402 0.0042735 0.0023138 0.0017926 0.0016919 ↵
        0.0017926 0.0023138 0.0042735 0.0063402 0.0084639 0.010545  0.012512  0.014387  0.014755
10      0.026181  0.025754  0.023367  0.020461  0.016507  0.012425  0.0084001 0.0046288 0.0035856 0.0033841 ↵
        0.0035856 0.0046288 0.0084001 0.012425  0.016507  0.020461  0.023367  0.025754  0.026181
15      0.0314    0.03108   0.029233  0.025606  0.020942  0.016102  0.011285  0.0069414 0.0053792 0.0050767 ↵
        0.0053792 0.0069414 0.011285  0.016102  0.020942  0.025606  0.029233  0.03108   0.0314
20      0.033689  0.033353  0.032043  0.02795   0.023331  0.018452  0.013607  0.0091879 0.0071738 0.0067702 ↵
        0.0071738 0.0091879 0.013607  0.018452  0.023331  0.02795   0.032043  0.033353  0.033689
25      0.035092  0.034841  0.032886  0.029641  0.02535   0.020558  0.015752  0.01134   0.0089691 0.0084643 ↵
        0.0089691 0.01134   0.015752  0.020558  0.02535   0.029641  0.032886  0.034841  0.035092];
```

```
%%------ Input magnitizing data
magdata;
magdata1=magdata;
%%------ Enter number of position from present aligned position to next align position.
%%------ This general data position can be used with 3 phase machine data.
pos=19;
%%------ Initiate matrix
A=zeros(pos,3);        %% contain fitting coeff.                    %%***
xcol=1;                     %% define column#1 as angle
lambdai=magdata1 ;    %% interchange name
%%------ initial number for loops
j=1;
jj=1;
r=0:2.5:22.5;
r=r';
%%------ start loop#1 to calculate fitting number
for ycol=2:pos+1,       %% data starts from col.#2

Data = [lambdai(:,xcol),lambdai(:,ycol)]    %% dummy matrix using for calculate fitting coeff in
% each position.

   x = Data(:,1);              %% set variable col.1 of 'Data' to use in function 'expstp'
   y = Data(:,2)        %% set variable col.2 of 'Data' to use in function 'expstp'

%%------ initial condition of each fitting number
   a(1) = 0.0005;
   a(2) = -0.001 ;
   a(3) = 0.001;
%%------ call function'expstp'
   a = leastsq('expstp',a,[],[],Data)
   a;
```

```
   %%------ put the coeff. back to fit curve
   yfit = a(1)*x + a(2)*x.^2 + a(3)*x.^3 ;
   fit_data(:,ycol-1) = yfit;
 %%------ store data in matrix 'A'
   A(j,1)=a(1)
   A(j,2)=a(2)
   A(j,3)=a(3)

%%------ Statements to plot progress of fitting:
ro=0:1:5;
   hold on;
   if j<= 9
      %plot(x,yfit,'k',x,y,'ko');    %% black is negative slope
   else
      plot(x,yfit,'r',x,y,'ro');    %% red is positive slope
   end
   j=j+1;                      %% incerase j
end                            %% end loop
grid;
xlabel('Current (A)');
ylabel('Flux Linkage (Wb)');
%%------ Generate curve a1, a2 and a3
Avstheta=[theta' A]
figure;plot(Avstheta(:,1),Avstheta(:,2)); % Show a1 curve from curve fitting.
grid;title('a1');
xlabel('Rotor position (deg)');
ylabel('Inductance (H)');
figure;plot(Avstheta(:,1),Avstheta(:,3)); % Show a2 curve from curve fitting.
grid;title('a2');
xlabel('Rotor position (deg)');
ylabel('a2 (Wb/A^2)');
```

```
figure;plot(Avstheta(:,1),Avstheta(:,4)); % Show a3 curve from curve fitting.
grid;title('a3');
xlabel('Rotor position (deg)');
ylabel('a3 (Wb/A^3)');
graph=Avstheta;
%....................Find the spline coefficients of a1, a2 & a3.........................
% Using Natura spline method
% AM=b
A=[     4 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
        1 4 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
        0 1 4 1 0 0 0 0 0 0 0 0 0 0 0 0 0
        0 0 1 4 1 0 0 0 0 0 0 0 0 0 0 0 0
        0 0 0 1 4 1 0 0 0 0 0 0 0 0 0 0 0
        0 0 0 0 1 4 1 0 0 0 0 0 0 0 0 0 0
        0 0 0 0 0 1 4 1 0 0 0 0 0 0 0 0 0
        0 0 0 0 0 0 1 4 1 0 0 0 0 0 0 0 0
        0 0 0 0 0 0 0 1 4 1 0 0 0 0 0 0 0
        0 0 0 0 0 0 0 0 1 4 1 0 0 0 0 0 0
        0 0 0 0 0 0 0 0 0 1 4 1 0 0 0 0 0
        0 0 0 0 0 0 0 0 0 0 1 4 1 0 0 0 0
        0 0 0 0 0 0 0 0 0 0 0 1 4 1 0 0 0
        0 0 0 0 0 0 0 0 0 0 0 0 1 4 1 0 0
        0 0 0 0 0 0 0 0 0 0 0 0 0 1 4 1 0
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 4 1
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 4];
h=2.5;
n=19;
for r=2:4,
    x=graph(:,1);
    y=graph(:,r);
```

```
    for i=1:n-2,
      b(i)=y(i)-2*y(i+1)+y(i+2);
      B(i,:)=b(i);  %use to store b value.
    end
%Fine value of Matrix M
    M=A\(B.*(6/h^2));
% According to Natural spline M0 =Mn =0;
    M=[0;M;0];
% Find Natural spline coefficients for a1, a2 & a3
    if r = =2  % Find spline coefficients of a1
      for i=1:n-1
        a(i)=(M(i+1)-M(i))/(6*h);
        c1(i,1)=a(i);

        b(i)=M(i)/2;
        c1(i,2)=b(i);

        c_1(i)=((y(i+1)-y(i))/h)-((M(i+1)+2*M(i))*h)/6;
        c1(i,3)=c_1(i);

        d(i) = y(i);
        c1(i,4)=d(i);
      end
    end

    if r = =3  % Find spline coefficients of a2
      for i =1:n-1
        a(i) = (M(i+1)-M(i))/(6*h);
        c2(i,1) = a(i);

        b(i) =M(i)/2;
```

```
        c2(i,2) = b(i);

        c_1(i) = ((y(i+1)-y(i))/h)-((M(i+1)+2*M(i))*h)/6;
        c2(i,3) = c_1(i);

        d(i) = y(i);
        c2(i,4) = d(i);
      end
    end

    if r = =4   % Find spline coefficients of a3
       for i =1:n-1
        a(i) = (M(i+1)-M(i))/(6*h);
        c(i,1) = a(i);

        b(i) = M(i)/2;
        c(i,2) = b(i);

        c_1(i) = ((y(i+1)-y(i))/h)-((M(i+1)+2*M(i))*h)/6;
        c(i,3) = c_1(i);

        d(i) = y(i);
        c(i,4) = d(i);
      end
    end
 u = 0:1:45;
theta = [0:1:45];

if r = = 2 % for a1
  for k=1:length(u)
```

```
 if u(k) >= 0 & u(k) <= 2.5
  y_fit2(k) = (u(k))^3*c1(1,1) + (u(k))^2*c1(1,2) + (u(k))*c1( 1,3)+ c1(1,4)
  d_fit2(k) = 3*(u(k))^2*c1(1,1) + 2*(u(k))*c1(1,2) + c1(1,3);
  fit(r,k) = y_fit2(k);
  d_yfit2(r,k) = d_fit2(k);
elseif  u(k) > 2.5 & u(k) <= 5
  u(k) = u(k)- 2.5;
  y_fit2(k) = (u(k))^3*c1(2,1) + (u(k))^2*c1(2,2) + (u(k))*c1(2,3) + c1(2,4)
  d_fit2(k) = 3*(u(k))^2*c1(2,1) + 2*(u(k))*c1(2,2) + c1(2,3);
  fit(r,k) = y_fit2(k);
  d_yfit2(r,k) = d_fit2(k);

elseif u(k) > 5 & u(k) <= 7.5
  u(k) = u(k)-5;
  y_fit2(k) = (u(k))^3*c1(3,1 ) + (u(k))^2*c1(3,2) + (u(k))*c1(3,3) + c1(3,4)
  d_fit2(k) = 3*(u(k))^2*c1(3,1) + 2*(u(k))*c1(3,2) + c1(3,3);
  fit(r,k) = y_fit2(k);
  d_yfit2(r,k) = d_fit2(k);
 elseif u(k)> 7.5 & u(k) <=10
   u(k) = u(k)-7.5;
  y_fit2(k) = (u(k))^3*c1(4,1) + (u(k))^2*c1(4,2 ) + (u(k))*c1(4,3 ) + c1( 4,4)
  d_fit2(k) = 3*(u(k))^2*c1(4,1) + 2*(u(k))*c1(4,2) + c1(4,3);
  fit(r,k) = y_fit2(k);
  d_yfit2(r,k)= d_fit2(k);
elseif u(k)> 10 & u(k) <=12.5
  u(k) = u(k)-10;
  y_fit2(k) = (u(k))^3*c1(5,1) + (u(k))^2*c1( 5,2 ) + (u(k))*c1(5,3) + c1(5,4)
  d_fit2(k) = 3*(u(k))^2*c1(5,1) + 2*(u(k))*c1(5,2) + c1(5,3);
  fit(r,k) = y_fit2(k);
  d_yfit2(r,k) = d_fit2(k);
```

```
elseif u(k) > 12.5 & u(k) <=15
  u(k) = u(k)-12.5;
  y_fit2(k) = (u(k))^3*c1( 6,1) + (u(k))^2*c1( 6,2 ) + (u(k))*c1(6,3 ) + c1(6,4 )
  d_fit2(k) = 3*(u(k))^2*c1(6,1) + 2*(u(k))*c1(6,2) + c1(6,3);
  fit(r,k) = y_fit2(k);
  d_yfit2(r,k) = d_fit2(k);
elseif u(k) > 15 & u(k) <=17.5
  u(k) = u(k)-15;
  y_fit2(k) = (u(k))^3*c1( 7,1 ) + (u(k))^2*c1( 7,2  ) + (u(k))*c1(7,3 ) + c1(7, 4)
  d_fit2(k) = 3*(u(k))^2*c1(7,1) + 2*(u(k))*c1(7,2) + c1(7,3);
  fit(r,k) = y_fit2(k);
  d_yfit2(r,k) = d_fit2(k);
elseif u(k) > 17.5 & u(k) <=20
  u(k) = u(k)-17.5;
  y_fit2(k) = (u(k))^3*c1(8,1 ) + (u(k))^2*c1( 8,2 ) + (u(k))*c1( 8,3) + c1(8,4 )
  d_fit2(k) = 3*(u(k))^2*c1(8,1) + 2*(u(k))*c1(8,2) + c1(8,3);
  fit(r,k) = y_fit2(k);
  d_yfit2(r,k) = d_fit2(k);
elseif u(k) > 20 & u(k) <=22.5
  u(k) = u(k)-20;
  y_fit2(k) = (u(k))^3*c1(9,1 ) + (u(k))^2*c1( 9,2 ) + (u(k))*c1( 9,3) + c1(9,4 )
  d_fit2(k) = 3*(u(k))^2*c1(9,1) + 2*(u(k))*c1(9,2) + c1(9,3);
  fit(r,k) = y_fit2(k);
  d_yfit2(r,k) = d_fit2(k);
elseif u(k)> 22.5 & u(k) <=25
  u(k) = u(k)-22.5;
  y_fit2(k) = (u(k))^3*c1(10,1 ) + (u(k))^2*c1( 10,2 ) + (u(k))*c1( 10,3) + c1(10,4 )
  d_fit2(k) = 3*(u(k))^2*c1(10,1) + 2*(u(k))*c1(10,2) + c1(10,3);
  fit(r,k) = y_fit2(k);
  d_yfit2(r,k) = d_fit2(k);
```

```
elseif u(k) > 25 & u(k) <=27.5
  u(k) = u(k)-25;
  y_fit2(k) = (u(k))^3*c1(11,1 ) + (u(k))^2*c1( 11,2 ) + (u(k))*c1( 11,3) + c1(11,4 )
  d_fit2(k) = 3*(u(k))^2*c1(11,1) + 2*(u(k))*c1(11,2) + c1(11,3);
  fit(r,k) = y_fit2(k);
  d_yfit2(r,k) = d_fit2(k);
elseif u(k)> 27.5 & u(k) <=30
  u(k) = u(k)-27.5;
  y_fit2(k) = (u(k))^3*c1(12,1 ) + (u(k))^2*c1(12,2 ) + (u(k))*c1( 12,3) + c1(12,4 )
  d_fit2(k) = 3*(u(k))^2*c1(12,1) + 2*(u(k))*c1(12,2) + c1(12,3);
  fit(r,k) = y_fit2(k);
  d_yfit2(r,k) = d_fit2(k);
elseif u(k)> 30 & u(k) <=32.5
  u(k) = u(k)-30;
  y_fit2(k) = (u(k))^3*c1(13,1 ) + (u(k))^2*c1( 13,2 ) + (u(k))*c1( 13,3) + c1(13,4 )
  d_fit2(k) = 3*(u(k))^2*c1(13,1) + 2*(u(k))*c1(13,2) + c1(13,3);
  fit(r,k) = y_fit2(k);
  d_yfit2(r,k) = d_fit2(k);
elseif u(k)> 32.5 & u(k) <=35
  u(k) = u(k)-32.5;
  y_fit2(k) = (u(k))^3*c1(14,1 ) + (u(k))^2*c1( 14,2 ) + (u(k))*c1( 14,3) + c1(14,4 )
  d_fit2(k) = 3*(u(k))^2*c1(14,1) + 2*(u(k))*c1(14,2) + c1(14,3);
  fit(r,k) = y_fit2(k);
  d_yfit2(r,k) = d_fit2(k);
elseif u(k)> 35 & u(k) <=37.5
  u(k)  = u(k)-35;
  y_fit2(k) = (u(k))^3*c1(15,1 ) + (u(k))^2*c1( 15,2 ) + (u(k))*c1( 15,3) + c1(15,4 )
  d_fit2(k) = 3*(u(k))^2*c1(15,1) + 2*(u(k))*c1(15,2) + c1(15,3);
  fit(r,k) = y_fit2(k);
  d_yfit2(r,k) = d_fit2(k);
```

```
 elseif u(k)> 37.5 & u(k) <=40
   u(k) = u(k)-37.5;
   y_fit2(k) = (u(k))^3*c1(16,1 ) + (u(k))^2*c1( 16,2 ) + (u(k))*c1( 16,3) + c1(16,4 )
   d_fit2(k) = 3*(u(k))^2*c1(16,1) + 2*(u(k))*c1(16,2) + c1(16,3);
   fit(r,k) = y_fit2(k);
   d_yfit2(r,k) = d_fit2(k);
 elseif u(k)> 40 & u(k) <=42.5
   u(k) = u(k)-40;
   y_fit2(k) = (u(k))^3*c1(17,1 ) + (u(k))^2*c1( 17,2 ) + (u(k))*c1( 17,3) + c1(17,4 )
   d_fit2(k) = 3*(u(k))^2*c1(17,1) + 2*(u(k))*c1(17,2) + c1(17,3);
   fit(r,k) = y_fit2(k);
   d_yfit2(r,k) = d_fit2(k);
 elseif u(k)> 42.5 & u(k) <=45
   u(k) = u(k)-42.5;
   y_fit2(k) = (u(k))^3*c1(18,1 ) + (u(k))^2*c1( 18,2 ) + (u(k))*c1( 18,3) + c1(18,4 )
   d_fit2(k) = 3*(u(k))^2*c1(18,1) + 2*(u(k))*c1(18,2) + c1(18,3);
   fit(r,k) = y_fit2(k);
   d_yfit2(r,k) = d_fit2(k);
 else
   fprintf('wrong number');
 end
end

d_yfit2(r,1)=0;
d_yfit2(r,length(u))=0
figure;
plot(x,y,'-o',theta,fit(r,:),':r');  % comparison between a1 and spline
legend('a1','spline');
xlabel('Rotor position (deg)');
ylabel('a1 (H)');
```

```
elseif  r = =3      % for a2
   for k = 1:length(u)
      if u(k) >=0 & u(k) <= 2.5
         y_fit2(k) = (u(k))^3*c2(1,1) + (u(k))^2*c2(1,2) + (u(k))*c2( 1,3)+ c2(1,4)
         d_fit2(k) = 3*(u(k))^2*c2(1,1) + 2*(u(k))*c2(1,2) + c2(1,3);
         fit(r,k) = y_fit2(k);
         d_yfit2(r,k) = d_fit2(k);
      elseif  u(k)> 2.5 & u(k) <= 5
         u(k) = u(k)- 2.5;
         y_fit2(k) = (u(k))^3*c2(2,1) + (u(k))^2*c2(2,2) + (u(k))*c2(2,3) + c2(2,4)
         d_fit2(k) = 3*(u(k))^2*c2(2,1) + 2*(u(k))*c2(2,2) + c2(2,3);
         fit(r,k) = y_fit2(k);
         d_yfit2(r,k) = d_fit2(k);
      elseif u(k)> 5 & u(k) <= 7.5
         u(k) = u(k)-5;
         y_fit2(k) = (u(k))^3*c2(3,1 ) + (u(k))^2*c2(3,2) + (u(k))*c2(3,3) + c2(3,4)
         d_fit2(k) = 3*(u(k))^2*c2(3,1) + 2*(u(k))*c2(3,2) + c2(3,3);
         fit(r,k) = y_fit2(k);
         d_yfit2(r,k) = d_fit2(k);
       elseif u(k)> 7.5 & u(k) <=10
          u(k) = u(k)-7.5;
         y_fit2(k) = (u(k))^3*c2(4,1) + (u(k))^2*c2(4,2 ) + (u(k))*c2(4,3 ) + c2( 4,4)
         d_fit2(k) = 3*(u(k))^2*c2(4,1) + 2*(u(k))*c2(4,2) + c2(4,3);
         fit(r,k) = y_fit2(k);
         d_yfit2(r,k) = d_fit2(k);
      elseif u(k)> 10 & u(k) <=12.5
         u(k) = u(k)-10;
         y_fit2(k) = (u(k))^3*c2(5,1) + (u(k))^2*c2( 5,2 ) + (u(k))*c2(5,3) + c2(5,4)
         d_fit2(k)= 3*(u(k))^2*c2(5,1) + 2*(u(k))*c2(5,2) + c2(5,3);
         fit(r,k) = y_fit2(k);
         d_yfit2(r,k) = d_fit2(k);
```

```
  elseif u(k)> 12.5 & u(k) <=15
    u(k) = u(k)-12.5;
    y_fit2(k) = (u(k))^3*c2( 6,1) + (u(k))^2*c2( 6,2 ) + (u(k))*c2(6,3 ) + c2(6,4 )
    d_fit2(k) = 3*(u(k))^2*c2(6,1) + 2*(u(k))*c2(6,2) + c2(6,3);
    fit(r,k) = y_fit2(k);
    d_yfit2(r,k) = d_fit2(k);
 elseif u(k)> 15 & u(k) <=17.5
    u(k) = u(k)-15;
    y_fit2(k) = (u(k))^3*c2( 7,1 ) + (u(k))^2*c2( 7,2  ) + (u(k))*c2(7,3 ) + c2(7, 4)
    d_fit2(k) = 3*(u(k))^2*c2(7,1) + 2*(u(k))*c2(7,2) + c2(7,3);
    fit(r,k) = y_fit2(k);
    d_yfit2(r,k) = d_fit2(k);
 elseif u(k)> 17.5 & u(k) <=20
    u(k) = u(k)-17.5;
    y_fit2(k) = (u(k))^3*c2(8,1 ) + (u(k))^2*c2( 8,2 ) + (u(k))*c2( 8,3) + c2(8,4 )
    d_fit2(k) = 3*(u(k))^2*c2(8,1) + 2*(u(k))*c2(8,2) + c2(8,3);
    fit(r,k) = y_fit2(k);
    d_yfit2(r,k) = d_fit2(k);
 elseif u(k)> 20 & u(k) <=22.5
    u(k) = u(k)-20;
    y_fit2(k) = (u(k))^3*c2(9,1 ) + (u(k))^2*c2( 9,2 ) + (u(k))*c2( 9,3) + c2(9,4 )
    d_fit2(k) = 3*(u(k))^2*c2(9,1) + 2*(u(k))*c2(9,2) + c2(9,3);
    fit(r,k) = y_fit2(k);
    d_yfit2(r,k) = d_fit2(k);
 elseif u(k)> 22.5 & u(k) <=25
    u(k) = u(k)-22.5;
    y_fit2(k) = (u(k))^3*c2(10,1 ) + (u(k))^2*c2( 10,2 ) + (u(k))*c2( 10,3) + c2(10,4 )
    d_fit2(k) = 3*(u(k))^2*c2(10,1) + 2*(u(k))*c2(10,2) + c2(10,3);
    fit(r,k) = y_fit2(k);
    d_yfit2(r,k) = d_fit2(k);
```

```
elseif u(k)> 25 & u(k) <=27.5
   u(k) = u(k)-25;
   y_fit2(k) = (u(k))^3*c2(11,1 ) + (u(k))^2*c2( 11,2 ) + (u(k))*c2( 11,3) + c2(11,4 )
   d_fit2(k) = 3*(u(k))^2*c2(11,1) + 2*(u(k))*c2(11,2) + c2(11,3);
   fit(r,k) = y_fit2(k);
   d_yfit2(r,k) = d_fit2(k);
elseif u(k)> 27.5 & u(k) <=30
   u(k) = u(k)-27.5;
   y_fit2(k) = (u(k))^3*c2(12,1 ) + (u(k))^2*c2( 12,2 ) + (u(k))*c2( 12,3) + c2(12,4 )
   d_fit2(k ) = 3*(u(k))^2*c2(12,1) + 2*(u(k))*c2(12,2) + c2(12,3);
   fit(r,k) = y_fit2(k);
   d_yfit2(r,k) = d_fit2(k);
elseif u(k)> 30 & u(k) <=32.5
   u(k) = u(k)-30;
   y_fit2(k) = (u(k))^3*c2(13,1 ) + (u(k))^2*c2( 13,2 ) + (u(k))*c2( 13,3) + c2(13,4 )
   d_fit2(k) = 3*(u(k))^2*c2(13,1) + 2*(u(k))*c2(13,2) + c2(13,3);
   fit(r,k) = y_fit2(k);
   d_yfit2(r,k) = d_fit2(k);
elseif u(k)> 32.5 & u(k) <=35
   u(k) = u(k)-32.5;
   y_fit2(k) = (u(k))^3*c2(14,1 ) + (u(k))^2*c2( 14,2 ) + (u(k))*c2( 14,3) + c2(14,4 )
   d_fit2(k) = 3*(u(k))^2*c2(14,1) + 2*(u(k))*c2(14,2) + c2(14,3);
   fit(r,k) = y_fit2(k);
   d_yfit2(r,k) = d_fit2(k);
elseif u(k)> 35 & u(k) <=37.5
   u(k) = u(k)-35;
   y_fit2(k) = (u(k))^3*c2(15,1 ) + (u(k))^2*c2( 15,2 ) + (u(k))*c2( 15,3) + c2(15,4 )
   d_fit2(k) = 3*(u(k))^2*c2(15,1) + 2*(u(k))*c2(15,2) + c2(15,3);
   fit(r,k) = y_fit2(k);
   d_yfit2(r,k) = d_fit2(k);
```

```
.

 elseif u(k)> 37.5 & u(k) <=40
   u(k) = u(k)-37.5;
   y_fit2(k) = (u(k))^3*c2(16,1 ) + (u(k))^2*c2( 16,2 ) + (u(k))*c2( 16,3) + c2(16,4 )
   d_fit2(k) = 3*(u(k))^2*c2(16,1) + 2*(u(k))*c2(16,2) + c2(16,3);
   fit(r,k) = y_fit2(k);
   d_yfit2(r,k) = d_fit2(k);
 elseif u(k)> 40 & u(k) <=42.5
   u(k) = u(k)-40;
   y_fit2(k) = (u(k))^3*c2(17,1 ) + (u(k))^2*c2( 17,2 ) + (u(k))*c2( 17,3) + c2(17,4 )
   d_fit2(k) = 3*(u(k))^2*c2(17,1) + 2*(u(k))*c2(17,2) + c2(17,3);
   fit(r,k) = y_fit2(k);
   d_yfit2(r,k) = d_fit2(k);
 elseif u(k)> 42.5 & u(k) <=45
   u(k) = u(k)-42.5;
   y_fit2(k) = (u(k))^3*c2(18,1 ) + (u(k))^2*c2( 18,2 ) + (u(k))*c2( 18,3) + c2(18,4 )
   d_fit2(k) = 3*(u(k))^2*c2(18,1) + 2*(u(k))*c2(18,2) + c2(18,3);
   fit(r,k) = y_fit2(k);
   d_yfit2(r,k) = d_fit2(k);
 else
   fprintf('wrong number');
 end
end
d_yfit2(r,1) = 0;
d_yfit2(r,length(u)) = 0
figure;
%hold on;
plot(x,y,'-o',theta,fit(r,:),':r');          % comparison between a2 and spline
legend('a2','spline');
xlabel('Rotor position (deg)');
ylabel('a2 (Wb/A^2)');
```

```
else    % for a3
    for k=1:length(u)
        if u(k)>=0 & u(k) <= 2.5
            y_fit2(k) = (u(k))^3*c(1,1) + (u(k))^2*c(1,2) + (u(k))*c( 1,3)+ c(1,4)
            d_fit2(k) = 3*(u(k))^2*c(1,1) + 2*(u(k))*c(1,2) + c(1,3);
            fit(r,k) = y_fit2(k);
            d_yfit2(r,k) = d_fit2(k);
        elseif  u(k)> 2.5 & u(k) <= 5
            u(k) = u(k)- 2.5;
            y_fit2(k) = (u(k))^3*c(2,1) + (u(k))^2*c(2,2) + (u(k))*c(2,3) + c(2,4)
            d_fit2(k) = 3*(u(k))^2*c(2,1) + 2*(u(k))*c(2,2) + c(2,3);
            fit(r,k) = y_fit2(k);
            d_yfit2(r,k) = d_fit2(k);
        elseif u(k)> 5 & u(k) <= 7.5
            u(k) = u(k)-5;
            y_fit2(k) = (u(k))^3*c(3,1 ) + (u(k))^2*c(3,2) + (u(k))*c(3,3) + c(3,4)
            d_fit2(k) = 3*(u(k))^2*c(3,1) + 2*(u(k))*c(3,2) + c(3,3);
            fit(r,k) = y_fit2(k);
            d_yfit2(r,k) = d_fit2(k);
         elseif u(k)> 7.5 & u(k) <=10
             u(k) = u(k)-7.5;
            y_fit2(k) = (u(k))^3*c(4,1) + (u(k))^2*c(4,2 ) + (u(k))*c(4,3 ) + c( 4,4)
            d_fit2(k) = 3*(u(k))^2*c(4,1) + 2*(u(k))*c(4,2) + c(4,3);
            fit(r,k) = y_fit2(k);
            d_yfit2(r,k) = d_fit2(k);
        elseif u(k)> 10 & u(k) <=12.5
            u(k) = u(k)-10;
            y_fit2(k) = (u(k))^3*c(5,1) + (u(k))^2*c( 5,2 ) + (u(k))*c(5,3) + c(5,4)
            d_fit2(k) = 3*(u(k))^2*c(5,1) + 2*(u(k))*c(5,2) + c(5,3);
            fit(r,k) = y_fit2(k);
            d_yfit2(r,k) = d_fit2(k);
```

```
elseif u(k)> 12.5 & u(k) <=15
  u(k) = u(k)-12.5;
  y_fit2(k) = (u(k))^3*c( 6,1) + (u(k))^2*c( 6,2 ) + (u(k))*c(6,3 ) + c(6,4 )
  d_fit2(k) = 3*(u(k))^2*c(6,1) + 2*(u(k))*c(6,2) + c(6,3);
  fit(r,k) = y_fit2(k);
  d_yfit2(r,k) = d_fit2(k);
elseif u(k)> 15 & u(k) <=17.5
  u(k) = u(k)-15;
  y_fit2(k) = (u(k))^3*c( 7,1 ) + (u(k))^2*c( 7,2  ) + (u(k))*c(7,3 ) + c(7, 4)
  d_fit2(k) = 3*(u(k))^2*c(7,1) + 2*(u(k))*c(7,2) + c(7,3);
  fit(r,k) = y_fit2(k);
  d_yfit2(r,k) = d_fit2(k);
elseif u(k)> 17.5 & u(k) <=20
  u(k) = u(k)-17.5;
  y_fit2(k) = (u(k))^3*c(8,1 ) + (u(k))^2*c( 8,2 ) + (u(k))*c( 8,3) + c(8,4 )
  d_fit2(k) = 3*(u(k))^2*c(8,1) + 2*(u(k))*c(8,2) + c(8,3);
  fit(r,k) = y_fit2(k);
  d_yfit2(r,k) = d_fit2(k);
elseif u(k)> 20 & u(k) <=22.5
  u(k) = u(k)-20;
  y_fit2(k) = (u(k))^3*c(9,1 ) + (u(k))^2*c( 9,2 ) + (u(k))*c( 9,3) + c(9,4 )
  d_fit2(k) = 3*(u(k))^2*c(9,1) + 2*(u(k))*c(9,2) + c(9,3);
  fit(r,k) = y_fit2(k);
  d_yfit2(r,k) = d_fit2(k);
elseif u(k)> 22.5 & u(k) <=25
  u(k) = u(k)-22.5;
  y_fit2(k) = (u(k))^3*c(10,1 ) + (u(k))^2*c( 10,2 ) + (u(k))*c( 10,3) + c(10,4 )
  d_fit2(k) = 3*(u(k))^2*c(10,1) + 2*(u(k))*c(10,2) + c(10,3);
  fit(r,k) = y_fit2(k);
  d_yfit2(r,k) = d_fit2(k);
```

```
elseif u(k)> 25 & u(k) <=27.5
  u(k) = u(k)-25;
  y_fit2(k) = (u(k))^3*c(11,1 ) + (u(k))^2*c( 11,2 ) + (u(k))*c( 11,3) + c(11,4 )
  d_fit2(k) = 3*(u(k))^2*c(11,1) + 2*(u(k))*c(11,2) + c(11,3);
  fit(r,k) = y_fit2(k);
  d_yfit2(r,k) = d_fit2(k);
elseif u(k)> 27.5 & u(k) <=30
  u(k) = u(k)-27.5;
  y_fit2(k) = (u(k))^3*c(12,1 ) + (u(k))^2*c( 12,2 ) + (u(k))*c( 12,3) + c(12,4 )
  d_fit2(k) = 3*(u(k))^2*c(12,1) + 2*(u(k))*c(12,2) + c(12,3);
  fit(r,k) = y_fit2(k);
  d_yfit2(r,k) = d_fit2(k);
elseif u(k)> 30 & u(k) <=32.5
  u(k) = u(k)-30;
  y_fit2(k) = (u(k))^3*c(13,1 ) + (u(k))^2*c( 13,2 ) + (u(k))*c( 13,3) + c(13,4 )
  d_fit2(k) = 3*(u(k))^2*c(13,1) + 2*(u(k))*c(13,2) + c(13,3);
  fit(r,k) = y_fit2(k);
  d_yfit2(r,k) = d_fit2(k);
elseif u(k)> 32.5 & u(k) <=35
  u(k) = u(k)-32.5;
  y_fit2(k) = (u(k))^3*c(14,1 ) + (u(k))^2*c( 14,2 ) + (u(k))*c( 14,3) + c(14,4 )
  d_fit2(k) = 3*(u(k))^2*c(14,1) + 2*(u(k))*c(14,2) + c(14,3);
  fit(r,k) = y_fit2(k);
  d_yfit2(r,k) = d_fit2(k);
elseif u(k)> 35 & u(k) <=37.5
  u(k) = u(k)-35;
  y_fit2(k) = (u(k))^3*c(15,1 ) + (u(k))^2*c( 15,2 ) + (u(k))*c( 15,3) + c(15,4 )
  d_fit2(k) = 3*(u(k))^2*c(15,1) + 2*(u(k))*c(15,2) + c(15,3);
  fit(r,k) = y_fit2(k);
  d_yfit2(r,k) = d_fit2(k);
```

```
elseif u(k)> 37.5 & u(k) <=40
  u(k) = u(k)-37.5;
  y_fit2(k) = (u(k))^3*c(16,1 ) + (u(k))^2*c( 16,2 ) + (u(k))*c( 16,3) + c(16,4 )
  d_fit2(k) = 3*(u(k))^2*c(16,1) + 2*(u(k))*c(16,2) + c(16,3);
  fit(r,k) = y_fit2(k);
  d_yfit2(r,k) = d_fit2(k);
elseif u(k)> 40 & u(k) <=42.5
  u(k)=u(k)-40;
  y_fit2(k) = (u(k))^3*c(17,1 ) + (u(k))^2*c( 17,2 ) + (u(k))*c( 17,3) + c(17,4 )
  d_fit2(k) = 3*(u(k))^2*c(17,1) + 2*(u(k))*c(17,2) + c(17,3);
  fit(r,k) = y_fit2(k);
  d_yfit2(r,k) = d_fit2(k);
elseif u(k)> 42.5 & u(k) <=45
  u(k) = u(k)-42.5;
  y_fit2(k) = (u(k))^3*c(18,1 ) + (u(k))^2*c( 18,2 ) + (u(k))*c( 18,3) + c(18,4 )
  d_fit2(k) = 3*(u(k))^2*c(18,1) + 2*(u(k))*c(18,2) + c(18,3);
  fit(r,k) = y_fit2(k);
  d_yfit2(r,k) = d_fit2(k);
else
  fprintf('wrong number');
end
end
d_yfit2(r,1) = 0;
d_yfit2(r,length(u)) = 0;
figure;
plot(x,y,'-o',theta,fit(r,:),':r');          % comparison between a3 and spline
legend('a3','spline');
xlabel('Rotor position (deg)');
ylabel('a3 (Wb/A^3)');
end
end % end r loop
```

```
%%% plot new magnetizing curve
pos=19;
cur = 0:5:25;
figure;
hold;
lam = zeros(pos,length(c));
for p=1:length(u)
   for n=2:20
   mag=magdata;
   mag1=mag(:,1);
   mag2=mag(:,n);
   plot(mag1,mag2,'bo')
   end
% reconstruct of flux linkage obtained by cubic spline.
 new_fit = fit(2,p)*cur + fit(3,p)*cur.^2 + fit(4,p)*cur.^3;
 % plot current vs flux linkage obtained by spline
         if p <=length(u)/2
           % plot(c,new_fit,'k-');    %% black is negative slope
   else
         plot(cur,new_fit,'r--');    %% red is positive slope

   xlabel('Current (A)');
   ylabel('Flux linkage (Wb)');
   grid on;
   end

 end
 % Now let find the inductance profile.
 c=0:3:15;
 for p=1:length(u)
         L = fit(2,p) + 2*fit(3,p)*c + 3*fit(4,p)*c.^2;
```

```
        %S_L= store inductance variable
        S_L(p,:)=L;
        % lam(p,:)=new_fit;
end
figure;
hold on;
grid on;
theta2=0:0.5:45;
xlabel('Rotor position (deg)');
ylabel('Inductance (H)');
for x=1:length(c)
   hold on;
   if x = =1
      plot(theta,S_L(:,x),'-*');
   end
   if x = = 2
      plot(theta,S_L(:,x),'-o');
   end
   if x = =3
      plot(theta,S_L(:,x),'-s');
   end
   if x = =4
      plot(theta,S_L(:,x),'-d');
   end
   if x = =5
      plot(theta,S_L(:,x),'-v');
   end
   if x = =6
      plot(theta,S_L(:,x),'-^');
   end
   legend('0A','3A','6A','9A','12A','15A');
end
```

```
%%% Calculate static torque
d_yfit2=d_yfit2.*180/pi;
c  = [0:5:25];
%torque_1=zeros(pos,length(c));
figure;hold on;
x = [0:2.5:45];
for p =1:length(u),
    torque = ((1/2)*c.^2*d_yfit2(2,p) + (1/3)*c.^3*d_yfit2(3,p) + (1/4)*c.^4*d_yfit2(4,p));
    torque_1(p,:) = torque ;
end
t;   % use t variable that declared at the beginning.
plot(x,t(1,:),'--',x,t(2,:),'--',x,t(3,:),'--',x,t(4,:),'--',x,t(5,:),'--',x,t(6,:),'--');
t1=linspace(0,45,18);
for p=1:length(c),
        plot(theta,torque_1(:,p));
        %plot(x,torque_1(:,p));
        xlabel('Rotor position (deg)');
        ylabel('Torque (N.m)');
        grid on;
end
```

The next piece of verbatim code is the polynomial fitting equation that sends error to the main program.

```
function f = expstp(a,Data)
%EXPSTP used by dd.m to return errors in fitting data to a function.
%EXPSTP returns the error between the data and the values computed by the
%current function of a.  EXPSTP assumes a function of the form
x = Data(:,1);
y = Data(:,2);
z = a(1)*x + a(2)*x.^2  + a(3)*x.^3;                    %used in Magnetic_Model.m
f = z - y;
```

BIOGRAPHY

Mr. Sisavath Khotpanya was born in June 19, 1979 in Borlikhamxay province, Laos. He received B. Eng in Electrical Engineering Department, Faculty of Engineering and Architecture, National University of Laos, in the year of 2002. Eversince, he has been an assistance lecturer in Electrical Engineering Department, National University of Laos. He is a postgraduate student of KMITL, Thailand under the AUN/SEED-Net program.