3/?

MICROPROCESSOR BASED STEP MOTOR CONTROLLER

by

NEERAJ MAGOTRA

B.Tech. Indian Institute of Technology, Bombay

India 1980

A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

College of Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1982

Approved by :

W. W. Koepsel

W.W.Koepsel

Major Professor

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

STEP MOTORS

1.0 INTRODUCTION

With the advent of digital electronics and associated devices like microprocessors and digital controllers, it has become increasingly important to be able to convert signals from the analog form to the digital form and vice-versa. This would be necessary to allow the interfacing of digital devices with the real world environment, which is an analog one.

The step motor is one of many such devices which provide an interface between the digital and analog worlds. It is essentially an electromagnetic device which converts input pulses (digital signal) to angular motion (analog shaft rotation). One of the earliest forms of a step motor was used by the British Royal Navy in the early 1930's to transmit shaft rotations(1)

At the present time,step motors are used in a wide variety of computer peripheral equipment (such as printers and disk drives) as well as in industrial control systems like machine tool systems, process control systems and robotics.

There are many different types of step motors, such as the solenoid and ratchet motor,the harmonic drive motor and the phase pulsed motor to name a few. However the most commonly used types are the VARIABLE RELUCTANCE motor and the PERMANENT MAGNET motor. The variable reluctance motor has a wound stator and an unexcited rotor, while the permanent magnet motor has a wound stator and a permanent magnet rotor. Another type of step motor developed recently in 1969, is the linear step-motor (1) . This motor, as the name implies, has a linear motion output and eliminates the need for rotary to linear motion conversion, as is the case with rotary step motors. The rotor of a step motor moves through a predetermined angle (a design constant of the particular motor) in response to an input pulse. In the case of a linear step motor, the response consists of the armature moving linearly along the stator bed . When properly controlled, the discrete steps (linear or angular) of the armature maintain synchronism with the input pulses.

Besides electrical step motors, industrial needs also led to the development of the electro-hydraulic step motor. These motors are capable of providing high power output more efficiently than electrical motors, which provide high power output at the expense of high heat losses. The electro-hydraulic step motor consists of an electrical step motor servo-valve and hydraulic motor. By virtue of their construction the electro-hydraulic motors are slower in responding to control signals than their electrical counterparts.

1.1 MOTOR OPERATING PRINCIPLES

The step motor is a means of converting electrical energy to mechanical energy, just like the more common direct current and alternating current motors. However, it differs greatly from both these types of motors as regards its principle of operation. The stator phase windings, in the step motor, are driven by square waves (digital signal) of the same frequency but shifted in phase. This applies to both, the variable reluctance and permanent magnet step motors, the major difference between the two is the rotor which is a magnet in the case of the permanent magnet motor and made of magnetic material in the variable reluctance motor. The motor used in this study was a permanent magnet type. A summary of the characteristics of the permanent magnet step motor and the variable reluctance step motor is included in the appendix on step motors.

In the permanent magnet motor rotary motion is obtained by energizing the stator phases in a particular sequence, causing the permanent-magnet rotor to follow the excitation sequence. The following example illustrates the principle of operation of a permanent magnet step motor. Fig.1.1 shows a motor with four stator windings and a two pole permanent-magnet rotor with a step size of ninety degrees. The step size represents the angle through which the rotor moves for a sequential change in the phase excitation pattern.

The arrow associated with each phase winding indicates the direction of the magnetic flux produced inside the coil when it is energised. shows the single phase excitation sequence: A_2, B_2, A_1, B_1. In this mode of operation only one phase of the motor is energized at a time. As the excitation pattern changes (Fig.1.2a----- d) , the rotor follows the changing pattern and thus rotates in the clockwise direction in steps of ninety degrees. To reverse the direction of rotation the sequence must be inverted to read: B_1, A_1, B_2, A_2. Next, one can consider the double phase excitation sequence: A_2B_2, A_1B_2, A_1B_1, A_2B_1, in this mode of operation two phases of the motor are excited simultaneously (Fig.1.3a----- d). Once again the rotor moves in steps of ninety degrees as it follows the changing excitation sequence, but the distinct equilibrium positions are forty-five degrees displaced from those of the single phase excitation sequence. Again, to reverse the direction of rotation the sequence must simply be inverted.

The fact that the rotor-equilibrium positions are forty-five degrees out of phase in the two modes of operation mentioned above gives rise to the concept of the mixed-mode excitation sequence: A_2, A_2B_2, B_2, A_1B_2, A_1, A_1B_1, B_1, A_2B_1 . The rotor again moves in the clockwise direction but with a step size of forty-five degrees.

The motor used in this study had a step size of 1.8 degrees for the single phase excitation mode of operation and double phase excitation mode of operation, respectively and a step size of 0.9 degrees for the mixed-mode of operation. Figures 1.4, 1.5, 1.6 show the timing diagrams as well as excitation tables for the three modes of operation.

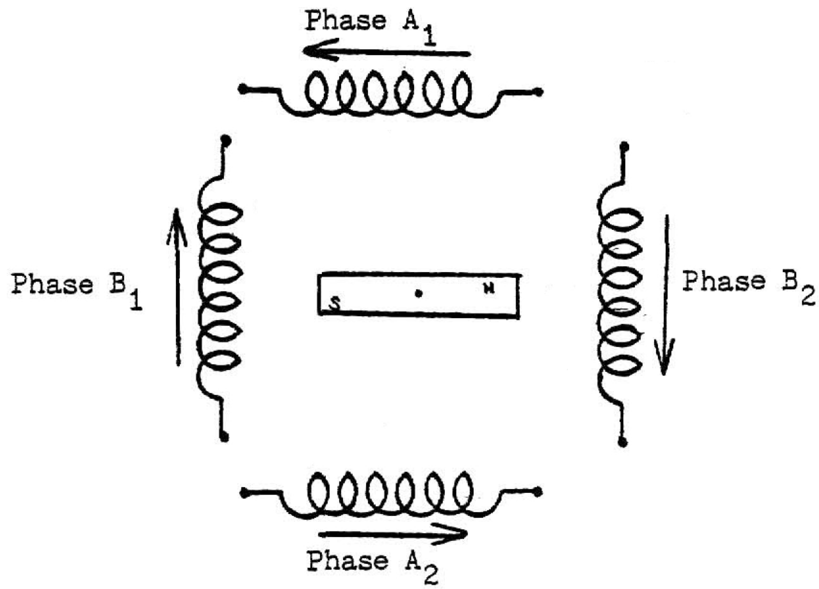


Fig.1.1 Motor Representation

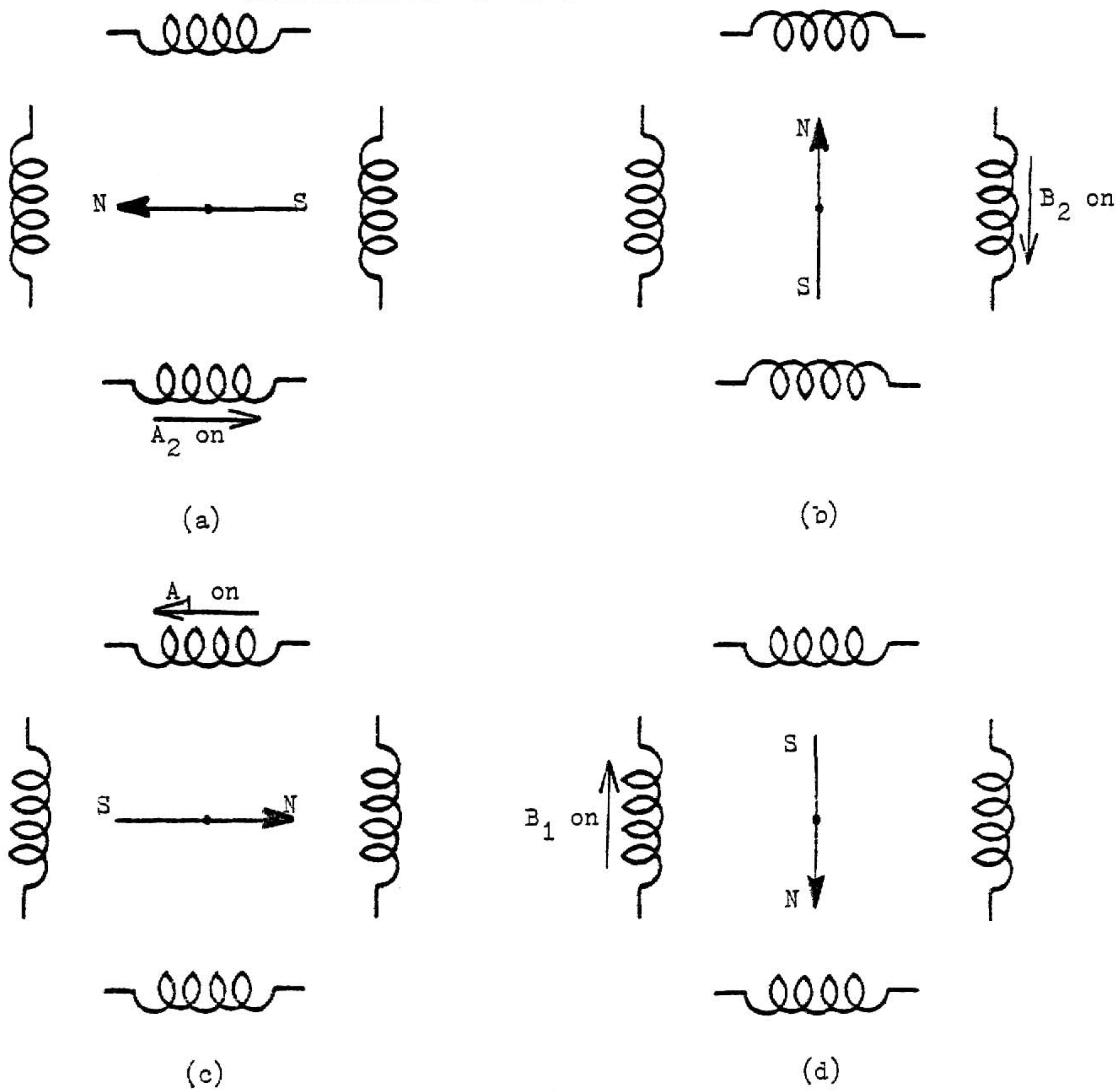


Fig.1.2 One phase on mode of operation

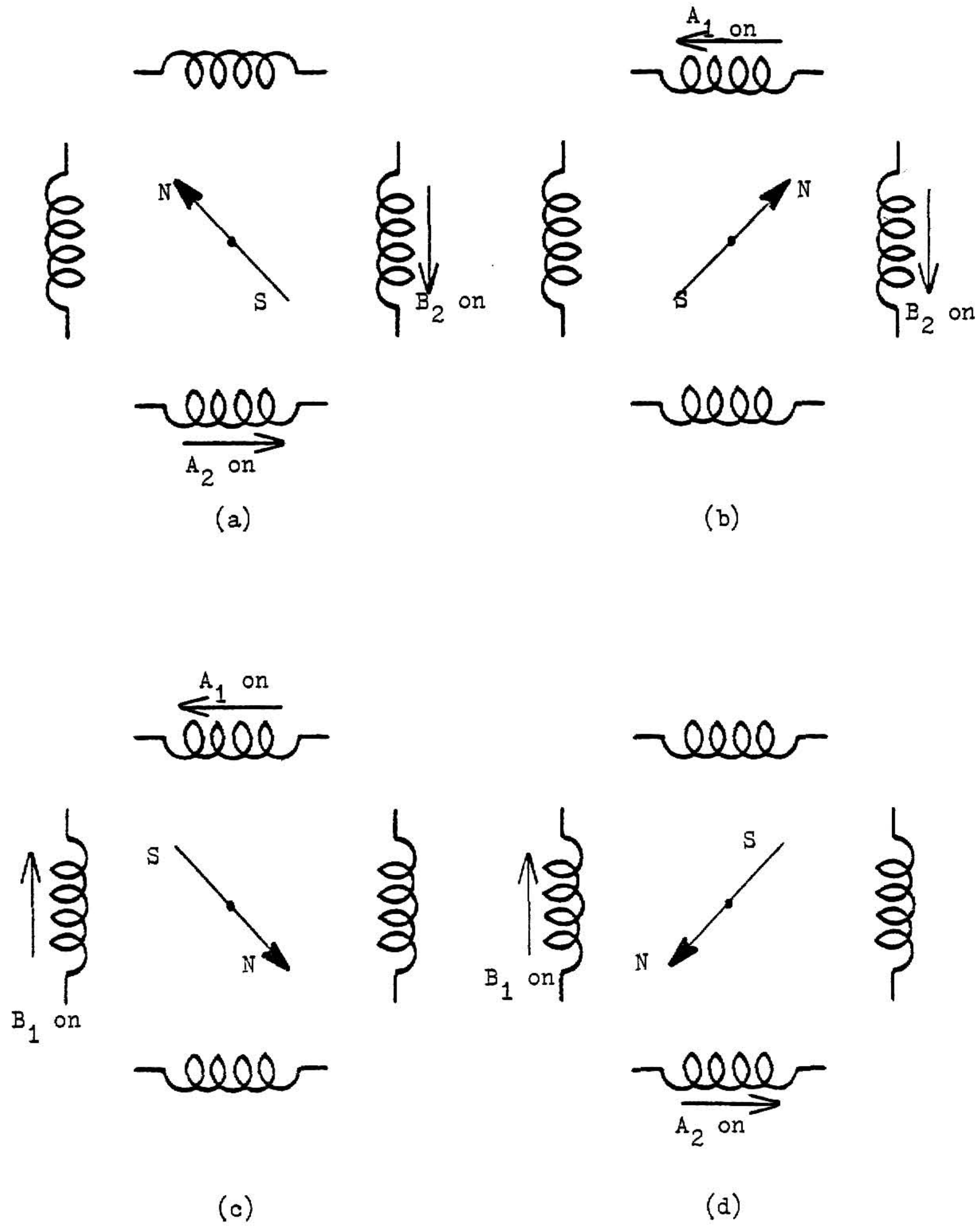
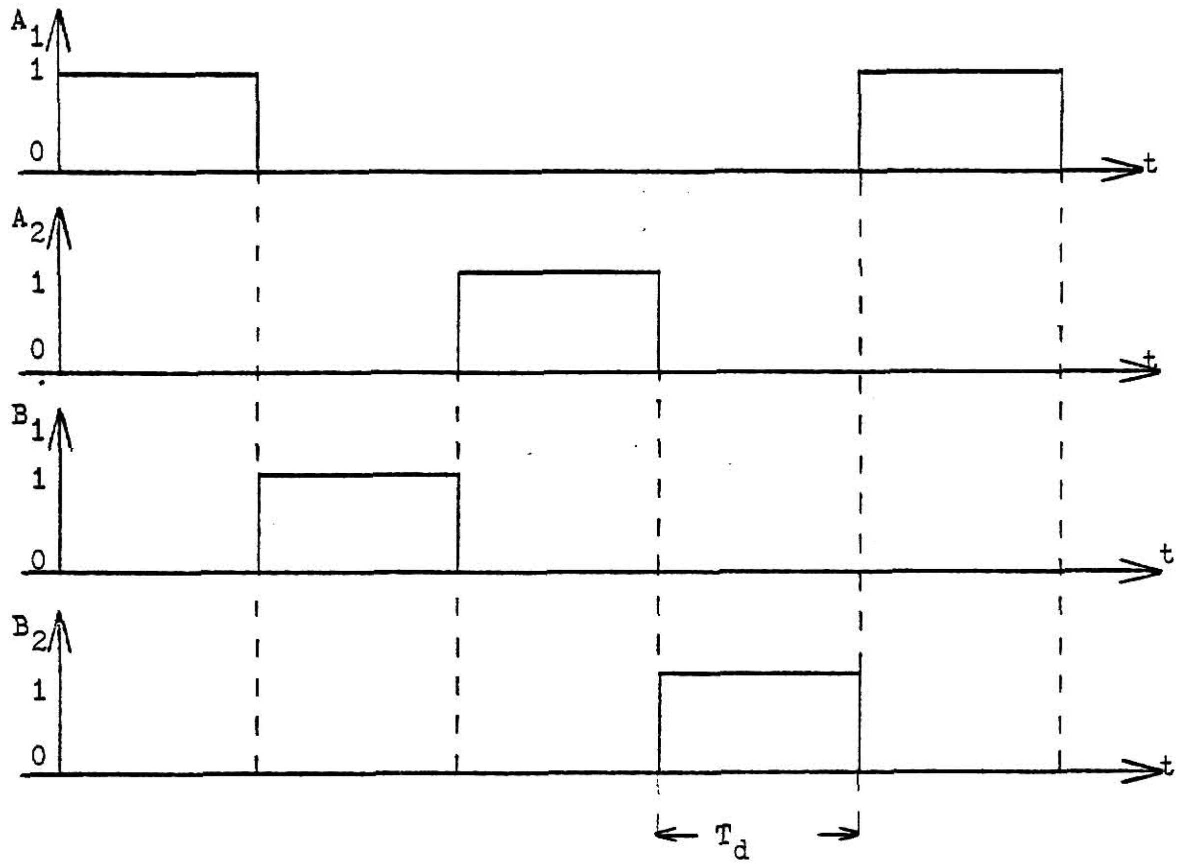


Fig.1.3 'Two phase on' mode of operation



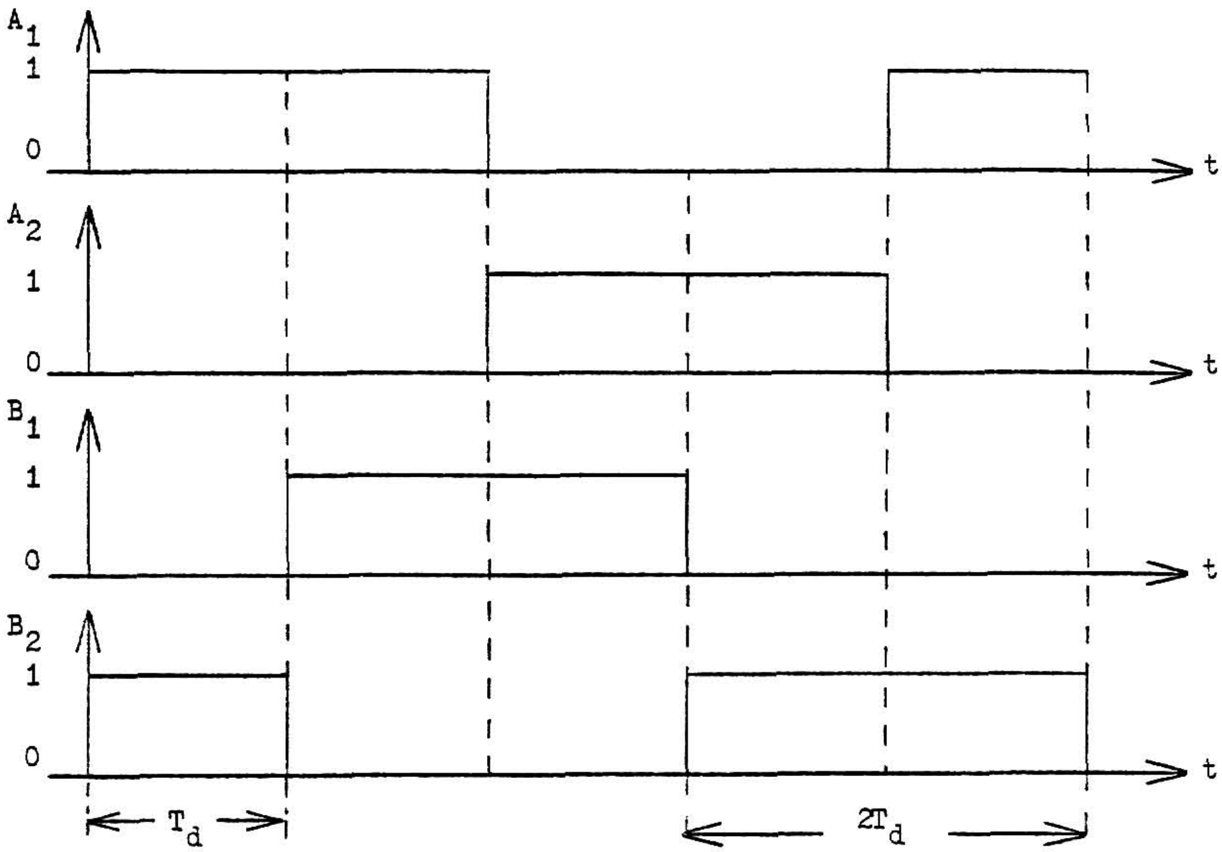
A_1	A_2	B_1	B_2
1	0	0	0
0	0	1	0
0	1	0	0
0	0	0	1

'0' = 'OFF'

'1' = 'ON'

T_d = Time for which a particular excitation pattern remains constant.

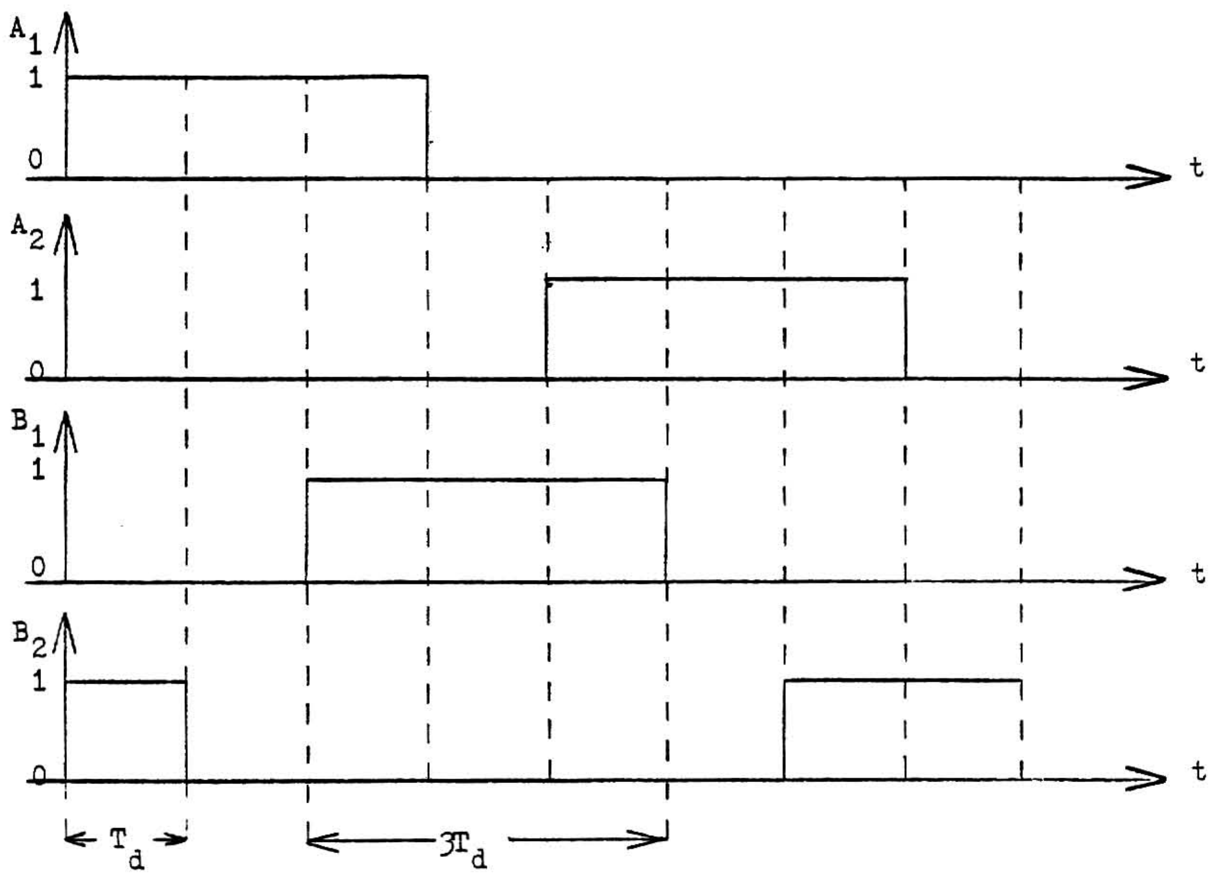
Fig1.4 Timing diagram and Excitation table for 'One phase on' mode



A_1	A_2	B_1	B_2
1	0	0	1
1	0	1	0
0	1	1	0
0	1	0	1

'0' = 'OFF'
'1' = 'ON'

Fig.1.5 Timing diagram and Excitation table for 'Two phase on' mode



A_1	A_2	B_1	B_2
1	0	0	1
1	0	0	0
1	0	1	0
0	0	1	0
0	1	1	0
0	1	0	0
0	1	0	1
0	0	0	1

'0' = 'OFF'
'1' = 'ON'

Fig.1.6 Timing diagram and Excitation table for Mixed mode

1.2 POWER DRIVERS

For a permanent magnet motor to step effectively (deliver rated torque) at high stepping rates, it is essential to ensure that, whenever a particular phase is energized and subsequently de-energized, the phase current swings between zero and its rated maximum . This becomes critical especially when the reciprocal of the stepping rate approaches the time constant of the motor phase winding.

The phase current waveform shown in Fig.1.7c results when the stepping rate reciprocal becomes comparable to the phase time constant (L /R). This is not desirable, since in the time interval (t_2-t_1), some other phase of the motor would be on and the residual current in this phase, which is supposedly 'off', would weaken the net magnetomotive-force acting on the rotor and thereby reduce the resulting torque.

Some of the schemes adopted to overcome the above problem, that is to permit effective motor operation at high stepping rates, are as follows:

(A)RESISTANCE DRIVE

(B)CHOPPER DRIVE

(C)BILEVEL DRIVE

(D)CONSTANT CURRENT SOURCE DRIVE

Of the different schemes listed above the first two were implemented in the course of this study. Implementation details for these are included in the chapter on hardware aspects. Each of the above schemes is outlined in some detail in the remainder of this section on power drivers. Theoretical equations have been developed for the first two schemes and computer simulation results of some of these equations have been included as an appendix.

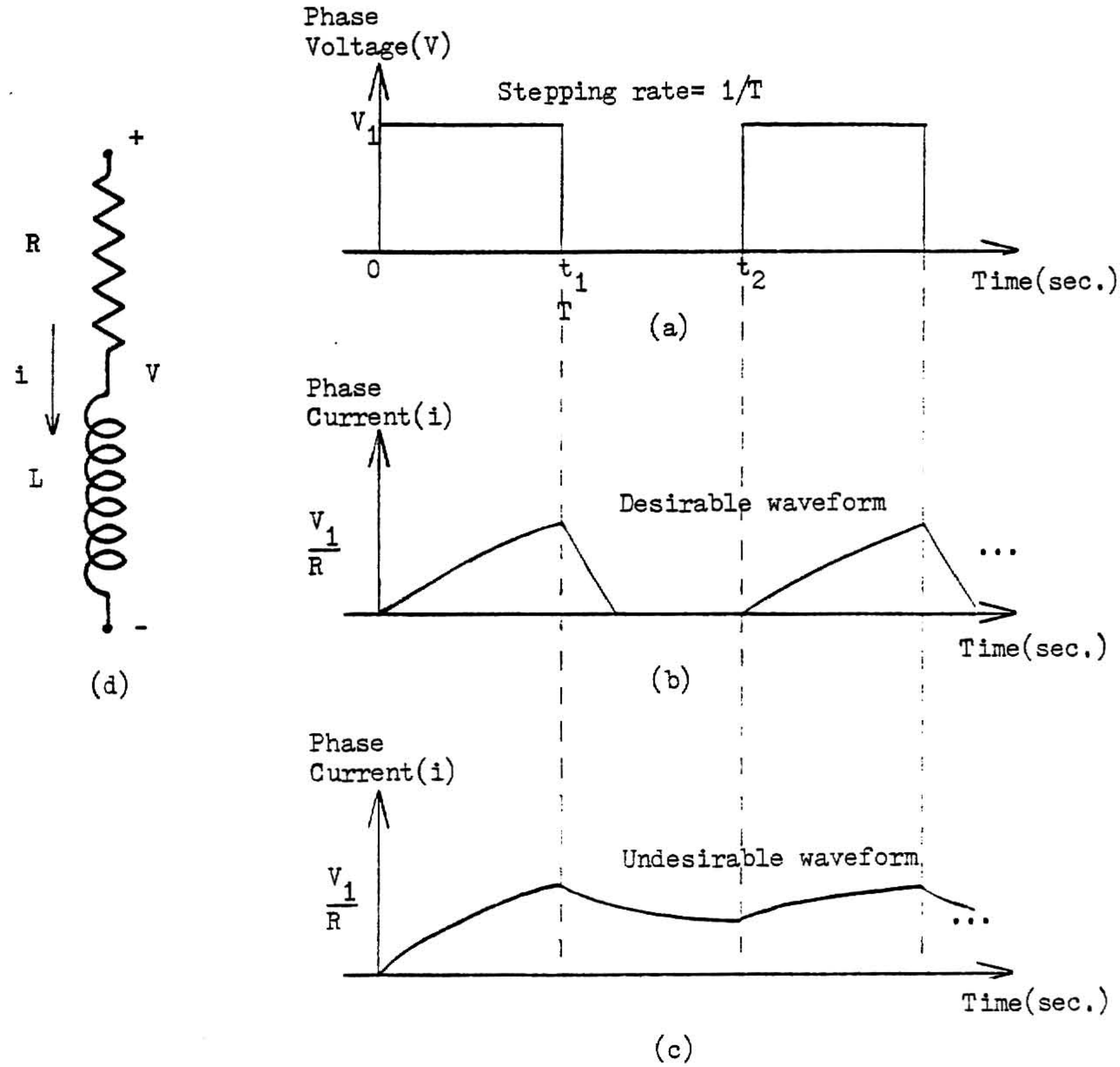


Fig.1.7 Phase voltage and current waveforms

A . Resistive drive

The resistive drive is the simplest of all the step motor drives. It is based on the fact that the time constant of a L-R circuit is inversely proportional to the resistor in series with the inductor. As shown in Fig.1.8a, a series resistance is added to each motor phase winding to reduce its time constant. The following are assumed:

a) Transistor Q_1 'on' state resistance = R_q

b) Diode D_1 'on' state resistance = R_d

c) The motor winding inductance/resistance = L_m/R_m

The charging τ_c and discharging τ_d time constants can be expressed as

$$\tau_c = \frac{L_m}{R_m + R_s + R_q} = L_m/R_c \qquad (1.1a)$$

$$\tau_d = \frac{L_m}{R_m + R_s + R_d} = L_m/R_c \quad . \qquad (1.1b)$$

The corresponding charging and discharging equations are as follows

Charging (Phase on) :

The charging current i_c is expressed as

$$i_c = (V_s \,/\, R_c)(1- e^{-t/\tau_c}) \qquad (1.2a)$$

with the initial condition

$$i_c(0_+) = 0 \quad .$$

The discharging current i_d is given by the equation

$$i_d = (V_s \,/\, R_d)e^{-t/\tau_d} \qquad (1.2b)$$

with the initial condition

$$i_d(0_+) = (V_s \,/\, R_c)$$

this initial condition is true assuming that the phase current reaches its maximum value while the phase is switched on.

This scheme is fairly expensive, power-wise, since the entire phase current flows through the series resistance and causes a considerable I^2R loss. Another point to note is the fact that when the phase is off, the entire supply

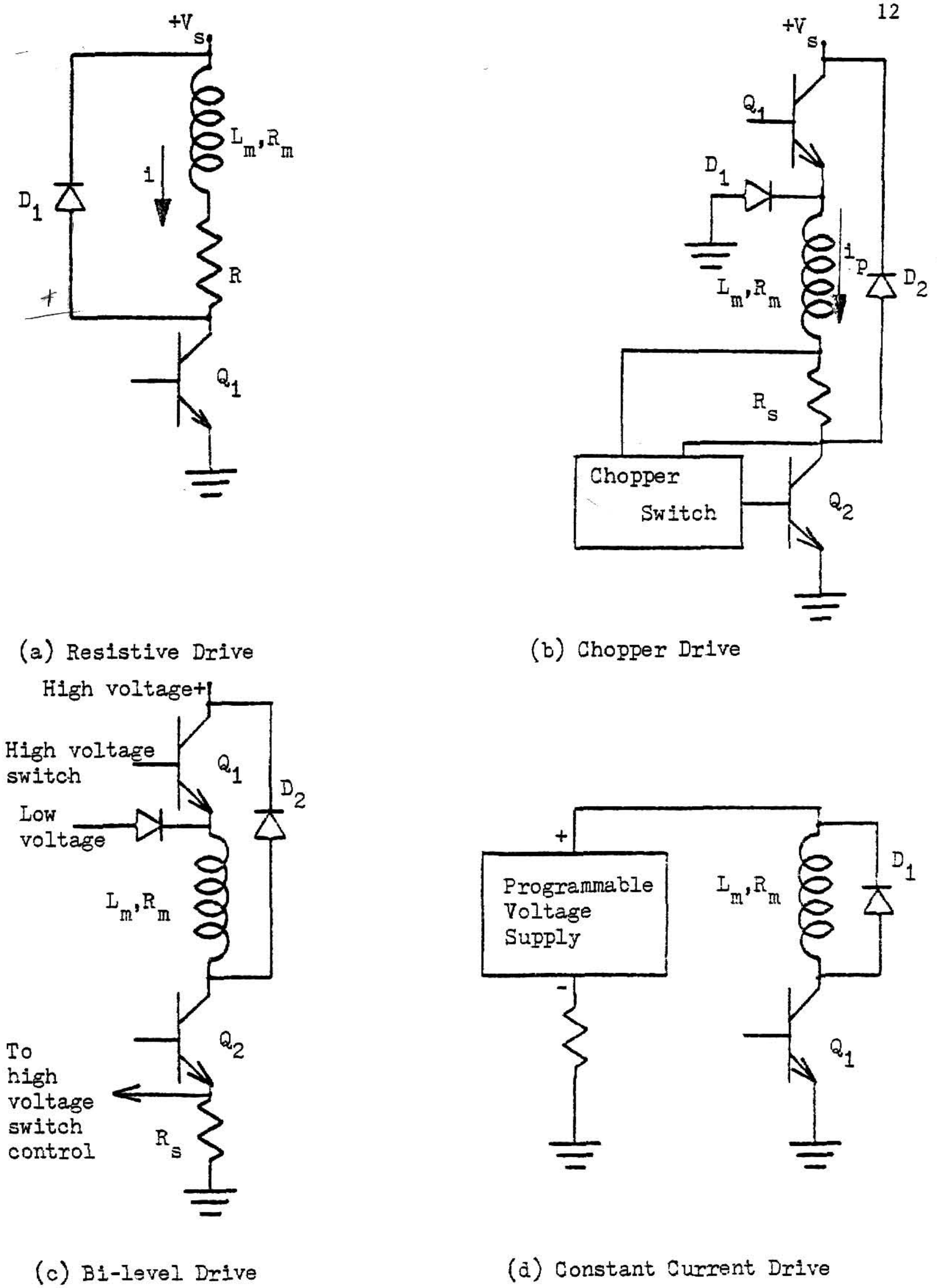


Fig. 1.8 Stepmotor Power Drives

voltage (V_s) appears across the transistor Q_1, which implies that its V_{CEO} rating must equal or exceed the value of the supply voltage.

B. Chopper drive

The chopper drive is a form of voltage modulation. In Fig.1.8b transistor Q_1 switches the phase winding on and off, and while Q_1 is on transistor Q_2 acts as the chopping switch. The voltage supply used (V_s) is in the range of ten times the motor's rated voltage. As soon as the phase current overshoots the rated value by a certain percentage (a circuit design parameter usually about 10%) (the chopper switch) Q_2 goes off and the current free wheels via diodes D_1 and D_2 until it undershoots the rated value by a certain percentage (again, about 10%).

To derive expressions for the phase current $i_p(t)$, it is convenient to consider the following time intervals, shown in Fig. 1.9

(1) In the interval $(0<t<t_1)$, with the initial condition

$$i_p(0_+) = 0$$

and the assumptions

a) $\tau_1 = L_m / R_1$

b) $R_1 = R_m + R_{q1} + R_{q2} + R_s$

$i_p(t)$ would be expressed as

$$i_p(t) = (V_s / R_1)(1 - e^{-t/\tau_1}). \tag{1.3a}$$

With a 10% overshoot, and assuming a rated current value of i_r the value of the phase current at the end of this interval can be expressed as

$$i_p(t_1) = 1.1\, i_r \quad .$$

(2) To derive an expression for $i_p(t)$ in the interval $(t_1 < t < t_2)$ with with diodes D_1 and D_2 on, the following voltage loop equation is used

$$V_s = R_2 i_s(t) + L_m \frac{di_s(t)}{dt} \tag{1.4a}$$

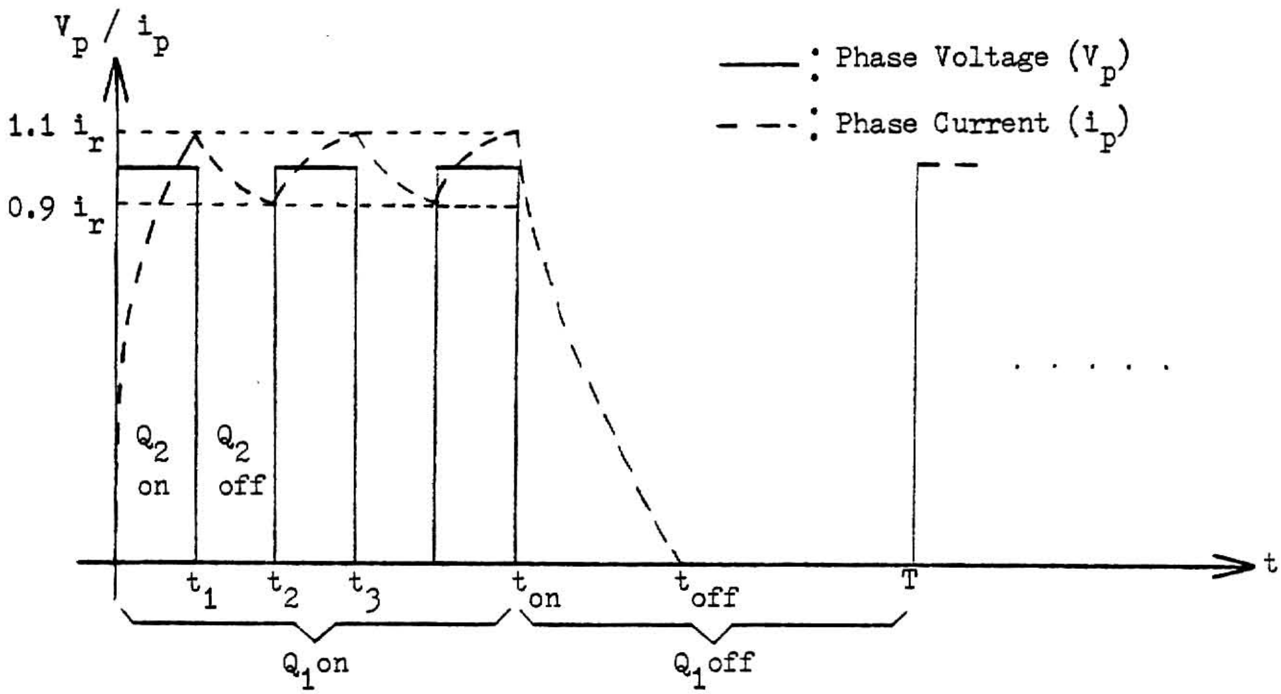


Fig. 1.9 Chopper drive waveform

where $R_2 = R_m + R_{d1} + R_{d2} + R_s$

and $i_s(t) = - i_p(t)$

The initial condition is $i_s(0_+) = -1.1\, i_r$

and time constant is $\tau_2 = L_m/R_2$.

Upon solving for $i_s(t)$ one finds

$$i_s(t) = \frac{V_s}{R_2} - (1.1 i_r + \frac{V_s}{R_2}) e^{-t/\tau_2} \qquad (1.4b)$$

which implies

$$i_p(t) = -\frac{V_s}{R_2} + (1.1 i_r + \frac{V_s}{R_2}) \, e^{-t/\tau_2} \qquad (1.4c)$$

at the end of this time interval

$$i_p(t) = 0.9\, i_r \ .$$

(3) The expression for $i_p(t)$ in the next time interval $(t_2 < t < t_3)$ is similar to the one derived in the interval $(0 < t < t_1)$ except for the initial condition

$$i_p(t_2) = 0.9\, i_r \ .$$

The phase current is given by the equation

$$i_p(t) = \frac{V_s}{R_1} + (0.9 i_r - \frac{V_s}{R_1}) e^{-t/\tau_1} \qquad (1.4d)$$

at the end of this interval $i_p(t_3) = 1.1\, i_r$.

(4) Finally in the interval $(t_{on} < t < t_{off})$ with the initial condition

$$i_p(t_{on}) = 1.1\, i_r$$

the phase current

$$i_p(t) = -\frac{V_s}{R_2} + (1.1 i_r + \frac{V_s}{R_2}) e^{-t/\tau_2} . \qquad (1.4e)$$

The time for the phase current to decay to zero is given by the equation

$$(t_{off} - t_{on}) = -\tau_2 \ln\left[\frac{(V_s / R_2)}{1.1 i_r + V_s / R_2}\right] .$$

C Bilevel drive

The bilevel drive (Fig.1.8c) involves the use of a dual voltage supply, having a high voltage level to build up current at a rapid rate in the phase winding, and a low level supply to maintain the rated current once it is reached.

A sensing resistor, in series with the phase winding, is used to switch from the high voltage supply to the low voltage supply by controlling the transistor switches Q_1 and Q_2.

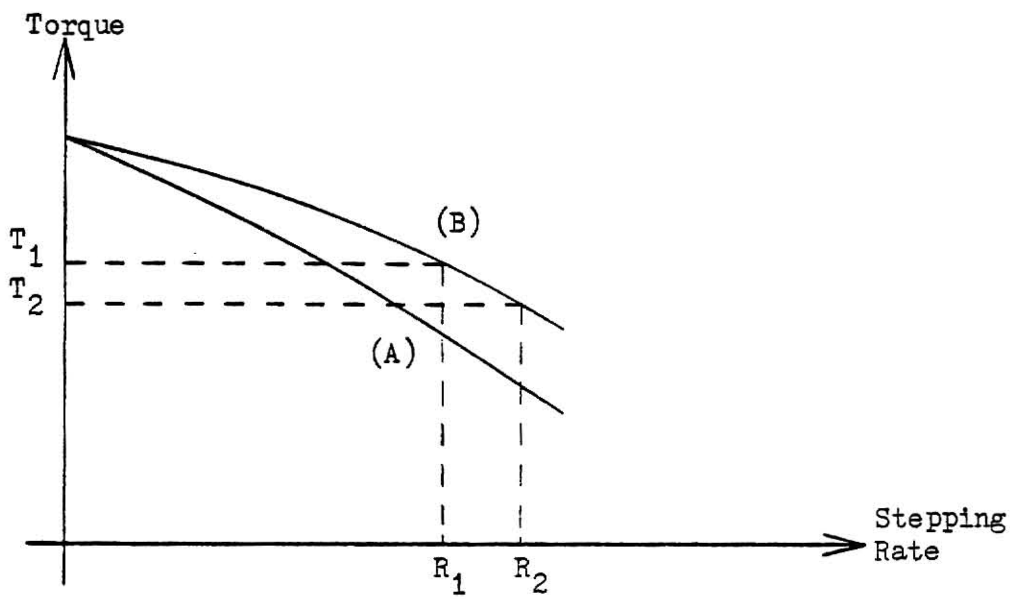
D Constant current source drive (Fig.1.8d)

This scheme involves the use of a programmable voltage source which ensures rated current supply regardless of the stepping rate. A major drawback of this drive is its slow response time, the motor speed would be limited by the slew rate of the voltage supply, thus limiting the motor's start-stop operation. However, in a system where the motor is ramped to its operating speed, this drive would be most suitable.

1.4 CONTROL ASPECTS

When energi ed, the step motor rotates and comes to rest in one of its distinct detent positions. Depending on the application, the motor can either be used in the single stepping mode (e.g. for positioning control) or in the continuous stepping mode (slewing mode). Since the rotor moves through a fixed angle for each input pulse, the step motor can be used in an open loop control scheme provided synchronism is maintained between the rotor steps and input pulses. This implies ensuring that the motor start / stop speed and operating speed (all measured in steps per second) do not exceed their rated values for a given load torque (Fig. 1.10).

For most hard-wired controllers, using an open loop mode of control, that is , with no feedback at all, it is fairly easy for the motor to lose synchronism, and the motor speed fluctuates widely. Closed loop control aids



(A) Torque vs. operating speed

(B) Torque vs. start-stop speed

Fig. 1.10 Step motor torque vs. speed characteristic in achieving higher speeds and improved speed stability. The feedback is obtained by means of optical encoders or magnetic Hall effect devices. Fig.1.11 shows a block diagram of a step motor control system.

One of the problems encountered in step motor control is the phenomenon of resonance. If the motor is operated at no-load over its entire speed range one or more natural resonance points might be detected, to compensate for this problem external dampers must be employed. A permanent magnet motor, as a result of its high rotor inertia and detent torque, does not show the instability and loss of steps often found in variable reluctance step motors.

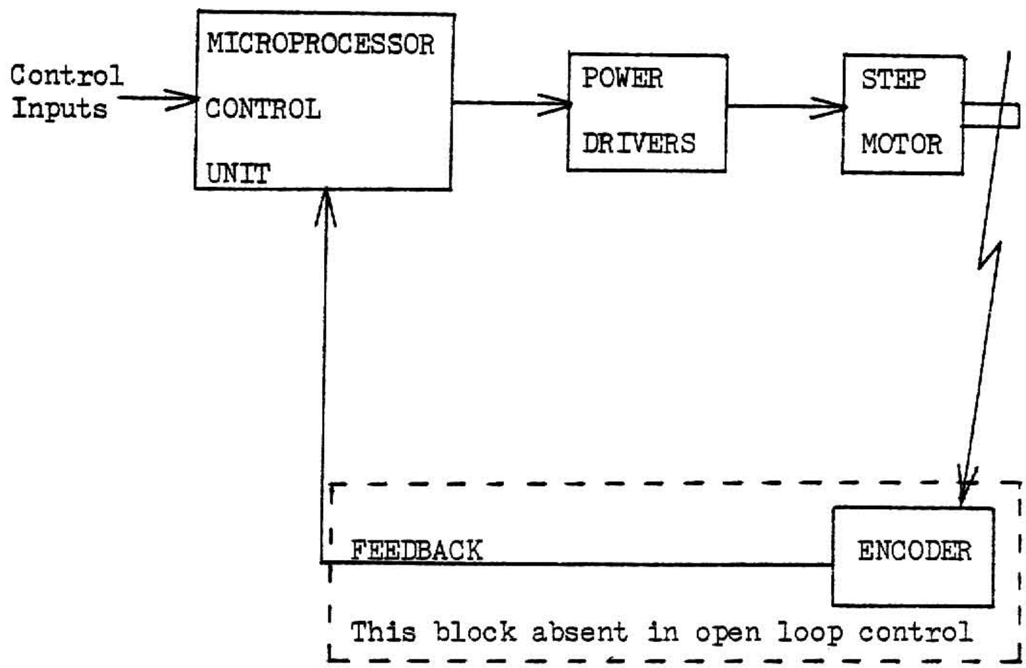


Fig. 1.11 Block diagram of step motor controller

CHAPTER 2

MATHEMATICAL MODELLING OF STEP MOTORS

2.0 DEVELOPMENT OF ELECTROMAGNETIC EQUATIONS

Essentially the theory underlying the operation of step-motors is the same as that of any rotating electromagnetic energy converter. It involves four laws :

1) Law of conservation of energy
2) Laws of electrical circuits
3) Laws of magnetic fields
4) Laws of mechanics

A physical device implies nonlinearities and losses which have to be suitably modelled. A primary model would incorporate the losses (for example, electrical and mechanical losses) while neglecting the nonlinearities. The results of such a study could then be corrected for the effects of the neglected factors.

The step-motor model for which the equations are developed is assumed to have four phases, but the equations can easily be extended for an n-phase motor. Fig. 2.1 shows a functional circuit diagram of a step motor. The flux linkage λ for each coil is a function of the rotor angle θ , winding currents (i_{1-4}) and the rotor current i_r

$$\lambda_j = f_j(\theta, i_1, i_2, i_3, i_4, i_r) \qquad 1 < j < 4 \quad . \qquad (2.1)$$

The phase voltages would then be

$$v_j = i_j r_j + \frac{d\lambda_j}{dt} \qquad 1 < j < 4 \quad . \qquad (2.2)$$

For a permanent magnet step-motor, the magnetomotive force (MMF_r) developed by the rotor can be expressed as

$$MMF_r = N_r i_r$$

where i_r is the effective equivalent current and N_r is the effective number of

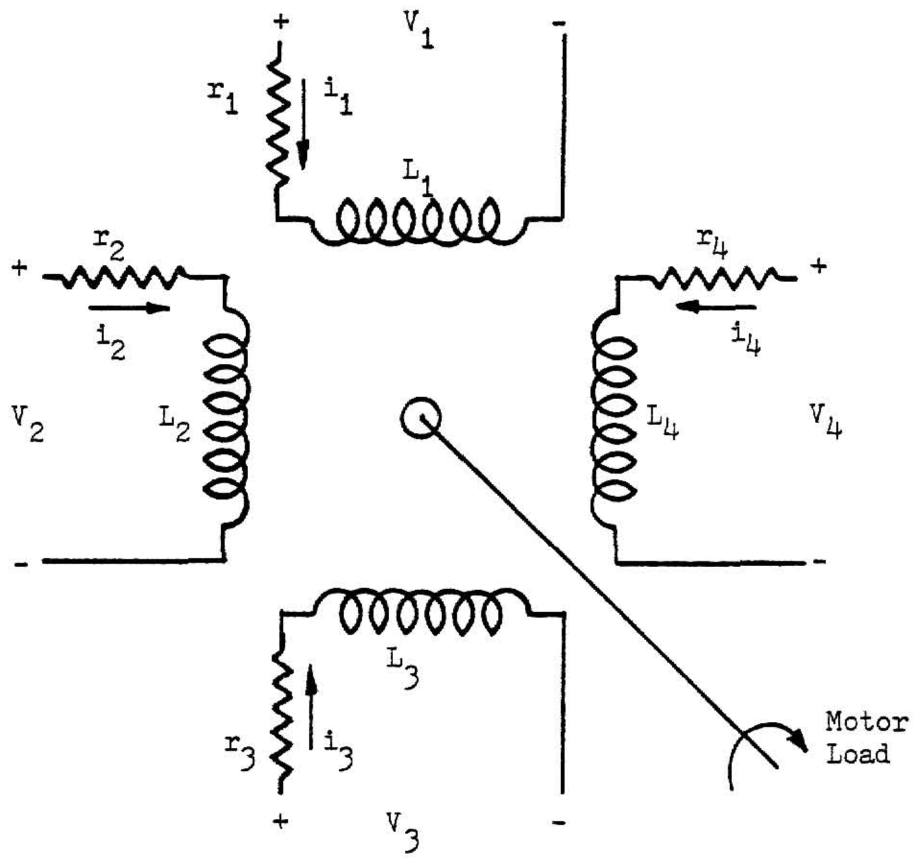


Fig. 2.1 Motor circuit representation

of turns.

The flux linkage of each stator coil (λ_j) can be expressed as

$$\lambda_j = \sum_{k=1}^{4} L_{jk} i_k + L_{jr} i_r \qquad 1 < j < 4 \, . \qquad (2.3)$$

The above equation assumes that the magnetic circuit does not saturate. L_{jj} represents the self inductance of each stator coil (j=1 to 4), L_{jk} ($j \neq k$) represents the mutual inductance between the j^{th} and k^{th} stator coil, L_{jr} represents the coupling between the j^{th} stator coil and the rotor. Combining equations 2.1 and 2.2, and applying the chain rule of differentiation yields the following result

$$v_j = i_j r_j + \sum_{k=1}^{4} \frac{d\lambda_j}{di_k} \frac{di_k}{dt} + \frac{d\lambda_j}{d\theta} \frac{d\theta}{dt} \qquad 1 < j < 4 \, . \qquad (2.4)$$

Assuming a linear magnetic circuit equations 2.3 and 2.4 can be combined to yield

$$v_j = i_j r_j + \sum_{k=1}^{4} L_{jk} \frac{di_k}{dt} + i_r \frac{dL_{jr}}{d\theta} + \sum_{m=1}^{4} i_m \frac{dL_{jm}}{d\theta} \frac{d\theta}{dt} \qquad 1 < j < 4 \, . \qquad (2.5)$$

In the above equations r_j represents the resistance of the j^{th} stator phase, the inductance terms (L_{jk}) are all functions of θ, the rotor angle measured from a fixed reference. This is to be expected since the rotor forms a closing link in the magnetic circuit for each of the stator phases.

The equations developed so far describe the electrical circuit of the step-motor where the phase voltages represent the external stimuli.

To develop an equation for the torque developed by the motor, the electrical and magnetic circuits must be interlinked. For a linear magnetic circuit, energy and coenergy can be used interchangeably.The energy W and coenergy W' of a magnetic system,with flux linkage λ and magnetising current i , are related as

$$W' = i\lambda - W \, .$$

The torque developed by a linear magnetic circuit, can be expressed as

$$\text{Torque}(T) = \frac{-dW(\lambda,\theta)}{d} = \frac{dW'(i,\theta)}{d} \quad . \qquad \begin{array}{l} W = \text{magnetic fld. energy} \\ W' = \text{magnetic fld. coenergy} \end{array}$$

In the above equation it is assumed that the magnetizing current is held constant.

The torque developed acts in a direction so as to decrease the stored energy at constant flux or to increase the stored energy and coenergy at constant magnetomotive-force (MMF), or to decrease the reluctance and to increase the permeance and inductance.

The magnetic field energy relationship, when used to develop an equation

$$W = W(\lambda_1,\lambda_2,\lambda_3,\lambda_4,\lambda_r,\theta) \quad . \tag{2.6}$$

For the static case

$$dW = i d\lambda \quad .$$

Hence,

$$W = \int_0^1 i_1 d\lambda_1 + \int_0^2 i_2 d\lambda_2 + \int_0^3 i_3 d\lambda_3 + \int_0^4 i_4 d\lambda_4 + \int_0^r i_r d\lambda_f \quad . \tag{2.7}$$

Using equations 2.3 and 2.7 yields the result

$$W = \frac{1}{2}\sum_{j=1}^{4} L_{jj} i_j^2 + \frac{1}{2} L_{rr} i_r^{\,2} + \sum_{k=1}^{4} \sum_{m=1}^{j-1} L_{km} i_k i_m + \sum_{n=1}^{4} L_{nr} i_n i_r \quad . \tag{2.8}$$

In deriving the above equation the rotor has been represented by a coil carrying a current of i_r, self inductance of L_{rr}, and mutual inductance of L_{jr} (between the rotor coil and the j^{th} stator coil).

Since energy and coenergy can be used interchangeably, the torque can be expressed as

$$\text{Torque}(T) = \frac{dW}{d\theta}$$

combining this with equation 2.8 yields

$$T = \frac{1}{2}\sum_{j=1}^{4} i_j^2 \frac{dL_{jj}}{d\theta} + \frac{1}{2} i_r^2 \frac{dL_{rr}}{d\theta} + \sum_{k=1}^{4}\sum_{m=1}^{j-1} i_k i_m \frac{dL_{km}}{d\theta} + \sum_{n=1}^{4} i_n i_r \frac{dL_{nr}}{d\theta} \qquad (2.9)$$

2.1 PERMANENT MAGNET STEP MOTOR MODEL

The following discussion is based on a paper by H.D.Chai(1). Some of the equations have been developed to a greater length,and also take into consideration additional harmonic terms.

The motor under consideration in the following discussion is the same as the one used in the project, and has been described in the hardware appendix. However, for the purpose of developing the step motor model some of the physical design considerations have to be dealt with. The rotor has the permanent magnet attached to its length, parallel to its axis, and it has two pole pieces attached at either end (Fig.2.2). Each pole piece has 50 teeth.

The stator windings (four phases) are bifilar wound. As shown in Fig. 2.3 this implies that phases A_1 and A_2 are wound on the same set of four pole pieces, but in opposite directions so that they produce opposing magnetic flux. Fig.2.3 represents the stator pole pieces corresponding to the left pole piece on the rotor, a similar set of pole pieces surround the right rotor pole piece as well. The above description applies to phase windings B_1 and B_2 as well. Each stator pole piece has 48 teeth and the right and left rotor pole pieces are offset by one-half of a rotor tooth pitch. This construction gives the step motor a step size of 1.8 degrees for the single phase on or the two phase on modes of operation, and a step size of 0.9 degrees for the mixed mode of operation.

Fig.2.4, which represents the motor equivalent circuit, has been constructed on the basis of the principle of superposition. Upon supressing

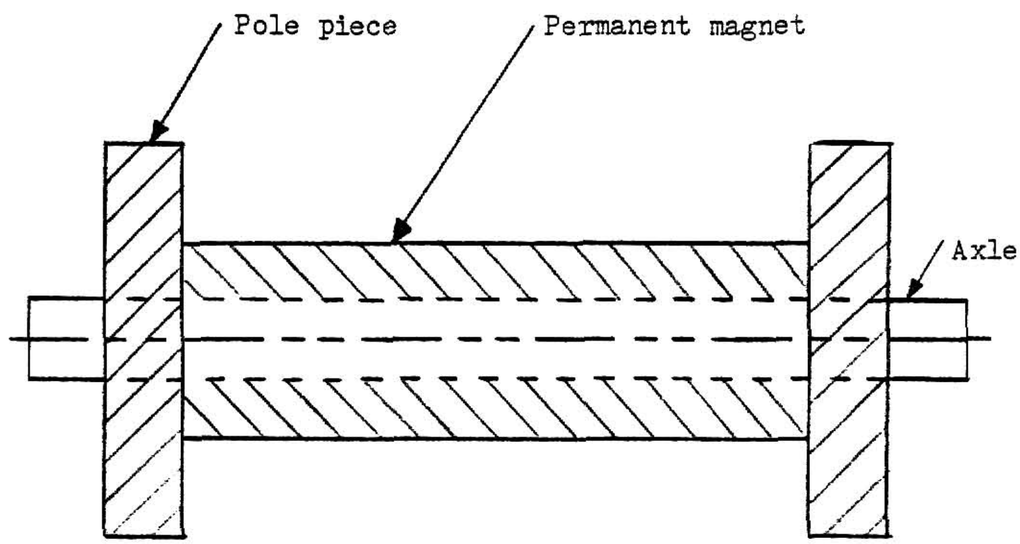


Fig. 2.2 Permanent magnet rotor

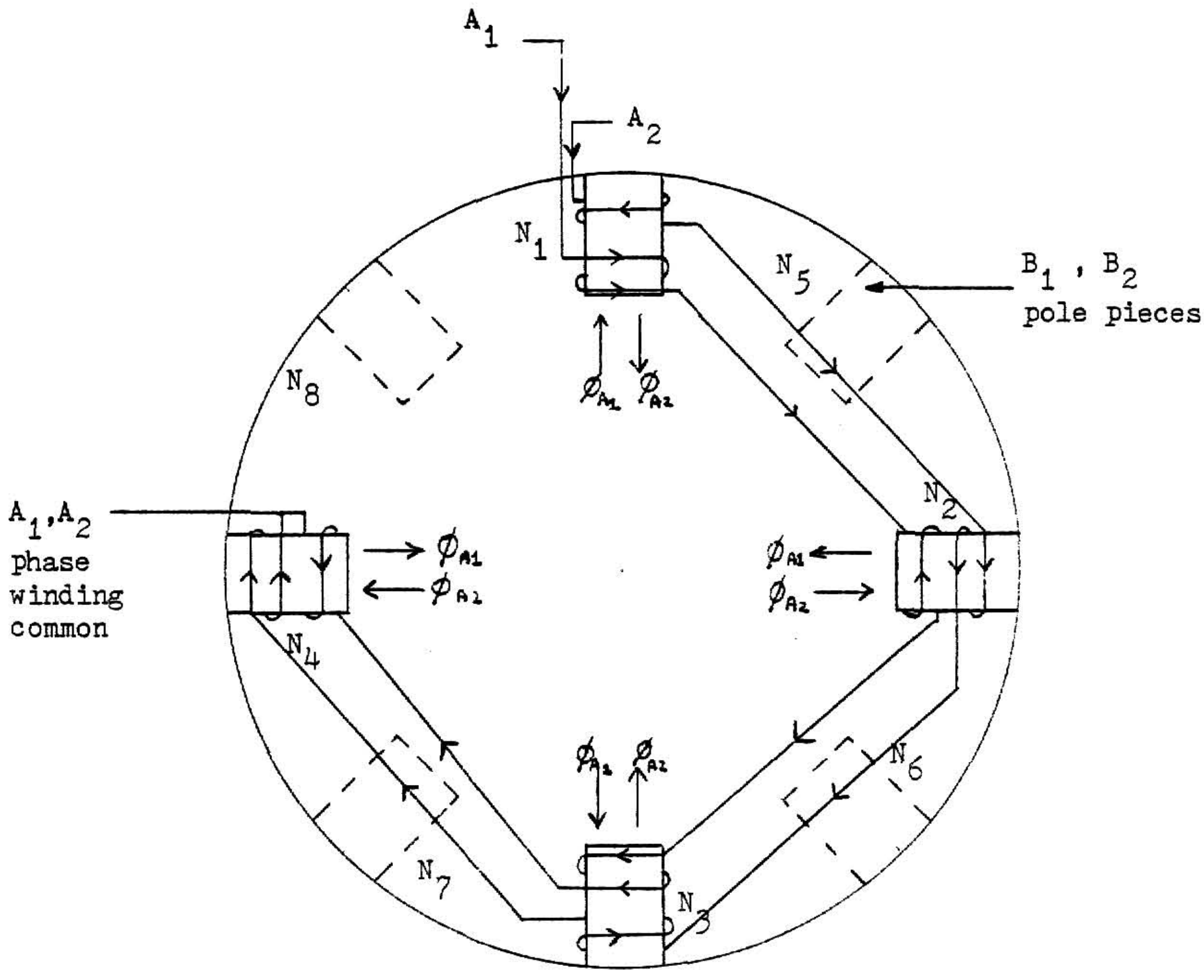
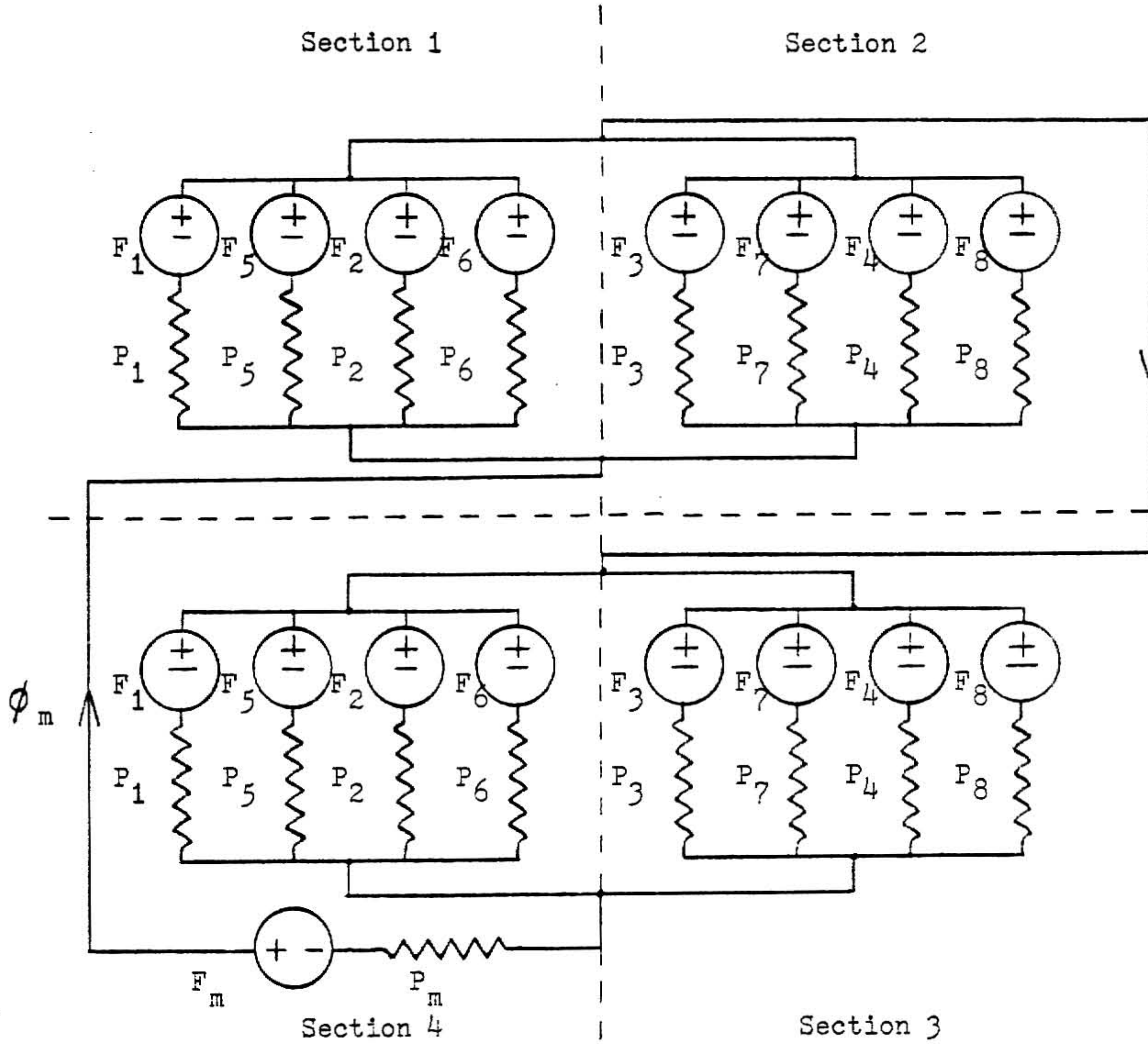


Fig. 2.3 Bifilar phase winding for stator phases



P_j : Respective stator winding air gap permeances. $1 < j < 8$

F_j : Respective stator winding MMF's

P_m : Permanent magnet permeance

F_m : Permanent magnet MMF

ϕ_m : Flux in magnetic circuit

Fig. 2.4 Magnetic circuit equivalent of step motor

MMF sources save for the permanent magnet (F_m), Fig. 2.4 represents the effective reluctance seen by the permanent magnet. To complete the equivalent circuit the stator coil MMF's are superposed. The construction symmetry is such that when a particular phase is energized (phase A_1 in Fig. 2.3) the flux enters the rotor via poles which are diameterically apart (N_1, N_3) and exists via a pair of poles (N_2, N_4) which are 90 degrees out of phase with the poles through which the flux entered the rotor. The symmetry implies that the magnetic equivalent circuit can be split into four identical sections, numbered 1 through 4 in Fig. 2.4. This reduces the analysis of the step motor circuit to the analysis of one of these sections, since the effect of each one of the four sections is additive. Fig. 2.5 shows the equivalent circuit of one of these sections.

Performing a Thevenin-Norton transformation on the circuit shown in Fig. 2.5 yields the Norton equivalent (Fig. 2.6). It is this circuit which is used to develop most of the following equations. In Fig. 2.6, since all the circuit elements are in parallel, the following equation holds true

$$F_0 = \frac{P_1F_1 + P_2F_2 + P_3F_3 + P_4F_4 + P_mF_m}{P_1 + P_2 + P_3 + P_4 + P_m} \qquad (2.10)$$

The circuit in Fig. 2.5 yields the following equations

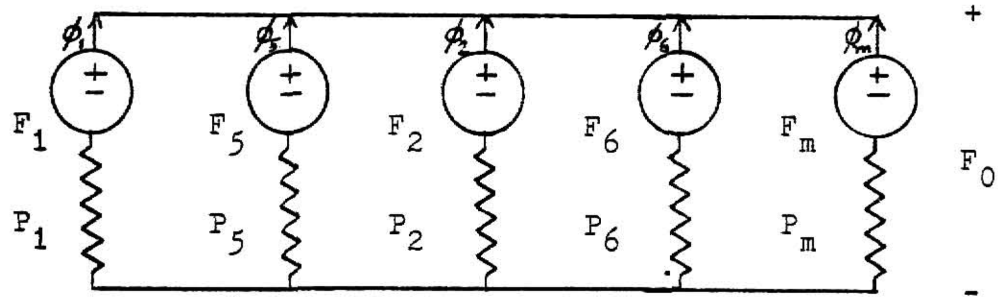
$$\phi_i = (F_i - F_0)P_i \qquad i = 1, 2, 3, 4 \qquad (2.11)$$

To derive an equation for the static torque on the rotor, the law of conservation of energy is used. Since the change in mechanical energy is zero for the static case

$$dW_{mech} = 0$$

therefore the electrical energy input to the motor will be equal to the change in stored magnetic field energy

$$dW_{elec} = dW_{fld} = Fd\phi$$



F_j : Corresponding MMF values.

P_j : Corresponding permeance values.

ϕ_j : Corresponding flux in each branch of the circuit.

F_0 : MMF drop across stator rotor gap.

Fig. 2.5 Magnetic circuit equivalent of Sec. 1 of fig.2.4

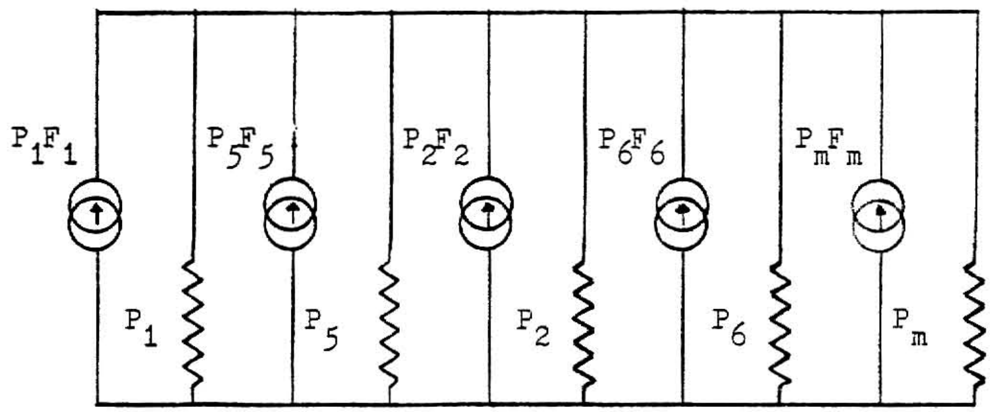


Fig. 2.6 : Norton's equivalent of circuit shown in fig.2.5

The above definition for magnetic field energy combined with the definition of torque as a function of the stored magnetic field energy

$$T_j = \frac{-dW_{fld_j}}{d\theta} \qquad j = 1,2,3,4 \qquad (2.12)$$

yields a solution for the static torque as follows

$$W_{fld_j} = \int_0^{\phi_j} (F_j - F_0)\,d\phi_j \qquad j = 1,2,3,4 \qquad (2.13)$$

substituting equation 2.11 in equation 2.13 and integrating yields

$$W_{fld} = \tfrac{1}{2}\,{}^{2}_{j}\,R_j \qquad j = 1,2,3,4 \qquad (2.14a)$$

which, when substituted in equation 2.12 results in

$$T_j = \frac{-d}{d\theta}\left(\tfrac{1}{2}\phi_j^2 R_j\right) \qquad j = 1,2,3,4\ . \qquad (2.14b)$$

Using the principle of superposition to obtain the total torque

$$T = \sum_{j=1}^{4} T_j$$

implies

$$T = -\tfrac{1}{2}\sum_{j=1}^{4} \phi_j^2 \frac{dR_j}{d\theta} \qquad (2.15)$$

since $\quad R_j = 1/P_j\ ,\ dR_j = -P_j^2 dP_j\ . \qquad j = 1,2,3,4$

Upon substituting this in equation 2.15

$$T = \tfrac{1}{2}\sum_{j=1}^{4} \frac{\phi_j^2}{P_j^2}\frac{dP_j}{d\theta}$$

which when combined with equation 2.11 yields

$$T = \tfrac{1}{2}\sum_{j=1}^{4} (F_j - F_0)\frac{dP_j}{d\theta_m}\ . \qquad (2.16)$$

In the last equation θ_m represents the mechanical (physical) angle moved by the rotor. For the step motor, it would be easier to express angles in electrical degrees or radians rather than mechanical degrees or radians, since the number of permeance cycles for every mechanical cycle (revolution) performed by the rotor will equal the number of rotor teeth, to this end one defines

$$\theta_e = N_r \theta_m \tag{2.17}$$

where θ_e = Electrical angle

θ_m = Mechanical angle

N_r = Number of rotor teeth

Therefore $d\theta_e = N_r d\theta_m$

and substituting this in equation 2.16 yields

$$T = (N_r/2)\sum_{j=1}^{4}(F_j - F_0)^2 \frac{dP_j}{d\theta_e} \tag{2.18}$$

Next one needs to derive an expression for the permeance seen by each stator coil. These expressions will all be periodic with respect to θ_e, assuming symmetric teeth on the rotor, however they will differ in phase. This implies that the permeance values (P_j , j=1,2,3,4) can be expressed as a Fourier Cosine series. Retaining terms only till the fourth harmonic, the expressions are as follows

$$P_1 = P_{10} + P_{11}\text{Cos}\,\theta_e + P_{12}\text{Cos}2\theta_e + P_{13}\text{Cos}3\theta_e + P_{14}\text{Cos}4\theta_e \qquad \text{--(2.19a)}$$

$$P_2 = P_{10} + P_{11}\text{Cos}(\theta_e - 90) + P_{12}\text{Cos}2(\theta_e - 90) + P_{13}\text{Cos}3(\theta_e - 90) + P_{14}\text{Cos}4(\theta_e - 90) \qquad \text{--(2.19b)}$$

$$P_3 = P_{10} + P_{11}\text{Cos}(\theta_e - 180) + P_{12}\text{Cos}2(\theta_e - 180) + P_{13}\text{Cos}3(\theta_e - 180) + P_{14}\text{Cos}4(\theta_e - 180) \qquad \text{--(2.19c)}$$

$$P_4 = P_{10} + P_{11}\text{Cos}(\theta_e - 270) + P_{12}\text{Cos}2(\theta_e - 270) + P_{13}\text{Cos}3(\theta_e - 270) + P_{14}\text{Cos}4(\theta_e - 270)\,. \qquad \text{--(2.19d)}$$

Adding up the above permeance expressions and applying some trignometric identities yields

$$\sum_{j=1}^{4} P_j = 4P_{10} + 4P_{14}\text{Cos}4\theta_e \quad . \tag{2.20}$$

Differentiating the permeance expressions and adding up the results yields, after some manipulations,

$$\sum_{j=1}^{4} \frac{dP_j}{d\theta_e} = - 16P_{14}\mathrm{Sin}4\theta_e \quad . \tag{2.21}$$

Permanent magnet step motors posses detent torque and an expression is derived for this torque (T_{rd}) using the equations developed so far. From equation 2.10, setting $F_1 = F_2 = F_3 = F_4 = 0$, to signify zero stator excitation, yields

$$F_0 = \frac{P_m F_m}{4P_{10} + 4P_{14}\mathrm{Cos}4\theta_e + P_m} \quad . \tag{2.22}$$

Substituting the above result for F_0 in equation 2.18 and neglecting the fourth order harmonic coefficient (P_{14}) in comparison to P_{10} and ($P_m/4$), in the denominator, leads to the result

$$T_{rd} = - (N_r P_m^2 F_m^2 P_{14}) \,/\, 2(P_{10} + \tfrac{1}{4}P_m)^2 \;\; \mathrm{Sin}4\theta_e \quad . \tag{2.23}$$

The above equation shows that T_{rd} varies sinusoidally with respect to θ_e. The stable points (minimum reluctance with respect to one of the stator poles) exist at θ_e= 0,90,180, and 270 degrees respectively, while the unstable points are at θ_e= 45,135,225, and 315 degrees. The high order harmonic terms which have been neglected tend to shift these stable/unstable points but the effect is negligible.

The same strategy which was used to develop an equation for the detent torque, can be used to develop an equation for the static torque when the stator windings are energized. As an example the expression for static torque for the case of single phase excitation has been derived. The basic assumption made is that opposing stator windings in the 4-phase single section equivalent motor are connected in series. In fig.2.5, this would imply that F_1 and F_3 are in series and F_2 and F_4 are in series. Furthermore the coils are connected such that $F_1 = -F_3$ and $F_2 = -F_4$. Thus, if coils F_1 and F_3 are excited

$$F_1 = -F = -F_3$$

and $$F_2 = F_4 = 0$$

Substituting the above values in equation 2.10 and using the permeance values from equations 2.19(a-d) yields

$$F_0 = \frac{-2F(P_{11}\mathrm{Cos}\theta_e + P_{13}\mathrm{Cos}3\theta_e) + F_m P_m}{4P_{10} + 4P_{14}\mathrm{Cos}4\theta_e + P_m} \quad . \qquad (2.24)$$

Denoting the static torque developed by the step motor in the one phase on mode of operation, as T_{2s} and applying the superposition rule, one gets the following expression from equation 2.18 after substituting for F_j, $(j=1,2,3,4)$ and the derivatives of the permeances

$$T_{2s} = 4(N_r/2)\Big[F^2(-4P_{12}\mathrm{Sin}2\theta_e - 8P_{14}\mathrm{Sin}4\theta_e) - 16F_0^2P_{14}\mathrm{Sin}4\theta_e - 4FF_0(P_{11}\mathrm{Sin}\theta_e + 3P_{13}\mathrm{Sin}3\theta_e)\Big] \qquad \text{--}(2.25)$$

Substituting for F_0 from equation 2.24 (neglecting the P_{14} term in the denominator) and ignoring cross product terms involving the product $P_{13}P_{14}$ yields

$$T_{2s} = 2N_r\Bigg[\frac{-4FF_mP_mP_{11}}{(4P_{10}+P_m)}\mathrm{Sin}\theta_e + \Bigg[-4F^2P_{12} - \frac{16F^2P_{11}^2P_{14}}{(4P_{10}+P_m)^2} + \frac{4F^2P_{11}^2}{(4P_{10}+P_m)} - \frac{4F^2P_{11}P_{13}}{(4P_{10}+P_m)} + \frac{24F^2P_{11}P_{13}}{(4P_{10}+P_m)}\Bigg]\mathrm{Sin}2\theta_e + \Bigg[\frac{32FF_mP_mP_{11}P_{14}}{(4P_{10}+P_m)^2} - \frac{12FF_mP_{13}P_m}{(4P_{10}+P_m)}\Bigg]\mathrm{Sin}3\theta_e + \Bigg[-8F^2P_{14} - \frac{16P_{14}(F_m^2P_m^2 + 2F^2P_{11}^2)}{(4P_{10}+P_m)^2} + \frac{28F^2P_{11}P_{13}}{(4P_{10}+P_m)}\Bigg]\mathrm{Sin}4\theta_e\Bigg] \quad . \qquad \text{--}(2.26)$$

The above equation for T_{2s} contains terms upto its fourth harmonic component deleting terms involving higher order permeance coefficients (P_{12} and on), the equation simplifies to

$$T = -2N_r \left(\frac{4FF_m P_m P_{11}}{(4P_{10} + P_m)} \cdot \sin\theta_e + \frac{4F^2 P_{11}^2}{(4P_{10} + P_m)} \cdot \sin 2\theta_e \right) \qquad (2.27)$$

normalizing the above equation by a factor (NF) equal to

$$NF = \frac{4N_r FF_m P_m P_{11}}{(4P_{10} + P_m)}$$

results in

$$T'_{2s} = -(\sin\theta_e + \frac{2FP_{11}}{N_r F_m P_m} \cdot \sin 2\theta_e) \quad . \qquad (2.28)$$

Equation 2.28 represents the normalized static torque developed by the step motor in the single phase on mode of operation, a similar analysis to the one used above yields an expression for the static torque developed in the two phase on mode of excitation. In this case the stator MMF's are as follows

$$F_1 = -F_3 = -F$$

and $$F_2 = -F_4 = -F$$

The result is the same as equation 2.27 with F replaced by 2F and θ_e replaced by $(\theta_e - 45^o)$.

Another interesting observation that can be made from equation 2.28 is the fact that as the magnitude of the stator coil MMF (F) is increased the effect of the second harmonic component increases and the variation of torque with θ_e approaches that of a variable reluctance step motor (proportional to $\sin 2\theta_e$). This is true assuming that the magnetic circuit remains linear with the increasing MMF.

2.2 CSMP SIMULATION

A computer simulation was carried out for a simplified motor phase model for the resistive and chopper drives. The programs were written in CSMP and the listings have been included as an appendix.

In both cases, resistive drive and chopper drive, the simulation considers a single motor phase as a simple L-R circuit, subject to a pulse train voltage input V(t), with different charging τ_c and discharging τ_d time constants ($\tau_d << \tau_c$). If the amplitude of the pulse train is assumed to be V_o, then the Laplace transform of V(t) is given by the equation

$$V(s) = \frac{V_o}{s} \cdot \frac{(1 - e^{-ps/2})}{(1 - e^{-ps})} \tag{2.20a}$$

p: reciprocal of pulse train frequency

The above equation holds true for the two phase on mode of operation. For the one phase on mode of operation the equation becomes

$$V(s) = \frac{V_o}{s} \cdot \frac{(1 - e^{-ps/4})}{(1 - e^{-ps})} \tag{2.20b}$$

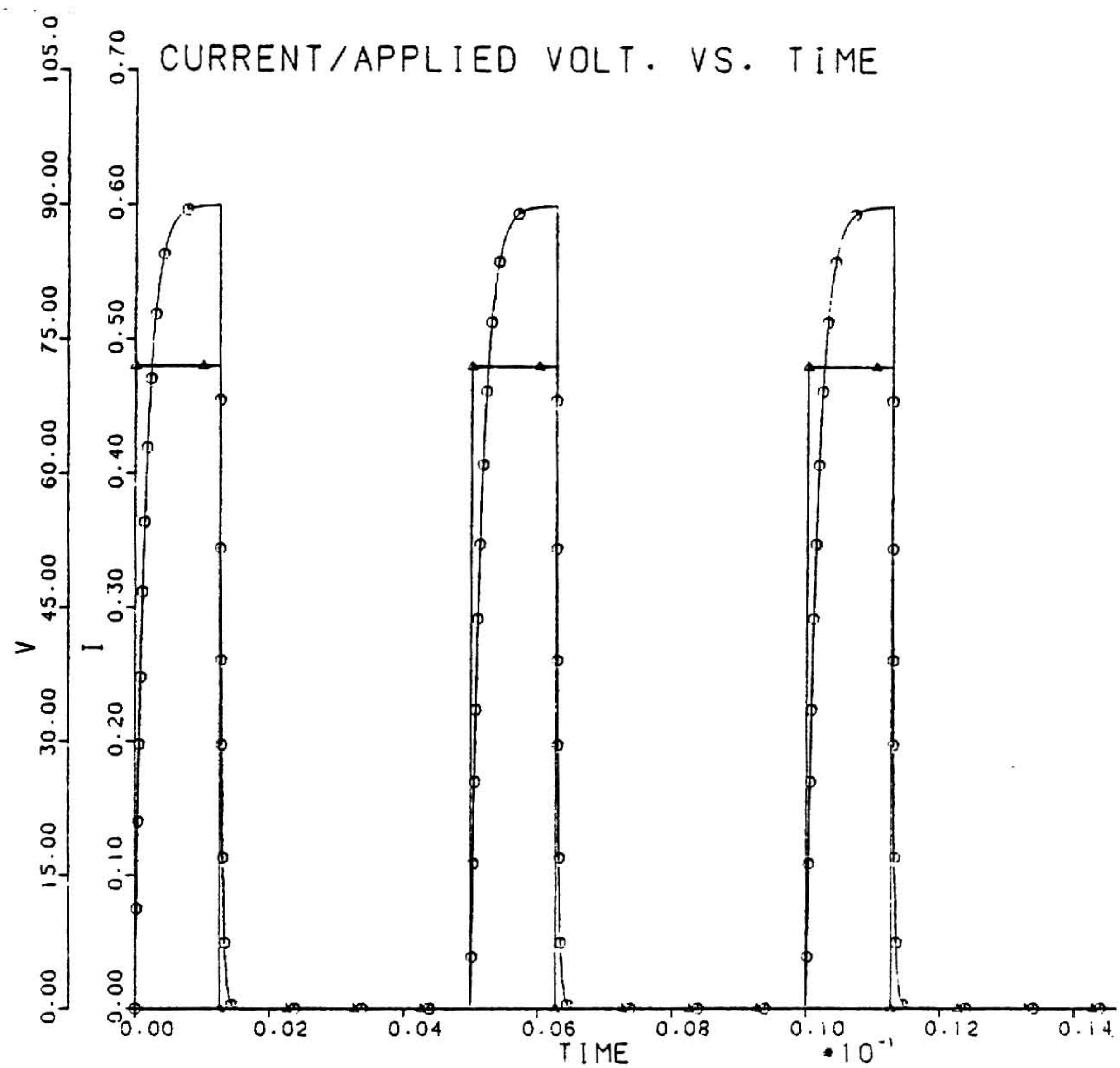
The Laplace transform I(s) of the phase current i(t) is then given by the equation

$$I(s) = \frac{V(s)}{R + sL} \tag{2.21}$$

In the above equation L is the phase inductance in Henries while R is the phase resistance in ohms.

In the CSMP programs, a set of FORTRAN statements accomplish the simulation of different charging and discharging time constants and, in the case of the chopper drive, the chopping action. The resulting phase current versus time plots are shown in Figs. 2.7 - 2.9.

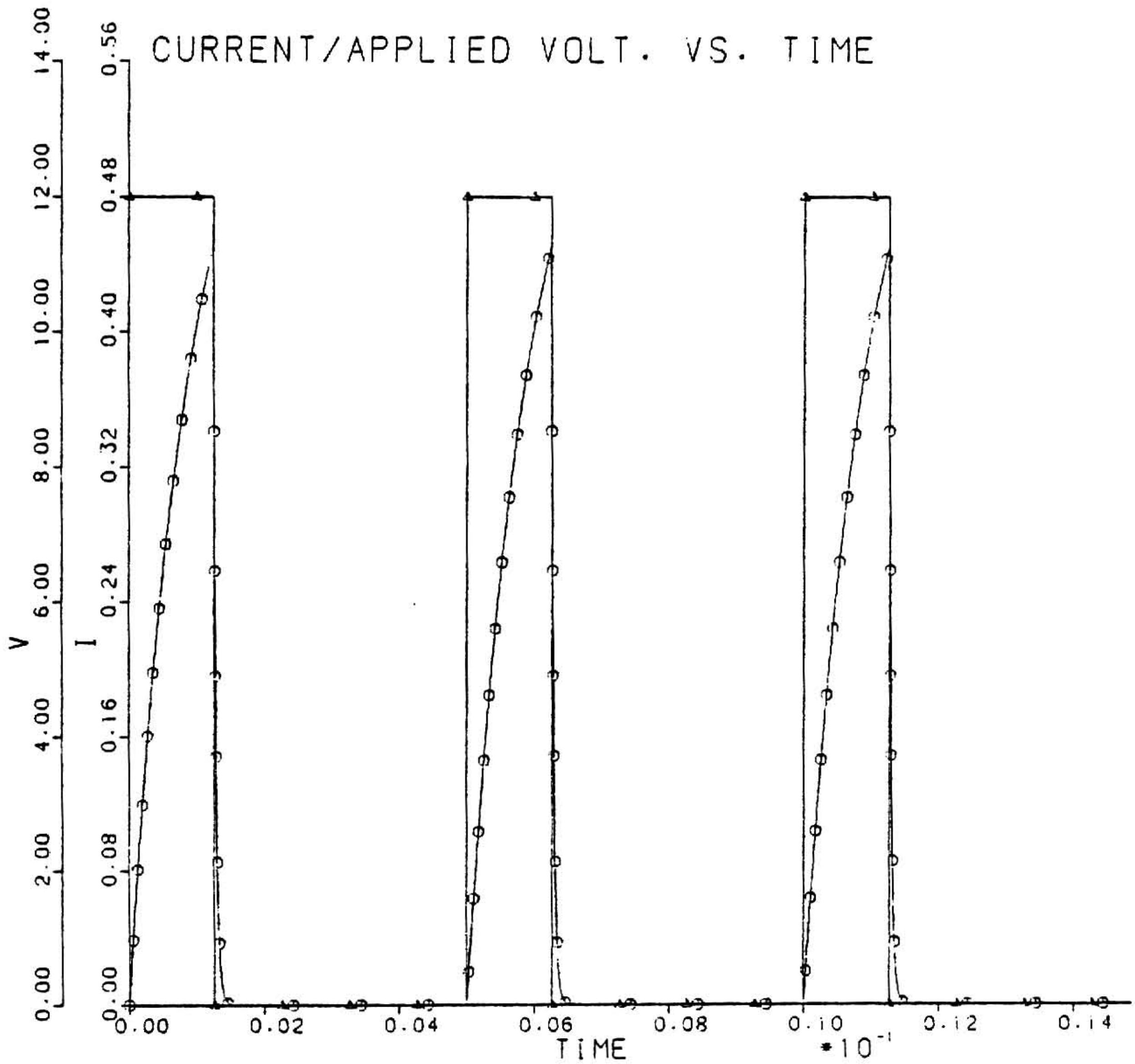
The plot shown in Fig. 2.7 results when no special drives are used and the phase current barely reaches a maximum of 0.45 amp., which is far below the rated phase current (0.6 amp.). Hence the motor tends to stall due to insufficient output torque. Fig. 2.8 shows the current plot for a resistive drive with an external resistor of 100 ohms and supply voltage of 72 volts. The phase current rapidly attains its maximum value, thus ensuring a strong output torque. However, it also implies an I^2R loss of approximately 36W in the external resistor. The final plot shown in Fig. 2.9 is the resulting phase current for a chopper drive. Only a fraction of the pulse time was plotted since the rapidly changing waveform drastically increased the program execution cost. This chopper drive yields a strong phase current and high output torque without increasing the I^2R losses.



Stepping rate = 800 sps.

One phase on mode of operation.

<u>Fig. 2.7 Simulation Plot #1</u>



Stepping rate = 800sps.

One phase on mode of operation.

Fig. 2.8 Simulation Plot #2

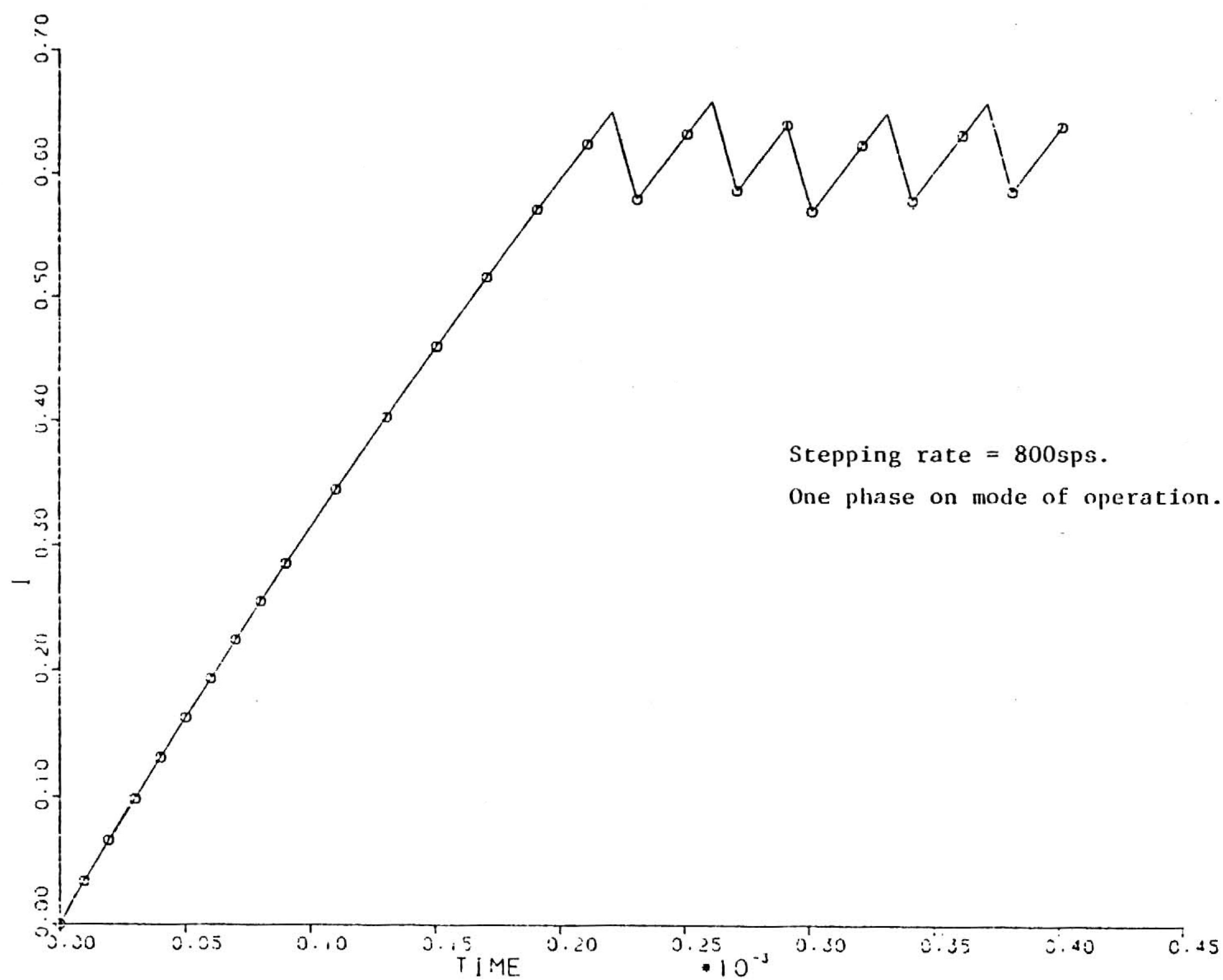


Fig. 2.9 Simulation Plot #3

CHAPTER 3

HARDWARE IMPLEMENTATION

3.0 OVERVIEW

The step motor used in the course of this study was a permanent magnet, bifilar wound four phase motor with a step size of 1.8°. The step size remains the same for the single phase and double phase modes of operation. However, for the mixed phase mode (single and double phase modes alternating) the step size is 0.9°.

The controller for the step motor was built around the KIM-1 microcomputer. This is a 8-bit micrccomputer system with a 6502 - nmos CPU chip. Further details about the KIM, including layout diagrams, are included in the following control section . The rest of the system consisted of the power drivers to energize the motor phases in a chosen fashion , power supplies and the feedback loop.

Two different types of power drives were implemented, the resistive drive and the chopper drive, for each of these types, different configurations were tried out and in both cases, the motor operated most smoothly if one end of each phase could be held at ground potential in the de-energizing, or free-wheeling, mode.

For the purpose of providing feedback, for closed loop control, an encoder disc with fifty slots was used along with a matched LED-photo transistor pair to provide a pulse train which was fed back to the microcomputer via a 4-bit counter.

The last section of this chapter lists some observations as well as comparisons between the resistive and chopper drive . Details of the hardware implementation including specifications and ratings have been included in the hardware appendix.

3.1 CONTROL SYSTEM

The overall system comprised of the KIM-1 microcomputer , the step motor and a protoboard setup with all the drive circuitry on board. A block diagram of the KIM-1 is shown in Fig. 3.1a. The KIM-1 system is built around the MCS6502 nmos 8-bit microprocessor. It has two 6530 multifunction device interrupts units on board, each of which provides 1-K of ROM, 64 bytes of RAM, 15 I/O pins and an interval timer. There is also an additional 1-K of RAM on board. The system clock operates at a frequency of 1 MHz. The KIM operates in the memory mapped mode of I/O and has dual means of program input, output, the keyboard display unit and the TTY. For application purposes the system provides 15 I/O lines.

The least significant four bits of port B, on the 6530-3 chip, are used to control the motor phases (PB_0 for phase B_2, PB_1 for phase B_1, PB_2 for phase A_2, PB_3 for phase A_1). The least significant four bits of port A (PA_{0-3}) were used to read in the counter feedback, PA_6 was used to reset the counter, PA_5 controlled the LED which indicated the end of deramping and PA_4 was set up as an input bit to read in the 'start deramping' switch status.

Fig. 3.1b shows a memory map of the KIM-1 system. A description of the protoboard is included as part of the motor drive description in following sections.

3.2 RESISTIVE POWER DRIVE

The resistive power drive is the simplest step motor drive to implement. Fig. 3.2 shows the overall drive circuit. Each motor phase L_m, R_m has a resistor R_s in series with it. Since the phase time constant τ_p is given by the equation

$$\tau_p = \frac{L_m}{R_m + R_s}$$

As the value of R_s is increased, τ_p decreases thus permitting higher motor operating speeds. The cost entailed by this scheme is the I^2R loss in the series

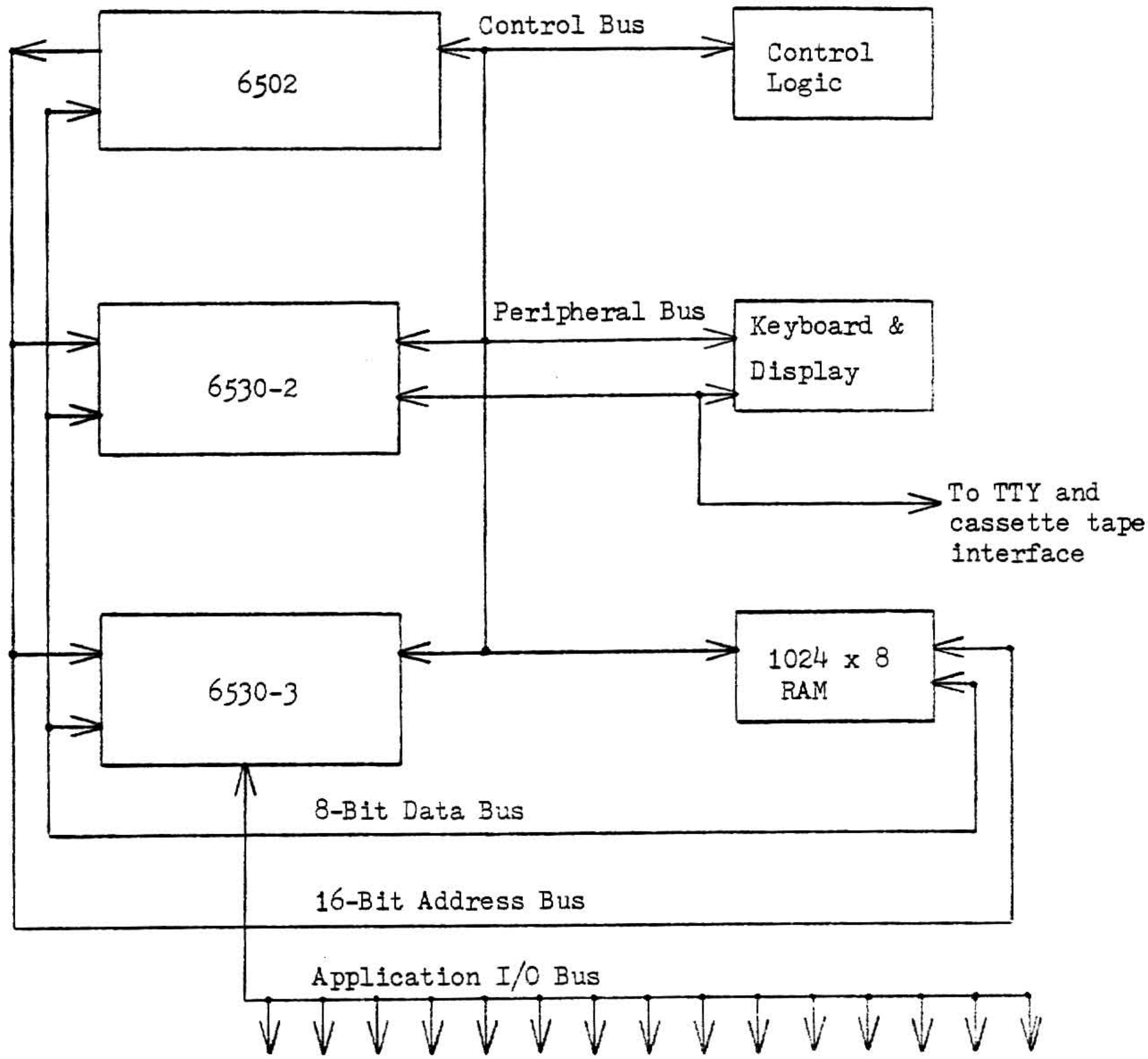
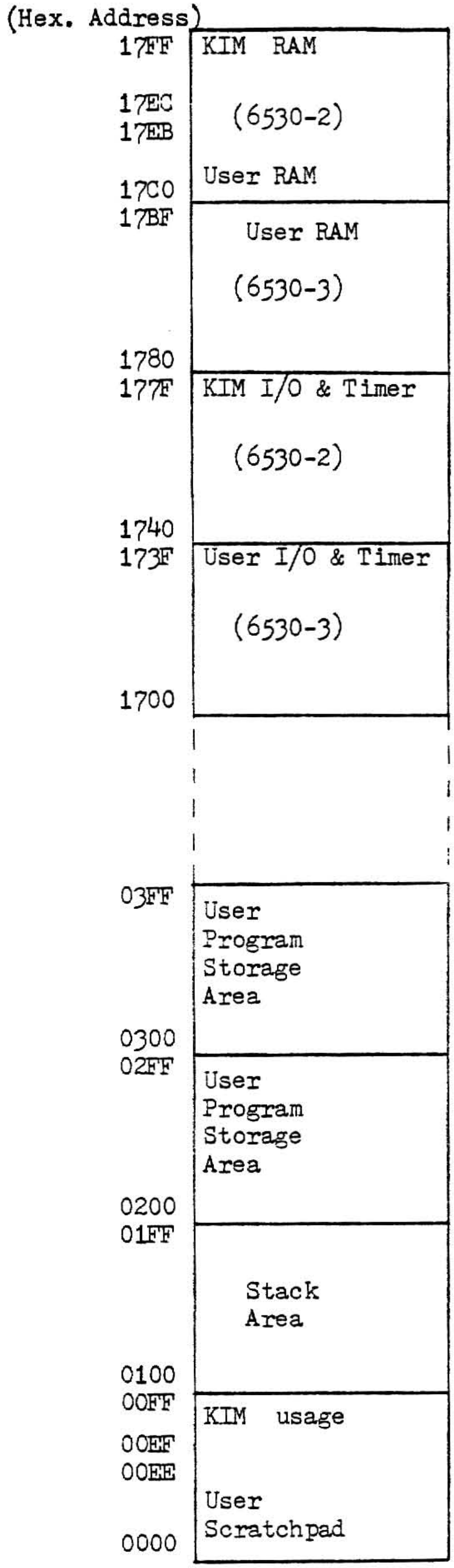


Fig. 3.1a KIM-1 block diagram



Note : Address space 0400_{16} to $16FF_{16}$ and 1800_{16} is available for expansion.

Fig. 3.1b KIM Memory Map

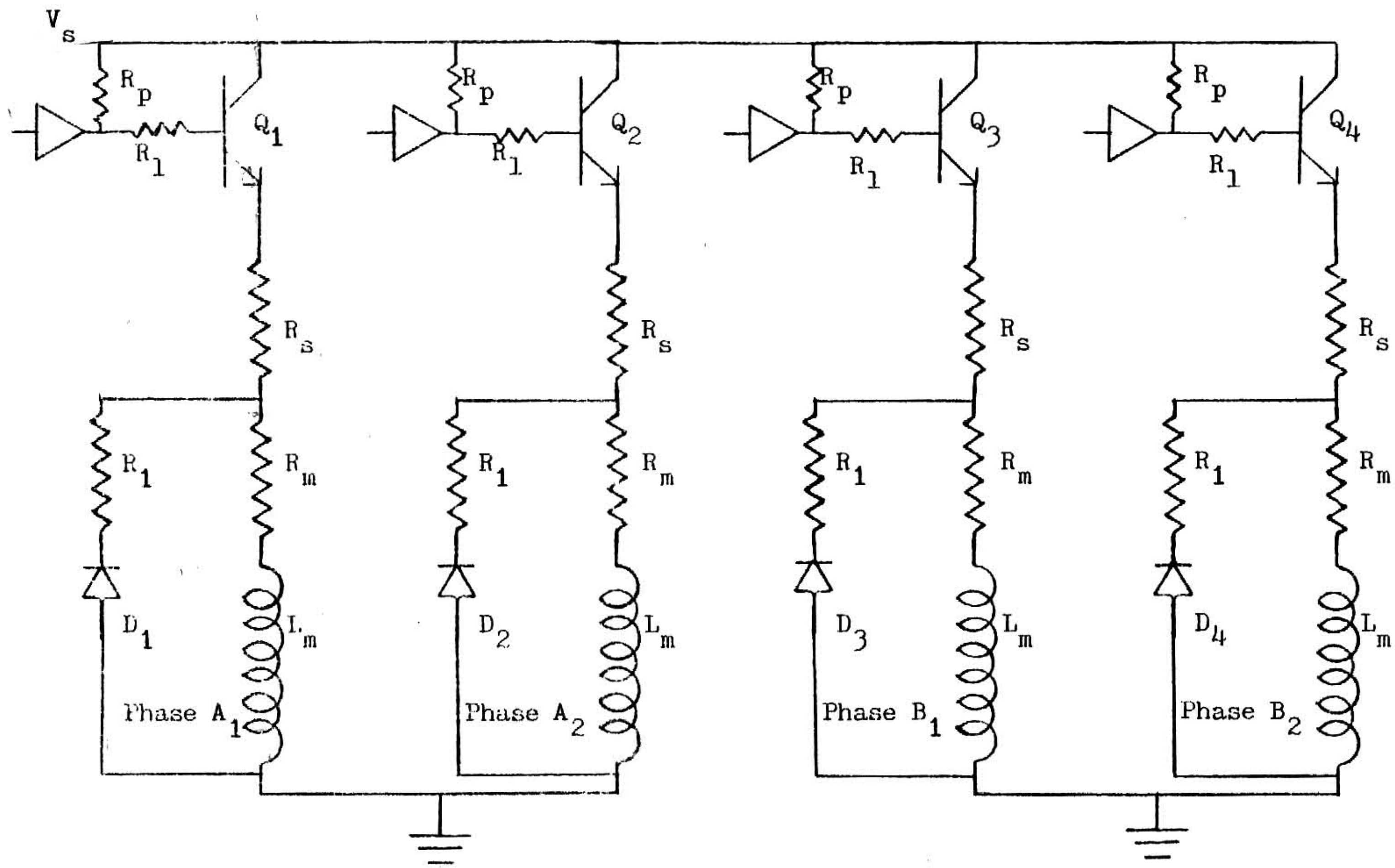


Fig. 3.2 Resistive step motor drive

resistor. Diodes D_j and resistors R_j $(1 < j < 4)$ allow the phase current i_j to free-wheel when the particular phase is de-energized.

Another problem encountered with the resistive drive is the fact that as the value of the external resistor is increased, the voltage supply rating must also be increased to maintain the flow of rated current in the windings. This implies that the V_{ceo} rating requirement of the transistors, used as switches to control the motor phases, must also increase.

The control bits from the KIM-1 microcomputer were connected to the base inputs (via buffers) of the respective transistor switches via which they controlled the motor phase switching. The buffers used were TTL 7407's, noninverting, open collector buffers. These were connected in the manner shown in Fig. 3.2. The values of R_p (pull up resistor) and R_1 (current limiting resistor) were selected so as to ensure a V_{be} for each transistor of approximately 0.7 - 0.75 V. One can consider phase A as a typical example:

$$V_{E1} = i_R(R_m + R_s)$$

$$i_B = i_R/h_{FE}$$

$$V_s = V_{E1} + 0.75 + i_B(R_p + R_1)$$

Typical values for R_1 and R_p were as follows:

R_p is approximately 1 K

R_1 is approximately (500-600) ohms

In order to limit excessive I^2R losses in the external resistor the supply voltage was raised to a maximum value of 30 volts allowing a value of 20 ohms for R_s, this allowed the motor to be operated at a maximum speed of 550 steps per second, as compared to 400-450 steps per second with no external resistors in series with each phase.

3.3 CHOPPER DRIVE

The chopper drive is a high efficiency step motor drive. Fig. 3.3 shows the circuit diagram of the system implemented. Each motor phase, instead of having a resistor in series with it, has an extra transistor(QC_i) in series with it. For each phase, transistor Q_i controls the phase switching and these transistors are driven by the control bits from the microprocessor. Transistors QC_j (j= 1,2,3,4) control the chopping action for their respective phase windings when the particular phase is turned on by its Q_i transistor switch.

To study the action of the chopping transistors, one can take a particular phase A_1 for example, since all four phase drives are identical. The chopping action is controlled by a means of current feedback. Each phase has a current sensing resistor (R_E = 2.2 ohms) in series with it and the voltage across this resistor is fed to a Schmitt trigger circuit (ST_i block in Fig. 3.3) via a differential amplifier stage (DA_i in Fig. 3.3). When the phase is turned on transistor Q_1, C_1 is in the on state, as the current in the motor phase increases and crosses an upper threshold value (approximately 0.66 amp.) the Schmitt trigger is turned off by the voltage across the sensing resistor so the phase current starts to decrease as it freewheels through diode D_1. When the current drops to a lower threshold value (approximately 0.54 amp.) the Schmitt trigger turns transistor on again. This chopping action continues for as long as phase A_1 is on.

Fig. 3.4 shows a detailed circuit diagram of the differential amplifier stage. It has two stages, a buffer stage to ensure that the current through the sensing resistor is not disturbed and an amplifier stage. Both stages utilize operational amplifiers. The output voltage (V_o) is given by the equation

$$V_o = n\,(V_1 - V_2)\ .$$

A block diagram of the Schmitt trigger is shown in Fig. 3.5. As long as the input voltage (V_i) is less than V_1, the output of the Schmitt trigger stays high. When V_i exceeds V_1, the output goes to zero and stays at zero volts until

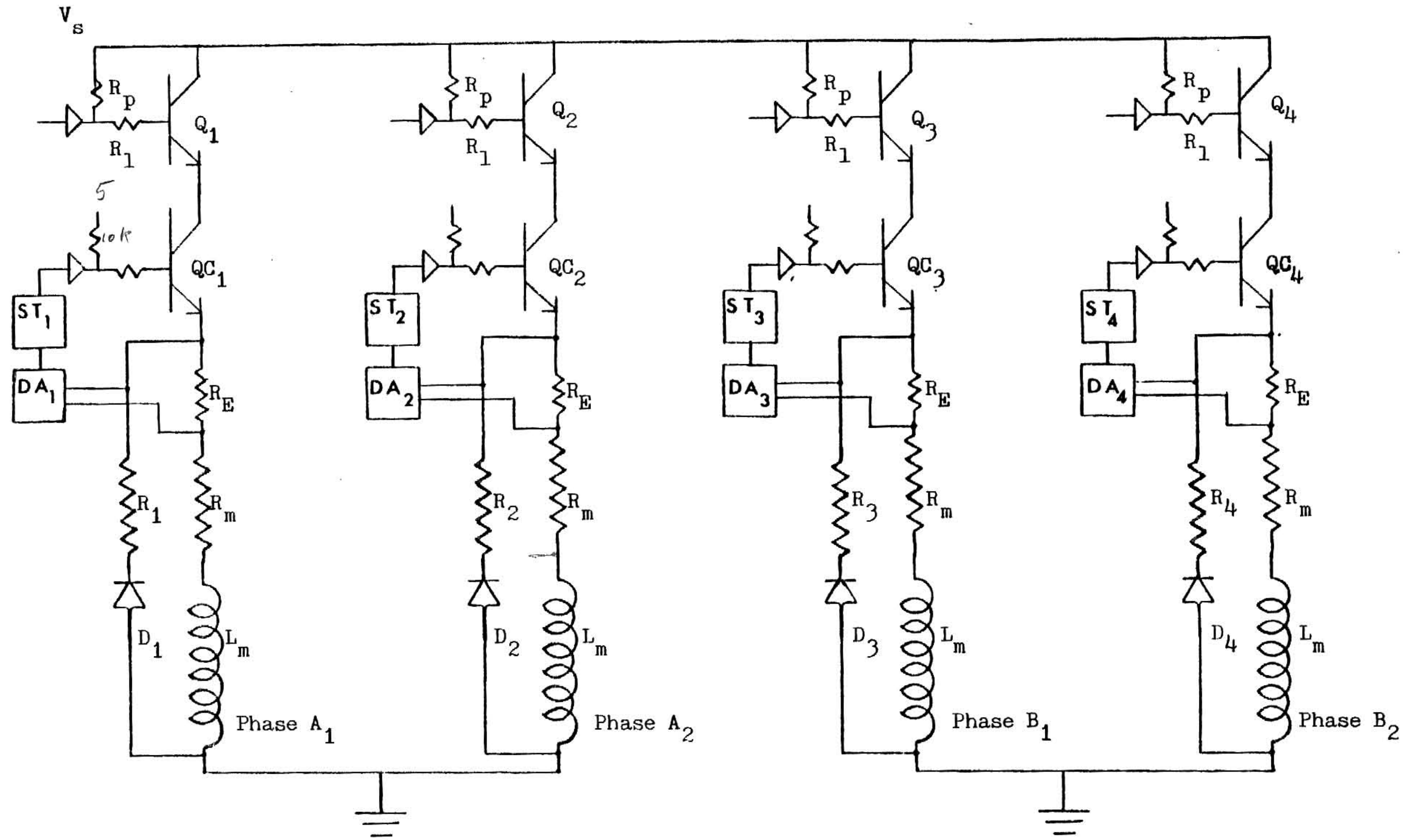


Fig. 3.3 Chopper step motor drive

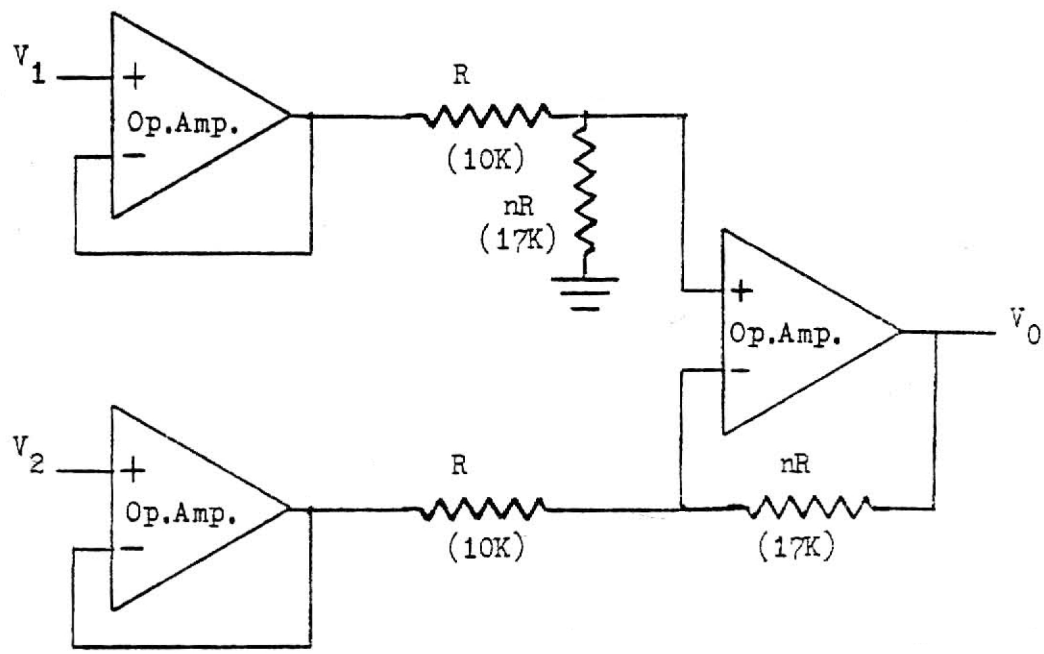


Fig. 3.4 Differerential Amplifier Stage (DA_1)

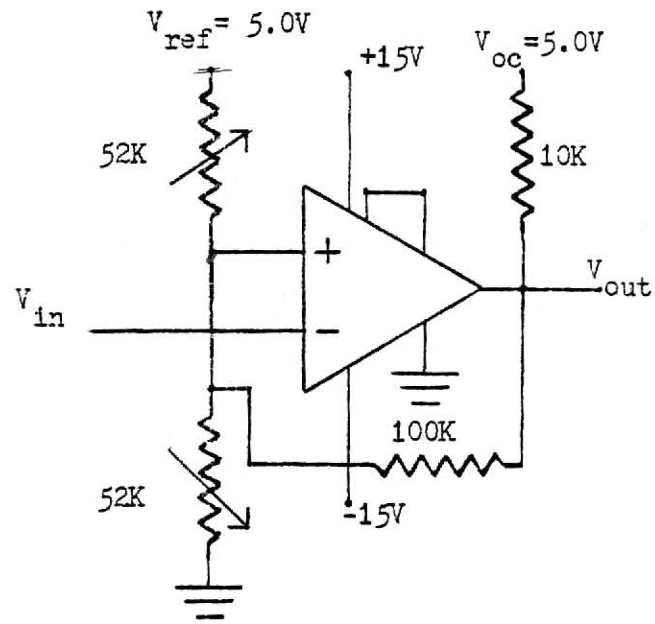


Fig.3.5 Schmitt trigger cicuit (ST_i) with LM311 comparator

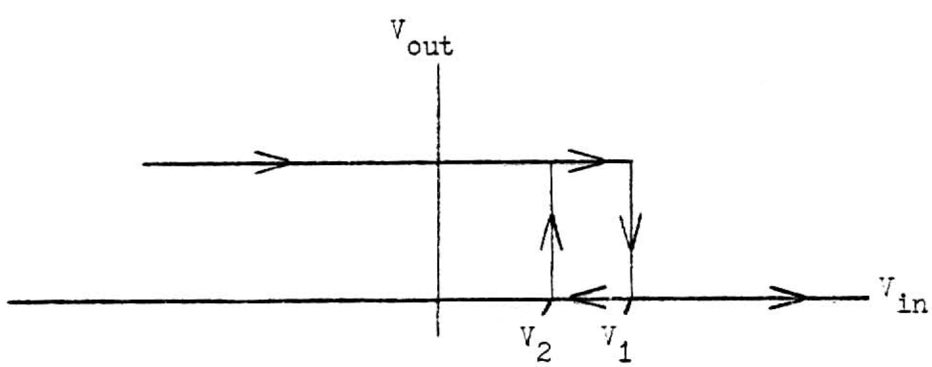


Fig.3.6 Schmitt trigger transfer characteristic

V_1 falls to a value of V_2. The equations for V_1 and V_2 can be expressed as:

$$V_1 = \frac{R_2R_3}{R_2R_3 + R_1(R_2 + R_3)} \cdot V_{ref} + \frac{R_1R_2}{R_1R_2 + R_3(R_1 + R_2)} \cdot V_{out} .$$

$$V_2 = \frac{R_2R_3}{R_2R_3 + R_1(R_2 + R_3)} \cdot V_{ref} .$$

R_1 and R_2 are variable resistors, hence the trigger points V_1 and V_2 can be varied. This implies that the upper and lower phase current threshold values can be set to a value, in a desired range, by varying the values of R_1 and R_2. The transfer characteristic of the Schmitt trigger is shown in Fig. 3.6.

Fig. 3.7 shows the circuit diagram for the matched LED photo-transistor pair used in the feedback loop. The output of the photo-transistor was fed to the counter clock input via a Schmitt-triggered buffer (TTL 7414). The Schmitt-triggered gate changed the output of the photo-transistor from a slow-changing waveform to a fast-changing one with well-defined positive and negative edges. The counter is reset by pin 6 of output port A and its output bits are connected to pins (0 - 3) of port A (the lsb to pin 0 and the msb to pin 3).

Fig. 3.8 shows the circuit diagran for the LED used to indicate the end of deramping. A circuit diagram for the 'start deramping' switch is shown in Fig. 3.9, the switch is debounced before being connected to bit 4 of port A.

Details of the circuit components have been included in the hardware appendix.

Normally, for a chopper drive, the supply voltage used is of the order of ten times the motor's rated supply voltage since this would imply that the current would reach its upper threshold value faster. However,for comparison purposes the supply voltage was restricted to a maximum of 30 V as in the resistive drive . In spite of this limitation on the maximum supply voltage, the motor operated satisfactorily at a maximum speed of 650 - 700 steps per second, about

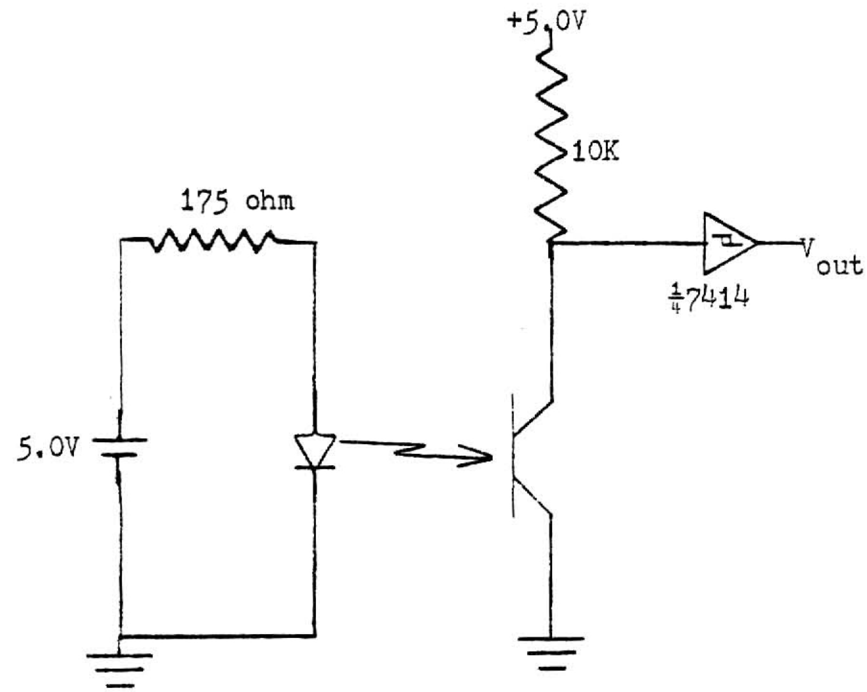


Fig.3.7 MCT-8 matched LED photo-transistor pair

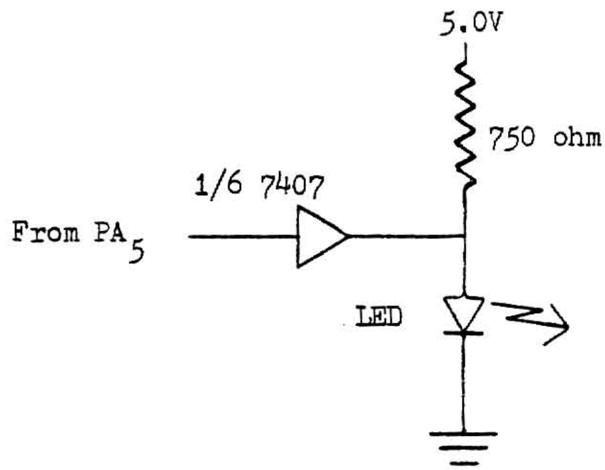


Fig. 3.8 LED circuit to indicate end of deramping

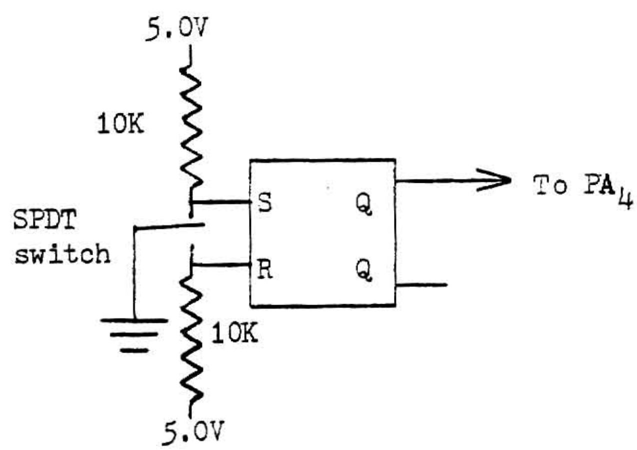


Fig. 3.9 Debouncing circuit for deramping switch

100 steps per second more than the maximum speed attained in the case of the resistive drive.

The chopper drive is a high efficiency step motor drive since it enables high stepping speeds without the high I^2R losses associated with the resistive drive. However, since it maintains a higher level of current in the motor phase windings, it also causes the motor to heat up rapidly thus necessitating some means of aiding the step motor to dissipate heat more rapidly. This problem though is associated with all high speed step motor drives.

CHAPTER 4

CONTROLLER SOFTWARE

4.0 OVERVIEW

For any microprocessor based control system, it is the system software which offers the greatest amount of flexibility in system modification and adaptability as compared to a hardwired control system. The software developed for controlling the step motor can be grouped under three headings:

(A) Position Control Program

(B) Ramping/De-ramping Table Generating Program

(C) Stepping Control Program.

All the programs were written in the 6502 assembly language, and a detailed listing of each, including "User Instructions", has been included in the software appendix.

Initially a couple of test programs were written which served the dual purpose of testing out the system hardware and comparing two alternative software schemes for driving the step motor.

4.1 TEST PROGRAM

The two software schemes for driving the step motor were implemented via programs "TLU" and "ROW". In both cases, the four motor phase windings were each controlled by one output bit thus implying a four-bit motor control word (MCW). This was necessary since, in the one-phase-on and mixed modes of operation (Fig. 1.4 and 1.6), none of the motor phase states are complementary as is the case in the two-phase-on mode of operation (Fig. 1.5).

Program TLU (Flowchart in Fig. 4.1) uses the Table Look-up Method. The excitation tables (Fig. 1.4-1.6) were stored in memory; each word in the table occupied a byte of memory and each table was stored in the sequence shown in Fig. 1.4 through Fig. 1.6 as well as in a reversed sequence to offer directionality (clockwise/counterclockwise) controls. The resulting look-up table is shown

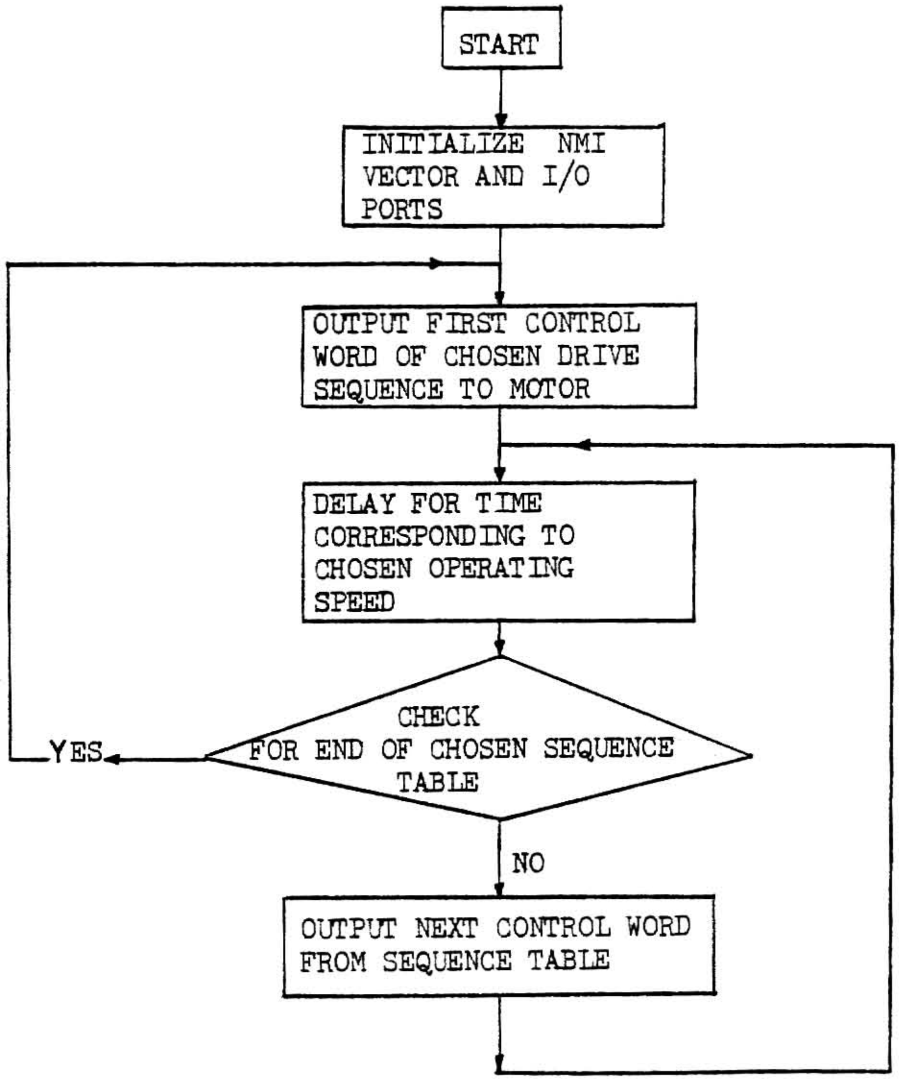


Fig. 4.1 Flowchart for program 'TLU'

in Fig. 4.2. To drive the motor the program needs a specified delay time (in milliseconds) and sequence code. The delay time is the reciprocal of the desired speed (steps per second) while the sequence code is the starting memory address of the chosen excitation table.

For example, if it were desired to step the motor in a clockwise direction in the two-phase-on mode of operation at a stepping rate of 200 steps per second, the corresponding delay time and sequence code would be as follows

$$\text{Delay time} = \frac{1000}{200} \text{ milliseconds} = 5 \text{ ms.}$$

$$\text{Sequence code (from Fig. 4.2)} = (00)_{16}$$

With this information the program outputs a control word from the chosen table to the MCW, delays for the specified time and then outputs the next control word in the table It keeps stepping through the table sequentially (Fig. 4.3) thereby stepping the motor in the desired manner.

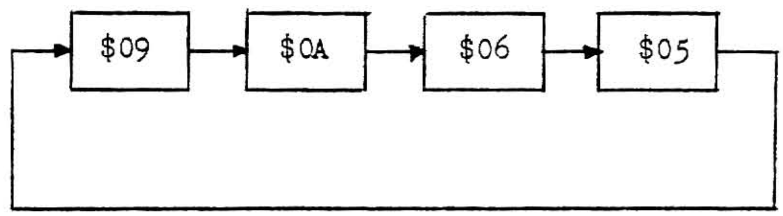
Control of the direction of rotation could have been carried out in the software program itself by traversing the same excitation table in a particular direction for clockwise rotation and in a reversed direction for counterclockwise rotation, but this merely increased the complexity of the program and did not save any significant amount of memory space. Hence the option of storing the same excitation table for a particular mode of operation (one-phase-on, two-phase-on, mixed) in reversed sequence was used to control the choice of direction of rotation. This drive technique has also been used in the positioning control and stepping control programs.

Program "ROW" (Flowchart in Fig. 4.4) uses the Rotating Control Word Method to achieve the same result as program TLU. Instead of using look-up tables this technique consists of rotating the MCW one bit, to the right or to the left depending on the direction of rotation required, after delaying for a specified time (depending on the required motor speed). Since the MCW is a 4-bit word (nibble) and the KIM microcomputer handles byte-sized words, a pseudo-nibble rotate instruction was implemented as follows. An 8-bit memory location was loaded

Starting Address(Hex)	Memory Contents(Hex)	
\$0000	\$09	
	\$0A	Two phase on clockwise sequence
	\$06	
	\$05	
\$0004	\$08	
	\$02	One phase on clockwise sequence
	\$04	
	\$01	
\$0008	\$09	
	\$08	Mixed mode clockwise sequence
	\$0A	
	\$02	
	\$06	
	\$04	
	\$05	
	\$01	
\$0010	\$05	
	\$06	Two phase on counter-clockwise sequence
	\$0A	
	\$09	
\$0014	\$01	
	\$04	One phase on counter-clockwise sequence
	\$02	
	\$08	
\$0018	\$01	
	\$05	Mixed mode counter-clockwise sequence
	\$04	
	\$06	
	\$02	
	\$0A	
	\$08	
	\$09	

Note : The two least significant hexadecimal digits of the starting address of each seqn. is the seqn. code.

Fig. 4.2 Memory map for motor drive sequence tables



Sequence code = $00 (from fig.4.2)

Two phase on, clockwise drive sequence

Fig. 4.3 Program example for 'TLU'

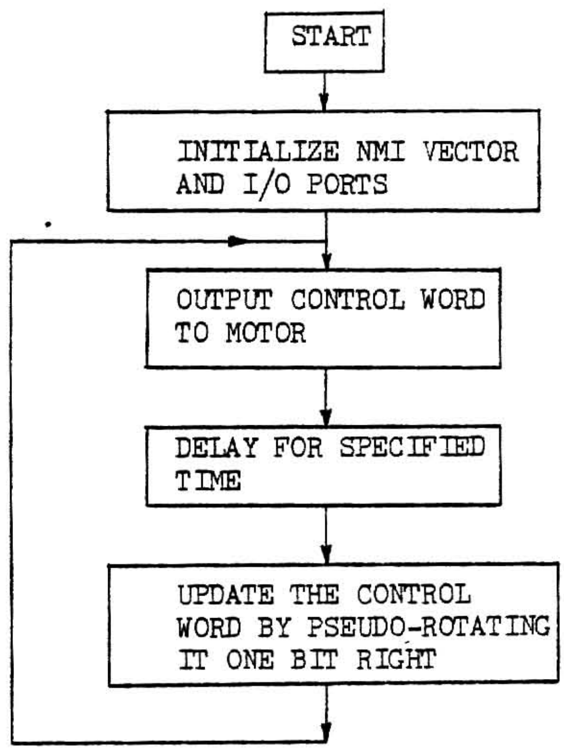


Fig. 4.4 Flowchart for program 'ROW'

with the control word (different for the one-phase-on mode of excitation and the two-phase-on mode of excitation), its four least significant bits comprised the MCW while the four most significant bits merely duplicated the same bit pattern (Fig. 4.5), next, before a rotate instruction is performed the carry is set to equal the least significant bit of the 8-bit control word or the most significant bit, depending on whether a rotate right or rotate left instruction is to be performed, respectively. Fig. 4.5 illustrates an example of this procedure with a control word for the two-phase-on mode of excitation.

The drive method used in program ROW saves the trouble of storing look-up tables, but it does so at the cost of increased program complexity. The program would have to have different segments for clockwise and counterclockwise rotation as a result of the pseudo-nibble rotate instruction and also, to incorporate the mixed mode of operation, the program would have to alternately output and update the two-phase-on mode control word as well as the one-phase-on mode control word. Incorporating these features in the program would offset any memory requirement savings as a result of eliminating the look-up tables.

The same method of generating time delays via software was used in both programs, ROW and TLU. The KIM-1 system has on-board timers which can also be used to generate time delays. Once they have been initialized to the delay count, they generate an interrupt when the delay is over.

4.1 POSITION CONTROL PROGRAM

Many applications of the step motor require it to act as a position control device. In profile cutting, for example, step motors can be used to control the X and Y coordinates of the cutting tool. In machine tools step motors can be used to position the drill or drill bit; in assembly line robots these motors could be used to position the robot's arms. With the growing sophistication and availability of digital controllers, there has been a corresponding increase in the application of a step motor as a digital position control actuator. Being an inherently discrete motor device with noncumulative position error, it is possible

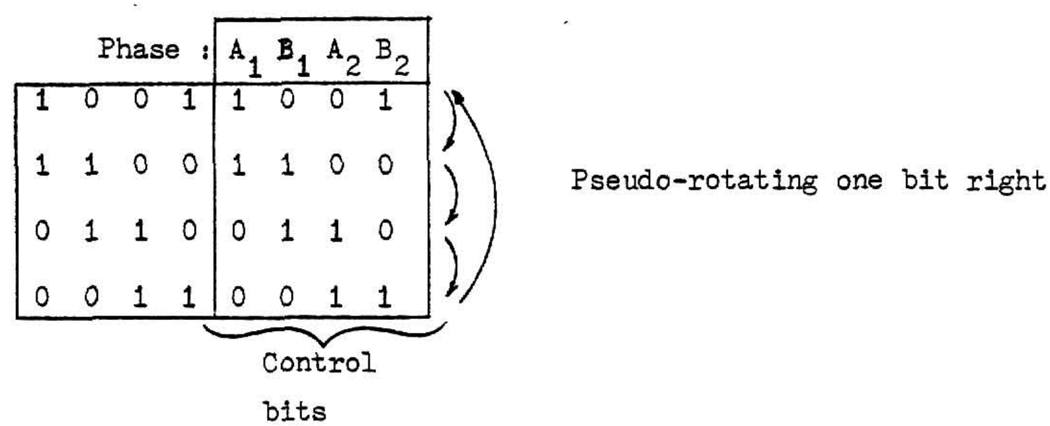


Fig. 4.5 Program example for 'ROW'

to achieve accurate position control with a step motor in an open-loop system, eliminating many instability problems and transducers such as tachometers and the need for gear trains.

The position control program (POSCON) controls the step motor in the positioning mode of operation. A flowchart of the program is shown in Fig. 4.6. This routine causes the motor to oscillate. Each oscillation consists of two phases. Since the table look-up method of driving the motor is used, the user can specify the number of steps (less than 255) to be moved, the speed, and the particular sequence table to be used for each phase of the oscillation and also the total number of oscillations(less than 255) the motor must complete before coming to a halt.

The parameters to be specified for each phase of the oscillation can be chosen independently for each phase, this implies that the phases could differ in direction of movement, number of steps, and choice of sequence table. This offers a considerable amount of flexibility in controlling the step motor. However, to ensure that no loss of steps occurs in switching back and forth between the phases, the sequences chosen must each be subsets of the other. This is so because the program, while switching from the old phase of the oscillation to the new phase, searches the new phase sequence table for the last control word output by the old phase sequence table and once a match is found, it outputs the next control word in the new phase sequence. This ensures stepping continuity. For a greater degree of flexibility, programs in which each oscillation consists of more than two phases could be developed, using essentially the same techniques as POSCON.

Program usage details and a listing have been included in the software appendix.

4.3 RAMP GENERATOR PROGRAM

The acceleration of the step motor from its start-stop speed to its operating speed is known as ramping and the deceleration process is referred to

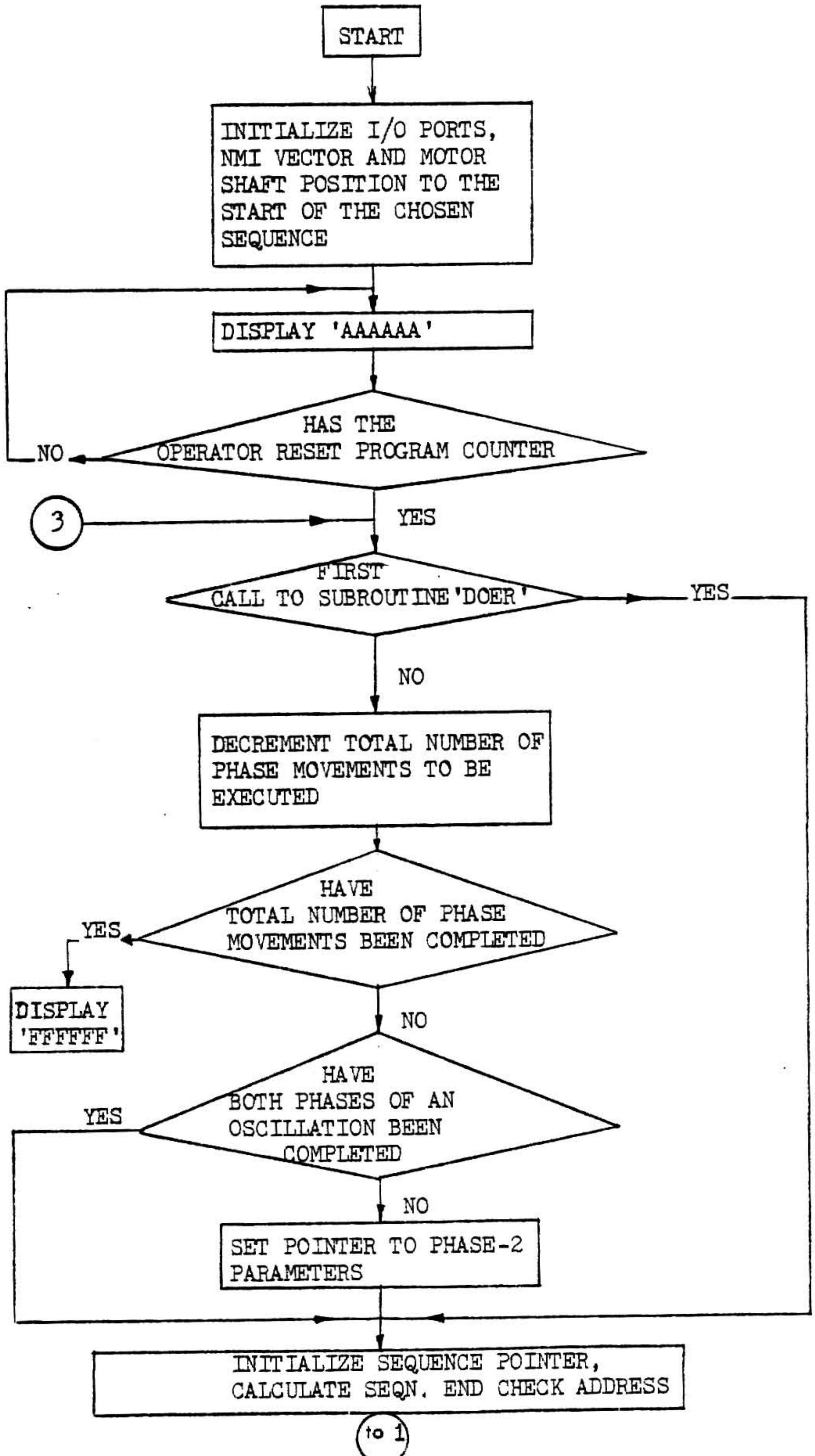
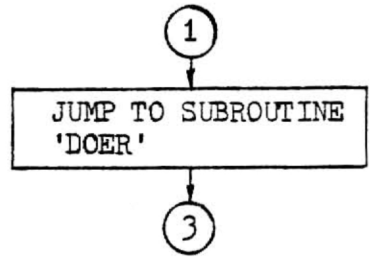


Fig.4.6 Flowchart for program 'POSCON'



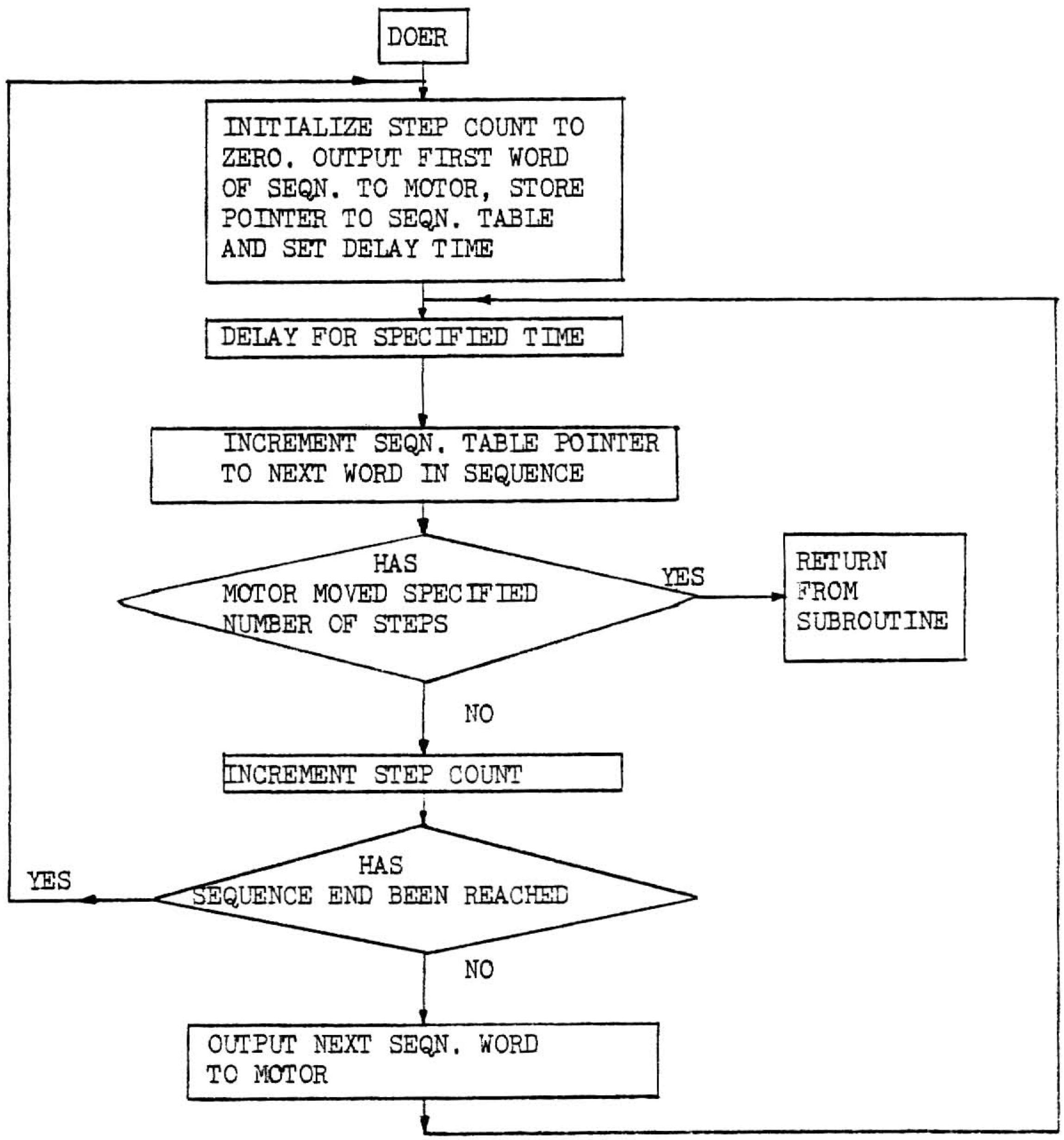


Fig.4.6 (contd.) Subroutine 'DOER'

as de-ramping. Generally, this task is carried out by analog circuits; an example of such a circuit has been presented in a paper by V. V. Athani and J. G. Mundhada (2). Such hardware circuits usually permit a particular ramping pattern. The speed can be varied, for example, in a linear, parabolic or exponential manner, but each pattern would require special hardware.

Program STEPCON which controls the motor in the slewing mode, incorporates software ramping/deramping via a ramping/deramping look-up table. Hence, the same program and drive hardware can be used to implement virtually any ramping/deramping pattern by merely modifying the look-up table. This leads to the idea of ramping/deramping look-up table generating programs. RAMGEN is an example of such a program. It generates a linear ramping/deramping look-up table.

RAMGEN, in essence, solves the linear equation

$$v = u + at \quad . \qquad (4.1)$$

over a specified time (t) range. Given the motor's starting (initial) velocity (u), the final velocity (v_f) and the time in which it has to attain its final velocity (t_f), the program calculates the acceleration constant (a) as

$$a = \frac{v_f - u}{t_f} \quad . \qquad (4.2)$$

It then divides the final time (t_f) into a specified number of intervals, the minimum interval resolution being one-quarter (1/4) of a second. The next task that it handles is the calculation of the motor velocity at the boundaries of each interval. For example, if the time specified (t_f) were ten seconds and the specified time interval was one second, the program would evaluate the motor speed at time 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, and 10 seconds (Fig. 4.8). Once this procedure is completed the program computes the delay times (Td) corresponding to the calculated velocities. Each delay time is calculated to the nearest tenth of a millisecond. Finally, corresponding to each velocity, the program calculates the number of steps the rotor would move in the specified time interval.

The ramping/deramping table consists of stored values of delay time (one byte) followed by the corresponding number of rotor steps for which this

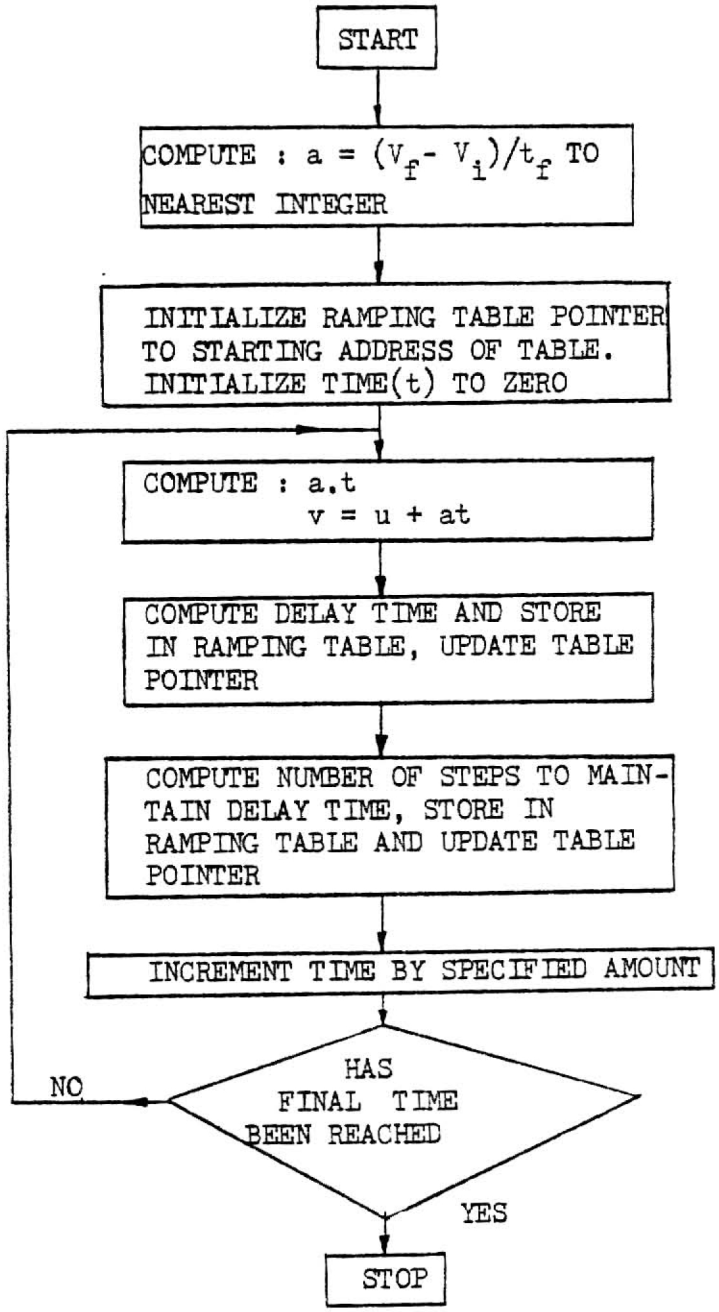


Fig. 4.7 Flowchart for program 'RAMGEN'

delay time value is to be used. The number of steps is stored in a two-byte location, most significant byte followed by the least significant byte (Fig. 4.9). The acceleration constant is stored in a 8-bit location. Of the values which must be supplied to the program to the program, t_f is an 8-bit word, while the initial and final motor speed values are each 16-bit words. A flowchart for the program is shown in Fig. 4.7.

Program RAMGEN generates a ramping table, which starts the motor at its start-stop speed and accelerates it, linearly, up to its operating speed; de-ramping is accomplished by merely traversing the table in reverse order. This implies that the acceleration constant (a) computed is always positive, as are the delay time values and the number of steps for maintaining each calculated delay time. Hence the program calls subroutines MPLY and DIVN which perform 8 bit by 8 bit unsigned division. RAMGEN also uses a form of "pseudo-floating point" arithmetic to permit delay time calculating with an accuracy of one tenth of a millisecond, and a time interval resolution of 1/4 of a second.

To permit the one-quarter of a second time interval resolution the program requires the user to specify the time interval (SPIN) as a multiple of 1/4 of a second and also initialize a constant (FRAC) to any non-zero value, the program then uses the time value specified and the divides all calculations in which the value is used by four (rotate right twice) resulting in a fractional time interval specification. If the constant is initialized to zero the program assumes the time interval specification to be an integer.

For the accuracy required in delay time specification, a process of repeated division is employed to yield a result which has a unit specification of one-tenth of a second rounded off to the nearest integer. The rounding off is accomplished by comparing the remainder of the division process with the divisor and if it turns out to be equal to or greater than half the divisor, the quotient is incremented by one.

Time (Sec.)	Speed (Hex.) (sps)
0	00C8
1	00F0
2	0118
3	0140
4	0168
5	0190
6	01B8
7	01E0
8	0208
9	0230
10	0258

Initial speed : 200 sps

Final speed : 600 sps

Acceleration time : 10 sec.

Time interbal : 1 sec.

Fig. 4.8 Program example for 'RAMGEN'

Table entry number	Delay time (0.1 ms,Hex.)	Number of steps to maintain delay time(Hex.)
1	32	00C8
2	2A	00F0
3	24	0118
4	20	0140
5	1B	0168
6	19	0190
7	17	01B8
8	15	01E0
9	14	0208
10	11	0230
11	11	0258
	1-byte	2-bytes

Fig. 4.9 Table output produced by RAMGEN (corresponding to fig.4.8)

Program RAMGEN has been developed as a "sample" program to give some idea of the versatility of software controlled open-loop ramping of a step motor. More sophisticated programs can be developed to generate parabolic or exponential ramping curves. These programs would essentially be equation-solving routines which would solve parabolic or exponential equations with enhanced arithmetic capability, generating their respective ramping curves. Hence, it is evident that this technique, of open-loop ramping of a step motor, facilitates virtually any form of ramping curve which can be expressed as an equation.

Details on program usage have been included in the software appendix along with a listing of the program.

4.4 STEPPING CONTROL PROGRAM

The step motor, besides being used in the single stepping (position control) mode is also used in the continuous stepping (slewing) mode. To operate the step motor in this mode, the most important control task is to accelerate (ramp) the step motor from its start-stop speed to its operating speed, for a given load torque, and then again decelerating (deramping) the motor from its operating speed to its start-stop speed before bringing it to a complete halt. This is necessary to ensure that the motor does not lose stepping synchronism with the input voltage pulses.

Program STEPCON (flowchart shown in Fig. 4.10) is designed to ramp the step motor from its start-stop speed up to its operating speed and, when it is time to halt the motor, to deramp the motor from its operating speed back to its start-stop speed before halting it. It is essentially an open-loop scheme, but it does use a feedback loop, via an encoder disc and a 4-bit binary counter to ensure that the motor is not stalling. It does this by reading in the counter after a specified number of steps and then comparing the count, obtained via the counter, with the specified number of steps. If the two do not match to within a

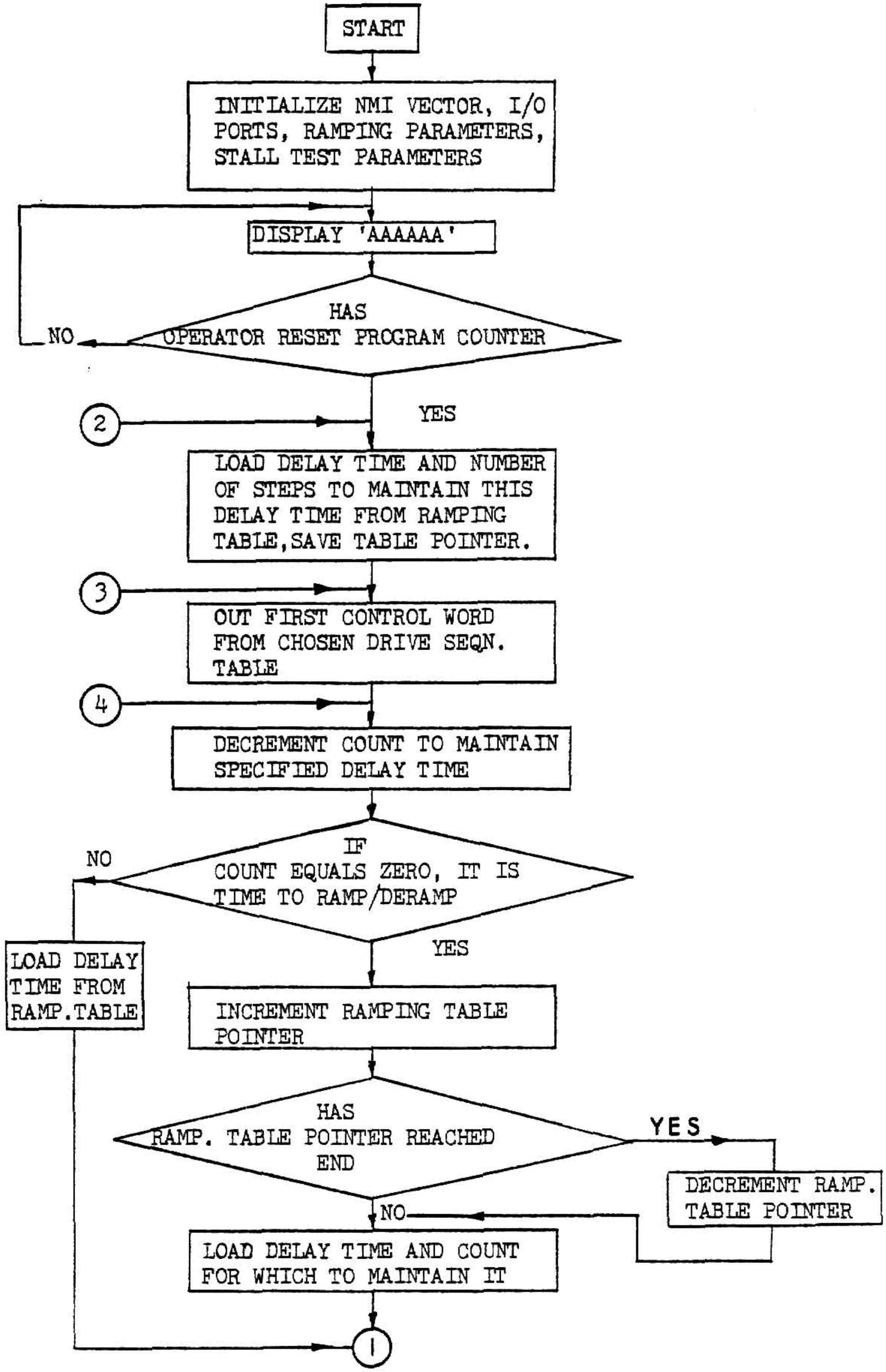


Fig. 4.10 Flowchart for program 'STEPCON'

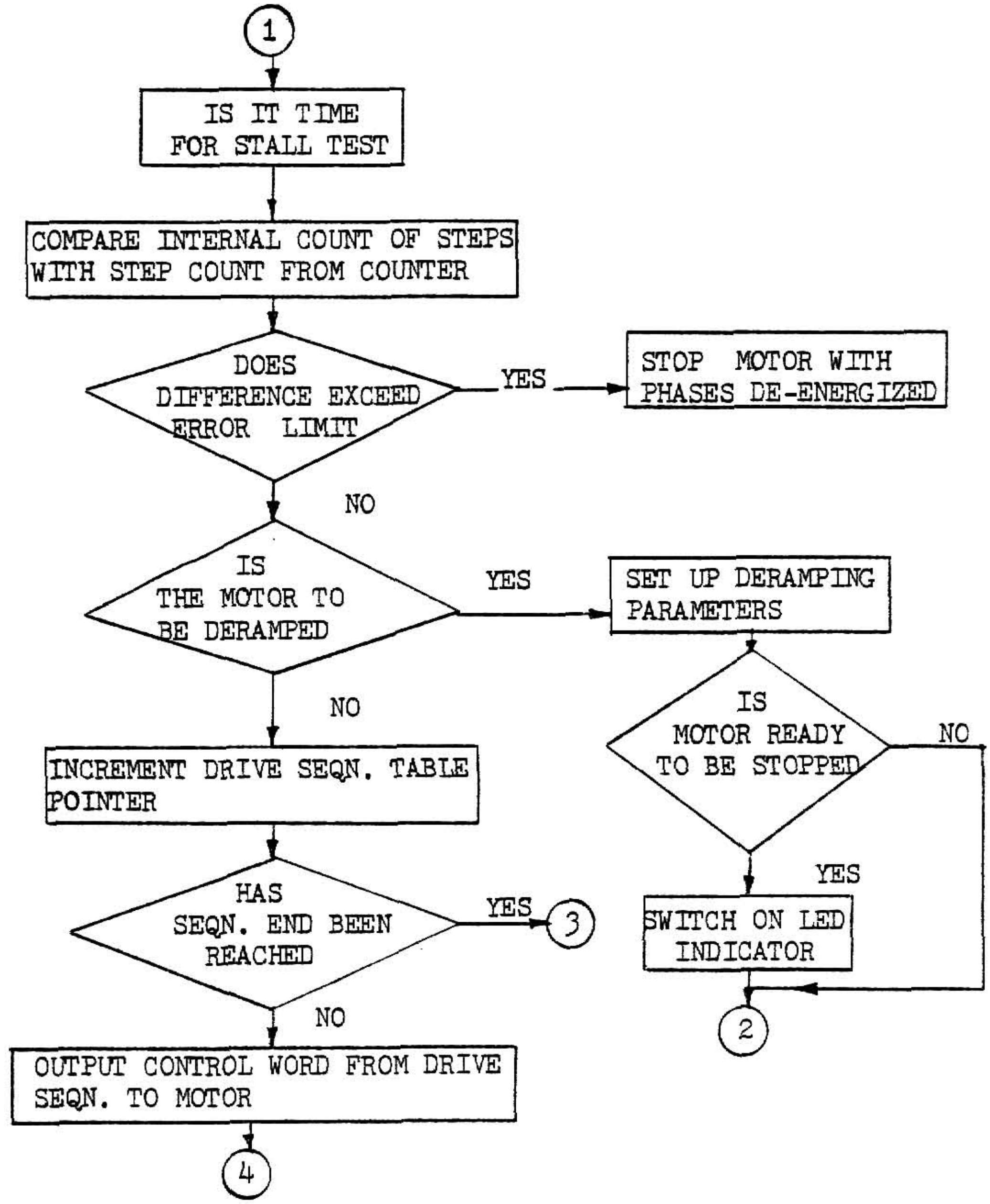


Fig. 4.10 (contd.)

specified error limit, the program halts the motor with its phases in the de-energized state. The maximum specified number of steps to perform the stall test is restricted to fifteen since a 4-bit counter is used.

STEPCON uses the ramping table, generated by program RAMGEN, to ramp and deramp the motor. STEPCON accesses three bytes of the table at a time. The first byte contains the delay time corresponding to the required motor speed while the next two bytes contain a 16-bit count of the number of steps for which the particular speed is to be maintained. Once this count reaches zero the program accesses the next three bytes from the table. The same table is used to ramp and deramp the step motor; while ramping, successive accesses to the table speed up the motor and while deramping, the sequence of accessing the table is anti-parallel to that of ramping the motor.

Program STEPCON uses the table look-up method of driving the step motor thus giving a choice of direction control, step size control (1.8° for the two-phase-on mode and for the one-phase-on mode, 0.9° for the mixed mode of operation), and torque output level since the output torque, at a given speed, is highest for the two-phase-on mode of operation, lowest for the one-phase-on mode of operation and in-between for the mixed mode of operation.

In a study done by W.D. Page, G. Singh and B.C. Kuo (4) on open-loop acceleration of step motors, it was observed that the most commonly used acceleration schemes, in which the speed is increased linearly or exponentially with time, do not necessarily yield the best result. The results were obtained for a particular step motor and for a given load, but they still indicate that the best acceleration scheme for a particular application could well depend on the application itself. In the case of the motor used by Page, et al., both the linear and exponential acceleration schemes produced an overshoot and oscillation in final speed (while ramping). The parabolic scheme, however, yielded a nearly constant final speed.

Implementing different ramping schemes via hardware would greatly increase the system complexity, especially if a choice of ramping schemes were required of the same control unit. Implementation via software, however, would simply require different ramping table generating programs (of which RAMGEN is an example) and the same stepping control program (STEPCON). Hence the software implementation would merely require additional memory space to implement a variety of acceleration schemes.

In view of the results presented by Page, et al.,namely, the dependence of the step motor acceleration scheme on factors such as the choice of step motor and loading conditions it would be advantageous to have a controller with the ability to implement different acceleration schemes with a minimum of difficulty in switching from one scheme to another. A software control scheme, of the type described in this section, would seem to satisfy the above requirement.

RESULTS AND CONCLUSIONS

In the course of this study two different kinds of step motor drives were implemented and tested, within limitations. The first of these drives the resistive drive increased the maximum stepping rate of the motor from 450 sps (with no external modifications to the motor phases) to about 550 sps. However, it entailed high energy loss (I^2R) in the external series resistance connected to each motor phase. The chopper drive allowed a maximum stepping rate in the range of (650-700) sps, this was in spite of the fact that the supply voltage was limited to 30 V (the maximum value used in the resistive drive). Normally the supply voltage used in a chopper drive is as high as ten times the motor's rated supply voltage (12 V for the motor used).

In the position control mode, the motor exhibited good start-stop capability, with a loss of accuracy limited to ± 2 steps. The linear ramping/de-ramping table used to accelerate/decelerate the motor had its delay time entries calculated to an accuracy of one-tenth of a millisecond which provided for smooth motor acceleration. In the continuous mode of operation the feedback loop also performed satisfactorily in switching off the power to the motor phases whenever the motor tended to stall.

The software developed for the microprocessor based controller demonstrated the flexibility of software control. The controller could be used in any positioning or continuous stepping application of the step motor and would offer different motor acceleration schemes without requiring hardware modifications. The idea of a microprocessor based controller seems ever more attractive in view of the growing availability of low cost single chip microprocessors which are virtual mini-computers.

APPENDIX A: STEP-MOTORS

Step-motor Terminology:

(1) Step angle: This is the angle through which the motor shaft moves for a change in excitation pattern.

(2) Accuracy: Step-motor accuracy is the per step deviation from the rated step angle. This error is non-cumulative, that is, even after n-steps, if synchronization between the input pulses and rotor steps has been maintained by design constraints, the total error is still equal to a percentage (about 5%) of a single step angle.

(3) Holding torque: Torque required to deflect the rotor a full step with the motor energized and in a standstill condition.

(4) Detent torque: Torque exerted on the rotor in the un-excited state.

(5) Ramping: The term ramping implies the acceleration/deceleration of the motor from/to a safe start-stop speed to/from its operating speed (Fig. A.1).

(6) Torque vs. Stepping rate curve: The motor, for a given load torque T_1 (Fig. A.2) can be started at a rate R_1 given by its response (start-stop) curve (R), and can attain a maximum operating speed R_2 given by the slewing (operating) curve (S).

(7) Pull in torque: Maximum torque against which the motor can start at a given speed. In Fig. A.2, this equals T_1 for a speed of R_1.

(8) Pull out torque: This is the maximum operating torque the motor can supply for a given speed, and it equals T_0 for a speed of R_1 (Fig. A.2).

(9) Pull in rate: The maximum start-stop speed for a given load torque, equals R_1 for T_1 in Fig. A.2.

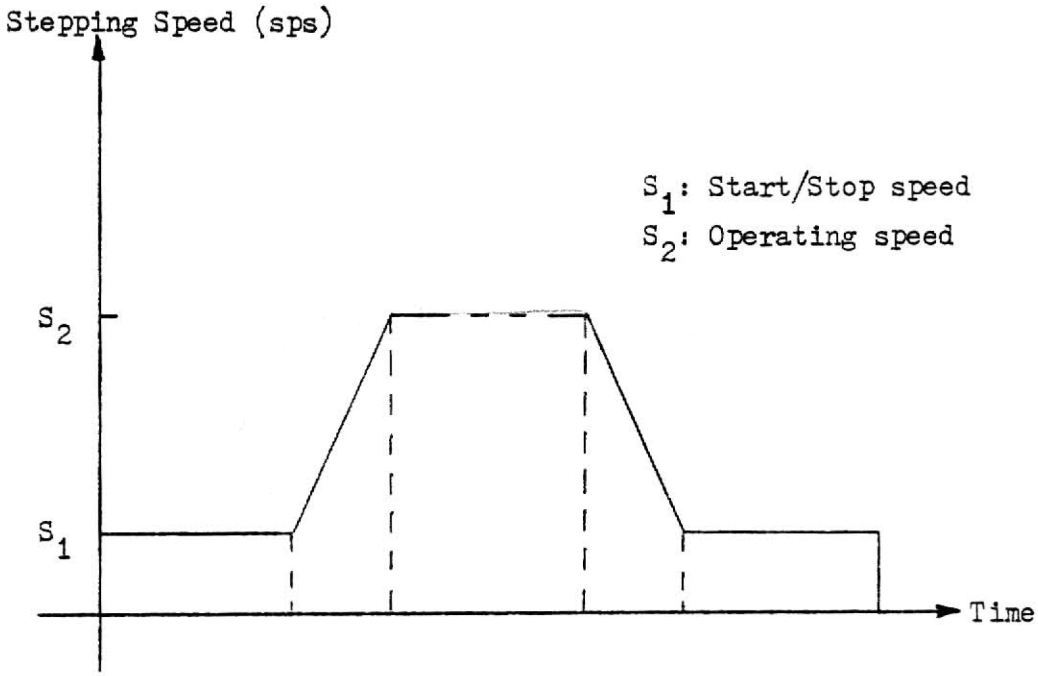


Fig. A.1 Linear Ramping/Deramping graph

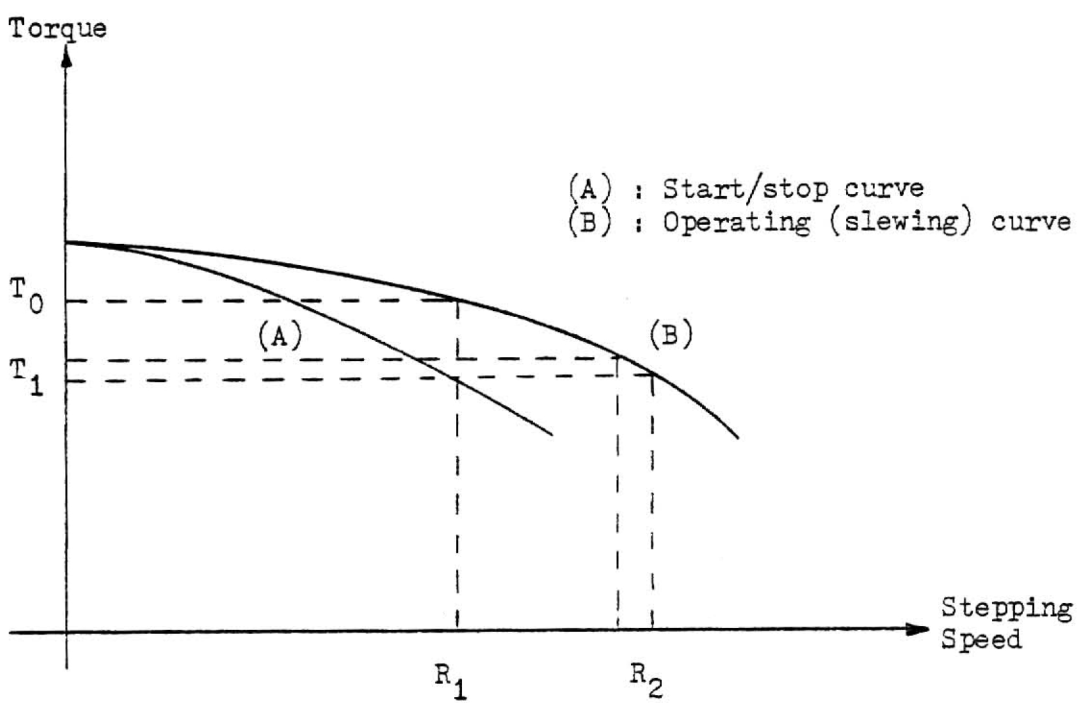


Fig. A.2 Torque vs. speed characteristic

(10) Pull out rate : Maximum operating speed for a given load torque, equals

R_2 for T_1 in Fig. A2.

Summary of Permanent magnet and Variable reluctance motor characteristics :

Variable reluctance motor :

(a) High stepping rates possible.

(b) High torque to inertia ratio.

(c) Fast step response.

(d) Ability to free wheel.

(e) Bi-directional rotation, requires at least three phases for directional control.

(f) No detent torque.

(g) Oscillatory step response with overshoots.

Permanent magnet motor :

(a) High torque to inertia ratio.

(b) Bi-directional rotation possible.

(c) Detent torque, implies no free-wheeling.

(d) Low power consumption.

(e) Less oscillatory than a variable reluctance motor.

Motor selection & System design criteria :

(a) Estimate the output torque required of the motor as the sum of acceleration torque, load torque, and frictional torque.

(b) From the positional accuracy required, determine the step angle.

(c) From the response and slewing curves, using the torque value obtained in (a) above, determine the allowable start-stop and operating speeds.

(d) From the minimum allowable time to reach the operating speed from the start-stop speed, determine the angular acceleration required.

APPENDIX B : HARDWARE SPECIFICATIONS

The step motor used in the project was a model manufactured Eastern Air Devices. Its construction details were as follows :

a) Four, bifilar wound, phase windings.

b) Two rotor pole pieces with 50 teeth each and out of phase by one-half of a tooth pitch.

c) Permanent magnet fitted along the lengthof the rotor axis.

d) 48 teeth on the stator, 6 teeth per pole piece.

e) 1.8^{o} basic angle.

f) For a step motor, the relationship between the pole pitch (P_p) and the tooth pitch (P_t) is given by

$$P_p = (m \pm 1/n)P_t$$

where m is a positive integer chosen from packaging considerations, and n equals the number of steps used for one tooth pitch rotation.

For the given motor, n=4 and m=6

$$\text{therefore } P_t = 360^{o}/50 = 7.2^{o}$$

$$P_p = 45^{o}$$

Motor specifications

EAD : LA23GCK-20

Rated voltage = 12V (d.c.)

Rated current = 0.6 amps (per phase)

Resistance = 20 ohms (per phase)

Inductance = 18 mh (per phase)

Step accuracy = 3 %

Holding torque = 30 oz.in.

Rotor inertia = 57 $gm.cm^2$.

Weight = 15 oz.

Power Transistor ratings :

GE-273 : medium power NPN silicon transistors.

Power dissipation = 7W @ 25^{o}C

I_C = 1.5 amps.

BVCBO =50 V

BVCEO = 40 V

BVEBO = 5 V

Freq. = 70 MHz

H_{fe} = 300 (max.)

Figures D.1a-c show the pin outs for the comparator (LM311) and operational amplifiers (MC1747 & TL084C) used.

Figures D.2a-c show the pin outs for the counter (CMOS 4516), the buffers (TTL 7407) and D-Flip-Flops used in the circuit.

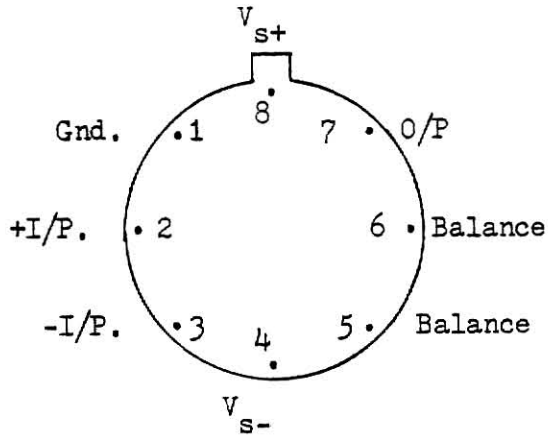


Fig. D.1a Pin out for LM311 comparator(Top view)

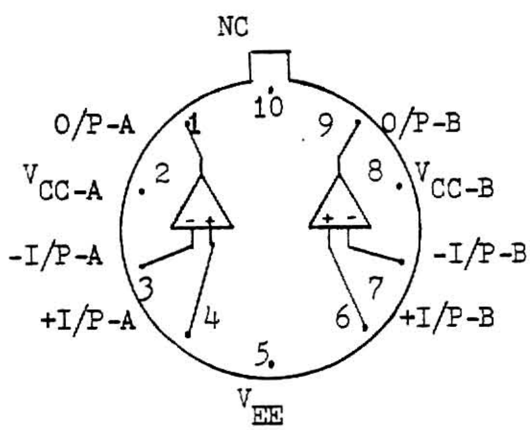


Fig. D.1b Pin out for 747 dual op. amp. (Top view)

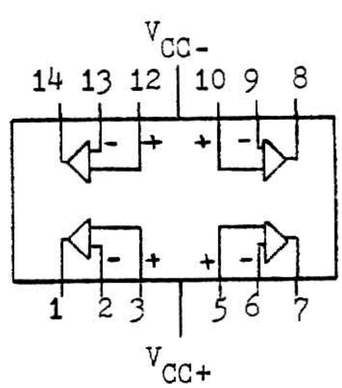


Fig. D.1c Pin out for TL084C quad. op. amp.

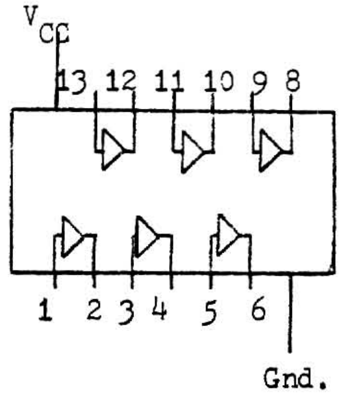


Fig.D.2a Hex. buffers (7407)

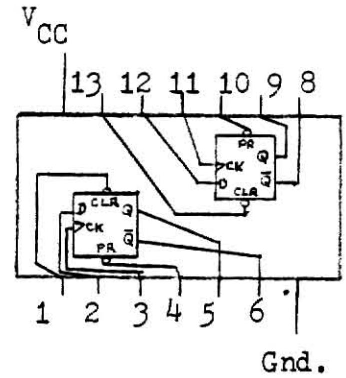


Fig.D.2b Dual D-F/F (7474)

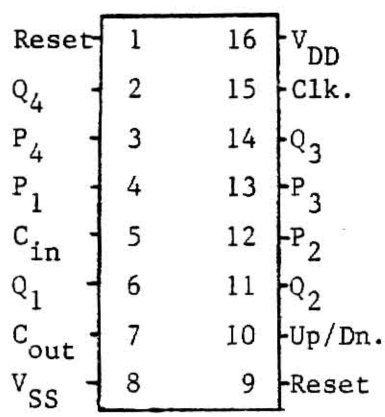


Fig.D.2c Pin designation for CD4516 4-bit CMOS counter

APPENDIX C SOFTWARE LISTINGS

The system software has been developed for the KIM-1 system described in the hardware appendix. The programs were stored on magnetic tape and entered via the teletype. The keypad was used to run the programs once they had been loaded into the KIM-1 system memory. All parameters, required as user inputs by any one of the programs, are entered in hexadecimal ($), and all numerical outputs from the programs are specified in hexadecimal too.

TEST PROGRAMS:

(a) TLU : Program TLU implements the table look-up method of driving the step motor. It requires, as inputs, a delay time value which corresponds to the motor speed required and an excitation table. To run the program, the above inputs must be specified and the 'GO' key, on the keypad, depressed. The motor can be halted by depressing the 'ST' key.

(b) ROW : This program drives the step motor by rotating the control word. To run the program, the delay time and chosen control word must be specified. The motor can then be started by depressing the 'GO' key and halted by depressing the 'ST' key.

```
LOC        CODE          CARD  10          20          30          40          50
0000                     ;****************************************************
0000                     ;MOTOR DRIVE PROGRAM (TLU)
0000                     ;****************************************************
0000                     ;VARIABLE STORAGE ADDRESSES
0000                     ;****************************************************
0000                             ;DRIVE TABLE STORED IN LOCN. $00-$03
0000                             TEMP=$0004
0000                             DELTI=$0005   ;DELAY TIME (MS)
0000                     ;****************************************************
0000                     ;OPERATIONAL
0000                     ;****************************************************
0000                             *=$0006
0006   A9 00                     LDA #$00      ;INIT. NMI VECTOR
0008   8D FA 17                  STA $17FA
000B   A9 1C                     LDA #$1C
000D   8D FB 17                  STA $17FB
0010   A9 7F                     LDA #$7F      ;INIT. PB (0-6) AS O/P BITS
0012   8D 03 17                  STA $1703
0015   A2 00             BLOOP   LDX #$00
0017   B5 00             SLOOP   LDA $00,X     ;O/P CONTROL WORD TO MOTOR
0019   8D 02 17                  STA $1702
001C   86 04                     STX TEMP
001E   A4 05                     LDY DELTI     ;DELAY LOOP
0020   A2 C7             DELAY   LDX #$C7
0022   CA                DLY1    DEX
0023   D0 FD                     BNE DLY1
0025   88                        DEY
0026   D0 F8                     BNE DELAY
0028   A6 04                     LDX TEMP      ;UPDATE TABLE PNTR.
002A   E8                        INX
002B   E0 04                     CPX #$04
002D   F0 E6                     BEQ BLOOP
002F   4C 17 00                  JMP SLOOP
0032                             .END
```

```
CARD # LOC      CODE          CARD  10          20          30          40          50
       0000                   ;************************************************
       0000                   ;MOTOR DRIVE PROGRAM (RCCCN)
       0000                   ;************************************************
       0000                   ;VARIABLE STORAGE ADDRESSES
       0000                   ;************************************************
       0000                          SECO=$0000    ;SECN. CODE
       0000                          DELT=$0001    ;DELAY TIME (MS)
       0000                   ;************************************************
       0000                   ;OPERATIONAL
       0000                   ;************************************************
       0000                          *=$0005
       0005  A9 00     START  LDA #$00      ;INIT. NMI VECTOR
       0007  8D FA 17         STA $17FA
       000A  A9 1C            LDA #$1C
       000C  8D FB 17         STA $17FB
       000F  A9 7F            LDA #$7F      ;INIT.PB (0-6) AS O/P BITS
       0011  8D 03 17         STA $1703
       0014  A5 00            LDA SECO      ;O/P MOTOR DRIVE WORD
       0016  85 02            STA $02
       0018  8D 02 17  LOOP   STA $1702
       001B  A4 01            LDY DELT
       001D  A2 C7     DELAY  LDX #$C7      ;DELAY LOOP
       001F  CA        DLY1   DEX
       0020  D0 FD            BNE DLY1
       0022  88               DEY
       0023  D0 F8            BNE DELAY
       0025  A9 01            LDA #$01      ;SET CARRY TO L.S.B. OF MOTOR
       0027  24 02            BIT $02       ;DRIVE WORD
       0029  F0 04            BEQ NEXT1
       002B  38               SEC
       002C  4C 30 00         JMP OUT
       002F  18        NEXT1  CLC
       0030  66 02     OUT    ROR $02       ;UPDATE MOTOR DRIVE WORD
       0032  A5 02            LDA $02
       0034  4C 18 00         JMP LOOP
       0037                   .END
```

POSITION CONTROL PROGRAM:

Before running program POSCON, the following initializations must be carried out:

(i) Specify number of steps to be moved by the motor in phase one of its oscillation.

(ii) Specify number of steps to be moved by the motor in phase two of its oscillation.

(iii) Specify stepping speed (corresponding delay time in ms.) for the first phase of the oscillation.

(iv) Specify stepping speed (corresponding delay time in ms.) for the second phase of the oscillation.

(v) Specify the chosen excitation table code for phase one.

(vi) Specify the chosen excitation table code for phase two.

(vii) Specify the total number of phase movements required.

Once the above parameters have been initialized, the program execution is started at address $3E by initializing the address display to this value and depressing the 'GO' key. The program initializes the rotor position and displays 'AAAAAA' on the keypad display, at which point program execution is started at address $5D by depressing the 'ST' key, initializing the address display to $005D and depressing the 'GO' key. When the positioning job is over, the display reads 'FFFFFF'.

```
CARD # LOC      CODE          CARD  10          20          30          40          50
       0000                   ;**************************************************
       0000                   ;POSITION CONTROL PROGRAM (PCSCON)
       0000                   ;**************************************************
       0000                   ;VARIABLE STORAGE ADDRESSES
       0000                   ;**************************************************
       0000                          NOST=$002A   ;NO. OF STEPS MOVED IN PHASE UF OS
       0000                          RATE=$002C   ;STEPPING RATE FOR PHASE U OF OSC.
       0000                          SECO=$002E   ;SECN. CODE FOR PHASE U OF OSC.
       0000                          ENCH=$0030   ;END CHECK ADD. FOR CHOSEN SEQN.
       0000                          TIME=$0031   ;NO. OF REPTITIONS OF OSC.
       0000                          INTI=$0032   ;TEMP. STORAGE FOR INCO
       0000                          INCO=$0033   ;NO. OF PHASES IN ONE OSC (MAX.=2)
       0000                          STEC=$0034   ;MOTOR STEP COUNT
       0000                          TEM1=$0035   ;TEMP. STORAGE LOCN.
       0000                          TEM2=$0036   ;TEMP. STORAGE LOCN.
       0000                          TEM3=$0037   ;TEMP. STORAGE L
       0000                          TEM4=$0038   ;TEMP. STORAGE LOCN.
       0000                   ;**************************************************
       0000                   ;INITIALIZATIONS
       0000                   ;**************************************************
       0000                          *=$0040
       0040   A9 7F                  LDA #$7F     ;INIT. PB (0-6) AS O/P
       0042   8D 03 17               STA $1703
       0045   A2 00                  LDX #$00     ;O/P CONTROL WORD TO MOTOR
       0047   A1 2E                  LDA (SECO,X) ;TO INIT. ROTOR POSITION
       0049   8D 02 17               STA $1702
       004C   A9 AA                  LDA #$AA     ;DISPLAY 'AAAAAA'
       004E   85 F9                  STA $F9
       0050   85 FA                  STA $FA
       0052   85 FB                  STA $FB
       0054   20 1F 1F      IS1      JSR $1F1F
       0057   4C 54 00               JMP IS1
       005A                   ;OPERATIONAL
       005A                   ;**************************************************
       005A   A2 00         MAST     LDX #$00     ;INIT. INTI TO VALUE OF INCO
       005C   A5 33                  LDA INCO
       005E   85 32                  STA INTI
       0060   B5 2E         MS1      LDA SECO,X   ;CALC. SEQN. END CHECK ADD.
       0062   C9 08                  CMP #$08
       0064   F0 0C                  BEQ MS2
       0066   C9 18                  CMP #$18
       0068   F0 08                  BEQ MS2
       006A   18                     CLC
       006B   69 04                  ADC #$04
       006D   85 30                  STA ENCH
       006F   4C 77 00               JMP MS3
       0072   18            MS2      CLC
       0073   69 08                  ADC #$08
       0075   85 30                  STA ENCH
       0077   86 35         MS3      STX TEM1     ;STORE PHASE NO. PNTR.
       0079   20 98 00               JSR DOER
       007C   A6 35                  LDX TEM1
       007E   C6 31                  DEC TIME     ;CHECK FOR END OF JOB
       0080   F0 08                  BEQ MS4
       0082   C6 32                  DEC INTI     ;CHECK FOR END OF OSC.
```

```
CARD # LOC     CODE         CARD  10          20          30          40          50
       0084   F0 D4                  BEQ MAST
       0086   E8                     INX
       0087   4C 60 C0               JMP MS1
       008A   A9 FF        MS4       LDA #$FF        ;DISPLAY 'FFFFFF'
       008C   85 F9                  STA $F9
       008E   85 FA                  STA $FA
       0090   85 FB                  STA $FB
       0092   20 1F 1F     MS5       JSR $1F1F
       0095   4C 92 C0               JMP MS5
       0098                ;**************************************************
       0098                ;SUB. : DOER
       0098                ;**************************************************
       0098   A9 00        DOER      LDA #$00        ;INIT. STEC TO ZERO
       009A   85 34                  STA STEC
       009C   85 37                  STA TEM3
       009E   A4 35        DS1       LDY TEM1
       00A0   B6 2E                  LDX SECO,Y
       00A2   86 36                  STX TEM2
       00A4   A5 37                  LDA TEM3        ;CHECK TO MAINTAIN STEP SYNCH.
       00A6   D0 13                  BNE DS2         ;BETWEEN PHASES
       00A8   E6 37                  INC TEM3
       00AA   AD 02 17     MAT       LDA $1702
       00AD   29 0F                  AND #$0F
       00AF   85 38                  STA TEM4
       00B1   B5 00                  LDA $00,X
       00B3   C5 38                  CMP TEM4
       00B5   F0 04                  BEQ DS2
       00B7   E8                     INX
       00B8   4C AA C0               JMP MAT
       00BB   B5 00        DS2       LDA $00,X       ;O/P CONTROL WORD TO MOTOR
       00BD   8D 02 17               STA $1702
       00C0   A6 35                  LDX TEM1
       00C2   B4 2C                  LDY RATE,X
       00C4   A2 C7        DS3       LDX #$C7        ;DELAY LOOP
       00C6   CA           DS4       DEX
       00C7   D0 FD                  BNE DS4
       00C9   88                     DEY
       00CA   D0 F8                  BNE DS3
       00CC   E6 36                  INC TEM2        ;UPDATE SEQN. TABLE PNTR.
       00CE   A6 35                  LDX TEM1
       00D0   B5 2A                  LDA NOST,X
       00D2   C5 34                  CMP STEC
       00D4   F0 0B                  BEQ DS5
       00D6   E6 34                  INC STEC
       00D8   A6 36                  LDX TEM2
       00DA   E4 30                  CPX ENCH
       00DC   F0 C0                  BEQ DS1
       00DE   4C BB C0               JMP DS2
       00E1   60           DS5       RTS
       00E2                ;**************************************************
       00E2                ; DRIVE SEQUENCE TABLES
       00E2                ;**************************************************
       00E2                          *=$0000
       0000   09                     .BYTE $09,$0A,$06,$05,$08,$02,$04,$01,$05
       0001   0A
```

```
CARD # LOC     CODE        CARD  10          20          30          40          50
  109  0002  06
  109  0003  05
  109  0004  08
  109  0005  02
  109  0006  04
  109  0007  01
  109  0008  09
  110  0009  08                 .BYTE $08,$0A,$02,$06,$04,$05,$01
  110  000A  0A
  110  000B  02
  110  000C  06
  110  000D  04
  110  000E  05
  110  000F  01
  111  0010  05                 .BYTE $05,$06,$0A,$09,$01,$04,$02,$08,$01
  111  0011  06
  111  0012  0A
  111  0013  09
  111  0014  01
  111  0015  04
  111  0016  02
  111  0017  08
  111  0018  01
  112  0019  05                 .BYTE $05,$04,$06,$02,$0A,$08,$09
  112  001A  04
  112  001B  06
  112  001C  02
  112  001D  0A
  112  001E  08
  112  001F  09
  113  0020                     .END
```

RAMP TABLE GENERATION PROGRAM:

Program RAMGEN needs the following parameter initializations to be carried out before it can generate the ramp table:

(i) Final (operational) velocity required (16 bits).

(ii) Initial (start-up) velocity required (16 bits).

(iii) Total acceleration time (8 bits).

(iv) Incremental time for changing the motor speed (8 bits).

(v) Fractional (0.25 sec.) incremental time required (FRAC = 1) or integer incremental time required (FRAC = 0).

Once the initializations have been carried out, program execution is started at address $200 and when execution ends, the display reads 'FFFFFF', which implies that the ramp table has been stored, starting at location $30.

```
LOC     CODE          CARD  10        20        30        40        50
0000                  ;**************************************************
0000                  ;PROGRAM:RAMGEN.
0000                  ;       THIS PROGRAM COMPUTES THE RAMP/DERAMP
0000                  ;TABLE, FOR THE LINEAR CASE AND STORES IT START-
0000                  ;ING AT ADDRESS $0030.
0000                  ;**************************************************
0000                  ;VARIABLE STORAGE ADDRESSES.
0000                  ;**************************************************
0000                        FHI=$0060    ;M.S.B. OF FINAL VELOCITY
0000                        FLO=$0061    ;L.S.B. OF FINAL VELOCITY
0000                        IHI=$0062    ;M.S.B. OF INITIAL VELOCITY
0000                        ILO=$0063    ;L.S.B. OF INITIAL VELOCITY
0000                        FINT=$0064   ;ACCN. TIME IN SEC.
0000                        TEMP1=$0065  ;TEMP. RAMP./DERAMP TABLE PNTR.
0000                        AAY=$0066    ;ACCN. CONSTANT
0000                        LDIV=$0067   ;L.S.B. OF DIVIDEND
0000                        MDIV=$0068   ;M.S.B. OF DIVIDEND
0000                        DSCR=$0069   ;DIVISOR
0000                        QUO=$006A    ;QUOTIENT
0000                        REM=$006B    ;REMAINDER
0000                        MAD=$006C    ;MULTIPLICAND
0000                        MER=$006D    ;MULTIPLIER
0000                        MMPL=$006E   ;M.S.B. OF MULT. RESULT
0000                        LMPL=$006F   ;L.S.B. OF MULT. RESULT
0000                        SPIN=$0070   ;INCREMENTAL TIME
0000                        FRAC=$0071   ;EQUALS ONE IF SPIN IS FRACTIONAL
0000                        TIME=$0072   ;STORES TOTAL TIME
0000                        TEMP2=$0073  ;TEMP. REMAINDER
0000                        COUNT=$0074  ;TEMP. STORAGE FOR SPIN
0000                        TEMP3=$0075  ;TEMP. STORAGE FOR X-REG.
0000                  ;**************************************************
0000                  ;OPERATIONAL
0000                  ;**************************************************
0000                        *=$0200
0200    38            RAG   SEC          ;CALCULATE ACCN. CONST. TO
0201    A5 61               LDA FLO      ;NEAREST INTEGER
0203    E5 63               SBC ILO
0205    85 67               STA LDIV
0207    A5 60               LDA FHI
0209    E5 62               SBC IHI
020B    85 68               STA MDIV
020D    A5 64               LDA FINT
020F    85 69               STA DSCR
0211    20 27 03            JSR DIVN
0214    A5 64               LDA FINT
0216                        LSRA
0216    C5 6B               CMP REM
0218    10 06               BPL RG1
021A    E6 6A               INC QUO
021C    A5 6A               LDA QUO
021E    85 66               STA AAY
0220    A9 30         RG1   LDA #$30     ;TEMP1 POINTS TO START OF
0222    85 65               STA TEMP1    ;RAMP/DERAMP TABLE
0224    A9 00               LDA #$00     ;INIT. TIME LOCN. TO ZERO
0226    85 72               STA TIME
```

```
LOC    CODE       CARD 10          20       30         40        50
0228   A5 72      RAGO   LDA TIME      ;CALCULATE (ACCN.CONST.)(TIME)
022A   85 6C             STA MAD
022C   A5 66             LDA AAY
022E   85 6D             STA MER
0230   20 44 C3          JSR MPLY
0233   A5 71             LDA FRAC      ;CHECK & ADJUST FOR FRAC. TIME
0235   F0 17             BEQ RG2
0237   46 6E             LSR MMPL
0239   66 6F             ROR LMPL
023B   46 6E             LSR MMPL
023D   66 6F             ROR LMPL
023F   90 0D             BCC RG2
0241   A5 6F             LDA LMPL
0243   18                CLC
0244   69 01             ADC #$01
0246   85 6F             STA LMPL      ;ROUND OFF RESULT
0248   A5 6E             LDA MMPL
024A   69 00             ADC #$00
024C   85 6E             STA MMPL
024E   A5 63      RG2    LDA ILO       ;CALC. VEL. AT END OF INC. TIME
0250   18                CLC           ;ROUNDED OFF TO NEAREST INTEGER
0251   65 6F             ADC LMPL
0253   85 63             STA ILO
0255   85 67             STA LDIV
0257   A5 62             LDA IHI
0259   65 6E             ADC MMPL
025B   85 62             STA IHI
025D   85 68             STA MDIV
025F   A9 0A             LDA #$0A
0261   85 69             STA DSOR
0263   20 27 C3          JSR DIVN
0266   A5 6B             LDA REM
0268   C9 05             CMP #$05
026A   30 02             BMI RG3
026C   E6 6A             INC CUC
026E   A9 64      RG3    LDA #$64      ;CALC. DELAY TIME TO NEAREST
0270   85 67             STA LDIV      ;TENTH OF A MILLISECOND
0272   A9 00             LDA #$00
0274   85 68             STA MDIV
0276   A5 6A             LDA CUC
0278   85 69             STA DSOR
027A   20 27 C3          JSR DIVN
027D   A9 0A             LDA #$0A
027F   85 6C             STA MAD
0281   A5 6A             LDA CUC
0283   85 6D             STA MER
0285   20 44 C3          JSR MPLY
0288   A5 6B             LDA REM
028A   85 73             STA TEMP2
028C   A9 00             LDA #$00
028E   85 68             STA MDIV
0290   A5 69             LDA DSOR
0292   85 67             STA LDIV
0294   A9 0A             LDA #$0A
0296   85 69             STA DSOR
```

```
LCC      CODE         CARC  10          2C           30           40           50
0298   2C 27 C3           JSR DIVN
0298   A5 68              LDA REM
C29D   C9 05              CMP #$05
C29F   30 02              BMI RG4
02A1   E6 6A              INC QUO
C2A3   A9 00        RG4   LDA #$00
02A5   85 68              STA MDIV
02A7   A5 73              LDA TEMP2
02A9   85 67              STA LDIV
02AB   A5 6A              LDA QUO
02AD   85 69              STA DSOR
02AF   20 27 03           JSR DIVN
02B2   A5 69              LDA DSOR
02B4   4A                 LSR A
C2B5   C5 68              CMP REM
02B7   1C 02              BPL RG6
02B9   E6 6A        RG5   INC QUO
02BB   FC FC        RG6   BEC RG5
02BC   A5 6A              LDA QUO
02BF   18                 CLC
02C0   A6 65              LDX TEMP1     ;STORE DELAY TIME IN TABLE
02C2   95 00              STA $00,X
02C4   A5 7C              LDA SPIN
02C6   85 74              STA COUNT     ;CALC. # OF STEPS TC MAINTAIN
C2C8   A9 00              LDA #$00      ;DELAY TIME
02CA   85 60              STA FHI
02CC   85 61              STA FLO
02CE   18           RG7   CLC
02CF   A5 63              LDA ILO
02D1   65 61              ADC FLO
02D3   85 61              STA FLO
02D5   A5 62              LDA IHI
02D7   65 60              ADC FHI
02D9   85 60              STA FHI
02DB   C6 74              DEC COUNT
02DC   DC EF              BNE RG7
02DF   A5 71              LDA FRAC
C2E1   FC 08              BEC RG8
02E3   46 6C              LSR FHI
C2E5   66 61              ROR FLO
02E7   46 60              LSR FHI
02E9   66 61              ROR FLO
C2EB   A5 60        RG8   LDA FHI
02ED   95 01              STA $01,X
C2EF   A5 61              LDA FLO
C2F1   95 02              STA $02,X
02F3   E8                 INX
C2F4   E8                 INX
02F5   E8                 INX
02F6   18                 CLC
02F7   A5 72              LDA TIME
02F9   65 70              ADC SPIN
02FB   85 72              STA TIME
02FD   A5 71              LDA FRAC
02FF   FC 09              BEC RG9
```

```
CARD # LCC    CODE        CARD  10           20          30          40          50
       0301  A5 64             LDA FINT
       0303  0A                ASL A
       0304  0A                ASL A
       0305  85 75             STA TEMP3
       0307  4C 0E C3          JMP RGA
       030A  A5 64      RG9    LDA FINT       ;CHECK IF RAMP/DERAMP TABLE
       030C  85 75             STA TEMP3      ;HAS BEEN COMPLETED
       030E  A5 72      RGA    LDA TIME
       0310  C5 75             CMP TEMP3
       0312  F0 02             BEQ RGM
       0314  10 03             BPL RGB
       0316  4C 28 C2   RGM    JMP RAGO
       0319  A9 DD      RGB    LDA #$DD       ;DISPLAY 'DDDDDD' TO INDICATE
       031B  85 F9             STA $F9        ;COMPLETION OF TABLE
       031D  85 FA             STA $FA
       031F  85 FB             STA $FB
       0321  20 1F 1F   RGC    JSR $1F1F
       0324  4C 21 C3          JMP RGC
       0327             ;**************************************************
       0327             ;SUB. : DIVN
       0327             ;      THIS SUB. DIVIDES A 16-BIT UNSIGNED NO. BY
       0327             ;AN 8-BIT UNSIGNED NO., GIVING AN 8-BIT QUOTIENT
       0327             ;AND AN 8-BIT REMAINDER
       0327             ;**************************************************
       0327  86 75      DIVN   STX TEMP3
       0329  A2 08             LDX #$08
       032B  A5 67             LDA LDIV
       032D  85 6A             STA QUO
       032F  A5 68             LDA MDIV
       0331  06 6A      DI1    ASL QUO
       0333  2A                ROL A
       0334  C5 69             CMP DSOR
       0336  90 04             BCC DI2
       0338  E5 69             SBC DSOR
       033A  E6 6A             INC QUO
       033C  CA         DI2    DEX
       033D  D0 F2             BNE DI1
       033F  85 6B             STA REM
       0341  A6 75             LDX TEMP3
       0343  60                RTS
       0344             ;**************************************************
       0344             ;SUB. : MPLY
       0344             ;      THIS SUB. MULTIPLIES TWO UNSIGNED 8-BIT
       0344             ;BINARY NOS., GIVING A 16-BIT RESULT.
       0344             ;**************************************************
       0344  86 75      MPLY   STX TEMP3
       0346  A9 00             LDA #$00
       0348  85 6E             STA MMPL
       034A  A2 08             LDX #$08
       034C  0A         MP1    ASL A
       034D  26 6E             ROL MMPL
       034F  06 6D             ASL MER
       0351  90 07             BCC MP2
       0353  18                CLC
       0354  65 6C             ADC MAD
```

```
CARD # LOC      CODE        CARD  10        20        30        40        50
  221  0356  9C 02               BCC MP2
  222  0358  E6 6E               INC MMPL
  223  035A  CA          MP2     DEX
  224  035B  DC EF               BNE MP1
  225  035D  85 6F               STA LMPL
  226  035F  A6 75               LDX TEMP3
  227  0361  60                  RTS
  228  0362                      .END
```

STEPPING CONTROL PROGRAM:

Program STEPCON requires the following parameter initializations:

(i) Specify chosen excitation table (sequence) code.

(ii) Specify length of ramping table, i.e., total number of stages in accelerating from the start-stop speed to the operating speed.

(iii) Specify the starting address of the ramping table.

(iv) Specify the ending address of the ramping table.

(v) Specify the number of steps after which the stall test must be performed.

(vi) Specify if the motor should be halted with its phases energized or de-energized.

Program execution is started at address \$200. The program initializes the rotor position and displays 'AAAAAA', at which point program execution is started at address \$300 by depressing the 'ST' key, initializing the address display to \$0300 and depressing the 'GO' key. The motor gets ramped up to its operating speed and continues to operate at this speed until the control switch on the hardware board is switched from '0' to the '1' position. The program then deramps the motor and lights up the LED on the hardware board, which implies that the motor is running at its start-stop speed and can be halted by depressing the 'ST' key.

```
LOC    CODE        CARD  10          20          30          40          50
0000               ;*************************************************
0000               ;STEPPING CONTROL PROGRAM (STEPCON)
0000               ;*************************************************
0000               ;VARIABLE STORAGE ADDRESSES
0000               ;*************************************************
0000                      STEP=$0020   ;STEP CNT. LOCN. FOR STALL TEST
0000                      SECQ=$0021   ;CHOSEN SEQN. CODE
0000                      ENCH=$0022   ;SEQN. END CHECK ADDRESS
0000                      CHEL=$0023   ;STOP CHECK LOCN.
0000                      TALEN=$0024  ;RAMP/DERAMP TABLE LENGTH
0000                      TEM4=$0025   ;TEMP. STORAGE FOR TALEN
0000                      RADN=$0026   ;END ADD. OF RAMP/DERAMP TABLE
0000                      RAUP=$0027   ;START ADD. OF RAMP/DERAMP TABLE
0000                      TEM1=$0028   ;TEMP. RAMP TABLE PNTR.
0000                      TEM2=$0029   ;TEMP. SEQN. TABLE PNTR.
0000                      RATEH=$002A  ;M.S.B. OF #OF STEPS
0000                      RATEL=$002B  ;L.S.B. OF # OF STEPS
0000                      STAL=$002C   ;# OF STEPS TO REPEAT STALL TEST
0000                      TEM3=$002D   ;TEMP. STORAGE LOCN.
0000                      POWR=$002E   ;POWER ON/OFF SPEC. LOCN.
0000               ;THE RAMP/DERAMP TABLE IS STORED STARTING AT $0030
0000               ;*************************************************
0000               ;INITIALIZATIONS
0000               ;*************************************************
0000                      *=$0200
0200   A9 E0       INIT   LDA #$E0     ;INITIALIZE PORT A,BITS(0-4) I/P
0202   8D 01 17           STA $1701    ;AND BITS(5-7) O/P
0205   A9 1F              LDA #$1F     ;INITIALIZE PORT B,BITS(0-4) O/P
0207   8D 03 17           STA $1703    ;AND BITS(5-7) I/P
020A   A9 00              LDA #$00     ;INITIALIZE STEP-COUNT LOCATION
020C   85 23              STA CHEL     ;AND STEP CHECK LOCATION TO ZERO
020E   85 20              STA STEP
0210   8D 00 17           STA $1700    ;INITIALIZE PORT A O/P TO ZERO
0213   A6 21              LDX SECQ     ;O/P FIRST CONTROL WORD TO MOTOR
0215   B5 00              LDA $00,X
0217   8D 02 17           STA $1702
021A   A9 00              LDA #$00     ;INITIALIZE NMI VECTOR
021C   8D FA 17           STA $17FA
021F   A9 1C              LDA #$1C
0221   8D FB 17           STA $17FB
0224   A5 24              LDA TALEN    ;INITIALIZE TEM4 TO R/D TABLE
0226   85 25              STA TEM4     ;LENGTH
0228   A5 21              LDA SECQ     ;COMPUTE CONTROL WORD SEQUENCE END
022A   C9 08              CMP #$08     ;CHECK ADDRESS
022C   F0 0C              BEQ IN1
022E   C9 18              CMP #$18
0230   F0 08              BEQ IN1
0232   18                 CLC
0233   69 04              ADC #$04
0235   85 22              STA ENCH
0237   4C 3F 02           JMP IN2
023A   18          IN1    CLC
023B   69 08              ADC #$08
023D   85 22              STA ENCH
023F   A9 AA       IN2    LDA #$AA     ;DISPLAY "AAAAAA"
```

```
CARD # LOC     CODE          CARD  10          20           30           40           50
       0241    85 F9                STA $F9
       0243    85 FA                STA $FA
       0245    85 FB                STA $FB
       0247    20 1F 1F    IN3      JSR $1F1F
       024A    4C 47 02             JMP IN3
       024D                ;***************************************************
       024D                ;UPERATIONAL
       024D                ;***************************************************
       024D                         *=$0300
       0300    A9 E2       RAM      LDA #$E2       ;RE-INITIALIZE NMI VECTOR
       0302    8D FA 17             STA $17FA
       0305    A9 03                LDA #$03
       0307    8D FB 17             STA $17FB
       030A    A6 27                LDX RAUP       ;LOAD Y-REG. WITH DELAY TIME
       030C    B4 00                LDY $00,X
       030E    B5 01                LDA $01,X      ;LOAD 16-BIT LOCN. RATEH-RATEL
       0310    85 2A                STA RATEH      ;WITH STEP COUNT FOR WHICH ABOVE
       0312    B5 02                LDA $02,X      ;DELAY TIME IS TO BE MAINTAINED
       0314    85 2B                STA RATEL
       0316    86 28                STX TEM1       ;SAVE R/D TABLE POINTER
       0318    A6 21       R1       LDX SECC       ;C/P CONTROL WORD TO MOTOR
       031A    B5 00       R2       LDA $00,X
       031C    8D 02 17             STA $1702
       031F    86 29                STX TEM2       ;SAVE SECN. TABLE POINTER
       0321    A2 14       R3       LDX #$14       ;DELAY GENERATING LOOP
       0323    CA          R4       DEX
       0324    D0 FD                BNE R4
       0326    88                   DEY
       0327    D0 F8                BNE R3
       0329    A6 28                LDX TEM1       ;X-REG. POINTS TO R/D TABLE
       032B    A5 25                LDA TEM4       ;TEST FOR END OF R/D OPERATION
       032D    F0 32                BEQ RH
       032F    A5 2B                LDA RATEL      ;DEC. RATEH-RATEL TO CHECK IF
       0331    38                   SEC            ;SPEED MUST BE CHANGED
       0332    E9 01                SBC #$01
       0334    85 2B                STA RATEL
       0336    A5 2A                LDA RATEH
       0338    E9 00                SBC #$00
       033A    85 2A                STA RATEH
       033C    05 2B                ORA RATEL
       033E    D0 21                BNE RH
       0340    A5 23                LDA CHEL       ;CHECK IF RAMPINGOR DERAMPING
       0342    F0 06                BEQ RL
       0344    CA                   DEX            ;SET R/D POINTER FOR DERAMPING
       0345    CA                   DEX
       0346    CA                   DEX
       0347    4C 4D 03             JMP R5
       034A    E8          RL       INX            SET R/D TABLE POINTER FOR RAMPING
       034B    E8                   INX
       034C    E8                   INX
       034D    C6 25       R5       DEC TEM4       ;DECREMENT R/D TABLE LENGTH COUNTE
       034F    D0 03                BNE R6
       0351    4C 03 03             JMP FIX
       0354    B4 00       R6       LDY $00,X      ;LOAD Y-REG. WITH NEW DELAY TIME
       0356    B5 01                LDA $01,X      ;AND RATEH-RATEL WITH STEP COUNT
```

```
CARD # LOC    CODE         CARD  10          2C         30           40          50
       0358   85 2A               STA RATEH    ;FOR WHICH TO MAINTAIN THIS TIME
       035A   B5 02               LDA $02,X
       035C   85 2B               STA RATEL
       035E   4C 63 03            JMP AN
       0361   B4 00        RH     LDY $00,X    ;LOAD CLC DELAY TIME
       0363   86 28        AN     STX TEM1     ;SAVE R/D TABLE POINTER
       0365   A5 25               LDA TEM4     ;ENSURE R/D OVER BEFORE PERFORMING
       0367   D0 31               BNE R8       ;STALL TEST
       0369   E6 20               INC STEP     ;INC. STALL TEST CNTR AND CHECK
       036B   A5 2C               LDA STAL     ;IF TIME TO PERFORM STALL TEST
       036D   C5 20               CMP STEP
       036F   D0 29               BNE R8
       0371   A9 00               LDA #$00     ;STALL TEST: CLEAR STALL TEST CNTR
       0373   85 20               STA STEP
       0375   AD 00 17            LDA $1700    ;READ IN CNTR. AND MULT. BY 4
       0378   29 0F               AND #$0F
       037A   0A                  ASL A
       037B   0A                  ASL A
       037C   85 2D               STA TEM3
       037E   A9 40               LDA #$40     ;CLR. CNTR. VIA PORT A BIT 6
       0380   8D 00 17            STA $1700
       0383   A5 2D               LDA TEM3     ;CHECK IF EXT. CNTR. COUNT EQUALS
       0385   38                  SEC          ;THE NO. OF STEPS THE MOTOR SHOULD
       0386   E5 2C               SBC STAL     ; HAVE MOVED BY FINDING THE ABS.
       0388   10 07               BPL R7       ;VALUE OF DIFF. AND BRANCHING TO
       038A   38                  SEC          ;STALL ROUTININE IF DIFF. EXCEEDS
       038B   85 2D               STA TEM3     ;A SET LIMIT
       038D   A9 00               LDA #$00
       038F   E5 2D               SBC TEM3
       0391   29 FC        R7     AND #$FC
       0393   D0 54               BNE INTS
       0395   A9 00               LDA #$C0     ;REACTIVATE CNTR.,END OF STALL TES
       0397   8D 00 17            STA $1700
       039A   AD 00 17     R8     LDA $1700    ;STOP CHECK VIA PORT A BIT 4
       039D   29 10               AND #$10
       039F   F0 25               BEQ M7
       03A1   A5 23               LDA CHEL     ;CHECK IF DERAMPING OVER
       03A3   D0 14               BNE M6
       03A5   A5 24               LDA TALEN    ;INITIALIZE DERAMPING PARAMETERS
       03A7   85 25               STA TEM4
       03A9   A5 28               LDA TEM1
       03AB   AA                  TAX
       03AC   B5 01               LDA $01,X
       03AE   85 2A               STA RATEH
       03B0   B5 02               LDA $02,X
       03B2   85 2B               STA RATEL
       03B4   E6 23               INC CHEL
       03B6   4C C6 03            JMP M7
       03B9   A5 25        M6     LDA TEM4     ;O/P A '1' VIA PORT A BIT 5 IF
       03BB   D0 09               BNE M7       ;DERAMPING IS OVER
       03BD   A5 23               LDA CHEL
       03BF   F0 05               BEQ M7
       03C1   A9 40               LDA #$40
       03C3   8D 00 17            STA $1700
       03C6   A6 29        M7     LDX TEM2
```

```
CARD # LOC     CODE         CARD  10          20         30         40         50
  166   03C8   E8                  INX
  167   03C9   E4 22               CPX ENCH
  168   03CB   D0 03               BNE R9
  169   03CD   4C 18 03            JMP R1
  170   03D0   4C 1A 03   R9       JMP R2
  171   03D3   A5 23      FIX      LDA CHEL       ;FIX R/D POINTER FOR RAMPING/
  172   03D5   F0 06               BEQ M1         ;DERAMPING
  173   03D7   E8                  INX
  174   03D8   E8                  INX
  175   03D9   E8                  INX
  176   03DA   4C E0 03            JMP M2
  177   03DD   CA         M1       DEX
  178   03DE   CA                  DEX
  179   03DF   CA                  DEX
  180   03E0   86 28      M2       STX TEM1
  181   03E2   4C 54 03            JMP R6
  182   03E5   A5 2E      STOSU    LDA POWR       ;CHECK IF MOTOR IS TO BE STOPPED
  183   03E7   D0 05               BNE FINI       ;WITH PHASES ON OR OFF.
  184   03E9   A9 00      INTS     LDA #$00       ;STOP MOTOR
  185   03EB   8D 02 17            STA $1702
  186   03EE   4C 00 1C   FINI     JMP $1C00
  187   03F1                       ;**********************************************
  188   03F1                       ; DRIVE SEQUENCE TABLES
  189   03F1                       ;**********************************************
  190   03F1                                *=$0000
  191   0000   09                           .BYTE $09,$0A,$06,$05,$08,$02,$04,$01,$09
  191   0001   0A
  191   0002   06
  191   0003   05
  191   0004   08
  191   0005   02
  191   0006   04
  191   0007   01
  191   0008   09
  192   0009   08                           .BYTE $08,$0A,$02,$06,$04,$05,$01
  192   000A   0A
  192   000B   02
  192   000C   06
  192   000D   04
  192   000E   05
  192   000F   01
  193   0010   05                           .BYTE $05,$06,$0A,$09,$01,$04,$02,$08,$01
  193   0011   06
  193   0012   0A
  193   0013   09
  193   0014   01
  193   0015   04
  193   0016   02
  193   0017   08
  193   0018   01
  194   0019   05                           .BYTE $05,$04,$06,$02,$0A,$08,$09
  194   001A   04
  194   001B   06
  194   001C   02
  194   001D   0A
```

```
CARD # LOC      CODE        CARD  10         20         30         40         50
  194  001E  08
  194  001F  C9
  195  0020                    .END
```

APPENDIX : COMPUTER SIMULATION

A computer simulation was carried out for the resistive and chopper drives. The supply voltage used in the simulation was much higher than could be applied to the test circuit, thus permitting high speed simulation.

The programs were written in CSMP and a listing of the programs used, along with plots of the results obtained, has been included in this appendix. In both drive circuit simulations the charging and discharging time constants are different, the discharging time constant being much smaller than the charging time constant. The plotted results show satisfactory results for both the drives under the simulated conditions. The simulation conditions have been included in the program listings.

```
$$$CONTINUOUS SYSTEM MODELING PROGRAM  III   V1M3   TRANSLATOR OUTPUT$$$

*
*PROGRAM: RESIM
*       THIS PROGRAM SIMULATES THE RESPONSE OF A STEP MOTOR PHASE
*WHEN DRIVEN BY A RESISTIVE POWER DRIVE. THE INPUT VOLTAGE(VI) IS A
*PULSE TRAIN WITH FREQUENCY (1/(TON+TOF)). TON IS THE PULSE-ON
*TIME (WITHIN A PERIOD) AND TOF IS THE PULSE-OFF TIME, BOTH SPECIFIED
*IN SECONDS.ALL OTHER VARIABLES ARE DESCRIBED BELOW.
*  L=PHASE INDUCTANCE(HENRIES).
*  R=TOTAL SERIES RESIST. IN THE PHASE(OHMS).
*  SR=REQUIRED STEPPING RATE(STEPS PER SECOND)
*  RB=FREE-WHEELING RESISTANCE(OHMS).
*FOR 1-PHASE ON MODE: TOF=3*TON
*FOR 2-PHASE ON MODE: TOF=2*TON
*FOR BOTH MODES : TON+TOF=4/SR
*
INITIAL
CONSTANT R=20.0,L=0.012,SR=200,VL=12.0,RB=500
COR=1/R
TON=1/SR
TOF=3/SR
TOT=TON+TOF
DYNAMIC
X1=IMPULS(0.0,TOT)
V=VL*PULSE(TON,X1)
       NOSORT
        IF (V.EQ.0.0) GO TO 10
       TAU=L/R
       GO TO 20
 10    TAU=L/RB
 20    CONTINUE
       SORT
IT=REALPL(0.0,TAU,V)
I=IT*COR
TERMINAL
TIMER FINTIM=0.015,OUTDEL=0.00001
OUTPUT TIME,I,V
PAGE XYPLOT
LABEL CURRENT/APPLIED VOLT. VS. TIME
END
STOP
```

```
$$$CONTINUOUS SYSTEM MODELING PROGRAM  III   V1M3   TRANSLATOR OUTPUT$$$

*
*PROGRAM: RESIM
*       THIS PROGRAM SIMULATES THE RESPONSE OF A STEP MOTOR PHASE
*WHEN DRIVEN BY A RESISTIVE POWER DRIVE. THE INPUT VOLTAGE(VI) IS A
*PULSE TRAIN WITH FREQUENCY (1/(TON+TOF)). TON IS THE PULSE-ON
*TIME (WITHIN A PERIOD) AND TOF IS THE PULSE-OFF TIME, BOTH SPECIFIED
*IN SECONDS.ALL OTHER VARIABLES ARE DESCRIBED BELOW.
*  L=PHASE INDUCTANCE(HENRIES).
*  R=TOTAL SERIES RESIST. IN THE PHASE(OHMS).
*  SP=REQUIRED STEPPING RATE(STEPS PER SECOND)
*  RB=FREE-WHEELING RESISTANCE(OHMS).
*FOR 1-PHASE ON MODE: TOF=3*TON
*FOR 2-PHASE ON MODE: TOF=2*TON
*FOR BOTH MODES : TON+TOF=4/SP
*
INITIAL
CONSTANT R=120.0,L=0.015,SP=500,VL=72.0,RB=500.0
CCR=1/4
TON=1/SP
TOF=3/SP
TOT=TON+TOF
DYNAMIC
X1=IMPULS(0.0,TOT)
V=VL*PULSE(TON,X1)
       NOSORT
        IF (V.EQ.0.0) GO TO 10
       TAU=L/R
       GO TO 20
 10    TAU=L/RB
 20    CONTINUE
       SORT
IT=REALPL(0.0,TAU,V)
I=IT*CCR
TERMINAL
TIMER FINTIM=0.015,OUTDEL=0.00001
OUTPUT TIME,I,V
PAGE XYPLOT
LABEL CURRENT/APPLIED VOLT. VS. TIME
END
STOP
```

```
$$$CONTINUOUS SYSTEM MODELING PROGRAM   III   MIMS   TRANSLATOR OUTPUT$$$

*
*PROGRAM: CHOPSIM
*       PROGRAM CHOPSIM SIMULATES THE CHOPPER DRIVE FOR A SINGLE
*PHASE OF THE STEP MOTOR. ALL VARIABLES ARE THE SAME AS THOSE DEFINED
*FOR PROGRAM RESIM. THE MOTOR'S RATED CURRENT IS 0.6 AMPS, HENCE THE
*UPPER LIMIT FOR THE PHASE CURRENT HAS BEEN SET EQUAL TO 0.66 AMPS
*AND THE LOWER LIMIT EQUAL TO 0.54 AMPS.
*
INITIAL
CONSTANT R=20,L=0.010,SR=500,VL=50.0,RR=500,CNT=0.0
COR=1/R
TON=1/SR
TOF=3/SR
TOT=TON+TOF
DYNAMIC
X1=IMPULS(0.0,TOT)
V=VL*PULSE(TON,X1)
      NOSORT
       IF (I.LT.0.66) GO TO 3
       CNT=1.0
 2     IF (I.LT.0.54) GO TO 4
       V1=0.0
       GO TO 6
 3     IF (CNT.NE.0.0) GO TO 2
 4     CNT=0.0
       V1=V
 6     CONTINUE
       IF (V1.EQ.0.0) GO TO 10
      TAU=L/R
      GO TO 20
 10   TAU=L/RR
 20   CONTINUE
      SORT
IT=REALPL(0.0,TAU,V1)
I=IT*COR
TERMINAL
TIMER FINTIM=0.0004,OUTDEL=0.00001
OUTPUT TIME,I,V
PAGE XYPLOT
LABEL CURRENT/APPLIED VOLT. VS. TIME
END
STOP
```

BIBLIOGRAPHY

1) Kuo B.C., "Theory and Applications of Step Motors", West Publishing Co.,1974.

2) Athani V.V.,Mundhada J.G., "High-Performance Translator for a High-Torque PM Stepping Motor", IEEE Transactions (IECI), Vol.25 No.4, Nov 1978.

3) Fitzgerald A.E., Kingsley C., Kusko A. "Electric Machinery", McGraw-Hill Kogakusha, 1971.

4) Page W.D., Singh G., Kuo B.C., "Application of a computer control to a Study of Open-loop acceleration of Step Motors", IEEE Transactions on Industrial Electronics and Control Instrumentation, Vol.22 No.2, May 1975.

5) Millman J., Halkias C.C., "Integrated Electronics", McGraw-Hill Kogakusha 1972.

6) Toacse G., "Translator Design of the Stepping Motor", IEEE Transactions on Industrial Electronics and Control Instrumentation, Vol.22 No.2, May 1975.

7) Cantoni A.," A Stepper motor system for Profile following and Cutting", IEEE Transaction on Industrial Electronics and Control Instrumentation, Vol. 21, No. 2, May 1974.

8) Peatman J.B., "Microcomputer based design", McGraw-Hill Kogakusha, 1977.

9) Speckhart F.H., Green W.L., "A Guide to Using CSMP", Prentice Hall, 1976.

10) Hayt H., Kemmerly J.E., "Engineering Circuit Analysis", McGraw-Hill Kogakusha, 1971.

11) Hilburn J.L., Julich P.M., "Microcomputers/Microprocessors : Hardware, Software and Applications", Prentice Hall India, 1979.

12) KIM-1 User Manual, MOS Technology Inc., 1976.

13) KIM-1 Programming Manual, MOS Technology Inc., 1976.

14) Texas Instruments, "The TTL Data Book for Design Engineers", 1976.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Prof. W.W.Koepsel for all his help and encouragement in the course of completing this study.

Thanks are also due to my wife, Corrinne, for her patience and invaluable help in editing and typing the final copy of this thesis. I would also like to thank my friend Ken Albin for the encouragement he gave me and his friendship.

Finally, I would like to express my sincere gratitude to my parents for making all of this possible.

MICROPROCESSOR BASED STEP MOTOR CONTROLLER

by

NEERAJ MAGOTRA

B.Tech. Indian Institute of Technology, Bombay (India)

1980

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

College of Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1982

ABSTRACT

The rapid growth in recent years in the field of digital electronics has spurred a growing interest in devices which interface between the digital and analog worlds. The step motor, the subject of this study, is one such device which has found numerous applications in the fields of robotics, industrial control and computer peripheral equipment.

Two different kinds of step motor drives were tested out, within equipment limitations, in the course of this study. The first of these drives, the resistive drive, improved the stepping rate of the motor but at a cost of high energy (I^2R) losses. The second drive, a form of chopper control, further improved the stepping speeds attainable by the motor without entailing high energy losses since it eliminates the need for a series resistance being connected to the motor phase windings.

The study also dwelled on the software aspects of controlling the step motor. Several programs were developed for the microprocessor based controller to demonstrate the flexibility of software control. The first of these programs (POSCON) controlled the motor in its positioning mode of operation, a mode of control used frequently in machine tools and robotics. The second program (RAMGEN) solves a linear velocity equation to develop a linear acceleration/decceleration table used by program STEPCON, which controls the motor in the slewing (continous running) mode of operation.

Most hardware controllers permit linear or exponential acceleration of the step motor. However in a study done by Page,Singh and Kuo (4),several different acceleration schemes were tried out and the results seemed to suggest that other schemes such as parabolic and different forms of parabolic and exponential acceleration schemes could result in performance improvement. This is where the flexibility of software control is a decided advantage over hardware control.

By simply storing several different software routines to generate acceleration/ decceleration tables, the user could implement any one of the acceleration schemes with a minimum of trouble, using the same drive software (STEPCON) and hardware. Using one of the new "piggy-back" microprocessors, the drive and control software could be stored in on-board memory, while the table (acceleration) generating programs could be stored in the piggy-back memory chip. This would virtually comprise a single chip step motor controller far more powerful than any of the single chip controllers presently available.

Program STEPCON uses the acceleration table generated by RAMGEN to accelerate the motor from a safe starting speed to its operating speed. When the motor is to be turned off, the program deccelerates the motor to a safe stopping speed before bringing it to a halt.

The study served to highlight the versatility of a microprocessor based step motor controller. An idea which seems ever more attractive with the growing availability of low-cost single chip microprocessors which are virtual mini-computers.