UNIVERSITY OF ALBERTA

STEP CONFIRMATION FOR CLAW-POLED PERMANENT MAGNET STEPPING MOTORS

BY

© CORNELIUS VERBURG

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF ELECTRICAL ENGINEERING

EDMONTON, ALBERTA

FALL, 1990

ISBN 0-315-64975-5

UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: CORNELIUS VERBURG

TITLE OF THESIS: STEP CONFIRMATION FOR CLAW-POLED PERMANENT MAGNET STEPPING MOTORS

DEGREE: MASTER OF SCIENCE

YEAR THIS DEGREE GRANTED: 1990

C. Verburg

8915 - 146 STREET

EDMONTON, ALBERTA

DATE: 30 Aug 1990

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

THE UNDERSIGNED CERTIFY THAT THEY HAVE READ, AND RECOMMEND TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH FOR ACCEPT 'ICE, A THESIS ENTITLED *STEP CONFIRMATION FOR CLAW-POLED PERMANENT MAGNET STEPPING MOTORS* SUBMITTED BY CORNELIUS VERBURG IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN ELECTRICAL ENGINEERNG.

Y.J. Kingma

D.H. Kelly

R.W. Toogood

DATE: 22-May-1990

Abstract

A claw-poled stepping motor driven robotic manipulator can suffer from positioning errors due to missed steps. To provide step confirmation, a simple and inexpensive position feedback encoder has been implemented. The mechanism uses readily available optical reflective sensors and a lightweight optical disc which does not load the motor. In arriving at this solution, several other technologies were considered including several types of waveform detection and Hall-effect sensors.

Acknowledgement

I would like to thank a number of people who have helped me with this work.

First thanks goes to Professor Kingma; not only for ideas and financial support, but also for providing me with an excellent environment in which to carry out this work and for ensuring that I had all the necessary tools.

Thanks also to my fiancé Heidi Hansen, who has taken loving care of me, especially during the last few weeks as I prepared this thesis. I hope that I can do as well in return, as she completes her studies.

Thanks to my father for sparking my interest in electrical engineering and encouraging me throughout my studies.

Much of this work would have taken much longer but for the ideas and help provided by Frank Niscak. In addition to academic help, he has also been a good friend: always ready to help when needed ; always forcing me to *go for my run* when I didn't really want to; and always ready to show me how to play tennis—by winning time and time again.

Ed Sumbar was an invaluable aid in the preparation and layout of this manuscript.

Table of Contents

List of Tables

List of Figures

Abbreviations and Symbols

$^\circ$	degrees
δi	range of *i* during chopping
Hz	Hertz
i	instantaneous phase current
I_{rated}	rated current
i/o	input-output
ℓ	incremental inductance
m	number of phases
N_r	number of rotor poles
s	seconds
t_{rise}	current rise-time during chopping
t_{decay}	current decay-time during chopping
Ψ	flux linkage
PM	permanent magnet
S	steps per revolution
θ	rotor position in mechanical radians
θ_S	step angle in degrees
VR	variable-reluctance

1. Introduction

Experience with a robotic manipulator used as an aid to quadriplegics has demonstrated the need for step confirmation from its stepping motor actuators [Kimura] In this research project, several methods of acquiring this feedback have been investigated. Since traditional forms of feedback such as optical encoders, which must be coupled to the motor shaft, are expensive and require significant modification of the existing manipulator, only alternative forms of feedback have been considered; namely observation of winding current characteristics and the placement of small sensors near the motor to detect motion of its permanent magnet rotor. Based on past experience with the manipulator, two specifications for the scheme have emerged: First, the manipulator was designed with strong emphasis on low cost. Any feedback method developed in this research must address this same concern. Second, simple step confirmation is considered sufficient; directional information, though desirable, is of secondary importance.

Evaluation of a number of feedback methods (including current waveform detection and two types of magnetic field sensors) proved them to be inadequate for this application. Finally, a variant of the conventional incremental optical shaft encoder was developed. The encoder consists of a sectored disk cemented to the exposed rear face of the rotor and two reflective optical sensors used to detect motion of the sectored disc. The reflective sensors provide a quadrature phase signal from which both step and direction information can be determined.

What follows is first a discussion of the goals of the project, second a description of the motors used, third discussions of the schemes that did not work and finally the development of the optical disc method.

2. Project Specification

Simply stated, the goal of this research project is to develop a device capable of providing confirmation of rotor steps in response to step commands. Several other important criteria are listed below:

- The device is to be designed specifically for the type of stepping motor known as the *claw-poled permanent magnet stepping motor*, which is the type of actuator used for the robotic aid. These motors are commonly used because they are inexpensive and readily available off-the-shelf with various gear-head arrangements.
- The device is to be inexpensive in order to comply with the requirement that the robotic aid be affordable. A related concern is that in order to minimize maintenance costs, the device should be easy to use and maintain.
- For safety reasons, the device must be reliable.
- Further simplification of the overall design can be realized if the feedback scheme can also be used for the manipulator's learning mode.
- Implementation should require minimal modification to the existing manipulator.

3. Stepping Motors

3.1. General Description

A stepping motor is a digitally controlled motor; each input pulse to the control circuitry causes the rotor to rotate through a fixed angle and come to rest at a fixed position. Although several types of stepping motors exist, only the permanent magnet and variable-reluctance types are described here; particular attention is paid to the so-called *claw-poled* motor used in the manipulator.

Stepping motors are available with step resolutions from four to 1000 steps per revolution. The most common structures, though, have 24, 48, or 200 steps per revolution which correspond to step angles of 15°, 7.5° and 1.8° respectively.[Kenjo] The manipulator uses only 7.5° motors.

3.2. Advantages and Disadvantages of Stepping Motors.

3.2.1. Advantages

The following features make stepping motors an attractive choice for many applications:

- they allow economical open-loop speed control.
- they allow economical open-loop position control (with non-cumulative position error [Kenjo]). This makes them the only choice for simple robotics applications which do not employ sophisticated closed-loop control.
- they are easily controlled using microprocessors and other digital circuits; they are therefore used in many computer peripherals.
- they are durable and reliable because they have no brushes or commutator to wear out, an ongoing problem with standard DC motors.

3.2.2. Disadvantages

- positioning accuracy is limited by step size (unless a sophisticated microstep drive is used).
- they have limited speed and acceleration capabilities compared to DC motors.[Kenjo]

3.3. Principles of Operation

Two basic types of stepping motors are in common use today: permanent magnet (PM) and variable reluctance (VR). For reasons of simplicity, this description is based on the PM type. As a background to some of the material on position detection, a short description of variable reluctance motors will also be given.

Figure 1 shows a simplified representation of a bipolar stepping motor. It is called a PM motor because its rotor is a permanent magnet. Although the motors in the manipulator are all unipolar, the conceptually simpler bipolar motor will be described in this discussion; unipolar motors will be dealt with in a separate section.

From figure 1, it is seen that current flow through one of the windings will force the rotor poles to line up with the excited stator poles. By exciting the windings in a specific sequence, the rotor is forced to line up in one orientation, then another, then another and so on; the motor is said to *step* from one position to the next. This is illustrated in figure 2, in which the windings are excited in the sequence AB, CD, BA, DC. BA means that current flows in the opposite direction to that in AB, causing the opposite magnetic polarity to appear on the stator poles. With this excitation sequence, the rotor rotates clockwise with steps of 90°. Reversing the sequence causes counter-clockwise rotation.

This operating mode is known as *wave-drive* or *single-phase-on* since only a single phase is excited at a time. Some other modes of operation, which provide various advantages, are discussed under *Alternative Drive Modes*.

3.3.1. Unipolar Motors

A unipolar motor works in the same way as a bipolar motor except that the stator's magnetic polarity is reversed using a second stator winding, rather than by reversing the current in a single stator winding. Each stator pole has two windings; forward current flow in one winding causes one magnetic polarity and forward current flow in the other winding causes the opposite polarity. The windings arranged in this manner are said to be *bi-filar* wound; the bipolar motor has *mono-filar* windings.

Doubling the number of windings on each stator pole requires thinner wire which in turn has higher resistance. For the same amount of torque producing magnetic flux, a unipolar motor must be driven with the same current as a bipolar motor with the same number of stator turns. The difference in winding resistance obviously results in the production of more heat in unipolar motors.

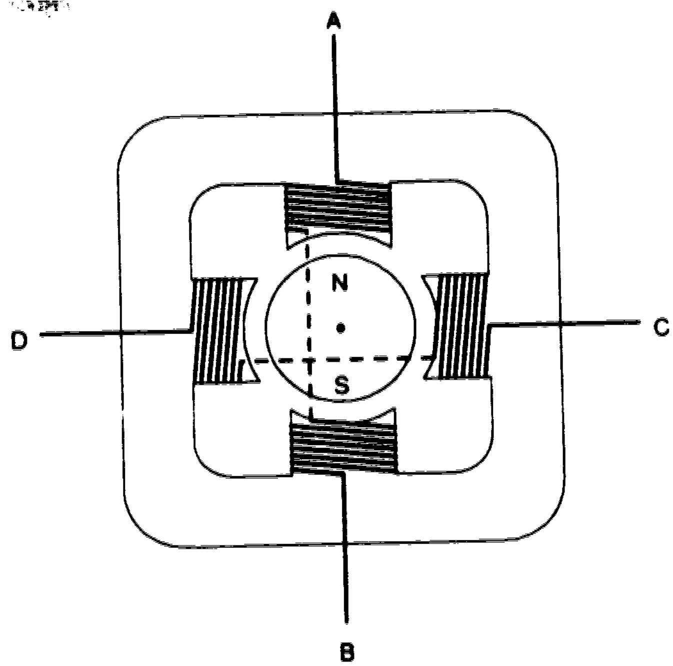

Figure 1. Cross-section of a simple PM stepping motor.[based on"*AN239*"]

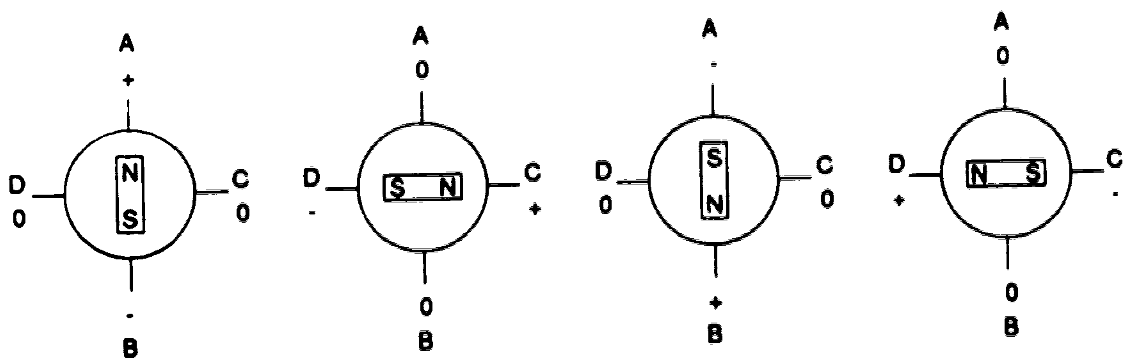

Figure 2. Wave drive of a PM motor.[based on "*AN239*"]

Since bipolar motors can carry higher current than unipolar motors without generating as much heat, they are capable of producing more torque. Although a natural limit to the available torque is imposed by magnetic saturation, mono-filar windings can produce twenty-five to thirty percent more torque than bi-filar windings.[Kenjo, Sax]

Despite these limitations, unipolar motors have been popular because of their apparently simpler driver requirements. A bipolar drive requires either an expensive split voltage supply (positive and negative) or a bridge circuit for each winding. Unipolar motors, on the other hand, can be driven using a single supply voltage and a simple transistor switch for each winding. Thanks to modern integrated circuit technology, the stepping motor system designer is no longer limited by these problems; completely integrated bridge drivers are available at reasonable cost.["AN239", Sax]

Increased torque can be obtained from certain unipolar motors by operating them as if they were bipolar; an explanation of how this is achieved is given in section 3.6.2.3.

3.4. Stepping Motor Structure

This section will show how various structures are used to produce steps and how step size is determined. The discussion addresses only a few of the many stepping motor structures in existence and its depth is limited to considerations useful in implementing a simple closed-loop drive system for the manipulator. Two basic stepping motor structures will be described: PM and VR. The descriptions of the motors will be followed by a discussion of how step size is calculated.

3.4.1. Permanent Magnet

The basic structure and operation of a PM motor were presented in section 3.3. This section presents some of the terminology used to describe the structure of the motor, then introduces the claw-poled motor.

Figure 1 shows a motor whose rotor has two poles and whose stator has four poles. Angular resolution can be improved by increasing the number of poles on both stator and rotor. A limit exists, however, on the number of poles a motor can have. The magnet is limited by the available materials and the stator by the physical size of the windings. Without sufficient turns, the motor cannot produce a useful torque.[Kenjo] Note that although the number of stator poles is increased, the number of windings need not change.

Another way to increase the resolution is to stack two or more stators; each stator is called a *stack*. Smaller steps are produced by placing one phase on each stack and offsetting the stacks. In a two stack motor, for example, the offset would be a quarter of a stator *pole pitch*, the angular separation between like poles (e.g. the separation between two north poles when a winding is excited). PM motors typically use two stacks, but some VR motors have five or more.[Kenjo]

From figure 1 it is clear that with the windings unexcited, the rotor will align itself with the stator poles. This is due to the permanent magnet rotor; the positions in which the rotor tends to align itself are called *détente positions*. If the motor is run in single-phase-on mode, its equilibrium positions are the same as its détente positions.

3.4.1.1. Claw-Poled Motor

A diagram of a claw-poled, also known as *can-stack* , motor is shown in figure 3. Since its construction is very simple, the claw-poled motor is inexpensive; it is produced by several manufacturers and can be found in many applications.[Kenjo, *Stepper Motor Handbook*]

From the figure, it can be seen that the stator consists of two stacks; each consisting of a coil cupped between two metal cans. The cans are stamped from a circular sheet of soft iron and pressed into a cup shape with a set of claw-like teeth forming the inside wall of the cup. These teeth become the stator poles in an assembled motor. Two of the cans are used to sandwich a coil (or if the motor is bi-filar wound, two coils) so that their poles inter-mesh. When current flows through the coil, alternating north and south magnetic poles appear on the stator teeth; one can's teeth become north poles and the other's south poles. A motor is made by stacking two of these sandwich structures such that the stacks are offset with respect to one another by a quarter pole pitch. In a bipolar motor, each stack contains one winding; in a unipolar motor, one stack contains phases *A* and *C*; the other *B* and *D*.

Figure 4 shows the magnetization of the rotor; it is cylindrical with alternating north and south magnetic poles parallel to its axis. This magnetization pattern is possible only through the use of a ferrite core, which can be magnetized in bands — with multiple sets of poles — instead of only at its ends. A 7.5° motor has a pole pitch of 30° which means that the rotor has twelve pole pairs. Increasing the number of poles beyond this is difficult [Kenjo], so 7.5° is typically the smallest step size

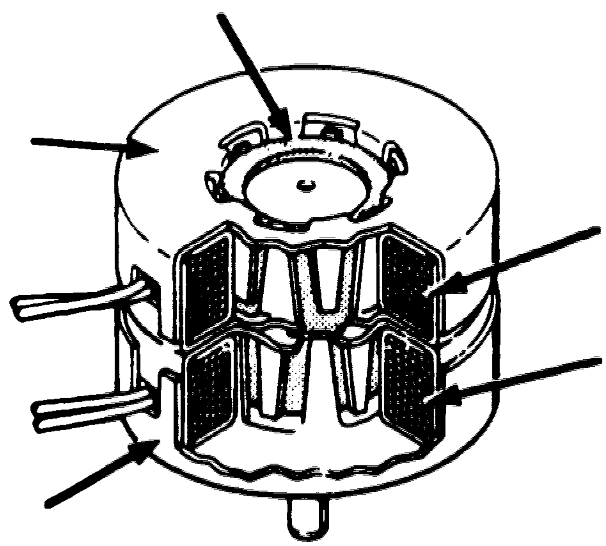

Figure 3. Cutaway diagram of a claw-poled motor.
[based on *"Stepper Motor Handbook"*]

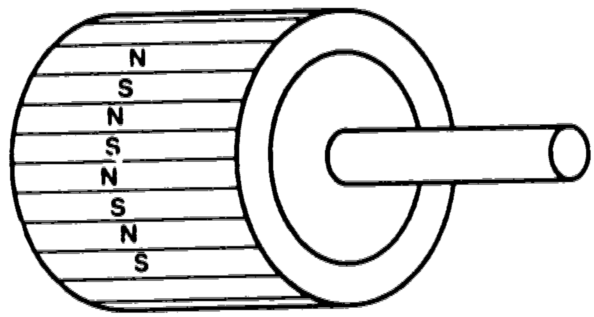

Figure 4. Rotor magnetization for a claw-poled motor.
[based on Kenjo]

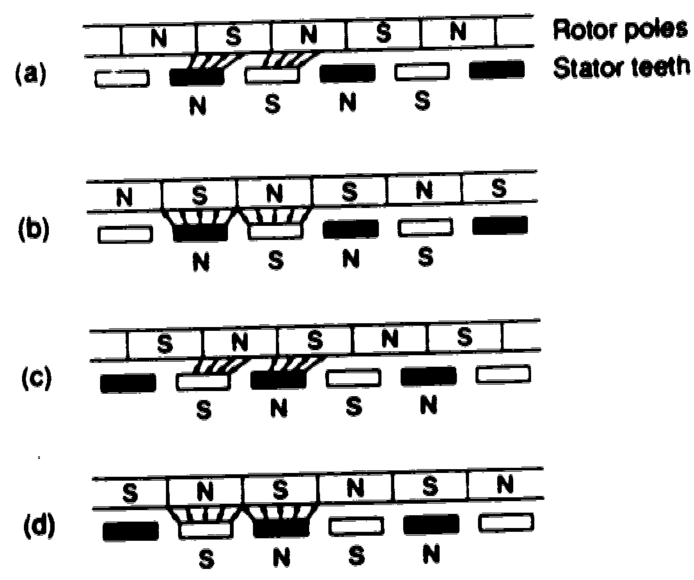

Figure 5. Step progression of a claw-poled PM motor. In (a), the rotor is out of position. In (b), the rotor has moved into position. In (c) and (d), the same operation takes place except that the winding excitation has changed state.[based on Kenjo]

available. If a smaller step size is required, a different type of stepping motor should be used.

When the rotor is inserted into the stator structure, the air gap between the rotor and stator is much smaller than that between the stator teeth, so the preferred path for magnetic flux is from one stator pole, through the rotor, to another stator pole. The number of poles on the rotor is the same as the number of poles on each stack of the stator, so in order for the motor to step, the stator stacks must be offset by a quarter pole pitch. Each time the phase excitation changes state, the rotor is drawn toward the next stator pole, a quarter pole pitch away. This is illustrated in figure 5. If the offset between the stacks is increased to a half pole-pitch, step direction is unpredictable; the rotor experiences forces from both directions.

Specifications for the AIRPAX 82701-P2 claw-poled motor used in the experiments are included in appendix A.

3.4.2. Variable Reluctance Motor

The *VR* motor was used in the initial development of the schemes for detecting rotor position using current waveforms.[Acarnley, Kuo 1977] This discussion of a simple VR motor is intended to clarify the concepts referred to in the discussions of those techniques.

Figure 6 shows the cross-section of a simple VR motor. This motor has three phases, each wound on two opposing stator poles in such a way that when current flows through the winding, opposite magnetic poles are induced on the pole faces. Magnetic flux flows from one stator pole through the rotor to the opposing stator pole.

Since the rotor is not a permanent magnet, its poles are defined by physically separated teeth which are needed for it to come to rest in particular positions. In the case of figure 6, the rotor has four teeth.

When a winding is excited, the rotor experiences a torque which tends to minimize the magnetic reluctance in the circuit, i.e. align one set of rotor poles with the excited stator poles. This process is shown in figure 7. A complete cycle of winding excitation states is shown in figure 8 in which the motor makes three 30° steps, rotating a total of 90° counter-clockwise.

This example illustrates why the motor is called *variable-reluctance*: as the rotor changes position, the reluctance of the magnetic circuit changes. When the rotor is fully aligned with the excited poles, the reluctance is at a minimum. The effect of rotor position on reluctance is utilized in determining rotor position using a technique introduced in sections 4.2.2. ff.

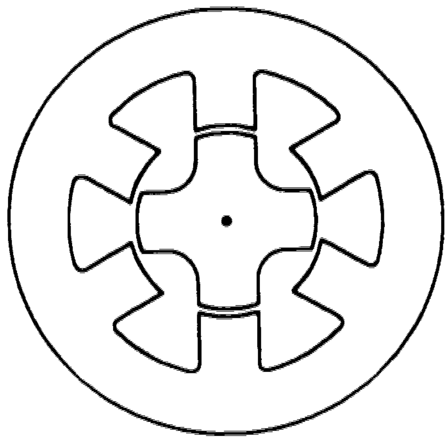

Figure 6. Cross-section of a simple VR motor.[based on Kenjo]

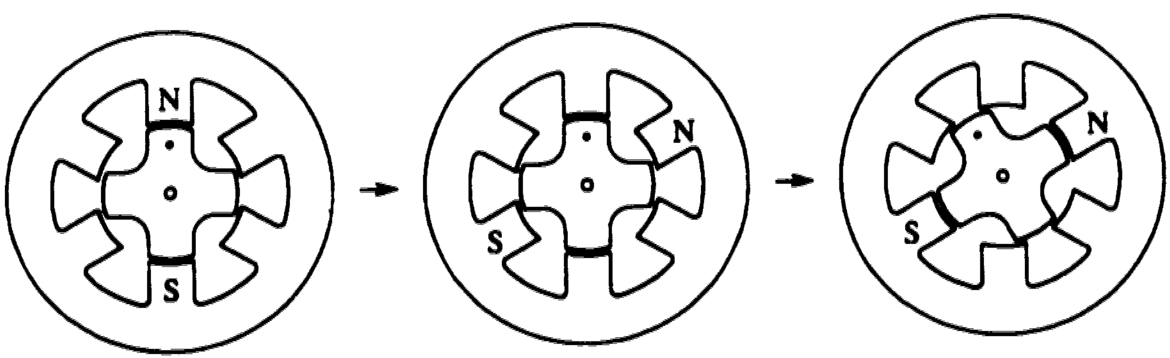

Figure 7. A single step. Winding excitation is switched from phase one to phase two at time t_0. At time t_1, the rotor has stepped to its new equilibrium position.[based on Kenjo]

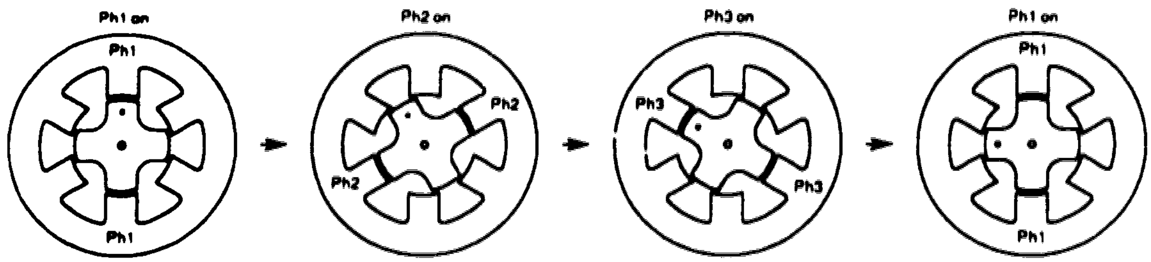

Figure 8. Sequence of steps for a three phase VR motor.[based on Kenjo]

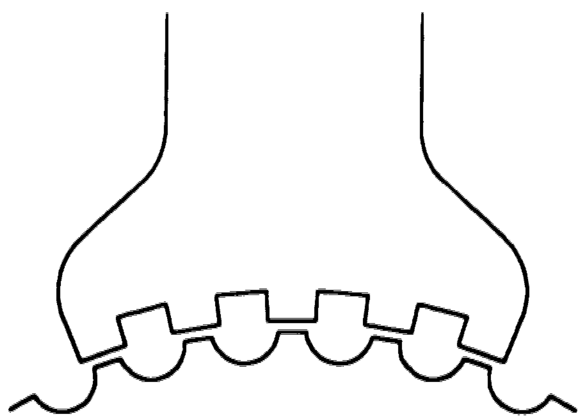

Figure 9. Close-up view of stator and rotor teeth. One stator pole is shown.[based on Kenjo]

In a single stack VR motor, the stator and rotor have a different number of poles; were this not the case, no torque would be generated, the rotor would simply be held in one position. A multiple stack motor usually has the same number of poles on stator and rotor; in this case, the stacks are offset with respect to one another, just as with the claw-poled motor.[Kenjo]

To reduce step size, VR stator poles are often constructed with teeth, as shown in figure 9. For a motor with a toothed stator, the rotor teeth are evenly distributed around the circumference of the rotor.

Maximum torque production again requires the smallest possible air-gap between stator and rotor. With a smaller air-gap, the same magneto-motive force will produce more flux and hence more torque.[Kenjo]

3.4.3. Hybrid Motor

Hybrid motors combine the operating principles of the VR and PM motors. Since this type of motor is not used in this application and its structure offers no additional insight into the feedback schemes considered here, it will not be discussed further.

3.4.4. Step Size

For reasons of clarity, the discussion in *Principles of Operation* was based on a simple 90° permanent magnet motor. Although 90° motors exist, they are not common and most stepping motors have smaller step sizes.

For a two phase bipolar motor, a smaller step angle is realized by increasing the number of poles on the stator and rotor; the number of windings is not changed. To decrease the step angle from 90° to 45°, for example, the number of stator and rotor poles is doubled.[Kenjo]

For a bipolar stepping motor, the number of steps per revolution is given by [Kenjo]:

$$S = \frac{360}{\theta_s} = mN_r \qquad (1)$$

Where S = steps per revolution

θ_s = step angle

m = number of phases

N_r = number of rotor poles

Several features of this equation require explanation: The number of phases, *m*, applies to a bipolar motor. A typical unipolar motor is said to have four phases, but since it is bi-filar wound, each pair of phases serves the same function as one phase of a bipolar motor; in the context of this equation, a unipolar motor is effectively a two phase motor. Note also that the number of stator poles does not appear in the equation. If the number of rotor poles is increased, the number of stator poles must be increased correspondingly in order for the motor to work.[Kenjo]

When the motor is run in a full-step mode, a complete sequence of winding polarity changes moves the rotor one pole pitch. For a two-phase motor, a winding excitation sequence consists of four states. The 90° motor in figure 2 has two phases, so four steps result in a total rotation of 360° which is one rotor pole pitch. Similarly, for the two-phase 7.5° claw-poled motor, a complete excitation sequence of four polarity changes results in a total angular displacement of 30°, again one pole pitch.

3.5. Alternative Drive Modes

In the operating mode discussed in *Principles of Operation*, only one phase of the motor was excited at a time. Improvements in motor performance can be obtained by running the motor in *two-phase-on* mode. Improvement of the resolution is possible using a *half-step* drive or a *microstepping* drive.

3.5.1. Two-Phase-on

As its name suggests, two-phase-on mode steps the motor by exciting two phases at a time. In this mode, rather than stopping directly adjacent to the stator poles, the rotor stops between poles; this can be seen in figure 10. The winding excitation sequence used to produce rotation for the motor in figure 10 is shown in table 1.

Two-phase-on mode is the most common drive used because it produces higher torque than single-phase-on mode.[Sax] More of the iron produces torque to move the rotor.

Perhaps even more significant for some applications is the fact that two-phase-on mode provides superior transient response. Both overshoot and settling time are significantly improved [Kenjo] making two-phase-on drive attractive for use with the manipulator.

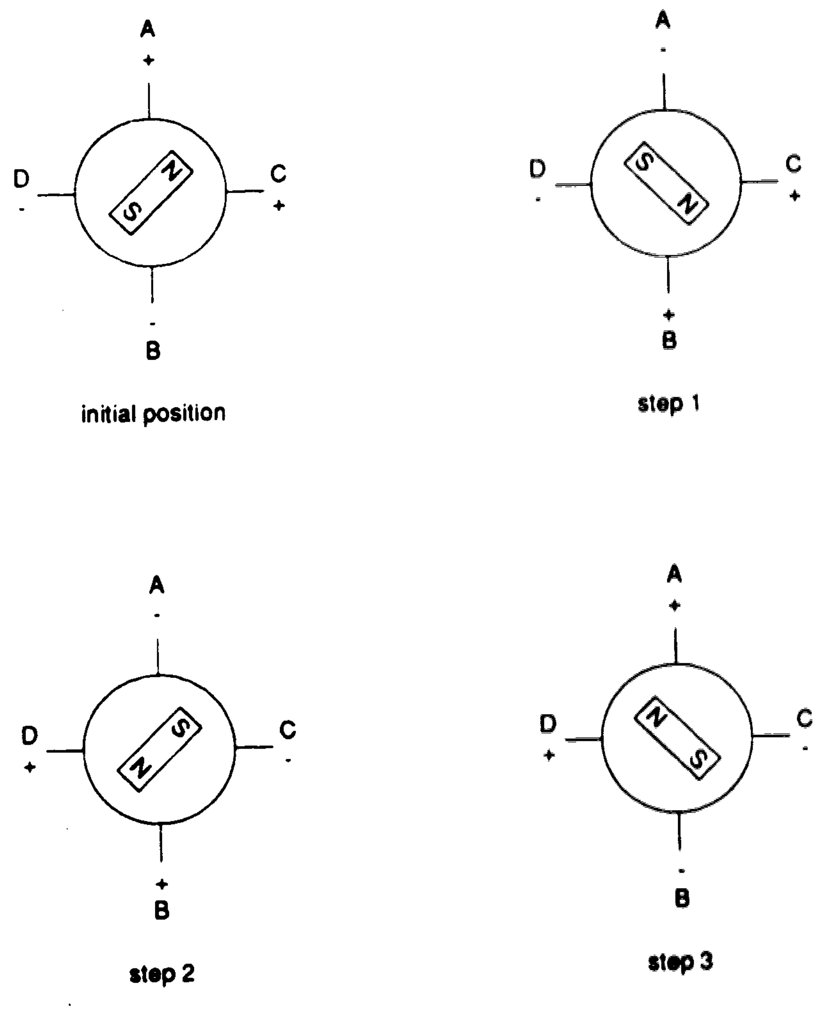

Figure 10. Two-phase-on drive of a PM motor.[based on "*AN239*"]

State	Phase				
	A	B	C	D	CW ↓ / CCW ↑
1	+	-	+	-	
2	-	+	+	-	
3	-	+	-	+	
4	+	-	-	+	

Table 1. Winding excitation sequence for two-phase-on operation of a two phase bipolar motor.

3.5.2. Half-Step

By combining the single- and two-phase-on drives, it is possible to double the resolution of the motor. In this mode, the rotor is stepped in half step increments by utilizing the equilibrium positions of both single- and two-phase-on modes; winding excitation alternates between the two modes.

Unfortunately, combining the two modes also combines the strengths and weaknesses of their performances which results in an irregular torque characteristic and transient response.

3.5.3. Microstepping

Rather than switching the windings either fully on or off, it is possible to allow only fractions of their rated currents to flow. By dividing the current between the windings in appropriate proportions, a single full step can be electronically divided into many smaller ones. Apart from vastly increased resolution (at least one manufacturer claims up to 20 000 steps per revolution [*Positioning Systems and Components*]) a motor run using this drive can accelerate and decelerate quickly and is capable of producing higher torque.

This method is only mentioned for the sake of completeness. It is not dealt with further because its implementation is complex and expensive and will not be considered for use in this application.

3.6. Electronics

Every stepping motor requires two circuit elements for operation: a sequencer, which controls phase excitation, and a driver, which performs the actual current switching. A block diagram of a typical sequencer-driver-motor arrangement is shown in figure 11.

3.6.1. Phase Sequencer

A sequencer is simply a digital state machine which changes its outputs in response to step and direction commands supplied by an external controller. Sequencers typically have a minimum of two inputs: step and direction. A pulse on the step input causes the winding control outputs to change to the next state in the sequence, using the direction indicated by the logic level on the direction input.

In the past, sequencers were constructed using discrete logic gates. Today, many varieties are available in integrated circuit form, some even complete with drivers for

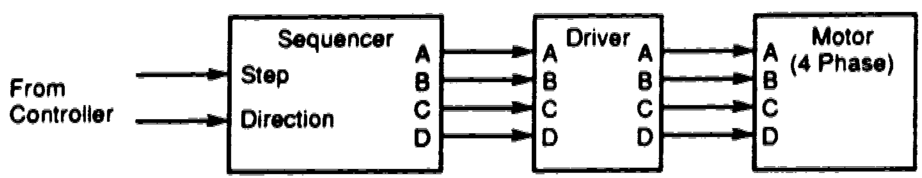

Figure 11. Block diagram of a complete stepping motor drive.

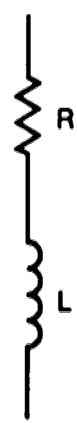

Figure 12. A simple phase model.

low current motors.[Harold] Some modern sequencers implement more than just a basic state machine; many include circuitry for control of a chopper-drive; the most sophisticated even implement micro-stepping.[Harold]

3.6.2. Drives

Stepping motor drives fall into three categories: constant-voltage, chopper and linear. This section deals with constant-voltage and chopper drives; linear drives, usually modified chopper drives, are used for microstepping and are of little interest in the context of this research.

Since the manipulator uses only four phase unipolar motors, the discussion will concentrate on drives for that type of motor.

3.6.2.1. Constant Voltage Drive

A constant-voltage drive, as its name suggests, applies a constant voltage to the excited windings. A simple model of the phase winding, shown in figure 12, shows that the winding has both inductance and resistance. The applied voltage is chosen such that at steady state, rated current flows through the winding.

Figure 13 shows a typical constant-voltage drive circuit for a four phase unipolar motor. Other circuits can be used, but this one provides good solutions to several important problems. Explanations of some of the circuit elements are given below.

Free-Wheeling Diodes

Without the so-called *free-wheeling* diodes, D_1-D_4, in the circuit, a damaging voltage spike, due to $L\,di/dt$ would appear at the collectors of the transistors whenever they are switched off. The diodes protect the transistors by providing a path for current decay.

Zener Suppression Diode

D_z, a zener diode, forces the free-wheeling current to decay quickly. If a mechanism to speed current decay is not included, the winding remains partially excited for an extended period, producing a braking torque.[Kenjo] D_z forces the current to decay quickly by making it work against the zener voltage (instead of only the much smaller forward voltage drop across the free-wheeling diode).

Forcing Resistors

The forcing resistors, R_1 and R_2, force the phase current to build rapidly when the transistor is switched on. This is necessary, especially at higher stepping rates,

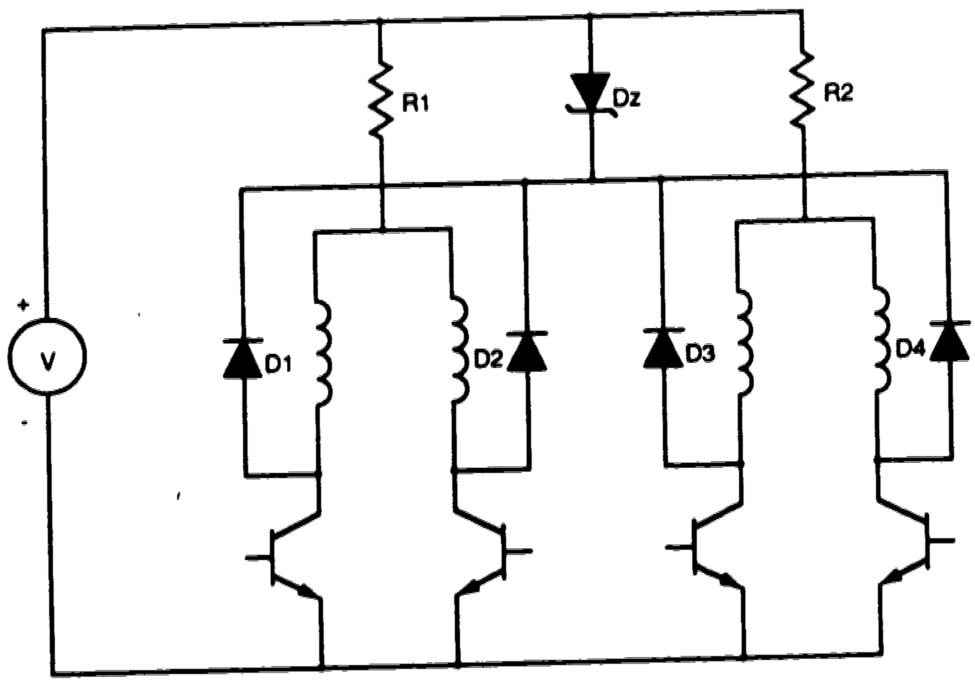

Figure 13. Constant-voltage drive circuit.[based on Kenjo]

because low current rise-time decreases the torque produced by the motor.[Kenjo] From basic circuit theory, it is known that the time-constant of an R-L circuit is governed by the ratio L / R . Current rise-time can clearly be reduced (hence torque improved), by increasing R; rated current is maintained by increasing the supply voltage correspondingly.

The main disadvantage of this solution is the increased i^2R losses caused by R_1 and R_2. Since the forcing resistance recommended for claw-poled motors is three times that of the phase resistance, which results in the so-called $L / 4R$ *drive*, the power lost through the forcing resistors is three times that lost in the motor (neglecting iron and mechanical losses). Though better ways of improving current rise exist, this method is popular because of its low production costs.

An excellent alternative to the constant-voltage $L / 4R$ drive is the *chopper* drive, which is presented in the next section.

3.6.2.2. Chopper Drive

Chopper, or switched-mode, drives ensure very fast current rise by applying a high voltage to the windings. Uninterrupted, the applied voltage would result in a current far above rated, but by modulating the current supply, the drive limits the average current to rated. Since the current rise-time is shortened by increasing the applied voltage, a forcing resistor is not required and the drive is considerably more efficient than its constant-voltage counterpart. Not only is less energy lost in the form of heat, the circuit enclosure can be smaller and simpler because it doesn't have to dissipate as much heat.

Another advantage available through the use of a chopper drive is that it permits a simple means of implementing a reduced holding current when the rotor is stationary. When the motor has reached its equilibrium position, it is no longer performing mechanical work, yet rated current flows through its windings and energy is lost as heat. For certain types of loads, rotor position can be maintained at equilibrium with only a fraction of rated current.[Kenjo] A reduced holding current is expensive to implement with other drives, which then require a second power supply. Since a chopper drive uses a reference voltage to regulate the amount of current , reducing the current is easily achieved by changing the reference voltage.

In the past, a disadvantage of chopper drives was their relatively complex circuitry; today, however, sequencers equipped with all the circuitry necessary to implement a chopper drive are readily available.[AN239, Harold]

An example of a chopper drive is given in section 4.2.2.2.

3.6.2.3. Bipolar Drive

With today's readily available integrated power electronic devices, a bipolar drive is only a little more expensive than a unipolar drive. A single package can contain all the necessary circuitry to run a two phase motor, including free-wheeling diodes.[Harold, Sax]

As mentioned earlier, by driving a bi-filar wound four-phase motor with a bipolar drive, a twenty-five to thirty percent increase in torque can be realized. This is because all four of the motor's windings conduct current at all times and thus produce torque at all times. An upper limit on the total torque available is due to saturation of the motor iron.

A bipolar drive for a unipolar motor is realized by connecting it in mono-filar fashion, in effect forcing a reverse current to flow in one winding of each pair at all times. This connection is only possible if all eight leads of the motor can be accessed.

Though called *bipolar*, this drive is often implemented with a single voltage supply using a bridge circuit to control the direction of current flow. An example of such a circuit is shown in figure 14. Note that with some modification, this drive can also be implemented as a chopper. In fact, a complete sequencer and bipolar chopper drive is available as a two chip set.["AN239"]

3.7. Control

Stepping motors are popular because they allow accurate positioning with open-loop control. Significant improvements in performance can be achieved by operating the motor as a closed-loop system. Block diagrams for open- and closed-loop stepping motor systems are shown in figure 15. While control in the open-loop system is based strictly on appropriate timing of step pulses, the closed-loop system uses knowledge of the rotor position to regulate switching.

Open-loop control is particularly well suited to applications where a constant load is moved in some repeated trajectory, such as the head in a computer disc drive. It is often economically advantageous to use a more powerful motor under open-loop control than to use a smaller and cheaper motor which requires expensive closed-loop control.

A more compelling reason for the use of closed-loop control exists when high motor performance such as high speed or smooth torque is desired. Another reason for using closed-loop control is found in robotics applications, where the motors used as actuators sometimes miss steps.[Kimura]

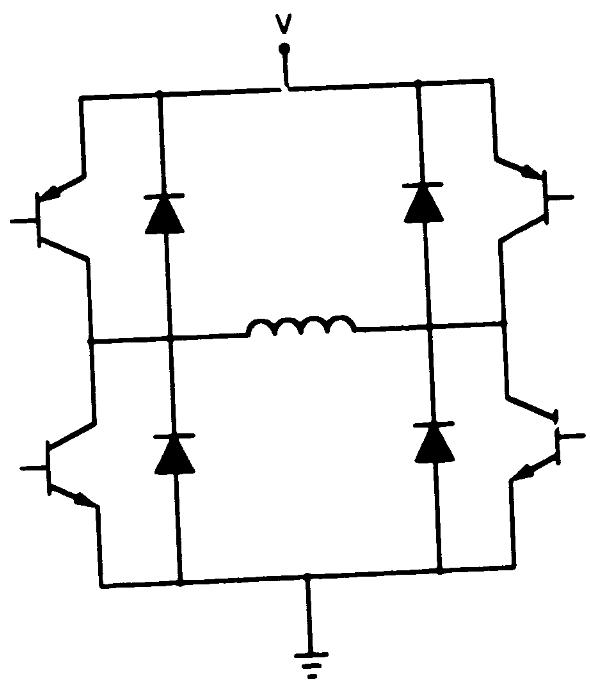

Figure 14. Bridge drive circuit for one phase.

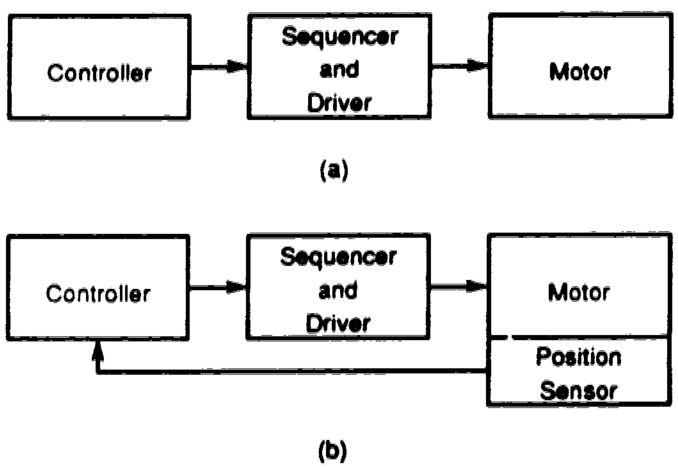

Figure 15. Block diagrams for open-(a) and closed-loop (b) stepping motor systems.

Since the manipulator is run using *learned* trajectories [Kimura], its closed-loop control will require only step confirmation; the application does not require the optimization of motor performance, which is also possible through closed-loop control.

This section presents some concerns encountered in both types of control.

3.7.1. Open-Loop

Under open-loop control, prior knowledge of motor characteristics is used to determine the pulse timing necessary to perform a given task. Some applications simply require step pulses at a preset rate. When speed becomes important, more sophisticated timing of the pulses is needed to ensure that the motor performs all of the steps requested.[Kimura]

For high speed operation, acceleration and deceleration must be gradual to ensure no loss of synchronization during starting or stopping. A stepping motor and its load are characterized by a starting step rate, above which the motor cannot start without missing steps, and a speed, below which the motor cannot stop abruptly, without several steps of overshoot.[Kenjo] These maximum rates are significantly influenced by load conditions; inconsistent loads, such as those encountered by robotic manipulators, impose a limit on the maximum acceleration and deceleration. A closed-loop system does not suffer from this limitation since acceleration and deceleration are controlled by the actual as opposed to assumed rotor position.

The method of gradually accelerating and decelerating the motor is known as *ramping*. Various timing circuits can be used to implement ramping, but maximum flexibility is possible using a microprocessor.

Even with ramping, experience has shown that at low stepping rates, the motion is still oscillatory and the motor can easily be stalled.[Kenjo] A method known as *back phasing*, a form of *dead beat* control, can be used as a means of electronically damping the rotor. During a step, when the rotor is already moving toward its new position, a braking torque is applied by commanding a temporary step in the opposite direction so that the rotor stops at the equilibrium position for the initial step; at this time, the winding excitation is again restored to hold the rotor in the new position. Clearly this technique is useful in open-loop systems only when the system dynamics are known so that rotor position can be accurately predicted; it is better suited to a closed-loop system where knowledge of rotor position is provided by a feedback mechanism.

3.7.2. Closed-Loop

Instead of waiting a fixed time before advancing to the next step, a closed-loop system can advance as soon as the rotor has reached a certain point in each step. As a result, the motor is always producing torque and the motion is quicker and smoother.[Kenjo]

By varying the position within each step that the phase windings are switched, known as the ***lead angle***, the controller is able to vary the motor's torque; speed is then determined by the load.[Kenjo]

Since the form of closed-loop control proposed for the manipulator calls only for step confirmation, the more sophisticated forms of closed-loop control will not be dealt with further.

3.8. Conclusion

This concludes the discussion of stepping motors and stepping motor systems. We have endeavored to describe how several motors work and the concerns encountered in determining the requirements of their drive. The remainder of this paper is a discussion of various forms of position feedback considered for use with the manipulator concluded with a discussion of the method chosen as a solution.

4. Rotor Position Detection

During the course of this research, several approaches have been considered for acquiring rotor position feedback from stepping motors. These include conventional position encoders, current detection methods and several types of position sensors. This section presents the methods considered; in each case, relevant theory and experimentation are presented along with an explanation of why each method except the last is unsuitable for use with the manipulator.

4.1. Conventional Position Detection Methods

Conventional methods of detecting angular position include such sensors as shaft mounted optical encoders and potentiometers. Since the motors in the manipulator have integrated reduction gear assemblies coupled directly to the motor shaft., there is no means by which to mount a conventional encoder. Furthermore, the space requirements of conventional encoders cannot be accommodated by the manipulator's present structure.

4.2. Current Detection

Current waveform detection uses the property that current drawn by the windings of a stepper motor is affected by the motion and position of the rotor; inspection of the winding current can thus provide information about rotor motion.

Current detection is an attractive form of feedback since all of the detection circuitry is located in the same enclosure as the drive circuitry instead of in the often adverse environment in which the motor operates. A further advantage with current detection is that it is fully electronic; no machined parts or mechanical coupling with the motor are necessary.

While current detection can be used effectively in high speed applications, its utility at low stepping rates is limited to step confirmation at best. Directional information is not readily available and detection of slipped steps is limited only to the number of steps slipped, leaving the controller with insufficient knowledge to make corrections.

Two approaches to current waveform detection have been investigated in this research. The first utilizes the effect of rotor *motion* on winding current (induced

EMF); the second utilizes the effect of rotor *position* on winding inductance which in turn affects the easily measured current rise-time.

Both of these methods were initially developed using variable-reluctance motors. Although variable-reluctance motors have significantly different electrical and magnetic characteristics than claw-poled motors, both methods merited further investigation.

4.2.1. Waveform Extrema Detection

In an R-L circuit, current rise in response to a step in applied voltage is exponential. If a stepping motor's rotor is fixed, its windings behave like simple R-L circuits, but due to the modulating effect of induced EMF caused by rotor motion, several detectable characteristics are present in the winding current.[Frus, Kuo 1977 and 79, Lin 1979b, Langley]

The simplest of these characteristics to detect was used by Langley and Kidd in a step confirmation system for a linear stepping motor.[Langley] The method was adapted for use with claw-poled motors, but it was found that it did not work reliably.

4.2.1.1. General Principle

When a stepping motor executes a step successfully, the current drawn by the most recently switched on winding exhibits a deflection or *dip*; if the motor is unable to execute the step (that is, no rotor motion takes place), the dip does not appear and the winding current closely resembles an exponential rise. Figure 16a shows the waveform produced when a step is executed successfully and figure 16b shows the waveform generated when a step does not occur. The dip changes position and size depending on the load. Step confirmation can be made by detection of this dip.

Detection of the dip can be accomplished by the method shown in the timing diagram in figure 17 and the circuit shown in figure 18. Each step command issued by the controller triggers a delayed exponential prototype waveform generated using a simple R-C circuit. The circuit has the same time-constant as that formed by the motor winding and it has a slightly lower final voltage than the current sense signal. The two waveforms are compared using a comparator whose output reflects which waveform has the higher voltage. If the dip occurs, the current sense signal crosses the prototype signal and the output of the comparator changes state, indicating the successful completion of the step. If the step does not occur, no crossing of the waveforms takes place and the controller detects a time-out which signals a missed step.

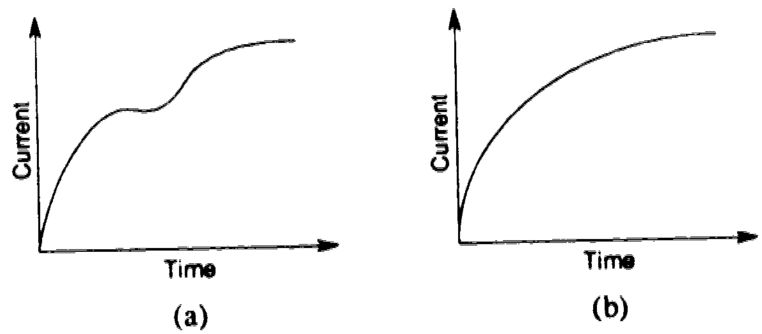

Figure 16. Current waveforms generated for successful (a) and unsuccessful (b) steps.[based on Langley]

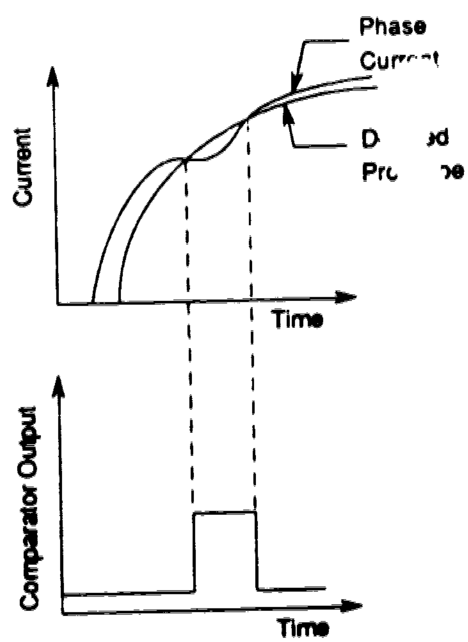

Figure 17. Timing diagram for the circuit of figure 18.

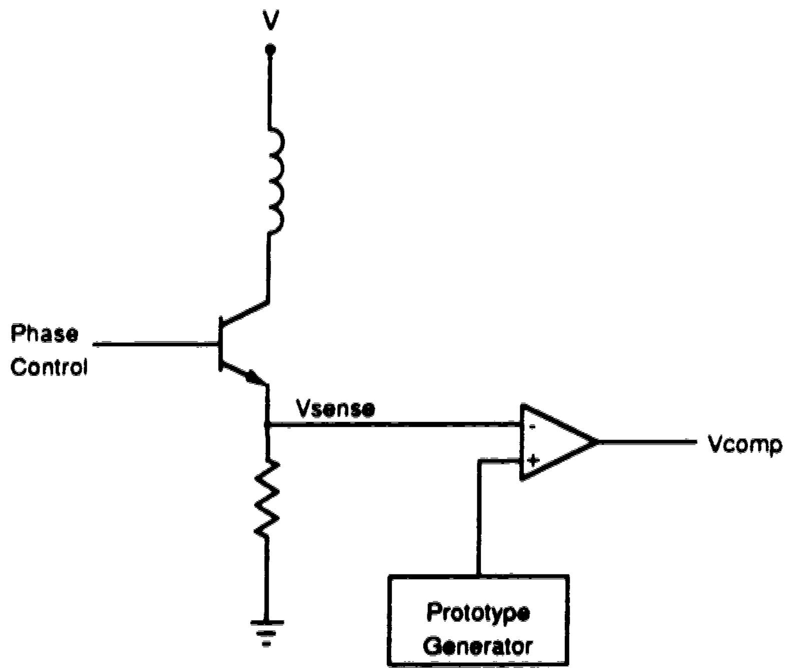

Figure 18. Circuit for step confirmation using dip detection.

4.2.1.2. Testing

Although dips in the winding current waveforms were observed using an oscilloscope, tests performed using the circuit shown in appendix B showed that the method is not reliable under certain load conditions. When the rotor was allowed to move quickly enough, the dips occurred and steps were properly detected, but as the load was increased and the stepping rate decreased, it was found that steps which did occur were not detected.

Since the dip is the result of induced EMF, which is proportional to the velocity of the moving magnetic field, slow rotor motion during a step can suppress or even eliminate the dip.

4.2.1.3. Conclusions

For reliable step detection using this method, the rotor must move fast enough during each step to produce a detectable dip in the current waveform. Observations have shown that at low stepping rates with heavy or high inertia loads, the dip becomes undetectable which makes this method unsuitable for robotics applications, in which loads such as these are expected.

Although other more sophisticated forms of extrema detection exist [Frus, Kuo 1977, Kuo 1979, Lin 1979b], they all depend on the effect of rotor motion on the current waveform [Acarnley, Hill] and are therefore unsuitable in this application.

4.2.2. Chopper Timing Detection

Another characteristic of winding current which is useful for detecting rotor position is based on the effect of rotor position on winding inductance; winding inductance in turn effects the rate-of-change of winding current when a voltage step is applied.

Since the method uses a characteristic of the current which is dependant primarily on rotor position instead of rotor motion, it was considered worthwhile to attempt its application to claw-poled motors.

Advantages to the use of this method include:

- Chopper drives are used, which provide improved performance and efficiency.
- The learning mode of the manipulator can be implemented using only slightly more complex circuitry than that used for step confirmation.

4.2.2.1. General Principle

An excited phase consists of a voltage source connected across a winding which can be modeled as a resistor in series with an inductance.(figure 19). A change in rotor position changes the magnetic path which affects the inductance seen by the voltage source. Since the time constant of an R-L circuit is dependent on inductance, the time constant of the winding circuit is dependent on rotor position. By monitoring the current rise- or decay-times during chopping, rotor position can be deduced.

4.2.2.2. Implementation

Before considering a theoretical explanation, an example of how this timing characteristic can be used may help the reader understand it.

Although various implementations of the scheme are suggested by the inventors [Acarnley, Amaratunga, Hill], this investigation is limited to the simplest one, namely: Detection of a specific rotor position in a step by inspection of current rise-time during chopping.

A typical chopper circuit for one winding is shown in figure 20. The drive uses a high supply voltage, which, if applied without some form of control, would cause a current several times I_{rated}, the rated current for the motor. By modulating the applied voltage, however, the drive limits the average current to the rated value. Referring to the timing diagram in figure 21: when the winding is first switched on, the current rises quickly to I_{rated}. When the current rises to a level slightly above I_{rated}, Q_2 is switched off and current begins to decay, flowing through the loop formed by D_1, Q_1 and R_{sense}. When the current has decayed to a level slightly below I_{rated} (an amount δi), Q_2 is again switched on and the current rises again. This cycle repeats itself as long as the phase is excited. When the excitation period is over, both Q_1 and Q_2 are switched off and the current decays through D_1 and D_2.

The rate at which the switching occurs is dependent on the rise and decay times of the current which will be shown to be dependant on rotor position.

Another popular chopper drive uses an external oscillator to switch on the supply; when the current has risen to the desired level, the supply is switched off and the current decays until the oscillator again switches the supply on. In this drive, the amount of current ripple is determined by the oscillator frequency.[Kenjo, Sax]

As illustrated in figure 22, when a step command is issued and the rotor rotates towards its new equilibrium position, the current rise-time during chopping first increases then decreases. When the rotor reaches its equilibrium position, the rise-

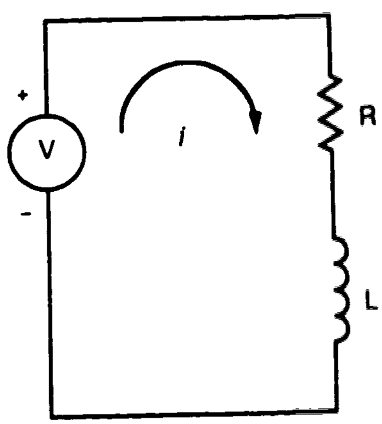

Figure 19. A simple phase model with a voltage source.

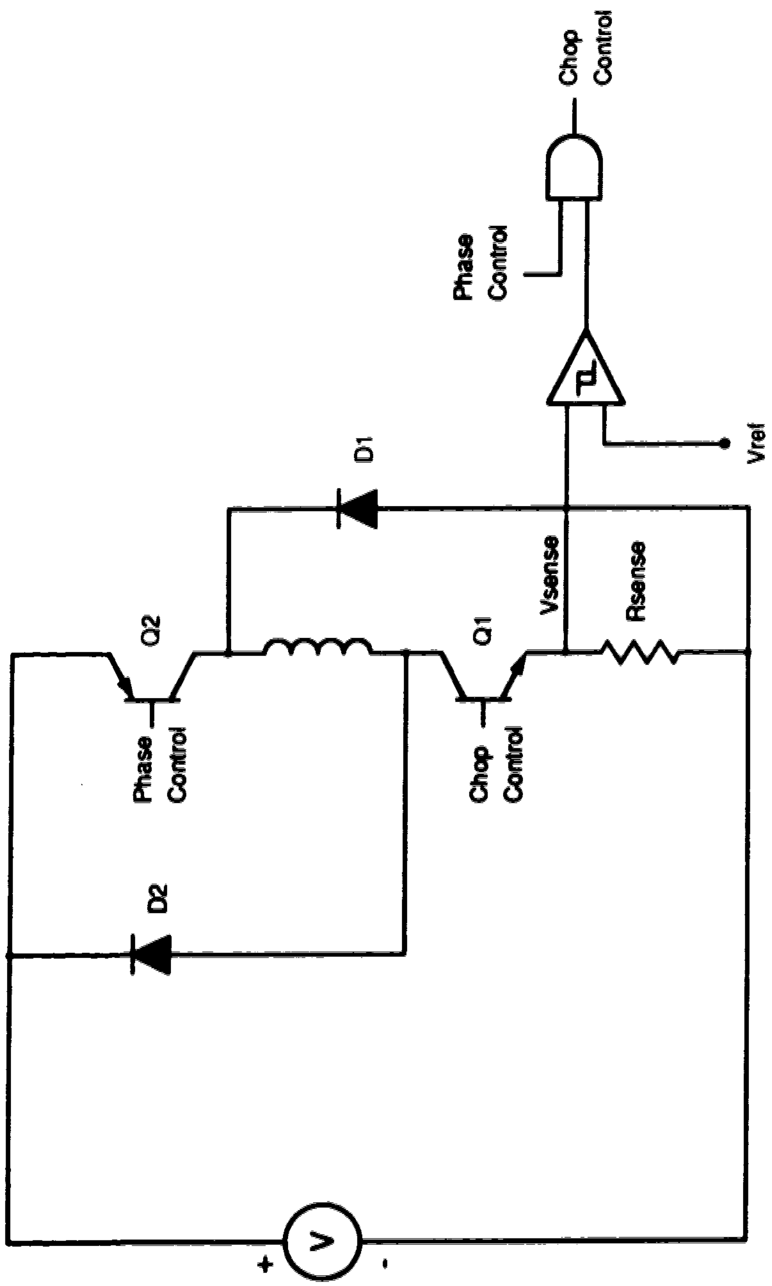

Figure 20. A chopper circuit. [based on Hill]

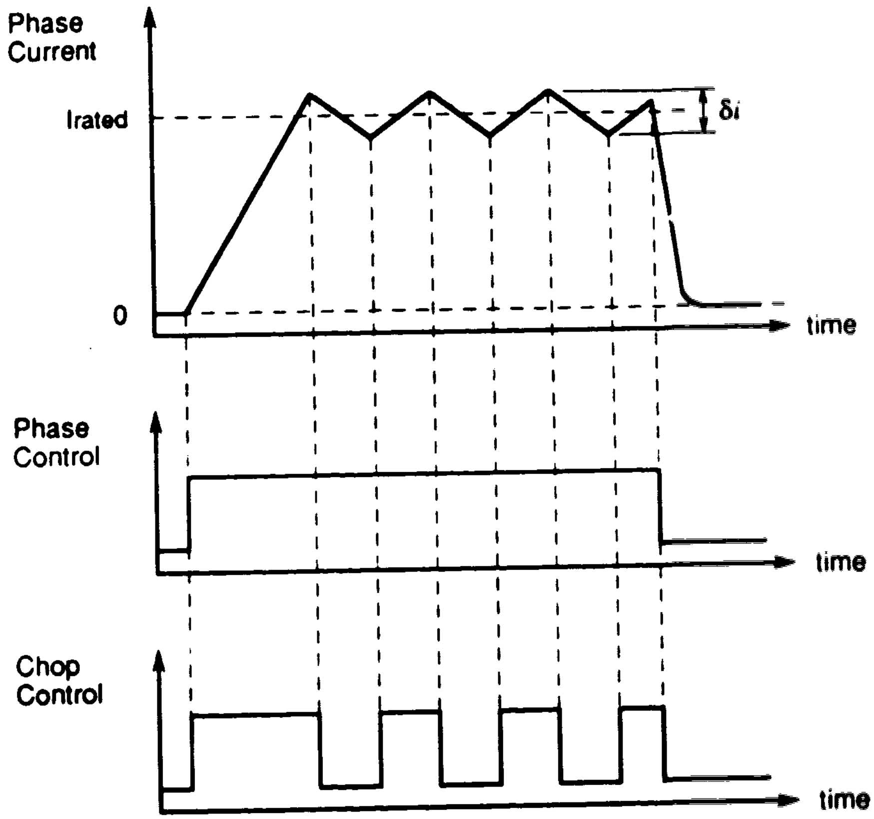

Figure 21. Chopper circuit timing for a single phase.

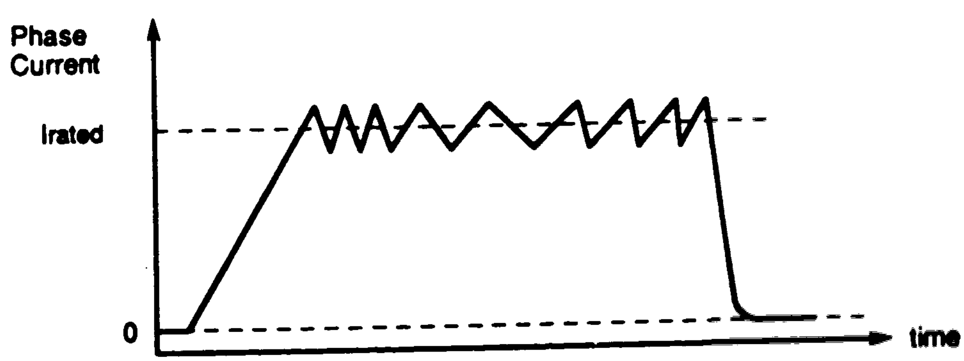

Figure 22. Chopped current waveform showing the variation of rise- and decay-times as a step occurs.[based on Hill]

time is at a minimum. This may not seem right since a maximum in inductance is expected when the rotor is at its equilibrium position and hence a maximum in rise-time; this apparent discrepancy is explained in the next section. By detecting a rise-time longer and then a rise-time shorter than an experimentally determined value, step confirmation is possible.

If the rotor is unable to move, the current rise-time remains constant, so that step confirmation is not made; a time-out can then be used to trigger an error signal.

A problem arises when the rotor slips in the direction opposite to the commanded direction. In this case, the rotor may reach an equilibrium position, but will be displaced from the desired position. Before stopping, however, a slipping rotor may pass through one or more equilibrium positions for the applied excitation. These positions would be interpreted by the detection circuitry as steps, which is obviously an error. Further investigation may reveal whether or not this situation can be detected.

A block diagram for a circuit to detect rise-times longer and shorter than a preset value is shown in figure 23a and the timing diagram associated with this circuit in figure 23b. When the rise-time has become longer and then shorter than the pre-determined value, the step has occurred and the confirmation signal is generated.

A circuit developed to implement this method is shown in appendix C. Observations made using the circuit and conclusions made on the basis of these observations are discussed in section 4.2.2.4.

4.2.2.3. Theory

Though this method was developed for variable reluctance motors, it is thought that it can also be applied to permanent magnet motors.[Amaratunga] Some assumptions used in the following discussion are more appropriate for variable reluctance motors, but they can also be applied to permanent magnet motors.

A variable reluctance stepping motor is designed in such a way that the rotor and stator teeth are in a state of magnetic saturation when they are aligned; in this state, the flux linkage for each winding is non-linear. When out of alignment (the magnetic circuit has a large air gap) or at low currents, the stator and rotor teeth are unsaturated and the flux linkage is linear and a function of current.[Hill] The rotors of claw-poled motors, since they are permanent-magnets, are constantly in a state near magnetic saturation, but the stator iron is not and the flux linkage for their windings still changes as the rotor moves. The relationship between flux linkage and current for a variable reluctance motor is illustrated in figure 24. From measurements made on

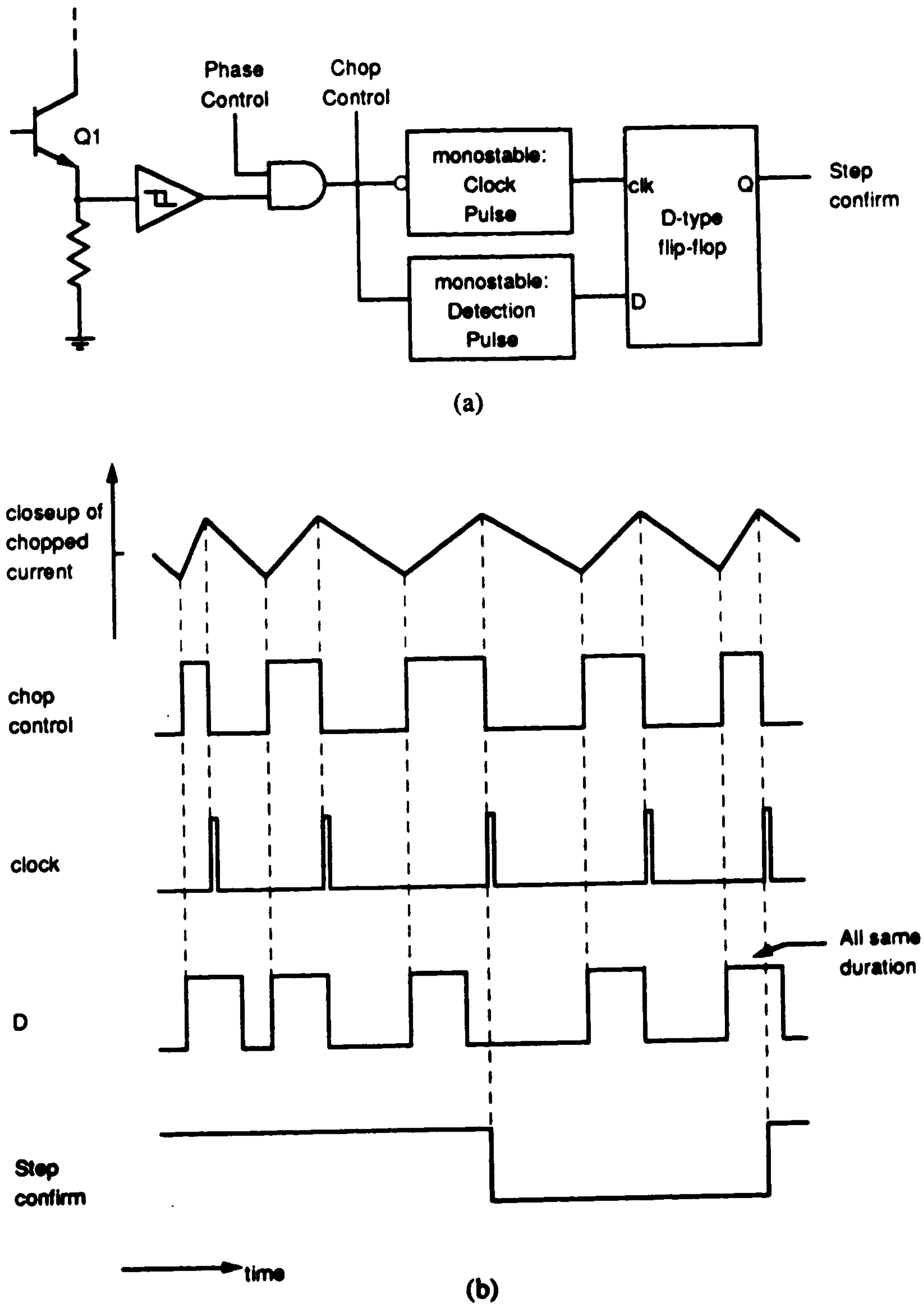

Figure 23. Circuit and timing diagram for step confirmation by monitoring current rise-time during chopping.[based on Acarnley]

their motor, Acarnley et al. show the effects of rotor position on incremental inductance. The results of these measurements are shown in figure 25 where it can be seen that the phase incremental inductance, $\ell = \partial\psi / \partial i$ (the change in flux linkage due to a small change in current), depends both on rotor position and current.

Assuming negligible mutual coupling between phases (though four phase claw-poled motors have a high degree of coupling between phase pairs, these pairs are never excited at the same time, so mutual coupling between excited phases windings is low), the voltage across the phase winding is given by [Acarnley]:

$$V = Ri + \frac{d\Psi}{dt} \qquad (2)$$

Where
$V =$ supply voltage
$R =$ phase circuit resistance
$i =$ instantaneous phase current and
$\Psi =$ flux linkage

Ψ is a function of i and θ, the position in mechanical radians, so

$$V = Ri + \frac{d\Psi}{\partial i}\frac{di}{dt} + \frac{d\Psi}{\partial\theta}\frac{d\theta}{dt}$$

$$= Ri + \ell\frac{di}{dt} + \frac{d\Psi}{\partial\theta}\frac{d\theta}{dt}$$

Rearranging,

$$\frac{di}{dt} = \frac{V - Ri - \frac{d\Psi}{\partial\theta}\frac{d\theta}{dt}}{\ell} \qquad (3)$$

At constant current and speed , di / dt is a function of incremental inductance which is in turn a function of position.

Neglecting the induced EMF term, which is much smaller than V at low speeds, equation (3) shows that during chopping, the rate of change of current is a function of the supply voltage, circuit resistance and incremental inductance.

During chopping, the current waveforms are small parts of larger exponential variations, so they can safely be treated as linear segments. Using this simplification, t_{rise} and t_{decay} can be approximated according to figure 26 as follows [Acarnley]:

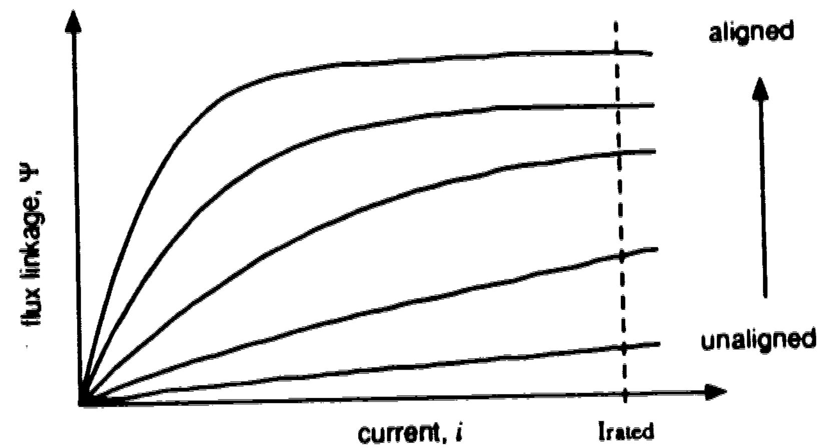

Figure 24. Flux linkage per winding versus phase current.
[based on Acarnley]

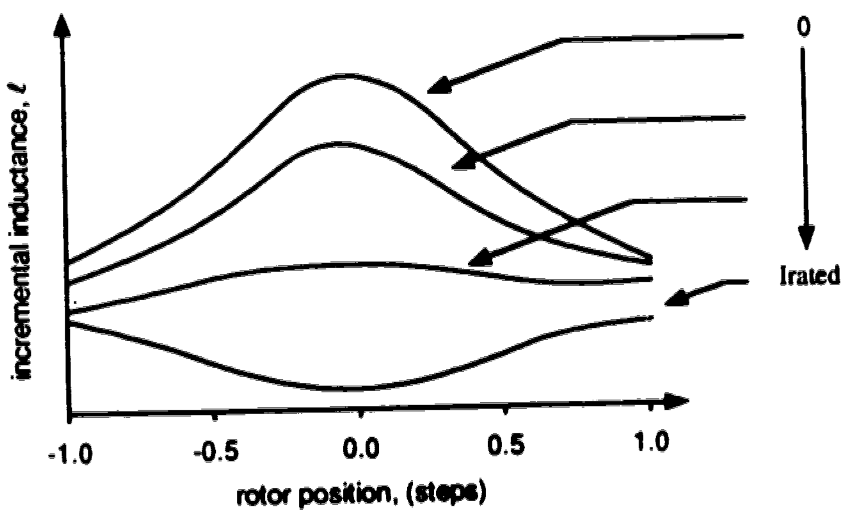

Figure 25. Incremental inductance as a function of rotor position.
[based on Hill]

$$t_{rise} = \frac{\ell \delta i}{V - Ri - \frac{d\Psi}{\partial \theta}\frac{d\theta}{dt}} \quad (4)$$

$$t_{decay} = \frac{\ell \delta i}{Ri + \frac{d\Psi}{\partial \theta}\frac{d\theta}{dt}} \quad (5)$$

Since the expression for t_{rise} has the supply voltage in its denominator, it is much less sensitive than t_{decay} to the effects of induced EMF; t_{rise} is therefore used for position detection. The dependance of t_{rise} on rotor position and the measured relationship between incremental inductance and position suggest the implementation discussed above.

A potential problem arises with the induced EMF term in the denominator of the equation. At low speeds, the effect of induced EMF is minimal, but as the rotor speed increases, the induced EMF term also begins to contribute to the equation. Figure 27 shows the effect of induced EMF on t_{rise} for a variable reluctance motor. Though claw-poled motors are unlikely to be used at speeds as high as 2000 steps per second [*Stepper Motor Handbook*], the effect of EMF could also be significant at lower speeds; since we chose not to pursue this method further, no tests were made to determine adverse effects.

Several alternative implementations using this feature of the current waveform are suggested in the literature. Namely *off-phase chopping*, in which the most recently switched off phase is observed (the example presented in section 4.2.2.2. used *on-phase chopping*), and inspection of the initial current gradient.[Acarnley, Hill] Since neither of these alternatives present any clear advantage over on-phase detection, and since both approaches require more complex circuitry, no attempt was made to implement them.

4.2.2.4. Testing

Using the circuit introduced above and a simple computer program, the following observations were made:

After careful adjustment of gains and pulse widths, the motor could be operated in closed-loop fashion in both directions. The resulting motion and torque were considerably higher and smoother than so far achieved with open-loop control.

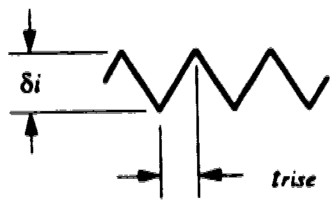

Figure 26. Current waveform with definitions of t_{rise} and δi.

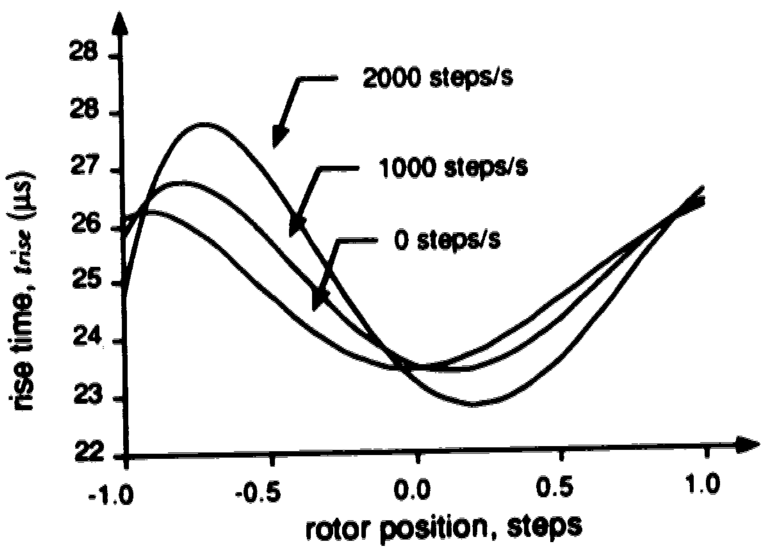

Figure 27. t_{rise} versus rotor position at various speeds.[based on Hill]

Unfortunately, the step confirmation signal was produced even when the rotor was prevented from moving. This problem persisted even after considerable effort was made to eliminate it by adjustment of the circuit; because of the complexities anticipated in overcoming this problem, the method was abandoned and no quantitative measurements were made.

4.2.2.5. Conclusions

Inspection of several waveforms in the circuit indicate several ways in which the circuit can be improved, but for several reasons, a different approach was considered first.

First, since t_{rise} varies only about one μs during a step, the circuit is very difficult to adjust. Even if step confirmation using this circuit worked, the difficulty in adjusting the circuit alone makes it a poor solution. Significant improvements of the circuit would have to be made before it would be easy to apply.

Papers describing the method claim t_{rise} ranges from four to twenty-five μs during a step.[Acarnley, Amaratunga] The difference between these results and those found with the claw-poled motor may be explained by the fact that whereas the rotor in a variable reluctance motor is unsaturated during part of the step, the permanent magnet rotor in claw-poled motors is almost continually saturated. Additionally, and considerably more significant, the winding inductance of the VR motor changes due to the changing air gap between stator and rotor poles; claw-poled motors, on the other hand, have a constant air gap, so the effect of rotor position on rise-time is due only to small variations in the magnetic path caused by rotor magnetization. Had it been possible to reproduce longer rise-times with claw-poled motors, adjustment of the feedback circuit would be simplified, but realization of the necessarily long rise times results in chopping rates under twenty kHz. At chopping frequencies in the audible range, claw-poled motors produce a disturbing whine; this is unacceptable for use around people.

Observations also indicate that there is no guarantee that the method can reliably detect missed steps at very low stepping rates. Also, since this method provides no indication of direction, slipped steps can be misinterpreted as successful steps. Although a computer software solution may exist for this problem, such software would likely be complex.

Though there are some indications that chopper timing detection can be used for closed-loop control of permanent magnet motors, further investigation of the method requires verification of the inventors' results using a variable reluctance motor;

following this verification, the same measurements could be made on claw-poled motors. Rather than investing significant additional time in pursuing this method, with no assurance of success, it was decided to find a simpler solution to the problem. Indeed, it has been found that the implementation of timing detection is considerably more complex than the simple encoder developed.

4.3. Shaft Encoder Alternatives

For reasons indicated earlier, conventional shaft encoders cannot be used with claw-poled motors which are fitted with gear-heads. To circumvent this problem, a shaft encoder was designed which is integrated directly into the motor. The initial approach was to sense the magnetic fields produced by the permanent-magnet rotor in much the same way that magnetic sensors are used to control commutation for brushless DC motors.[Klafter] As will be shown, this approach is impractical. A second approach, which proved successful, employs a sectored disc cemented to the rear face of the rotor; through sensing the light and dark regions of the disc with a type of optical sensor, both step confirmation and direction detection are possible.

4.3.1. Disadvantages With Respect to Conventional Encoders

This method is not readily capable of the high resolutions available from some conventional encoders. Since the goal of this project is to provide step confirmation, the resolution available from the encoder developed here is sufficient.

4.3.2. Disadvantages With Respect to Current Detection

Compared to current detection, this approach has several disadvantages, many of which are avoidable by integration of the encoder into the initial design of the motor.

With respect to current detection, this approach:

- will work only with claw-poled motors. Though claw-poled motors are commonly used and the concept could find many applications, current detection is, at least in theory, applicable to many types of stepping motors.[Acarnley, Kuo 1977, Langley, Lin 1979a] In this project, the primary objective is the development of a working system for a particular class of motors. It should be noted that there are also some claw-poled motors to which this method cannot easily be applied.
- requires modification of existing motors. To gain access to the rear face of the rotor, the rear cover of the motor must be removed. In the case of smaller motors, this cover consists of a thin metal plate glued to the rear stator cup and

its removal is trivial. Slightly more challenging, larger motors are encased in an aluminum housing, part of which must be removed; a circular portion can easily be removed from the housing using a lathe. Care must be taken, however, to prevent cuttings from falling into the housing and interfering with the operation of the motor.

- requires custom fixtures for use with motors of different sizes. The manipulator employs two motor sizes, each requiring a different sensor mount. Fortunately, both mounts can be similar, differing mainly in dimension. With current detection, one circuit design would be sufficient for use with all of the motors.
- is not all electronic, nor is it located away from the often adverse environment in which the motor operates.

4.3.3. Advantages With Respect to Conventional Encoders

With respect to conventional shaft encoders, this scheme:

- can be used with the manipulator's motors. Due to the gear train, conventional encoders do not have access to the motor's shaft and cannot be used.
- has no moving parts and is thus cheaper to manufacture.
- does not require the alignment of rotating parts. Since the assembly is not coupled to the shaft, critical mechanical alignment is not required.
- does not require a supporting structure; the assembly is fixed to the body of the motor. If conventional encoders were used, the manipulator would require extensive modification.
- has negligible effect on motor performance. Since the only load added to the motor is a small paper (or plastic) disc, no appreciable loading takes place. Any mechanical device coupled to the motor shaft, would affect its performance; this is particularly undesirable for small motors.

4.3.4. Advantages With Respect to Current Detection

With respect to current detection, this scheme:

- requires significantly shorter development time.
- has much simpler electronics and requires simpler software on the controlling computer. This in turn reduces the required processing power.
- provides for simple implementation of the manipulator's learning mode, without hardware modification.

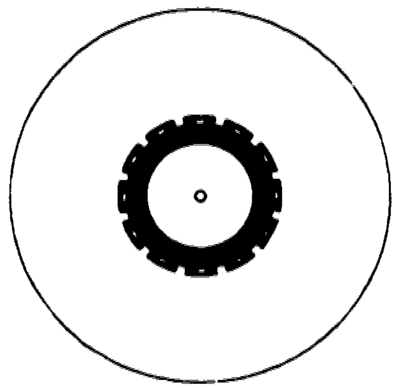

Figure 28. Rear view of motor showing rotor face.

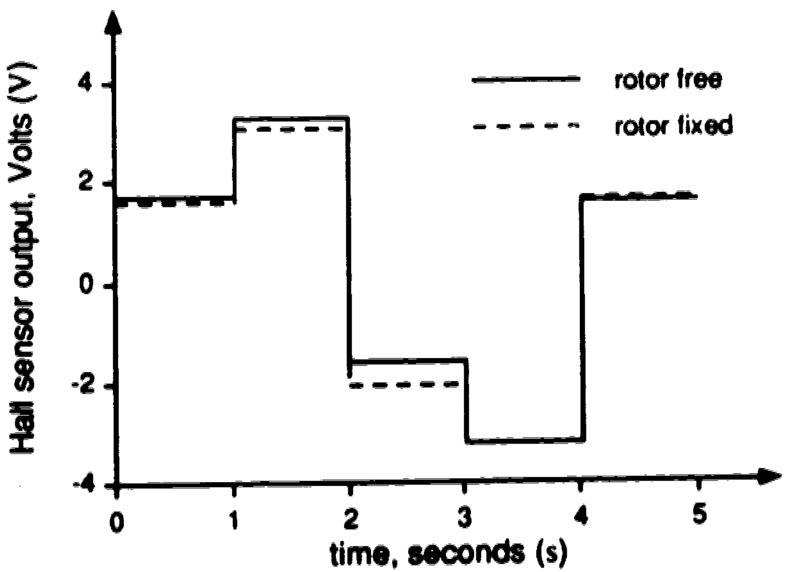

Figure 29. Linear Hall sensor output for five steps.

4.3.5. Magnetic Field Sensors for PM Stepping Motors

Removal of the rear cover of a claw-poled motor provides access to the rear face of the rotor as shown in figure 28.

The magnetic structure of the rotor used in claw-poled motors was discussed earlier in the section dealing with stepping motors. It was hoped to detect the magnetic pole faces on the exposed face of the rotor using magnetic field sensors. The rotor poles of a 7.5° claw-poled motor each occupy a 15° slice of the rotor. By detecting the magnetic polarity of the rotor under two sensors (displaced with respect to each other by an odd multiple of 7.5°), each step, including its direction, can be detected. For a discussion of how the detection of step direction is performed, see section 4.3.6.1.

Two methods of sensing magnetic fields were considered: small pick-up coils near the rotor and Hall Effect sensors.

4.3.5.1. Pick-up Coils

According to basic physical principles, a coil of wire with its axis parallel to a changing magnetic field has a voltage induced across it. The magnitude of the induced voltage is proportional to the rate of change of magnetic flux. Based on this principle, it was hoped that a small coil of wire placed near the rotor could be used to detect the changing rotor field as the rotor moved.

These sensors were not given much consideration for several reasons. As with the extrema detection method discussed earlier, the magnitude of the signal would be proportional to rotor velocity instead of position. Also, the analog signals would be subject to noise problems as a result of the harsh electrical conditions through which they would have to pass before reaching conditioning circuitry.

Testing

A coil sensor was built to determine whether or not the magnetic field could be measured. As long as the motor windings were unexcited, pulses could be produced by rotation of the rotor. When the motor windings were excited and the motor was stepped at a constant rate (to produce a high rotor velocity), the coil signal consisted of large spikes and noise produced by the switching stator field; the signal component due to rotor motion was visible, but undetectable by simple electronic means.

4.3.5.2. Hall Effect Sensors

Hall sensors, used to measure magnetic field strength, are a good alternative to pick-up coils. When a current carrying conductor is placed into a magnetic field, a voltage is generated perpendicular to both the magnetic field and the current. This principle is known as the *Hall effect* and the generated voltage as the *Hall voltage*. In a practical Hall effect sensor, a constant current is passed through a thin sheet of semiconducting material; the Hall voltage is measured across the edges perpendicular to the direction of current flow and is proportional to the perpendicular component of the magnetic field.[*Hall Effect Transducers*]

Commercially available Hall sensors have integrated signal conditioning circuitry; most have digital outputs, which indicate magnetic field strength above (or below) some threshold. The high noise immunity of these devices makes them ideal for this application. Linear Hall sensors are also available; their output voltage is linearly proportional to magnetic flux density.[*Hall Effect Transducers*]

Testing

A test was performed which showed that this approach is not practical. A linear Hall sensor fitted with an assembly to focus the magnetic field was used to measure the magnetic field near the rotor face. A focusing assembly was required to direct the rotor's field to the sensor. Over a period of five seconds, the motor was stepped at a rate of one step per second. While the motor was stepped, data was collected at a rate of 100 samples per second. A plot showing the sensor output is shown in figure 29; the solid line shows the measured field strength when the rotor was allowed to step normally and the dashed line shows the field with the rotor held stationary. Although there is a difference between the two curves, this difference is too insignificant to allow for simple detection of rotor position. From this graph, it is clear that the dominant component of the magnetic field is produced by the stator. In order to sense rotor position using digital Hall sensors, the contribution of the rotor field would have to be the dominant feature; a sensor could then be selected which would react to the rotor field without being affected by the stator. Linear Hall sensors could be used, but would require more sophisticated electronics and software than required for the optical sensors discussed below.

The circuit used to collect the data for figure 29 is included in appendix D.

4.3.5.3. Conclusions

From the experiments with coil and Hall sensors, it can be concluded that because of interference from the stator field, detection of rotor position by measurement of its magnetic field is not practical.

4.3.6. Reflective Sensors and a Sectored Disc

A reliable step detection system has been implemented using a sectored disc and reflective sensors. Its implementation is conceptually identical to that used for conventional optical shaft encoders, but due to the low resolution demands of this application, it was possible to integrate the encoder directly into the motor. The sectored disc, shown in place in figure 30, which consists of alternating light and dark regions is cemented to the exposed rear face of the rotor. Two reflective optical sensors sense the light and dark areas in an arrangement which provides direction information. Details of the implementation, along with test results, are provided in the next sections followed by some recommendations for the use of the encoder with the manipulator.

4.3.6.1. Implementation

Direction Detection With an Incremental Encoder

For the purpose of simplification, the following discussion will use an analogy to a linear incremental encoder. Operation of this encoder is identical to that of a rotary encoder except that it detects linear motion instead of rotational motion. The sensor positions relative to a striped tape are shown in figure 31a. In this analogy, a displacement of the tape by λ corresponds to one motor step.

As the tape is moved to the right (which corresponds, for example, to clockwise rotor rotation), the sensor outputs change according to figure 31b. For every displacement λ, one sensor switches. With the sensor arrangement shown, in which the sensors are placed ninety electrical degrees apart, the encoder can detect displacements of one half the width of each stripe.

For the rotational encoder, the striped tape is replaced with a sectored disc. Just as each stripe on the tape is twice the width of the desired resolution, each sector of the disc occupies twice the step angle. Since the motors used in the manipulator all have a 7.5° step angle, the disc is divided into fifteen degree sectors. With this arrangement, for every 7.5° rotation of the rotor, one sensor will switch, thereby making every step detectable.

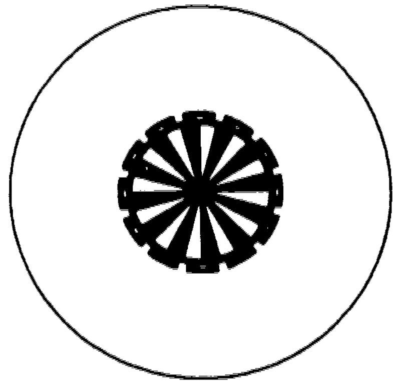

Figure 30. Rear view of motor with striped disc in place.

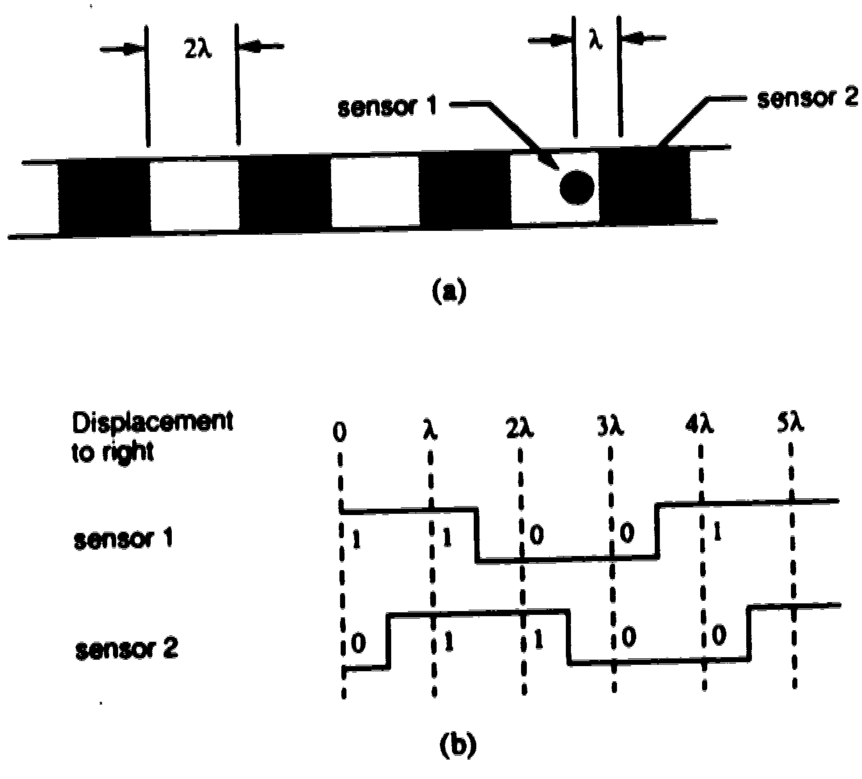

Figure 31. Striped tape and timing diagram for tape motion to the right.

It should be noted that the sensors must be separated by ninety electrical degrees. This can be accomplished with a physical separation by any odd multiple of λ (or 7.5° for the rotational encoder).

A useful characteristic of the waveforms generated by the two sensors is that, depending on the direction of displacement (or rotation, in the case of a motor), one leads the other by ninety electrical degrees. In the case of figure 31b, the output of sensor one leads that of sensor two, signifying displacement to the right. This information can be used by a digital circuit to control an up/down counter, for example.

A simple way for a computer to deduce directional information using these signals is through the use of a look-up table. Whenever a step is detected, the new state of the sensors is read and compared to that expected for clockwise rotation. If the comparison fails, either counter-clockwise rotation is assumed or it is verified using a look-up table extended to include the counter-clockwise case. The second alternative, though more expensive computationally, provides a facility for the detection of some error conditions.

Note that the relationship between the sensors must be known in order to correctly determine direction. Ultimately, this problem can be solved by designing the controller software to learn the state sequence.

Example:

A sample look-up table, based on figure 31b is shown in table 2. The table contains the expected next state for motion of the tape to the right. If the state of the sensors (S1,S2) is (1,0) and the tape is moved right by λ, the new state becomes (1,1). The controller compares this new state with (1,1), the content of the look-up table at an offset determined by the old state ((1,0) corresponds to binary *10* or decimal *2* and can be used as a pointer to the third element in the table). In this case, the new state and expected state match and the computer concludes that the motion was to the right. Had the motion been to the left, the new state would have been (0,0) which is different than the expected state and motion to the left could be assumed.

Sectored Disc

Good results were achieved using a disc made of paper, with the pattern applied using a simple dot-matrix printer; a later version of the disc was printed using a laser printer. Final implementation, however, should employ a more durable material such as plastic or aluminum.

present state		next state (clockwise)	
S1	S2	S1	S2
0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	1

Table 2. Next state lookup table for determination of direction.

Since the rotor's rear face is circular, mounting a disc with the same diameter is not difficult. Due to the encoder's low resolution, mounting the disc slightly off-centre results in state changes at different points in the steps as the rotor turns. Experiments have shown that a slight misalignment does not adversely effect step confirmation.

The angular position of the disc relative to the rotor is irrelevant since the sensor position, adjusted using the sensor mount, is used for alignment.

Reflective Sensors and Electronics

The following discussion refers to the circuit diagrams included in appendix E.

Two of the circuits, the sequencer/driver and the strobe generator are simple utility circuits and are only given cursory treatment. The sequencer/driver uses an off-the-shelf sequencer and a constant voltage driver and serves only to run the motor. The strobe generator provides a short pulse after every state change; it latches the new data into the computer input port and generates an interrupt.

Signal conditioning for the sensor outputs is performed by a separate circuit; it is more complex than necessary in order to provide flexibility in adjustment, an affordable luxury for a laboratory prototype.

Reflective sensors use a light emitting diode (LED) and a photo-transistor to detect reflective surfaces. This structure is shown schematically as U1 in figure E.3. When the sensor is near a reflective surface, light generated by the LED is reflected back from the surface to the photo-transistor; the amount of light returned controls the sensor output voltage.

To make the sensor's analog output useful in a digital circuit, it must be converted to digital form. This conversion is performed using a comparator; hysteresis is added to avoid high frequency switching due to noise and rotor oscillations during settling.

To prevent the generation of erroneous states, the sensors must have very similar switching characteristics. To this end, the circuit allows adjustment of the LED current, reference voltage and hysteresis for each sensor. The LED current adjustment controls sensor sensitivity; the other adjustments control the sensor switching characteristic. Note that the LED does not require rated current for the sensor to work. Not all of these adjustments are required in a practical design.

Two types of reflective sensors are available: focused and unfocused. Focused sensors use lenses to focus the transmitter and receiver on a common point which gives them good selectivity and makes them well suited to this application. Due to availability and package size considerations, unfocused sensors were used for the prototype. Fortunately, the poor selectivity of the sensors chosen can be compensated for in the design of the sensor mount.

Sensor Mount

The sensor mount is made up of two parts: a body, which is clamped to the motor, and a holder into which the sensors are inserted. Drawings of the two parts are shown in appendix F. This arrangement was initially intended for use with a different type of sensor; a simpler and cheaper version can be designed for this application. Though the details of the mount will not be discussed here, several considerations should be noted.

Since the field of view of the sensors is wider than a single sector of the encoder disc (at least for small 7.5° motors), they do not sense a single sector in all equilibrium positions and thus their outputs cannot be used to generate logic signals. For example, if the field of view is 15°, the sensors detect either all black, all white or half black and half white when the rotor is in its equilibrium positions. Since half black and half white is an undefined state, the field of view must be limited so that at equilibrium positions, the sensors detect only a single sector. Poor selectivity is improved in two ways. First, the gap between the sensor and encoder disc is determined using the manufacturer's data for optimum performance (in this case, the gap is 0.050 inches or 1.27 mm). Second, the sensor is masked using a window which effectively limits its field of view to 7.5°, the resolution of the encoder. Although narrowing the field of view yet further may improve the switching performance of the sensors, 7.5° windows provide a convenient means for aligning the sensors with the encoder disc.

Encoder alignment is required so that the encoder operation is independent of direction and so that its output changes state for each step. When out of alignment, the sensors can change state twice for certain steps and not at all for others. Alignment of the encoder is performed by rotating the sensor assembly with respect to the encoder disc while the rotor is in an equilibrium position. The alignment mechanism must be made available in the mount design. Positioning is aided by the removal of a sensor from its holder and aligning an edge of the now visible slot with the border between a light and dark area.

Note that alignment should be made with the motor windings excited. This is to ensure that the rotor is at an equilibrium position. Although this is not as important if the motors are driven using single-phase-on drives (in which case the natural détente positions correspond to equilibrium positions), it is not likely that the motors will be used in that mode.

Selection of material for the mount is based on the following considerations:

- the material should not interfere with the magnetic fields produced by the motor.
- the material should not inhibit the ability of the motor to dissipate heat.
- the material should be lightweight so as to minimize loading of the manipulator.
- the sensors should be shielded from ambient light so that the encoder operation is not affected by its environment.

To satisfy these conditions, the mount body and sensor holder were made of aluminum and black plastic respectively.

4.3.6.2. Testing

Testing of the encoder was broken into two stages: verification of phase relationship and operation as an incremental encoder.

Verification and adjustment of the circuitry was performed by running the motor at 100 steps per second and observing the comparator outputs using an oscilloscope. Each channel was adjusted to provide a square wave output with 50% duty cycle. When these adjustments were complete, the waveforms showed the correct phase relationship.

To ensure that the phase relationship between the two sensors would remain correct under poor conditions, another test was performed in which the comparator outputs were sampled while the motor was stepped. Five steps were executed at one hundred millisecond intervals and the comparator outputs sampled at 2000 samples per second. Figure 32 shows the data collected during this test and shows that despite multiple state changes following some steps, the phase relationship between the waveforms remains correct. Multiple state changes are caused by rotor overshoot and oscillations during settling.

To make this an accurate test, it was performed under worst-case conditions. Any difficulty with the phase relationship will manifest itself when the state changes occur in rapid succession. Rapid state changes take place either at high stepping rates (which are not expected in this application) or during single step response as the rotor settles to its equilibrium position. The problem is exacerbated when the load is light and has low inertia; in short, no load. Further testing shows that although higher inertia loads cause a larger overshoot and longer settling time than low inertia loads,

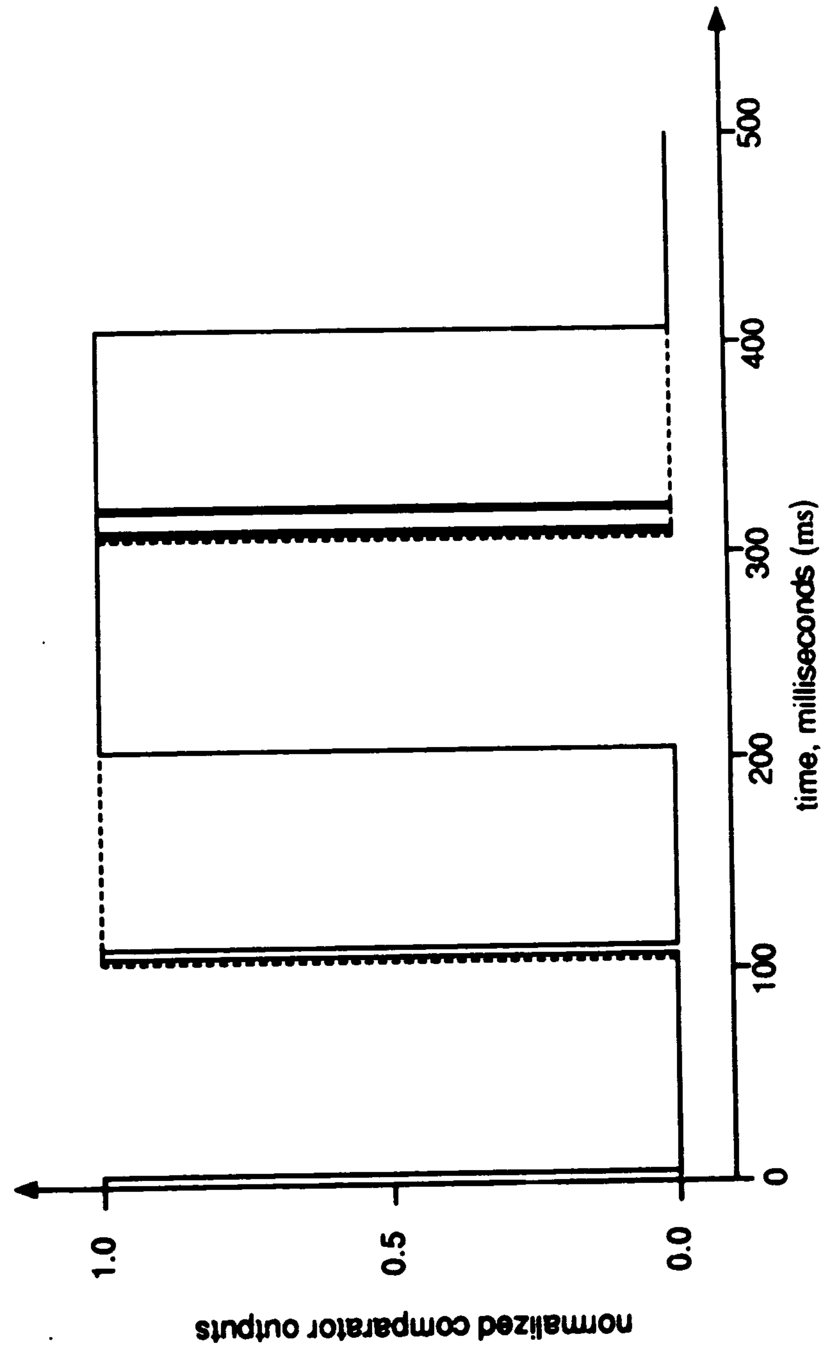

Figure 32. Normalized reflective sensor outputs for five steps clockwise. Note the correct phase relationship.

the frequency of the state changes during settling is lower. Thus the no load condition presents the worst case; for figure 32 the motor was run with no load.

In the second test, the encoder was used in the control of a motor. In this test, the motor executes a series of motions and then stops. The motions are performed under closed-loop control. That is, the first step is commanded by the algorithm and subsequent steps are triggered upon confirmation of the previous step. When the rotor comes to within a few steps of the target position, delays are inserted to cause the deceleration necessary to minimize overshoot. As long as the rotor is not in the desired position, i.e. at the setpoint, the controller attempts to move it to that position. This can be observed by displacing the rotor from its rest position or by preventing it from moving. In either case, the controller attempts to make the appropriate correction. Note that since stepping motor control is not within the scope of this project, the control algorithm used here is simple and only serves to test the encoder.

A listing of a program which implements the algorithm described above is included in appendix G. The program steps the motor through a trajectory and then holds it in a final position until terminated.

To avoid missing steps, step detection is performed by an interrupt driven routine which updates a counter and sets flags used by the main program. If an erroneous state is detected, the program increments a counter which is later displayed on the computer monitor.

Erroneous states occur when the detected state does not correspond to either clockwise or counter-clockwise rotation. These can result from actual sensor errors or if state changes are missed during interrupt processing. It is possible to generate erroneous states by spinning the rotor very fast or by holding it loosely so that it vibrates; the first condition is again not expected in this application and the vibration in the second is a condition brought on by the primitive control algorithm. Under normal conditions, no erroneous steps are detected.

Verification that the system behaves correctly was made visually using a flywheel attached to the rotor shaft. Before the program was started, a mark was made at a point on the flywheel. Since the program always stops the motor at multiples of a full rotation, the mark should be in the same position whenever the rotor is stationary. Whenever a disturbance was introduced, the rotor was returned to the correct position, indicating that the scheme works.

4.3.6.3. Recommendations for Implementation on the Robotic Aid

Implementation of the scheme on the manipulator requires the following:

•

- design and manufacture of mounts for the two sizes of motor used.
- design and manufacture of sectored discs for the two sizes of motor used.
- simplification of the signal conditioning circuitry.

Other considerations are:

- Simplification of the manipulator's wiring harness can be made by mounting the sensors on a circular printed circuit board which can in turn be mounted on the mount body.
- A simple solution for control of the manipulator can be realized through the use of a micro-controller for each joint. Although this may at first seem like overkill, micro-controllers have become cost effective alternatives to digital circuits. In this application, a separate controller for each joint will not only significantly reduce the computational load but will also simplify the software requirements for the controlling computer.
- A chopper drive is recommended because of its high efficiency and performance [Acarnley ,Kenjo, Sax]. Since all eight of the winding leads are available, improved torque can be generated using a bipolar drive. Were the micro-controller used as a sequencer, a task of which it is certainly capable, additional circuitry would be required to implement the chopper. If additional circuitry is needed anyway, a better solution is the use of a sequencer with built-in chopper control such as the one used the experiments.

A block diagram for the proposed system is shown in figure 33.

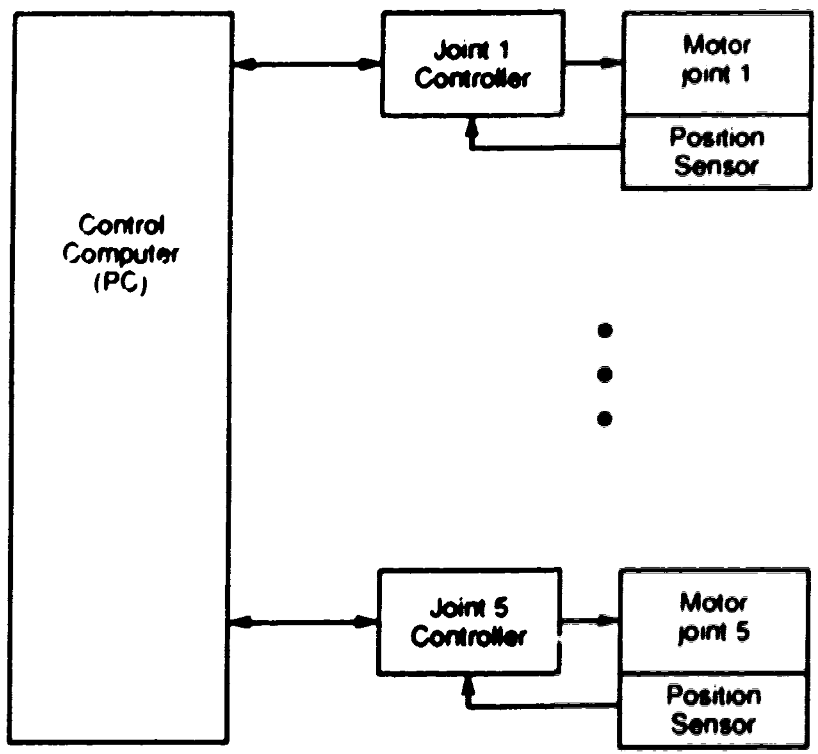

Figure 33. Block diagram of the manipulator with an individual controller for each joint.

5. Conclusion

Implementation of several step confirmation systems have been attempted. Of these, a simple variation of the ubiquitous incremental optical encoder has been developed as a solution to the problem of acquiring step confirmation for claw-poled stepping motors. With some additional design work, this scheme can be implemented on a manipulator as a position encoder for both step confirmation and learning. Looking further, the structure of claw-poled motors is such that a position detection scheme similar to the one developed here could be designed directly into the motor thus producing an inexpensive actuator capable of closed-loop operation.

References

Acarnley, Paul P., Roland J. Hill, and Clive W. Hooper. "Detection of Rotor Position in Stepping and Switched Motors by Monitoring of Current Waveforms." *IEEE Transactions on Industrial Electronics*, IE-32, (August 1985), 215-22.

Amaratunga, Gehan, Kin-Wah Kwan, M. Tso, and David Crawley. "A Single-Chip CMOS IC for Closed-Loop Control of Step Motors." *IEEE Transactions on Industrial Electronics*, 36 (November 1989), 539-44.

"AN239: The L297 Stepper Motor Controller." *Smart Power Application Manual*. Phoenix, Arizona: SGS-THOMPSON Microelectronics, 1989, pp. 131-47.

Frus, J.R., and B.C. Kuo. "Closed-Loop Control of Step Motors Using Waveform Detection." *Proceedings of the International Conference on Stepping Motors and Systems*, (1976), 44-53.

Hall Effect Transducers: How to Apply Them as Sensors. Freeport, Illinois: MICRO SWITCH, 1982.

Harold, Peter. "Monolithic Stepper-Motor Drivers Achieve Higher Power Levels and Greater Versatility." *EDN*, 21 Jan. 1988, pp. 69-80.

Hill, R.J., and P.P. Acarnley."Interacting Stepping Motor Systems: Closed Loop Control Using Waveform Detection." *Real Time Control of Electromechanical Systems*, 58 (March 1984), 43-8.

Kenjo, Takashi. *Stepping Motors and their Microprocessor Controls*. Oxford: Clarendon Press, 1984.

Kimura, Randall. *Design Study of a Robotic Arm for use by Quadriplegics*, M.Sc. Thesis, University of Alberta, 1987.

Klafter, Richard D., Thomas A. Chmielewski, and Michael Negin. *Robotic Engineering An Integrated Approach*. Englewood Cliffs, New Jersey: Prentice Hall, 1989.

Kuo, B.C., and A. Cassat "On Current Detection in Variable-Reluctance Step Motors." *Proceedings of the 6th Annual Symposium on Incremental Motion Control Systems and Devices*, (1977), 205-20.

Kuo, B.C., W.C. Lin and U. Goerke. "Waveform Detection of Permanent-Magnet Step Motors, Part II." *Proceedings of the 8th Annual Symposium on Incremental Motion Control System and Devices*, (1979), 243-56.

Langley, Lawrence W., and Howard Keith Kidd. "Closed-Loop Operation of a Linear Stepping Motor under Microprocessor Control." *Proceedings of the Southeastcon '79*, (1979), 13-7.

Lin, William Chin-Woei. *The Waveform Detection of Permanent Magnet Step Motors and its Application to Closed-Loop Control*, Ph.D. Dissertation, University of Illinois at Urbana-Champaign, 1979a.

Lin, W.C., and B.C. Kuo. "Waveform Detection of Permanent-Magnet Step Motors, Part I." *Proceedings of the 8th Annual Symposium on Incremental Motion Control System and Devices*, (1979b), 227-41.

Positioning Systems and Components. Harrison City, Pennsylvania: Daedal, 1989.

Sax, H. "AN235: Stepper Motor Driving." *Smart Power Application Manual*. Phoenix, Arizona: SGS-THOMPSON Microelectronics, 1989, pp. 95-110.

Stepper Motor Handbook. Cheshire, Connecticut: AIRPAX, North American Philips Controls Corp., 1989.

Appendix A

This appendix contains specifications for the AIRPAX 82701-P2 claw-poled stepping motor used in the experiments. More complete specifications can be found in the *Stepper Motor Handbook* which is available from AIRPAX.

DC Operating Voltage (V)	12
Resistance per Winding (Ω)	36
Inductance per Winding (mH)	38
Holding Torque (Measured with 2 phases energized)(mNm)	105
Rotor Moment of Inertia (gm^2)	3.4×10^{-3}
Détente Torque (mNm)	9.9
Step Angle (°)	7.5
Step Angle Tolerance (Measured with 2 phases energized)(°)	±0.5
Steps per Revolution	48
Maximum Operating Temperature (°C)	100
Ambient Temperature Range(°C)	
Operating	-20 to 70
Storage	-40 to 85
Mass (g)	281
Lead Wires (AWG)	26

Appendix B

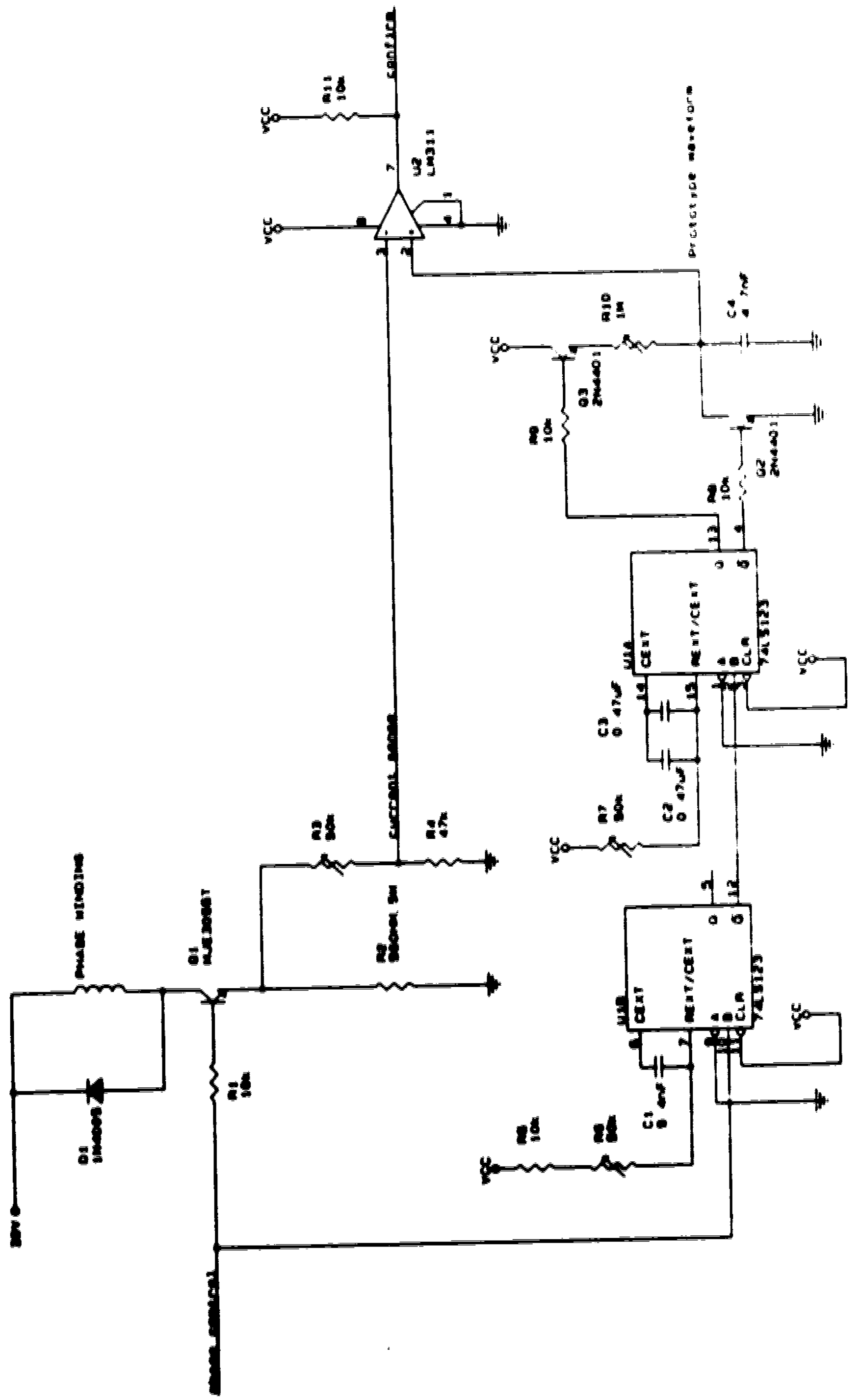

Figure B.1. Circuit for step confirmation using delayed exponential prototype.

Appendix C

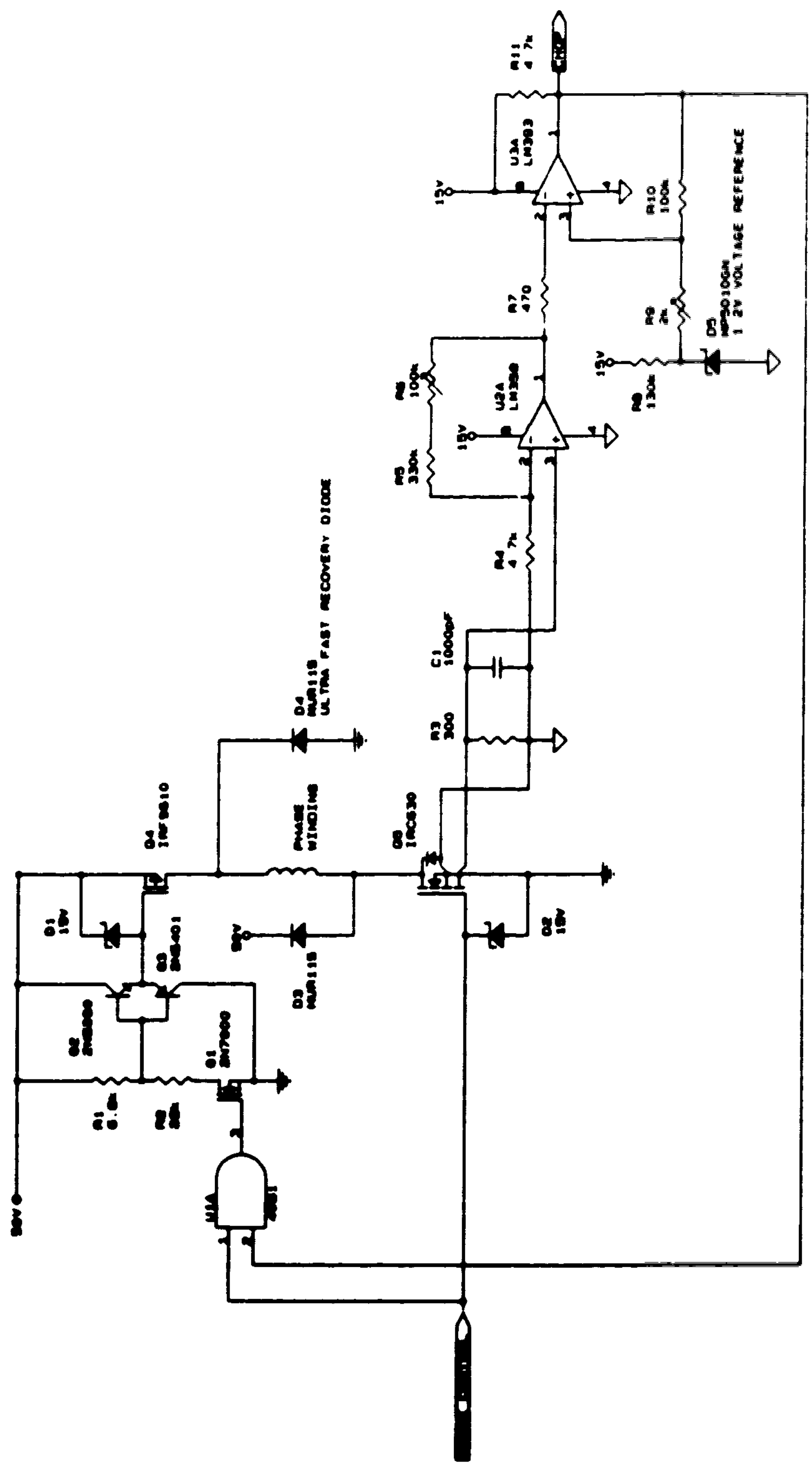

Figure C.1. Chopper circuit for a single phase.

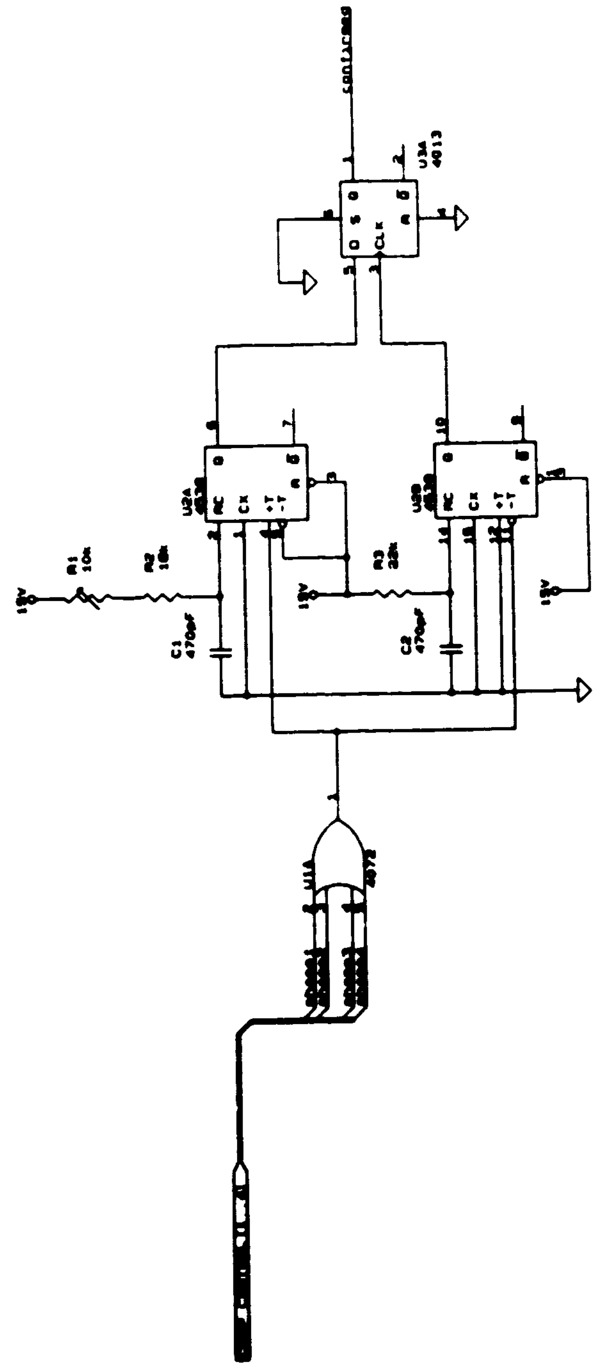

Figure C.2. Circuit for step confirmation using current rise-time detection.

Appendix D

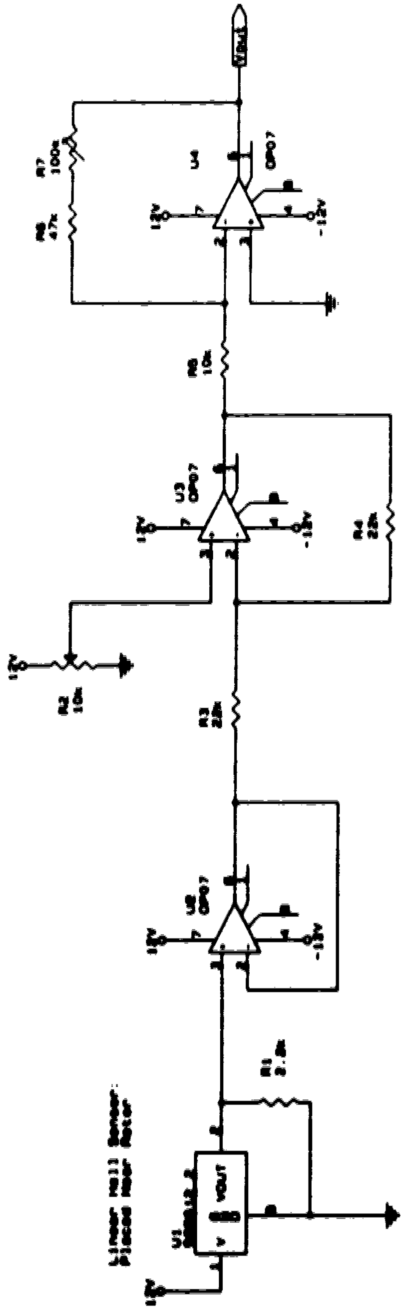

Figure D.1. Amplifier and offset correction for linear Hall sensor.

Appendix E

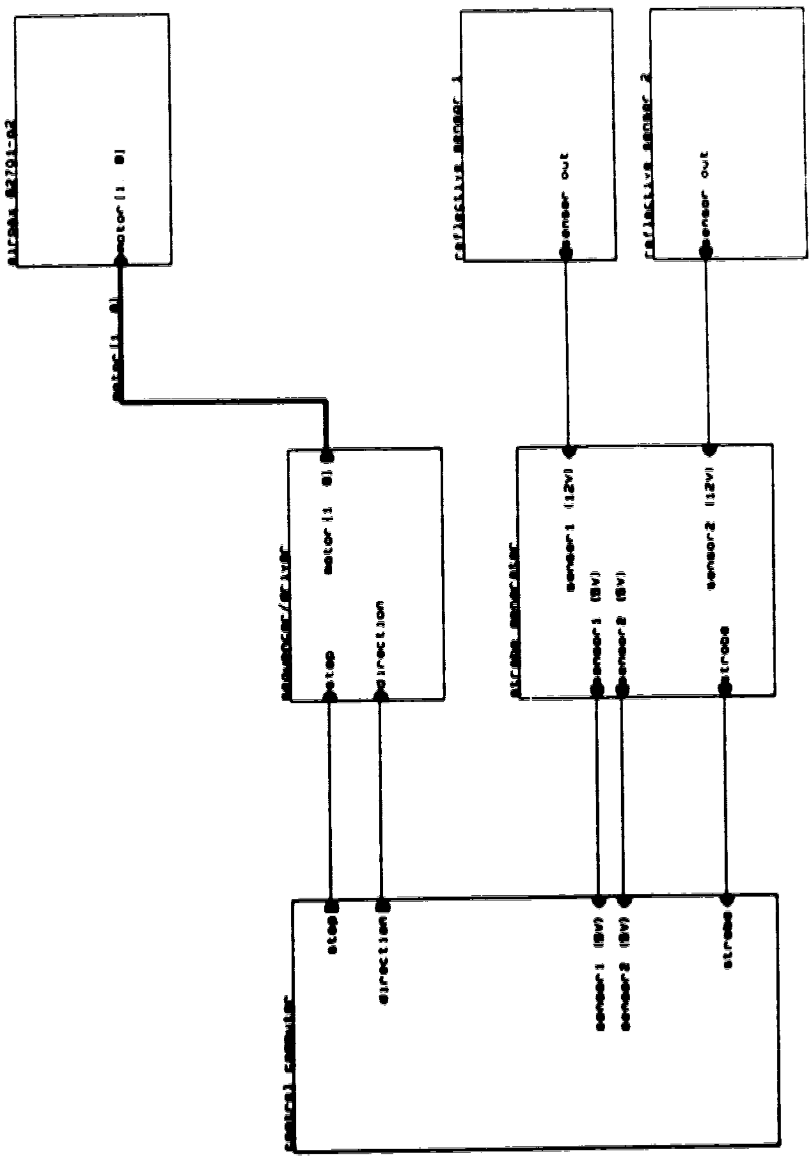

Figure E.1. Block diagram of circuit for reflective sensor position encoder.

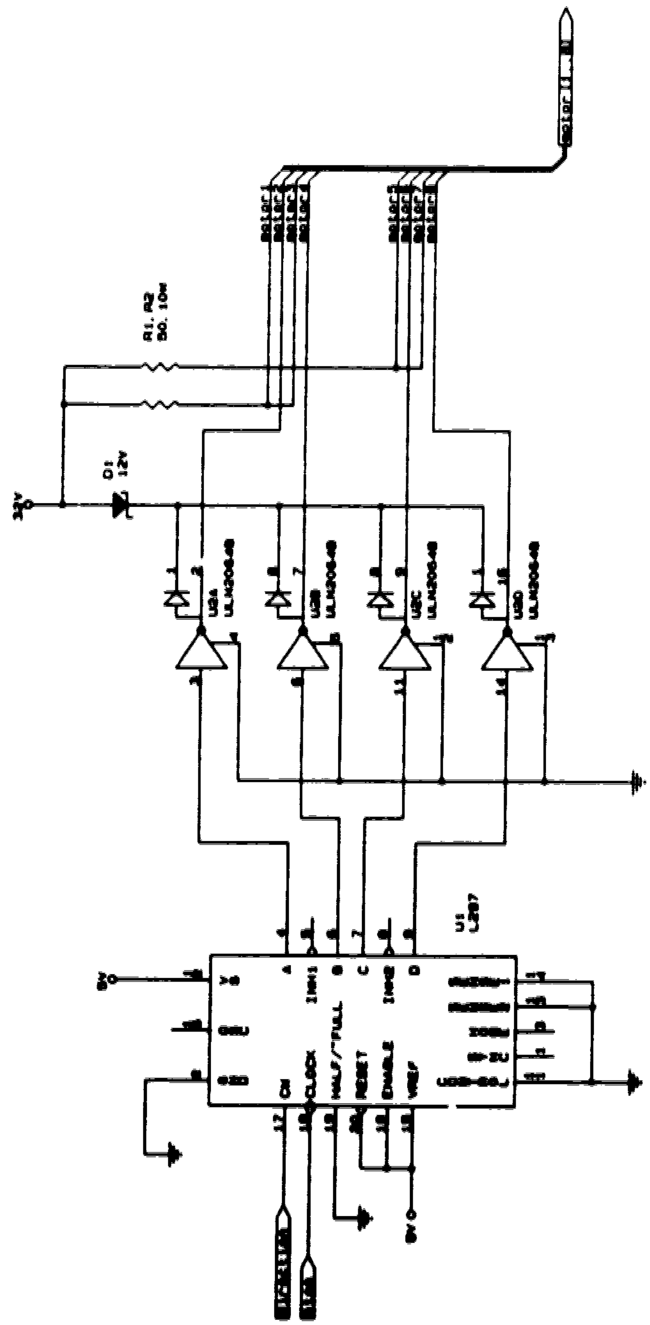

Figure E.2. Constant voltage drive circuit.

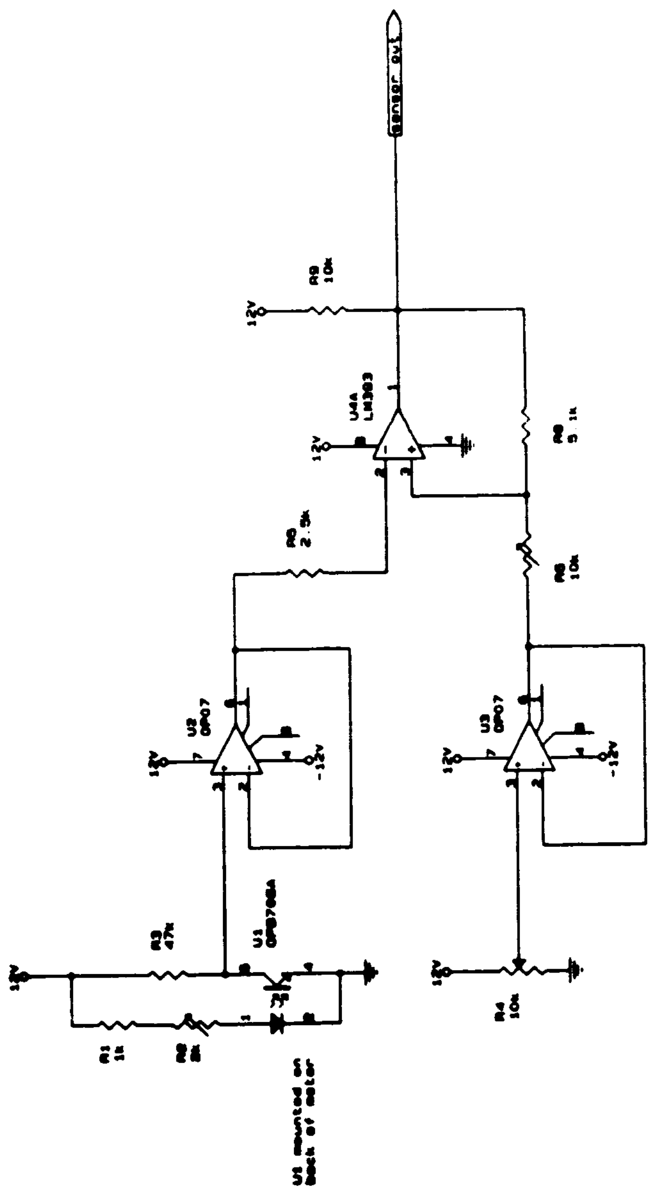

Figure E.3. Reflective sensor amplifier and comparator circuits.

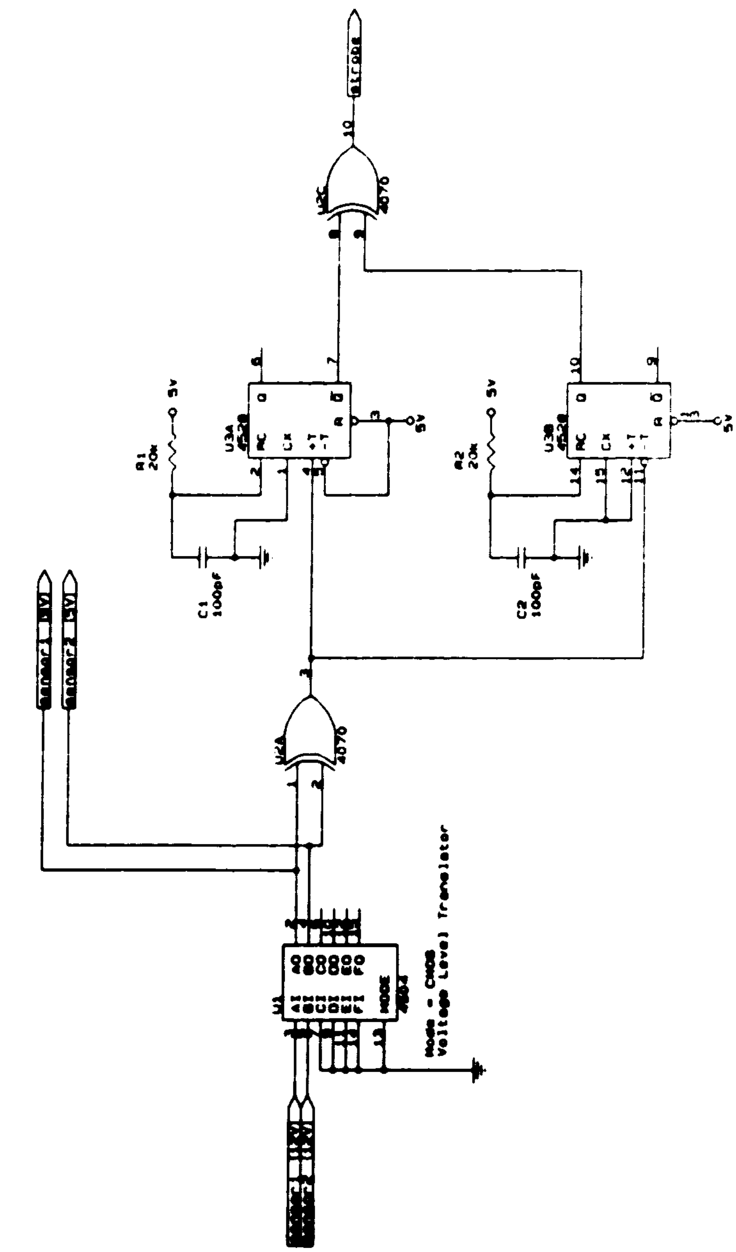

Figure E.4. Strobe generator for loading the sensor outputs into th: computer and causing an interrupt.

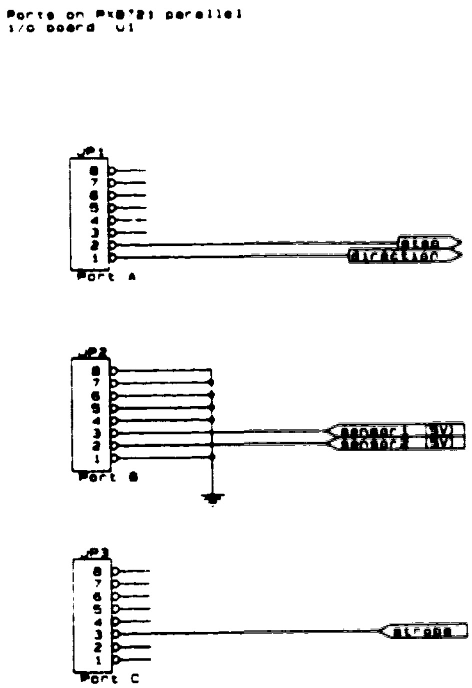

Figure E.5. Wiring diagram for connection of circuits to control computer.

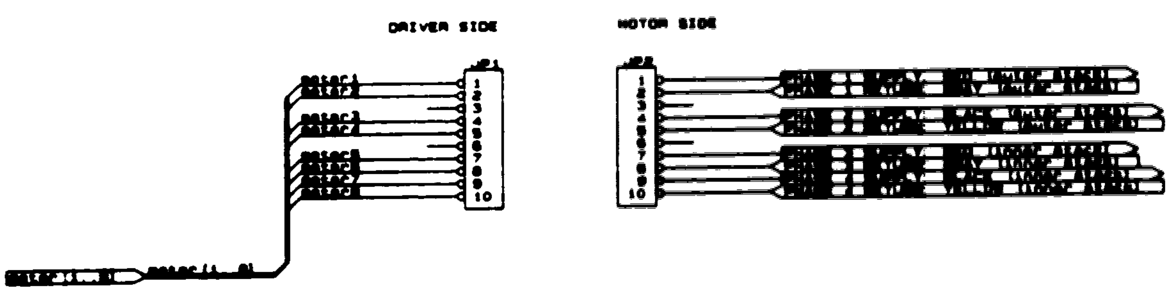

Figure E.6. Wiring diagram for connection of driver to AIRPAX 82701-P2 stepping motor.

Appendix F

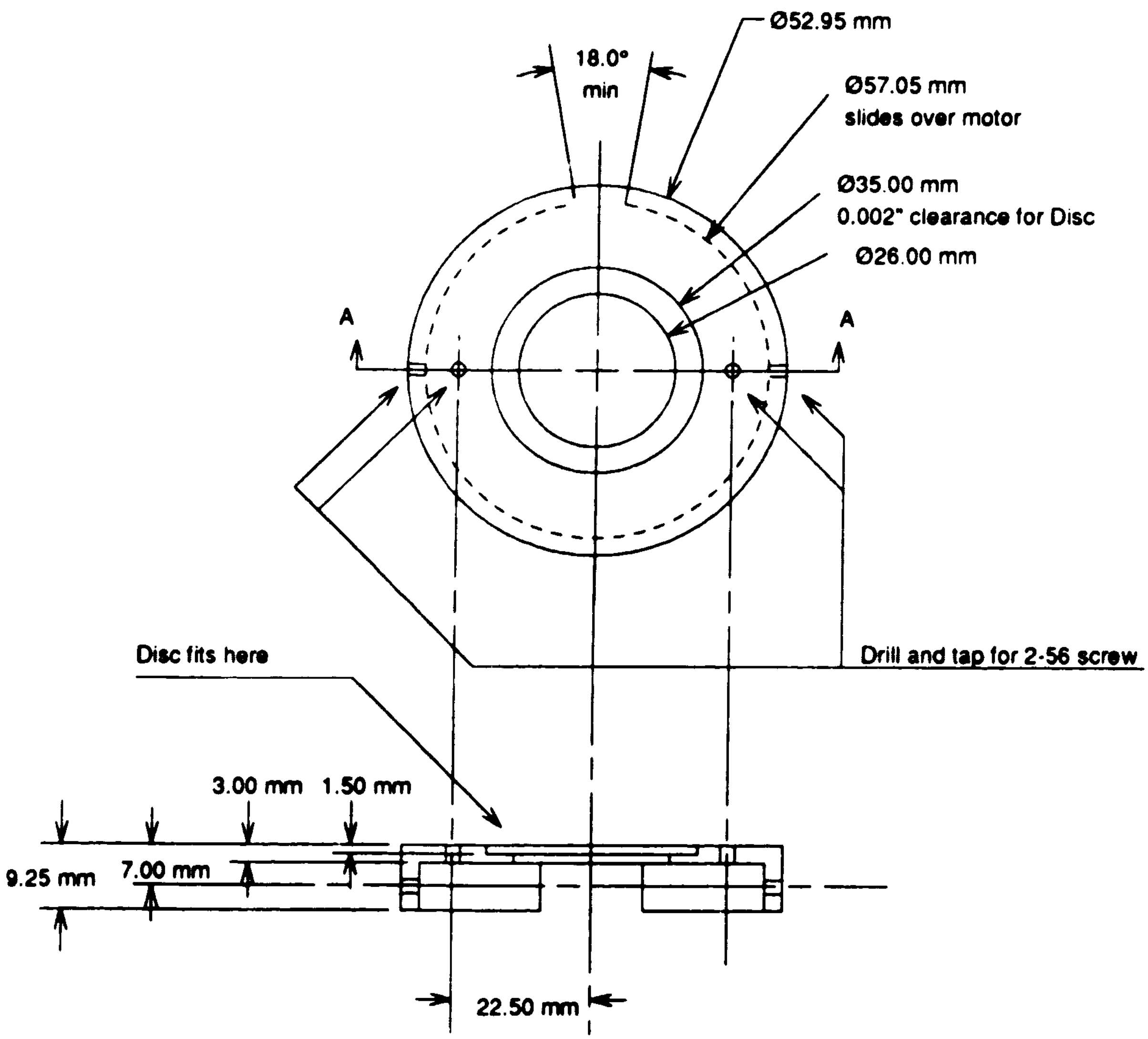

Figure F.1. This mount is fixed to the rear of the motor. The disc mounts into the recessed center portion.

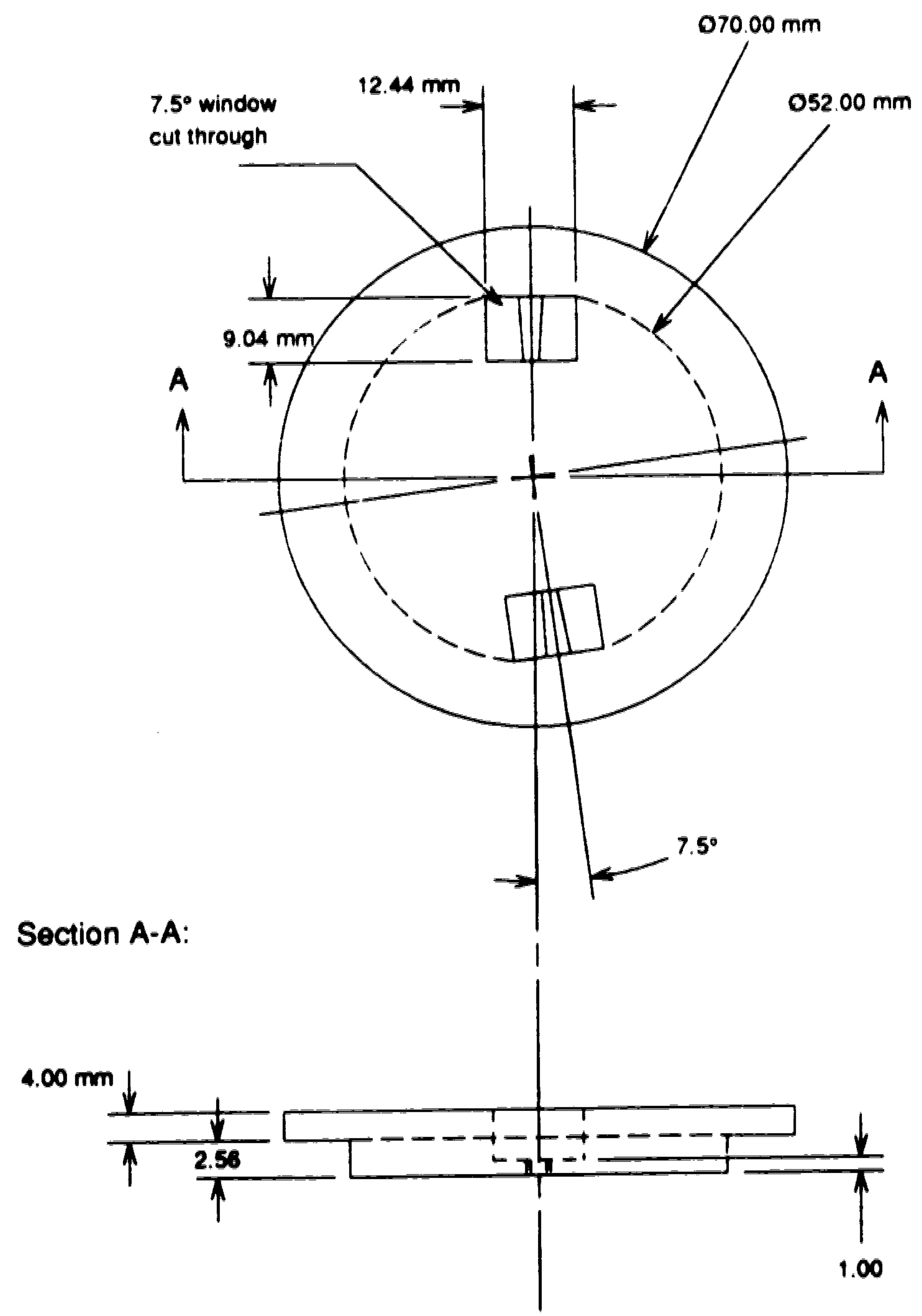

Figure F.2. Sensor disc. The sensors are inserted into the rectangular holes.

Appendix G

This appendix contains four program listings:

goi.c	main program
step.c	contains routines for stepping motor
tecmar.c	contains routines for using timer
int.asm	interrupt service routine

The programs are written for an MS-DOS compatible computer and assume that the computer is equipped with a Tecmar data acquisition and timer system and a PXB-721 digital i/o board.

goi.c

```
/* goi.c by C. Verburg on 27 March 1990
   modified April 3 1990
/* goi.c - program to run motor connected via L297/driver to PXB_297.

   goi: "go" with interrupts

   NOTE THAT THE STEP DETECTION IN THIS PROGRAM IS INTERRUPT
   DRIVEN. WHEN A STEP OCCURS, THE PARALLEL PORT ON THE PXB721
   IS PROGRAMMED TO INTERRUPT USING IRQ5. This is to ensure that
   no steps are missed during other operations.

   The motor is stepped after a step is detected via the parallel
   port inputs connected to the optical sensors (with signal
   conditioning). The main program knows that a step has occurred by
   continually inspecting a flag which is set in the interrupt service
   routine.

   The program attempts to position the rotor relative to its position
   when the program is started. The rotor is moved the number of steps
   defined in the #define: TOTAL_STEPS.

   If a step command is executed and no step occurs in the timeout
   given by TIMEOUT (in milliseconds), the flag 'stepDetected' is set
   in order to cause the step algorithm to try again.

   Assumed that the sequencer direction input connected to PXB721
   port 1A bit 0, the clock is connected to bit 1 and the optical
   sensor electronics are connected to port 1B bits 1 and 2. (1B
   bit 0 causes an error for some reason.)

   This program calls functions in the files tecmar.c and step.c.
*/

/************************************************************************
   to run, type:  goi<ret>
************************************************************************/

#include    "pxb721.h"
#include    <stdio.h>
#include    <process.h>
#include    <conio.h>
#include    <fcntl.h>
#include    <io.h>
#include    <stdlib.h>
#include    <dos.h>

/*_______________________________________________________________________*/
/* Function prototype declarations.*/
void main(void);
void irq5_c(void);        /* irq5_c is the Int. Serv. Routine called
                             by int.asm */
void initInterrupts(void); /* initializes interrupt vector and saves
                             old context */
```

gol.c

```
void restore(void);           /* restores original IRQ5 vector and
                                 interrupt mask */
void delay(unsigned);         /* wastes time, argument in ms */
unsigned int cs(void);        /* returns contents of Code Segment
                                 register */
unsigned int ds(void);        /* returns contents of Data Segment
                                 register */

/* in int.asm */
extern void far SERV5(void);  /* irq5_asm is the Int. Serv. Routine in
                                 int.asm */

/* in step.c */
extern void stepCW(void);
extern void stepCCW(void);
extern void initPXB(void);              /* init for simple i/o */
extern void initPXBInterrupts(void);    /* init for interrupt driven i/o
                                           */
extern void disablePXB721(void);        /* disable interrupts */

/* in tecmar.c */
extern void timerGo(unsigned);          /* start timer in ms */
extern unsigned timerRead(void);

/*______________________________________________________________________*/
#define SETPOINT  480       /* default number of steps. */
#define TIME_OUT  100       /* timeout */
#define CW        0         /* offsets for state lookup table */
#define CCW       7         /* 4 if using bits 0 and 1 */

#define TRUE      1
#define FALSE     0
#define VALID     1
#define INVALID   0
#define HIGH      1
#define LOW       0
#define MOTORIN   PXB1B     /* Input port used to read motor status */
#define MOTOROUT  PXB1A     /* Output port used to output to motor */

#define IRQ5   0x0D*4   /* location of irq5 vector in memory */

/*______________________________________________________________________*/
/* Macro to build a longword in the form: (segment:offset) out of two
   integers. */
#define FP_INT_ADDR(seg,off)(unsigned int far *)((((unsigned
   long)seg)<<16)|(unsigned long)off)

/*______________________________________________________________________*/
/* Globals for use with interrupts. */

struct int_vector
   {
   unsigned int off;    /* Offset of interrupt vector   */
   unsigned int seg;    /* Segment of vector            */
```

goi.c

```
   };

struct int_vector old_vec={0};    /* for saving previous Interrupt
                                     Vector */

union REGS inregs,outregs;     /* for initializing the interrupt
                                  service */
struct SREGS segregs;

unsigned int far *vector;      /* interrupt vector address */
unsigned int far *data_seg;    /* pointer to where the DS
                               register is saved */
int int_mask;      /* for saving original content of IMR of 8259 */

/*______________________________________________________________________*/
/* Globals for keeping track of position. */
int running=FALSE,direction,desiredDirection;
long int actualSteps=0,desiredSteps=SETPOINT;
unsigned newPosition,oldPosition,stepDetected=FALSE,sensorError=0;

/* for determining direction. Sensors connected to bits 1 and 2
   (the -1's indicate unused entries) */
int stateTable[14]={4,-1,0,-1,6,-1,2,2,-1,6,-1,0,-1,4};

/*______________________________________________________________________*/
void main()
{

   initPXB();
   oldPosition=inp(MOTORIN);

   if(desiredSteps!=0)
      stepDetected=TRUE;

   initInterrupts();
   initPXBInterrupts();

   while(!kbhit())
   {
      /* if motor is in position, continually check to see if it
            is moved. If moved, change status to running. */
      if(!running)
      {
         if(actualSteps!=desiredSteps)
            running=TRUE;
      }
      if(running)
      {
         /* check to see if timeout has occurred */
            /* if so, insert phantom step */
         if(timerRead()<5)
         {
            /* set flag to force another attempt */
            stepDetected=TRUE;
```

goi.c

```
            /* make phantom step in correct direction */
            direction=desiredDirection;
         }
         if(stepDetected)
         {
            /* if close to setpoint slow down */
            if(labs(actualSteps-desiredSteps)<2)
               delay(20);
            /* if step detected in wrong direction, */
            if(direction!=desiredDirection)
               delay(100); /*  let rotor stop before continuing */

            timerGo(TIME_OUT);

            /* if below setpoint, step CW */
            if(actualSteps<desiredSteps)
            {
               desiredDirection=CW;
               timerGo(TIME_OUT);
               stepCW();
            }
            /* if below setpoint, step CCW */
            if(actualSteps>desiredSteps)
            {
               desiredDirection=CCW;
               timerGo(TIME_OUT);
               stepCCW();
            }

            stepDetected=FALSE;      /* clear the flag */
         }
      }
   }

   /* print summary of position and the number of sensor transitions
      detected which did not indicate either CW or CCW */
   printf("\nActual number of steps: %li\tErroneous readings: %u\n",
      actualSteps,sensorError);

      disablePXB721();
      restore();
      exit(0);
}

/*____________________________________________________________________*/
/* cs(): return current Code Segment register */
unsigned int cs()
{
   struct SREGS segregs;
   segread(&segregs);
   return(segregs.cs);
}

/*____________________________________________________________________*/
```

goi.c

```
/* ds(): return current Data Segment register */
unsigned int ds()
{
   struct SREGS segregs;
   segread(&segregs);
   return(segregs.ds);
}

/*__________________________________________________________________*/
/* irq5_c(): functional part of interrupt service */
void irq5_c()
{
   newPosition=inp(MOTORIN);

   if(newPosition==stateTable[oldPosition+CW])
   {
      actualSteps+=1;
      direction=CW;
   }
   else if(newPosition==stateTable[oldPosition+CCW])
   {
      actualSteps-=1;
      direction=CCW;
   }
   else
      sensorError+=1;

   oldPosition=newPosition;
   stepDetected=TRUE;
}

/*__________________________________________________________________*/
/* initInterrupts(): initialize computer for interrupts */
void initInterrupts()
{
   data_seg=FP_INT_ADDR(cs(),SERV5-2);
      /* initialize pointer to address where we're saving DS register.
         This space is reserved in the interrupt service routine just
         before the _SERV5 label. */


   /* save the contents of the data segment register */
   *data_seg=ds();/* save the contents of the data segment register */

   /* build pointer to the irq5 vector in low memory */
   vector=FP_INT_ADDR(0,IRQ5);

   /* store the old vector (for later restoration) */
   old_vec.seg=*(vector+1);
   old_vec.off=*vector;

   /* load new vector for the interrupt service routine in int.asm
      The start of the code is labeled _SERV5 */
   /* interrupt code loaded into same segment as this code */
```

goi.c

```
   *(vector+1)=cs();
   *vector=(unsigned int) SERV5;     /* offset of interrupt service */

   int_mask=inp(0x21);               /* save original 8259 IMR */
   outp(0x21,(int_mask&0xdf));       /* enable IRQ5 */
   outp(0x20,0x20);                  /* tell 8259 to proceed */
}

/*______________________________________________________________________*/
/* restore(): return computer to original state */
void restore()
{
   disablePXB721();

   /* restore the original interrupt vector */
   *(vector+1)=old_vec.seg;
   *vector=old_vec.off;

   outp(0x20, 0x20);     /* clear all pending interrupts */

   /* restore normal priority setting defined in 8259 documentation */
   outp(0x20,0xe7);
   outp(0x21,int_mask);     /* restore original IMR to 8259 */
}

/*______________________________________________________________________*/
void delay(delayTime)
unsigned delayTime;
{
   timerGo(delayTime+5);
   while(timerRead()>5);
}
```

step.c

```
/*  step.c by C. Verburg 1989.
   modified Feb 1990.
   modified 29 Mar 1990.
*/
/* step.c - routines to step a motor whose driver is connected to the
   PXB-721 parallel i/o board. */


#include    "pxb721.h"
#include    <conio.h>

#define MOTORIN    PXB1B     /* Input port used to read motor status */
#define MOTOROUT   PXB1A     /* Output port used to output to motor */
#define CONT_REG   PXB1CR    /* Control register for I/C ports */
#define CW         1      /* used in specification of mc .or direction */
#define CCW        0      /* " */
#define HOME_BIT   0x02   /* home pin on L297 connected to port B */

extern void timerGo(unsigned);
extern unsigned timerRead(void);

void initPXB(void)
{
   outp(CONT_REG,0x8b); /* INITIALIZE A as output B,C as input */
}

void initPXBInterrupts(void)
{
   /* see notes 29 Mar 1990 for bit selection */
   /* Initialize port A=mode 0 output, port C upper=input port B=mode 1
      input */
   outp(CONT_REG,0x8f);
   /* Enable INTR_B using bit set/reset function. */
   outp(CONT_REG,0x05);
}

void disablePXB721(void)
{
   /* disable interrupts and make port A=mode 0 output make port
      B,C=mode 0 input */
   outp(CONT_REG,0x04); /* disable INTR_B */
   outp(CONT_REG,0x8b); /* INITIALIZE A as output B,C as input */
   inp(MOTORIN);        /* dummy read to clear any pending interrupts */
}

void stepCW(void)
{
   outp(MOTOROUT,0x00);
   outp(MOTOROUT,0x02);
}

void stepCCW(void)
{
   outp(MOTOROUT,0x01);
```

step.c

```
   outp(MOTOROUT,0x03);
}

int goHome(void)
{
   int i=0;
   while(!(inp(MOTORIN)&HOME_BIT))
   {
      stepCW();
      timerGo(10);        /* Give the motor time to respond */
      if(i++==10)         /* If haven't found home by now, exit */
         return(0);
      while(timerRead()!=1);
   }
   timerGo(300);          /* Let the rotor settle for a while */
   while(timerRead()!=1);
   return(1);             /* Found home successfully */
}
```

tecmar.c

```
/* tecmar.c, routines for using the tecmar data acquisition hardware */
/* Basic functions (except for minor modifications) written by Frank
      Niscak */
/* 1989ff modifications made by C. Verburg */
#include    <conio.h>
#include    <math.h>

#define     TECMAR     0x710

/* Function prototype definitions */
void timerGo(unsigned);
void timerGoMicro(unsigned);
unsigned timerEnd(void);
unsigned timerRead(void);
double convert(int);

/***********************************************************************
 * Procedure:          timerGo
 * Function:           Initializes and starts Tecmar counter #5
                          (counts down from time)
 * Input Frequency:    1 kHz (counts milliseconds)
 * Maximum count:      16 bit unsigned word
 * To stop:            call timerEnd
 * To read time:       call timerRead
***********************************************************************/
void timerGo(time)
   unsigned time;
{
   unsigned timeh;

   timeh=time;
   time=time & 0x00ff;
   timeh=(timeh>>8) & 0x00ff;

   outp(TECMAR+9, 0xff);    /* Master Reset of 9513 */
   outp(TECMAR+9, 0xe8);    /* Disable Data Pointer Sequencing */
   outp(TECMAR+9, 0x17);    /* Data Pointer to Master Mode Reg. */
   outp(TECMAR+8, 0x10);    /* Low byte, Master Mode Register */
   outp(TECMAR+8, 0xd1);    /* High byte, Master Mode Register */
   outp(TECMAR+9, 0x05);    /* Data Pointer to Counter Mode Reg. */
   outp(TECMAR+8, 0x05);    /* Low byte to Counter Mode Register */
   outp(TECMAR+8, 0x0e);    /* High byte, Counter Mode Register */
   outp(TECMAR+9, 0x0d);    /* Data Pointer to Load Register */
   outp(TECMAR+8, time);    /* Low byte to Load Register */
   outp(TECMAR+8, timeh);   /* High byte to Load Register */
   outp(TECMAR+9, 0x70);    /* Load and Arm Counter #5 */
}

/***********************************************************************
 * Procedure:          timerGoMicro
 * Function:           Initializes and starts Tecmar counter #5.(counts
                          down from time)
 * Input Frequency:    1 MHz (counts microseconds)
 * Maximum count:      16 bit unsigned word
```

tecmar.c

```
 * To stop:          calltimerEnd
 * To read time:     call timerRead
***************************************************************************/
void timerGoMicro(time)
   unsigned time;
{
   unsigned timeh;

   timeh=time;
   time=time & 0x00ff;
   timeh=(timeh>>8) & 0x00ff;

   outp(TECMAR+9, 0xff);    /* Master Reset of 9513 */
   outp(TECMAR+9, 0xe8);    /* Disable Data Pointer Sequencing */
   outp(TECMAR+9, 0x17);    /* Data Pointer to Master Mode Reg. */
   outp(TECMAR+8, 0x10);    /* Low byte into Master Mode Register */
   outp(TECMAR+8, 0xd1);    /* High byte into Master Mode Register */
   outp(TECMAR+9, 0x05);    /* Data Pointer to Counter Mode Reg. */
   outp(TECMAR+8, 0x05);    /* Low byte to Counter Mode Register */
   outp(TECMAR+8, 0x0b);    /* High byte to Counter Mode Register */
   outp(TECMAR+9, 0x0d);    /* Data Pointer to Load Register */
   outp(TECMAR+8, time);    /* Low byte to Load Register */
   outp(TECMAR+8, timeh);   /* High byte to Load Register */
   outp(TECMAR+9, 0x70);    /* Load and Arm Counter #5 */
}

/**************************************************************************
 *Procedure:   timerEnd
 *Function :   Stops counter #5 and returns its content.
***************************************************************************/
unsigned timerEnd()
{
   outp(TECMAR+9, 0x90);    /* Disarm and Save Counter #5 to Hold */
   outp(TECMAR+9, 0x15);    /* Data Pointer to Hold Register #5 */
   return((inp(TECMAR+8)<<8) | inp(TECMAR+8));
}

/**************************************************************************
 *Procedure:   timerRead
 *Function :   Reads counter #5 and returns its content.
***************************************************************************/
unsigned timerRead()
{
   outp(TECMAR+9, 0xb0);    /* Disarm and Save Counter #5 to Hold */
   outp(TECMAR+9, 0x15);    /* Data Pointer to Hold Register #5 */
   return((inp(TECMAR+8)<<8) | inp(TECMAR+8));
}

/**************************************************************************
 *Procedure:   convert
 *Function :   Starts and waits for A/D conversion.
 *Parameters:  channel number (int). Returns double in volts.
***************************************************************************/
double convert(channelNumber)
```

tecmar.c

```
int channelNumber;   /* number of AD channel to convert */
{
/* conversion constant for A/D conversion = 10.0 / 2048.0  */
#define CONVERSION_CONSTANT 0.0048828125

   unsigned twobyte;

   outp(TECMAR + 4, 0x80);              /* Initalize for nonindexing */
   outp(TECMAR + 5, channelNumber);     /* Channel Max_number */
   outp(TECMAR + 6, 0x0);               /* Start the AD conversion */

   /* Waste some time: (only for 25MHz 386 machine using DTK
         motherboard)
         1. to compensate for timing problem with timerRead
         2. for some reason, this time wastage (before waiting for
            conversion to finish) speeds the execution of this routine.
   */
   twobyte++;
   twobyte++;
   twobyte++;
   twobyte++;
   twobyte++;
   twobyte--;
   twobyte--;
   twobyte--;
   twobyte--;
   twobyte--;

   while (inp(TECMAR + 4) < 0x80);  /* Wait for end of AD conversion */
   twobyte = inpw(TECMAR + 5);
   return(((double)((int)twobyte)) * CONVERSION_CONSTANT);
}
```

Int.asm

```
; Interrupt handler. Saves context and calls functional part (usually
;  written in C) called irq5_c.
        PAGE 58,132
        TITLE    irq5_asm.asm - irq5 interrupt service routine
;
_TEXT   SEGMENT BYTE PUBLIC 'CODE'
_TEXT   ENDS
_DATA   SEGMENT WORD PUBLIC 'DATA'
_DATA   ENDS
CONST   SEGMENT WORD PUBLIC 'CONST'
CONST   ENDS
_BSS    SEGMENT WORD PUBLIC 'BSS'
_BSS    ENDS
DGROUP  GROUP CONST,_BSS,_DATA
        ASSUME CS:_TEXT,DS:DGROUP,SS:DGROUP,ES:DGROUP
;-----------------------------------------------------------------
;
;
;
;-----------------------------------------------------------------
;
;
_TEXT   SEGMENT
; reserve space for main program to store DS register (at SERV5-2)
_DATA_SEG   DW      (0)
_SERV5      PROC    FAR
        PUBLIC      _SERV5
        PUBLIC      _DATA_SEG
        EXTRN   _irq5_c:NEAR        ; define functional part of handler
;
        PUSHF               ; Save the flags
        PUSH    AX          ; Save all registers
        PUSH    BX          ;
        PUSH    CX          ;
        PUSH    DX          ;
        PUSH    SI          ;
        PUSH    DI          ;
        PUSH    BP          ;
        PUSH    DS          ;
        PUSH    ES          ;
;
        MOV     AX,CS:_DATA_SEG     ; Make sure DS register contain
        MOV     DS,AX               ;   pointer to main program data
        CLC                         ; Clear the carry flag
        CLD                         ; Set direction flag forward
        CALL    _irq5_c             ; Call C functional part
;
; Standard interrupt return procedure:
        MOV     AL,20H
        OUT     20H,AL              ; address the 8259
;; Restore all registers
        POP     ES
        POP     DS
        POP     BP
```

```
            POP     DI
            POP     SI
            POP     DX
            POP     CX
            POP     BX
            POP     AX
            POPF
;
            IRET                    ; Interrupts on with return
_SERV5      ENDP
_TEXT       ENDS
        END
```