Practical Indirect Position Sensing for a Variable Reluctance Motor

BY

WALTER D. HARRIS

S.B., Massachusetts Institute of Technology (1986)

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

at the

Massachusetts Institute of Technology

May 1987

(c) Walter D. Harris

The author hereby grants to MIT permission to reproduce and to distribute copies of this thesis document in whole or in part.

Signature of Author

Department of Electrical Engineering and Computer Science, May 1987

Certified by

Thesis Supervisor

Accepted by

Chairman, Department Committee on Graduate Students

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Archives JUL 0 8 1987
LIBRARIES
Practical Indirect Position Sensing for a Variable Reluctance Motor

Walter D. Harris

Submitted to the Department of Electrical Engineering and Computer Science on May 8, 1987, in partial fulfillment of the requirements for the Degree of Master of Science in Electrical Engineering

ABSTRACT

This thesis describes an algorithm for estimating the position of a variable reluctance motor (VRM) shaft using the known inductance characteristics of idle motor windings. Short excitation pulses are used to estimate inductance and the inductance-position relation is then inverted to determine position. This position information is suitable for closed-loop control.

VRM structure is discussed and the position estimation algorithm is developed. Eddy currents in the VRM, switching noise from its inverter, sensor resolution, and interphase magnetic coupling all pose problems which complicate the implementation of the algorithm. Each problem is individually addressed and a modified algorithm developed. The resulting algorithm is sufficiently simple to run on a low-cost single-chip microcomputer.

As a practical demonstration, the position estimation algorithm is incorporated into the closed-loop speed control of the VRM. This combined estimation and control algorithm is implemented experimentally and its operation is successfully demonstrated. Performance characteristics and limitations of this system are examined and found to be consistent with expectations.

Thesis Supervisor: Jeffrey H. Lang
Title: Associate Professor of Electrical Engineering

- 2 -
Acknowledgements

I would like to thank my thesis supervisor, Professor Jeffrey H. Lang. He was always ready with useful suggestions and encouragement. If I were to become a teacher, I would strive to be like him.

Thanks also go to Karen Walrath for the excellent magnetic flux simulations shown in Chapter 3. They come from her ongoing thesis research.

This thesis was supported by the Copeland Corporation of Sidney, Ohio.
To my parents
# Table of Contents

Abstract 2

Acknowledgements 3

Chapter 1 Introduction 11
   1.1 Problem Statement 11
   1.2 Motivation 11
   1.3 Background 12
   1.4 New Solution 14
   1.5 Thesis Organization 16

Chapter 2 The Variable Reluctance Motor Drive 17
   2.1 Introduction 17
   2.2 The Variable Reluctance Motor 17
   2.3 The Control Algorithm 25
   2.4 The Control System Hardware 26
Chapter 3  Position Detection  

3.1 Basic Algorithm for Position Detection ........ 29  

3.2 Shortcomings of The Basic Algorithm ........ 32  

3.2.1 Eddy Currents .................. 33  

3.2.2 Interphase Coupling .................. 56  

3.2.3 Noise ............................. 65  

3.2.4 Current Sensor Resolution .............. 65  

3.3 Enhancements to Basic Position Sensing  

Algorithm ...................................... 66  

3.3.1 Eddy Currents .................. 66  

3.3.2 Magnetic Coupling .................. 69  

3.3.3 Noise ............................. 72  

3.3.4 Sensor Resolution .................. 73  

3.4 Revised Algorithm For Position Detection ... 74  

Chapter 4  A Practical Application: Speed Control  

4.1 Dynamics of Motor .......................... 77  

4.2 The Controlled Variable: Reciprocal Speed .. 83  

4.3 Structure of Compensator .................. 85  

4.4 Effect of Control Limits .................. 95  

Chapter 5  Implementation  

5.1 Microcontroller Hardware .................. 101  

5.2 The Inverter ............................ 105
List of Figures

Figure 2.1: Three phase VRM .................................. 19
Figure 2.2: Experimental VRM ................................. 20
Figure 2.3: Phase inductance vs. shaft position, three
phase VRM ......................................................... 22
Figure 2.4: Flux Vs. Current for Experimental VRM ... 24
Figure 2.5: Torque Vs. Current and Shaft Position for
Experimental VRM .............................................. 24
Figure 3.1: Theoretical Operation of Position Sensing
Algorithm ......................................................... 32
Figure 3.2: Flux path in experimental VRM with rotor
aligned ............................................................... 35
Figure 3.3: Flux path in experimental VRM with rotor
misaligned .......................................................... 36
Figure 3.4: Magnetic model for eddy current analysis . 40
Figure 3.5: Experimental impedance of VRM winding vs.
Frequency ......................................................... 55
Figure 3.6: Experimental response to sense pulses, with
motor spinning ................................................... 56
Figure 3.7: Winding response to sense pulses with
torque-producing phase driven and undriven. . . . 59

Figure 3.8: Mutual Inductance/Self Inductance, or
Transimpedance/Impedance, vs. Shaft Position . . . 63

Figure 3.9: Mutual Induct. (Transimpedance/Freq.) Vs.
Shaft Angle ........................................ 64

Figure 3.10: Experimental response to sense pulses, with
motor spinning ................................. 67

Figure 3.11: Winding impedance/frequency as a function
of rotor position, at two frequencies (magnitude
only). ............................................. 70

Figure 3.12: Motor operating using final algorithm. . . 76

Figure 4.1: Schematic model of motor, drive, and load
system .............................................. 78

Figure 4.2: Open-loop speed of VRM, plotted versus
chopping current for small, constant dynamometer
torque ........................................ 79

Figure 4.3: Step response of open-loop motor .......... 82

Figure 4.4: Block Diagram of Basic Speed Control Loop .87

Figure 4.5: Root-Locus of Compensated Motor System . 89

Figure 4.6: Block Diagram of Generalized Unity-Gain
Feedback System .................................. 91

Figure 4.7: Simulated and Actual Command Step Response
of Closed-Loop Motor System ...................... 94

Figure 4.8: Block Diagram of Simulated Motor System . 97
Figure 4.9: Simulated and Actual Closed-Loop Large Step Responses, Showing Actuator Saturation .......... 98

Figure 4.10: Compensated system response to square-wave command perturbation ................. 100

Figure 5.1: Block Diagram of Control System .......... 102

Figure 5.2: Control Program Flowchart ............... 109
Chapter 1

Introduction

1.1 Problem Statement

High performance operation of a variable reluctance motor requires accurate position information because the motor is synchronous. This information is typically provided by a specialized sensor such as a shaft encoder or resolver. The presence of the required information in the terminal voltages and currents of the motor, however, permits an alternative, indirect method of position sensing which eliminates the specialized position sensors. By this method, the voltages and currents are measured and filtered to produce useful position information. This thesis addresses the problem of such filtering, particularly at high data rates. This thesis also addresses the problem of incorporating such position information into the closed-loop control of a variable reluctance motor.

1.2 Motivation

The indirect method of position sensing described above is desirable because dedicated direct position sensors can be relatively
expensive, unreliable, and can have high inertia and large size. Replacing these mechanical components with sophisticated algorithms and electronic components reduces system size, weight, and cost, and improves reliability. In particular, using the motor as its own sensor cuts size, weight, and inertia where it is most important, namely at the shaft of the motor. Finally, indirect position sensing allows the use of existing power wires for both actuation and sensing, rather than requiring additional sensor leads. Once again, reliability is improved.

1.3 Background

Other researchers have developed several methods of deriving motor position from electrical terminal voltages and currents. These methods have been applied to many types of motors and range from simple to sophisticated and complex. A simple class of methods is called waveform detection.\textsuperscript{III,VI,IX,XI} Often used with both permanent magnet and variable reluctance motors, these methods monitor the terminal current and voltage of the motor and look for single event extrema or zeroes. They then use known motor models to relate the detected events to position. The events of interest, however, are often a result of back electromotive forces, which prevents the methods from working at zero speed. Similarly, they become unreliable at very low speed, because back electromotive force is small. This problem makes waveform detection impractical for a significant range of speeds. This class of methods is also
susceptible to single errors. Since waveform detection operates on only enough state information to choose one correct position from the periodic operation of the motor, one inaccurate measurement can cause a false determination of position. This problem can not be ignored, and can be solved with recursive filtering.

Acarnley\(^1\) has suggested a second method of sensing rotor position for the variable reluctance motor. In one form, this method also detects a single, known position. Methods of this type involve applying short voltage pulses to the idle windings and measuring the changes in current. From this information, inductance is deduced and then inverted as a function of position. Commutation usually occurs at the position corresponding to minimum inductance, so only one inductance value need be found. When a minimum is detected, the controller initiates commutation. A similar method is developed here.

A third method, developed by Lumsdaine\(^2\), uses an observer to follow the detailed dynamics of the motor itself. With this method, a sophisticated model of the motor dynamics, driven by the same voltages applied to the actual motor and corrected by differences between simulated and measured currents, continuously estimates rotor position. This method produces continuous position estimates with very high accuracy. However, research indicates that inexpensive contemporary microprocessors can not support its
operation for motors operating at high speeds, such as thousands of revolutions per minute.

1.4 New Solution

The method of indirect position sensing developed and implemented here combines the advantages of the simplicity and low computational load of waveform detection with the flexibility of an observer. As suggested by Acarnley¹, this method uses short excitation pulses to sense extrema in the inductance of idle phases. However, it also incorporates enhancements necessary for a practical implementation and demonstration. Specifically, a complex threshold current sensing algorithm seeks the inductance minima of idle phases while simultaneously regulating active phase current. Minimum inductance occurs at or near desired phase commutation positions and is used to drive a simple observer algorithm. Unlike the observer described by Lumsdaine², this simpler observer uses only one measurement per electrical cycle, namely that of the interval between minimum inductance points. If necessary, other shaft positions are found by interpolating between these detected points using the observer. By eliminating the computation associated with continuously tracking all motor currents to generate a high accuracy estimate of position, this algorithm can run much faster. Such a tradeoff of speed and accuracy makes high-speed control possible with inexpensive electronics. In summary, without becoming too computationally complex for real-time use, the present method allows
operation over a wide speed range with reliable position indications at positions of interest.

While this research is directed toward the implementation of indirect position sensing and current control at low computational and monetary cost, such sensing is of little use without closed-loop speed control. Therefore, the indirect position sensing algorithm is embedded in a system which accurately controls both motor speed and phase commutation.

To make the indirect position sensing algorithm work, several problems must be addressed. First, noise presents problems. Some noise problems are solved with hardware, while others are solved within the software of the position sensing algorithm. Second, problems arise from the mutual inductance of different motor phases. The simple algorithm mentioned above for estimating position from the measurements of idle phase self inductances assumes that coupling between windings is not significant. When it is significant, this coupling causes current in a given phase to depend upon the behavior of the other phases, as well as on the voltage applied across it. This can seriously interfere with estimation of self inductance and therefore corrupt position estimates based on inductance estimates. Therefore, this thesis is concerned with understanding and measuring coupling effects and then ensuring that they do not interfere with position estimation. Third, eddy currents in the motor alter the response of phase currents to applied
voltages. This thesis addresses ways of circumventing such effects to produce accurate estimates of inductance and position.

To demonstrate the success of the indirect position sensing algorithm, an experimental system is developed and successfully operated. The system includes a three-phase variable reluctance motor with six stator poles and four rotor poles. The position sensing and controller works well at all speeds up to 6500 rpm and at all power levels within the range of the power electronics. The speed limit of 6500 rpm is directly related to the processing speed of the Intel 8031 microcomputer with which the sensing algorithm and controller are implemented.

1.5 Thesis Organization

This thesis presents issues in the natural order of its research. Chapter Two describes the variable reluctance motor and its excitation. Chapter Three outlines the position sensing problem and presents a basic algorithm. Then, the difficulties with that algorithm are analyzed from a theoretical standpoint. Necessary improvements to the algorithm appear next. The use of the position sensing algorithm for speed control is described in Chapter Four. Chapter Five describes the hardware used in the experimental implementation of the sensing algorithm and controller. Finally, Chapter Six provides a summary, conclusions, and suggestions for future research.
Chapter 2

The Variable Reluctance Motor Drive

2.1 Introduction

The indirect position sensing algorithm developed in this thesis depends intimately on the special nature of the variable reluctance motor (VRM) and its control algorithm. So, this chapter examines the operation of the motor, focusing on the characteristics of it and its control which are important for position sensing.

2.2 The Variable Reluctance Motor

As shown by the example in Figure 2.1, a VRM consists of steel laminations forming a rotor and stator, each with several salient poles. Current through a phase wound around opposing stator poles magnetizes both the rotor and stator poles and produces a torque to align the rotor poles with those excited on the stator. The magnetic effect is always attractive, so the torque produced by the current exciting a particular stator pole is independent of the sign of the current. Continuous rotational torque is produced by exciting the stator poles in sequence, attracting the rotor poles to points
successively further around the stator. Note that this implies that in proper operation, the motor and control signal are synchronous. Thus, the controller must have accurate information concerning the position of the rotor.

Figure 2.1 depicts a typical three-phase VRM similar to that which is used as the experimental system in this thesis. Figure 2.2 shows laminations from the actual experimental motor. In this type of VRM, six poles are placed symmetrically around the stator and wound with alternating magnetic polarity. The windings around opposing poles are connected in magnetic reinforcement to form a single phase. The rotor has four equally spaced poles. The symmetry of the motor means that there is negligible magnetic coupling between the motor phases when considering normal motoring or generating operation.

A VRM generates torque through the spatial variations of the inductance of each phase. For each phase, torque is given by

\begin{equation}
T = \frac{\partial E}{\partial \theta}\,i_1,
\end{equation}

where $T$ is torque, $\theta$ is shaft position, $I$ is phase current, and $E$ is stored magnetic coenergy.\textsuperscript{xiv} In general, the coenergy stored in a phase is given by

\begin{equation}
E = \int \delta \lambda (I, \theta) \, dI,
\end{equation}

where $\lambda$ is the flux linked by the phase. Substitution of (2.2) into (2.1) yields

- 18 -
(2.3) \[ T = \langle \delta \delta \theta \rangle ; \int \delta \lambda (I', \theta) dI', \]

where \( I' \) is a dummy variable of integration representing current.

If the VRM is magnetically linear, that is, in the absence of magnetic saturation, a simple relation exists between \( \lambda \) and \( I \). This relation is
The air gap between the rotor and stator poles is .010 inches and the depth of the motor is 3.5 inches. Fifty turns are wound on each stator pole. The ruler shows actual lamination size, though the air gap is exaggerated slightly for clarity.

Figure 2.2: Experimental VRM
\( \lambda = L(\theta)I. \)

where \( L(\theta) \) is the inductance of the phase. Figure 2.3 shows the idealized \( L(\theta) \) for the VRM shown in Figure 2.1. Substitution of (2.4) into (2.3) gives

\( T = \frac{1}{2} I^2 \frac{dL}{d\theta}, \)

which will be valid for every VRM at sufficiently low currents. From (2.5), it is seen that, if the motor is excited so that each phase has high current during regions of rising inductance and low current at other times, net torque is generated. Figure 2.3 shows the idealized inductance vs. shaft position curves for the type of VRM used in this project.

As the VRM is excited with increasing currents, beyond its magnetically linear range, torque continues to increase. However, the increase is no longer quadratic, as in (2.5). Figure 2.4 shows the flux linkage of the motor studied in this thesis, as a function of current, with position as a parameter. Substitution of this data into (2.3) yields the torque shown in Figure 2.5. Note the magnetically linear operation below approximately five Amperes.

The terminal voltage and current in each phase are related by
Figure 2.3: Phase inductance vs. shaft position, three phase VRM

\[ V = RI + d\lambda/dt, \]

where \( V \) is the terminal voltage, \( I \) is the current, and \( \lambda \) is the flux linked by the phase, as above. \( R \) is the Ohmic resistance of the phase winding. Often, in variable reluctance motors,

\[ RI \ll V \]

In the case of the motor used in this thesis, values for the relevant parameters are
Figure 2.4: Flux Vs. Current for Experimental VRM

\begin{align*}
(2.8) & \quad R = 0.1 \text{ Ohm}, \\
(2.9) & \quad I < 10 \text{ Amperes}, \\
\text{and} \\
(2.10) & \quad V > 100 \text{ Volts}
\end{align*}

This implies that the quantities in (2.7) differ by at least two orders of magnitude. Thus, the experimental VRM can be modelled as having zero winding resistance. With this approximation, (2.6) simplifies to
Figure 2.5: Torque Vs. Current and Shaft Position for Experimental VRM

(2.11) \[ V = \frac{d\lambda}{dt} \]

The mechanical characteristics of the VRM are described by

(2.12) \[ J \frac{d\omega}{dt} = \tau_m - \tau_f, \]

where \( J \) is the rotational inertia of the motor, \( \omega \) is the speed of the shaft, \( \tau_m \) is the torque generated by the motor, and \( \tau_f \) is the torque due to friction and any other load. \( \omega \) is defined as
(2.13) \[ \omega = \frac{d\theta}{dt}, \]

where \( \theta \) is shaft position, as above.

2.3 The Control Algorithm

The control algorithm determines when to build up current in one phase and reduce current in another, as the shaft rotates. In the VRM, this process, known as commutation, typically occurs at or near the point of minimum inductance for each phase (see Figure 2.3). Thus, if phase A is ON, when phase B reaches its minimum inductance, phase A will be turned OFF and phase B will be turned ON. When phase C reaches its minimum, B is turned off and C is fired. Finally, C will be turned off and A will once again be fired. Note that it is not strictly necessary to turn a phase OFF immediately when another is fired. However, doing so allows time to decrease current in the newly OFF phase before it enters a region of decreasing inductance and causes negative torque.

The speed of the motor is indirectly controlled by the level of current in each excited phase, because current controls torque via (2.3) and torque controls speed via (2.12). The control algorithm, therefore, must also determine the level of current in each excited phase to maintain the desired speed. Note that in this implementation, only current level is varied to control torque. Commutation position is held constant.
2.4 The Control System Hardware

To implement such an algorithm, a control system is needed with certain features. Specifically, it must be able to switch currents on and off in several motor phases synchronously with shaft rotation. This entails a switch transition at least twice per phase per electrical cycle, and some form of position sensing. Furthermore, closed loop control of current demands measurement and feedback of current flow and the ability to chop current around a desired level.

In the following sections, a demonstration controller and inverter combination is used which can apply a fixed positive or negative voltage across each stator winding independently. Schematic diagrams of the controller and inverter appear in Appendix A and Appendix B. This combination can measure winding current with a resolution of one part in 256. The demonstration motor normally operates at current levels of about three-fourths of full scale on the current sensors. As will be discussed later, the algorithm is programmed into a microcontroller with appropriate analog sensors attached.

Though the control system can sense current directly and can apply sufficient voltage to control current, it cannot sense shaft position directly. Yet, knowledge of position is vital for motor operation. Therefore, either some form of mechanical position sensing must be added or a method must be used to infer position from other
information. This thesis studies the latter choice. In the next chapter, one such method is introduced.
Chapter 3

Position Detection

3.1 Basic Algorithm for Position Detection

As noted in Figure 2.3, the self-inductance of each motor phase varies smoothly and predictably as a function of shaft position. This suggests that a measurement of inductance can be used to generate an estimate of position. The present position detection algorithm uses such a method. Unfortunately, inverting the position dependence of inductance yields a multivalued relation, meaning that a single measurement of inductance is not sufficient to estimate position. Several positions will exhibit the same value of inductance. By using simultaneous inductance measurements of several phases, however, this problem is eliminated. Further, by taking sequential measurements of the inductance of each phase, better position estimation accuracy can be obtained by proper filtering. An observer is a common means of filtering such sequential measurements.\textsuperscript{ix}

The observer-based method of position estimation is computationally complex. Lumsdaine\textsuperscript{x} describes an even more complex
and accurate observer along these lines. In contrast, this thesis
describes a simplified algorithm based on the observation that the
inductance of each phase passes through a unique minimum point
once per electrical cycle. By detecting this point, the multiple-value
ambiguity of the inductance function is eliminated. For the three-
phase motor of Figure 2.1, three such points are available per
electrical cycle, with four electrical cycles occurring per shaft
revolution. For detection of those three positions, no additional
filtering is needed, assuming accurate inductance measurements. If
additional positions are to be detected, an observer can be used for
interpolation.

As noted in Chapter 2, it is desirable to excite a VRM phase
near its minimum inductance and to remove its excitation near its
maximum inductance. For some variable reluctance motors, this would
cause multiple phases to be excited simultaneously. In this thesis, a
simpler commutation scheme is used instead, with each phase turned
off when the next one is turned on. This commutation is performed
at the minimum inductance of the phase being excited. Since this is
near the maximum inductance position of the phase being turned off,
this commutation method approximates the ideal scheme. Since this
method places commutation at the position most easily detected, it
demonstrates the underlying position detection algorithm quite well.
Different activation and deactivation points can be achieved at
additional computational cost with the use of an observer to perform
interpolation, as suggested above.
Immediately prior to its excitation, a phase is idle, carrying no current. At this time, it can be interrogated with small pulses to determine its inductance without producing significant torque. Because small currents are used, this scheme also avoids the effects of magnetic saturation, which are difficult to model. The measurement of inductance, using short current pulses, and hence indirect position detection, is based on the relation derived by substituting (2.4) into (2.11),

\[
(3.1) \quad V = \frac{d[L(\theta)I]}{dt},
\]

where \( \theta \) is the rotor position, \( L \) is the phase winding inductance, \( V \) is the phase voltage, and \( I \) is the phase current. Note that this relation is valid for the case where magnetic coupling between phases is small. In Chapter 2, it was observed that this is usually true. The effect of nonzero coupling is discussed in a later section of this chapter.

Integrating (3.1) while holding \( V \) constant gives

\[
(3.2) \quad [t_2-t_1] \ V = L(\theta(t_2))I(t_2) - L(\theta(t_1))I(t_1)
\]

Following Figure 3.1, \( I(t_1) \) is zero, so

\[
(3.3) \quad I(t_2) = \frac{[t_2-t_1] \ V}{L(\theta(t_2))}
\]

If the pulse width is constant, then as the inductance reaches its minimum, \( I(t_2) \) reaches a maximum. Since the minimum inductance is
The slope of the rising current is $V/L(\theta)$. When $L(\theta)$ nears its minimum, the current peaks cross a threshold.

**Figure 3.1: Theoretical Operation of Position Sensing Algorithm**

Known, or can be measured, a simple threshold detection of current at the end of each pulse allows detection of the commutation position. When the current rise is large enough, minimum inductance has been reached. Figure 3.1 shows this procedure.
3.2 Shortcomings of The Basic Algorithm

The discussion above addresses the basic principle of the position detection algorithm. However, in practice, several significant problems arise from the existence of independent phenomena. Among these phenomena are eddy currents, mutual inductance, noise, and current sensor resolution. Each is discussed below.

3.2.1 Eddy Currents

Applying pulses of voltage to a phase wound around a steel core will cause both the desired short ramps of current in the phase and undesired currents in the steel core. As the initial rise of current in the phase establishes flux in the core, secondary eddy currents arise in the core which temporarily shield the flux from passing through it. These eddy currents, by removing the low reluctance steel from the flux path, cause more current to be required for a given flux linkage. This reduces the apparent inductance of the phase and the current rises far more rapidly than originally expected. After a short time, the eddy currents relax due to core resistance and the apparent inductance tends to its low-frequency value.

The present algorithm for position detection depends on the position-varying response of a phase to short voltage pulses.
Therefore, it is important to know the effects of eddy currents. To do so, first consider the fundamental magnetic operation of the motor. Figures 3.2 and 3.3 show the flux path in the experimental VRM for the aligned and misaligned cases, respectively. These Figures show only one quarter of the motor, which was shown in full in Figure 2.2. However, because of the symmetry of the motor, that section can be used to describe the behavior of the remainder. In the Figures, the top stator pole is excited. Magnetic flux must pass through that stator pole, through the rotor, through the matching stator pole on the opposite side of the motor, and back around the outside of the stator. In doing so, the flux twice passes across the small air gap between the rotor and the stator, and through the steel laminations of both parts.

Magnetic reluctance, the analog of electrical resistance, can be thought of as the resistance of a block of material to magnetic flux passing through it.\textsuperscript{iv} By definition,

\begin{equation}
\lambda = N^2 I / (R_{\text{air}} + R_{\text{steel}}),
\end{equation}

where \( R_{\text{air}} \) and \( R_{\text{steel}} \) are the reluctances of the air and steel portions of the path followed by flux driven by the stator winding in Figure (3.2) and \( N \) is the number of winding turns. In the steady state, that is, at zero frequency, reluctance is given by
Figure 3.2: Flux path in experimental VRM with rotor aligned

\[ R = \frac{1}{A \mu}, \]

where \( A \) is the cross-sectional area of the block of material, \( l \) is the length of the flux path passing through it, and \( \mu \) is its permeability. At high frequencies, however, eddy currents arise in blocks composed of conducting material, increasing the apparent reluctance of those blocks.

Since the air gap in the VRM is non-conducting, its reluctance is a function only of its geometry, which in turn is governed by the distance of the rotor pole from the stator pole. The apparent reluctance of the conductive steel, though not dependent on shaft position, rises greatly at high frequencies. Expressing these dependencies, (3.4) becomes
(3.6) \[ \lambda(s) = \frac{N^2 I(s)}{[R_{\text{air}}(\theta) + R_{\text{steel}}(s)]}, \]

where \( s \) is a frequency domain variable. At low frequencies, \( R_{\text{steel}} \) is essentially constant, so (3.6) reduces to

(3.7) \[ \lambda(s) = \frac{N^2 I(s)}{[R_{\text{air}}(\theta) + R_{\text{steel}}(0)]} \]

This is similar to the familiar

(3.8) \[ \lambda(s) = L(\theta) I(s), \]

However, at high frequencies, when eddy currents act to oppose flux through the steel,
\( R_{\text{steel}} \gg R_{\text{air}} \)

so (3.6) becomes

\[ \lambda(s) = N^2 I(s)/R_{\text{steel}}(s) \]

Here, the effect of the air gap is negligible, due to the high reluctance of the steel. To determine the behavior of the flux in the motor in this case, it is convenient to consider a motor model which contains no air gap. Now, the flux path consists only of steel, which is simplified further into the form of a ring. The ring is considered to have a large enough diameter that its curvature is small. As shown by (3.10), the frequency-dependence of the flux in this model will represent that of the unsimplified motor in the case of high frequency excitation. A cross section of the simplified high-frequency motor model is shown in Figure 3.4; each lamination can be treated in isolation.

The effects of eddy currents are examined using the model of Figure 3.4. To do so, begin with Maxwell's equations in their magneto-quasistatic form\textsuperscript{iv}, which are
(3.11) \[ \text{curl}(H) = J \]
(3.12) \[ \text{div}(B) = 0 \]
(3.13) \[ \text{curl}(E) = -\partial B/\partial t \]

Here, \( H \) is the magnetic field, \( B \) is the magnetic flux density, \( E \) is the electric field, and \( J \) is the current density. The relevant material constitutive relations are

(3.14) \[ J = \sigma E \]
(3.15) \[ B = \mu H \]

The former is Ohm's law and the latter is a magnetization law. The value for \( \mu \) depends on the material being magnetized. Note that the magnetization law is a linear model, and is not valid for saturated steel. However, during the sense pulses, the flux density levels are low, keeping the important steel unsaturated.

Eddy currents evolve according to the magnetic diffusion equation, which follows from Maxwell’s equations. Substituting (3.14) into (3.11) and taking the curl of both sides yields

(3.16) \[ \text{curl(curl}(H)) = \text{curl}(\sigma E) \]

Using the vector identity

(3.17) \[ \text{curl(curl}(X)) = \text{grad(div}(X)) - \text{grad}^2(X), \]

Equation (3.16) becomes
(3.18) \[ \text{grad(div}(\mathbf{H})) - \text{grad}^2(\mathbf{H}) = \text{curl}(\sigma \mathbf{E}) \]

Substituting (3.12) into the left side and (3.13) into the right side of (3.18) gives

(3.19) \[ \text{grad}^2(\mathbf{H}) = -\sigma \mu \frac{\partial \mathbf{H}}{\partial t} \]

which is the magnetic diffusion equation.

The magnetic diffusion equation is applied to this model to find a time domain expression for the flux linkage evolution in the lamination. The lamination model is effectively infinite in the \( z \)-axis, so the magnetic field \( \mathbf{H} \) consists only of the \( z \)-directed component. Similarly, \( D \gg \omega \), so the model also appears infinite in the \( y \) direction and solutions can have no \( y \) dependence. Thus, a magnetic field solution of the form

(3.20) \[ \mathbf{H} = z' \times \mathbf{H}_z(x,t) \]

is assumed, where \( z' \) is a unit vector in the positive \( z \) direction.

Substituting (3.20) into (3.19) yields

(3.21) \[ \text{grad}^2(\mathbf{H}) = \frac{\partial^2 \mathbf{H}_z(x,t)}{\partial x^2} \]

(3.22) \[ = \mu \sigma \frac{\partial \mathbf{H}_z(x,t)}{\partial t} \]

Consider the application of a step of external magnetic field from \( \mathbf{H}(t) = 0 \) to \( \mathbf{H}(t) = \mathbf{H}_0 \) at \( t = 0 \), using the windings. Thus, the
The model is shown in cross-section with windings and from the side without windings. The bar shown at left is closed on itself to form the ring shown at right. Note that the inner and outer diameters of the ring are assumed near enough that its curvature is small. Also, $D \gg W$.

Figure 3.4: Magnetic model for eddy current analysis

magnetic field in the air space between the windings and the steel steps to the values

\[(3.23) \quad H_s(x<0, t) = H_0\]

and
(3.24) \[ H_z(x > W, t) = H_0 \]

This establishes the boundary conditions for the solution of the magnetic diffusion equation within the steel lamination, namely for \( 0 < x < W \). At \( t = 0 \), however, the steel lamination responds with eddy currents on its surface which shield out the magnetic field. The appropriate initial condition is therefore

(3.25) \[ H_z(0 < x < W, t=0) = 0 \]

within the laminations. Note this implies that at \( t = 0 \), a step of \( H \) occurs at the surface of the lamination. This step is supported by a surface eddy current.

To solve the diffusion equation within the lamination subject to the preceding initial and boundary conditions, it is convenient to separate temporal and spatial variables by expressing \( H_z(x, t) \) as

(3.26) \[ H_z(x, t) = X(x) \ast T(t) + H_0, \]

where \( X(x) \) and \( T(t) \) are as yet unknown functions. The amplitude of the applied external magnetic field, \( H_0 \), is chosen as the constant term to match the requirement that at \( t = \infty \), when the eddy currents have died away, the magnetic field inside the steel should equal that in the surrounding air.

Substitution of (3.26) into the diffusion equation (3.19) gives
(3.27) \[ T \left( \frac{\partial^2 X}{\partial x^2} \right) = \mu \sigma X \left( \frac{\partial T}{\partial t} \right) \]

Rearranging terms yields

(3.28) \[ (\mu \sigma)^{-1} \frac{\partial^2 X}{\partial x^2} = (T)^{-1} \frac{\partial T}{\partial t} \]

Since both sides of the equality depend entirely on different independent variables, they must both be constant to maintain equality. Thus,

(3.29) \[ (\mu \sigma)^{-1} \frac{\partial^2 X}{\partial x^2} = -\tau^{-1} \]

and

(3.30) \[ (T)^{-1} \frac{\partial T}{\partial x} = -\tau^{-1} \]

where \( \tau \) is an as yet unknown time constant. Solving (3.30) for \( T(t) \) gives

(3.31) \[ T(t) \propto \exp(-t/\tau) \]

Solving (3.29) for \( X(x) \) gives

(3.32) \[ \frac{\partial^2 X}{\partial x^2} = -\mu \sigma / \tau X, \]

which has solutions of the form

(3.33) \[ X = \{ \sin[(\mu \sigma / \tau)^{1/2} x] \text{ or } \cos[(\mu \sigma / \tau)^{1/2} x] \} \]

The boundary condition (3.23), substituted into (3.26) at \( x=0 \) requires that
(3.34) \[ X(0)T(t) = 0 \]

Only the sine solutions of (3.33) satisfy this requirement. At \( x=W \), (3.24), substituted into (3.26) requires

(3.35) \[ X(W)T(t) = 0 \]

For the sine solutions of (3.33), this implies

(3.36) \[ \sin([\mu_\sigma/\tau]^{1/2}W) = 0, \]

or

(3.37) \[ (\mu_\sigma/\tau)^{1/2} = n\pi/W \quad (n = 1,2,3...) \]

giving

(3.38) \[ \tau_n = \mu_\sigma W^2/(n^2\pi^2) \]

The family of solutions for \( X_n(x) \), (3.33), is thus

(3.39) \[ X_n(x) = \sin((\mu_\sigma/\tau_n)^{1/2} x), \]

with a corresponding family for \( T(t) \) from (3.31) of

(3.40) \[ T(t) \propto \exp(-t/\tau) \]

where in (3.39) and (3.40),
\[ \tau_n = \frac{\mu \sigma W^2}{n^2 \pi^2} \quad (n = 1, 2, 3, \ldots) \]

A properly weighted sum of these sine expressions will satisfy the initial condition over \(0 < x < W\). This gives a combined solution for \(H_z(x,t)\) of

\[ H_z(x,t) = H_0 + \sum_{n=1}^{\infty} H_n \sin\left(\frac{n \pi x}{W}\right) \exp\left(-\frac{t}{\tau_n}\right), \]

To satisfy the initial condition that all fields are zero inside the steel, the constant term must cancel the summation term. Thus

\[ -H_0 = \sum_{n=1}^{\infty} H_n \sin\left(\frac{n \pi x}{W}\right) \]

To solve for the Fourier coefficients, \(H_n\), multiply both sides of (3.43) by a new sine term and integrate to yield

\[ \int_{-W}^{W} \left( \sum_{n=1}^{\infty} H_n \sin\left(\frac{n \pi x}{W}\right) \right) \sin\left(\frac{m \pi x}{W}\right) \, dx = \int_{-W}^{W} -H_0 \sin\left(\frac{m \pi x}{W}\right) \, dx \]

Thus,

\[ H_m \ast \left(\frac{W}{2}\right) = \int_{-W}^{W} -H_0 \sin\left(\frac{m \pi x}{W}\right) \, dx \]

\[ = -H_0 \left(\frac{W}{m \pi}\right) \left[-\cos\left(\frac{m \pi x}{W}\right)\right]_{-W}^{W} \]

\[ = \{0, \text{ for } m \text{ even; or } -2H_0 W/m \pi, \text{ for } m \text{ odd} \} \]

so that

- 44 -
(3.48) \[ H_m = \begin{cases} 0, & \text{for } m \text{ even}, \\ -4H_0/m\pi, & \text{for } m \text{ odd} \end{cases} \]

The summation using these coefficients describes the magnetic field inside the lamination

(3.49) \[ H = H_0 + \sum_{\omega=0} \left[ -4H_0/(2p+1)\pi \right] \sin((2p+1)\pi x/W) \exp[-(2p+1)^2\pi^2 t/(\mu\sigma)^{-1}W^{-2}] \]

This is the desired result.

In an effort to simplify this equation and then derive a relation between phase terminal voltage and current, consider the frequency domain relationship between the internal and external magnetic fields.\textsuperscript{xiii} The transform of (3.49), the response to a step of external field at \( t = 0 \), is

(3.50) \[ H_0(s) = H_0/s - \sum_{\omega=0} \left[ 4H_0/(2p+1)\pi \right] \sin((2p+1)\pi x/W) / (s+\pi^2(2p+1)^2(\mu\sigma)^{-1}W^{-2}) \]

Differentiating this in time and normalizing yields

(3.51) \[ \frac{H(s)}{H_0(s)} = 1 - \sum_{\omega=0} \left[ 4/(2p+1)\pi \right] s \sin((2p+1)\pi x/W) / [s+\pi^2(2p+1)^2(\mu\sigma)^{-1}W^{-2}] \]

Next, applying the definition of magnetic flux linkage to the phase winding gives
\begin{equation}
\lambda(s) = \int \Psi N \mu H(s) \, D \, dx
\end{equation}

This integral must be evaluated over a cross-sectional slice of the ring of Figure 3.4. This consists of two areas: the air-filled regions between the winding and the steel and the region containing the steel itself. Assuming the windings are placed tightly around the poles, the area of the air-filled region will be negligible. Thus, only the contribution of the flux through cross-sectional area of the steel need be considered in (3.52). The magnetic field inside the steel varies as given by (3.49) and the cross-sectional area is \( A_s \). Thus, the flux integral of (3.52) evaluates to

\begin{equation}
\lambda(s) = NH_0(s)\mu A_s \left[ 1 - \sum_{p=0}^{\infty} \left( \frac{4}{(2p+1)\pi} \right) \left( \frac{1}{2p+1}\pi \right) \right] \\
\left[ -\cos((2p+1)\pi W/W) + \cos(0) \right] \left[ s/(s+\pi^2(2p+1)^2(\mu\sigma)^{-1}W^{-2}) \right]
\end{equation}

This simplifies to

\begin{equation}
\frac{\lambda(s)}{H_0(s)} = N\mu A_s \left[ 1 - \sum_{p=0}^{\infty} \left( \frac{8}{(2p+1)^2\pi^2} \right) \left[ s/(s+\pi^2(2p+1)^2(\mu\sigma)^{-1}W^{-2}) \right] \right]
\end{equation}

A contour drawn on the surface of the ring shaped steel lamination, in the direction perpendicular to the windings, will enclose a surface pierced only by the current in the winding, not by any eddy currents. Therefore, such a contour may be used to evaluate \( H_0 \), the field adjacent to the steel, which drives the field inside the steel. Ampere's law, Equation 3.11, in its magnetooquasistatic integral form is
where \( C \) is a closed contour and \( S \) is any surface bounded by that contour. Applying this to a contour drawn on the ring surface gives

\[
(3.56) \quad H_0(s) \cdot l_s = N I(s),
\]

where \( l_s \) is the length of the flux path around the ring. Substituting (3.56) into (3.54) yields

\[
(3.57) \quad \frac{\lambda(s)}{I(s)} = N^2 l_s^{-1} \mu A_s \left[ 1 - \sum \mathcal{P} \left( \frac{8}{(2p+1)^2 \pi^2} \right) \left( \frac{s}{s+\pi^2(2p+1)^2(\mu_0)^{-1}W^2} \right) \right]
\]

In the low frequency limit, when eddy currents must be negligible, this expression simplifies to

\[
(3.58) \quad \frac{\lambda(s)}{I(s)} = N^2 l_s^{-1} \mu A_s,
\]

By definition, this quantity in the low-frequency limit is the DC inductance

\[
(3.59) \quad L_0 = N^2 l_s^{-1} \mu A_s
\]

Thus, the effect of the eddy currents, shown by (3.57), is to reduce the effective inductance from its DC value noted in (3.59). This corresponds to increasing the effective reluctance of the steel at high frequencies.

With the relationship between flux and current given by (3.57), one need only differentiate (3.57) with respect to time to get a
relationship between voltage and current. Proceeding in the frequency domain,

\[(3.60) \quad V(s) = \lambda(s) \cdot s\]

\[(3.61) \quad = I(s) \left[ \lambda(s)/I(s) \right] \cdot s\]

so that

\[(3.62) \quad V(s)/I(s) = Z(s)\]

where

\[(3.63) \quad Z(s) = sN^2\lambda_1^{-1} \mu A_0[1 - \Sigma \Phi [8/(2p+1)^2\pi^2] \left[ s/(s+\pi^2(2p+1)^2(\mu\sigma)^{-1}W^{-2}) \right] ]\]

As noted above, because of the frequency dependence of the expression for \(Z(s)\), the system does not respond as an ideal inductor at all frequencies. Two characteristics of the response are of interest, namely, the frequency at which the system ceases acting like an inductor and its behavior above that frequency. The characteristics must be interpreted in the time domain in which the controller will operate.

The question of the first critical frequency can be answered by finding the location of the first pole in the summation in (3.63). Lower numbered terms break first and have larger magnitudes as well. In fact, because of the dependence on \((2p+1)^2\), the magnitude of the second term is one ninth of the magnitude of the first term.
The complete series can therefore be reasonably approximated by its first term. The break frequency in the series occurs at

\[ \omega_0 = \frac{\pi^2}{\mu \sigma W^2} \]  

(3.64)

For the steel used in this project, approximate parameter values are

\[ \mu = 2000 \quad \mu_0 \]

(3.65)

\[ \sigma = 2 \times 10^6 \quad \text{N}^{-1}\text{m}^{-1}, \]

(3.66)

and

\[ W = 0.018 \text{ in} = 4.57 \times 10^{-4}\text{m} \]

(3.67)

Substituting these parameters into (3.64) gives

\[ \omega_0 = 9.4 \times 10^3 \text{ rad/s} = 1.5 \times 10^3 \text{ Hz} \]

(3.68)

Similarly,

\[ \omega_1 = 13.5 \times 10^3 \text{ Hz} \]

(3.69)

and

\[ \omega_2 = 37.5 \times 10^3 \text{ Hz} \]

(3.70)

By neglecting the high order terms in the summation of (3.63), \( Z(s) \) can be simplified into the form
\[(3.71)\] \[Z(s) \equiv [b + a_\omega s/(s + \omega_0)]s\]

where \(b\) and \(a_\omega\) are constants and \(\omega_0\) is the first break frequency. The values for the constants come from (3.63). They are

\[(3.72)\] \[b = N^2 \mu A_s / l_s\]

and

\[(3.73)\] \[a_\omega = -8b / \pi^2\]

Note that \(b\) is the low frequency inductance,

\[(3.74)\] \[b = L_0\]

Thus, \(Z(s)\) can be rewritten

\[(3.75)\] \[Z(s) = s(bs + b\omega_0 + a_\omega s)/(s + \omega_0)\]

\[(3.76)\] \[= s[(b + \omega_0)s + b\omega_0]/(s + \omega_0)\]

\[(3.77)\] \[= [b + a_\omega] s (s + b\omega_0[b + a_\omega]^{-1}) / (s + \omega_0)\]

Since \(a_\omega\) is negative, the pole is at a lower frequency than the zero. So, for frequencies less than \((b\omega_0[b + a_\omega]^{-1})\), this is

\[(3.78)\] \[Z(s) \equiv (b + a_\omega) s (b\omega_0[b + a_\omega]^{-1}) / (s + \omega_0)\]

\[(3.79)\] \[= b\omega_0 s/(s + \omega_0)\]

For the present purpose, the important response of the system is the response of the current to a step of voltage. The
transfer function from voltage to current is the admittance, $Y(s)$, defined by

\[(3.80) \quad Y(s) = 1/Z(s),\]

giving

\[(3.81) \quad I(s) = Y(s) V(s)\]

Substitution of (3.79) into (3.80) yields

\[(3.82) \quad Y(s) = (s+\omega_0)/b\omega_0 s.\]

This is an expression for the voltage to current transfer function of the system, valid through the low end of the frequency range where eddy currents have significant effects.

To complete the analysis, (3.82) is returned to the time domain and solved for the response to a step in voltage. For this purpose, $Y(s)$ is rewritten as

\[(3.83) \quad Y(s) = [1/b\omega_0] + [1/bs]\]

Applying the inverse Laplace transform$^{xiii}$ to (3.83) gives the time-domain impulse response
\[ y(t) = \left[ \frac{1}{b} \right] \left[ \delta(t)/\omega_0 + u(t) \right], \]

where \( u(t) \) represents a unit step at \( t = 0 \) and \( \delta(t) \) represents a unit impulse at \( t = 0 \). Integrating this impulse response gives the step response

\[ y(t) = \left[ \frac{1}{b\omega_0} \right] \left[ u(t) + \frac{1}{\omega_0} u(t) \right], \]

namely a ramp superimposed on a step. In the limit of large \( t \), this tends to

\[ y(t) = tu(t)/b \]

Substituting (3.74), this becomes

\[ y(t) = tu(t)/L_0, \]

which again confirms that in the long term, the system behaves like a normal inductor.

Equation 3.85 was derived for the case in which the flux path never crosses an air gap. However, (3.6) shows that in a more realistic model of a motor, with an air gap, that there will be a smooth transition between the high frequency, short time interval case, which behaves as if there were no air gap, and the low frequency, long time interval case, which behaves like a normal inductor. Over short time intervals, (3.63) and (3.85) will accurately describe the system, in the frequency and time domains, respectively, because the reluctance of the steel will completely dominate the
reluctance of the air gap, making the effect of the air gap negligible. Over long time intervals, the reluctance of the steel to the inductance tends to a constant, since eddy currents die out, and the reluctance of the air has no time dependence, so the combined effect of the air gap and the steel must be a constant reluctance. Thus, the air gap will have no effect in short time intervals and will alter the inductance in long time intervals. The complete step response of the system is therefore a step and ramp, followed by a smooth transition of the ramp to the low-frequency ramp of the ideal inductor, with an air gap.

As noted in (3.6), the reluctance of the steel has no dependence on shaft position. However, the reluctance of the air gap is strongly dependent on position, since the geometry of the air gap varies, as shown in Figures (3.2) and (3.3). It is the effect of this dependence on low-frequency inductance which causes the large inductance variations described in Chapter 2, which are used to generate torque. Fortunately, since the eddy currents do decay to zero, those low-frequency inductance values are accurate, though one must wait some time to measure them.

Several experiments on the experimental VRM confirm the previous results. The impedance of the motor was measured by applying a sine wave of voltage to a phase and recording the response. Figure 3.5 shows the resulting impedance. The phase
impedance demonstrates a roll-off centering at $3 \times 10^3$ Hz, which is very close to that of (3.68). It is also worth noting that the phase of the impedance tends to forty-five degrees. The infinite sum in (3.63) also tends to forty-five degrees, so the experimental results tend to confirm that the model for $Z(s)$.

A time-domain experiment also confirms the eddy-current theory. Applying a square pulse of voltage to the winding yields a current step followed by a ramp, when the shaft is aligned so that the steel dominates the flux path, as in Figure (3.2). This position is referred to as maximum alignment. Notably, when the shaft is positioned so that the air gap dominates the flux path, the current waveform looks like a pure ramp. Upon close inspection, the step still exists, however the large scale shape of the response is dominated by the much larger slope of the low inductance of the air-gap dominated magnetic circuit. These experiments are shown in Figure 3.6. From Figure 3.6, it is also possible to note the curvature of the short time response, as the step becomes a ramp, which is dominated by $\omega_1$. Taking $\omega_0$ to be $3 \times 10^3$ Hz, from Figure 3.5, (3.63) predicts $\omega_1$ to be approximately $27 \times 10^3$ Hz or $81 \times 10^3$ rad/sec, so that $\tau_1$ is predicted to be 12 $\mu$sec. This is very close to the relaxation time observed in Figure 3.6.
Figure 3.5: Experimental impedance of VRM winding vs. Frequency
Figure 3.6: Experimental response to sense pulses, with motor spinning

Supply voltage: 35V  Speed: 1300 RPM

--- > Time, 10 us/div
3.2.2 Interphase Coupling

The original development of the position detection algorithm, we assumed that the sense voltage pulse train was the only input to the motor. However, during realistic operation, large currents must be present in at least one phase to produce torque. This leads to the problem of coupling between the torque-producing phases, with high currents being switched by high voltages, and the position-detecting phases. Thus, the phenomenon of magnetic coupling via mutual inductance must be considered.

To first order, there should be no magnetic coupling between phases. Each phase is divided into two equal windings, placed on poles on opposite sides of the stator. By symmetry, equal amounts of flux enter the rotor from one winding and leave the rotor to enter the other winding section. In the limiting case of zero steel reluctance, all of the entering flux passes across the rotor and makes up the exiting flux. In that case, there is no coupling between phases, since flux from one phase passes only through that phase. Coupling occurs because the steel does not have zero reluctance. Some flux passes back across the air gaps into other windings and through the air space between windings, causing the phases to have noticable mutual inductance. As eddy currents act to increase the apparent steel reluctance, this becomes a more significant factor.
Many designers assume that magnetic coupling is minimal in variable reluctance motors; for example, see [X]. Often, it is indeed negligible, particularly when one is concerned only with torque production and related characteristics. However, at high pulse rates, eddy currents increase the apparent reluctance of the rotor steel to the point where coupled flux is important to a position detection algorithm. The problem is yet more severe because position sensing is performed at low current levels, so that high currents in other phases coupling through low mutual inductance can still induce significant currents in the sensing phase. Figure 3.7 shows an actual winding response to small sense pulses when another winding is being chopped at high current. The coupling is quite evident.

To understand the effect of mutual inductance, consider the electrical operation of the motor in the absence of saturation. The three phases are treated simultaneously using a motor inductance matrix; for clarity, underlining denotes matrix quantities. \( \theta \) denotes shaft position, with zero chosen arbitrarily at the maximum self-inductance point of phase A. The matrix \( L(\theta) \) is defined by:

\[
\begin{align*}
\Lambda &= L(\theta)I
\end{align*}
\]

where \( \lambda \) is now a vector of phase flux linkages and \( I \) is now a vector of phase currents. Assuming negligible winding resistance as was shown in (2.11), phase voltage is by definition
The effect of interphase coupling is clearly visible here, where the single sense pulse which occurs while the torque-producing phase is driven produces a dramatically smaller current.

(Note that the current sensor measuring $I_A$ is disabled when its associated phase is undriven.)

**Figure 3.7:** Winding response to sense pulses with torque-producing phase driven and undriven.

\[(3.89) \quad V = \frac{d\lambda}{dt},\]

(3.89) expands to become
(3.90) \[ V = L \frac{dI}{dt} + I \frac{dL}{dt} \]

Under the normal operating conditions of the VRM studied in this thesis, \( I \frac{dL}{dt} \) is negligible compared to \( L \frac{dI}{dt} \). Therefore, (3.90) simplifies to

(3.91) \[ V = L \frac{dI}{dt} \]

Without loss of generality, consider the case in which phase A is being driven at high current, phase B is being sensed, and phase C carries no current. The quantity of concern is the flux through the sense phase B. The relevant terms of (3.91) are

(3.92) \[ \lambda_B = L_{AB}I_A + L_{BB}I_B + L_{BC}I_C, \]

where \( L_{AB}, L_{BB}, \) and \( L_{BC} \) are mutual inductances. Since \( I_C \) is defined to be zero here, only the first two terms contribute. Rearranging (3.92) gives

(3.93) \[ I_B = L^{-1}_{BB} (\lambda_B - L_{AB}I_A), \]

which shows that the current in the sense phase B is a function of the flux in that phase and of the current through the driven phase. Next, differentiation of (3.93) and substitution of (3.89) yields

(3.94) \[ \frac{dI_B}{dt} = L^{-1}_{BB} (V_B - L_{AB}dI_A/dt) \]

Assume here that \( dI_A/dt \) is approximately a function of \( V_A \) alone. Furthermore, since the inverter used for experiments in this thesis
has only one supply voltage, all applied voltages have equal magnitude \( V \). These conditions yield

\[
(3.95) \quad \frac{dI_A}{dt} = L_{-1AA} V_A
\]

and

\[
(3.96) \quad V = V_B = [+/-] V_A.
\]

Consequently,

\[
(3.97) \quad \frac{dI_B}{dt} = L_{-1BB} (V [+/-] L_{AB} L_{-1AA} V)
\]

\[
(3.98) \quad = L_{1BB} V (1 [+/-] L_{AB} L_{-1AA})
\]

The important issue in (3.98) is the relative magnitudes of coupling term and the self-inductance term. The coupling term introduces error. A typical measurement at the point of commutation between phases \( A \) and \( B \) is

\[
(3.99) \quad L_{AB}(\theta)/L_{AA}(\theta) = .07
\]

The commutation point, in this case, is defined as the point of minimum \( L_{BB} \). Inductances were measured by applying 3000 Hz sine waves of current at one terminal pair and observing coupled voltage at the other terminal pair. The resulting seven percent error is significant, especially since it is at a moderate frequency and can be either positive or negative, introducing an uncertainty of fourteen percent into self inductance measurements.
Eddy currents cause additional phase shift between current and voltage, so the V/I characteristics are again not those of pure inductors. For this reason, it is helpful to think of the ratio of "Mutual Inductance / Self Inductance" as "Transimpedance / Impedance" -- ideal inductors cause a constant phase shift, whereas these do not. The phase of the coupling can be neglected, however, because the magnitude of the coupling accurately indicates the worst-case error.

Figure 3.8 shows the magnitude of the mutual inductance as a function of rotor position. Figure 3.9 shows the magnitude of the mutual inductance divided by the magnitude of the self inductance (transimpedance/frequency) as a function of rotor position. As noted in (3.98), it is this ratio that determines the effect of magnetic coupling on self inductance measurements. Notably, the effect is large, varies with shaft position, and has a distinct minimum at the point of commutation. Fortunately, though, this last observation places the minimum error at the most critical measurement point.
Figure 3.8: Mutual Inductance/Self Inductance, or Transimpedance/Impedance, vs. Shaft Position
Mutual Inductance Vs. Shaft Angle (Exp.)

Magnitude (I = 150 mA)

Figure 3.9: Mutual Induct. (Transimpedance/Freq.) Vs. Shaft Angle

Figure 64: Mutual Inductance Angle (I = 150 mA)
3.2.3 Noise

Switching transients caused by the inverter can interfere with analog-to-digital conversions and can cause false threshold detections. Either of these events disrupts the position detection algorithm. In the worst case, switching transients can stop the microcontroller program. Fortunately, the latter problem can be avoided by careful grounding. The solutions to the former problem appear in the next section.

3.2.4 Current Sensor Resolution

Motor torque is generated by one or more phases carrying high currents. Simultaneously, the sense phase carries small current pulses for inductance measurement. Consequently, there is a problem with the dynamic range and resolution of the current sensor. In practice, the same sensor on a given phase will likely measure the high current, for regulation, and the low current, for position detection. Since all current measurements are digitized by some element of the hardware, it is possible that the sense pulses of current may be lost in the graininess of the digitization of the torque-producing currents. That is, the sense currents may be only
as large as the smallest measurable current deviation. This problem is discussed in the next section.

3.3 Enhancements to Basic Position Sensing Algorithm

In the previous section, several problems with the basic position detection algorithm were presented. To accomodate these problems, the basic position detection algorithm must be modified. In the subsections which follow, the required improvements are presented and discussed.

3.3.1 Eddy Currents

As determined above, eddy currents in the rotor and stator steel greatly alter the current response of the motor phases to short pulses of voltage. In particular, it was shown in Section 3.2.1 that the time domain current response was approximately a step, dominated by the eddy currents, followed by a ramp, dominated by the low-frequency phase inductance. Finer detail is also evident from Figure 3.10.

The low frequency coil inductance is well known, as shown in Figure 2.3. The effects of eddy currents can be regarded as modifications to this behavior. While the general frequency-domain character of these modifications is also known, the changes in
Steel/air composition of the VRM magnetic circuit as the rotor goes between minimum and maximum alignment make the exact effects of the eddy currents difficult to predict analytically and difficult to estimate rapidly. However, in the time domain, after the eddy-
currents and their effects have decayed, the ramp always has the slope of the low-frequency inductance. This fact allows a simple algorithm modification which allows one essentially to ignore the eddy current effects henceforth. This modification is to use a two point measurement to determine the slope of the ramp and thus eliminate the effect of the step. This modification works well. Figure 3.10 shows sense pulses and currents in the experimental motor. The initial step is clearly visible when the rotor is at its maximum alignment position, where steel dominates the flux path. At the minimum alignment position, the step is barely visible. In fact, it is still present, but since the air gap dominates the flux path, the ramp is much larger and the step is less obvious in comparison. At all positions, the position detector uses a two point measurement to measure the slope of the ramp. It takes the first data point approximately 15 microseconds after applying the voltage. The voltage is applied for a total of 62 microseconds with the second data point taken just before it is removed. Pulses are applied at 125 microsecond intervals, yielding a 50% duty cycle.

Figure 3.3.1 shows apparent winding self inductance, |Z(s)/s|, for two shaft positions, as a function of excitation frequency. Notably, apparent self inductance tends to a constant, independent of position, as excitation frequency is increased. This is an effect of the eddy currents. Unfortunately, to improve the temporal resolution of the position detection algorithm, it is desirable to take many
inductance measurements during each electrical cycle. However, as Figure 3.3.1 shows, increasing the frequency of the sense pulses will also tend to reduce their effectiveness from another standpoint, since it reduces the position dependence of inductance on which they rely. A compromise between the two effects leads to the choice of 8 kHz as the sense pulse rate for the experimental motor. This is reasonably high, compared to the electrical cycle time of the experimental motor, yet as Figure 3.3.1 shows, the position dependence of inductance is still easily measurable at such a frequency.

3.3.2 Magnetic Coupling

As noted above, interphase coupling due to mutual inductance is a problem because it alters phase current during sense pulses. Because this coupling is observed to become minor around the rotor angles where commutation must occur, one method of avoiding coupling problems is to begin sensing only when commutation is expected, which means shortly before the inductance minima. The obvious difficulty with this approach is that one must first know when minima will occur in order to detect them. Fortunately, magnetic coupling is not severe enough to prevent low speed operation of the motor using the pulse method of position detection. Thus, one can start up the system with sense pulses occurring
Figure 3.11: Winding impedance/frequency as a function of rotor position, at two frequencies (magnitude only).

throughout the shaft rotation. Later, when changes in speed due to motor or load torques are small compared to total speed, so commutation intervals are relatively constant, one can operate a predictor and refrain from sensing during intervals predicted to
contain no inductance minima. In this way, one can avoid the intervals suffering from high interphase coupling.

Since this control system must run at high speed, the predictor must be simple. A zeroeth-order system is implemented in the experimental system. Here, each phase transition interval is predicted to be exactly as long as the measured length of the preceding one. No sense pulses are issued during the first half of the predicted interval.

A hidden benefit of a predictor is that it allows the controller to do something else when sensing is not needed. In the experimental system, a single microcontroller handles both driven-phase current regulation and sense-phase pulsing. Its computational resources are limited, so at high supply voltages, it has trouble chopping the driven phase rapidly enough to maintain a smooth current profile. In particular, issuing sense pulses distracts the controller from chopping, so the driven phase current may change undesirably. However, after predicting the interval length and setting a timer, the controller need not perform the sensing function and can chop the driven phase rapidly. Since the first part of the transition interval is the time of low inductance for the driven phase and thus the time of rapid current change, the controller devotes itself exclusively to current regulation when that task is most difficult.
3.3.3 Noise

Care in designing the control system electronics will reduce electrical noise and its effects on sensor readings. Many bypass capacitors, short signal paths, wide circuit board trace spacing, and a large ground plane limit noise. However, when many Amperes of current are being switched within centimeters of low-voltage control circuitry, noise may still interfere.

Since physical remedies may fail, the control algorithm itself must tolerate some amount of electrical noise. Most of that noise is caused by switches changing state while carrying high current. Such noise can be predicted, however, because it occurs only when the controller commands a switch to change state. Here, the fact that a single microcontroller performs all functions helps. To eliminate effects of switching noise, the control program need only avoid setting switches while it is measuring an analog quantity, such as sense pulses or chopping currents. This method works well and is used in the experimental system.

The control system also tolerates external noise. Small noise signals cause small variations in commutation times which do not affect motor operation noticeably. Strong interference, such as a dip in supply voltage, may cause the controller to miss a transition
entirely. As long as this occurs at high speed, it may not affect operation. If the motor can rotate one electrical cycle on its own inertia, the algorithm will detect the commutation point when it comes around a second time, and the phases will advance properly once more.

Under normal conditions, the controller never misses a transition. However, tests show that it will in fact recover from doing so. Finally, if the system is required to operate from a varying supply voltage, this particular problem can be eliminated by sensing the supply voltage and adjusting the transition threshold appropriately.

3.3.4 Sensor Resolution

To avoid producing counter-productive torque, the controller must make the sense pulses as small in amplitude as possible. However, as noted above, small pulses may be undetectable on any sensor calibrated to measure large, torque-oriented currents. Two methods may be used to get meaningful numbers for sense pulse response. The first is to use separate sensors for sense pulse currents and torque-producing currents. Thus, the sense pulses can generate currents measuring a significant percentage of full scale on their sensors, without having made the drive currents unreadable.
Alternatively, if a single sensor is used to measure all currents in a phase, or even in the entire motor, pulse sizes must be selected to remain detectable. The experimental system uses the latter approach. Since the microcontroller represents all analog quantities with eight bit integers, sense currents can be only as small as about ten percent of full scale before their internal representation loses too much resolution. This means that analog quantities reach an integer representation of approximately 25. Counter-productive torque, when sense pulses are being issued, is then much less than approximately ten percent of productive torque.

One can operate the control system with even smaller sense pulses, but with a loss of flexibility. Obviously, if currents due to sense pulses are represented by integers with values of 9, 10, and 11, changes of less than ten percent are undetectable. Thus, the controller cannot vary the internal thresholds by such small, now unmeasurable amounts dynamically. However, the designer can, simply by varying either the pulse time or the supply voltage, a change which might well be decided in advance of operation, at the product design stage. Thus, as long as one can set a reliable threshold, low sensor resolution can be tolerated. The experimental system is operated using this approach with current changes registering only ten units on the sensors.
3.4 Revised Algorithm For Position Detection

Figure 3.12 shows the final position-detection algorithm in operation, combined with a current regulator. The algorithm follows the following pattern. After it determines that the sensed phase should be switched on, it allows that phase to continue increasing its current. The previously driven phase is turned off, and a timer is started; this timer is preset to one half of the length of the previous commutation interval. While the timer is running, the controller devotes itself to regulating the current in the newly driven phase by chopping. Once the timer expires, the controller begins issuing sense pulses to the next phase, while continuing current regulation. For each sense pulse, the controller turns on the drive voltage to the sensed phase, waits a short amount of time, measures the current flowing at that time, waits a longer amount of time, and measures the current again. If the current increases enough, the controller decides the time has come to commutate and the process repeats. If not, the controller allows the sense current to relax before the phase is interrogated again. During the relaxation interval, the controller chops the current in the main phase, if necessary. This algorithm is repeated for each phase, as the motor rotates.
Time, 1 ms/div

1500 RPM, $V_s = 35$ volts, $I_{avg}=2.5$ A

Note the delay between phase B firing and the sense pulses for phase C. Also, the current sensors used here read zero when their associated phases are turned off.

Figure 3.12: Motor operating using final algorithm.
Chapter 4

A Practical Application: Speed Control

To be useful, a position detection algorithm must be suitable for use in an actual motor control system. To demonstrate the suitability of the present algorithm, this chapter describes such a system. Specifically, the position sensing algorithm developed above is used in a closed-loop speed controller.

4.1 Dynamics of Motor

In order to design a controller for the experimental motor, it is first necessary to understand the dynamics of the complete motor system, including controller, motor, and load. Figure 4.1 shows a suitable model of the experimental closed-loop system. An electronic drive circuit regulates the currents of the motor phases. The motor generates torque according to the applied current, which turns the shaft. A dynamometer provides a known, adjustable load to the rotating shaft. Information concerning the currents in the motor is fed back to the drive controller.

A suitable equation for the behavior of the motor and load is
Figure 4.1: Schematic model of motor, drive, and load system

\[
T_{\text{motor}}(I) = J \frac{d\omega}{dt} + F\omega + T_{\text{load}},
\]

where \( \omega \) is the rotational speed, \( J \) is the motor inertia, \( F \) is a constant of viscous frictional drag, \( T_{\text{load}} \) represents the drag torque of the dynamometer and of all other non-viscous loads on the system, and \( I \) is the motor current. The goal is to control \( \omega \) via \( I \).

To simplify the analysis, note that although \( T_{\text{motor}} \) is a function of the shaft position and of the instantaneous current in each of the motor phases, fixed commutation angles, such as those determined in Chapter 2 allow approximating instantaneous torque by average torque, dependent only on the chopping level of the current; this is expressed by (4.1). To determine this dependence, \( T_{\text{motor}}(I) \) is balanced in steady state, yielding \( \omega \) as a function of \( I \) as shown in Figure 4.2. Note that throughout this thesis, the dynamometer is set to a small, constant torque. At all of the resulting equilibria,
Figure 4.2: Open-loop speed of VRM, plotted versus chopping current for small, constant dynamometer torque

\[(4.2) \quad T_{\text{motor}}(I) = Fw + T_{\text{lead}},\]

where \( T_{\text{lead}} \) is a small, constant value. At any equilibrium, define

\[(4.3) \quad I = I_0 + \delta I,\]

and
\[ (4.4) \quad \omega = \omega_0 + \delta \omega. \]

Now, for perturbations around the equilibrium, (4.1) becomes

\[ (4.5) \quad T_{\text{motor}}(I_0) + \delta I \frac{dT_{\text{motor}}}{dI}|_{I_0} = J \frac{d\delta \omega}{dt} + F \omega_0 + F \delta \omega \]

Subtracting (4.2) from both sides of (4.5) yields

\[ (4.6) \quad \delta I \frac{dT_{\text{motor}}}{dI}|_{I_0} = J \frac{d\omega}{dt} + F \delta \omega \]

Steady state operation allows simplifying (4.6) to

\[ (4.7) \quad \delta I \frac{dT_{\text{motor}}}{dI}|_{I_0} = F \delta \omega \]

Figure 4.2 shows that a small perturbation in current will produce a linearly related change in speed, for a wide range of operating points. This implies that \( dT_{\text{motor}}/dI \) is constant over a wide range of operating speeds. Thus, by defining

\[ (4.8) \quad K_m = \left[ \frac{dT_{\text{motor}}}{dI} \right]|_{I_0}, \]

for operation in that linear range one can write (4.6) as

\[ (4.9) \quad \delta I K_m = J \frac{d\delta \omega}{dt} + F \delta \omega \]

Since electronic switches respond much faster that the mechanical time constants of the system, inverter dynamics can be ignored. Therefore, (4.9) represents all of the important dynamics of the motor system.
Applying the Laplace transform to (4.9) yields

\[(4.10) \quad \delta I(s) \ K_m = sJ \ \delta \omega(s) + F \ \delta \omega\]

and the transfer function of the system is thus

\[(4.11) \quad M(s) = \frac{\delta \omega(s)}{\delta I(s)} = \frac{K_m}{(sJ + F)}\]

This simple model indicates that the mechanical friction and inertia of the motor interact to give it a first-order response at its equilibrium. Figure 4.3 shows the observed open-loop response of the motor to a step change in current. As predicted by the first-order system model, the motor rises exponentially to its new setpoint and exhibits no measurable overshoot. From the figure, the time constant of the open-loop system can be measured as approximately ten seconds.
Figure 4.3: Step response of open-loop motor


For both charts, the horizontal scale is one second per major division. Note that the 63% risetime is approximately 10.0 seconds.
Additional poles will appear in the response due to such factors as control delay and electrical response times, but they will appear at much higher frequencies than the fractional-Hertz mechanical pole just described. Thus, the model above is valid for control up to at least several Hertz. Control delay and control magnitude restrictions will also limit loop bandwidth, so higher frequency models of the motor are not necessary.

4.2 The Controlled Variable: Reciprocal Speed

The raw speed data is actually measured in the form of a commutation period rather than a rotational frequency. Thus, to control speed directly would require inversion of the period. Division, however, uses excessive processor time and would slow down the speed control loop, particularly since the experimental system uses an eight-bit processor, with no multi-precision divide instruction. Regulation of period, rather than frequency, then, saves this time. As shown below, small signal period changes are directly proportional to small signal speed changes, except for a sign difference.

Let $P$ be the period of one complete rotation of the motor shaft. Expressing $P$ as a function of $\omega$ in radians/second gives
(4.12) \[ P(\omega) = 2\pi/\omega, \]

Differentiating (4.12) with respect to \( \omega \) gives

(4.13) \[ \frac{dP}{d\omega} = -2\pi \omega^{-2} \]

By the chain rule of calculus,

(4.14) \[ dP = d\omega(dP/d\omega) \]

Substituting (4.13) into (4.14) yields

(4.15) \[ dP = d\omega(-2\pi \omega^{-2}) \]

This shows that incremental changes in period are indeed proportional to incremental changes in rotational speed and that the constant of proportionality is \((-2\pi \omega^{-2})\). Control gains are adjusted to account for this scaling and produce the desired transfer function, from motor speed error to output current.

Approximations based on (4.15) will be valid for deviations from equilibrium of \( \delta \omega \) which are small compared to \( \omega \), as shown below. From (4.12),

- 84 -
\[ P(\omega + \delta \omega) - P(\omega) = 2\pi [1/((\omega + \delta \omega) - 1/\omega) \]

\[ = 2\pi (\omega - [\omega + \delta \omega]) / \omega [\omega + \delta \omega] \]

\[ = -2\pi \delta \omega / \omega [\omega + \delta \omega] \]

\[ = -2\pi \delta \omega / \omega^2, \text{ for } \delta \omega << \omega, \]

which was to be shown.

4.3 Structure of Compensator

The drive current to the motor is chosen to be a combination of terms proportional to the speed error and to its integral. Henceforth, the algorithm for determining this current level is called the compensator, to distinguish it from the controller hardware used for its implementation. Figure 4.4 shows a block diagram of the control signal generation. This compensation scheme is often referred to as the proportional-integral, or simply PI, system. The integral term ensures that the system sustains no steady-state error, while the proportional term provides it with faster response.

A high frequency, digital low-pass filter limits the frequency response of the control signal. This filter is not shown in Figure 4.4. It eliminates jitter due to asymmetry in the motor phases. Any pole which is slightly different from the others in the motor causes a longer or shorter commutation interval and thus a sudden change in
Figure 4.4: Block Diagram of Basic Speed Control Loop

apparent speed. Without a low-pass filter, this would immediately cause a change in control current. The filter smooths out such jumps.

Together, the proportional, integral, and low-pass terms result in the full loop transfer function \( A(s) \), given by

\[
A(s) = [K_is^{-1} + K_p] \left[ \omega_f(s+\omega_f)^{-1} \right] M(s),
\]

where \( \omega_f \) is the frequency of the low-pass filter pole, \( K_i \) and \( K_p \) are the compensator integral and proportional term gains, respectively, and \( M(s) \) is the single-pole current-to-speed transfer function of (4.11). The low-pass filter is included only to prevent sensor jitter from causing audible jitter in the output, so it can be neglected.
when considering relatively slow motor speed response. Therefore, for the purpose of control, $A(s)$ reduces to

$$
(4.21) \quad A(s) = (K_is^{-1} + K_p) M(s).
$$

Substituting (4.11) for $M(s)$, the open-loop transfer function of the combined motor and compensator is

$$
(4.22) \quad A(s) = (K_is^{-1} + K_p) \frac{K_m\omega_m}{(s+\omega_m)} \]
$$

This can also be written as

$$
(4.23) \quad A(s) = K_i\frac{(K_pK_i^{-1}s + 1)/s}{[K_m\omega_m/(s+\omega_m)]}.
$$

This reveals that the compensator introduces a pole at the origin and a zero at $s = K_iK_p^{-1}$.

Analysis of $A(s)$ provides insight into the behavior of the system when the loop is closed as in Figure 4.4. The root-locus method of feedback system analysis allows prediction of the closed-loop system poles. Specifically, if the gains $K_i$ and $K_p$ in the compensator are low, the poles are near their open-loop locations. As the gains are increased, maintaining a constant ratio between $K_i$ and $K_p$ to avoid moving the compensator zero, the closed-loop poles move toward the open-loop system zeroes, which remain stationary. The reason for avoiding moving the compensator zero is discussed later. Figure 4.5 shows the progression of the closed-loop poles.
Figure 4.5: Root-Locus of Compensated Motor System

The root-locus shows that with the PI compensator, the system poles move so that there is a pole near the low-frequency zero introduced by the compensator and another pole higher in frequency, both on the real axis of the complex plane. For moderate gains, the system has a complex pole pair. For high gains, however, the zero approximately cancels the low-frequency pole, so the higher-frequency pole dominates. Once again, the system shows first-order
behavior, though with much faster response than it had with no feedback control.

The compensator zero seen in (4.23) is placed by choosing the ratio \( K_l/K_p \) to be near the open-loop motor pole. As shown by the root-locus analysis above, this means that the zero will tend to cancel the motor pole when the control loop is closed and a moderate amount of loop gain is applied. The zero is not placed exactly at the motor pole because the pole is at a quite low frequency. If the zero were placed there, though the pole would almost be cancelled even for low gains, imperfect cancellation would result in a pole-zero pair, or doublet, at a low frequency instead of a high one. As is shown later in this section, a low-frequency doublet is undesirable. A compromise between having a very low frequency pole-zero pair and requiring high gain for cancellation suggests placing the compensator zero a factor of ten higher than the open-loop motor pole.

An algebraic analysis of the system transfer function gives the exact location of the closed-loop poles and zeroes. Given the general form of a unity-gain feedback system as shown in Figure 4.6, the transfer function from the error \( e(s) \) to the output \( Y(s) \) is

\[
(4.24) \quad y(s) = A(s)e(s).
\]

The error can be written as
(4.25) \[ e(s) = u(s) - A(s)e(s), \]

So,

(4.26) \[ e(s)[1 + A(s)] = u(s) \]

(4.27) \[ e(s) = u(s) [1 + A(s)]^{-1} \]

Finally,

(4.28) \[ y(s) = A(s) [1 + A(s)]^{-1}u(s) \]

The poles of the first term in (4.28) are the poles of the open-loop transfer function. However, those will be cancelled by their inverses generated by the second term. Thus, the poles of the system are thus only the poles of the second term, given by the roots of the desensitivity function,
The desensitivity function can be used to solve for the poles of the closed-loop motor system described above. From (4.22),

\[
S^{-1}(s) = [1 + A(s)].
\]

\[
S^{-1}(s) = 1 + (K_s^{-1} + K_p) [K_m \omega_m/(s+\omega_m)]
\]

\[
= 1 + (K_ps + K_i) (K_m \omega_m) / s(s+\omega_m)
\]

\[
= [s(s+\omega_m) + (K_ps + K_i)K_m \omega_m] / [s(s+\omega_m)]
\]

\[
= [s^2 + \omega_m s + K_m \omega_ms + K_m \omega_m K_i] / [s(s+\omega_m)]
\]

\[
= S^{-1}(s) = [s^2 + \omega_m(1 + K_m K_p)s + K_m \omega_m K_i] / [s(s+\omega_m)]
\]

The zeroes of \( S^{-1}(s) \) and the poles of the closed-loop system are thus the roots of the numerator of (4.34), given by the quadratic formula

\[
p_n = [-b (+/-) (b^2 - 4ac)^{1/2}]/2a
\]

where \( a, b, \) and \( c \) are given by

\[
a = 1
\]

\[
b = \omega_m(1 + K_m K_p)
\]

\[
c = (K_m \omega_m)K_i.
\]

With the motor pole known to be at 0.1 radians/second, the design decision to put the compensator zero, \( z_{comp} \), a decade above
the motor pole yields an approximate zero frequency of 1 radian/second. As can be seen from (4.22), this implies

\[(4.39) \quad z_{\text{comp}} = K_i/K_p = 1,\]

or

\[(4.40) \quad K_i = K_p.\]

The second degree of freedom is the low-frequency loop gain of the system, \(K_i\). Due to the above decision, this will also be the value of \(K_p\). Experiments and simulations show that to control the system in response to moderate disturbances, the control gain cannot be higher than

\[(4.41) \quad K_{i,\text{max}} = 0.2\]

without generating unreasonably high control signals. So, this value is used as a compromise between fast theoretical response and real-world control limits.

Figure 4.7 shows the compensated motor responding to a commanded speed increase. The response time is improved by two orders of magnitude from that of the uncompensated system. Figure 4.7 also shows some overshoot in the step response. This is due to the presence of an integrator in the control loop. While the motor is accelerating to the new, higher speed, the integral of the speed error increases, due to the positive error during this time.
Consequently, when the motor reaches its target speed, the integrator has too high a value, so the motor must run over the setpoint for a short time, to reduce the integral of the error to the value needed to hold the motor at the desired speed. One way to avoid this effect is to make the integral gain so low, $K_i \ll 1$, that the effect of the initial speed jump is small. However, this means that the system will take a long time to reach its exact target speed.

An explanation in root-locus terms of the overshoot is that at the gains chosen, the system pole does not exactly reach the compensator zero. Thus, there is a pole-zero doublet on the real axis, with the pole at the higher frequency. The Laplace transform of such a doublet is

\begin{equation}
H_4(s) = \frac{s+z}{s+p},
\end{equation}

where $z$ and $p$ are the radian frequencies of the zero and pole, respectively. The step response is therefore $(s+z)/[(s+p)s]$. The inverse Laplace transform of this response is

\begin{equation}
H(t) = \frac{z}{p} + \exp(-pt)(1 - \frac{z}{p})
\end{equation}

If, as in this case, the pole frequency, $p$, is only slightly greater than the zero frequency, $z$, then (4.43) reduces to

\begin{equation}
H(t) \approx 1 + \exp(-pt)[(p-z)/p].
\end{equation}

This explains the long tail effect of pole-zero doublets. Since the magnitude of the overshoot, or undershoot, if $z>p$, is proportional to
Steps from 2000 to 2200 rpm. Horizontal scales are top, .3 sec/div; bottom, 1 sec/div.

Figure 4.7: Simulated and Actual Command Step Response of Closed-Loop Motor System

the difference in the pole and zero frequencies, it is desirable to
have the two close together, which can be achieved by having high loop gain in the control system. However, as noted, infinite loop gains are impossible due to control signal limits, among other reasons. Also, (4.44) explains why the compensator zero was not placed exactly as low in frequency as the open-loop motor pole. Had that been done, the tail would have died off extremely slowly, causing the motor to deviate from its target speed for a correspondingly long time. The numbers chosen, therefore, represent a good compromise among all of these effects.

4.4 Effect of Control Limits

If there were no limits on the size of the drive torque of the motor, one could achieve excellent performance. However, in reality, it is not possible to apply a drive current outside of certain maximum and minimum bounds. In the demonstration system, the control algorithm cannot apply a current greater than six amperes and cannot apply currents in such a way as to create a negative torque. The upper limit is a function of the particular hardware used and the lower limit is a software limitation. Both limits could be made less strict, but they serve to illustrate how drive signal limits affect system response time.

To investigate the effects of drive signal limits, a simulation of the motor and drive system was created. This system, shown in Figure 4.8 models the nonlinear dynamics of the motor shown in Figure 4.2, the upper and lower limits on drive current just
mentioned, and the linear dynamics of the motor and compensator described in (4.23).

Figure 4.9 shows both the simulated and actual motors responding to a large command step and reaching the upper bound on drive current. The results are quite similar. In both cases, to follow the large command step, the PI controller attempts to apply a very large initial current to the motor. However, real-world factors intervene. Specifically, this current is limited at the maximum output of the inverter. Consequently, the risetime of the motor speed is greatly degraded from that shown in response to small steps, which do not cause the actuator to saturate. An example of such a small step was shown in Figure 4.7. As noted, the risetime there was approximately three times shorter. The risetime in Figure 4.7 corresponds to the predicted time constant of Equation (4.34), whereas the results in the nonlinear, saturated case of Figure 4.9 depart markedly from the linear model, as expected.
Figure 4.8: Block Diagram of Simulated Motor System
Figure 4.9: Simulated and Actual Closed-Loop Large Step Responses, Showing Actuator Saturation

Top: Simulated results, 1900-2500 RPM step
Bottom: Actual results, 1900-2500 RPM step
Because the experimental system is not programmed for dynamic braking, the only retarding torque which can be applied to the experimental motor is that of friction. This is in no way a consequence of the algorithms used. It is merely a result of the decision to program the controller to excite phases only during regions of positive torque. Negative torque could be implemented simply by changing the programmed commutation angle. However, since this was not done, the system provides an example of the effects of different upper and lower control torque limits. The lower limit of zero torque greatly limits the response of the system to negative-going steps in commanded speed. Figure 4.10 illustrates this problem by contrasting the responses to equal positive and negative command steps. For the positive step, the motor drive stays within the linear, unbounded drive current range. For the negative step, however, the motor current quickly limits at its lower bound of zero. Thus, the rising response is much quicker than the falling response and the only shape of the rising step is as predicted by the linear model of the complete control system.
Figure 4.10: Compensated system response to square-wave command perturbation

1 sec/div, 2000 to 2200 RPM
Chapter 5

Implementation

This chapter describes implementation aspects of the experimental demonstration of the position detection algorithm with integrated speed control. As shown in Figure 4.1, the experimental system can be divided into four blocks: the microcontroller and support hardware, the inverter, the motor, and the dynamometer used to apply a load. The microcontroller, its software, and the inverter are described in this chapter.

5.1 Microcontroller Hardware

The experimental controller system consists of an Intel 8031 microcontroller and associated peripheral devices. A detailed schematic is presented in Appendix A. As shown in the block diagram of Figure 5.1, the peripheral devices allow the microcontroller to adjust eight switches, sense twelve analog quantities, read a pulse-type shaft encoder, and communicate with other computers. Other memory devices allow the microcontroller to store data and to execute previously stored programs.
Figure 5.1: Block Diagram of Control System
The controller works with an IBM personal computer to form a complete development station. A monitor program for the microcontroller allows control programs to be composed using the personal computer, then transferred into the temporary memory of the microcontroller and executed. Thus, the high-level capabilities of the personal computer direct the control functions of the microcontrol system.

It is important to note that the experimental system described here is specifically designed to be versatile, not small or inexpensive. It is oriented towards rapidly developing algorithms. The system design permits testing of widely differing control methods without changing equipment. Of course, a production controller could be optimized to run a single algorithm, well known by construction time. Such an optimized system would be much smaller and cheaper than the one here, but would not useful for research.

The 8031 microcontroller combines many functions at a low cost. On this one chip reside a control-oriented eight-bit microprocessor, data memory, 32 bits of parallel input or output, two timers, and a serial port. More self-contained versions are available with permanent program memory (ROM), which provides a convenient upgrade path for production versions. However, the prototype uses alterable program memory, so programs can be developed and tested rapidly. Specifically, this off-chip program memory allows the
microcontroller to execute a monitor program and to accept downloaded control programs. A part of it can also be used to store large amounts of experimental data for later analysis. The monitor program resides permanently in a 2764 erasable, programmable, read-only memory (EPROM). The temporary memory is a 6264 static random-access memory (RAM).

The experimental system can measure analog quantities with two subsystems. The first converts analog quantities directly into integers. The Analog Devices AD7828 analog-to-digital (A/D) converter has eight inputs and converts each to an eight-bit word, for a maximum reading of 255. The converter is quite fast, requiring only 2.5 microseconds for a conversion. The second analog input subsystem works indirectly, comparing a variable threshold with external analog quantities. When the threshold is reached, the processor is interrupted. Four quantities can be sensed in this manner. An Analog Devices AD7226 digital-to-analog (D/A) converter generates the thresholds. This subsystem also uses eight-bit words.

To control the solid-state switches in the inverter, the system has eight output bits, driven by a 74HC373 latch. Eight light-emitting diodes allow the operator to observe the output states during operation and allow error codes to be flashed.

Although the control system implements an indirect position sensing algorithm, during development, a reference is needed for
comparison and to prove that the algorithm is working. Thus, three bits of input are provided. On the schematic diagram, they are designated "Home", "Quadrature", and "Main", for use with a pulse-type shaft encoder.

To transfer data between the personal computer and the controller, the controller includes a serial port, with RS-232 signal levels. New programs are loaded using this port and data can be transferred to the host computer for analysis after a test. Monitor software simplifies this data transfer by allowing block data transfer in standard file formats. Programs thus downloaded can be executed with a simple command. The monitor is completely described in reference XII.

5.2 The Inverter

A separate power electronics circuit, shown completely in Appendix B, applies voltage to each of the motor windings following the commands of the microcontroller. The experimental system uses a simple three-phase driver circuit consisting of FETs and flyback diodes. One terminal of each winding is connected to a DC supply, and the other to the input of a FET and the input of a diode. The FET connects the second winding terminal to ground, causing current to flow. When the FET is switched off, winding inductance drives current through the flyback diode. However, a capacitor on the other side of the diode provides a back voltage which reduces the
current as the capacitor charges. In this way, both positive and negative voltage can be applied to a motor winding using only one FET and one diode. Energy recovered from the windings when they are switched off is dissipated as heat in a resistor, connected across the capacitor. This energy could be recovered by a DC-to-DC converter and dumped back into the supply, but the experimental system does not include that circuitry.

5.3 Microcontroller Software

The control program has two functional levels, the foreground and the interrupt routines. Timer overflows cause interrupts which initiate all time-critical functions. These include issuing position detection pulses, commutating the motor phases, and regulating drive current. The foreground, in contrast, performs each of its functions as often processor load permits. Foreground functions include processing the reading from the front panel potentiometer into a desired motor speed and then converting it to a commutation period. Then, the foreground routine calculates the desired output current according to the speed control algorithm developed in Chapter 4. This calculation consists of subtracting the measured commutation period from the desired period, passing it through the proportional-integral control system shown in Figure 4.8, and storing the desired drive current level. There is also an alarm interrupt routine, which shuts down the system and flashes an alarm code if any current or voltage exceeds its preset threshold.
Otherwise, it has no role in normal motor operation. Figure 5.2 shows a flowchart of the control program. A complete listing appears in Appendix C.
Interrupt:

Short term timer expires

Slow mode?

Y

1/2 interval over?

N

Uses long timer →

N

Pulse ON next phase

Time for commutation?

N

Turn Sense Phase OFF

Y

- 108 -
Figure 5.2: Control Program Flowchart

(Continued on following pages)
The block above labeled, "Time for commutation?" expands into the following sequence:

1. Implicit Delay
2. A/D conversion of current
3. Add desired DI and store
4. A/D conversion of front panel potentiometer, put in memory (fixed delay)
5. Delay loop 10-20 usec.
6. A/D conversion of current
Above target?

Y: State change

N: No state change
The alarm handler does the following:

1. **Alarm threshold reached**

2. **Decode threshold number**

3. **Turn off everything**

4. **Flash LED coded number of times**
The foreground loop performs these tasks:

1. Read Setpoint from memory
2. If new value?
   - No, go to Within Bounds?
   - Yes, store in different memory
3. If Within Bounds?
   - No, go to Subtract actual period from set period
   - Yes, go to Read Setpoint from memory

FSTART
Scale by $K_p$

Save as proportional term

Add result to integral accumulator

Scale integral by $K_i$

Add to proportional term

Average result with previous drive current

Store result as new drive current

to FSTART
5.4 Production Design of a Microcontroller

The experimental system is designed for flexibility. This means that its components must be as general purpose as possible. Since this is the antithesis of custom design, the system uses only off-the-shelf components, connected as generally as possible. This leads to a large system, designed to handle every imaginable task. A production controller, however, would be designed differently. This section presents a few suggestions for using the core structure of the experimental controller, still with its inexpensive off-the-shelf components, in a much smaller, dedicated controller.

The 8031 microcontroller is available with on-chip ROM. Since a production controller need not accept downloaded programs, this will suffice for program memory. Thus, the external EPROM and RAM chips are not needed.

Comparators and D/A converters can perform all analog input functions, if this solution is much less expensive than the eight-channel A/D chip used in the experimental system. Also, in the test program, the four comparators only detect alarms. Their thresholds are set only once, which suggests that they could be fixed by resistors, instead of adjusted by D/A converters. This would eliminate more parts.
Both digital inputs can be simplified. The tachometer input is not used, so it can be eliminated entirely. Finally, the serial port need not have RS-232 levels in a dedicated controller, so its level shifters can be removed as well.

The result of the above simplifications is that the controller can be implemented using only a microcontroller, two comparator chips, a digital-to-analog converter, an output latch, and approximately two simple logic chips. This is a total of seven chips, only half the number used in the experimental system.
Chapter 6

Conclusion

A complete and functional indirect position sensing system for variable reluctance motors was developed and incorporated within a speed control loop. This development began with an examination of the structure of the variable reluctance motor. The principles of torque conversion were discussed and the ensuing requirement of large variation in phase inductance was explained. This variation also provided a means of indirect position detection. Furthermore, the synchronous nature of the motor meant that phase currents must be regulated according to shaft position, which meant that accurate position information was important.

Next, a basic algorithm for position detection was developed. This algorithm examined short pulses of current in idle windings to estimate the inductances of those windings and then work backwards through the known relation of inductance to position to estimate position. Immediately, difficult issues presented themselves. The motor and controller could easily handle large currents which changed relatively slowly, but behaved differently for small, rapidly changing currents. One such problem was caused by eddy currents,
arising due to the high rate of change of current in the small sense pulses. Another such problem was poor sensor resolution, since control sensors must handle both the large drive currents and the small sense currents. Sensor noise caused a problem, since small quantities were being measured. Additionally, problems with interphase coupling occurred, since the motor had little interphase coupling at the low frequencies used by drive currents, but had high coupling at the high sense pulse frequencies, when eddy currents altered the magnetic flux paths.

Solutions to these problems were presented one by one. Each problem was analyzed extensively both theoretically and experimentally to obtain and verify its solution. Such analysis revealed that the problem of eddy currents was solvable with the observation that although the currents cause a jump in sense-pulse current in the short term, their effect is minor immediately thereafter. Adjustment to the sensing algorithm allowed it to avoid the critical initial interval. The sensor resolution problem was surmountable by proper choice of pulse magnitude. Interphase coupling could not be reduced, but could be avoided by ensuring that critical data was not taken during intervals of strong coupling. This required giving the position detection algorithm the ability to predict position over short intervals even with no direct information from the motor. Hence, the algorithm was enhanced by the addition
of a simple observer. These changes produced a fully functional position sensor.

Next, a set of hardware and software was described which was capable of implementing the position sensing algorithm. The computational simplicity of the algorithm allowed its implementation with a small, inexpensive control package.

The usefulness of the position sensor was demonstrated by embedding it into a closed-loop speed controller. The VRM was modelled as a first-order single-input, single-output plant and a compensator was developed for it. Numerical values for the parameters in the compensator were derived with attention to performance and practical constraints. The resulting compensator improved the dynamic response of the open-loop motor system by a factor of a hundred. The limits on the performance of the compensated motor system were analyzed and explained.

In summary, a complete, physically compact, inexpensive motor control system was presented, based on the method of indirect position sensing developed in this thesis. To complete the presentation, suggestions for future research follow. First, the simple observer algorithm incorporated in the position sensing routine could be extended to allow for phase commutation at other than the minimum inductance points. This is necessary for high performance control over a wide range of speeds and for dynamic
braking. One simple method for advancing the firing was mentioned, specifically, raising the threshold used as the minimum, so that the algorithm uses a point slightly earlier than the true minimum. However, as noted in the section on eddy currents, the apparent inductance, measured by the high frequency pulses, is approximately constant over a large range of shaft positions. Thus, changing the firing angle greatly by detecting a different value of inductance is not practical because there is only a small range of angles which have distinct apparent inductance values. If great flexibility of firing angle is needed, it will be necessary to drive the observer with the minimum inductance point, but commutate at a different, unrelated position. Such commutation is quite possible and is a logical extension to the algorithms presented in this thesis.
Appendix B

Schematic Diagram of Power Electronics

MIT - CORP LAND VRU CONTROLLER, POWER CIRCUIT

JULY 29, 1986

C1 = C2 = 430 uF
DIODES = IN914
R01 = 220 kΩ
Q1 = 6550 D2
Q2 = LM353

A, B, C:  
R1 = 0.1 Ω, 3 W (two in parallel)
R2 = R4 = 1 kΩ
R3 = RC = 10 kΩ

D:  
R2 = 470 kΩ
R3 = 22 kΩ
R4 = 390 kΩ
R5 = 10 kΩ

- 122 -
Appendix C

Listing of Control Program

; Walter D. Harris 8/12/86  Rev WDH 10/20/86. Rev WDH 12/16/86 and 4/87.

; Controller program for Copland VRM with indirect position sensing.
; Has integral control, runs at variable speed.

; This program runs on the controller package with revision approx.
8/1/86.

; Originally SPEED5.MSA
Closed-loop speed regulation
; Assemble with Enertec A8051 Cross Assembler for IBM PC.

assemble_main equ 1 ; assemble main part of program

; To assemble diagnostics, uncomment 'include' command at end of this file.

; ALARM CODES:
; 1, Ia limit detected in bkgn; 2, Ib limit in bkgn; 3, Ic, same;
; 4, Vdmp limit, bkgn; 6, Vdmp limit, chop sbtrn;

; TIMER USAGE:
; TM0L delineates sensing/chopping interval
; TM1 measures time between phases (K/shaft_speed)

; REGISTER USAGE:
; Foreground uses register bank 0, R0-R5.
; Background uses register bank 0, registers R6 and R7, and DPTR.
; DPTR is NOT saved by int_enter macro.
;MNEMONICS:
;Dump Voltage readings are biased high by Vsupply. (I.e., voltage reads as
;the sum of the capacitor and supply voltages, or the voltage across
;the FETs.)

;Current readings are actually 10 x voltage across a .05 Ohm resistor, so
; internal_I = (255/5) * 10 * .05 * I_external
;                = 25.5 * I_external

Iscale_num    = 255 ; fraction by which to multiply desired current
Iscale_den    = 10  ; to get internal number
Vscale_num    = 64  ; fraction by which to multiply desired voltage
Vscale_den    = 100 ; to get internal number
min_setpoint  = 143 ; limit value, 1400 RPM
max_setpoint  = 255 ; no upper limit on speed setpoint

; Speed control parameters
max_current   = 6 * Iscale_num/Iscale_den ; limit value
pgain_div     = 1 ; log2(divisor for error-->output function)
igain_div     = ; programmed to fixed value (see code)

; Position sensing parameters
MAX_SENSE_TIME EQU 7 ; ramp up time for sense, /2 usec (39 us ovrrd).
ITSENSE       EQU 28 ; x.039A/div (28 for 100V sply, mst=13)
    sense_interval EQU 10 ; phase current sense threshold
    rise_time      EQU 20 ; state transition check intrvl
                        ; sense_intrvl usd when phase trnd on (max 255)

; Mode related parameters
TCHOP          EQU 15 ; chop period, used only -- during prestart
startup_current = 4 * Iscale_num/Iscale_den ; chopping lvl for prestart
max_preset    EQU 4096 ; 1250 RPM -- slow/fast mode changeover point
;Alarm parameters
alarm_current = 254 ;(Almost full scale, about 10 amps)
               ; 10 * Iscale_num/Iscale_den ;shutdown at this val
alarm_voltage = 300 * Vscale_num/Vscale_den ;shutdown at this val
                ;(Vsupply + resistors)

;Set up facilities for reading analog inputs and writing switch outputs

;Analog input channel numbers
la = 0 ;Current in phase A
lb = 1 ;
lc = 2
vdmp = 3 ;Voltage on dump capacitor
vdc = 4 ;DC Supply voltage
setpt = 5 ;Analog input for setpoint (from user)

;Identity Masks for active low binary outputs
sa = 1b ;Switch driving phase A
sb = 10b ;
sd = 100b ;
sdmp = 1000b ;Switch driving dump inductor
led4 = 10000b ; ... LEDs used for monitoring ...
led5 = 100000b
led6 = 1000000b
led7 = 10000000b
all = 11111111b ;all binary outputs
all_transistors = 00001111b ;all transistors
all_phases = 00000111b ;all phase windings

;Configuration words for timers
TCON_DATA = 00000100b ;init value for TCON -- lvs
                ;tmrs off,
                ;sets ext int1 to edge trg, 0
to lvl
TMOD_DATA = 00010010b ;Tmr0: 8b auto-reload, TMR1: 16b
                ;both are internally clocked

;Configuration for interrupt priority structure
INT_PRIOR = 00000001b ;all interrupts prior 0 except
all_alarms = 00001111b ;interrupt mask (all on, int0 positive pol)

;External addresses:
DAC = 4000H ;base address for DAC
ADC = 6000H ;analog mux control
SWITCHES = 8000H ;switch port

;******************************************************************************
;  *                              MACROS                                   *
;******************************************************************************

macro %lock ;mnemonic
clr EA
endmac

macro %unlock ;other mnemonic
setb EA
endmac

macro %switch ;assert or unassert binary output
;format: 'switch #LED4 ON' or 'switch <src> ON' time: 10
OFF/11
ON
mov dpdr, SWITCHES ;address of switch port
if '\1' = 'ON'
mov a, \0 ;get mask for turning on
cpl a ;outputs are negative true
anl a, switch_copy ;get rest of bits
else
mov a, \0 ;get mask for turning off
orl a, switch_copy ;get rest of bits
endif
mov switch_copy, a ;store back to memory
movx @dpdr, a ;put on port
endmac

macro %toggle ;toggle output(s). Fmt: 'toggle #mask'
mov DPTR, #SWITCHES
mov a, \0
xrl a, switch_copy
mov switch_copy, a
movx @dpdr, a
endmac
macro %analog_rd ;read an analog input channel to
  reg/mem ;fmt 'analog_rd reg, #Ia' time: 11 or
12 us
        mov   DPH, #(high ADC) ;load address of DAC
        mov   DPL, \1 ;get channel number, used as DAC
        offset
        movx  a, @dptr ;initiate read
        nop
        nop
        nop
        movx  a, @dptr ;get result of conversion
        if ('\0' NE 'a') and ('\0' NE 'A') ;not going into
        endif
        accumulator
        mov   \0, a ;put into destination register
        endif
endmac

macro %int_enter ;entry code for interrupt
setb   INT1 ;&&&& test indicator of processor
          time
        push  acc ;format: '%int_enter'
push   psw
endmac

macro %int_exit ;exit code for interrupt
pop    psw ;format: '%int_exit'
pop    acc
clr    INT1 ;&&&& test indicator
reti
endmac

macro %set_threshold ;set interrupt threshold
us
  ;fmt: 'set_thresh #Ia, #100d' time: 7
    if '\l' NE 'a' ;get value into accum
        mov   a, \1
    endif
        mov   dph, #(high DAC) ;high byte of DAC base adr
        (low is 0)
        mov   dpl, \0 ;low byte selects section (0-3)
        movx  @dptr, a ;set value
endmac

;Note: In mode 3, timer 0 is split into two 8b timers and
;uses the control bits TR1 and TF1 normally assigned to tmr 1.

- 127 -
macro %start_tm0L ;set timer 0 to overflow in (x) counts
;NOTE: only works for 8 bit args!!
fmt: 'start_tm0L

<interval_length>
clr TR0 ;stop timer

if ("\\L1\\L1\\L0\\"", EQ '#') ;Is macro argument of form "#ARG" ?
mov TLO, #LOW(-("\\L32\\L2\\L0\\" - 7)) ;YES, set timer to (ARG - 7)
else ;NO
mov a, #11 ;adjust for setting up time
clr c
subb a, \0 ;prepare for count up
mov TLO, a ;set low
endif
setb TR0 ;start timer
endmac

#include multiprecision arithmetic macros

;include multiprecision arithmetic macros

public loc0

; ************************************************************
; * VARIABLES                                          *
; ************************************************************

;byte variables
;addresses variables in internal RAM (above register banks 0-3 and bit space):
switch_copy equ 30H
setpoint equ 31H ;8 bit current level for chopping
sense_cur equ 32H ;sense phase channel number
sense_sw equ 33H ;sense phase switch mask
main_cur equ 34H
main_sw equ 35H
last_preset0 equ 36H ;memory for state transition timer
last_preset1 equ 37H
state_interval0 equ 38H ;interval of one state transition
state_interval1 equ 39H
pval equ 3AH ;copy of potentiometer reading

- 128 -
last_pval equ 3BH ; previous copy of
           potentiometer rdg
integral0 equ 3CH ; memory for integral control
term
integral1 equ 3DH
set_period0 equ 3EH ; setpoint, converted to period
set_period1 equ 3FH
setpt_filter equ 4AH ; state memory for setpoint
                     filter

; bit variables     (start at byte address 20H)
fast_mode_flag equ 00H

;******************************************************************************
;                        STACK                                            *
;******************************************************************************

stack_base equ 0060H ; start address, builds up fm here
                    ; (32 bytes before end of
                    ; RAM)

;******************************************************************************
;                        INTERRUPT VECTOR TABLE                           *
;******************************************************************************

rseg code ; back to code
loc0: ; this is location 0h or location 2000h, depending on
       ; loading method
if assemble_main EQ 1 ; Omit this if making diagnostics only
    jmp start ; this is the reset vector address
    org loc0 + 0003h ; external interrupt 0 vector address
    jmp exti0_handler
    org loc0 + 000BH ; timer 0 interrupt vector address
    jmp tm0_handler
    org loc0 + 0013h ; external interrupt 1 vector address
    jmp exti1_handler
    org loc0 + 001Bh ; timer 1 interrupt vector address
    jmp tm1_handler
    org loc0 + 0023h ; serial port vector address
    reti ; ignore incoming data

- 129 -
endif

;;;;;;;;;;;;;;;SUBROUTINES;;;;;;;;;;;;;;;

pause:                ;subroutine to pause for acc/100
                    ;seconds
    push acc         ;set up outer loop
ploop3:         mov a, #10 ;set up middle loop
ploop2:         push acc
    mov a, #250
ploop1:         dec a     ;1 cycle
      nop          ;1 cycle
      jnz ploop1   ;2 cycles
      pop acc     ;middle loop
      dec a
      jnz ploop2
      pop acc     ;outer loop
      dec a
      jnz ploop3
    ret

;CHOP is a procedure with 2 arguments. R0 is the switch mask (i.e.,
;#Sa)
;and R2 is the time to chop in 0.1 sec increments. The
;Chopping limits the current to the value set in the DACs.
;ACC, R1-4 are used, not restored.
CHOP:
    CLR a
    mov R1, a         ;clear R1, which is off-time flag and
counter
    mov R3, a         ;also clear inner loop counters
    mov R4, a

turn_on:
    %switch
    R0, ON               ;fire switch

;Main chopping loop returns to here
CHOP1:       cjne R1, #0, now_off ;if R1 nonzero, now counting off-time
        jnb P1.3, no_ovv ;check for alarm interrupt
        jnb P1.3, no_ovv ;check for alarm interrupt

- 130 -
jn b P1.3, no_ovv ;check for alarm interrupt
jn b P1.3, no_ovv ;check for alarm interrupt
jn b P1.3, no_ovv ;check for alarm interrupt
mov a, #6 ;overvoltage alarm code
jmp shutdown

no_ovv:

JB INT0,loop ;No, in on_time. Do nothing if < threshold.

;turn_off:
%switch

;Above threshold! turn off switch
#all_phases, OFF
mov R1, #TCHOP ;has exceeded threshold, start
off_time

now_off:
dj nz R1, loop ;more off time left? yes: continue
sjmp turn_on ;no: turn switch back on

loop:
dj nz R4, CHOP1
mov R4, #43 ;inner loop (R6) is 43 * 9 usec
DJNZ R3, CHOP1 ;next multiplies by 256
DJNZ R2, CHOP1 ;outer is argument to procedure
%switch
#all_phases, OFF ;make sure to leave phase off
RET

if assemble_main ;assemble following if soft switch set

;dummy routines:

tm1_handler:
extil_handler:

reti ;These interrupts are not used.

; ***********************************************************************
; *  I N I T   C O D E  *
; ***********************************************************************

public start, shutdown

start: ;program startup

;Simulate a hardware reset in case this is a soft start
mov IE, #0 ;ensure global, specific interrupts off
mov SP, #stack_base ;set up stack pointer
mov PSW, #00 ;Select register bank 0

- 131 -
mov TH0, #0 ;Make 8b auto-reload same as normal overflow

all_stop:
    mov IE, #0 ;jump here to stop everything.
    (again) make sure ints off
    mov IP, #INT_PRIOR ;interrupt priority settings

%switch #all, OFF
%switch #LED7, ON ;turn off all outputs
;signal all stopped using LED

;set up timers
mov TCON, #TCON_DATA ;stop timers, set modes of ext
    ints
mov TMOD, #TMOD_DATA ;set mode of timers

;Pause here. This ensures that potentiometer jitter doesn't cause
;immediate restart when operator is turning setpoint DOWN to zero.

mov R1, #0 ;1 sec pause here
wt1:
    mov R0, #0
    djnz R0, $
    djnz R1, wt1

setb EA ;enable global interrupts

;******************************************************************************
; * INITIALIZE SPEED CONTROLLE R *
;******************************************************************************

clr a
mov integral0, a
mov integral1, a

mov setpoint, a ;clear setpoint-related
    parameters
mov last_pval, a
mov setpt_filter, a ;clear state memory for filter

;******************************************************************************
; * WAIT FOR START UP SIGNAL * 
;******************************************************************************

;get_setpoint to see if time to start motor (setpoint nonzero, valid)
%analog_rd a, #Setpt ;read 8 bit setpoint from pot
mov R0, a ;save
;Now, test if too large or small:
clr c
subb a, #min_setpoint ;if very small setpt, stay stopped
JC all_stop ;stop and wait for higher setpoint
mov a, R0 ;fetch setpoint back
clr c
subb a, #max_setpoint
jnc all_stop ;reject if too large
%switch #LED7, OFF ;clear stop light
mov pval, R0 ;initialize copy of potentiometer reading

; *****************************************************************************************
; # MOVE SHAFT TO INITIAL POSITION
; *****************************************************************************************

; Align shaft for startup
sequence
;Leave alarms disabled.
;just for startup:
mov a, #startup_current ;calculate target current
%set_threshold #Ia, a ;install as chop value
%set_threshold #Ib, a ;TEMPORARILY init all to this value
%set_threshold #Ic, a ;
%set_threshold #Vdmp, #alarm_voltage ;install voltage alarm
mov R3, #0 ;pause for settling time
djnz R3, $
mov P1, #all_alarms ;set polarity of analog comparison, enable all
%switch #LED4, ON ;tell user aligning rotor
mov R0, #Sc ;fire phase C to align rotor
mov R2, #30 ;fire for three seconds
call CHOP
%switch #LED4, OFF ;tell user alignment done

; *****************************************************************************************
; BEGIN ROTATION

; SETUP STATE MACHINE
call init_state ; initializes all state machine pointers

; START STATE MACHINE
call start_state ; fires first phase (DPTR now resrved for bkgnd)

; LOOP WHILE UPDATING SETPOIN T

update_setpoint: ; loop to update memory copy of setpt pot
    mov a, pval ; fetch latest potentiometer reading
    xrl a, last_pval ; same as last time?
    jz setpoint_done ; YUP, don't need to change
    mov a, pval ; NO. Fetch pot reading again.
    mov last_pval, a ; save for next time

; bounds check for pot value:
    clr C ; see if too small
    subb a, #min_setpoint ; too small?
    jnc benuf ; no. proceed.
    jmp all_stop ; yes, stop motor.

benuf: subb a, #(max_setpoint-min_setpoint) ; No. too big?
    jnc setpoint_done ; yes. skip rest of setpt processing.

; end of bounds check

; calc_period: ; Convert speed setpoint to period spt: P = 32 * (255-pval)
    mov a, last_pval ; Fetch authenticated user setpoint
    cpl a ; calculate 255 - setpt
    mov B, #32 ; next, multiply by 32
    mul AB
    mov set_period0, a ; save for next time (low byte)
    mov set_period1, B ; (high byte)

setpoint_done:
mov R2, set_period0 ;fetch target period
mov R3, set_period1 ;R3/R2 now contains valid target
PERIOD
CLR EA ;locked transfer
mov R5, state_interval1 ;fetch actual state time (16
bit)
mov R4, state_interval0
SETB EA
clr c
mov a, R4
subb a, R2 ;determine P0 – Pactual
mov R4, a
mov a, R5
subb a, R3 ; and leave in R5/R4
mov R5, a
;calculate_current:
;scale pd difference to obtain output current

IF pgain_div = 1
%asr16 R5, R4
ELSE
mov . R0, #pgain_div ;how many bits to shift period
 diffrrnce right
sgain: %asr16 R5, R4 ;do sixteen bit arithmetic shift
djnz R0, sgain ;shift again if needed
ENDIF

;Signed proportional term now in R5/R4. Check whether want to
integrate.

mov a, R5 ;don’t integrate until within prop
band
jz Integrate ;(true for posnum if hibyte zero)
cpl a
jz Integrate ;(and for negnum if hibyte all ones)
mov a, R5 ;out of prop band, skip integration
sjmp integral_done

Integrate: ;prop term (integrand) is now in R5/R4
; prescale integrand down by 2, leave in R3/R2


```
; Calculating integral term based on prop term.
mov a, R2
add a, integral0
mov integral0, a
mov a, R3
addc a, integral1
mov c, acc.7
xch a, integral1

;CHECK FOR OVERFLOW OF INTEGRAL
; over/underflow if two positive numbers sum to a negative number
; or if two negative numbers sum to a positive one. All else OK.
jc negres

negres:
posres: rlc a
mov a, R3
anl c, acc.7
jnc inrange
mov integral1,#080h
mov integral0,#000h
sjmp inrange

negres:
rc l a
mov a, R3
orl c, acc.7
jc inrange
mov integral1,#07fh
mov integral0,#0ffh

inrange:
mov R3, integral1
mov R2, integral0
```

- 136 -
; Do 7 bit right shift of integral term
    mov  a, R2         ; fetch msb of low byte
    rlc  a
    mov  a, R3         ; fetch high byte
    rlc  a             ; shift left one, sign bit into carry
    mov  R2, a         ; shift right 8 bits by putting into
                         ; low byte
    clr  a             ; Sign extend through high byte.
    jnc  pint          ; positive number?
    cpl  a             ; NO, extend negative
    pint: mov  R3, a    ; Shift done. R3/R2 now has scaled
                         ; integ term.

    mov  a, R2         ; add integral, prop. terms
    add  a, R4
    mov  R4, a         ; save low byte of result
    mov  a, R3         ; add high bytes
    addc a, R5
    mov  R5, a

integral_done:
    ; validate_current:
    jz    no_ovr       ; R5/R4 has desired current, a=R5
                         ; Look for small positive number.
    jnb   acc.7, ovrflw ; If pos but not small, must be
                         ; overflow
    mov   R4, #0       ; If negative, make zero
    sjmp   valid_current

no_ovr:
    mov   a, R4         ; fetch low byte
    clr   c             ; Low byte of output below limit?
    subb  a, #max_current
    jc    valid_current ; YES

ovrflw: mov   R4, #max_current   ; Use max current instead.

valid_current:
    ; R4 now has desired current level.
    mov   a, R4         ; smooth setpoint by averaging
                         ; w/prev
    add   a, setpoint
    rrc   a

    - 137 -
mov setpoint, a ;pass to background

cpl T1 ;indicate control action to oscilloscope
jmp update_setpoint ;Repeat setpoint check and control adjustment

;***************************************************************
; * POSITION TRACER *
;***************************************************************

;This is a finite state machine. There are 3 distinct states. They
;represent 3 phases, each of which provides torque for 30 degrees.
;Then,
;it is shut off and the next is driven.

;State transitions occur when low inductance is detected on the next
;phase.
;The program estimates that inductance by applying V_s to the sense
;phase
;for a predetermined interval and observing whether the current
;rises
;more than ITSENSE. An allowance is made for feedthrough by
;beginning
;the slope measurement SETTLING_TIME after the switch is activated.

;During ON interval of the main phase, main current is limited
;by chopping. Chop decisions are made at fixed intervals, determined
;by timer0L overflows.

;****
;**** MACROS FOR FOLLOWING ALGORITHM
;****

;**** Chop main phase at desired current.
macro %chop_main ;Format: "chop_main"
%switch
main_sw, ON ;time: 11 us
clr c
%analog_rd a, main_cur ;sense current time:11 us
subb a, setpoint ;compare with target -- set C if low
jc too_low@ ; (@ is macro cmd:create nr, save it)
%switch #all_phases, OFF ; Turn main off if too high.
  too_low:<; (macro cmd: put created nr here)
endmac

;**** Main Inductance Sensing Routine // used by all modes

; Test sense winding. If state chg needed, jump to indicated adrs.
; Ovrhd:10 btwn ON and 1st point, 12 during, 15 between 2nd and OFF

macro %sensejump ; Format: sense+jmp <state_chg_rtn_adr>

%switch sense_sw, ON ; drive sense winding

; settling time is about 10us, during the following measurement

%analog_rd a, sense_cur ; take first current measurement

; Calculate target and wait a fixed interval for current to rise.
add a, #ITSENSE ; delta_I threshold
mov R7, a ; save target

; The following line takes a constant time, so is also used as delay.
%analog_rd pval, #Setpt ; pass user input to fgn_d
mov R6, #MAX_SENSE_TIME ; add some more ramp delay
djnz R6, $

clr c

%analog_rd a, sense_cur ; take second current measurement
subb a, R7 ; need state trans iff C==0 (NC)

jc no_sc\@ ; Need state change? (make unique label)

jmp \0 ; YES, go do it.
no_sc\:< %switch sense_sw, OFF ;NO, turn sense drive off 
endmac

;****
;**** STATE MACHINE CODE
;****

; THE INITIALIZATION ROUTINE
init_state:
mov    main_cur, #Ia    ;set up for firing phase A, sensing B
mov    main_sw, #Sa
mov    sense_cur, #Ib
mov    sense_sw, #Sb

%set_threshold   #Vdmp, #alarm_voltage     ;set alarms for intrpt
               monitor

mov    a, #alarm_current
%set_threshold   #Ia, a
%set_threshold   #Ib, a
%set_threshold   #Ic, a

mov    P1, #all_alarms     ;include all alarms
setb   EX0
               ;enable alarm interrupt

mov    lastpreset1, #(high max_preset)  ;1st state interval is
               long.
mov    lastpreset0, #(low max_preset)
clr    fast_mode_flag       ;start in slow mode
setb   TF1
               ;(ensure timer ovrfw)

mov    state_interval1, #(high 10000) ;init interval to very
               long
mov    state_interval0, #(low 10000)  
ret

; THE ACTIVATION ROUTINE
start_state:
setb   ETO     ;enable use of timer 0 interrupt by
               bckgnd
setb   TF0     ;simulate end of 0th chop interval
ret

; THE TRANSITION DECISION ROUTINE
;**** generic state handler -- uses masks, counters to address
               appropriate
;***** switches, sensors. Also chops selected main phase.
;***** Has two different algorithms, one for fast and one for slow speed.

tm0_handler:
;dispatch:
    %int_enter          ;save everybody
    jb         fast_mode_flag, fast_mode           ;use algorithm for fast rot?

;slow_mode:
    %sensejump        pset                   ;Note: main is NOT always on during sensng here
    ;Sense and jump to PSET if state change needed.
    ;(In this mode, sensing always occurs)
    ;No state chg. Chop main then lv
    %chop_main        ;(In this mode, main is always chopped)
    %start_tm0L       #sense_interval        ;start timer for next interval
    clr        TF0            ;clear pending timer interrupt
    %int_exit       ;continue in this state -- pointers unchanged

fast_mode:   ;In this mode, sensing begins only after TMR1 overflows
    jnb      TF1, no_sns    ;OK to sense?
    %sensejump       pcalc    ;Sense to see if need state change
    no_sns: %chop_main    ;No. Chop main and quit.
    %start_tm0L       #sense_interval    ;start timer for next interval
    clr        TF0            ;clear pending timer interrupt
    %int_exit       ;continue in this state -- pointers unchanged
pset; ;State change needed from slow mode:
clr TR1 ;stop state timer
jb TF1, still_slow ;if timer overflowed, stay in slow mode
mov TH1, #(high max_preset/2) ;transfer to fast mode
mov TL1, #(low max_preset/2)
sjmp pcalc_done ;save preset and change state

still_slow:
mov TH1, #(high max_preset) ;still in slow mode -- pin timer
mov TL1, #(low max_preset)
sjmp pcalc_done

pcalc: ;Calculate value for timer preset (fast mode)

;leave sense switch ON.
;Fast mode, Preset timer to 1/2 of actual interval length.

;Set up timer to inhibit transition checks for 1/2 of the last
;state interval. (This reduces false transitions, speeds chopping)

;First, how long was previous interval?
;Counter WAS preset to -(last_preset) then allowed to run up
;past that and continue going positive.
;Total elapsed interval is thus (TH1 TL1) + (last_preset)

;last_preset is abs(timer init val), so is positive value.
clr TR1 ;stop timer
%add16 TH1, TL1, last_preset1, last_preset0 ;calc total elapsed tm

;TH1 TH0 now contains length of last interval.
mov state_interval1, TH1 ;pass to foreground
mov state_interval0, TL1

not_a2; ;Now, calculate inhibit time (counter preset) for next interval
%shr16 TH1, TL1 ;Wait (state_trans_time/2) before sensing

pcalc_done:
mov last_preset1, TH1 ;save timer preset value
mov last_preset0, TL1
mov a, last_preset1 ;set mode flag
clr C
subb a, #high max_preset ;is max_preset bgr than
mov fast_mode_flag, C ;last_preset?
;
;xrl TH1, #0ffh ;yes, moving fast, set fast
;xrl TL1, #0ffh ;mode.
clr TF1 ;clear overflow from last cycle
setb TR1 ;start timer for next inhibit interval

;Advance state information to drive next phase
;initiate next state -- uses fact that switch masks go 0001,0010,0100
;and current pointers go 0,1,2

%switch
  main_sw, OFF ;prev main phase now unused
  mov main_cur, sense_cur ;advance state pointers
  mov main_sw, sense_sw
  mov a, sense_sw ;rotate switch mask to next
  rl a ;phase
  jb acc.3, next_is_a ;time to reset to state zero?
  mov sense_sw, a ;no, advance normally
  inc sense_cur

%switch
  #LED5, OFF ;signal user -- not state A
  sjmp state_changed

next_is_a:
  mov sense_sw, #0001b ;jump back to state zero
  mov sense_cur, #0

%switch
  #LED5, ON ;signal user -- state A

state_changed:
  %start_tm0L #rise_time ;allow current to build up B4
  clr TF0 ;sensing
  ;clear pending timer
  interrupt
%int_exit

endif ;end of main program

;*******************************************************************************
; * E M E R G E N C Y S H U T D O W N *
;*******************************************************************************

false_trigger:
    reti

exti0_handler:
    ;alarm interrupt comes here
    jb INTO, false_trigger ;deglitch interrupt
    jb INTO, false_trigger
    jb INTO, false_trigger
    mov a, #1 ;shutdown interrupt number
    jb P1.0, shutdown ;decode error -- 1 = Ia
    inc a
    jb P1.1, shutdown ;2 = Ib
    inc a
    jb P1.2, shutdown ;3 = Ic
    inc a ;4 = Vdmp

shutdown: ;trouble! Flash code in accumulator on LED.
    clr EA ;stop everything
    mov R0, a ;save code
    %switch #all, OFF ;turn off all switches

flash: mov a, R0 ;retrieve code
    rl a ;double code
    mov R1, a ;save doubled copy

 tog: %toggle #LED6 ;set alarm indicator
    mov a, #15 ;15 secs
    call pause
    djnz R1, tog ;flash n times
    mov a, #60 ;pause again
    call pause
    sjmp flash ;repeat entire sequence

;Comment out the following line to omit optional diagnostics:
;$spn_diag.msa ;include for diagnostic code
;NOTE: The dollar sign is local syntax for 'include'.
end
References


XI. A. Pittet and M. Jufer, "Closed-Loop Control without Encoder of Electromagnetic Step Motors", Proc. of the 7th Incremental

