Analysis and Design of a Novel Controller Architecture and Design Methodology for Speed Control of Switched Reluctance Motors

by

Terry W. Jackson

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

IN

ELECTRICAL ENGINEERING

APPROVED:

Dr. K. Ramu, Chair

Dr. C. E. Nunnally

Dr. H. F. VanLandingham

July 1996

Blacksburg, Virginia

Keywords: Switched Reluctance Motor, Electric Drives, Linearization, Controller, Control Design

ANALYSIS AND DESIGN OF A NOVEL CONTROLLER ARCHITECTURE AND DESIGN METHODOLOGY FOR SPEED CONTROL OF SWITCHED RELUCTANCE MOTORS

by

Terry W. Jackson

Dr. Krishnan Ramu, Chairman

Bradley Department of Electrical Engineering

(Abstract)

This paper presents a novel controller architecture and speed control design methodology suitable for low cost, low performance switched reluctance motor drives. By utilizing inexpensive components in a simple, compact architecture, a low cost controller is developed which achieves a performance level similar to many high performance controllers. A speed control design methodology is established and analyzed based on the linearized small signal model of the switched reluctance motor. This unique control methodology is simple and provides a starting point for further research into speed/current controller parameter design for switched reluctance motors. The analysis, design and realization of the speed controller are presented. The derivation of the design methodology for speed controlled, switched reluctance motor drives is discussed, along with computer simulations for verification. Experimental results utilizing the proposed architecture and design methodology verify the control design and performance capabilities of the speed controller system.

Disadvantages to SRM drive technology:

- Requires two many switching components!
- Requires complex commutation logic!
- Motor is rugged and cheap but very big and heavy!
- Noisy

Acknowledgments

I thank Prof. Krishnan Ramu for his continuing guidance, advice and support through the course of my study. A special thanks to Shiyoung Lee for building the converter and helping with the experimental setup. I would like to thank the United States Nuclear Regulatory Commission for their financial support of my study, and for funding the research activity. Most of all, I thank my Lord and Savior, Jesus Christ, who provided me with the opportunity and wisdom for graduate study. I dedicate this work to my wife Katrina, and thank her for the encouragement she has given.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

The Switched Reluctance Motor(SRM) is a rugged, inexpensive motor that is seeing use in several applications[1],[2],[3]. Because the SRM is doubly salient with concentric windings on the stator and no windings on the rotor, the cost of motor construction is low compared to other motors. Since each phase winding of the SRM is independent of the other phase windings, the SRM is able to operate with some of the phase windings disabled, at a lower power output[4]. In SRM converters, there is a phase winding between the two phase switches, making shoot-through faults less of a hazard. The independence of phase windings and decrease in the potential for shoot-through faults makes the SRM a very reliable motor. Other advantages of the SRM include high starting torque and high power density. The disadvantages of the SRM include large torque ripple and significant nonlinearities which complicate current, speed and position control.

Because of its reliability and cost, the SRM is a competitive candidate for many low performance drive applications[5]. In developing low performance drives, several aspects are important to a drive system. Some of the main aspects include cost, reliability and efficiency. Cost is most important, and is relative to the amount of performance a drive provides. Due to long periods of operation and the possibility of hostile environments, reliability is required. Because of the long operating periods, higher efficiency significantly reduces the electric utility cost to operate the drive.

The problem this work addresses is the need for a low cost SRM speed controller which maintains an acceptable level of performance[6]. The controller should be flexible for the various types of SRMs, as well as, the many applications in which they are applied. Currently, there are few analytical methods in which to design the SRM to meet speed performance specifications. This problem is also addressed in this work.

Several speed controllers have been proposed for SRMs. Most controllers are digitally based and require digital signal processors or high-end microcontrollers as part of

their architecture[7],[8]. While some of these controllers are capable of many control features, they are not suitable for low performance applications because the cost of the controllers reduces the cost effectiveness of the drive system. There have been proposals for low cost SRM controllers which enable the SRM to be competitive in the low performance drive market[9]. However, these controllers lack the necessary flexibility and performance capabilities, excluding the SRM from many of the low performance drive applications.

Besides the need for a low cost controller, there is a lack of systematic design techniques which allow engineers to set control parameters. Previous work has dealt with design of speed and current controllers using feedback linearization[10],[11]. This type of precise control is not necessary for low performance and adds to the cost and complexity of the controller.

The purpose of this work is to develop a speed controller architecture and control design methodology suitable for SRMs in low performance applications. There are three specific goals to obtain. First, the cost of the controller should fall below $20(US). Second, the controller must be able to maintain a level of performance which is common to general purpose motor drive controllers. Some primary capabilities of general purpose motor drive controllers include four quadrant operation, adaptability to specific applications and the ability to achieve the highest level of efficiency. The third goal is to develop a technique in which the speed/current controller gains are analytically calculated and the performance analyzed by simulation.

The architecture of the SRM controller is a hybrid architecture, meaning it consists of both analog and digital hardware. There are three control blocks in an SRM. The first two control blocks are the current control and the speed control. Both of these types of control are accomplished in analog hardware. The third control block is the phase firing control, which is performed in digital hardware. The controller provides an interface between the analog and digital hardware for control coordination, and it also provides an

interface to the converter, motor and user. The following lists the main features of the proposed SRM controller:

- Standard +/- 10V user speed command input
- Four quadrant speed control
- New current sensor placement on the converter allowing current control with one current sensor
- Experimental, low cost absolute/incremental encoder interface
- Capable of both advance and fall angle control
- Torque smoothing
- Compact

For industrial drives, such as induction motor and dc motor drives, there are systematic design techniques for current/speed control loop design, which allows engineers to analyze their designs before construction. This work provides a simple design technique based on the small signal linearized model of the SRM. For low performance drive applications, this design technique is sufficient for choosing PI gains for the speed and current controllers. A prototype SRM is used to illustrate the effectiveness of the design technique in determining the gains of the controllers. Experimental and simulation results are given to illustrate the effect of varying design choices in the design methodology.

The thesis is organized as follows. Chapter 2 details the derivation and use of the design methodology for obtaining speed/current loop PI gains. Chapter 3 describes the torque smoothing process and lists its advantages and limitations. Chapter 4 discusses the SRM controller architecture; both hardware and software. Chapter 5 discusses the experimental verification of the system. Conclusions are presented in chapter 6.

Chapter 2: Design of SRM Speed and Current Controllers

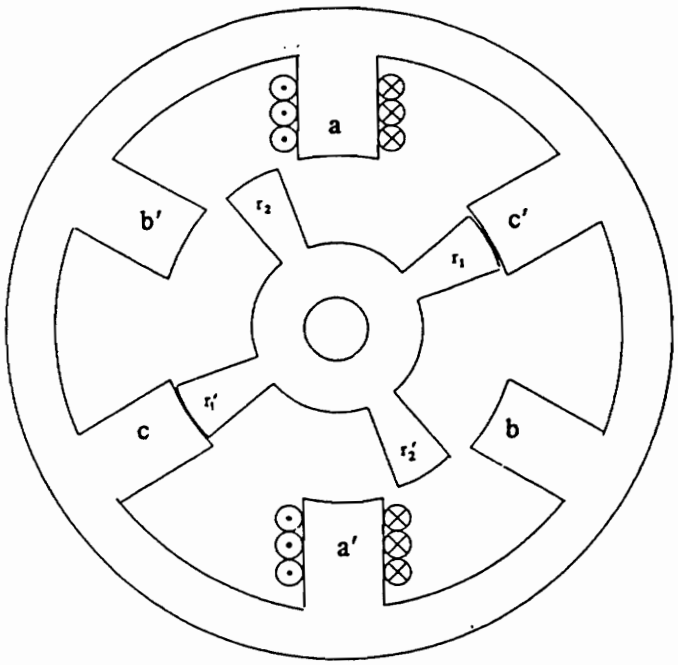
2.1 Introduction

The Switched Reluctance Motor(SRM) is a viable candidate for many variable speed applications. However, there has been little research into design methodologies for SRM speed controllers. Because of the SRM's nonlinear nature, it is difficult to construct a design model of the motor itself. The SRM's torque is proportional to the input current squared, and the back EMF is proportional to the product of current and speed. This work proposes the linearization of the SRM's voltage and torque equations to a small signal model about the rated current and speed. A design example, using a prototype 5hp SRM, is given.

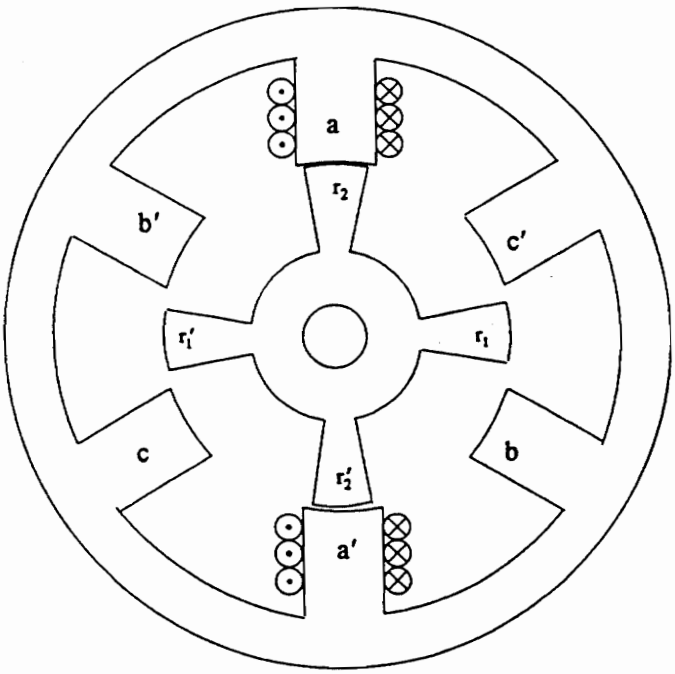
2.2 Basic Operation of the Switched Reluctance Motor

The SRM uses basic electromagnetic principles to convert electrical energy into mechanical energy. Consider the crossection of a 6/4 SRM shown on the next page in Fig. 2.1[12]. In Fig. 2.1(i), r_1 is aligned with c', and r_1' is aligned with c. To cause the rotor to move in a clockwise direction, the windings around a and a' are energized to create a magnetic force, pulling r_2 to a and r_2' to a'. The result is shown in Fig. 2.1(ii). To move the rotor counterclockwise in Fig. 2.1(i), the windings around b and b' are energized to pull r_2 and r_2' into alignment with b and b'.

Mechanical rotation is achieved by switching on the phases of the SRM in a sequence which depends on the direction of rotation. For example, if clockwise rotation is desired, the phases are energized in the a-b-c-d sequence. For counterclockwise rotation, the reverse of the sequence is executed.



(i) Phase c aligned.



(ii) Phase a aligned.

Fig. 2.1 Basic operation of the SRM

An ideal inductance profile of the SRM is used to determine the polarity of the torque produced. Shown in Fig. 2.2 is an ideal inductance profile of an 8/6 SRM.

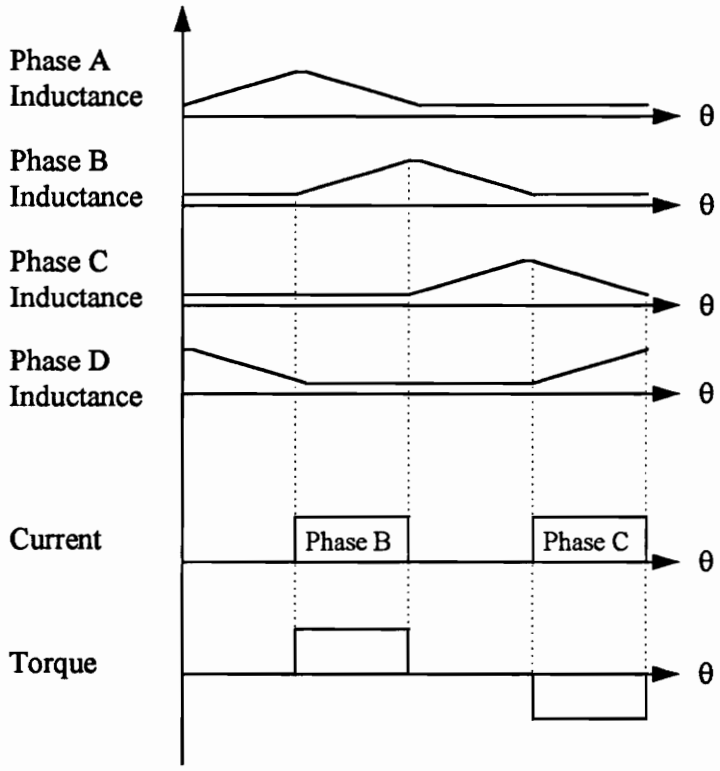


Fig. 2.2 Ideal inductance profile of an 8/6 SRM

The phase inductances are shown for 60 mechanical degrees only. For each 60 degrees in an 8/6 SRM, the phase inductance pattern repeats. At the points where inductance is the highest, a set of rotor poles are in alignment with two opposing stator poles. Two opposing stator poles make one phase.

Shown below the phase inductances in Fig. 2.2 is an example of torque production in the SRM. If current is applied at the rising slope of the inductance, positive torque is produced. If current is applied at the falling slope of the inductance, negative torque is

produced. Notice that the polarity of the current makes no difference. Also, notice that torque production is dependent on the rotor angle, or rotor position. Appendix C shows the ideal inductance profile for the prototype 5hp SRM.

Fig. 2.2 portrays an ideal situation. In reality, the inductance and torque are nonlinear functions dependent upon the current and rotor position. Fig. 2.3 shows the flux linkages for one phase of the prototype SRM. Flux linkages are equal to inductance times current. The flux linkages are only shown for 30 degrees of the full 60 degree cycle. The other 30 degrees of the flux linkages pattern are a mirror image of Fig. 2.3. At zero degree, a set of rotor poles are in full alignment with a set of stator poles.

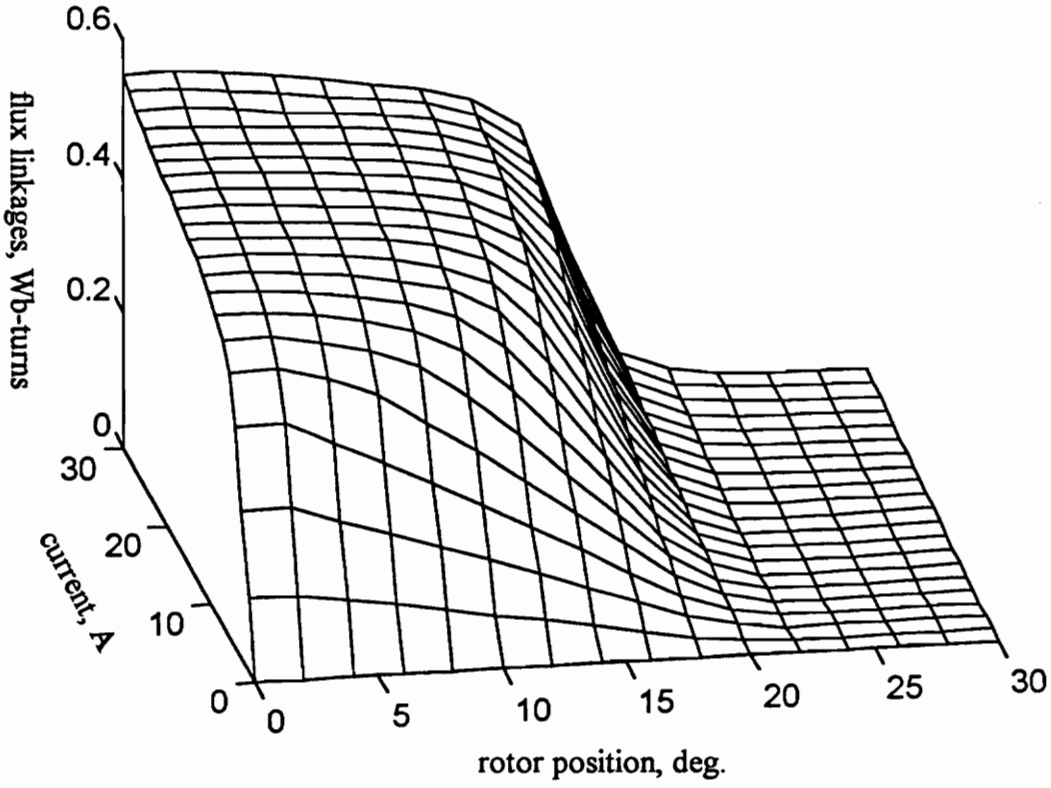


Fig. 2.3 Flux linkages as a function of rotor position and current

Fig. 2.4 shows the torque over one-half cycle for one phase of the prototype SRM. Again, zero degree is full alignment for one set of rotor poles and one set of stator poles. The other half cycle is a mirror image of Fig. 2.4.

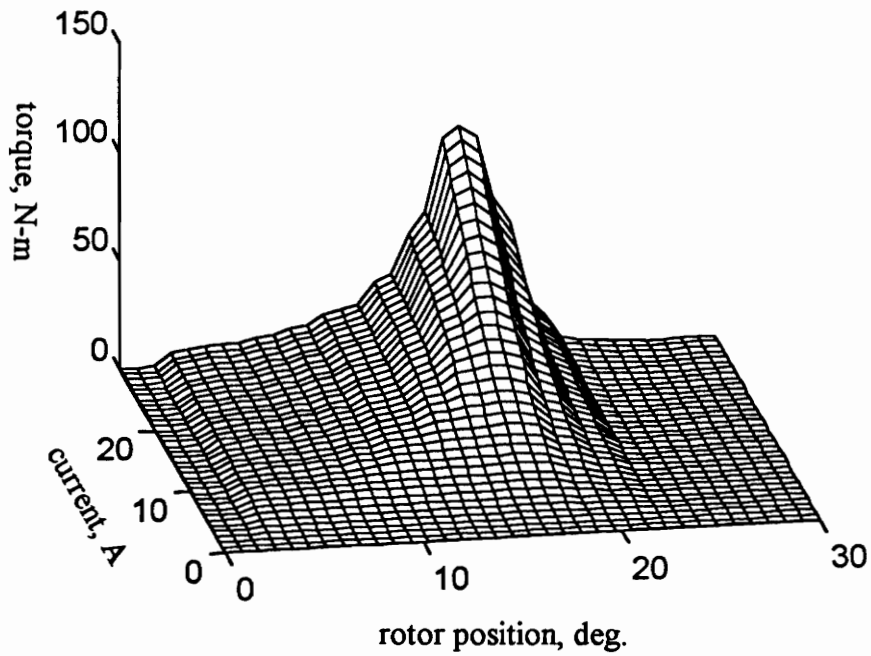


Fig. 2.4 Electromagnetic torque as a function of rotor position and current

2.3 Voltage and Torque Equations for the SRM

The small signal model of the SRM is derived from the SRM's voltage and torque equations shown below:

$$V = R_p i + \frac{d\lambda(\theta,i)}{dt}, \qquad \text{Voltage Equation} \qquad (2.1)$$

$$T_e(\theta,i) - T_l = J\frac{d\omega}{dt} + B\omega, \qquad \text{Torque Equation} \qquad (2.2)$$

In the above equations, the flux linkages, λ, and the airgap torque, T_e , are dependent upon the phase current and the rotor position. It is desirable to have the voltage equation and the torque equation in a form similar to the separately excited DC motor. The separately excited DC motor is the easiest of all electric motors to work with, and standard design procedures are established for it. By using the relation $L(\theta,i)i = \lambda(\theta,i)$ the voltage equation becomes the following:

$$V = R_p i + \frac{dL(\theta,i)i}{dt} = R_p i + L(\theta,i)\frac{di}{dt} + \frac{dL(\theta,i)}{d\theta}\omega_m i \tag{2.3}$$

In (2.3), the three terms on the right hand side represent the resistive, inductive and back EMF terms respectively; similar to the separately excited DC motor voltage equation.

Likewise, it is desirable to arrange the torque equation into a form similar to that of the separately excited DC motor torque equation. The torque is related to the current in the SRM phase windings. The following shows the relationship between the torque and the current in one phase winding of the SRM.

Energy in an inductor:

$$\frac{d}{dt}\left(\frac{1}{2}L(\theta,i)i^2\right) = L(\theta,i)i\frac{di}{dt} + \frac{1}{2}i^2\frac{dL(\theta,i)}{dt} \tag{2.4}$$

Power in the SRM:

$$P = Vi = R_p i^2 + i^2\frac{dL(\theta,i)}{dt} + L(\theta,i)i\frac{di}{dt} \tag{2.5}$$

By substituting (2.4) into (2.5), the result is the following:

$$P = R_p i^2 + i^2\frac{dL(\theta,i)}{dt} + \frac{d}{dt}\left(\frac{1}{2}L(\theta,i)i^2\right) - \frac{1}{2}i^2\frac{dL(\theta,i)}{dt}$$

$$P = R_p i^2 + \frac{1}{2} i^2 \frac{dL(\theta,i)}{dt} + \frac{d}{dt}\left(\frac{1}{2}L(\theta,i)i^2\right) \tag{2.6}$$

The power in the SRM is represented by the three terms in (2.6), which are the copper losses, airgap power and rate of change of stored energy in the phase. The airgap power is used to determine the airgap torque by the relation that airgap power is equal to airgap torque times the rotor speed.

$$P_a = \frac{1}{2} i^2 \frac{dL(\theta,i)}{dt} = \frac{1}{2} i^2 \frac{dL(\theta,i)}{d\theta}\omega_m, \qquad \text{Airgap Power} \tag{2.7}$$

$$T_e = \frac{1}{2} i^2 \frac{dL(\theta,i)}{d\theta}, \qquad \text{Airgap Torque} \tag{2.8}$$

The resulting torque equation for the SRM is the following:

$$\frac{1}{2} i^2 \frac{dL(\theta,i)}{d\theta} - T_l = J\frac{d\omega}{dt} + B\omega \tag{2.9}$$

2.4 Derivation of the SRM Small Signal Model

The states of the SRM plant are the rotor speed, ω_m, and the phase current, i. In examining the SRM voltage and torque equations, there are terms where states are multiplied together, resulting in a nonlinear system. For low performance, variable speed SRM drives, it is desirable to have a simple model in order to design the speed and current controllers. The simplest approach is linearization of the actual model. This work takes the approach of deriving a linearized small signal model of the SRM system equations, which has not been mentioned in literature to date.

The SRM system model may be written in the following form:

$$\dot{x} = \begin{bmatrix} \left(\frac{u_1}{L} - \frac{R_p}{L}x_1 - \frac{1}{L}\frac{dL}{d\theta}x_1x_2\right) \\ \left(\frac{1}{2J}x_1^2\frac{dL}{d\theta} - \frac{u_2}{J} - \frac{B}{J}x_2\right) \end{bmatrix} = f(x,u,t) \tag{2.10}$$

$x_1 = i \qquad x_2 = \omega_m \qquad u_1 = V \qquad u_2 = T_l$

In the above equation, the inductance $L = L(\theta,i)$ is assumed to be constant for sake of simplification. The inductance is chosen as the mean value between the aligned inductance and the unaligned inductance at rated current. The derivative of inductance with respect to rotor position is also assumed a constant and calculated between the conduction angles at the rated current value. This derivative has only a small change over the operating range of the motor.

By using the linearization analysis of small deviations from the nominal[13], the following is the derivation of the small signal model.

$$x(t) = x_n(t) - \delta x \tag{2.11}$$

$$u(t) = u_n(t) - \delta u \tag{2.12}$$

$$\dot{x}(t) = \dot{x}_n(t) - \delta\dot{x} = f(x_n,u_n,t) + \left[\frac{\partial f}{\partial x}\right]\delta x + \left[\frac{\partial f}{\partial u}\right]\delta u \tag{2.13}$$

$$\delta\dot{x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} \tag{2.14}$$

$$\delta\dot{u} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} \end{bmatrix} \tag{2.15}$$

$$\frac{\partial f_1}{\partial x_1} = -\frac{R_p}{L} - \frac{1}{L}\frac{dL}{d\theta}x_{20} \tag{2.16}$$

$$\frac{\partial f_1}{\partial x_2} = -\frac{1}{L}\frac{dL}{d\theta}x_{10} \tag{2.17}$$

$$\frac{\partial f_2}{\partial x_1} = -\frac{1}{J}\frac{dL}{d\theta}x_{10} \tag{2.18}$$

$$\frac{\partial f_2}{\partial x_2} = -\frac{B}{J} \tag{2.19}$$

$$\frac{\partial f_1}{\partial u_1} = \frac{1}{L} \tag{2.20}$$

$$\frac{\partial f_2}{\partial u_2} = -\frac{1}{J} \tag{2.21}$$

$$\frac{\partial f_1}{\partial u_2} = \frac{\partial f_2}{\partial u_1} = 0 \tag{2.22}$$

where x_{10} is the rated current and x_{20} is the rated speed

While the total linearized system contains a steady state portion, $x_n(t)$ and $u_n(t)$, that has a constant value determined by the rated speed and current of the SRM, the small signal deviation portion of the linearized system is of more interest in developing the design model. Therefore, the steady state portion is ignored in order to develop the design model. The following is the set of linearized small signal voltage and torque equations:

$$\frac{d\delta i}{dt} = \left(-\frac{R_p}{L} - \frac{1}{L}\frac{dL}{d\theta}\omega_{mo}\right)\delta i - \frac{1}{L}\frac{dL}{d\theta}i_o\delta\omega_m + \frac{\delta V}{L} \tag{2.23}$$

$$\frac{d\delta\omega_m}{dt} = \left(-\frac{1}{J}\frac{dL}{d\theta}i_o\right)\delta i - \frac{B}{J}\delta\omega_m + \frac{\delta T_l}{J} \tag{2.24}$$

After substituting the rated speed, ω_{mo}, and the rated current, i_o, a linear model is achieved which is used to design the controllers. For the rest of the derivation, the following substitutions are used:

$$R = R_p + \frac{dL}{d\theta}\omega_{mo} \tag{2.25}$$

$$K_b = \frac{dL}{d\theta} i_o \tag{2.26}$$

By using the small signal voltage and torque equations, the following block diagram can be drawn to illustrate the linearized SRM plant model. Notice that this model is similar to the separately excited DC motor model.

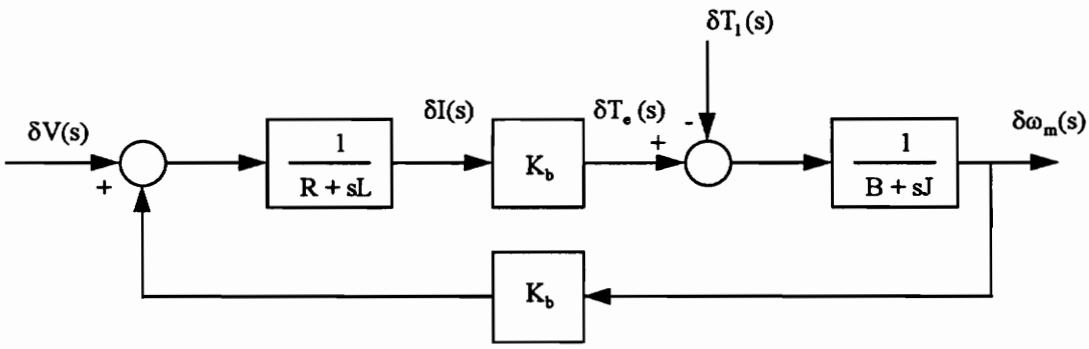


Fig. 2.5 Block diagram of SRM plant model

It is desirable to reduce the block diagram model in order to eliminate the back EMF feedback loop. The back EMF feedback loop crosses over the current feedback loop and complicates the controller design. Therefore, by making the torque load equal to a friction constant B_l times the speed, the block diagram is reduced to the following[14]:

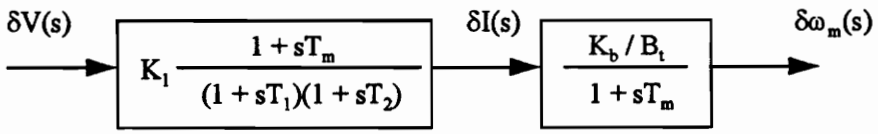


Fig. 2.6 Reduced block diagram of SRM plant model

where,

$$B_t = B + B_l \tag{2.27}$$

$$K_1 = \frac{B_t}{K_b^2 + RB_t} \tag{2.28}$$

$$T_m = \frac{J}{B_t} \tag{2.29}$$

$$-\frac{1}{T_1}, -\frac{1}{T_2} = -\frac{1}{2}\left[\frac{B_t}{J} + \frac{R}{L}\right] \pm \sqrt{\frac{1}{4}\left(\frac{B_t}{J} + \frac{R}{L}\right)^2 - \frac{K_b^2 + RB_t}{JL}} \tag{2.30}$$

2.5 Designing the Current Controller With the Linearized Model

For reference, the entire block diagram of the SRM speed-controlled drive is given in Fig. 2.7. For the rest of the report, the small signal model is treated as the design model, and therefore $\delta I(s) = I(s)$, $\delta\omega_m(s) = \omega_m(s)$, etc. Once the linearized model of the SRM plant is identified, the current controller is designed. A PI controller is selected for the current controller because of its simplicity to implement, and the integral portion of the controller prevents the control signal from varying erratically between the saturation levels of the controller.

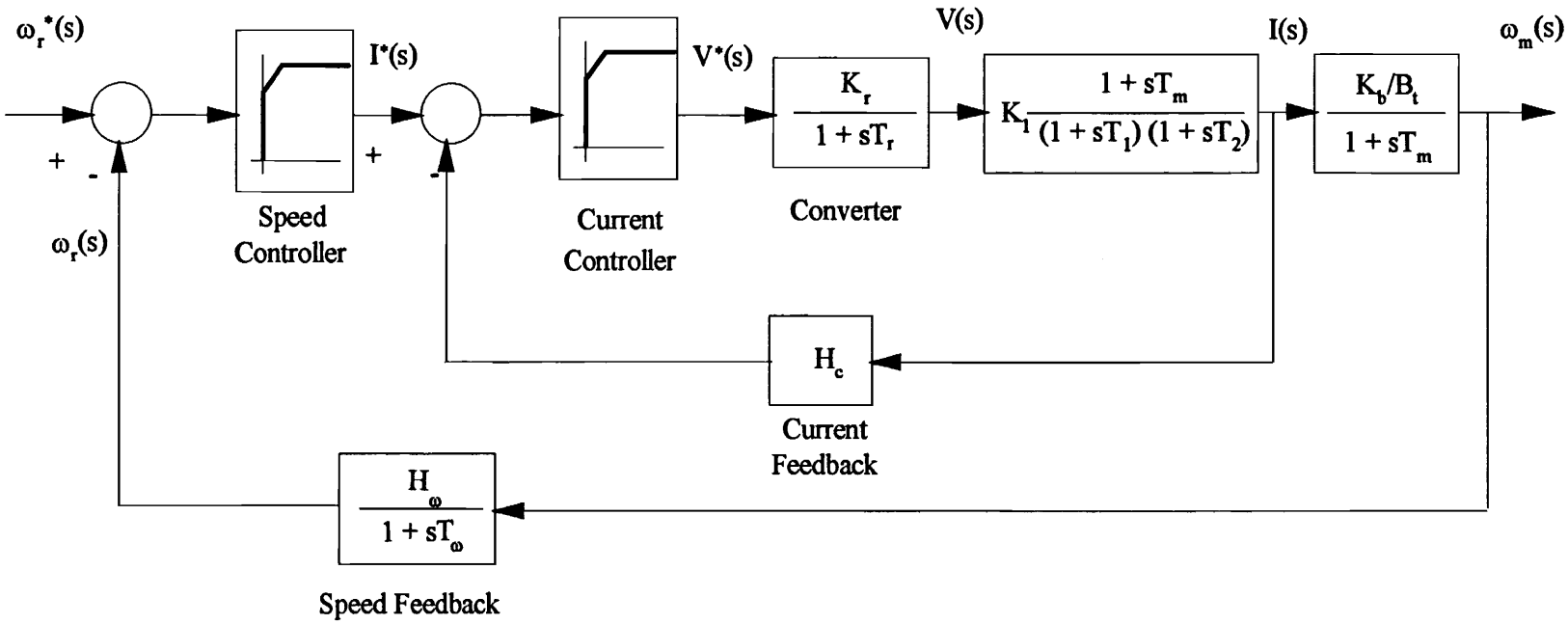


Fig. 2.7 Block diagram of SRM drive

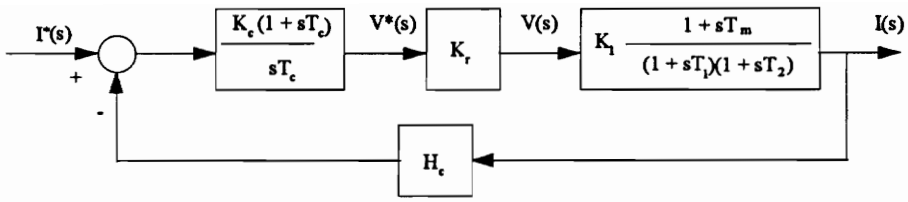


Fig. 2.8 Current loop block diagram

The block diagram of the current loop is given in Fig. 2.8. The converter has a delay, T_r, but it is assumed to be 10 times faster than the current loop delay and not considered in the design process for simplification purposes. The current loop transfer function is shown in (2.31):

$$\frac{I(s)}{I^*(s)} = \frac{b_0 + b_1 s + b_2 s^2}{a_0 + a_1 s + a_2 s^2 + a_3 s^3} \tag{2.31}$$

where,

$$b_0 = K_c K_r K_1 \tag{2.32}$$

$$b_1 = K_c K_r K_1 (T_c + T_m) \tag{2.33}$$

$$b_2 = K_c K_r K_1 T_c T_m \tag{2.34}$$

$$a_0 = H_c K_c K_r K_1 \tag{2.35}$$

$$a_1 = T_c + T_m H_c K_c K_r K_1 + T_c H_c K_c K_r K_1 \tag{2.36}$$

$$a_2 = T_c T_1 + T_c T_2 + T_c T_m H_c K_c K_r K_1 \tag{2.37}$$

$$a_3 = T_c T_1 T_2 \tag{2.38}$$

Since the mechanical time constant of the system, T_m, is large, $(1 + sT_m)$ is approximated as sT_m. With this approximation, the current loop becomes a second order system in which second order system design techniques are applied. Below is the approximated system:

$$\frac{I(s)}{I^*(s)} = \frac{K_c K_r K_1 T_m (1 + sT_c)}{T_c(1 + sT_1)(1 + sT_2) + H_c K_c K_r K_1 T_m (1 + sT_c)} \tag{2.39}$$

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

$$s^2 + s\left(\frac{T_1 + T_2 + H_c K_c K_r K_1 T_m}{T_1 T_2}\right) + \frac{H_c K_c K_r K_1 T_m + T_c}{T_c T_1 T_2} \tag{2.40}$$

Since it is a second-order equation, the natural frequency, ω_n, and damping ratio, δ, are used to obtain the current controller gain and time constant. Below, equations (2.41) and (2.42) specify the damping ratio and the natural frequency of the approximated system.

$$2\delta\omega_n = \frac{T_1 + T_2 + H_c K_c K_r K_1 T_m}{T_1 T_2} \tag{2.41}$$

$$\omega_n^2 = \frac{H_c K_c K_r K_1 T_m + T_c}{T_c T_1 T_2} \tag{2.42}$$

From (2.41) and (2.42), the gain, K_c , and the time constant, T_c , are solved for a given natural frequency and damping ratio. The following are equations for K_c and T_c.

$$K_c = \frac{2\delta T_1 T_2 \omega_n - T_1 - T_2}{H_c K_r K_1 T_m} \tag{2.43}$$

$$T_c = \frac{H_c K_c K_r K_1 T_m}{T_1 T_2 \omega_n^2 - 1} \tag{2.44}$$

Once the gain and time constant of the current controller are found, the unapproximated current loop is simulated for a step response and frequency response.

2.6 Designing the Speed Controller

Once the current loop is defined, the outer speed loop is designed. To design the speed controller gain and time constant, the symmetric optimum method is chosen. The symmetric optimum method gives a flat frequency response over the bandwidth of the system, optimum phase margin, and stability[14]. For low performance systems, the bandwidth is not as much of a concern as the tracking capability of the drive. This is the reason why the symmetric optimum method is chosen for the speed controller design over the bandwidth method.

When designing the speed loop, it is assumed that the delay of the current loop is ten times faster than the response of the speed loop.

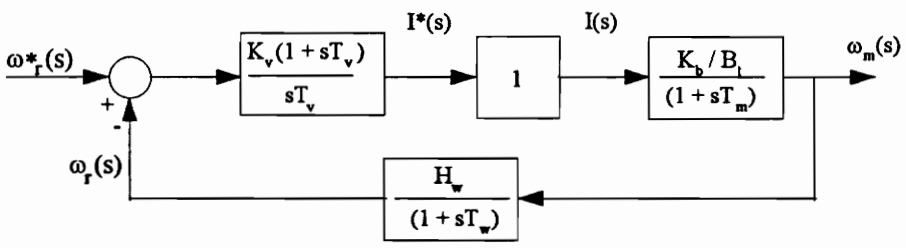


Fig. 2.9 Approximated speed loop block diagram

Therefore, to simplify the design equations, the current loop is approximated as a unity gain. Normally, the delay due to the speed feedback is neglected, which reduces the speed loop to a second-order system. But, when the speed feedback delay is slow compared to

the delay of the other subsystems it must be considered in the design process. Fig. 2.9 is the block diagram of the approximated speed loop. The transfer function for the approximated speed loop is the following:

$$\frac{\omega_m(s)}{\omega_r^*(s)} = \frac{K_b K_v (1 + T_v)(1 + T_w)}{T_v H_c B_t s(1 + sT_m)(1 + T_w) + H_w K_b K_v (1 + sT_v)} \tag{2.45}$$

The design by symmetric optimum is made easier if the transfer function is in the following form:

$$\frac{\omega_m(s)}{\omega_r^*(s)} = \frac{d_0 + d_1 s + d_2 s^2}{c_0 + c_1 s + c_2 s^2 + c_3 s^3} \tag{2.46}$$

where,

$$d_0 = K_v K_b \tag{2.47}$$

$$d_1 = K_v K_b (T_v + T_w) \tag{2.48}$$

$$d_2 = K_v K_b T_v T_w \tag{2.49}$$

$$c_0 = H_w K_b K_v \tag{2.50}$$

$$c_1 = H_w K_b K_v T_v + B_t T_v \tag{2.51}$$

$$c_2 = (T_m + T_w) B_t T_v \tag{2.52}$$

$$c_3 = T_m T_w B_t T_v \tag{2.53}$$

To develop a speed controller which gives symmetric optimum performance, the frequency magnitude response is taken of equation (2.46).

$$\left|\frac{\omega_m(j\omega)}{\omega_r^*(j\omega)}\right| = \sqrt{\frac{d_0^2 + (d_1^2 - 2d_0 d_2)\omega^2 + d_2^2\omega^4}{c_0^2 + (c_1^2 - 2c_0 c_2)\omega^2 + (c_2^2 - 2c_1 c_3)\omega^4 + c_3^2\omega^6}} \tag{2.54}$$

From equation (2.54), the two middle coefficients of denominator need to be zero for the system to be symmetric optimum. Therefore, the two following conditions must hold:

$$c_1^2 = 2c_0c_2 \tag{2.55}$$

$$c_2^2 = 2c_1c_3 \tag{2.56}$$

By using equation (2.56), the speed controller gain is obtained, and by using equation (2.55), the speed controller time constant is calculated.

$$K_v = \frac{B_t((T_m + T_w)^2 - 2T_mT_w)}{2K_bT_mT_wH_w} \tag{2.57}$$

$$T_v = \frac{2H_wK_bK_v(T_m + T_w)B_t}{(B_t + H_wK_bK_v)^2} \tag{2.58}$$

Following the calculation of the speed controller gains, the step response is obtained from the approximated speed loop. To get a better view of how the entire system responds, the unapproximated speed loop step response and frequency response must be obtained. The unapproximated speed loop transfer function includes the transfer function of the current loop and has the following form:

$$\frac{\omega_m(s)}{\omega_r^*(s)} = \frac{e_0 + e_1s + e_2s^2 + e_3s^3 + e_4s^4}{f_0 + f_1s + f_2s^2 + f_3s^3 + f_4s^4 + f_5s^5 + f_6s^6} \tag{2.59}$$

where a_0 to a_3 and b_0 to b_2 are from equations (2.32) to (2.38) and,

$$e_0 = b_0K_bK_v \tag{2.60}$$

$$e_1 = (b_1 + b_0(T_v + T_w))K_bK_v \tag{2.61}$$

$$e_2 = (b_2 + b_1(T_v + T_w) + b_0T_vT_m)K_bK_v \tag{2.62}$$

$$e_3 = (b_1T_vT_m + b_2(T_v + T_w))K_bK_v \tag{2.63}$$

$$e_4 = T_v T_m K_b K_v b_2 \tag{2.64}$$

$$f_0 = K_b K_v H_w b_0 \tag{2.65}$$

$$f_1 = B_t T_v a_0 + b_0 H_w K_b K_v + K_b K_v H_w b_1 \tag{2.66}$$

$$f_2 = B_t T_v a_1 + b_2 H_w K_b K_v + T_v K_b K_v H_w b_1 + T_v B_t (T_m + T_w) a_0 \tag{2.67}$$

$$f_3 = B_t T_v a_2 + a_0 T_w T_m B_t T_v + T_v K_b K_v H_w b_2 + T_v B_t (T_m + T_w) a_1 \tag{2.68}$$

$$f_4 = B_t T_v a_3 + a_1 T_w T_m B_t T_v + T_v B_t (T_m + T_w) a_2 \tag{2.69}$$

$$f_5 = a_2 T_w T_m B_t T_v + T_v B_t (T_m + T_w) a_3 \tag{2.70}$$

$$f_6 = a_3 T_w T_m B_t T_v \tag{2.71}$$

After simulating the step response of the unapproximated system, there is often a large overshoot due to the time constant of the speed controller. To minimize the overshoot of the system, it is desirable to cancel the zero due to the speed controller time constant with a pole of the same magnitude. This is a "soft start" for the drive system and it implemented with a simple R-C circuit. Upon designing the soft start, the system is simulated once more to observe the change in the step response and frequency response.

2.7 Calculating the Advance and Fall Angles

Besides controlling the current and speed of the machine, the points at which current is applied to each individual phase is controlled. It is desirable to apply a square current waveform at the conduction period of the phase. However, this is not practical since the motor phase has a considerable amount of inductance and it takes time for the current to rise and fall. Also, when speed begins to increase, the back EMF increases, reducing the effective applied voltage to the phase. The effect of back EMF increases the time required for the current to rise to the commanded level.

To combat this problem, the current for each phase is switched on earlier so the current is at commanded level when the rising slope of the inductance is reached.

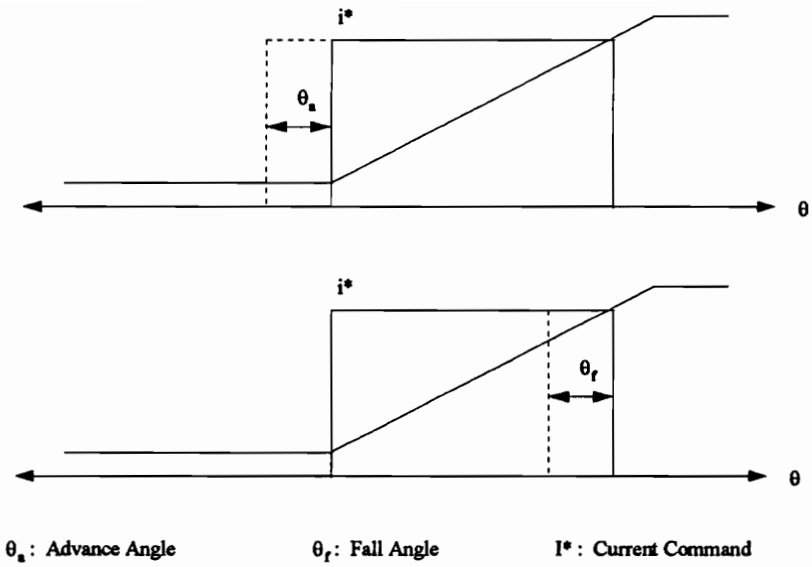


Fig. 2.10 Illustration of advance and fall angles

The switching of current is determined by the rotor position, and the advance angle, θ_a, is the amount of advancement in the current switch-on angle for a particular speed.

Similarly, the fall angle, θ_f, is the advancement in the current switch-off angle. When a phase is switched off, the current does not instantaneously fall, but takes time to recover to zero because of the inductance. The fall angle is used to prevent the recovery current from entering the opposite inductance slope region and producing an opposing torque. It is not uncommon to have both an advance angle and fall angle acting on a phase firing at the same instant. Fig. 2.10 illustrates the advance and fall angles.

Besides being used to obtain the desired current waveform in the presence of inductance, advance and fall angles are used to increase efficiency. For each SRM, the torque profile is different. For some SRMs, the majority of the torque is produced at the bottom of the inductance slopes, while others may have an equal amount of torque produced throughout the phase excitation. The advance and fall angles are used to focus the current to the rotor positions which produce the highest torque-to-current ratio. In doing this, the efficiency of the drive increases because the amount of current to obtain a certain amount of torque decreases.

The calculation of the advance and fall angles for a range of speeds is straightforward. Each phase of the SRM is an R-L circuit, and the time for the current to rise or fall to a specified current is found from the step response. The amount of time for the current to rise/fall is multiplied by the speed to find the amount of advance or fall angle.

$$i(t) = \frac{V - V_e}{R} + \left(i_o - \frac{V - V_e}{R}\right) e^{\frac{-t}{\tau}} \tag{2.72}$$

$$\theta = \omega_m \cdot t \tag{2.73}$$

where,

θ = advance or fall angle, rad

ω_m = rotor speed, rad / s

L = inductance, H

R = phase resistance, Ω

V = DC Link voltage + back EMF, V

$V_e = K_{b1}\omega_m$ (back EMF)

K_{b1} = assumed back EMF constant

τ = R-L time constant, H/Ω

t = time, sec

i_o = initial current

For advance angles, the unaligned inductance is used, and for fall angles, the aligned inductance is used. The largest amount of time for the current to fall is associated with the largest amount of current allowed in the SRM. Therefore, inductance is selected with respect to maximum current. At the point of advancement and current cut-off, there may be a back EMF present. The back EMF slows the rise of current and helps cut-off the current.

2.8 Design of the Prototype 5hp SRM Drive System

To validate the design technique using the linearized model, an SRM drive system based on a 5hp SRM is designed. The specifications for the 5 hp SRM drive are listed in Appendix B. The inductance of the SRM phases is assumed to be the mean value of the unaligned inductance and aligned inductance at the rated current. This value turns out to be 22.1mH. The slope of the inductance curve is needed in order to calculate the linearized torque/back EMF constant. Using the inductance values at rated current, the approximate slope of the inductance profile is 0.234 H/rad. By using (2.26) and (2.27), the linearized torque/back EMF constant, K_b, and the linearized phase resistance, R, are calculated.

$$R = R_p + \frac{dL}{d\theta}\omega_{mo} = 0.931 + (0.234)(261) = 62\,\Omega \tag{2.74}$$

$$K_b = \frac{dL}{d\theta} i_o = (0.234)(12) = 2.81 \tag{2.75}$$

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

(1) <u>Converter Gain</u>

$$K_r = \frac{V_{dc}}{V^*_{max}} = \frac{400}{10} = 40 \tag{2.76}$$

(2) <u>Current Transducer Gain</u>

$$H_c = \frac{i^*_{max}}{i_{max}} = \frac{10}{15} = 0.667\ V/A \tag{2.77}$$

(3) Speed Transducer Gain and Delay

$$H_\omega = \frac{\omega_{r-max}}{\omega_{m-max}} = \frac{10}{261} = 0.0382 \tag{2.78}$$

$$T_\omega = 0.01 \text{ sec.} \tag{2.79}$$

(4) Motor Transfer Function

$$B_t = B + B_l = 0.001 + 0 = 0.001 \tag{2.80}$$

$$K_1 = \frac{B_t}{K_b^2 + RB_t} = 0.000126 \tag{2.81}$$

$$T_m = \frac{J}{B_t} = \frac{0.006}{0.001} = 6 \text{ s} \tag{2.82}$$

$$-\frac{1}{T_1}, -\frac{1}{T_2} = -\frac{1}{2}\left[\frac{B_t}{J} + \frac{R}{L}\right] \pm \sqrt{\frac{1}{4}\left(\frac{B_t}{J} + \frac{R}{L}\right)^2 - \frac{K_b^2 + RB_t}{JL}} \tag{2.83}$$

$T_1 = 0.0464$ s $\qquad T_2 = 0.000359$ s

(5) Current Controller Design

To design the current controller gain and time constant, (2.43) and (2.44) are used. For the prototype, a bandwidth of 1600 Hz and damping ratio of 0.707 is targeted for the current loop performance. Shown below are the current controller gain and time constant. Fig. 2.11 contains the step response of the current loop transfer function described in (2.31). The current loop is given the maximum step current command of 10V and the result is the maximum current of 15A. The current loop has a rise time of 0.14ms, 13% overshoot, and settling time of 0.5ms. The unapproximated step response is shown, which does not approximate the mechanical time constant, T_m.

$$K_c = \frac{1.414T_1T_2\omega_m - T_1 - T_2}{H_cK_rK_1T_m} = 9.42 \tag{2.84}$$

$$T_c = \frac{H_c K_c K_r K_1 T_m}{T_1 T_2 \omega_m^2 - 1} = 0.000113 \tag{2.85}$$

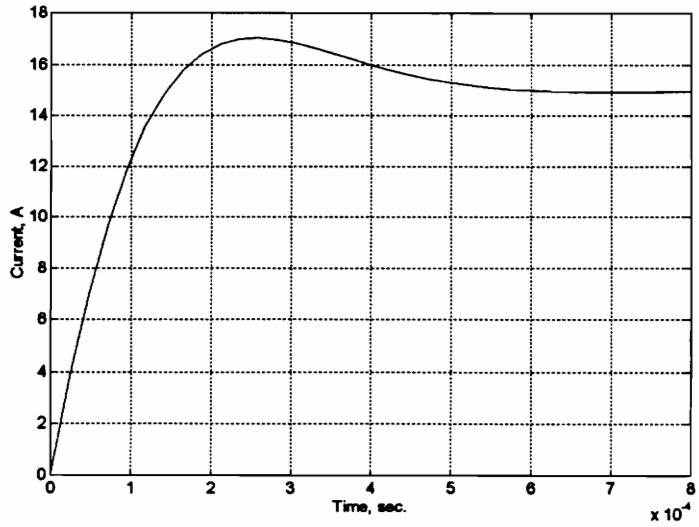


Fig. 2.11 Step response of current loop transfer function (2.31)

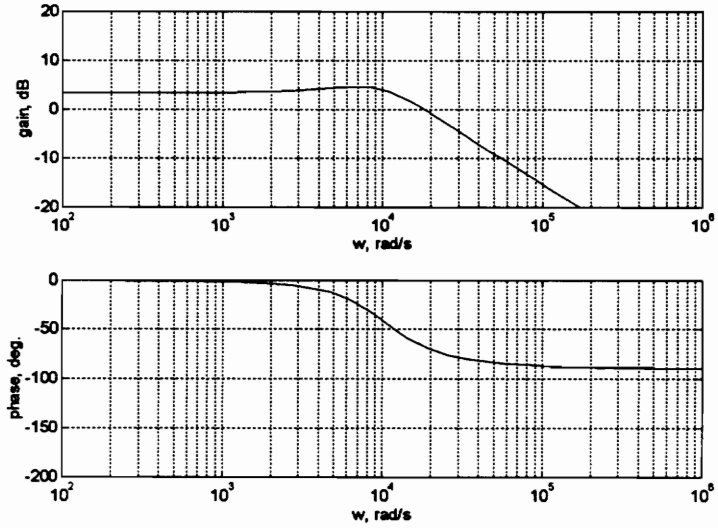


Fig. 2.12 Frequency response of the current loop transfer function (2.31)

The frequency response in Fig. 2.12 shows the designed control loop meets the bandwidth of 1600 Hz. With the exception of a slightly larger overshoot due to the unapproximated mechanical time constant, the design meets specifications.

(6) Speed Controller Design

To design the speed controller, equations (2.57) and (2.58) are used.

$$K_v = \frac{B_t((T_m + T_w)^2 - 2T_m T_w)}{2K_b T_m T_w H_w} = 2.79 \tag{2.86}$$

$$T_v = \frac{2H_w K_b K_v (T_m + T_w) B_t}{(B_t + H_w K_b K_v)^2} = 0.04\ \text{s} \tag{2.87}$$

Fig. 2.13 is the step response of the unapproximated speed loop transfer function described in (2.59). The unapproximated speed loop contains the current loop in the system transfer function.. The speed loop is given a maximum speed command of 10V and the result is the maximum speed of 261 rad/s.

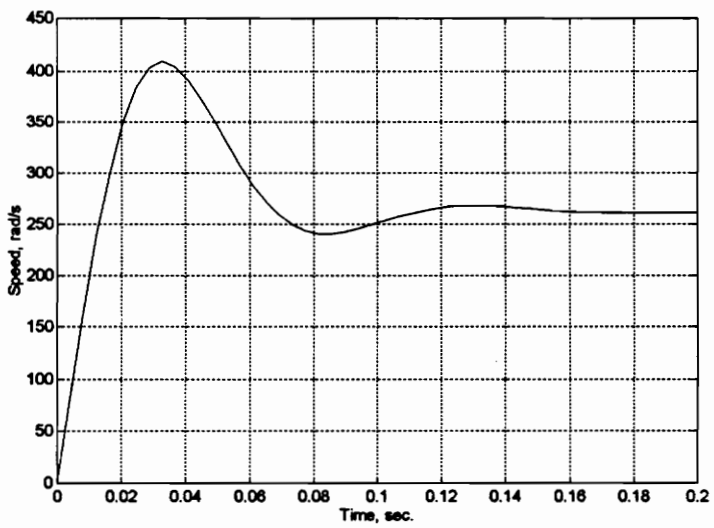


Fig. 2.13 Step response of speed loop transfer function (2.59)

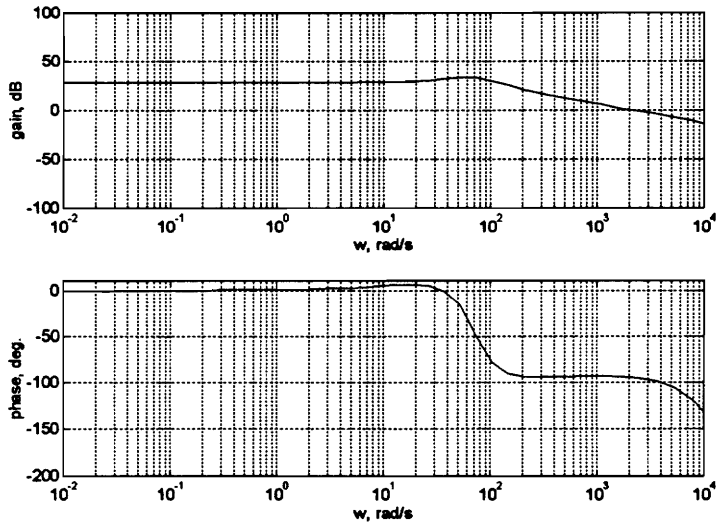


Fig. 2.14 Frequency response of speed loop transfer function (2.59)

The step response has a rise time of 0.015s, 57% overshoot, and a 0.11s settling time. Fig. 2.14 is the frequency response of the speed drive, showing a bandwidth of approximately 15.9 Hz.

To reduce the overshoot for the step response, a soft start is placed between the reference command and the SRM drive system. The time constant of the soft start is the same magnitude as the speed controller time constant. The following is the transfer function of the soft start:

$$S(s) = \frac{1}{1 + 0.04s} \tag{2.88}$$

Shown in Fig. 2.15 is the step response of speed loop transfer function with a soft start. The same conditions apply in this simulation as the other step speed response. Even though the rise time is increased to 0.055s and the settling time to 0.14s, the overshoot is eliminated. The bandwidth of the speed loop, shown in Fig. 2.16, has been reduced from 15.9Hz to 6.37Hz. For many industrial applications, the response time is not as important

as the speed overshoot of the drive. Therefore, a soft start is necessary part of an SRM speed drive.

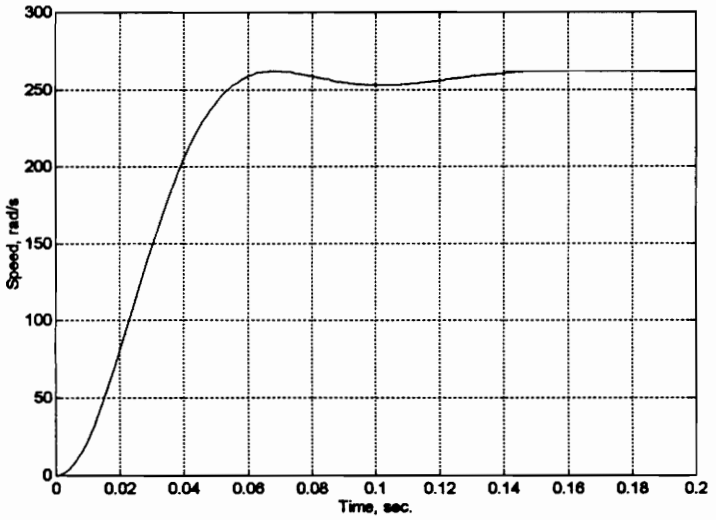


Fig. 2.15 Step response of speed loop transfer function with soft start

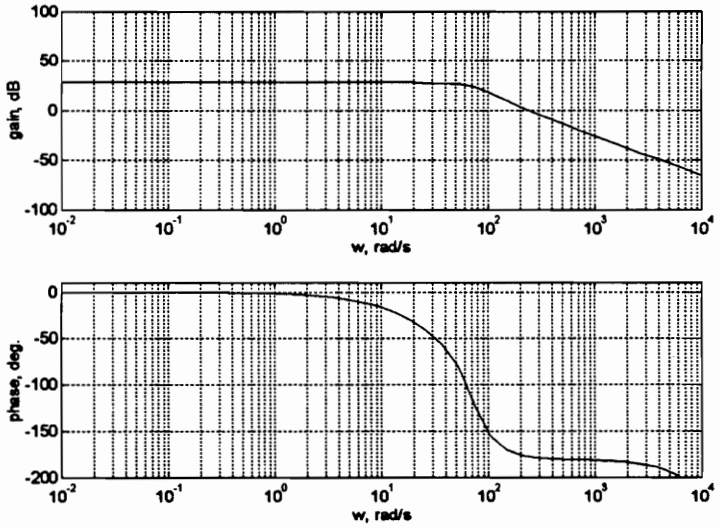


Fig. 2.16 Frequency response of speed loop with soft start

(7) Calculation of Advance and Fall Angles

Using (2.72) and (2.73), the advance and fall angles are calculated. For calculation of advance angles the following parameters are used:

DC Link Voltage:	400V
Back EMF constant:	0.96
Phase Resistance:	0.931Ω
Initial Current:	0A
Unaligned Inductance:	3.4mH

Table 2.1 Advance and fall angles for the prototype 5hp SRM

rpm	Advance Angle	Fall Angle		rpm	Advance Angle	Fall Angle
100	0.08	0.69		1400	1.70	7.75
200	0.16	1.35		1500	1.90	8.18
300	0.25	2.00		1600	2.11	8.60
400	0.35	2.61		1700	2.34	9.00
500	0.45	3.21		1800	2.60	9.39
600	0.55	3.78		1900	2.88	9.78
700	0.66	4.34		2000	3.19	10.15
800	0.78	4.87		2100	3.53	10.51
900	0.91	5.39		2200	3.92	10.86
1000	1.05	5.89		2300	4.35	11.20
1100	1.19	6.38		2400	4.84	11.53
1200	1.35	6.85		2500	5.40	11.86
1300	1.52	7.31				

Since the inductance changes at the rising slope, the back EMF is taken into consideration. The unaligned inductance is chosen as a function of the maximum current of 15A. For calculation of the fall angles, the aligned inductance is 31.8mH and the initial current is 15A. The back EMF constant is 0.75, since there is some change in inductance with respect to rotor position. Table 2.1 contains the advance and fall angles for the prototype 5hp SRM.

(8) Simulation of Current Loop

Shown in Fig. 2.17 is the step response of the current loop at rated speed of 2500rpm and rated current of 12A. All values are given in normalized units. The first plot shows phase A current, i_a , along with an idealized inductance profile for reference. The second plot shows phase B current, i_b , the third plot is the torque generated, T_e , and the fourth plot is the voltage applied, V_{ap}. The rise time for the current is 0.045ms which is much faster than the 0.14ms rise time in the step response shown in Fig. 2.17.

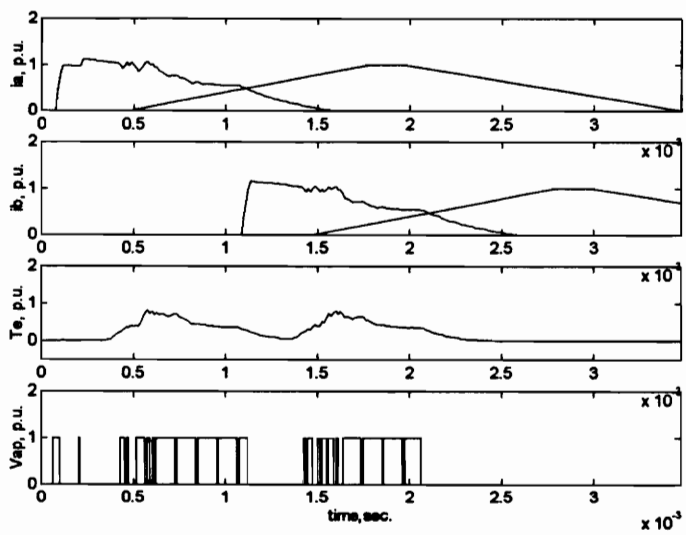


Fig. 2.17 Current loop simulation at rated speed and current

This is due to the smaller inductance at the beginning of the inductance slope, whereas the mean value of inductance is used in the current loop step response.

The simulation uses an advance angle of 6 degrees and a fall angle of 12 degrees. In the simulation, the current is able to rise to the commanded value before the rising edge of the inductance slope, and it falls before the negative inductance slope is reached.

In the simulation, the current is not able to stay at commanded value of 12A when the inductance begins to rise sharply, even with a maximum PWM cycle of 90%. To combat the problem, the DC link voltage is increased from 400VDC to 621VDC, which is the average DC voltage value for a 3 phase supply of 460VAC. Fig. 2.18 illustrates the increase in DC link voltage and current at 12A for the duration of the phase excitation.

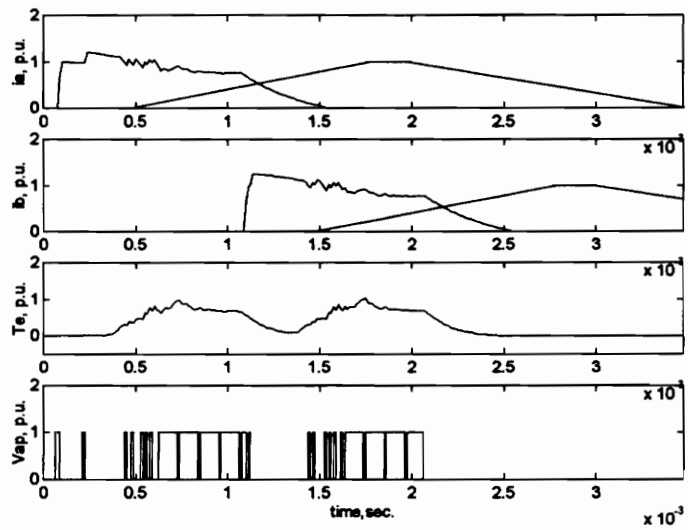


Fig. 2.18 Current Loop Simulation at Rated Speed and Current and Vdc = 621V

(9) Simulation of the Speed Loop

Fig. 2.19 shows the dynamic simulation of the speed loop in normalized units. Shown in the first plot is the speed command at 2500rpm and the speed response of the system, ω_m. The second, third and fourth plots show the speed error, ω_{me} , current

command, i^*, and torque, T_e, respectively. The simulation is carried out without a soft start and no external load. The rise time for the simulation is 0.09s which is much slower than the rise time of 0.015s in the predicted step response shown in Fig. 2.13. Due to the limitation of the torque command, the nonlinear dynamic simulation has a slower response time compared to the step response from transfer function (2.59).

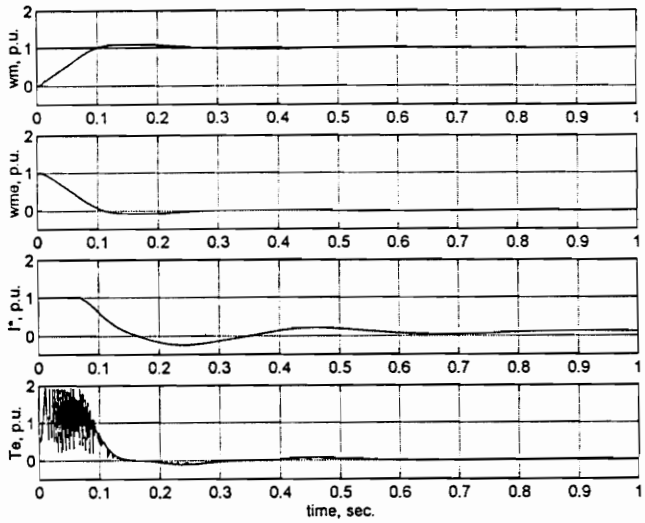


Fig. 2.19 Speed loop simulation for step speed command

(10) Simulation of Load Disturbance Rejection

An important performance feature of industrial drives is their insensitivity to load changes. The load is modeled as a disturbance to the system and subject to change, which affects the speed of the drive system. By using the symmetric optimum method in the speed controller design, the speed controller exhibits good disturbance rejection.

Fig. 2.20 illustrates a step load change from 0p.u. to 0.8p.u.at 0.5s. The load is a constant load. The drive is simulated at 2500rpm. At zero time, the simulation parameters are being adjusted, and at 0.5s, the simulation illustrates how the drive is able to adjust to the load change and recover to the speed command within 0.1s.

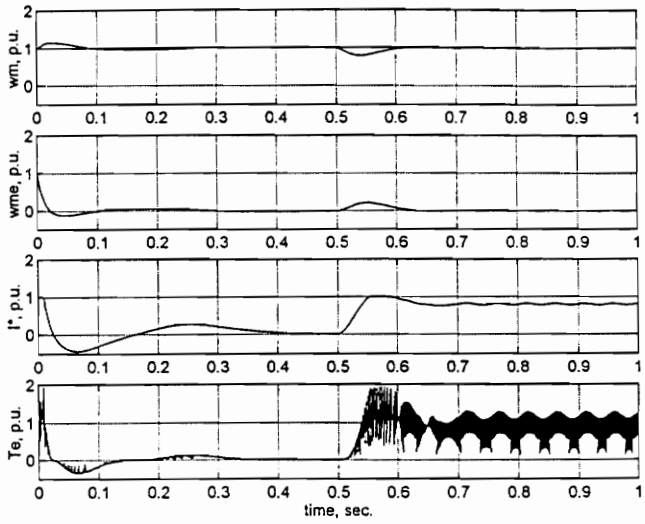


Fig. 2.20 Speed response to a constant step load change

2.9 Selection of Nominal Operating Points

In the variable speed drive design process, it is often desirable to adjust the sensitivity of the controller to the speed error signal.

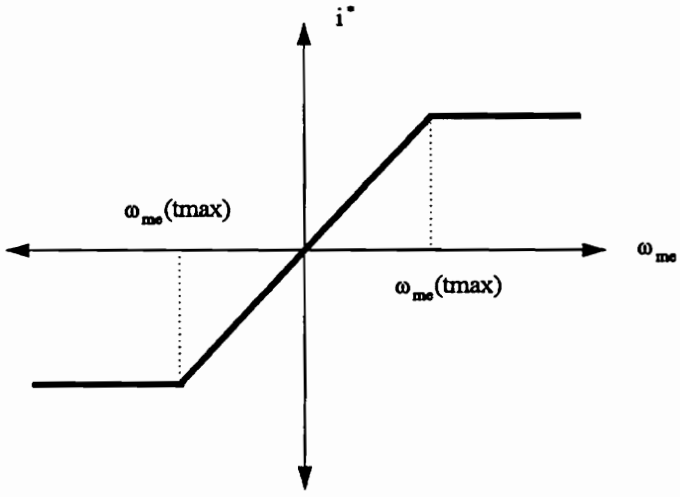


Fig. 2.21 Current command versus speed error

The speed response is directly related to the current command generated by the PI speed controller. At a specified speed error, ω_{er}(tmax), the maximum current command should be generated, as illustrated in Fig. 2.21. It is an iterative design process to adjust the PI gains, which in turn, changes the speed error at which the maximum current command is generated.

Selection of nominal operating points affect the PI gains for the current and speed controllers. For example, Fig. 2.22 illustrates the torque generated in the actual SRM and the assumed linear models.

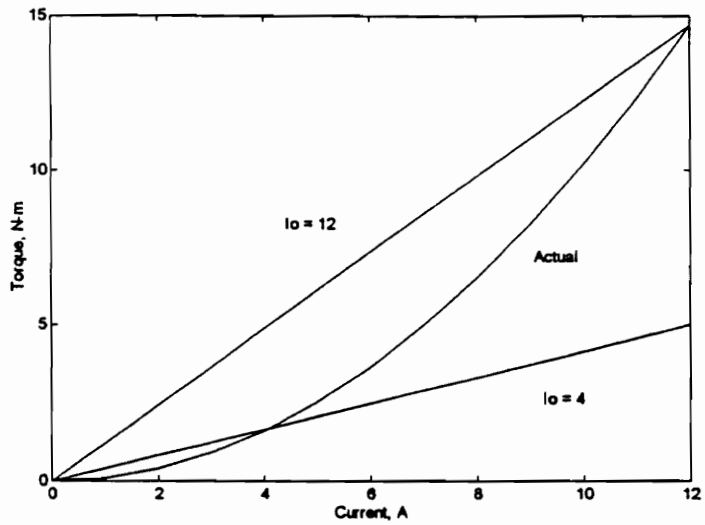


Fig. 2.22 Selection of nominal current

The actual torque is proportional to the current squared, whereas the linear models make the torque proportional to the current. If the nominal current is chosen as 12A, then for the range of currents below 12A, less torque is produced than expected in the linear model. However, if the nominal current is chosen as 4A, then for current below 4A, less-than-expected torque is produced. But, for current larger than 4A, more torque is produced than expected in the linear model.

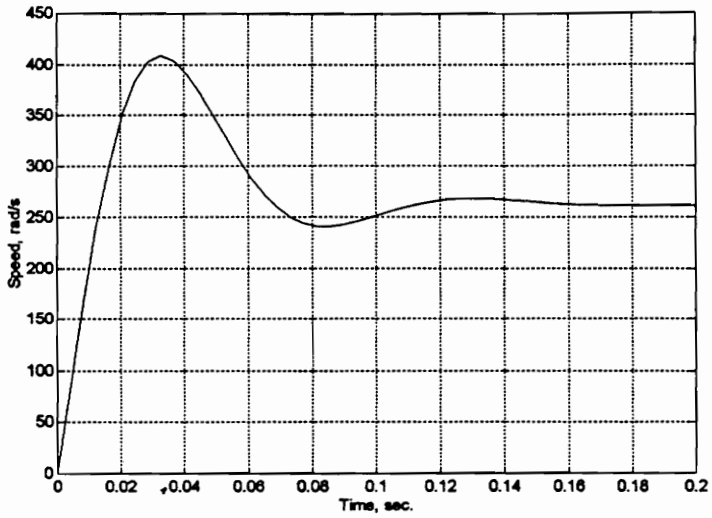


Fig. 2.23 Step response of speed loop transfer function with nominal current = 4A

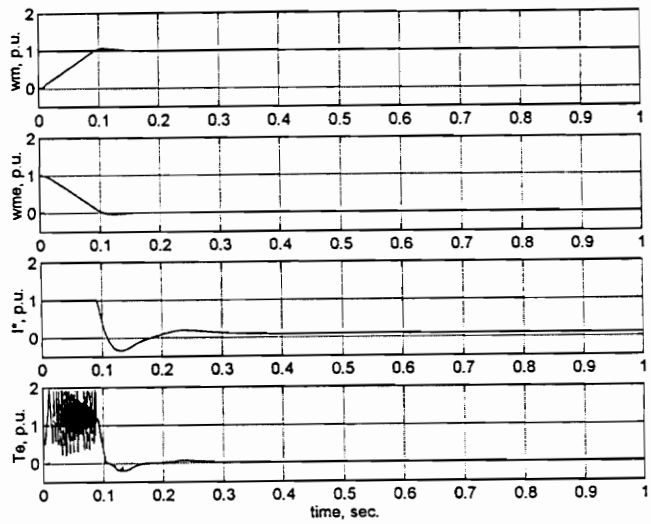


Fig. 2.24 Simulation of speed loop with nominal current = 4A

Fig. 2.23 shows the step response of the speed loop transfer function (2.59) with a nominal current of 4A. Fig. 2.24 is a nonlinear dynamic simulation with the same

conditions. The dynamic simulation shows a greater sensitivity to the speed error compared to the simulation in Fig. 2.19.

By selecting the nominal current of 4A, assumed back EMF/torque constant is decreased, increasing the sensitivity of the speed controller. A decrease in the back EMF/torque constant increases the speed controller proportional gain which is seen in (2.57). The following table shows the resulting changes in current and speed controller gains and time constants due to changes in inductance, nominal speed and current. Notice that by changing the inductance and nominal speed, the speed controller gain and time constant are not changed. Also, notice that by changing the nominal current, the current controller gain, current controller time constant, and the speed controller time constant are not changed. Since the response of the current controller is satisfactory, there is no need to adjust the nominal speed or inductance in the linearized model.

Table 2.2 Variation of gains with respect to nominal speed and current

L (H)	ω_{mo} (rad/s)	I_o (A)	K_v	T_v (s)	K_c	T_c (s)
0.0221	261	12	2.79	0.04	9.42	0.000113
0.0318	261	12	2.79	0.04	14.59	0.000121
0.0221	261	4	8.39	0.04	9.44	0.0000113
0.0221	131	12	2.79	0.04	10.60	0.000126

Chapter 3: Torque Smoothing

3.1 Introduction

Torque smoothing is a technique used to compensate for the nonlinearity in the SRM. Because the torque is dependent upon the current and rotor position, for a constant phase current input, a non-constant torque is created. This chapter describes the torque smoothing process, the approach taken in the prototype SRM drive and illustrations of torque smoothing through simulation.

3.2 Torque Smoothing Process

Fig. 3.1 illustrates the torque produced for 5A of constant current. Zero degree is the fully aligned position and 30 degrees is the fully unaligned position.

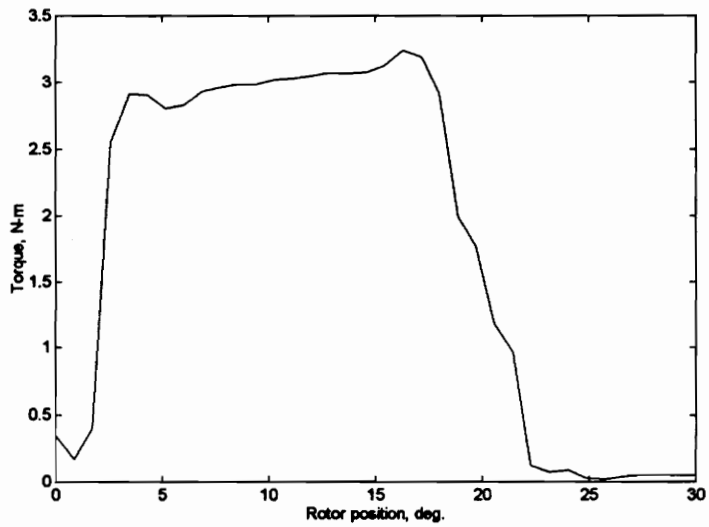


Fig. 3.1 Torque as a function of rotor position and current = 5A

If the motor is operating at a constant current command, the torque produced is not constant, causing vibration and higher acoustic noise[15]. Also, at very low speeds, smooth rotation is difficult to obtain because of the pulsating nature of the torque.

To achieve linear-type operation from a nonlinear plant, a nonlinear actuation command is needed. In SRM controllers, both the speed and current controllers are designed based on linear models, providing linear actuation signals. Torque smoothing provides a nonlinear mapping between the current command and the current loop. For a constant current command, torque smoothing adjusts the current command so a constant torque is produced. Fig. 3.2 illustrates torque smoothing in the SRM driver system.

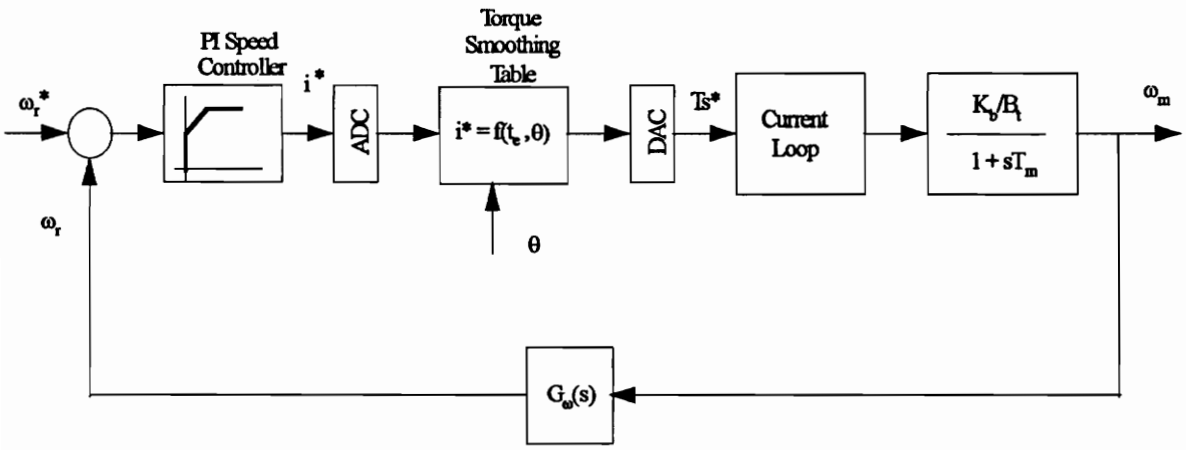


Fig. 3.2 Block diagram of SRM drive system with torque smoothing

Work reported in literature seems to use a common torque smoothing technique[16]. The torque smoothing map is created by obtaining torque data using static torque measurements, finite element analysis of the motor, or a CAD program. Once the relationship between the torque, current and rotor position is established, a table is created and stored in non-volatile memory, such as an EPROM. During machine operation, the current command, i^*, and the rotor position, θ, are used as an index into the table to get

the nonlinear current command, T_s^*. This nonlinear current command is then input into the current loop.

3.3 Torque Smoothing in the 5hp SRM

In the 5hp SRM prototype, the first step is to create the torque smoothing table. The torque smoothing table is created using the torque data from SRMCAD[17]. SRMCAD provides the torque in relation to rotor position and current. Using the torque and rotor position, an interpolation procedure obtains the appropriate current. The current values are then stored in an EPROM.

When the current command is output from the PI speed controller, it is digitized by an 8-bit analog/digital converter. The digitized current command and rotor position from the microcontroller are used to address the EPROM. The EPROM provides the nonlinear current command, which is converted back to an analog signal by a digital/analog converter.

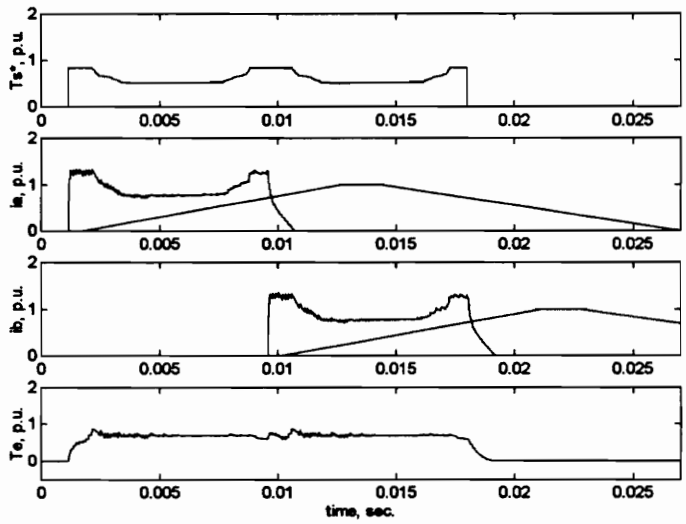


Fig. 3.3 Simulation of torque smoothing

This nonlinear current command is then applied to the current controller.

Although the system now has a digital nature, the sampling time of the A/D and D/A system is 100μs, which corresponds to a sampling frequency of 10kHz. The sampling frequency of the torque smoothing hardware is much greater than the bandwidth of the speed loop(approx. 15.9 Hz) so the digital aspects do not come into play.

Fig. 3.3 illustrates torque smoothing for a speed of 300rpm, a torque command of 10N-m, and a DC link voltage of 400V. The maximum nonlinear current command, Ts^*, is 10V, which is equivalent to commanding 15A. Shown in the first plot is the torque smoothing command to provide a constant torque at 10N-m. The second and third plots illustrate the response of the phase currents, ia and ib, to the torque smoothing command. The fourth plot is the resulting torque, T_e.

3.4 Advantages and Limitations of Torque Smoothing

The simulation, torque smoothing is able to keep the torque at 10N-m for the majority of the time. A comparison made between the torque in Fig. 3.3 and Fig. 2.17 illustrates the advantage of torque smoothing. The smooth torque will reduce the vibration and acoustic noise, which is a major problem with the SRM. Furthermore, the proposed torque smoothing technique is simple in that it consist of a few inexpensive, off-the-self components. Provided the correct mapping and rotor position is provided, torque smoothing enhances the drive's performance.

Torque smoothing has some limitations. It is not able to keep torque smooth at the switching off of one phase and the switching on of another. This is due to the torque created by the recovery current, which is not taken into account in the torque smoothing system. Also, torque in the simulation is not constant near the end of the phase excitation because the amount of current needed to keep the torque constant exceeds the maximum amount of current allowed in the motor.

Chapter 4: Controller Architecture

The controller architecture plays a key part in achieving a low cost SRM drive while maintaining satisfactory performance. The block diagram of a SRM controller architecture is shown in Fig. 4.1. There are two main parts to the controller architecture; the hardware and software. The following hardware sections discuss the hardware necessary for speed, current and phase firing control. They also present the types of current, speed and position sensing available for SRM controllers. The software sections discuss the microcontroller programming and the software involved in torque smoothing.

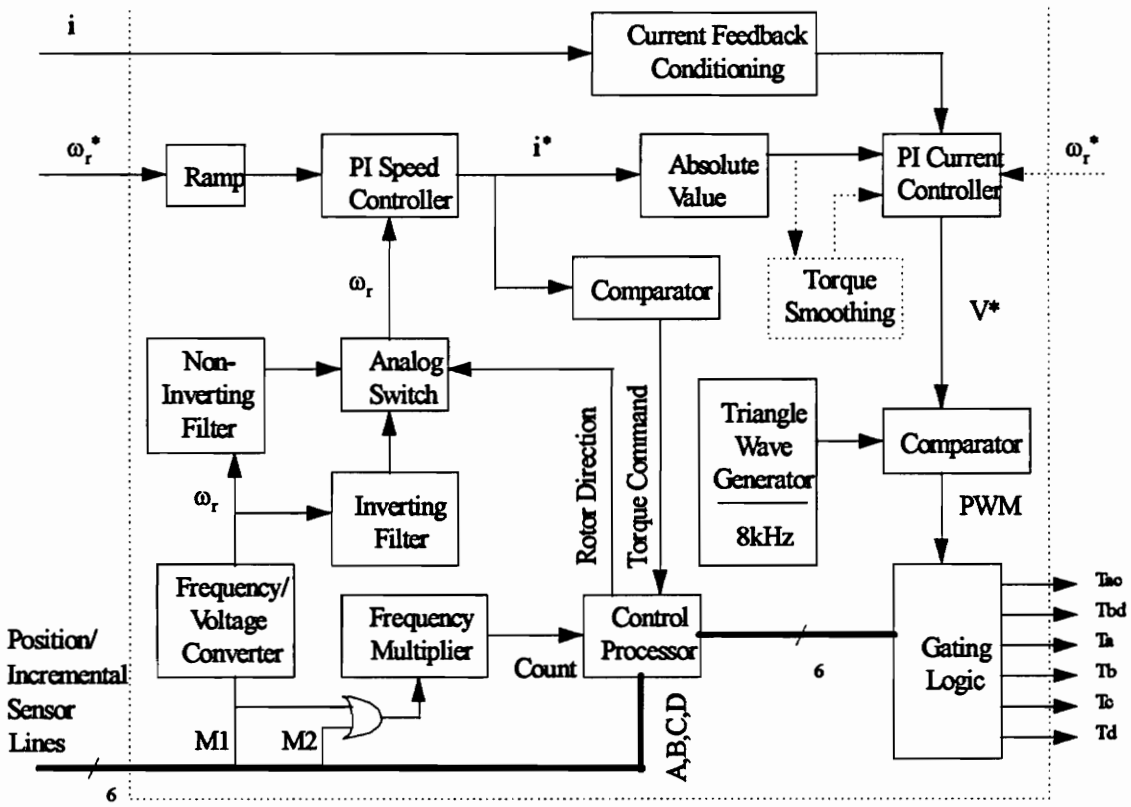


Fig. 4.1 SRM controller hardware architecture

4.1 Hardware Layout

There are three types of control functions carried out by the controller. The first two control functions are current and speed control. These two control functions are implemented in analog hardware because of its simplicity, cost and compactness. The third control function, phase firing control, is implemented in digital hardware because of its discrete nature. This type of mixed analog and digital hardware is a hybrid architecture. For the hybrid architectures to operate correctly, a well-defined interface between the analog and digital hardware is needed. The following hardware layout sections discuss the current sensing, current control loop, speed/position sensing, the speed control loop and the phase firing control.

4.1.1 Current Sensing

A large portion of the controller cost arises from the sensing equipment. Presently, current sensing for SRMs is carried out by placing a current sensor on each phase. Current sensors are normally hall effect sensors. For the 8/6 prototype SRM, four current sensors are required, and each current sensor may cost up to $20(US). For low cost, low performance SRM drives, this cost is unacceptable. To reduce the cost of current sensing, the number of current sensors must be reduced. This work introduces a unique, one-current sensor topology for SRM drives. On the next page, Fig. 4.2 shows the converter topology used in the experimental setup with the one sensor, *S1*, in place.

Each phase of the converter has three operating stages. For example, if *T5* and *T1* are on, current flows from the positive supply and goes through *T5*, *Ph A*, *T1* and *S1*, and returns to the negative supply. Once the current reaches commanded value, *T5* is off, and

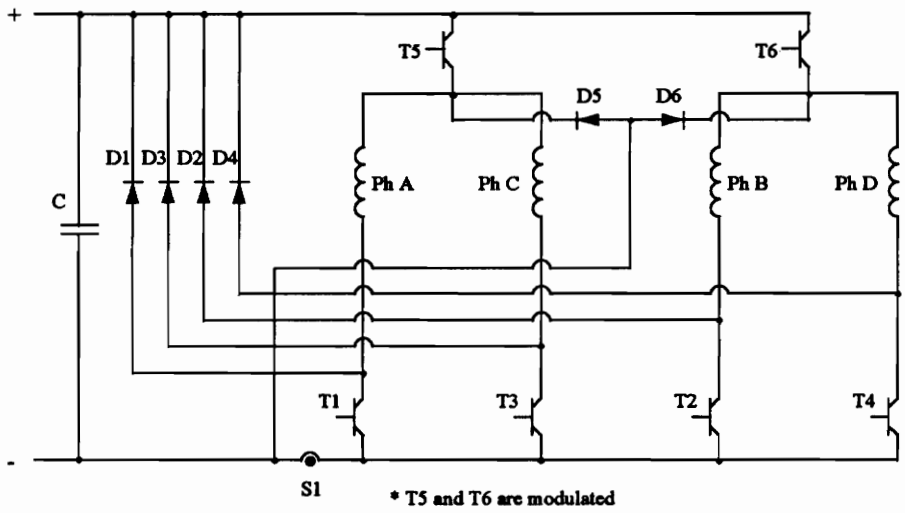


Fig. 4.2 Converter with one current sensor

the current circulates through *Ph A, T1, S1, D5* and back to *Ph A*. To de-energize *Ph A*, *T5* and *T1* are off, and the current circulates through *Ph A, D1, C, D5* and back to *Ph A*. A similar operation occurs with the other phases. Fig. 4.3 illustrates the current sensed by *S1*, where i^* is the current command and i is the current in the machine phase.

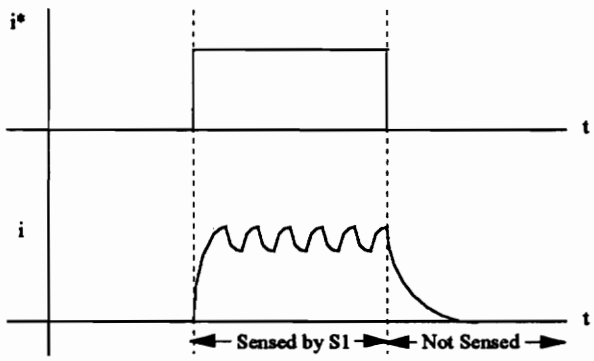


Fig. 4.3 Current sensed in one phase

Notice that the sensor only detects current while a current command is given and it cannot detect the recovery current once the phase is switched off. Normally, the recovery current is not sensed for current control in SRMs because, at the time one phase switches off, another is switching on, and current control switches to the new phase. Therefore, the new, one-current sensor topology senses the same current in the SRM as the multiple current sensor topology, at a lower cost. Because only one phase current can be sensed at a time, simultaneous conduction in two phases is not permitted. At the present, this topology has only been tested for the full-bridge converter topology but may be extended to other converter topologies.

4.1.2 Current Control Loop

The use of a current control loop has several benefits for a low cost, low performance SRM drive. First, if the drive system is used as a torque drive, the user command is applied to the current loop to achieve the desired torque. Second, a current loop provides overcurrent protection because the current in the phases is sensed and limited to a maximum value.

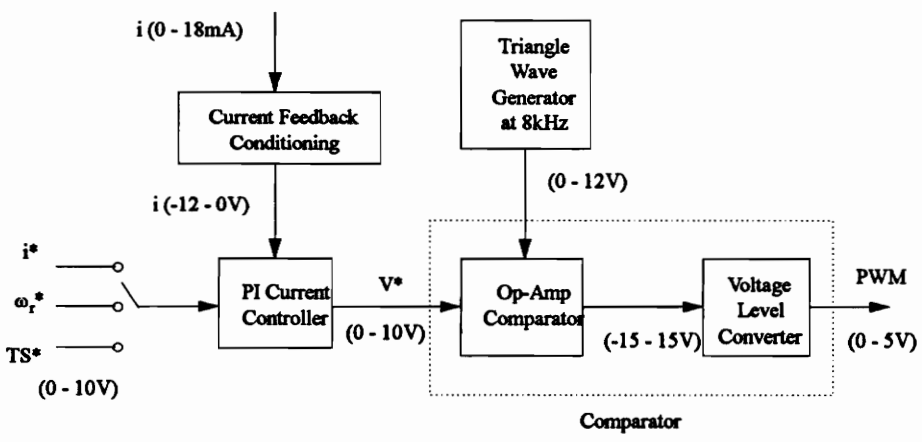


Fig. 4.4 Current control loop

Third, for advance control, such as torque smoothing, it is necessary to have a current control loop which follows the nonlinear current command, T_s^*. The current control loop structure for the experimental setup is shown in Fig. 4.4.

Another benefit of using the one-current sensor topology is simplification of the current control loop. With one current sensor, no analog switches or multiplexors are necessary to switch the current feedback. The current feedback from the sensor is a current signal which ranges from 0 - 18mA. The current feedback is fed to a current feedback conditioning circuit, which makes a one-to-one thousand current-to-voltage conversion using a resistor. The current feedback conditioner also inversely amplifies the feedback. The feedback gain for the experimental setup is 0.667 and the feedback ranges from -12V to 0V.

The current command may come from three sources. The first source is from the speed controller when the drive functions as a speed-controlled drive. The second source is directly from the speed command, ω_r^* , which enables the drive to function as a torque drive. The third source is from the torque-smoothing hardware, T_S^*. Since the current command is always positive, a noninverting PI controller is chosen as the controller. The integrator helps smooth the voltage command and reduce multiple switching which may damage the converter.

Once the voltage command, V^* , is obtained, it is compared with a positive triangle wave at 8kHz. The op-amp comparator's output is a bipolar -15V or 15V. To convert the PWM signal to a discrete signal level, a voltage divider circuit reduces the level of the voltage, and a diode is used to truncate the negative voltage.

The current control loop presented is typical of analog current control loops used in other drives. The switching frequency is adjusted by varying the frequency of the triangle wave generator. Because the current feedback and current command are unipolar, a unipolar switching scheme is chosen. In a unipolar switching scheme, the triangle wave is always a positive or negative voltage. A hysteresis current control loop may also be

used[18], but it was not chosen in the experimental setup because of switching frequency variations.

4.1.3 Speed and Position Feedback

For speed control of an SRM, both the speed and position of the rotor are needed. The position information is necessary to fire the phases of the SRM at the appropriate moments to achieve the desired torque. There are several ways to detect the position of the rotor. One method is using a resolver, which gives excellent position information, but is too costly for low cost, low performance SRM drives. The most common position sensing for SRMs are absolute position encoders. However, these devices are costly and a cheaper alternative needs to be available for low cost, low performance SRM drives.

The speed feedback sensing for drives is either done by a tachogenerator or an incremental encoder. Tachogenerators are the easiest to use in control because they return a bipolar voltage signal indicating the direction the rotor is turning. If a quadrature incremental encoder is used, the direction is detectable, as well as, the speed. If an incremental encoder is in quadrature, the two pulse tracks are 90 degrees out of phase, and the direction is detected by determining which track is leading. A tachogenerator has a slight advantage over the incremental encoder because the pulses of the incremental encoder must be processed to determine the speed.

In the experimental prototype, a new incremental-plus-absolute encoder [developed by R. Krishnan and S. Lee] is used. The relation of the encoder signals to the phases is shown in Appendix C. In this type of encoder, there are two incremental tracks, but they are not in quadrature. There are also four absolute tracks which give phase position information. The purpose of the encoder is to reduce the cost of speed and position sensing and it is being tested with the low cost, low performance SRM drive for feasibility.

4.1.4 Speed Control Loop

To achieve good speed performance, both speed and current control loops are required. The advantages of analog control is its simplicity in design and construction, low cost, and compactness. The disadvantages of analog control compared to digital control is its susceptibility to drift and noise. In low cost, low performance drives, accurate speed performance is not as important as the cost, simplicity, and compactness. Therefore, for the prototype system, both the current and speed control loops are implemented in analog hardware.

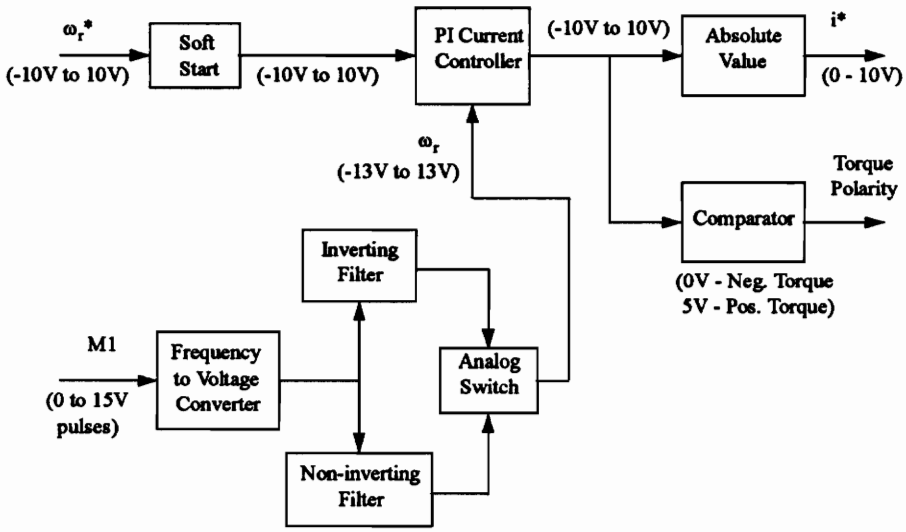


Fig. 4.5 Speed control loop

Fig. 4.5 illustrates the speed controller and its associated hardware. The speed command, ω_r*, is the industrial standard +/-10V signal. Often speed controllers filter the speed command in order to decrease the speed overshoot and acceleration. The smoothing is called a "soft start."

The speed feedback is obtained by processing one pulse train of the incremental encoder tracks, M1, through a Frequency-to-Voltage(F/V) converter chip. The F/V

converter is the simplest and least expensive of all speed feedback processing options. Although the F/V converter has a slow time constant(approx. 10ms in the prototype), it possesses good linearity over the entire speed range of the prototype SRM drive. Since the F/V converter output is unipolar, the output must be amplified and filtered by both an inverting and noninverting amplifier to receive both speed feedback polarities.

An analog switch is used to switch the appropriate speed feedback to the speed controller. The switch is controlled by the microcontroller which determines the correct polarity for the speed feedback. Normally, if the incremental encoder is in quadrature, the incremental portion of the encoder is used to determine the speed feedback selection. In the experimental prototype, the microcontroller uses the information given by the absolute portion of the encoder to sense any changes in direction. When rotor direction changes, the speed feedback selection is then changed by the microcontroller.

An inverting PI speed controller is chosen as the controller circuit for two reasons. First, the inverting circuit allows gains of less than 1. And secondly, the integrating portion of the control smoothes the torque command and eliminates speed error.

The output of the speed controller is the torque command, and at this point, the torque command polarity is detected. The torque command polarity is required by the microcontroller to control the phase firing. To determine the torque command polarity, an op-amp comparator circuit is used to give a bipolar output signal. The output of the op-amp comparator is -15V or 15V. A voltage divider, diode, and a buffer are used to convert the output signal voltage levels to digital levels. Another option to determine torque polarity is an A/D circuit, but this requires much more complexity and cost compared to the comparator circuit.

Because the feedback from the current sensor is unipolar, the current command should also be unipolar. Therefore, an absolute value circuit converts the torque command from a bipolar signal to a unipolar, positive signal.

4.1.5 Microcontroller

At the heart of the SRM controller architecture is an Intel 8748H microcontroller. The 8748H is an 8-bit microcontroller which operates at 740kHz. The 8748H is a suitable processor for the low cost, low performance SRM controller because of its ruggedness, cost, and capabilities. Details on the microcontroller are listed in Appendix D. Shown in Fig. 4.6 is the microcontroller, its supporting hardware, and the I/O signals.

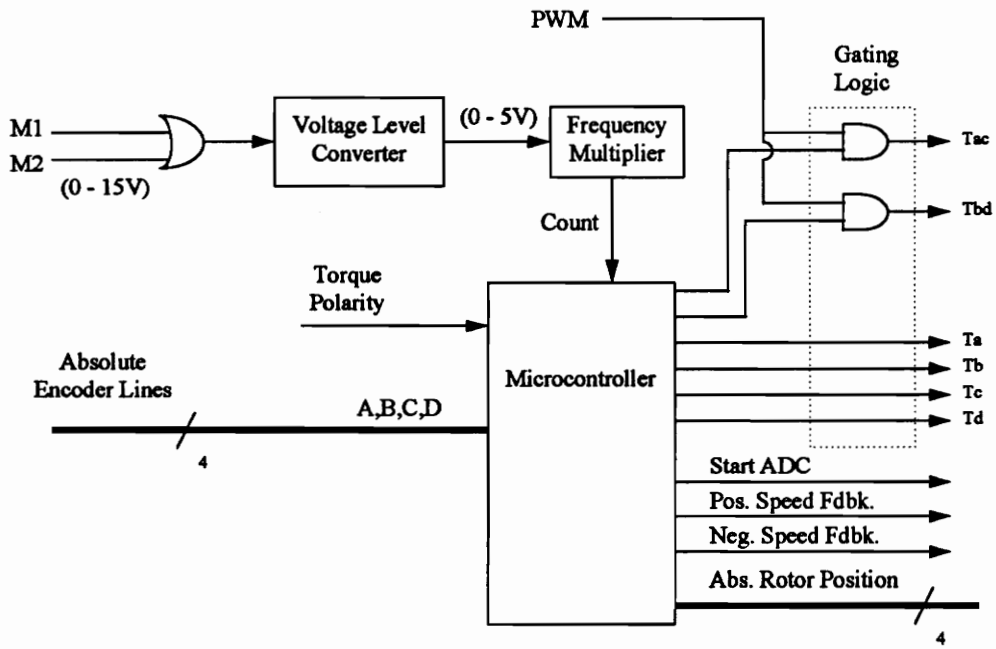


Fig. 4.6 Microcontroller and supporting hardware

The microcontroller performs the following functions listed below:

- Provide gate drive control signals
- Advance and fall angle control
- Provide the absolute rotor angle for torque smoothing
- Speed feedback selection

As mentioned earlier, the microcontroller determines the speed feedback selection by monitoring the absolute encoder signals. By recording the previous encoder signal, the microcontroller is able to determine in the next sample whether the rotor is continuing in the same direction, or if direction has changed.

The main function of the microcontroller is to provide the appropriate gate signals depending on which quadrant the drive is operating. Information received from the absolute portion of the encoder and *Count* determine which phase to fire. For the prototype SRM drive, the two incremental signals are combined and the frequency is multiplied by two to achieve a higher rotor position resolution(240 pulses per revolution).

Advance and fall angle control determines the rotor positions at which each phase is turn on and off. The advance and fall angle is determined by the speed of the motor and this calculation is done on the microcontroller. Calculation of advance and conduction angle control is discussed in more detail in section 4.2.2.2.

In order to carry out torque smoothing, the absolute rotor position is required. Once the new rotor position is determined, the microcontroller outputs it through the BUS port. To perform torque smoothing, the A/D converter needs a starting signal, which is the purpose of *Start ADC*. This signal stays high as long as power is supplied to the controller.

The microcontroller is one option for a low cost, low performance SRM controller. For the prototype system, the microcontroller is necessary to carry out the functions listed above. Provided that the speed feedback selection is done in discrete hardware and torque smoothing is not performed, it is possible to create an encoder which provides signals which are directly applied as the gate firing signals. In this case, the functions carried out by the microcontroller are carried out by discrete hardware. For this option, the hardware is simple, operation is fast and the software is eliminated. However, this option does not provide the flexibility which the microcontroller provides. Also, since

the 8748H is priced so low, there is little or no cost advantage in choosing the discrete hardware implementation for low speed applications.

4.1.6 Torque Smoothing

Torque smoothing is a method which helps decrease the amount of pulsating torque in the SRM. The cause of the pulsating torque and how torque smoothing is used to decrease its effect is discussed in detail in Chapter 3. This section describes the hardware to carry out torque smoothing.

There have been several proposed hardware circuits and architectures to perform torque smoothing[19],[20]. Most architectures use a high-end microcontroller, micro-

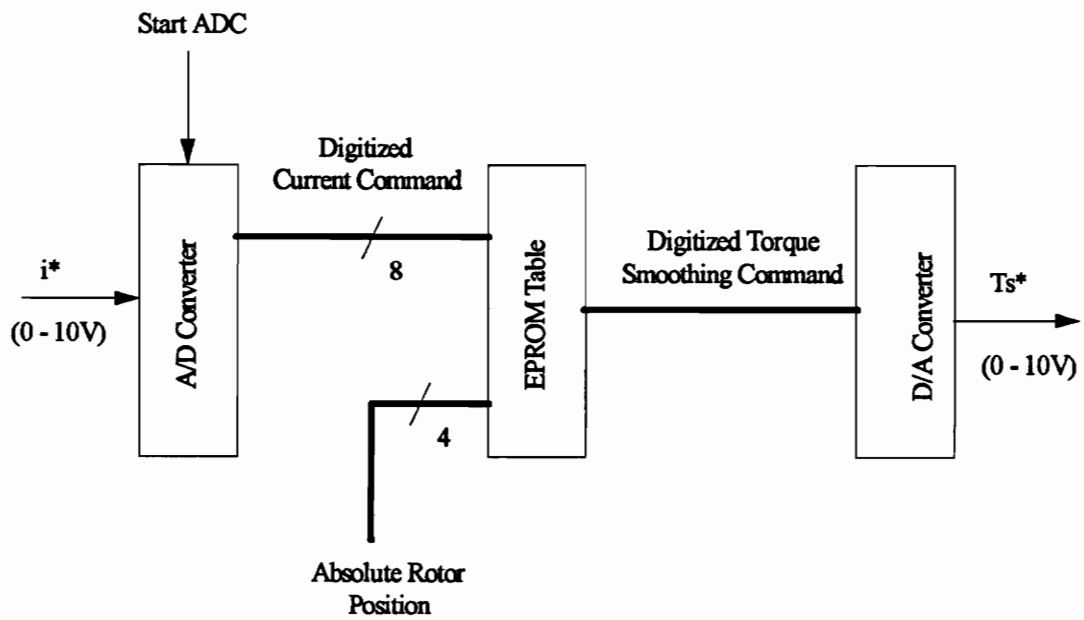


Fig. 4.7 Torque smoothing layout

processor, or DSP chip which access a table memory in order to create the torque smoothing map. These architectures are targeted for high performance applications and

the controllers are costly. Proposed in this work is a simple architecture which provides the same capabilities at a reduced cost. All that is needed for this torque-smoothing scheme is an A/D converter, EPROM, and D/A converter. The torque smoothing may be enabled or disabled by selecting jumpers.

Fig. 4.7 illustrates the hardware necessary for torque smoothing. The purpose of torque smoothing is to provide a nonlinear mapping between the current command and the current controller. First, the current command, i^*, is digitized by a free-running A/D converter. The mapping is carried out by an EPROM, where the address for the EPROM is composed of the digitized current command and the absolute rotor position from the microcontroller. The output of the EPROM is a new current command, T_s^*, which is converted back to a positive analog signal by a free-running D/A converter.

4.2 Software Layout

The use of a microcontroller in electric motor drives provides flexibility and increased performance. The amount of flexibility and performance depends upon the capabilities of the microcontroller and the algorithms used in programming. The software layout section looks at the capabilities of the Intel 8748H and the algorithms used to program the experimental prototype. Other low cost, low performance SRM drives may use different microcontrollers and algorithms, but the design considerations remain the same. It is the goal of the following sections to present the work performed and list the important design concepts achieved.

4.2.1 Microcontroller Capabilities

There is a wide selection of microcontrollers which may be chosen for the low cost, low performance SRM drive system. Selection of a microcontroller depends upon cost, performance, and complexity. The best microcontroller choice is the one with the

lowest cost that is able to meet the performance specifications. In selecting a microcontroller, the following aspects should be taken into consideration:

- Is the processor fast enough?
- Does it have enough memory?
- Does it have sufficient peripheral devices such as timers, ports, etc.?
- Is the instruction set sufficient?

For the prototype SRM drive system, the Intel 8748H is chosen as the microcontroller. The 8748H is one of the slowest 8-bit microcontrollers, which eliminates it from being used in drives running at medium to high speeds. The maximum speed of the prototype drive is 2500 rpm, and the processor's instruction cycle is 1.36μs long. The longest execution path in the software takes 90μs for completion. If the drive is running at maximum speed, and for each revolution there are 240 count pulses coming to the microcontroller, the microcontroller must complete each execution of the control algorithm in 100μs. Therefore, the 8748H is able to meet the speed specification for the prototype drive.

The 8748H comes with 1K of EPROM which limits the size of the program. Of the 1024 bytes of EPROM, 463 bytes are taken up in four different tables. These tables are especially useful in SRM speed drives because the absolute encoder signal, fall angles, torque polarity, and rotor direction are used to index information, such as phase firing signals. Since the size of the program is limited by the maximum execution time of the control algorithm, the small amount of memory is not a factor for the prototype drive.

Peripheral devices are very important when choosing a microcontroller. All three of the 8-bit I/O ports on the 8748H are used to input the absolute encoder signals and output the absolute rotor position and gate control signals. There is also an on-board hardware counter which keeps count of the pulses coming from the incremental encoder. The hardware counter frees the rest of the microcontroller to perform other functions and

not have the concern of losing an incoming pulse. Other helpful features include single input pins testable by special jump instructions.

Although the instruction set of the 8748H only consists of 96 instructions, the instructions where found to be sufficient for the prototype SRM drive. There are special instructions to handle look-up tables, as well as controlling the I/O ports. The instructions for the 8748H only take 1 or 2 cycles for completion.

4.2.2 Microcontroller Algorithms

The flow chart shown in Fig. 4.8 shows the main algorithms of the SRM controller and how they relate to each other.

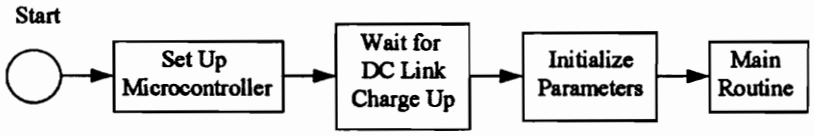


Fig. 4.8 Main algorithms of SRM control program

The main routine contains the algorithms for the desired control of the SRM drive. The other routines of the software work to move control to the main routine. Control is displaced from the main routine if the controller is reset, or if the direction of the rotor has changed.

Once the microcontroller is turned on, the memory banks are set up, the interrupts are disabled, flags are cleared and the gate control signals are off. For 500 milliseconds, the controller waits until the DC link capacitors charge to the full DC link voltage. In order to begin the main routine of the program, the parameters of the drive are established in "Initializing Parameters," and from there, control is passed to the main routine.

4.2.2.1 Initializing Parameters

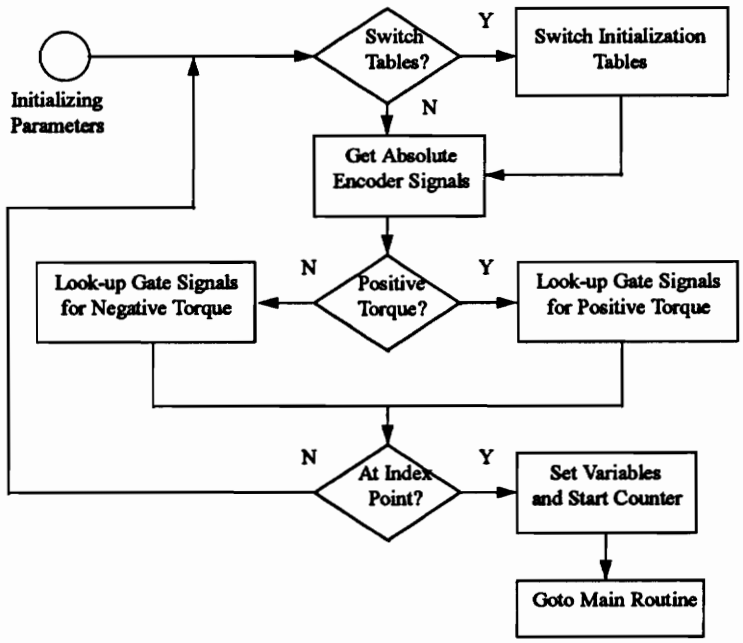


Fig. 4.9 Initializing parameters

When the controller is powered up, the only information the controller has concerning the rotor position are the four absolute encoder signals. Fig. 4.9 contains the flow chart describing the procedure to initialize the software control. The counter is used to determine the absolute rotor position and an index point is chosen so the counter receives a hard reset for every 60 mechanical degrees. The index point for the prototype is the boundary between the A and B absolute encoder signals.

Index

1	1	Tbl	T	D	C	B	A
7	6	5	4	3	2	1	0

Output

0	0	Td	Tc	Tb	Ta	Tbd	Tac
7	6	5	4	3	2	1	0

Tbl - Table Selection(1 - Table 1, 0 - Table 0)

T - Torque Polarity(1 - positive, 0 - negative)

D - Absolute Encoder Signal D

C - Absolute Encoder Signal C

B - Absolute Encoder Signal B

A - Absolute Encoder Signal A

Tac - Tac gate control signal

Tbd - Tbd gate control signal

Ta - Ta gate control signal

Tb - Tb gate control signal

Tc - Tc gate control signal

Td - Td gate control signal

Fig. 4.10 Initialization look-up table

In order to start the counter, the rotor has to be driven past the index point. Until then, phase firing control is performed by using the absolute encoder signals and torque polarity to look-up the gate control signals. Fig. 4.10 illustrates the index and output for this table. At the start of the initialization, the microcontroller chooses which table to select in order to index the gate control signals. Over a single absolute encoder signal, the torque varies for the phase in which to fire. At one end of the absolute encoder signal the phase is fired at, or near, alignment resulting in little torque being produced. To overcome this problem, two tables are created and alternated every 2 milliseconds. At the moment one table is producing a small torque, the other table is producing the maximum torque. Provided that the absolute encoder signals are aligned with the phases such that the low torque regions are avoided, one table for start-up will suffice.

Once the table has been selected, the absolute encoder signals are received at port 1 of the 8748H. By testing the signal at the T0 pin, the torque polarity from the speed controller is found, combined with the absolute encoder signals, and used to look-up the appropriate gate control signals. The gate control signals are output through port 2.

The absolute encoder signals are used to determine if the rotor has passed the index point. This is carried out by use of a second look-up table shown in Fig. 4.11.

Index

1	Dir	Ph1	Ph0	D	C	B	A
7	6	5	4	3	2	1	0

Output

0	0	Ph1	Ph0	0	0	Res	Chg
7	6	5	4	3	2	1	0

Dir - 1= For. Direction, 0 = Rev. Direction

Ph1 - msb of previous phase

Ph0 - lsb of previous phase

D - Absolute Encoder Signal D

C - Absolute Encoder Signal C

B - Absolute Encoder Signal B

A - Absolute Encoder Signal A

Ph1 - msb of present phase

Ph0 - lsb of present phase

Res - if 1, passed index point

Chg - if 1, direction has changed

Fig. 4.11 Look-up table for direction change and counter reset

Along with the encoder signals, the previous phase information and current rotor direction are needed to determine if the index point has been passed. This table also determines if the rotor direction has changed and is used in the main routine. If the index point has been passed, the variables are initialized and the counter started in order to calculate the absolute rotor position. Control is then passed to the main routine. If the index point has not been passed, control returns to the beginning of the initialization routine and the procedure is repeated.

4.2.2.2 Main Routine

Fig. 4.12 contains the flow chart for the main routine.

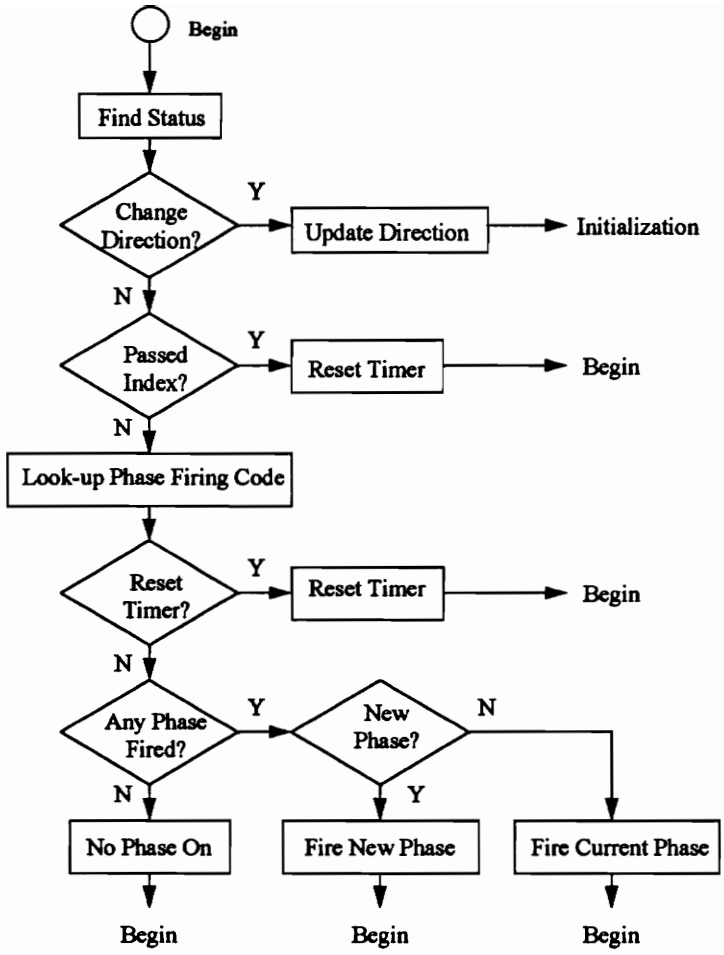


Fig. 4.12 Main routine

At the top of the main routine, the status of the drive is found. The two status details checked are the rotor direction and the index point. The same look-up table, as shown in Fig. 4.11, is used to check these status details. If a direction change is noticed, the microcontroller leaves the main routine and re-enters the initialization program to reset the software control. If a direction change has occurred, the absolute rotor position is lost and the status of the microcontroller must be reset. This is the reason for returning to the initialization procedure.

If the index point is passed, the counter is reset. A hard reset, such as the index point, ensures the controller does not have a corrupt absolute rotor position for longer than 60 degrees traveled by the rotor. Besides resetting the counter at this time, the advance angle and fall angle are recalculated. Fig. 4.13 contains the flow chart illustrating the counter reset and the angle calculations.

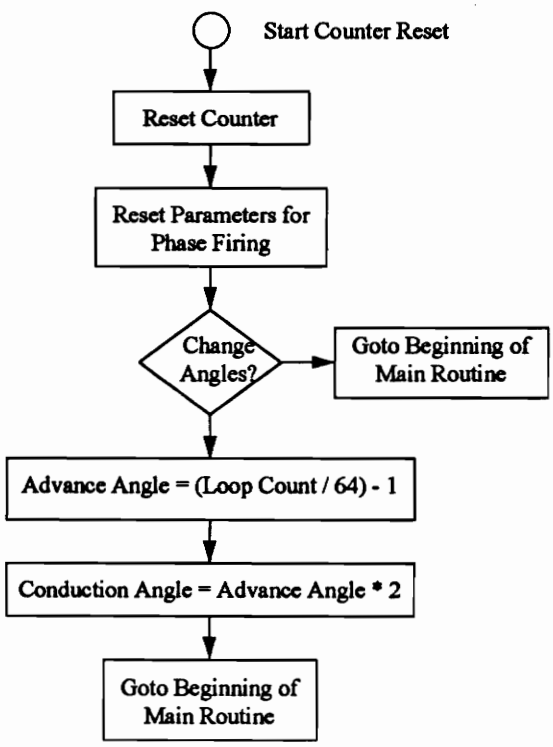


Fig. 4.13 Counter reset

To prevent the advance and fall angle from changing too frequently, a software counter is decremented from 255 to zero before a new calculation is performed. To calculate the new advance and fall angles, another software counter is decremented each time the main routine is executed. As the speed increases, the number of times the main routine executes decreases, leaving a larger count in the software counter. When the hard reset occurs, the software counter is reset to 255. When calculating the angles, the counter

value is divided by 64 plus one. This becomes the fall angle which has a maximum value of 12 degrees at 2500 rpm. The fall angle is divided by 2 to form the advance angle, which is 3 degrees at 2500 rpm.

If the index point has not been passed and the direction has not changed, then another look-up table is used to find the phase firing code. Fig. 4.14 details the index and output of the look-up table.

Index

Quad	FA2	FA1	FA0	C3	C2	C1	C0
7	6	5	4	3	2	1	0

Quad - 1= Regeneration, 0 = Motoring
FA2 - bit 2 of fall angle
FA1 - bit 1 of fall angle
FA0 - bit 0 of fall angle
C3 - bit 3 of counter value
C2 - bit 2 of counter value
C1 - bit 1 of counter value
C0 - bit 0 of counter value

Output

NP	PO	RC	1	Th3	Th2	Th1	Th0
7	6	5	4	3	2	1	0

NP - if 1, rotate phase indication
PO - if 1, gate control signals off
RC - if 1, reset counter
Th3 - bit 3 of absolute rotor position
Th2 - bit 2 of absolute rotor position
Th1 - bit 1 of absolute rotor position
Th0- bit 0 of absolute rotor position

Fig. 4.14 Look-up table for phase firing codes

To look-up the phase firing codes, the counter value, fall angle, and information on whether the drive is in regeneration or motoring is made necessary. To determine whether the drive is in regeneration or motoring, the controller looks at the direction and torque polarity. If the torque is positive and the direction is forward, or if the torque is negative and the direction is reverse, the drive is in a motoring mode. Otherwise, it is in a regeneration mode.

The outputs of the look-up table are the absolute rotor position, reset counter flag, phase off flag and new phase flag. The absolute rotor position is output through the

8748H BUS port for torque smoothing purposes. First, the controller determines if the counter should be reset. The phase firing patterns are the same for each phase so to reduce the number of bits need to index the phase firing code, the counter is reset each time it reaches the count of 11, which is the start of a new phase pattern. The next flag tested is the phase-off flag. If this flag is high, then no gate control signals are active. In the case where there is a fall angle, there may be periods where no phases are excited. The final flag tested is the new phase flag. In the software, a variable keeps track of the current phase to be fired. Each time the counter is reset, the variable is updated to the next phase to be fired according to the quadrant of operation. Before the counter resets, the next phase in the sequence must be fired. This flag indicates the next phase should be fired instead of the current phase indicated. If none of the flags are active, the current phase is fired.

4.2.3 Torque Smoothing Table

Separate from the look-up tables used in the microcontroller, a look-up table is used to store the nonlinear torque map of the prototype SRM. The information for the map is calculated by using SRMCAD. SRMCAD calculates the output torque for a given current and rotor position. For torque smoothing, an inverse map which maps the current for a given torque and rotor position is desired. To accomplish the inverse map, the torque data is entered into Matlab, and for each torque and rotor position, the current is found using an interpolation procedure. The inverse map is stored into one matrix and by using Matlab's "fwrite"[21] command, the map is written as a binary output file. The binary file is then loaded into an EPROM programmer and stored in the EPROM. Fig. 4.15 shows the indexing for the map.

Index

0	θ_3	θ_2	θ_1	θ_0	T_7	T_6	T_5	T_4	T_3	T_2	T_1	T_0

θ_3 - θ_0 : Absolute Rotor Position

T_7 - T_0 : Digitized Torque Command

Output

I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0

I_7 - I_0 : Digitized Torque Smoothing Command

Fig. 4.15 Index for torque smoothing map

Chapter 5: Experimental Verification

To verify the controller architecture and design methodology, a prototype controller is built according to the hardware schematic listed in Appendix E and the software listing in Appendix F. Verification covers the three control blocks: current control, speed control, and phase firing control. Through verifying these types of control, it is shown the proposed architecture meets performance levels necessary for low performance drives.

5.1 Current Control Verification

The current control loop is implemented using the analog PI controller and the PWM scheme as mentioned in Chapter 4. The gain for the current controller is 9.42 and the delay is 1.13×10^{-4}. These are the controller values calculated in Chapter 2 for the 5hp SRM prototype. Fig. 5.1 is a simulation of the current loop at 1000 rpm with a current command of 10A and a DC link voltage of 400V.

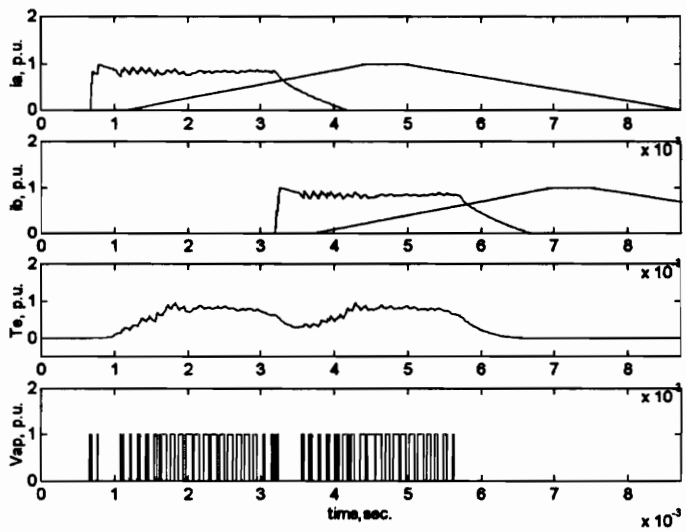


Fig. 5.1 Simulated phase current, torque and applied voltage

The advance angle is 3 degrees and fall angle is 9 degrees. The phase currents, i_a and i_b, are shown with their ideal phase inductance. The rise time for the current is approximately 0.05 - 0.1ms, and the fall time is approximately 0.85ms. Also shown is the electromagnetic torque, T_e, and the applied voltage, V_{ap}.

Fig. 5.2 is the actual phase current at a commanded value of 10A, 400V DC link voltage, and a speed of 1000rpm. In the figure, each vertical division equals 2.5A of current. The rise time is approximately 0.15ms and fall time is approximately 0.85ms. The errors between the actual and simulation may be attributed to deviations in the assumed phase inductance generated by the SRMCAD.

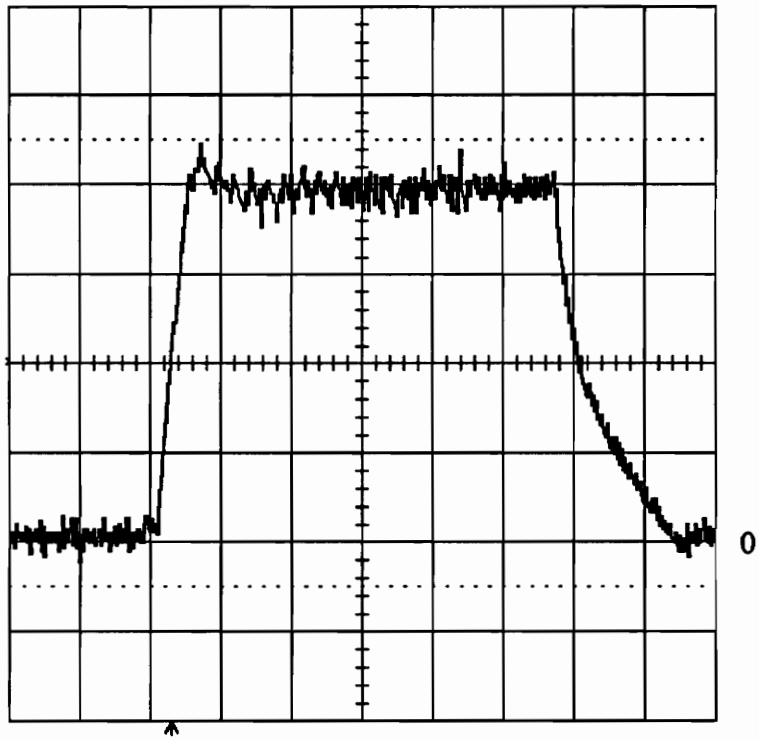


Fig. 5.2 Actual phase current at 1000rpm

x-axis: 0.5ms/div.
y-axis: 2.5A/div.

5.2 Speed Control Verification

The speed controller is designed by the same architecture as mentioned in Chapter 4. The gain for the speed controller is 8.39 and the delay is 0.04. Fig. 5.3 is a four quadrant simulation of the speed loop given a pos./neg. step speed command of 1250rpm. The load for the simulation is a 10% friction load. The rise time for the speed, ω_m , is

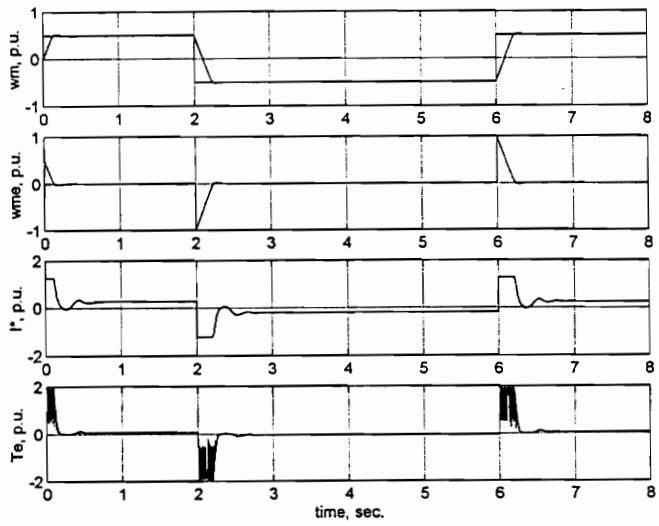


Fig. 5.3 Dynamic simulation of a bi-directional 1250rpm step speed command

0.25s, the percent overshoot is 5%. Also shown in Fig. 5.3 is the torque command, i^*, and the electromagnetic torque, T_e.

In Fig. 5.4 the actual results of the step speed command on the prototype 5hp SRM is shown. For the figure, each vertical division equals to 500rpm. The rise time for the actual is 0.3s and the percent overshoot is 10%. The reasons for the deviations are the following. First, the current loop is not simulated in the speed simulation. The commanded value of current is assumed the actual current. Second, the problem with small torque produced near alignment, which was mentioned in section 4.2.2.1, is not simulated. This problem reduces the torque generated in the regenerative quadrants.

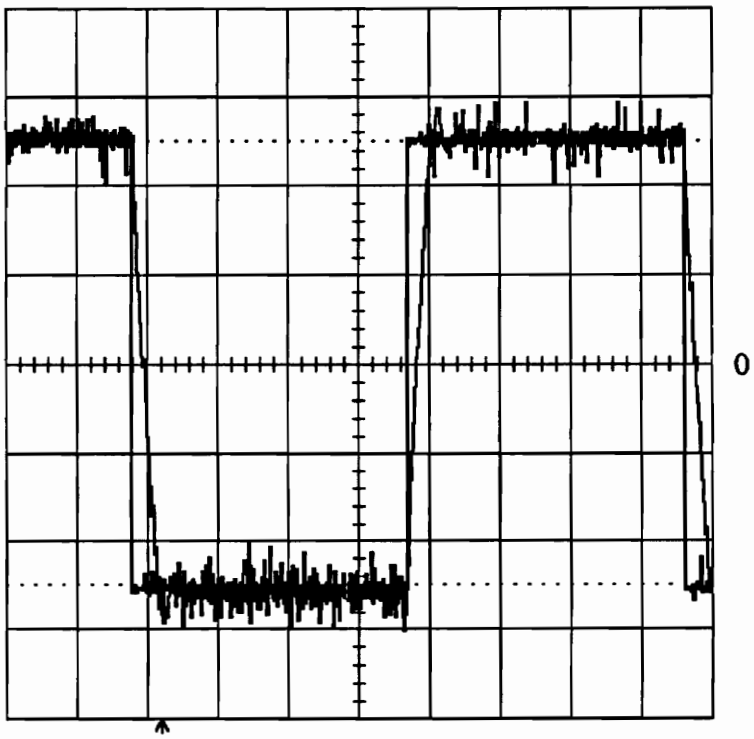


Fig. 5.4 Actual response to a bi-directional 1250rpm step speed command

x-axis: 1 sec/div.

y-axis: 500 rpm/div.

5.3 Torque Smoothing Verification

The Torque Smoothing tables embedded in the controller generate a nonlinear current command for the current controller. Shown in Fig. 5.5 is a simulation of torque smoothing for a speed of 500rpm. The commanded torque is 6.6 N-m from the PI speed controller. The nonlinear current command, T_s^*, is shown in the first plot of Fig. 5.5. The phase currents, i_a and i_b, illustrate how the phase current follows the nonlinear current command. The electromagnetic torque, T_e , demonstrates the effect of torque smoothing.

Because the recovery current from the previous phase contributes torque, the total output torque is not completely smooth.

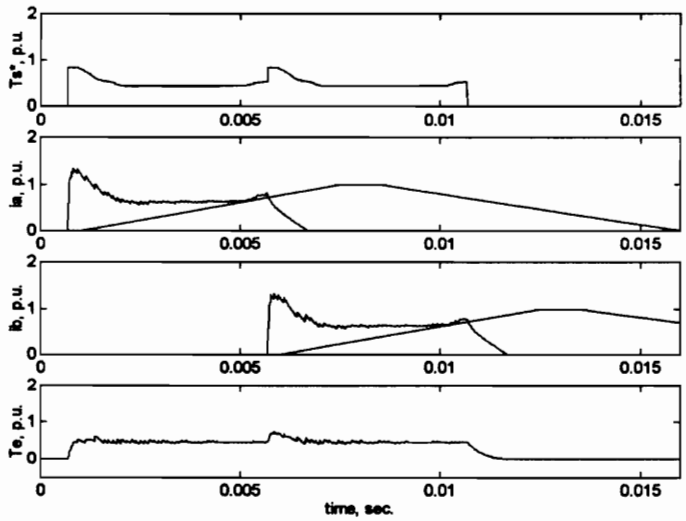


Fig. 5.5 Simulation of torque smoothing

Fig. 5.6 shows the actual nonlinear current command and one of the four phase currents for the same conditions as mentioned in the simulation. Channel 1 is the nonlinear current command where each vertical division equals a command of 7.5A. Channel 2 is the phase current, i_a, where 10mV equals 5A. The phase current is able to follow the torque smoothing command with little error. There are small differences in the actual nonlinear current command and that of the simulation. Most of the error is due to the resolution of the rotor position, which is 1.5 degrees.

T_s*:	Channel 1	i_a:	Channel 2
	x-axis: 1 msec./div.		x-axis: 1 msec./div.
	y-axis: 5 V/div.		y-axis: 5 A/div.

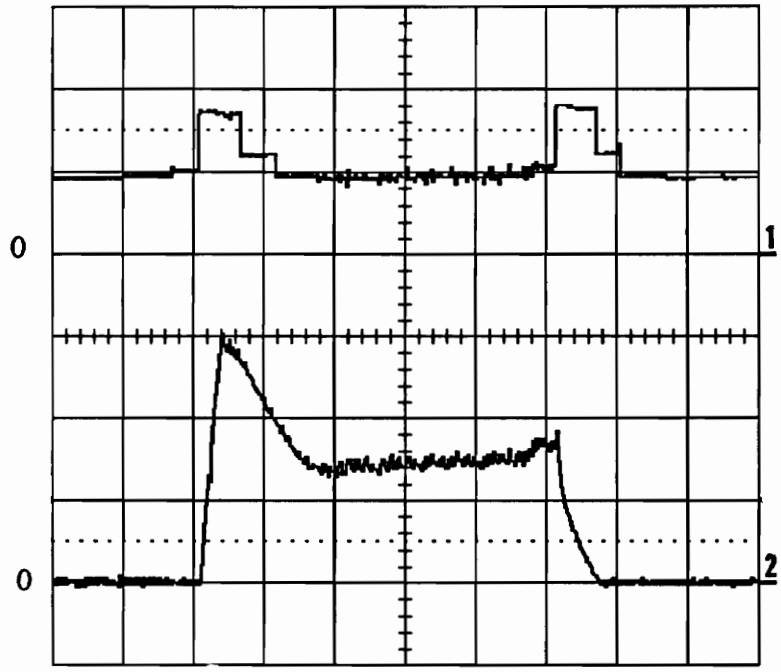


Fig. 5.6 Actual torque smoothing command and current waveform

5.4 Advance and Fall Angle Control

One advantage of possessing advance and fall angle control is the ability to achieve maximum efficiency over the speed range of the drive system. The prototype SRM is tested for efficiency over a range of advance and fall angles. Fig. 5.7 and 5.8 show the efficiency for four different advance and fall angles. Test #1, in Fig. 5.7, uses no advance angle and a fall angle of 6 degrees. Test #2 uses an advance of 1.5 degrees and a fall angle of 7.5 degrees. In Fig. 5.8, test #3 uses an advance of 1.5 degrees and a fall of 9 degrees, while test #4 uses an advance angle of 6 degrees and a fall angle of 12 degrees.

The efficiency tests were performed with 70% constant load and a DC link voltage of 621V. The efficiency is measured for the SRM drive between the DC link and the generating load. For some of the angle combinations, the prototype was not able to reach the maximum speed, which is the reason for no efficiency data at those points.

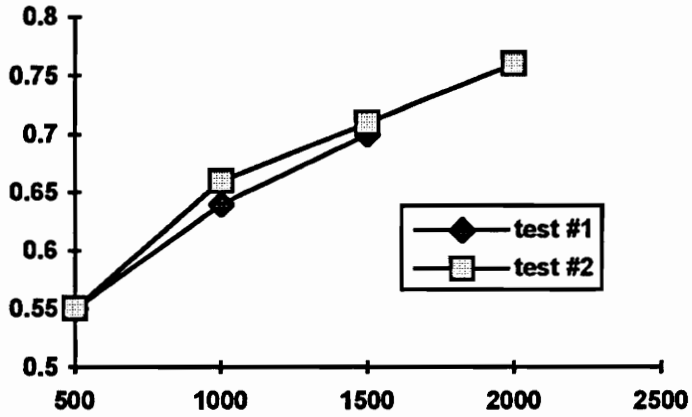


Fig. 5.7 Efficiency for test #1 and test #2

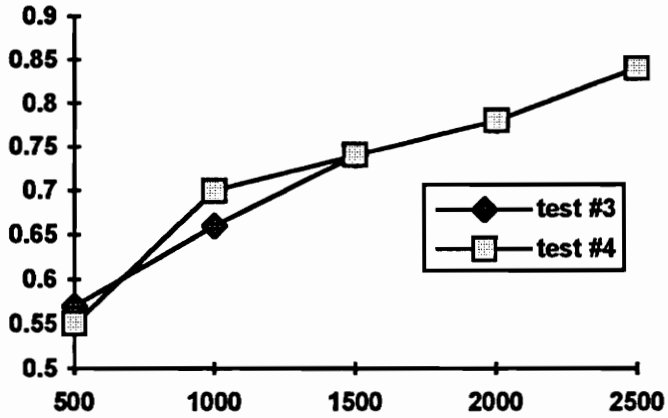


Fig. 5.8 Efficiency for test #3 and test #4

The plots illustrate the effect of advance and fall angles on the efficiency and on its ability to operate up to the maximum speed. Fig. 5.9 is the efficiency for the prototype SRM when self-calculation of angles is performed. The calculation of angles according to speed

is listed in Table 5.1. The efficiencies of the self-calculation match the maximum efficiencies over the speed range.

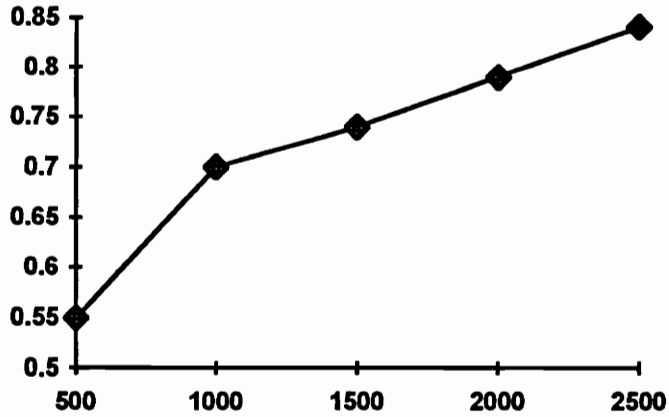


Fig. 5.9 Efficiency for self-calculation of advance and fall angles

Table 5.1 Controller calculated angles over the speed range

rpm	Advance Angle	Fall Angle
0 - 599	0	6
600 - 1199	1.5	7.5
1200 - 1799	3	9
1800 - 2399	4.5	10.5
2400 - 2500	6	12

Chapter 6: Conclusions and Recommendations

6.1 Conclusions

The following are the conclusions of this research study.

1. A SRM controller architecture was developed and verified. Through experimental tests, the controller proves its ability to meet low performance standards.

2. A design methodology for current and speed control design of variable speed SRM drives, based on the linearized small signal model, was developed, tested and experimentally verified. The unique methodology is simple and adequate for low performance applications.

3. A low cost method of torque smoothing was developed and verified. The proposed Torque Smoothing method uses simple components(A/D converter, D/A converter, and EPROM) to provide the nonlinear current command mapping.

4. A novel method of current sensing using one current sensor is experimentally verified. The new current sensing method significantly reduces the cost of the controller without impairing control performance.

5. The SRM controller architecture demonstrates its ability handle a new, low cost speed/position encoder.

6. The SRM controller is able to provide both advance and fall angle control. This type of control has been shown to maximize the efficiency of the drive system for the range of speed.

7. The SRM architecture provides four-quadrant operation and complies to industrial standard inputs.

8. The cost of construction for the prototype SRM controller is below $20(US).

6.2 Recommendations for Future Study

The following recommendations are made for future study.

1. Derivation of other control design methodologies which provide accurate speed and current control of the SRM.

2. Modification of the SRM architecture to accommodate more complex control and higher speed ranges.

3. Accurate modeling methods of the SRM for controller design

Appendix A: List of Symbols

λ	Flux	[B]
θ	Rotor Position	[rad]
θ_a	Advance Angle	[rad]
τ_a	time constant	[s]
ω_m	Rotor Speed	[rad/s]
ω_m^*	Speed Command	[V]
ω_{mo}	Rated Rotor Speed	[rad/s]
B	SRM Friction Constant	[N-m/rad/s]
B_l	Load Friction Constant	[N-m/rad/s]
H_c	Current Feedback Gain	[V/A]
H_w	Speed Feedback Gain	[V/rad/s]
i	Current	[A]
i^*	Current Command	[V]
i_o	Rated Current	[A]
J	Rotor Inertia	[kg-m^2]
K_b	Linearized Torque Constant	[N-m/A]
K_b	Linearized back EMF Constant	[V/rad/s]
K_r	Converter Gain	[V/V]
K_c	Current Controller Gain	[V/V]
K_v	Speed Controller Gain	[V/V]
L	Inductance	[H]
L_u	Unaligned Inductance	[H]
P	Power	[W]
P_a	Airgap Power	[W]
R	Linearized Phase Resistance	[Ω]
R_p	Non-linearized phase resistance	[Ω]
T_c	Current Controller Time Constant	[s]

T_e	Airgap Torque	[N-m]
T_l	Load Torque	[N-m]
T_r	Converter Time Constant	[s]
T_v	Speed Controller Time Constant	[s]
T_w	Speed Feedback Time Constant	[s]
V_{dc}	DC Link Voltage	[V]
v_c	PWM Duty Cycle Command	[V]

Appendix B: Motor and System Parameters

Command Signal Levels:	+/- 10V
DC Link Voltage:	400 V
Max. Current:	15 A
PWM Chopping Frequency:	8kHz
Phase Resistance:	0.931Ω
Power:	5 hp
Rated Current:	10 A
Rated Speed:	2500 rpm
Rotor Friction Constant:	0.001 N-m/rad/s
Rotor Inertia:	0.006 kg-m^2
Speed Feedback Gain:	0.0382 V/rad/s
Speed Feedback Time Constant:	0.01 s

Appendix C: Inductance Profile and Sensor Disk Signals

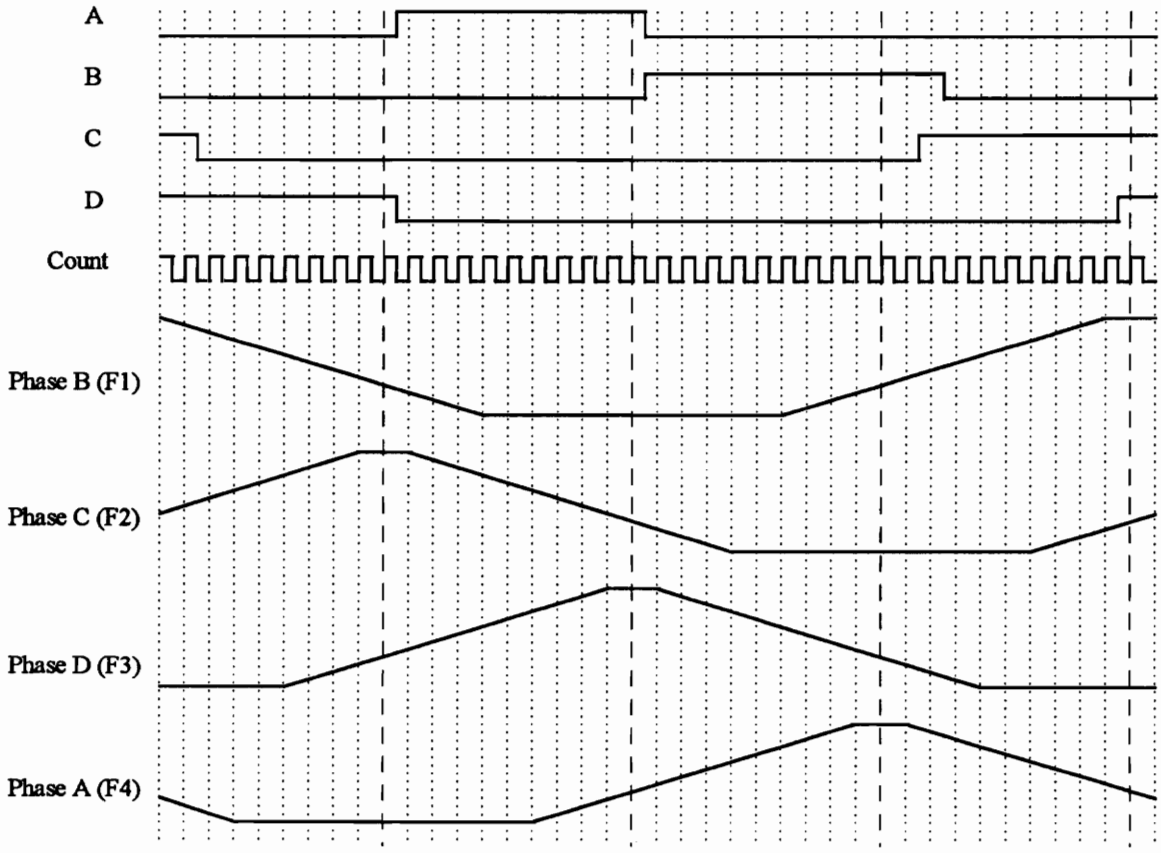


Fig. A.1 Ideal inductance profile and sensor disk signals

Appendix D: Microcontroller Specifications

Component:	Intel 8748H
Processing Speed:	740kHz
Memory Size:	1024 bytes of EPROM and 256 bytes of RAM
Data Bus Size:	8-bit
Input/Output Ports:	Ports 1 and 2 and BUS(All 8-bits)
Hardware Counter:	1 8-bit counter. Max. Frequency = 175kHz
Special Modes:	Power Down, Single Step, Sync Mode
No. of Instructions:	96

Appendix E: Prototype SRM Hardware Schematic

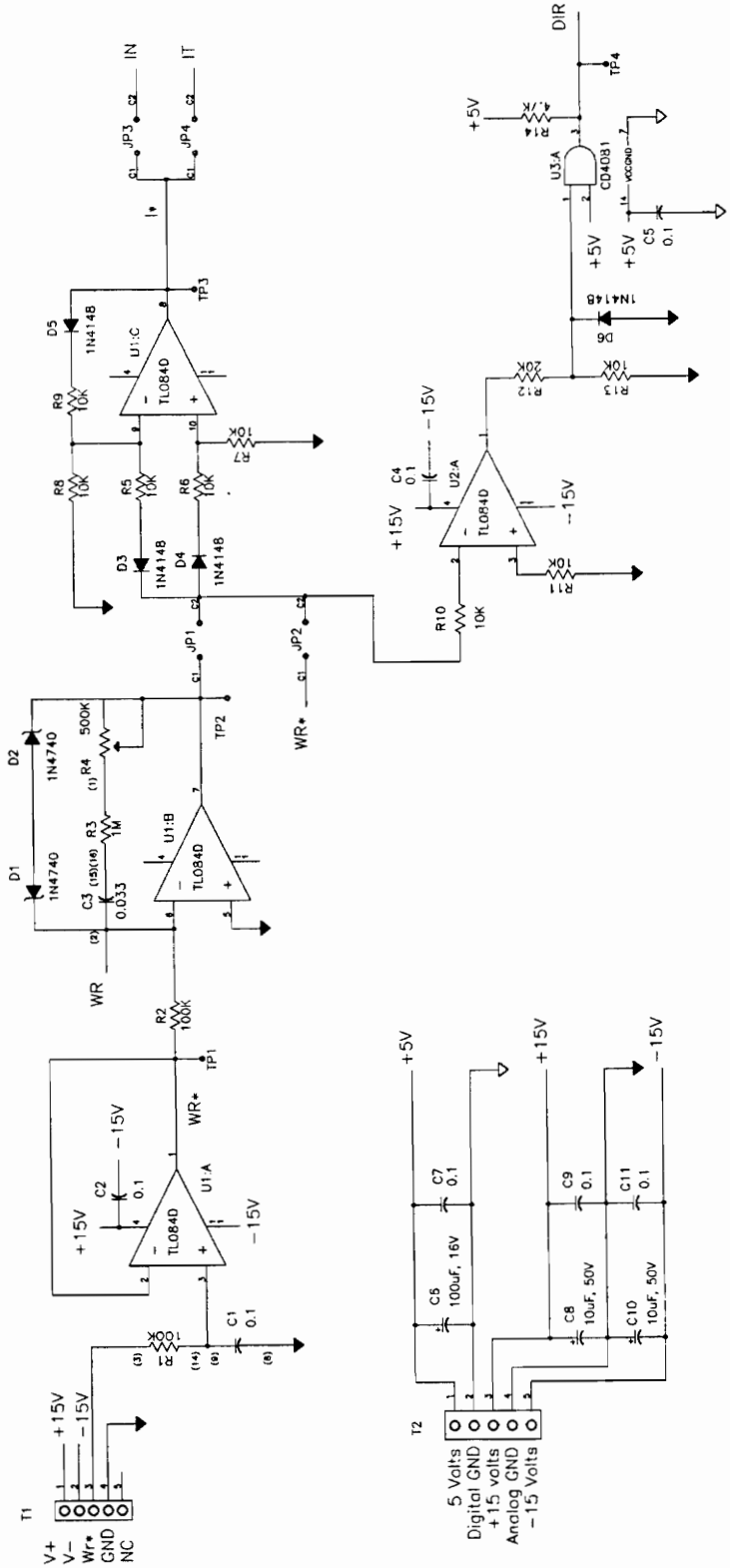


Fig. A.2 PI speed controller circuit

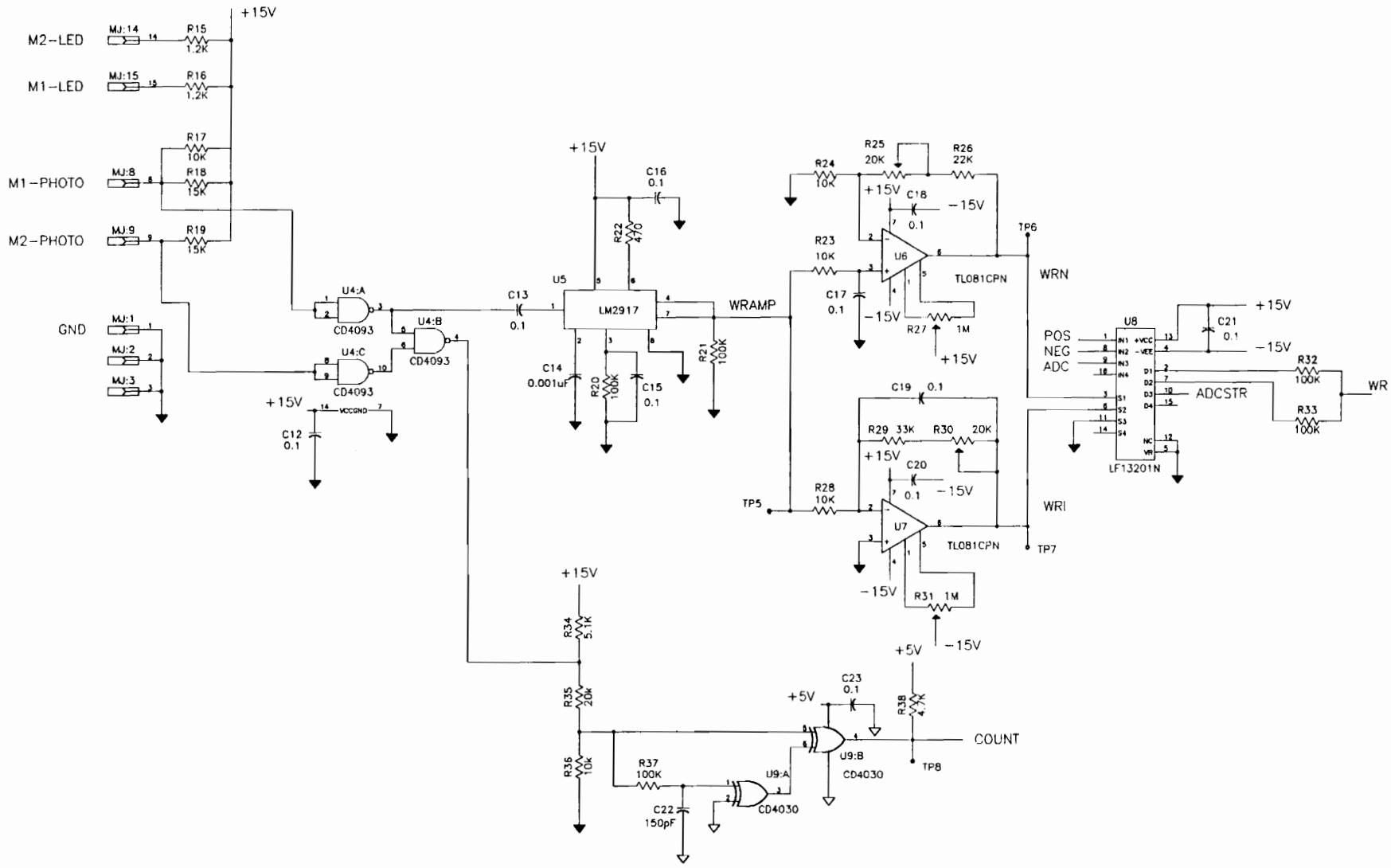


Fig. A.3 Speed feedback circuit

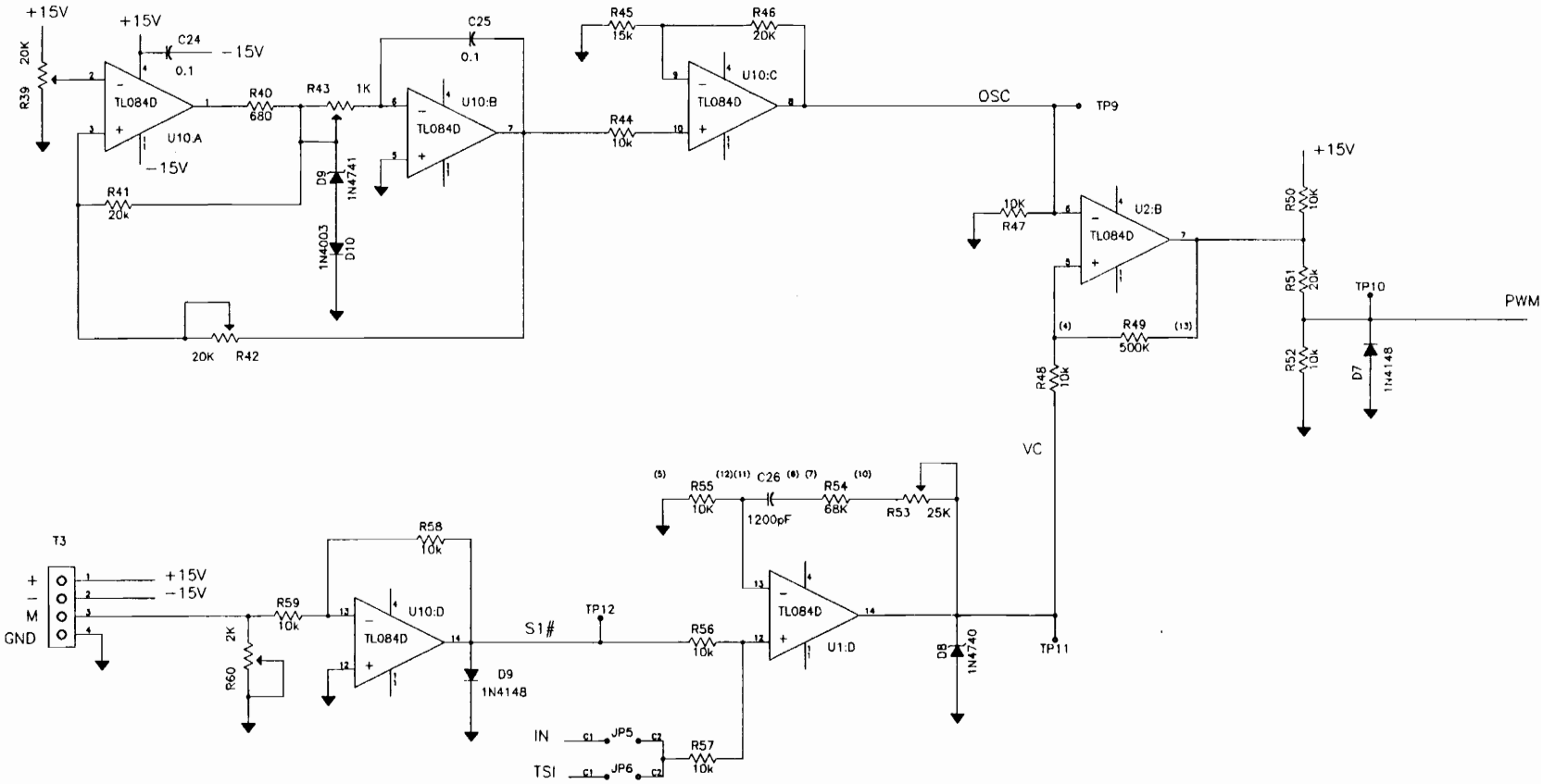


Fig. A.4 Current controller and feedback circuit

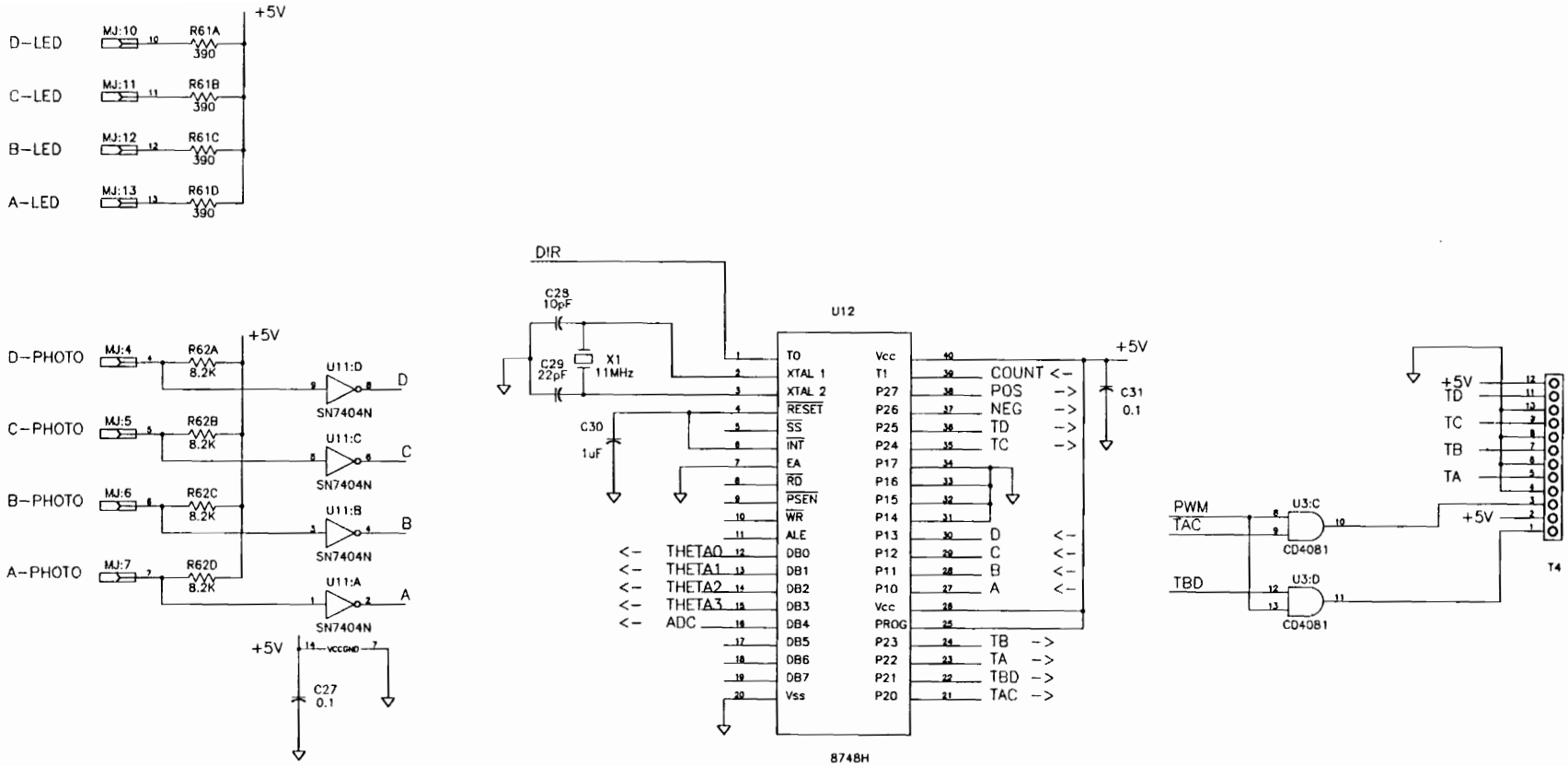


Fig. A.5 Microcontroller circuit

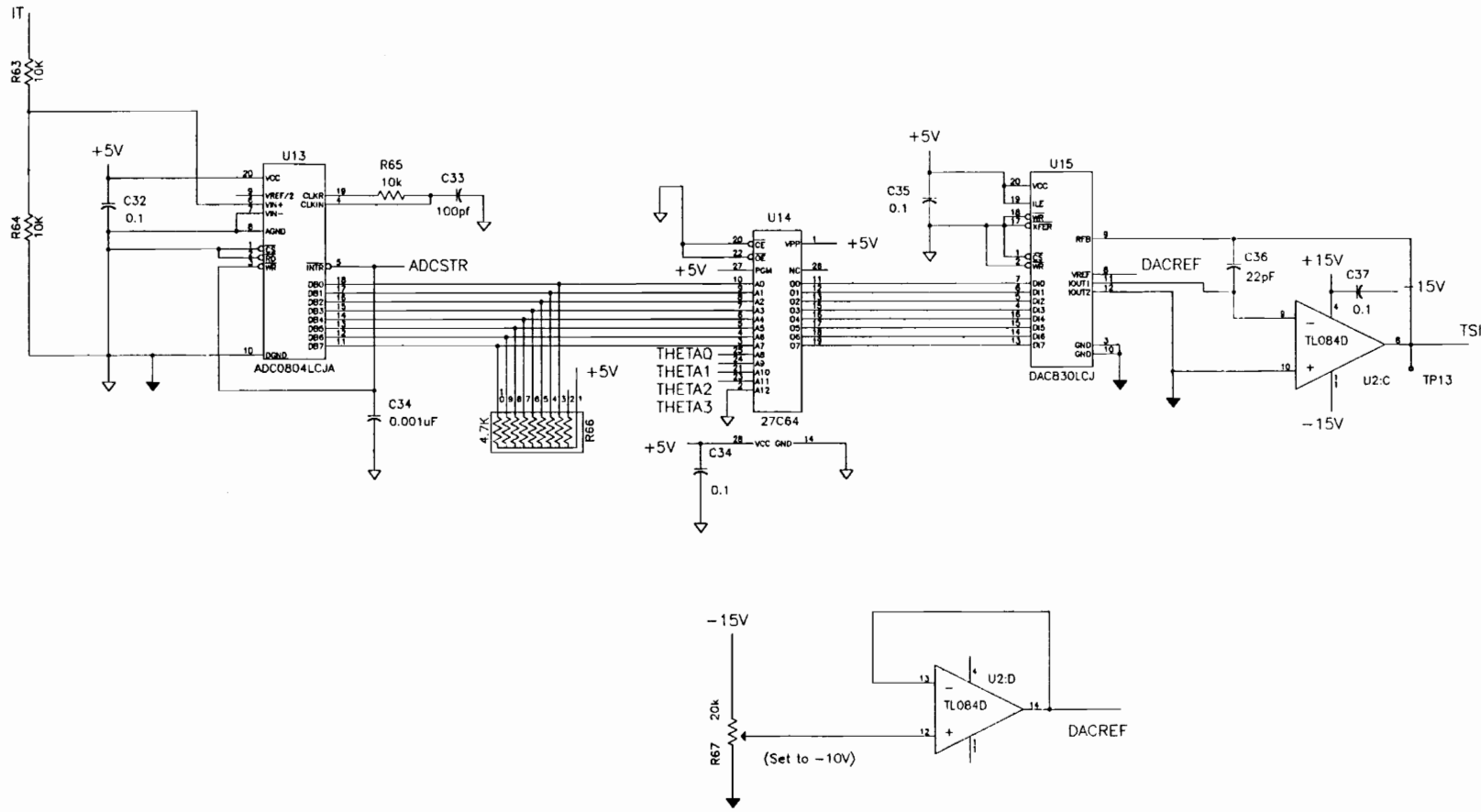


Fig. A.6 Torque smoothing circuit

Appendix F: SRM Control Program

```
;*************************************************
;*  SRM Control Program                          *
;*                                               *
;*  Filename:   SRM.A48                          *
;*  Date:       3.16.96                          *
;*                                               *
;*  Tasks:      1)  Provide Gate Drive Signals   *
;*              2)  Advance Angle Control        *
;*              3)  Provide Absolute Rotor Pos.  *
;*              4)  Speed Feedback Selection     *
;*              5)  Torque Smoothing Control     *
;*                                               *
;*  Inputs:     1)  DIR - Torque Com. Polarity   *
;*              2)  COUNT - Incremental Pulses   *
;*              3)  Absolute Encoder Signals     *
;*                                               *
;*  Outputs:    1)  POS & NEG - Speed Fbk. Sel.  *
;*              2)  THETA - Absolute Rotor Pos.  *
;*              3)  ADC, DAC, and EPROM  -       *
;*                  Torque Smoothing Control     *
;*              4)  Gate Drive Control Signals   *
;*                                               *
;*  Symbols:    R0 - TEMP, temporary reg.        *
;*              R1 - CYCNT, adv. ang. cycle      *
;*              R2 - PREV, previous sensor sig.  *
;*              R3 - PHASE, current phase        *
;*              R4 - LOOP, reset loop counter    *
;*              R5 -                             *
;*              R6 - DWL, dwell angle            *
;*              R7 - ADV, advance angle          *
;*              F0 -                             *
;*              F1 - Rotor Direction             *
;*              T0 - Torque Command Polarity     *
;*************************************************


;*************************************************
;* Initialize 8748                               *
;*************************************************
            .ORG    10H

START:      SEL     MB0                 ;Select Memory Banks
            SEL     RB0
            JMP     WAIT                ;Finish initialization

;*************************************************
;* Begins the main operation of the controller   *
;* when absolute rotor position can be found.    *
;*************************************************

OPER:       IN      A,P1                ;Get encoder signals
            ORL     A,R2                ;Make table index
            JF1     SETPOS

            ORL     A,#80H              ;Currently Neg. Direction
            MOV     R0,#00H             ;Load quadrant indication
```

```
            JMP     SKIP

SETPOS:     ORL     A,#0C0H         ;Currently Pos. Direction
            MOV     R0,#80H         ;Load quadrant indication

SKIP:       CALL     GETDIR         ;Check status
            JB0      CHGDIR         ;Changed Direction?
            JB1      TRES           ;Reset Timer?

PHASE:      MOV      A,T            ;Get counter value
            ORL      A,R6           ;Append dwell angle
            JT0      PTOR           ;Torque Polarity?

            ORL      A,R0           ;Load direction
            XRL      A,#00H         ;Neg. torque
            JB7      REGEN          ;In regeneration
            MOVP3    A,@A           ;Look-up table
            MOV      R0,#0E0H       ;Neg. torque
            JMP      GETCODE        ;Get phase signals

PTOR:       ORL      A,R0           ;Load direction
            XRL      A,#80H         ;Pos. torque
            JB7      REGEN          ;In regeneration
            MOVP3    A,@A           ;Look-up table
            MOV      R0,#0F0H       ;Pos. torque

GETCODE:    CPL      A              ;Subtract advance angle
            ADD      A,R7
            CPL      A
            OUTL     BUS,A          ;Output rotor position
            JB5      RESTIME        ;Reset timer and phase
            JB6      NOPHASE        ;No phase on
            JB7      NEWPHASE       ;Fire new phase
            MOV      A,R3           ;Load phase indication
            ANL      A,#0FH         ;Clear top four bits
            ORL      A,R0           ;Append torque polarity
            MOVP     A,@A           ;Get gate signals
            ORL      A,R5           ;Append speed feedback
            OUTL     P2,A           ;Output gate signals
            DJNZ     R4,OPER        ;See if cycle count zero
            MOV      R4,#01H        ;Keep at zero
            JMP      OPER

REGEN:      SEL     RB1             ;Select register bank 1
            DJNZ    R0,SWOFF        ;Switch between table1's?
            MOV     A,R1            ;Switch table
            XRL     A,#20H
            MOV     R1,A
            MOV     R0,#01FH        ;Reset switch counter

SWOFF:      IN      A,P1            ;Get sensor input
            JT0     POSREG          ;See if Positive or Neg. Torque

            ORL     A,#0C0H         ;Neg. Torque Command
            ORL     A,R1            ;Table selection
            SEL     RB0             ;Select register bank 0
            CALL    INISEQ          ;Look-up gate signals
            ORL     A,R5            ;Append Speed Fbk.
            OUTL    P2,A            ;Output gate signals
            JMP     OPER
```

```
POSREG:     ORL    A,#0D0H          ;Show positive torque
            ORL    A,R1             ;Table selection
            SEL    RB0              ;Select register bank 0
            CALL   INISEQ           ;Get Gate Signal
            ORL    A,R5             ;Append Speed Fbk.
            OUTL   P2,A             ;Output gate signals
            JMP    OPER

RESTIME:    CLR     A               ;Clear acc.
            MOV     T,A             ;Load count value
            MOV     A,R5            ;Load speed feedback
            JB6     RIGHT           ;Shift phase indication left

            MOV     A,R3            ;Load phase indication
            RL      A               ;Rev. Dir., shift right
            MOV     R3,A            ;Store phase indication
            JMP     OPER

RIGHT:      MOV     A,R3            ;Load phase indication
            RR      A               ;For. Dir., shift left
            MOV     R3,A            ;Store phase indication
            JMP     OPER

NEWPHASE:   MOV     A,R3            ;Load phase indication
            JF1     LROT            ;Which way rotate

            RR      A               ;Neg. Dir., rotate right
            JMP     CONT

LROT:       RL      A               ;Pos. Dir., rotate left

CONT:       ANL     A,#0FH          ;Clear top four bits
            ORL     A,R0            ;Pos. torque
            MOVP    A,@A            ;Get gate signals
            ORL     A,R5            ;Append speed feedback
            OUTL    P2,A            ;Output gate signals
            DJNZ    R4,OPER         ;See if cycle count zero
            MOV     R4,#01H         ;Keep at zero
            JMP     OPER

NOPHASE:    MOV     A,R5            ;Load speed feedback
            OUTL    P2,A            ;No gates on
            JMP     OPER

;*************************************************

CHGDIR:     STOP   TCNT             ;Stop Timer
            MOV    A,#2AH           ;Freeze Torque Smoothing
            OUTL   BUS,A            ;Output rotor position
            MOV    R0,#1FH          ;Load table switch counter
            MOV    R1,#00H          ;Initially table1_1
            JMP    INI

;*************************************************

TRES:       STOP   TCNT             ;Reset counter
            MOV    A,R7             ;Load advance angle
            MOV    T,A
            STRT   CNT
```

```
            JF1     TRESPOS             ;What phase to reset to?
            JT0     SECOND              ;In second quadrant?

            MOV     R3,#88H             ;Third Quad., load phase D
            JMP     CHECK

SECOND:     MOV     R3,#11H             ;Sec. Quad., load phase A
            JMP     CHECK

TRESPOS:    JT0     FIRST               ;First Quadrant?

            MOV     R3,#88H             ;Fourth Quad., load phase C
            JMP     CHECK

FIRST:      MOV     R3,#11H             ;Pos. direction, phase A

CHECK:      DJNZ     R1,INTERM          ;Not time to change advance
            MOV      A,R4               ;Acc = loop counter
            ANL      A,#0C0H            ;Clear bottom 6 bits
            SWAP     A                  ;Swap nibbles
            RR       A                  ;Divide by four
            RR       A
            JNZ      ADV                ;Is there some advance?
            MOV      R7,#01H            ;No advance
            MOV      R1,#0FFH           ;Reload cycle counter
            JMP      INTERM

ADV:        INC      A                  ;Increment acc.
            MOV      R7,A               ;New advance angle
            MOV      R1,#0FFH           ;Reload cycle counter

INTERM:     MOV      R4,#0FFH           ;Reload loop counter
            JMP      OPER               ;Continue in loop


;*************************************************
;* Table 1 is used to calculate the appropriate *
;* gate signal.                                 *
;*                                              *
;*  Input:   bit 0 - A phase                    *
;*               1 - B phase                    *
;*               2 - C phase                    *
;*               3 - D phase                    *
;*               4 - TORQUE POLARITY(1=POS.)    *
;*               5 - 0                          *
;*               6 - 0                          *
;*               7 - 0                          *
;*                                              *
;*  Output:  bit 0 - Tac gate signal            *
;*               1 - Tbd gate signal            *
;*               2 - Ta gate signal             *
;*               3 - Tb gate signal             *
;*               4 - Tc gate signal             *
;*               5 - Td gate signal             *
;*               6 - 0                          *
;*               7 - 0                          *
;*************************************************
```

```
            .ORG      0E0H

TABLE1:     .DB       00H             ;0 - Neg. Torque
            .DB       22H             ;1
            .DB       05H             ;2
            .DB       00H             ;3
            .DB       0AH             ;4
            .DB       00H             ;5
            .DB       00H             ;6
            .DB       00H             ;7
            .DB       11H             ;8
            .DB       00H             ;9
            .DB       00H             ;10
            .DB       00H             ;11
            .DB       00H             ;12
            .DB       00H             ;13
            .DB       00H             ;14
            .DB       00H             ;15

            .DB       00H             ;16 - Pos. Torque
            .DB       05H             ;17
            .DB       0AH             ;18
            .DB       00H             ;19
            .DB       11H             ;20
            .DB       00H             ;21
            .DB       00H             ;22
            .DB       00H             ;23
            .DB       22H             ;24
            .DB       00H             ;25
            .DB       00H             ;26
            .DB       00H             ;27
            .DB       00H             ;28
            .DB       00H             ;29
            .DB       00H             ;30
            .DB       00H             ;31


;*************************************************
;* Subroutine to determine the direction of the *
;* rotor.                                       *
;*************************************************

            .ORG    100H

GETDIR:     MOVP   A,@A               ;Get table value
            MOV    R2,A               ;Direction did not change
            ANL     A,#30H            ;EPROM = 0, then = 1
            XCH     A,R2

            RET

;*************************************************
;* Wait 500ms for DC Link Capacitor Charging.   *
;*************************************************

WAIT:       MOV    A,#0C0H            ;Ta = Tb = Tc = Td =
            OUTL   P2,A               ;Tac = Tbd = 0 and
                                           ;POS = NEG = 1

            ORL    A,#0AH             ;THETA = 10
```

```
            OUTL    BUS,A

            DIS     I                   ;Disable Interrupts
            DIS     TCNTI

            CLR     F0                  ;Clear user flags
            CLR     F1

            MOV     R0,#0FFH
WAIT1:      MOV     A,#0FFH

WAIT2:      DEC     A
            JNZ     WAIT2
            DJNZ    R0,WAIT1

            MOV     R0,#0FFH
WAIT3:      MOV     A,#0FFH

WAIT4:      DEC     A
            JNZ     WAIT4
            DJNZ    R0,WAIT3

            ORL     BUS,#20
            SEL     RB1                 ;Select register bank 1
            MOV     R0,#1FH             ;Switch table counter
            MOV     R1,#00H             ;Table_1 initially
            SEL     RB0                 ;Select register bank 0
            MOV     R0,#1FH             ;Switch table counter
            MOV     R1,#00H             ;Table1_1 initially
            MOV     R2,#20H             ;Prev. phase is C
            JMP     INI                 ;Start initialization

;*************************************************
;* Table 2 is used to calculate the evidence of *
;* direction change in the rotor.               *
;*                                              *
;*  Input:   bit 0 - A from sensor              *
;*               1 - B from sensor              *
;*               2 - C from sensor              *
;*               3 - D from sensor              *
;*               4 - T0 the lsb of prev. signal *
;*               5 - T1 the msb of prev. signal *
;*               6 - ROTOR prev. dir. of rotor  *
;*                                              *
;*  Output:  bit 0 - CHGDIR, 1 if dir. change   *
;*               1 - Timer Reset                *
;*               2 - 0                          *
;*               3 - 0                          *
;*               4 - T0 the lsb of current sig. *
;*               5 - T1 the msb of current sig. *
;*               6 - 0                          *
;*               7 - 0                          *
;*************************************************

            .ORG    180H

TABLE2:     .DB     00H                 ;0
            .DB     00H                 ;1
            .DB     13H                 ;2
            .DB     00H                 ;3
```

```
.DB     20H             ;4
.DB     00H             ;5
.DB     13H             ;6
.DB     00H             ;7
.DB     30H             ;8
.DB     00H             ;9
.DB     00H             ;10
.DB     00H             ;11
.DB     30H             ;12
.DB     00H             ;13
.DB     00H             ;14
.DB     00H             ;15
.DB     10H             ;16
.DB     02H             ;17
.DB     10H             ;18
.DB     10H             ;19
.DB     21H             ;20
.DB     10H             ;21
.DB     10H             ;22
.DB     10H             ;23
.DB     30H             ;24
.DB     02H             ;25
.DB     10H             ;26
.DB     10H             ;27
.DB     21H             ;28
.DB     10H             ;29
.DB     10H             ;30
.DB     10H             ;31
.DB     20H             ;32
.DB     00H             ;33
.DB     10H             ;34
.DB     10H             ;35
.DB     20H             ;36
.DB     20H             ;37
.DB     20H             ;38
.DB     20H             ;39
.DB     31H             ;40
.DB     31H             ;41
.DB     20H             ;42
.DB     20H             ;43
.DB     20H             ;44
.DB     20H             ;45
.DB     20H             ;46
.DB     20H             ;47
.DB     30H             ;48
.DB     01H             ;49
.DB     10H             ;50
.DB     01H             ;51
.DB     20H             ;52
.DB     30H             ;53
.DB     20H             ;54
.DB     30H             ;55
.DB     30H             ;56
.DB     30H             ;57
.DB     30H             ;58
.DB     30H             ;59
.DB     30H             ;60
.DB     30H             ;61
.DB     30H             ;62
.DB     30H             ;63
```

```
.DB     00H             ;64
.DB     00H             ;65
.DB     12H             ;66
.DB     00H             ;67
.DB     20H             ;68
.DB     00H             ;69
.DB     12H             ;70
.DB     00H             ;71
.DB     31H             ;72
.DB     00H             ;73
.DB     00H             ;74
.DB     00H             ;75
.DB     31H             ;76
.DB     00H             ;77
.DB     00H             ;78
.DB     00H             ;79
.DB     10H             ;80
.DB     03H             ;81
.DB     10H             ;82
.DB     10H             ;83
.DB     20H             ;84
.DB     10H             ;85
.DB     10H             ;86
.DB     10H             ;87
.DB     30H             ;88
.DB     03H             ;89
.DB     10H             ;90
.DB     10H             ;91
.DB     20H             ;92
.DB     10H             ;93
.DB     10H             ;94
.DB     10H             ;95
.DB     20H             ;96
.DB     00H             ;97
.DB     11H             ;98
.DB     11H             ;99
.DB     20H             ;100
.DB     20H             ;101
.DB     20H             ;102
.DB     20H             ;103
.DB     30H             ;104
.DB     30H             ;105
.DB     20H             ;106
.DB     20H             ;107
.DB     20H             ;108
.DB     20H             ;109
.DB     20H             ;110
.DB     20H             ;111
.DB     30H             ;112
.DB     00H             ;113
.DB     10H             ;114
.DB     00H             ;115
.DB     21H             ;116
.DB     30H             ;117
.DB     21H             ;118
.DB     30H             ;119
.DB     30H             ;120
.DB     30H             ;121
.DB     30H             ;122
.DB     30H             ;123
```

```
            .DB     30H             ;124
            .DB     30H             ;125
            .DB     30H             ;126
            .DB     30H             ;127


;*************************************************
;* INI initializes the controller parameters at *
;* startup and it is also called when a change  *
;* in the direction of the rotor has been de-   *
;* tected.  This routine drives the rotor to    *
;* the indexing point according to the torque   *
;* command given.                               *
;*************************************************

            .ORG    200H

INI:        CLR     F1              ;Clear Direction
            DJNZ    R0,NOSW         ;Switch between table1's?
            MOV     A,R1            ;Switch table
            XRL     A,#20H
            MOV     R1,A
            MOV     R0,#01FH        ;Reset switch counter

NOSW:       IN      A,P1            ;Get sensor input
            JT0     POSINI          ;See if Positive or Neg. Torque

            ORL     A,#0C0H         ;Neg. Torque Command
            ORL     A,R1            ;Table selection
            CALL    INISEQ          ;Look-up gate signals
            ORL     A,#40H          ;Make Pos. Speed Fbk.
            OUTL    P2,A            ;Output gate signals

            IN      A,P1            ;Create index for table 2
            ORL     A,R2            ;Prev. phase
            ORL     A,#80H          ;Reverse direction
            CALL    GETDIR          ;Look-up reset timer

            MOV     R3,#88H         ;Currently phase D
            MOV     R5,#40H         ;Speed feedback
            JB1     SYNC            ;At index point
            JMP     INI

POSINI:     ORL     A,#0D0H         ;Show positive torque
            ORL     A,R1            ;Table select
            CALL    INISEQ          ;Get Gate Signal
            ORL     A,#80H          ;Make Neg. Speed Fbk.
            OUTL    P2,A            ;Output gate signals

            IN      A,P1            ;Create index for table 2
            ORL     A,R2            ;Prev. phase
            ORL     A,#0C0H         ;Forward direction
            CALL    GETDIR          ;Look-up table

            MOV     R3,#11H         ;Currently phase A
            MOV     R5,#80H         ;Speed feedback
            CPL     F1              ;Set forward direction
            JB1     SYNC            ;At index point
            JMP     INI
```

```
SYNC:       MOV     R1,#0FFH            ;Load Adv. Cycle Counter
            MOV     R4,#0FFH            ;Loop counter
            MOV     R6,#00H             ;Load Dwell Angle
            MOV     R7,#01H             ;Load Adv. Angle
            CLR     A                   ;Load counter with 0
            MOV     T,A
            STRT    CNT                 ;Start Counter
            JMP     OPER                ;Go to main loop

;*************************************************
;* Subroutine to drive the rotor to the indexing*
;* position.                                     *
;*************************************************


INISEQ:     MOVP    A,@A                ;Get table value

            RET

;*************************************************
;* Table 3 is used to calculate the appropriate *
;* gate drive signals in order to drive the     *
;* rotor to the indexing position according to  *
;* the torque command.  There are two tables to *
;* drive the rotor to the indexing position and *
;* the tables are alternated to avoid a stuck   *
;* rotor.  Table3_1 has the gate signals chosen *
;* close to the aligned position and Table3_2   *
;* has gate signals chosen fartherest from the  *
;* aligned position.                            *
;*                                              *
;*  Input:   bit 0 - A from sensor              *
;*               1 - B from sensor              *
;*               2 - C from sensor              *
;*               3 - D from sensor              *
;*               4 - F0 flag indicating torque  *
;*                   command polarity           *
;*               5 - Table 1 (=0) or 2 (=1)     *
;*               6 - 1                          *
;*               7 - 1                          *
;*                                              *
;*  Output:  bit 0 - Tac                        *
;*               1 - Tbd                        *
;*               2 - Ta                         *
;*               3 - Tb                         *
;*               4 - Tc                         *
;*               5 - Td                         *
;*               6 - 0                          *
;*               7 - 0                          *
;*************************************************

            .ORG    2C0H

TABLE3_1:   .DB     00H                 ;0
            .DB     11H                 ;1
            .DB     22H                 ;2
            .DB     11H                 ;3
            .DB     05H                 ;4
            .DB     00H                 ;5
            .DB     22H                 ;6
```

```
            .DB     00H             ;7
            .DB     0AH             ;8
            .DB     0AH             ;9
            .DB     00H             ;10
            .DB     00H             ;11
            .DB     05H             ;12
            .DB     00H             ;13
            .DB     00H             ;14
            .DB     00H             ;15
            .DB     00H             ;16
            .DB     22H             ;17
            .DB     05H             ;18
            .DB     05H             ;19
            .DB     0AH             ;20
            .DB     00H             ;21
            .DB     0AH             ;22
            .DB     00H             ;23
            .DB     11H             ;24
            .DB     22H             ;25
            .DB     00H             ;26
            .DB     00H             ;27
            .DB     11H             ;28
            .DB     00H             ;29
            .DB     00H             ;30
            .DB     00H             ;31

TABLE3_2:   .DB     00H             ;0
            .DB     0AH             ;1
            .DB     11H             ;2
            .DB     11H             ;3
            .DB     22H             ;4
            .DB     00H             ;5
            .DB     22H             ;6
            .DB     00H             ;7
            .DB     05H             ;8
            .DB     0AH             ;9
            .DB     00H             ;10
            .DB     00H             ;11
            .DB     05H             ;12
            .DB     00H             ;13
            .DB     00H             ;14
            .DB     00H             ;15
            .DB     00H             ;16
            .DB     05H             ;17
            .DB     0AH             ;18
            .DB     05H             ;19
            .DB     11H             ;20
            .DB     00H             ;21
            .DB     0AH             ;22
            .DB     00H             ;23
            .DB     22H             ;24
            .DB     22H             ;25
            .DB     00H             ;26
            .DB     00H             ;27
            .DB     11H             ;28
            .DB     00H             ;29
            .DB     00H             ;30
            .DB     00H             ;31
```

```
;*************************************************
;* Table 4 is used to calculate the necessary    *
;* signals for gate firing and the absolute      *
;* rotor position.                               *
;*                                               *
;*  Input:   bit 0 - COUNT(0)                    *
;*               1 - COUNT(1)                    *
;*               2 - COUNT(2)                    *
;*               3 - COUNT(3)                    *
;*               4 - DWELL ANGLE(0)              *
;*               5 - DWELL ANGLE(1)              *
;*               6 - DWELL ANGLE(2)              *
;*               7 - QUADRANT                    *
;*                   0 -> Motoring               *
;*                   1 -> Regeneration           *
;*                                               *
;*  Output:  bit 0 - THETA(0)                    *
;*               1 - THETA(1)                    *
;*               2 - THETA(2)                    *
;*               3 - THETA(3)                    *
;*               4 - THETA(4)                    *
;*               5 - RESET COUNTER               *
;*               6 - PHASE ON                    *
;*               7 - NEW PHASE                   *
;*************************************************

            .ORG    300H
TABLE4:     .DB     1AH             ;0 - Dwell = 0, Motoring
            .DB     19H             ;1
            .DB     18H             ;2
            .DB     17H             ;3
            .DB     16H             ;4
            .DB     15H             ;5
            .DB     9EH             ;6
            .DB     9DH             ;7
            .DB     9CH             ;8
            .DB     9BH             ;9
            .DB     0BAH            ;10
            .DB     0B9H            ;11
            .DB     0B8H            ;12
            .DB     0B7H            ;13
            .DB     0B6H            ;14
            .DB     0B5H            ;15

            .DB     1AH             ;16 - Dwell = 1, Motoring
            .DB     19H             ;17
            .DB     18H             ;18
            .DB     17H             ;19
            .DB     16H             ;20
            .DB     55H             ;21
            .DB     9EH             ;22
            .DB     9DH             ;23
            .DB     9CH             ;24
            .DB     9BH             ;25
            .DB     0BAH            ;26
            .DB     0B9H            ;27
            .DB     0B8H            ;28
            .DB     0B7H            ;29
            .DB     0B6H            ;30
            .DB     0B5H            ;31
```

```
.DB     1AH             ;32 - Dwell = 2, Motoring
.DB     19H             ;33
.DB     18H             ;34
.DB     17H             ;35
.DB     56H             ;36
.DB     55H             ;37
.DB     9EH             ;38
.DB     9DH             ;39
.DB     9CH             ;40
.DB     9BH             ;41
.DB     0BAH            ;42
.DB     0B9H            ;43
.DB     0B8H            ;44
.DB     0B7H            ;45
.DB     0B6H            ;46
.DB     0B5H            ;47

.DB     1AH             ;48 - Dwell = 3, Motoring
.DB     19H             ;49
.DB     18H             ;50
.DB     57H             ;51
.DB     56H             ;52
.DB     55H             ;53
.DB     9EH             ;54
.DB     9DH             ;55
.DB     9CH             ;56
.DB     9BH             ;57
.DB     0BAH            ;58
.DB     0B9H            ;59
.DB     0B8H            ;60
.DB     0B7H            ;61
.DB     0B6H            ;62
.DB     0B5H            ;63

.DB     1AH             ;64 - Dwell = 4, Motoring
.DB     19H             ;65
.DB     58H             ;66
.DB     57H             ;67
.DB     56H             ;68
.DB     55H             ;69
.DB     9EH             ;70
.DB     9DH             ;71
.DB     9CH             ;72
.DB     9BH             ;73
.DB     0BAH            ;74
.DB     0B9H            ;75
.DB     0B8H            ;76
.DB     0B7H            ;77
.DB     0B6H            ;78
.DB     0B5H            ;79

.DB     1AH             ;80 - Dwell = 5, Motoring
.DB     59H             ;81
.DB     58H             ;82
.DB     57H             ;83
.DB     56H             ;84
.DB     55H             ;85
.DB     9EH             ;86
.DB     9DH             ;87
```

```
.DB     9CH             ;88
.DB     9BH             ;89
.DB     0BAH            ;90
.DB     0B9H            ;91
.DB     0B8H            ;92
.DB     0B7H            ;93
.DB     0B6H            ;94
.DB     0B5H            ;95

.DB     1AH             ;96 - Dwell = 6, Motoring
.DB     59H             ;97
.DB     58H             ;98
.DB     57H             ;99
.DB     56H             ;100
.DB     55H             ;101
.DB     9EH             ;102
.DB     9DH             ;103
.DB     9CH             ;104
.DB     9BH             ;105
.DB     0BAH            ;106
.DB     0B9H            ;107
.DB     0B8H            ;108
.DB     0B7H            ;109
.DB     0B6H            ;110
.DB     0B5H            ;111

.DB     1AH             ;112 - Dwell = 7, Motoring
.DB     59H             ;113
.DB     58H             ;114
.DB     57H             ;115
.DB     56H             ;116
.DB     55H             ;117
.DB     9EH             ;118
.DB     9DH             ;119
.DB     9CH             ;120
.DB     9BH             ;121
.DB     0BAH            ;122
.DB     0B9H            ;123
.DB     0B8H            ;124
.DB     0B7H            ;125
.DB     0B6H            ;126
.DB     0B5H            ;127

.DB     1AH             ;128 - Dwell = 0, Regen.
.DB     1BH             ;129
.DB     1CH             ;130
.DB     1DH             ;131
.DB     1EH             ;132
.DB     1FH             ;133
.DB     96H             ;134
.DB     97H             ;135
.DB     98H             ;136
.DB     99H             ;137
.DB     0BAH            ;138
.DB     0BBH            ;139
.DB     0BCH            ;140
.DB     0BDH            ;141
.DB     0BEH            ;142
.DB     0BFH            ;143
```

```
.DB     1AH             ;144 - Dwell = 1, Regen.
.DB     1BH             ;145
.DB     1CH             ;146
.DB     1DH             ;147
.DB     1EH             ;148
.DB     1FH             ;149
.DB     96H             ;150
.DB     97H             ;151
.DB     98H             ;152
.DB     99H             ;153
.DB     0BAH            ;154
.DB     0BBH            ;155
.DB     0BCH            ;156
.DB     0BDH            ;157
.DB     0BEH            ;158
.DB     0BFH            ;159

.DB     1AH             ;160 - Dwell = 2, Regen.
.DB     1BH             ;161
.DB     1CH             ;162
.DB     1DH             ;163
.DB     1EH             ;164
.DB     1FH             ;165
.DB     96H             ;166
.DB     97H             ;167
.DB     98H             ;168
.DB     99H             ;169
.DB     0BAH            ;170
.DB     0BBH            ;171
.DB     0BCH            ;172
.DB     0BDH            ;173
.DB     0BEH            ;174
.DB     0BFH            ;175

.DB     1AH             ;176 - Dwell = 3, Regen.
.DB     1BH             ;177
.DB     1CH             ;178
.DB     1DH             ;179
.DB     1EH             ;180
.DB     1FH             ;181
.DB     96H             ;182
.DB     97H             ;183
.DB     98H             ;184
.DB     99H             ;185
.DB     0BAH            ;186
.DB     0BBH            ;187
.DB     0BCH            ;188
.DB     0BDH            ;189
.DB     0BEH            ;190
.DB     0BFH            ;191

.DB     1AH             ;192 - Dwell = 4, Regen.
.DB     1BH             ;193
.DB     1CH             ;194
.DB     1DH             ;195
.DB     1EH             ;196
.DB     1FH             ;197
.DB     96H             ;198
.DB     97H             ;199
.DB     98H             ;200
```

```
.DB     99H             ;201
.DB     0BAH            ;202
.DB     0BBH            ;203
.DB     0BCH            ;204
.DB     0BDH            ;205
.DB     0BEH            ;206
.DB     0BFH            ;207

.DB     1AH             ;208 - Dwell = 5, Regen.
.DB     1BH             ;209
.DB     1CH             ;210
.DB     1DH             ;211
.DB     1EH             ;212
.DB     1FH             ;213
.DB     96H             ;214
.DB     97H             ;215
.DB     98H             ;216
.DB     99H             ;217
.DB     0BAH            ;218
.DB     0BBH            ;219
.DB     0BCH            ;220
.DB     0BDH            ;221
.DB     0BEH            ;222
.DB     0BFH            ;223

.DB     1AH             ;224 - Dwell = 6, Regen.
.DB     1BH             ;225
.DB     1CH             ;226
.DB     1DH             ;227
.DB     1EH             ;228
.DB     1FH             ;229
.DB     96H             ;230
.DB     97H             ;231
.DB     98H             ;232
.DB     99H             ;233
.DB     0BAH            ;234
.DB     0BBH            ;235
.DB     0BCH            ;236
.DB     0BDH            ;237
.DB     0BEH            ;238
.DB     0BFH            ;239

.DB     1AH             ;240 - Dwell = 7, Regen.
.DB     1BH             ;241
.DB     1CH             ;242
.DB     1DH             ;243
.DB     1EH             ;244
.DB     1FH             ;245
.DB     96H             ;246
.DB     97H             ;247
.DB     98H             ;248
.DB     99H             ;249
.DB     0BAH            ;250
.DB     0BBH            ;251
.DB     0BCH            ;252
.DB     0BDH            ;253
.DB     0BEH            ;254
.DB     0BFH            ;255

.END
```

References

[1] R. Krishnan and S. Lee, "Effect of Power Factor Correction Circuit on Switched Reluctance Motor Drives for Appliances," IEEE APEC Conference Record, 1994, pp. 83-89.

[2] J. Reinert, J. H. R. Enslin and E. D. Smith, "Digital Control and Optimization of a Rolling Rotor Switched Reluctance Machine," IEEE-IAS Conference Record, 1993, pp. 130-136.

[3] T. J. E. Miller and T. M. Jahns, "A Current-Controlled Switched Reluctance Drive for FHP Applications," Conference on Applied Motion Control, 1986, pp. 109-117.

[4] R. Krishnan and A. S. Bharadwaj, "A Comparative Study of Various Motor Drive Systems for Aircraft Applications," IEEE-IAS Annual Meeting, Conf. Record, Oct. 1991, pp. 252-258.

[5] R. Krishnan, "Criteria for the Comparison of Motor Drive Systems in Motion Control," International Conference on Intelligent Control and Instrumentation, Conf. Proceedings, Feb. 1992, pp. 127-133.

[6] W. F. Ray, R. M. Davis and R. J. Blake, "The Control of SR Motors," Conference on Applied Motion Control, Conf. Record, June 1986, pp. 137-145.

[7] Roche, Egan and Murphy, "Intelligent 8096-Based Multimode Control of a Variable Reluctance Motor Drive," *Proceedings of EPE*, Vol. 2, 1987, pp. 917-922.

[8] C. Elmas and H. Zelaya De La Parra, "A DSP Controlled Switched Reluctance Drive System for Wide Range of Operating Speeds," IEEE-PESC Conference Record, 1992, pp. 844-848.

[9] R. Krishnan, X. Mang and A. S. Bharadwaj, "A Low-Cost SRM Analog Controller," *Electronic MOTORtechnics*, Feb./Mar. 1990, pp. 19-23.

[10] Yang, Panda and Liang, "Experimental Investigation of Feedback Linearization Controller for Switched Reluctance Motor," IEEE-PESC, Conf. Record, June 1996, pp. 1804 - 1810.

[11] Marija Ilic'-Spong, R. Marino, S. M. Peresada and D. G. Taylor, "Feedback Linearization Control of a Switched Reluctance Motor," *IEEE Trans. on Automatic Control*, Vol. AC-32, No. 5, pp. 371-379.

[12] R. Krishnan, "Control of Switched Reluctance Motors," Class Notes, Fall 1995, VPI & SU, Blacksburg, VA.

[13] Brogan, W. L., *Modern Control Theory*, 3rd. ed., Prentice Hall, Englewood Cliffs, 1991.

[14] R. Krishnan, "Electronic Control of Machines," Class Notes, Fall 1995, VPI & SU, Blacksburg, VA.

[15] D. E. Cameron, J. H. Lang and S. D. Umans, "The Origin and Reduction of Acoustic Noise in Doubly Salient Variable Reluctance Motors," *IEEE Trans. on Industrial Applications*, Vol. IA-28, No. 6, May 1987, pp. 234-239.

[16] I. Husain and M. Ehsani, "Torque Ripple Minimization in Switched Reluctance Motor Drives by PWM Current Control," IEEE-APEC Conference Record, 1994, pp. 72-77.

[17] R. Krishnan, A. S. Bharadwaj and P.N. Materu, "Computer aided design of electrical machines for variable speed applications," *IEEE Trans. on Industrial Electronics*, Vol. 36, No. 4, pp. 560-576.

[18] A. R. Oza, R. Krishnan and S. Adkar, "A Microprocessor Control Scheme for Switched Reluctance Motor Drives," IEEE Industrial Electronics Conference, Conf. Proceedings, Nov. 1987, pp. 448-453.

[19] Sember, James W., "Control System for High Speed Switched Reluctance Motor," U.S. Patent # 5,012,171, April 30, 1991.

[20] J. G. O' Donovan, P. J. Roche, R. C. Kavanagh, M. G. Egan and J. M. D. Murphy, "Neural Network Based Torque Ripple Minimization in a Switched Reluctance Motor," IEEE Industrial Electronics Conference, Conf. Proceedings, 1994, pp. 1226-1231.

[21] The Math Works Inc., *The Student Edition of MatlabUser's Guide,* Ver. 4, Prentice Hall, Englewood Cliffs, 1994.

Vita

Terry Wayne Jackson was born September 3, 1971 in Spartanburg, South Carolina. As a freshman, he attended the University of South Carolina. Later, he transferred to Clemson University where he graduated with a Bachelor of Science in Computer Engineering. Upon graduation, he was employed by the U. S. Nuclear Regulatory Commission(USNRC). Funded by the USNRC, he began work on a Master of Science degree in the Fall of 1994.

Terry W. Jackson